ENGR9700: Masters Thesis

The Design and Development of a Sports Agility Tester

Reuben Smith

FAN: smit1461
Student ID: 2121143

Associate Supervisor

Sam Elliott

Academic Supervisor

David Hobbs & Mark Taylor

Submitted to the College of Science and Engineering in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Mechanical) (Honours) / Master of Engineering (Biomedical) at Flinders University, Adelaide, Australia.

Flinders UNIVERSITY
College of Science and Engineering

SHAPE Research Centre
SPORT, HEALTH AND PHYSICAL EDUCATION

Examination Submission Date: 22nd October 2018

Declaration

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Signature has been removed for Open Access Version.

____

Reuben Smith

Date: 11th January 2019

Acknowledgements

I would like to take this opportunity to give my thanks to several individuals for the successful outcomes of this project and for the ongoing support and assistance throughout:

My associate supervisor, Sam Elliott, for identifying the current gap in the market, coming up with the initial idea and for his enthusiasm, positivity and ongoing support throughout the project.

My academic supervisors, David Hobbs and Mark Taylor, for overlooking my progress, providing feedback, making suggestions and giving me assistance.

I would like to thank the Engineering Technical Services Team, especially Craig Dawson, who provided me with support, which made the research, design and development process a whole lot smoother.

Finally, I would like to thank the support from my friends and family, who helped me continue to push through during the most difficult times of this project.

Executive Summary

Performance testing protocols are a widely accepted way of evaluating the ability and skills athletes possess in sports clubs, sports science and physiology by coaches, trainers, physical educators and conditioning specialists. Through performance testing, athletes can be monitored, with their progress tracked for improvements or downfalls, as well as use for talent identification, differentiation between higher-skilled players from their lesser-skilled counterparts, and even usefulness in deciding best positions of play. The objective of this thesis was to design and develop a fully-functional system, capable of testing an athlete's agility with a primary focus for athletes in Australian rules football, but with capability of implementation in other sports such as netball, basketball or rugby.

Currently implemented agility tests, such as the Illinois agility, AFL agility, 5-0-5, T-test, Pro-agility and 3-cone drill, do not incorporate the perceptual and decision-making attributes of the skill as they involve running around objects only; thus, proving to be tests of pre-planned agility only. The proposed test conducted through the designed system differs from these currently implemented agility tests, as it evaluates agility performance through the athlete reacting to a series of random individual stimuli dispersed over a designated testing zone area. The athlete must deactivate the specific marker which presented the stimuli, thereby activating another component successively. Therefore, the test incorporates perceptual and decision-making factors, as well as physical components of strength, power, balance and coordination to sprint short-distances and make sudden change-of-directions to each stimulus. Additionally, sport-specific equipment (such as an AFL football, netball or basketball) can be incorporated to add a sport specific ball-handling element to the test which contributes to deactivation of the marker. Agility is inferred by evaluating the time taken to complete the test. The proposed agility testing system differs from commercially available products in the fact that it can incorporate sport-specific equipment as a stand-alone product and remain standardised, with future work planning to incorporate the option of programming custom tests for training regimes. The system, for the first time, boasts the capability of self-measuring the componentry layout for test replication and standardisation which no commercially available product can achieve.

The proposed agility testing system was designed and developed from the ground up. By employing a logical and systematic engineering design process, disciplines of mechanical, electrical and software engineering were brought together to bring what started as an idea, into an innovative, tangible product. The system consists of one master component (Figure 1a), containing an assembly that rotates a Light Detection and Ranging (LiDAR) scanner and low power laser at precise angles to assist positioning of six slave components (one of which is shown in Figure 1b). These components consist of two LED matrices and a speaker, so that both visual and auditory stimuli can be presented to the athlete. The LED matrices are positioned such that there is a wide viewing angle and can display any random alphanumeric character and colour. This component was sculpted so sport-specific equipment (such as game-ball) could be positioned on top of it with a sensor detecting when it has been removed. The system is capable of changing difficulty modes to increase or decrease the testing area, as well as being able to receive commands from either a laptop or smartphone using Bluetooth® technology.

Figure 1: Master and slave component prototypes

Although full functionality was not able to be implemented due to significant time constraints, the proposed agility testing system was still able to achieve 92.3% of first order specifications and a total of 86.3% of both first (essential) and second (desirable) order specifications. Thus, the proof-of-concept system (Figure 2) was deemed a success, acting as the first stepping stone in developing a commercial product to introduce into the sporting community, which may become a new standardised way of evaluating athlete agility performance across sporting clubs for use with a variety of field and court sports and athletic abilities that may be used in testing batteries, combines and testing regimes.

Figure 2: Integrated system with master and all slave components with integrated sport-specific equipment

The author would like to take this opportunity to mention that after submission of the thesis, further development and refinement of the system was undertaken, solving all issues which had impeded the system from achieving a complete test run-through. Thus, the system is now fully functional and has the capability of running a complete desired test. Further details have been mentioned in Section 9: Erratum Appendix.

Table of Contents

List of Figures

List of Figures in Appendix

List of Tables

List of Tables in Appendix

List of Equations

List of Definitions, Acronyms and Abbreviations

Term Used	Definition
Sport Associated	
AFL	Australian Football League
USAFL	United States Australian Football League
SANFL	South Australian National Football League
NBL	National Basketball League
NBA	National Basketball Association
NRL	National Rugby League
NCAA	National Collegiate Athletic Association
Testing battery/combine	A set of tests completed on an athlete in a short period of time that measure a variety of aerobic and anaerobic sporting attributes and skills.
CODS	Change-of-direction speed
RAT	Reactive Agility Test
ANOVA	Analysis of variance
ICC	Intraclass coefficient
Project Associated	
SHAPE Research Centre	Sport, Health and Physical Education Research Centre
Device/System	Refers to all components that make up the Sports Agility Tester.
Marker/Signal/Slave	The system is composed of individual devices defined in this project as "markers" or "signals", which will aim to hold the ball, present audio and visual stimulus and house electronics.
Master	The primary component of the system, which completes most data generation and collection, test adjustment, assists with test set-up and has the role of user interface connection. All slaves connect with the Master and listen to its instructions.
Testing zone	Is the area of space that is occupied between the markers where the athlete will perform the test; the athlete may exit this area at any point whilst undertaking the test without penalty.
QFD	Quality-function deployment (method)
CAD	Computed-Aided Design

HOQ	House of Quality (chart)
LED	Light Emitting Diode
LCD	Liquid crystal display
I/O Pin	Input/Output Pin (Referring to Microcontroller Pin)
FSD, SISD	14- and 16- segment LED Display
PWM	Pulse width modulation
LiDAR	Light detection and ranging
TX	Transmit
RX	Receive
RFID	Radio frequency identification
RF	Radio frequency
IR	Infrared
IP Rating	Defines the international standards for classification of protection against the intrusion of solid bodies or water into an electrical enclosure.
PVC	Polyvinyl chloride
PC	Polycarbonate
ALS	Ambient light sensor
HSV	Hue, saturation, value
RGB	Red, green, blue
DC	Direct current
VCC/VIN/V+	Power pin in electronic hardware
GND	Ground pin in electronic hardware
VREF	Reference voltage value
LSB	Lowest significant byte
ISR	Interrupt service routine

1. Introduction

1.1. Project Background

1.1.1. Athlete Performance Testing

Performance testing protocols are a widely accepted way of evaluating the ability and skills athletes possess in sports clubs, sports science and physiology by coaches, trainers, physical educators and conditioning specialists (McGee and Burkett, 2003; Pauole et al., 2000). Through performance testing, athletes can be monitored with their progress tracked for improvements or downfalls, as well as use for talent identification, differentiation between higher-skilled players from their lesser-skilled counterparts, and even usefulness in deciding best positions of play (Chu and Vermeil, 1983; McGee and Burkett, 2003; Pauole et al., 2000; Reilly et al., 2000). Agility is an ability that has had a lot of attention in sports and research, with interest in the evaluation of the skill in both field and court sports such as Australian rules football (Chalmers and Magarey, 2016; Henry et al., 2011, 2013; Mooney et al., 2011; Pyne et al., 2005; Robertson et al., 2015; Veale et al., 2010; Woods et al., 2015; Young et al., 2011; Young and Murray, 2017), American football (rugby) (Gabbett et al., 2009, 2008; Gabbett and Benton, 2009; Hoffman, 2006; Meir et al., 2001; Serpell et al., 2010; Till et al., 2011, 2013, 2015), soccer (Bidaurrazaga-Letona et al., 2015; Born et al., 2016; Brahim et al., 2013; Chaalali et al., 2016; Fiorilli et al., 2017; Hachana et al., 2014; Hoffman, 2006; Kaplan et al., 2009; Kutlu et al., 2012, 2017; Leon-Carlyle et al., 2012; Little and Williams, 2005; Matlák et al., 2016; Raven et al., 1976; Rouissi et al., 2016; Sporis et al., 2010; Trecroci et al., 2016), basketball (Chaouachi et al., 2009; Delextrat and Cohen, 2009; Lockie et al., 2014; Scanlan et al., 2013; Sekulic et al., 2017; Spiteri et al., 2014), netball (Barber et al., 2016; Farrow, 2010; Farrow et al., 2005), tennis (Cooke et al., 2011; Farrow and Abernethy, 2002; Parsons and Jones, 1998; Tenenbaum et al., 1996), hockey (Keogh et al., 2003; Morland et al., 2013) and handball (Chaalali et al., 2016; Hermassi et al., 2011; Spasic et al., 2015). Testing agility is a difficult concept to grasp, as so many currently implemented, standardised tests claiming to assess agility consist of a multitude of different elements in them; where implementation of one test may be significantly different to another test.

1.1.2. Defining Agility

The reason why so many currently implemented agility tests vary significantly from one another could be due to the ambiguity underlying the construct. The definition and classification of agility, to this day, remains an ambiguous term used in sports science, with no consistency throughout current or past literature. With an ever-growing number of publications acknowledging this ambiguity exists (Chelladurai, 1976; Cooke et al., 2011; Coulson and Archer, 2009; Haff and Triplett, 2016; Liefeith et al., 2018; Lloyd et al., 2009; Paul et al., 2016; Sheppard and Young, 2006; Stewart et al., 2014; Young et al., 2015), it is bringing the sports science community one step closer to coming to a common agreement on the subject. Sheppard and Young (2006) suggested that agility involves a multitude of components involving both physical, perceptual and decision-making factors, proposing a comprehensive definition that agility is “a rapid whole-body movement with change of velocity or direction in response to a stimulus” (Sheppard and Young, 2006, p.922), which has been accepted by a large amount of literature (Benvenuti et al., 2010; Born et al., 2017; Coulson and Archer, 2009; Haff and Triplett, 2016; Jeffreys, 2011; Lockie et al., 2014; Scanlan et al., 2013; Serpell et al., 2011; Spiteri et al., 2012; Stewart et al., 2014; Tanner et al., 2013; Young and Rogers, 2016). Through a thorough investigation of independent definitions of agility in the past 40 years in literature, the author was able to present a definition based on the frequency of keywords: *the ability of rapidly (or quickly) changing position (or direction) of the whole-body due to a stimulus with speed/velocity;* the definition aligned significantly with the definition proposed by Sheppard and Young (2006).

1.1.3. Limitations of Standardised Agility Tests

With an accepted definition and classification of agility presented in this thesis, it was now clear to see that agility tests presently conducted do not truly represent the skill. Agility tests such as the Illinois Agility, AFL Agility Run, 5-0-5, T-Test (T-Drill), Pro-Agility (5-10-5 Shuttle) and the 3-Cone Drill (L-Run) all only test the physical component of agility, involving pre-planned behaviour of sprints, change-of-direction speed (CODS) and navigation through a series of cones to predefined locations on a known path. These aspects therefore show these tests involve closed skills only, which are defined as movements in which a person knows exactly what is expected from them; some form of pre-programmed/pre-planned movement that does not involve a response to a stimulus (Cox, 2011; Verstegen et al., 2001). These tests therefore do not accurately represent the open skills associated with agility, which are used when a person must react to a stimulus from the surrounding environment perceived via their sensory input systems, in which the movement contains some form of ambiguity that is not automated or rehearsed (Cox, 2011; Verstegen et al., 2001). Although these tests do not accurately represent all components of agility, they are widely accepted protocols in testing batteries and draft combines conducted by popular sporting institutions such as the Australian Football League (AFL) (Chalmers and Magarey, 2016; Pyne et al., 2005; Robertson et al., 2015; Tanner et al., 2013; Woods et al., 2015) and the National Football League (NFL) (Hoffman, 2006; McGee and Burkett, 2003; Sierer et al., 2005).

1.1.4. Reactive Agility Tests

There has been a lot of emerging literature that has aimed to include the perceptual and decision-making factors of agility within tests known as Reactive Agility Tests (RATs). Traditional RATs incorporate a single straight sprint towards a 'Y' shaped fork, where the athlete is presented with a light, video or human stimuli, which indicates whether they need to traverse 45^{o} to the left or right. This factor is key in implementation of the athlete using perceptual and decision-making factors of agility, as well as being able to assess the physical components of strength, power, balance and change of direction speed from quickly responding to the stimulus. Inglis and Bird (2016) undertook a comprehensive review of RATs presented in literature, showing findings that RATs are more reliable and valid in assessing agility compared with traditional pre-planned and light agility drills. Another similarly extensive review on RATs by Paul et al. (2016) showed a high reliability and validity in RATs. Several studies have implicated similarly positive results for RATs, proving both reliability and validity (Farrow et al., 2005; Serpell et al., 2010; Veale et al., 2010), as well as differentiation between higher-skilled athletes compared to their lesser-skilled counterparts (Farrow et al., 2005; Gabbett and Benton, 2009; Henry et al., 2011; Lockie et al., 2014; Morland et al., 2013; Veale et al., 2010; Young et al., 2011). Many of the studies concluded that the performance differences in the RATs was attributed to the differences in the skill of perceptual and decision-making components of reactive agility (Farrow et al., 2005; Gabbett and Benton, 2009; Inglis and Bird, 2016; Scanlan et al., 2013; Serpell et al., 2010).

1.1.5. Limitations of RATs

Despite the incorporation of perceptual and decision-making factors through RATs with promising results, there exists several major limitations; they are not standardised, they require a lot of expensive equipment to perform and they are not able to test some physical components of agility that simple pre-planned tests evaluate. Among the literature that implemented a traditional 'Y' shaped RAT or slightly modified version (Farrow et al., 2005; Fiorilli et al., 2017; Gabbett et al., 2008; Henry et al., 2011; Lockie et al., 2014; Morland et al., 2013; Scanlan et al., 2013; Sekulic et al., 2017; Serpell et al., 2010; Spiteri et al., 2014; Veale et al., 2010; Young et al., 2011), only two of the tests conducted had the same testing protocol (number of decision-making components and distances ran between each stage) (Fiorilli et al., 2017; Lockie et al., 2014). All the other tests had different sprint distances, including up to the point where a change-of-direction is required and the remaining sprint thereafter. Some authors also made further modifications to the test to include an extra reactive component (Spiteri et al., 2014; Veale et al., 2010), adding a side-stepping motion (Farrow et al., 2005) or adding an additional running element at the end (Veale et al., 2010). The variations in each of the RATs employed by these authors emphasised that although the RATs were a viable method for differentiating athletic performance, there was no standard that could be used to compare the results between athletes in various clubs or sporting institutions. Additionally, the test usually required a lot of equipment, including timing gates, force sensing plates (or alternative sensing means), projectors, and some form of stimulus (visual light display, video or in some cases a human stimuli). The traditional RATs also only incorporate one 45^{o} change-in-direction, which means that the test does not incorporate a complete 180^{o} change-in-direction like some pre-planned tests integrate (the Illinois Agility Test, Pro-Agility or 5-0-5).

1.2. Proposal for a New Agility Test

With a clear gap in the research and in the market, this therefore presented the idea that if a system could be designed and developed, which is able to test all components of agility (including both physical, perceptual and decision-making factors) whilst incorporating sport-specific equipment (such as a game ball) while also remaining standardised, then it would be of great interest to sporting institutions and clubs, as well as researchers and academics. Thus, this thesis project had the aim to design and develop a novel system capable of testing all components of an athlete’s agility whilst incorporating use of sport-specific equipment and maintaining a form of standardisation.

1.3. Idea Conception

To further understand the significance of this project, it is important to recognise how this project emerged by going back to the roots of the conception of the idea, which was formulated by Dr. Sam Elliott.

The SHAPE Research Centre (Flinders University, Adelaide, Australia) is a multi-disciplinary cluster of academic researchers in Sport, Health, Performance, Exercise and Physical Education (SHAPE). The SHAPE Research Centre is responsible for several High Performance Programs (HPPs), which are delivered to numerous organisations including the South Australian National Football League (SANFL) Womens Talent Search, the South Adelaide Football Club's Development Academy, the West Adelaide Football Club's Development Academy and the Contax Netball Club and Surf Lifesaving South Australia (SHAPE Research Centre, 2018). One of the active members of the team is Dr. Sam Elliott, who is an early career researcher and lecturer in Sport, Health and Physical Activity in the College of Education, Psychology and Social Work at Flinders University. By playing an active role in consultancy and advisory capacities with the Football Federation Australia, Sport Australia and the Australian Council for Health, Physical Education and Recreation, Dr. Elliott has a wealth of knowledge and experience within the field of Sport, Health and Physical Education. He experiences first hand interactions with athletes as well as testing protocols implemented. As a result, he was able to identify a problem with currently implemented tests, that they require pre-planned agility only. He also observed how no standardised test incorporated sport-specific equipment such as the game ball, recognising the current gap in the market associated with agility testing protocols. Thus, the conception for the idea of designing and developing a new sports agility testing system was proposed, which SHAPE wishes to incorporate into their program so that it may be enhanced.

The summary proposal for the project offered by SHAPE (Figure A- 1) is presented in Appendix A.

1.4. Competing Products

A thorough investigation and market analysis was undertaken into competing products currently on the market, which were able to test an athlete's agility. Of the products analysed, all systems were found to be able to function indoors and outdoors, were wireless, could measure time, provide real-time data feedback post-test, were usable in a variety of sports, included customisable tests and generally had a precision accuracy of one hundredth or one thousandth of a second (0.01 and 0.001 seconds, respectively).

The systems that appeared to be of the highest threat included the FITLIGHT Trainer™ (FITLIGHT Sports Corp., Ontario, Canada), the Wireless Training Timer SEM (WittySEM) (Microgate Corporation, Bolzano, Italy), the SMARTfit® Strike Pods (Smartfit Inc., Camarillo, California) and photocell timing gate incorporated systems; SmartSpeed Pro (Fusion Sport, Brisbane, Australia) and SpeedLight (Swift Performance, Brisbane, Australia). All these systems were considered a high threat as they can present a stimulus to the athlete, enabling use of reactive agility tests and training. Furthermore, the SMARTfit® and Witty SEM were able to display a variety of characters, which added to the randomness and unpredictability of the test.

Another system that was of considerable threat was the Freelap Timing System (Freelap SA, Fleurier, Switzerland), as well as traditional photocell timing gate systems such as the TC Timing System (Brower Timing Systems, Draper, UT), Wireless Timing Network (WTN) (ALGE-TIMING GmbH, Vienna, Austria) or the Wireless Race Timing System (TAG Heuer, La-Chaux-de-Fonds, Switzerland). A major limitation to these devices was that being primarily timing systems, these products were not able to implement reactive agility training independently; they required external equipment to do so.

1.4.1. Limitations of Competing Products

It was found the major gap in the market was incorporating sport-specific equipment (such as a game ball) with the system itself. Additionally, if using sport-specific equipment alongside the system, there was no form of standardisation of the test that could be applied consistently across sporting clubs or institutions. Another significant finding was that although some systems provided guided assistance in setting-up, such as aligning timing gates together, no system was able to assist in the layout set-up of the test. That is, no test could assist the user in knowing where they needed to place their devices for a given test; the distances and angles needed to be measured independent of the system. Another understood limitation was that each system needed some form of accessory, such as a tripod or mount, which usually needed to be bought separately.

A more detailed explanation of the findings is presented in Section 2.4: Competitive Market Analysis.

1.5. Project Scope & Constraints

Establishing project scope and constraints in the early stages of the project was important so that it could be understood what would be achievable for this project and what would not, such that a set of expectations could be established.

It was understood early in the development phase of this project that a lot of work would be required to design and develop a system that can test all components of an athlete's agility. With the project to be completed in eight months, significant time constraints existed, which limited the refinement of the product. As a result, a set of essential and desirable features for the product were established in the initial kick-off meeting. The desirable features were not required to be completed in this project but if time permitted, they were pursued upon; generally, they were accepted that implementation would be incorporated in future work.

Although no official budget was agreed upon, development of the system would need to be implemented through contributions from the College of Science and Engineering Thesis Budget ($600), Dr. Elliott's staff account ($400) and personal contributions made by the author ($600); a total contribution of $1,600. As a result, thorough investigation and research needed to be conducted to carefully select components and materials that reflected the budget. Through both time and budget constraints, it was agreed upon that the system developed would be in the form of a 'proof-of-concept' prototype.

A strong belief existed between those associated with the project; that the proposed system to be designed and developed would have significant commercial appeal and value, since it was something that had not been implemented before. With the possibility of commercial value in the product, it was in the best interest to not publicly display or disclose any portion of the project with the public. As a result, the author and all associated parties entered a Contractual Agreement regarding intellectual property (IP) and confidentiality. In addition, the final seminars were held behind closed doors, with any external judges (individuals who are not Flinders University staff) being required to sign Non-Disclosure Agreements (NDAs) with Flinders University prior to assessing and judging the author's work, so that confidential information regarding the product was maintained. A private working space was granted for the author to work at in the Flinders University Medical Device Partnering Program (MDPP) laboratory, which is one of the few rooms that are isolated from the public, so that the work would remain confidential.

1.6. Project Aims & Objectives

The primary aim of this project was to design and develop a novel system capable of testing all components of an athlete's agility, primarily focused for athletes in Australian rules football, but with the possibility to make it diverse, such that it had compatibility with other sports such as netball, basketball or rugby. By discussing with Dr. Elliott, his thoughts and ideas on the system regarding what was essential to include and what was desirable, a clear understanding of the system was able to be made, so that the primary aim could be broken down into a set of aims and objectives. To measure the success of the project quantitatively, the aims, objectives and design requirements could be reflected upon by comparing the system against a given specification criteria.

It was established through an initial kick-off meeting and several subsequent meetings, that the system would consist of a minimum of four devices (markers), which would be positioned in a specific layout over a designated testing zone area. A marker would activate by presenting an audio and visual stimulus to the athlete to indicate they need to remove sport-specific equipment (such as a game ball) from it. The athlete would need to react as quickly as possible to the stimulus, sprinting towards it and removing the sport-specific equipment; thereby deactivating it. Upon deactivation, this would then cause a subsequent activation of another marker. This process would be repeated until all sport-specific equipment had been removed from all markers. It was essential that the sequence the markers activated would be random such that it was not possible for the athlete to anticipate which marker would activate next. Through this, the test would be able to evaluate all components of agility; the athlete would require perceptual cognitive factors to anticipate and visually scan for the next activated marker, then use decision-making when presented with the random stimuli to initiate execution of a physical response. This response would involve a subsequent change-of-direction speed, followed by a burst of rapid acceleration involving strength, power, balance and coordination to sprint in a straight line for a short-distance to the next marker, rapidly decelerating and accurately removing the sport-specific equipment to deactivate it. By measuring the time taken for an athlete to complete the test, this metric would be able to be used to infer a relative evaluation of skill of agility the athlete possesses, as a more agile individual would be expected to complete the test faster.

To assist in developing a clear path to follow and so that the primary aim of the project could be achieved, it was necessary to break it down into a set of definitive aims:

- Obtain a thorough understanding of athlete performance testing; why it is conducted and how to evaluate the results
- Attain a comprehensive understanding of the concept of agility and how it is tested
- Develop an in-depth understanding into the concept of test reliability and validity
- Identify and follow an appropriate engineering design process
- Establish the needs, design requirements and associated design specifications corresponding to the project
- Design and develop a prototype system, which meets the primary aim of testing all components of an athlete's agility.

The aims of the project could be accomplished by establishing a clear set of objectives to aspire to achieve; these were identified as follows:

- Investigate and understand currently implemented standardised agility tests.
- Undertake a competitive market analysis, identifying any commercially available threats to the proposed solution, as well as their advantages and limitations.
- Consult with Dr. Elliot and academic supervisors on the essential and desired features of the system to produce a corresponding set of design requirements.
- Translate the set of design requirements into a corresponding set of engineering specifications with target metrics that can be quantitatively measured against.
- Using design requirements and engineering specifications, ideate and conceptualise viable solutions, evaluating the alternatives and making refinements as required.
- Use Computer-Aided Design (CAD) to develop the final evaluated concept.
- Develop and construct the mechanical design of the system.
- Develop, assemble and program the electronic hardware componentry interface.
- Integrate the electrical, mechanical and software design into a tangible product
- Test and evaluate the system against the set of engineering specifications to determine the success of the system.
- Document all design detail including CAD assembly and part drawings, electronic schematics and software code.
- Understand and acknowledge the limitations with the final prototype system.
- Recognise and document all future work for additional improvements and refinement of the system.

1.7. Thesis Outline

This thesis documents the processes taken to achieve the primary aim of designing and developing a system capable of testing an athlete's agility. A review of literature was first undertaken to substantiate the importance for the proposal of designing and developing a new novel performance testing system capable of testing an athlete's agility. The definition and classifications of agility is analysed through recent and past literature. Discussion of the important fundamental concepts regarding measurement and evaluation of athlete testing is presented. The concepts of validity and reliability are analysed; how these are the most important qualities involving testing, how they are currently established in existing tests and when how they are recognised in a new test such as the one in this thesis. Agility tests that are well recognised and implemented in the field of sport science are analysed; how they are administered to athletes and what literature has to say about the validity and reliability for these to test agility. Emerging sports agility tests presented in research are discussed, as well as existing industry products in the contemporary market, establishing the gap in the market and how the proposed agility tester will fill these gaps.

A logical and systematic engineering design process was followed, which consisted of a series of stages; problem definition, conceptual design, solution concept, design embodiment and design detail. A needs identification was first conducted to understand the current problem and to clarify the objectives, primarily based upon the proposal brief supplied by the SHAPE Research Centre, the literature review and the competitive market analysis. Through the market research and analysis conducted, a set of design requirements were established that separated essential requirements from desirable ones. The conceptual design stage then targeted establishing functional structures, which were accomplished via an objective tree and functional decomposition to determine the function structure of the project. A set of subsequent design specifications were then able to be developed using the specification-performance method, based upon the initial stages of the design process. By establishing a set of qualitative, scalable values to the design specifications, this served as a framework for the following conceptualisation and development of the product, to work towards set target values that could be compared against when the product was complete to determine the success of the design. The quality function deployment process was followed, hereby developing a House of Quality chart that enabled correlations and conflicts to be determined between specifications and requirements, as well as assisting in determining a set of importance values for the specifications; so that the high importance ones could be watched over carefully through design and development.

A morphological chart was employed, which established a set of means for achieving given functions established in the function structure. The solution-neutral processes thus far left the door open for all possible designs to be considered. The alternatives were evaluated against one another using a decision matrix, which assisted in the selection process for refinement of the final solution concept. By integrating the most optimal performing means, an optimised solution layout was made that assisted in componentry selection for purchasing.

Preliminary design embodiment through CAD was conducted to visualise the solution layout. Using the CAD models as a framework, the physical embodiment of mechanical, electrical and software design could go underway. The process was iterative in nature, jumping back and forth between various stages of the design process. Finally, integration of the prototype was completed such that a final solution could be presented. The design details were documented including the final programs, testing procedures and electronic schematics. The actual metric values were compared with the engineering specifications to determine the success of the design. The significance and any limitations associated with the solution were presented here. Subsequent recommendations for future work were noted.

2. Literature Review

2.1. Agility

Current literature presents the definition of agility to be a clouded area, with no solid consistency amongst the sports science community. Terms such as “quickness” or “change-of-direction” are often used interchangeably with agility. However, it does seem a number of publications and textbooks have acknowledged and agreed that this ambiguity does exist regarding the definition and classification of agility (Chelladurai, 1976; Cooke et al., 2011; Coulson and Archer, 2009; Haff and Triplett, 2016; Liefeith et al., 2018; Lloyd et al., 2009; Paul et al., 2016; Sheppard and Young, 2006; Stewart et al., 2014; Young et al., 2015).

2.1.1. The Ambiguity in Defining Agility

Traditionally, agility has been defined in literature simply as *“the ability to change direction rapidly.”* (Baumgartner and Jackson, 1983; Lloyd et al., 2009; Mathews, 1978; Parsons and Jones, 1998). Other scientific literature have adapted to this, defining it as *“the ability to change direction rapidly and accurately”* (Barrow and McGee, 1979; Johnson and Nelson, 1986; Lacy, 2011). Adding further to the uncertainty, some literature has suggested that agility involves a change of direction using the whole-body, in conjunction with rapid changing limb direction and movement (Baechle, 1994; Draper and Lancaster, 1985). In an increasing number of emerging literature, a common trend is becoming present, with acknowledgement that agility involves a combination of physical and perceptual and decision-making factors (Benvenuti et al., 2010; Born et al., 2017; Chelladurai, 1976; Coulson and Archer, 2009; Haff and Triplett, 2016; Inglis and Bird, 2016; Nimphius, 2014; Paul et al., 2016; Sheppard and Young, 2006; Tanner et al., 2013). More recently, in a publication by Liefeith et al. (2018), the authors mention how agility is complex, involving a range of movements and coordination, indicating commonly presented definitions such as the ones mentioned above do not fit the complexity of the construct. Liefeith et al. suggests agility should be considered as a dynamic, complex and challenging integration of many abilities to provide movement solutions that satisfy the needs imposed by a physical context that has rapidly changed.

The independent definitions of agility provided in literature throughout the last 40 years were investigated (Table 1). It is very apparent that the definitions vary among authors, which could be a result of different backgrounds, thoughts and ideas on the construct of agility.

Table 1: Independent definitions of agility presented in literature

Author	Year & Page No.	Definition of Agility
Barrow and McGee	1979, p.113	*"The ability of the body or parts of the body to change directions rapidly and accurately."*
Baumgartner and Jackson	1983, p215	*"The ability to change direction of the body or body parts rapidly."*
Draper and Lancaster	1985, p.16	*"The ability to change the direction of the body rapidly and is a result of a combination of strength, speed, balance and coordination. "*
Johnson and Nelson	1986, p.215	*"The physical ability which enables an individual to rapidly change body position and direction in a precise manner. "*
Verstegen and Marcello	2001, p.140	*"The ability to react to a stimulus, start quickly and efficiency, move in the correct direction, and be ready to change direction or stop quickly to make a play in a fast, smooth, efficient, and repeatable manner."*
Hoffman	2006, p.112	*"The ability to change direction rapidly."*
Sheppard and Young	2006, p.922	*"A rapid whole-body movement with change of velocity or direction in response to a stimulus"*
Coulson and Archer	2009, p.156	*"Is made up of several discrete components such as balance, reaction or decision-making, coordination, technique, strength and power, which are all trainable components."*
Lacy	2011, p.206	*"The ability to rapidly and accurately change the position of the body in space."*
Nimphius	2014, p.185	*"The perceptual–cognitive ability to react to a stimulus such as a defender or the bounce of a ball in addition to the physical ability to change direction."*
Haff and Triplett	2016, p.522	*"The skills and abilities needed to change direction, velocity, or mode in response to a stimulus."*

Author	Year & Page No.	Definition of Agility
Liefeith et al.	2018, p.2	*"[Agility] should be conceived as describing movement solutions which require the dynamic integration of a number of sub-capacities (speed, forceful contraction, mobility, dexterity, balance, postural control, coordination, perceptual awareness, reflexive decision making, etc.) in some complex and challenging permutation to satisfy the demands imposed by a rapidly changing physical context…It may therefore be posited that a wide range of contributing sub-capacities enable agile behaviour."*

The frequency of certain keywords within the above definitions were tabulated (Table 2), where some keyword frequency has been added when a word has not necessary been stated, but the actual wording is representative of it.

Table 2: Frequency of keywords from independent definitions of agility

Term Used	Frequency	Sources
Change position/ direction	**9**	Lacy, Barrow and McGee, Baumgartner and Jackson, Sheppard and Young, Nimphius, Haff and Triplett, Verstegen and Marcello, Draper and Lancaster, Hoffman
Whole-body	**6**	Lacy, Barrow and McGee, Baumgartner and Jackson, Johnson and Nelson, Sheppard and Young, Draper and Lancaster
Body parts	**2**	Barrow and McGee, Baumgartner and Jackson
Balance	**3**	Coulson and Archer, Lancaster and Draper, Liefeith et al.
Coordination	**3**	Coulson and Archer, Lancaster and Draper, Liefeith et al.
Rapidly/Quickly	**7**	Lacy, Barrow, Baumgartner and Jackson, Verstegen and Marcello, Draper and Lancaster, Liefeith et al., Hoffman
Strength/Power	**3**	Coulson and Archer, Draper and Lancaster, Liefeith et al.
Technique	**1**	Coulson and Archer
Accurately/ Precisely	**3**	Lacy, Barrow and McGee, Johnson and Nelson, Verstegen and Marcello

Term Used	Frequency	Sources
Speed/Velocity	**4**	Sheppard and Young, Haff and Triplett, Draper and Lancaster, Liefeith et al.
Reaction/ Response	**3**	Coulson and Archer, Nimphius, Haff and Triplett
Efficiency	**1**	Verstegen and Marcello
Decision Making	**2**	Coulson and Archer, Liefeith et al.
Stimulus	**5**	Sheppard and Young, Nimphius, Haff and Triplett, Verstegen and Marcello, Liefeith et al.

A specific frequency criterion was used, where the key term must be present in at least four independent agility definitions. Thus, based on the current literature, it is clear to see that when referring to the key terms which met this criterion, the words could be put together to define agility as: *the ability of rapidly (or quickly) changing position (or direction) of the whole-body due to a stimulus with speed/velocity.* Now that key terms relating to agility had been identified, it was important to take an in-depth analysis of the components of agility presented in literature and recognize an established definition in current literature so that the ideas, theories and concepts presented in this thesis were substantiated.

2.1.2. Components of Agility

A deterministic model of agility components was proposed by Young et al. (2002), which was developed so that the main factors could be identified and applied to sports involving the ability. This was later modified by Sheppard and Young (2006), then further modified and expanded by Nimphius (2014), with the most recent model shown in Figure 3.

The multifactorial model suggests a range of components are related to agility, involving perceptual and decision-making factors while also incorporating physical qualities relating to change-of-direction speed. Physical sub-components of CODS includes anthropometric components such as body position or individual technique (foot placement, stride adjustment and body lean and posture), straight line sprint speed, strength (concentric, isometric and eccentric), power, force development and reactive strength (Young et al., 2002).

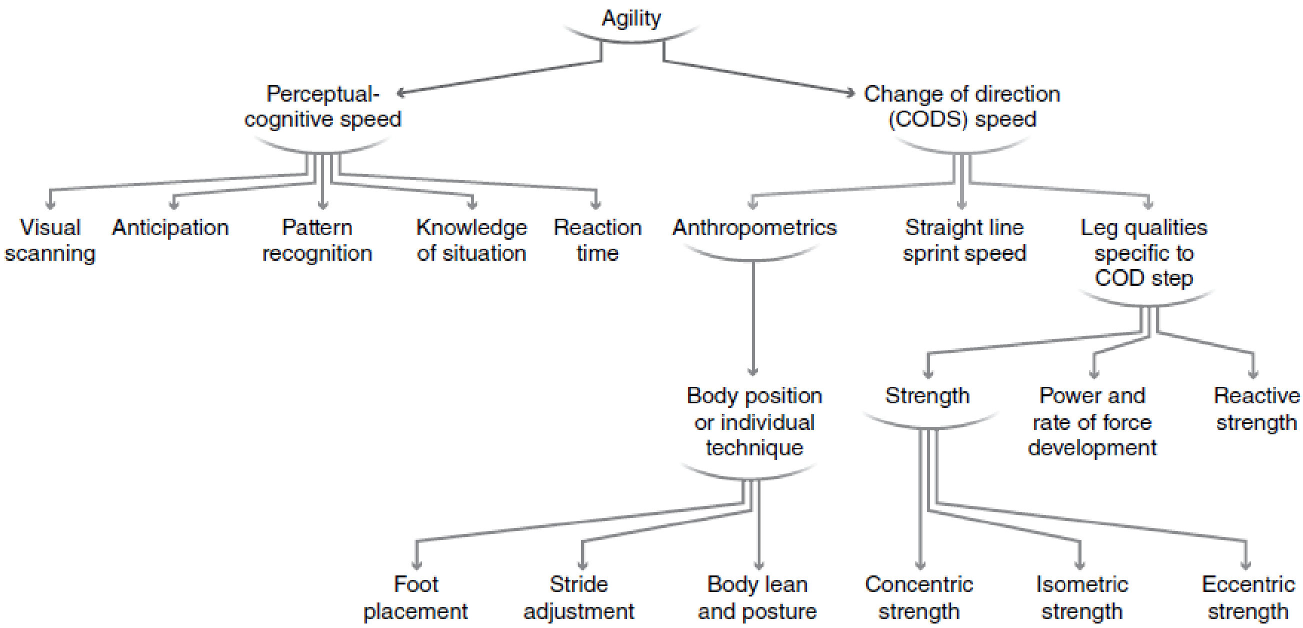


Figure 3: Model of the components of agility (Nimphius, 2014)

Young et al. suggests that although these physical and biomechanical qualities of agility are of great importance for higher performance CODS, it is the perceptual components such as visual scanning, anticipation, pattern recognition and knowledge of situations that are the significant components that differentiate higher-skilled performance sporting athletes from their lesser-skilled counterparts. This has been proven to be consistent in several studies (Farrow et al., 2005; Gabbett et al., 2008; Henry et al., 2012; Serpell et al., 2010; Sheppard et al., 2006). Liefeith et al. (2018) emphasised agility consists of a wide range of contributing sub-capacities, similarly to how Young et al. (2002) and Nimphius (2014) describe agility to contain a range of components in a multifactorial sense. Other scientific literature is in agreeance that agility involves a multitude of sub-components of physical and cognitive factors, where Morc and Coulson (2009) state agility consists of a variety of discrete components involving physical and cognitive; strength, power, balance, technique, coordination, decision-making and reaction. Dawes and Roozen (2012) mention that agility involves a multitude of component abilities including physical factors (speed, strength, eccentric strength, stabilisation strength, power, rate of force development, strength-shortening cycle, anthropometric variables and technique) and cognitive factors due to quickness (a factor affecting agility, which consists of information processing, knowledge of situations, decision-making skills, anticipation and arousal level).

A significant amount of literature has mentioned perceptual and decision-making factors are important components of agility (Table 3). The list of sources in not exhaustive, as there are many more research articles that the author has not read. However, the table does show that a significant amount of scientific literature agrees the cognitive factor is an important component of agility.

Table 3: Literature that suggest agility involves perceptual and decision-making factors

Frequency	Sources
18	Benvenuti et al., 2010; Born et al., 2017; Coulson and Archer, 2009; Dawes and Roozen, 2012; Haff and Triplett, 2016; Jeffreys, 2011; Lockie et al., 2014; Nawi and Homoud, 2015; Nimphius, 2014; Scanlan et al., 2013; Sheppard and Young, 2006; Šimonek et al., 2016; Spiteri et al., 2012, 2014; Stewart et al., 2014; Tanner et al., 2013; Verstegen et al., 2001; Young and Rogers, 2016

2.1.3. Defining Agility Based on Literature

Sheppard and Young (2006) undertook a thorough investigation into the classifications, training and testing of agility in literature, suggesting a comprehensive definition of agility would allow recognition for all components; the cognitive processes involved with motor learning, the technical skills in relation to biomechanics and the physical demands when considering the strength and conditioning aspect. Thus, they proposed agility to be defined as *"a rapid whole-body movement with change of velocity or direction in response to a stimulus" (*Sheppard and Young, 2006, p. 922). The definition proposed by Sheppard and Young (2006) correlates well with the definition developed for agility based on the frequency of keywords from independent definitions presented in literature: *the ability of rapidly (or quickly) changing position (or direction) of the whole-body due to a stimulus with speed/velocity.* Thus, it is reasonable to accept this definition of agility by Sheppard and Young. By accepting this statement as the definition for agility, both the cognitive and physical components are acknowledged.

An increasing number of scientific literature is accepting this definition by Sheppard and Young in their research, recognizing the importance of the reaction to a stimulus as a key component in agility (Benvenuti et al., 2010; Born et al., 2017; Coulson and Archer, 2009; Haff and Triplett, 2016; Jeffreys, 2011; Lockie et al., 2014; Scanlan et al., 2013; Serpell et al., 2011; Spiteri et al., 2012; Stewart et al., 2014; Tanner et al., 2013; Young and Rogers, 2016). More recently, Paul et al. (2016) and Inglis and Bird (2016) undertook systematic reviews of literature regarding reactive agility tests. In agreeance with Sheppard and Young (2006), they too emphasised the importance of a sport specific stimulus when testing agility. This makes sense, as a large majority of sports require an athlete to make perceptual and decision-making in response to some type of stimulus, either due to the location of a ball, equipment, or by the movement of a defending or attacking opponent. Australian football, soccer, rugby, netball, tennis, hockey, badminton, volleyball, boxing, or martial arts are all examples where an athlete will find themselves in these circumstances. Athletes will often need to make sudden changes of direction by decelerating and reaccelerating rapidly and accurately as they react to a situation (Nimphius, 2014).

2.1.4. Classifying Agility

Sheppard and Young (2006) went further to classify agility (Table 4), so that confusion would be avoided, and the terms used could become more standardized in application.

Table 4: Criteria for the classification of agility (Modified from Sheppard and Young, 2006, p. 930)

Agility	Other Physical or Cognitive Skills
• Must involve initiation of body movement, change of direction, or rapid acceleration or deceleration • Must involve whole-body movement • Involves considerable uncertainty, whether spatial or temporal • Open skills only • Involves a physical and cognitive component, such as recognition of a stimulus, reaction, or execution of a physical response	• Entirely pre-planned skills such as shot-put classified by their skill function rather than included as a type of agility • Running with directional changes classified as change of direction speed rather than agility or quickness • Closed skills that may require a response to a stimulus (e.g. the sprint start in response to the starter's pistol is pre-planned (closed), and therefore is not agility)

Several aspects already mentioned are included within the table, with the addition that agility involves considerable spatial or temporal uncertainty, as well as involving open skills only, while other physical or cognitive skills involve closed skills. A closed-skill involves a movement where a person knows exactly what is expected from them; some form of pre-programmed/pre-planned movement that does not involve a response to a stimulus (Cox, 2011; Verstegen et al., 2001). Open skills are used when a person must react to a stimulus from the surrounding environment perceived via their sensory input systems, in which the movement contains some form of ambiguity that is not automated or rehearsed (Cox, 2011; Verstegen et al., 2001). An example of a closed-skill is the movement required for throwing a javelin across the field, while an example of an open skill could be the reaction of an athlete to a passing of a ball or oncoming opponent. Therefore, open skills are an important aspect of agility performance.

2.1.5. Change-of-Direction Speed

Many publications consider change-of-direction speed and agility as two separate abilities (Haff and Triplett, 2016; Spiteri et al., 2014). CODS can be considered a pre-planned ability to change direction of movement where there is no requirement of a reaction to a stimulus (Haff and Triplett, 2016; Nimphius, 2014; Young et al., 2002), such as when a baseball player is running to first base after hitting the ball. An athlete changing direction of travel when the path of movement may be pre-planned is different to how an athlete may change direction when reacting to a stimulus such as an oncoming defender or the bounce of a ball (Nimphius, 2014). Sheppard and Young (2006) suggests conditioning exercises could be classified as either an agility test with sprints and change of direction in response to a stimulus, or a CODS tests that involves just sprints and change of direction. Therefore CODS can be considered a physical component of agility (Nimphius, 2014; Sheppard and Young, 2006; Young et al., 2002).

2.2. Athlete Testing

Athlete testing is a significantly useful method to evaluate the performance of athletes; acting to support improvement by allowing goals to be set and progress to be evaluated as well as differentiating one higher-performing athlete from another lower-performing one (Haff and Triplett, 2016). Both quantitative and qualitative information may be obtained about a given subject from athlete testing; quantitative measurements on numerical data such as weight, measurements or performance time, as well as qualitative observations such as through analysis or artefacts (Gratton and Jones, 2004; Pitney and Parker, 2001). Physiological and performance testing is the most common forms of athlete testing, which may have a focus on either health-related or skill-related physical fitness, such as aerobic or anaerobic fitness, endurance, balance, speed, coordination, strength, power, flexibility or agility (Lacy, 2011; Tanner et al., 2013). This form of testing can provide a way of obtaining relevant measurements to evaluate an athlete's physical abilities, performance or skills (Haff and Triplett, 2016; Tanner et al., 2013). It is important to establish the processes involved in physiological measurement and evaluation of athlete testing as this will provide the basis of understanding how to validate the measurements obtained from the proposed agility tester and have confidence in its results, thus, further in-depth analysis in this topic has been conducted in this literature review.

2.2.1. Evaluation of Test Quality

The reliability and validity of measurements obtained through athlete testing is of great importance as the tests needs to measure what it is supposed to measure as well as be repeatable. If the test does not possess these characteristics, then the measurement outcomes are not beneficial and would be thought of as questionable (Haff and Triplett, 2016; Lacy, 2011). It is important to ensure the reliability and validity of the proposed agility tester presented in this thesis. High reliability and validity of a test enables it to be conducted with confidence such that the measurements can be used to assess an athlete and differentiate them from one another; in terms of agility, determining the most agile from those who require further training (Johnson and Nelson, 1986; Lacy, 2011).

2.2.2. Reliability

For a test to be reliable, it therefore needs to have repeatability or reproducibility and consistency in its outcomes, such that the test may be repeated and produce similar results, under similar circumstances with no change in status or ability of the subject being tested (Barrow and McGee, 1979; Coulson and Archer, 2009; Haff and Triplett, 2016; Hopkins, 2000; Lacy, 2011; Vincent and Weir, 2012). Reliability can be quantitatively concerned with measurement error, which will always accompany any test measurement made (Baumgartner and Jackson, 1983; Bishop, 2008; Lacy, 2011; Vincent and Weir, 2012). An observed score of a test can therefore be described as a component of its true score and that of an error component (Equation 1).

Equation 1: Calculating observed score

$$Observed\ Score = True\ Score + Error$$

The averaged value of a subject's score obtained from an infinite number of tests provides a true score, while the difference between this true score and the observed score describes the error component. As the reliability of a test increases, the difference between the observed score and true score will decrease. Therefore, a highly reliable test indicates that the errors in measurement are small (Vincent and Weir, 2012).

Test reliability may be determined via three methods; analysis of variance, test-retest and split halves (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Vincent and Weir, 2012). Both Barrow and McGee (1979) and Baumgartner and Jackson (1983) state that the most preferred method of establishing reliability of a test is through the intraclass correlation method (via estimation through the analysis of variance). Baumgartner and Jackson also state the test-retest method is next preferred, with the split-halves method as the least desirable method. Therefore, the split-halves method will not be described in this literature review, nor will it be implemented in this thesis.

2.2.2.1. Analysis of Variance

One technique for establishing test reliability is through analysis of variance (ANOVA) via the intraclass correlation method. This technique accounts for subject test performance variability from test to test and also from day to day (Barrow and McGee, 1979; Baumgartner and Jackson, 1983). The total variability of the measured outcome for each test and each day is compared by dividing it into several parts, similarly to the observed score mentioned previously, using the ANOVA method (Baumgartner and Jackson, 1983). In doing this, the intraclass correlation coefficient (ICC) can be determined and will indicate the reliability of the test for a given number of trials over a given number of days (Barrow and McGee, 1979; Baumgartner and Jackson, 1983). Hopkins (2000) suggests for any more than two trials, it makes most sense to use the intraclass correlation method.

2.2.2.1.1. Spearman-Brown Prophecy Formula

The ANOVA method may be manipulated with the Spearman-Brown Prophecy Formula to obtain various values for the coefficient of reliability based on a combination of number of tests performed over a given set of days (Barrow and McGee, 1979). The Spearman-Brown Prophecy Formula (Equation 2) may be used as a way of estimating the adequacy of the coefficient for a given number of days and tests (Barrow and McGee, 1979; Baumgartner and Jackson, 1983).

Equation 2: Spearman-Brown Prophecy Formula

$$r_{k,k} = \frac{k(r_{1,1})}{1 + (k-1)(r_{1,1})}$$

Where k is the number of times the test has changed by either the number of trials undertaken or by the length of the test, $r_{k,k}$ is the reliability of the test increased by a length of k times and $r_{1,1}$ is the reliability of the original tests.

This formula can be used to determine the upper limit of the reliability coefficient (Baumgartner, 1968). An example of use for a given test, could be that it receives a 0.96 reliability coefficient by undertaking the test three times per day for six days. Alternatively, a 0.91 reliability coefficient may be achieved from undertaking the test just twice over two days.

Initially, one method for confirming the reliability of the proposed test to be developed in this thesis was going to be completed through analysis of variance (ANOVA) via the intraclass correlation coefficient by integration with the Spearman-Brown Prophecy Formula. However, due to time constraints of the project, it was decided that there would not be enough time to develop the system as well as test it on athletes. Thus, it was determined the reliability through ANOVA via the intraclass correlation coefficient integrated with the Spearman-Brown Prophecy Formula of the test would need to be completed in future work.

2.2.2.2. Test-Retest

Another technique for establishing test reliability is by conducting the test on a sample of subjects once and then repeating it at least once (Barrow and McGee, 1979; Bishop, 2008; Vincent and Weir, 2012). The test must not be repeated until the subjects are fully recovered, preferably on a different day following the initial test, and under very similar test conditions. The correlation coefficient may be determined by comparing the scores of the two tests (Barrow and McGee, 1979; Bishop, 2008).

2.2.2.3. Pearson Product-Moment Correlation

This correlation coefficient can be determined by using the Pearson Product-Moment Correlation method (Equation 3), which can compare two different independent measures (Barrow and McGee, 1979; Lacy, 2011).

Equation 3: Pearson Product-Moment Correlation

$$r = \frac{n\Sigma \mathrm{XY} - (\Sigma \mathrm{X})(\Sigma \mathrm{Y})}{\sqrt{[n\Sigma \mathrm{X}^2 - (\Sigma \mathrm{X}^2)][n\Sigma \mathrm{Y}^2 - (\Sigma \mathrm{Y}^2)}}$$

Where r is the Pearson product-moment correlation coefficient, n the number of subjects (or pairs of scores) and X and Y the test scores for the two independent measurements taken.

Use of the Pearson Product-Moment Correlation method is yet another ambiguous factor in the physical sciences field. Lacy (2011) recommends using this method when measuring test-retest reliability. Hopkins (2000) states it is an adequate estimate of retest correlation, although slightly biased. Bishop (2008) and Baumgartner and Jackson (1983) suggest this method is not appropriate in assessing the relationship between measures of the same measurement since it is conducted under the assumption that there is two independent variables. They propose the intraclass correlation method previously mentioned, using the ICC as a more suitable method for estimating reliability, which is in agreeance with Hopkins (2000).

Use of the Pearson product-moment correlation coefficient has been used to test between-test and re-test relationships in various research studies (Chaouachi et al., 2009; Čoh et al., 2018; Farrow and Abernethy, 2002; Fessi et al., 2016; Hachana et al., 2014; Haugen et al., 2014; Hermassi et al., 2011; Inglis and Bird, 2016; Nawi and Homoud, 2015; Young et al., 2011). Therefore, it was initially thought that the proposed agility test's reliability would be evaluated through the test-retest technique alongside the Pearson Product-Moment Correlation Equation. Due to the time constraints presented by the project discussed already, it was found that evaluating the reliability would not be feasible within this thesis; however, it should certainly be addressed in future work.

2.2.3. Validity

For a test to be considered valid, it must measure what it has specified it is to measure, as accurately as possible (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Bishop, 2008; Haff and Triplett, 2016; Lacy, 2011). Furthermore, to be truly valid, the test needs to simulate energy and movement requirements the sport would normally situate the athlete in, related to the ability being tested (Haff and Triplett, 2016). Of the characteristics of measurement, validity has been established as the most important (American Educational Research Association et al., 2014; Barrow and McGee, 1979; Bishop, 2008; Haff and Triplett, 2016; Lacy, 2011). A high validity test can be conducted with the measurements evaluated with confidence (Lacy, 2011). For a test to be valid, it must be reliable; therefore the validity of a test is influenced by its reliability (Baumgartner and Jackson, 1983; Bishop, 2008). However, it is possible that a test can be reliable, yet invalid (Baumgartner and Jackson, 1983; Bishop, 2008).

There exist several types of validity; content validity, construct validity and criterion-related validity. These are the general types mentioned from Lacy (2011), Haff and Triplett (2016), Barrow and McGee (1979) and Baumgartner and Jackson (1983), although some may group the types slightly differently.

2.2.3.1. Content Validity

Content validity is also known as face validity or logical validity. This type of validity refers to the degree at which the test can satisfy whether it is truly measuring a skill or ability (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Lacy, 2011). If the test satisfies the components of the ability, then it can be inferred to have content validity (Barrow and McGee, 1979; Lacy, 2011). Due to the ambiguity of the classifications and definition of agility, establishing content validity of an agility test is a difficult procedure. It is likely that this is the very reason why the term "agility test" refers to such a wide variety of tests involving dissimilar traits, which is discussed in Section 2.3 of this literature review. This type of validity is based on subjective decision making, usually through professional judgement and logic, instead of statistical procedures (Barrow and McGee, 1979; Baumgartner and Jackson, 1983). The content validity of the proposed agility test will be established by ensuring it does indeed test the components of agility established earlier.

2.2.3.2. Construct Validity

A construct is what is referred to as a fundamental concept, theory or idea that is made up of multiple simpler elements, which forms an underlying characteristic to be measured (Bishop, 2008; Lacy, 2011). Construct validity therefore refers to the degree a test measurement can accurately measure an underlying construct (Haff and Triplett, 2016). The construct in the case of this thesis is the concepts that underly the characteristics that make up the ability of agility. This interpretation of the term construct is confirmed from Liefeith et al. (2018), which discuss how the interpretation of the agility construct needs to be clarified. Therefore, the construct can be quite complex, as it is a multifactorial ability (Liefeith et al., 2018; Nimphius, 2014; Sheppard and Young, 2006).

Validating construct validity is normally achieved by comparing the results between two groups that are known to differ significantly regarding the construct being tested; such as elite athletes compared with beginners in a particular sport (Barrow and McGee, 1979; Lacy, 2011). It is known that theoretically the elite athletes would perform far better in the test as they would have been profoundly trained in the sport giving them expertise in it (Baumgartner and Jackson, 1983; Lacy, 2011). Therefore, the test would be said to have construct validity if there is a statistically significant difference in the scores of the two groups. As the proposed agility test will only be conducted internally and not be conducted on a group of elite athletes and non-athletes due to time constraints in getting ethics approval, it may not be achievable to determine the construct validity of the test. It would be very beneficial to determine the construct validity in future work.

2.2.3.3. Criterion-Related Validity

This type of validity refers to the degree in which the test score is associated with criteria that accurately measures the same ability (Haff and Triplett, 2016; Lacy, 2011). When establishing evidence of validity using this technique, the criteria being used as a reference needs to have a high level of confidence in the accuracy of the measurement (Lacy, 2011). Criterion-related validity can be grouped into two types; concurrent validity and predictive validity.

2.2.3.3.1. Concurrent Validity

The concurrent validity of a test is the degree to which it scores against another established test that measures the same ability (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Haff and Triplett, 2016; Lacy, 2011). Both the new test and the already established test are undertaken by a subject and correlated. This correlation can be determined via the Pearson product-moment correlation coefficient (Baumgartner and Jackson, 1983; Haff and Triplett, 2016). The new test can be considered to have concurrent validity if there is a high correlation between the tests (Barrow and McGee, 1979; Lacy, 2011). With increasing correlation coefficient comes greater confidence in the validity of the results of the new test (Lacy, 2011). To determine the concurrent validity of the proposed agility test, it could be compared against well-established agility tests such as the Illinois Agility Test or AFL Agility Test, mentioned later in Section 2.3 in this literature review. A major problem in this instance, is that these pre-planned agility tests do not truly represent all components of an athlete's agility. Therefore, the proposed agility test may not have a high correlation with these tests.

2.2.3.3.2. Predictive Validity

The predictive validity of a test is the degree to which the measurement may predict or correspond with some form of future measure or performance (Barrow and McGee, 1979; Haff and Triplett, 2016). To determine the predictive validity, the results of the test would need to be compared with some form of measurement such as the athlete's game performance in their sport (Haff and Triplett, 2016; Lacy, 2011). Based on the scores measured from the test, the athlete performance should be able to be predicted. If there is a high correlation between the two, then this test is said to have a high predictive validity (Lacy, 2011). Therefore, scoring highly on the test would predict that the athlete would have good game performance.

This project is focusing on the design and development of the agility test, which will incorporate many features. As this form of validity requires an initial test to be undertaken and then another criterion measurement to be administered at a future date to correlate the scores, the predictive validity will not be able to be determined in this project to assess the measurement. This is due to time constraints, as it would not be possible to go through the process of designing and the developing the agility tester as well as going through the process of obtaining ethics approval to administer the test on highly skilled athletes on two separate occasions with the test measure and another criterion measure.

2.2.4. Statistical Analysis

2.2.4.1. *Correlation Coefficients*

The validity and reliability of a test may be interpreted by using the ICC mentioned previously, which is a statistical technique (Barrow and McGee, 1979). As stated, the reliability coefficient can be determined by correlating the measurements obtained from a test with the measurements of the same test repeated at least once (test-retest), or though the intraclass approach through ANOVA.

2.2.4.2. *Interpretation of Correlation Coefficients*

A correlation coefficient can range from -1.00 to +1.00, with the latter being the optimal correlation (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Lacy, 2011). A correlation coefficient greater than 0 represents a positive correlation, whilst any value below 0 is classified as a negative correlation (Lacy, 2011). Interpretation of validity and reliability coefficients above .60 can be observed in Table 5, which has been modified from Barrow and McGee (1979). From the table, a validity coefficient needs to be above .70 to be acceptable, while a reliability coefficient must be .80 to be considered acceptable. These standards will be implemented into this thesis to determine the validity and reliability of the proposed agility test.

Table 5: Standards for interpreting correlation coefficients (Modified from Barrow and McGee, 1979)

Coefficients	Validity	Reliability
.95 to .99	Excellent	Excellent
.90 to .94	Excellent	Very good
.85 to .89	Excellent	Acceptable
.80 to .84	Very good	Acceptable
.75 to .79	Acceptable	Poor
.70 to .74	Acceptable	Poor
.65 to .69	Questionable (except for very complex tests)	Questionable (except for groups)
.60 to .64	Questionable	Questionable (except for groups)

2.3. Testing Agility

Testing the ability of agility has been of great importance in the sports science community as agility is essential in both field and court sports involving physical demands of changing direction quickly such as in team sports of Australian rules football, netball or rugby (Nawi and Homoud, 2015). Current established agility tests contain a variety of different factors, including change-of-direction, manoeuvrability and more recently, a stimulus that forces a reactive response (Haff and Triplett, 2016; Nimphius, 2014). Agility has been regarded as an important ability to evaluate the performance of athletes, capable of distinguishing higher-skilled athletes from lesser-skilled ones (Paul et al., 2016). Athletes that are highly agile can use this ability to help their performance within the sport they play, displaying quick changes in body position, footwork efficiency (Barrow and McGee, 1979), as well as helping to minimise the risk of injury (Verstegen et al., 2001). This section of the literature review discusses and analyses the most popular currently established agility tests conducted in athlete performance testing batteries, combines and training. Something to note for all tests discussed is that for higher accuracy and reliability of results, timing gates would be required, which are more expensive and harder to set-up (Haugen et al., 2014).

2.3.1. Illinois Agility Test

The Illinois Agility Test is one of the most common agility tests conducted extensively in a variety of sporting and research applications (Coulson and Archer, 2009). The test is used to assess multidirectional CODS, body control and straight sprinting speed and technique (Dawes and Roozen, 2012; Reiman and Manske, 2009). Traditionally, the test zone course area is 30 ft (9.15 m) by 12 ft (3.66m) (Lacy, 2011). However, there is another variation of the test, which incorporates values in metres (rounded), so that the course area is 10 m by 5 m (Figure 4) (Dawes and Roozen, 2012). Four cones are used to mark the start, finish and locations where the subject must turn around (cones A, B, C and D), with another four cones placed in the centre of the test zone, which the athlete will need to navigate around (cones 1, 2, 3 and 4). These middle cones are placed 10 ft (3.05 m) from each other in the traditional layout and 3.3 m apart in the alternative version (Coulson and Archer, 2009; Lacy, 2011).

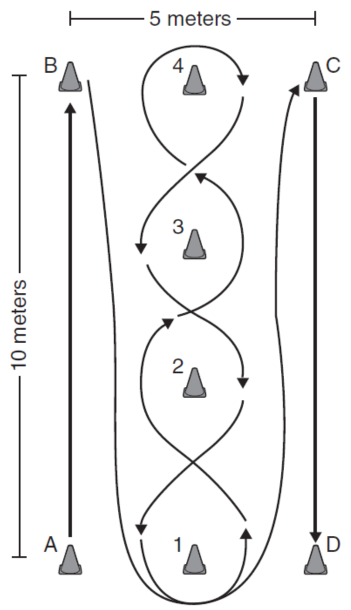


Figure 4: Illinois Agility Test (Dawes and Roozen, 2012)

The subject must begin laying in a prone position (laying face down on the floor), with hands at the sides of the chest, on the starting line (cone A), facing the direction of the course (Coulson and Archer, 2009; Lacy, 2011). On the test administrator's command, the subject must get up as fast as possible and sprint to the first turning point (cone B). The subject must have at least one foot cross the line, before changing direction to run back in the opposite direction towards cone 1. They must then zigzag their way through the centre cones 1-4, which will make them reach the far line and back to the start line again. They will then need to sprint towards the cone C, and once reaching the turning point, reverse directions and sprint to the finish line (cone D) crossing it. The time taken to complete the course is recorded.

The Illinois Agility Test is simple to set-up, administer and requires minimal equipment to conduct (just eight cones). Since this test is highly standardised and is a well-established test among a variety of sports, there exists large sets of data, and therefore results can be compared against general standards and normative data (Table 6).

Table 6: Normative Illinois Agility Test data for adults (Modified from Roozen, 2004)

	Rating				
	Poor	**Fair**	**Average**	**Good**	**Excellent**
Males	>18.3 s	18.3 - 18.2 s	18.1 - 16.2 s	16.1 - 15.2 s	<15.2 s
Females	>23.0 s	23.0 - 21.8 s	21.7 - 18.0 s	17.9 - 17.0 s	<17.0 s

The Illinois Agility Test is considered one of the "gold standard" agility tests; yet it is pre-defined from the beginning; the athlete knows the path they need to travel so this is a measure of a closed skill since they can pre-plan their movement. The test does not involve any reaction in response to a stimulus, so it lacks the cognitive component of agility described in this literature review. It also does not require them to use their hands, apart from getting up from the floor. In the standardised test, sporting equipment is not incorporated as it would be hard to establish a universal set of rules. However, during training, the athlete can use sporting equipment such as a player dribbling a basketball, soccer ball or football whilst undertaking the test.

2.3.2. AFL Agility Run Test

This planned agility test was specifically designed for the Australian Football League (AFL) and is part of the AFL National Draft Combine (Chalmers and Magarey, 2016; Pyne et al., 2005; Robertson et al., 2015; Tanner et al., 2013; Woods et al., 2015). The test is designed to evaluate an athlete's ability of overall agility and CODS (Tanner et al., 2013). The test involves the AFL player running in a twisting motion, in, out and around obstacles (typically large PVC piping ≈17.5 cm in diameter and ≈1.5 m high). The test set-up (Figure 5) covers approximately 25 m^2 of area.

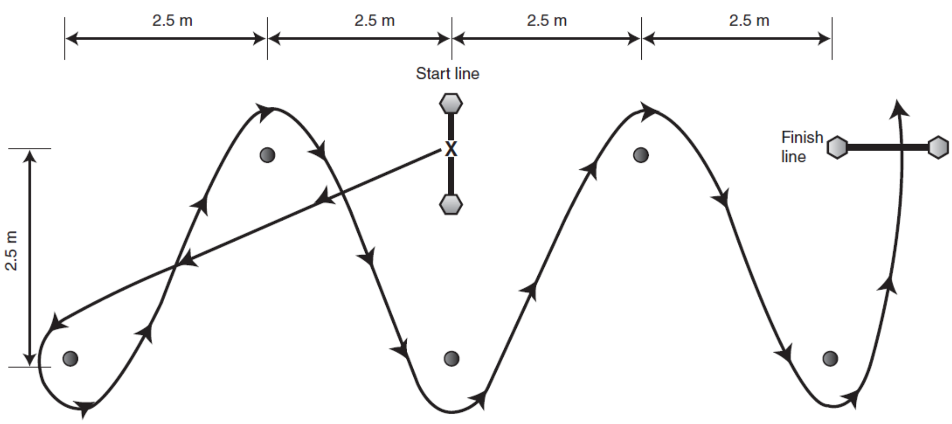


Figure 5: The Planned AFL agility run (Tanner et al., 2013)

The player starts in a stationary, upright position with the leading foot on the starting line (Tanner et al., 2013). On command, they must weave in and out of the obstacles in the direction shown in Figure 5, ensuring not to knock them; if an obstacle is moved at all, then the trial must be stopped and repeated. The time to complete the test is recorded in seconds. The best score of three efforts is recorded. Field marking tape is placed on the ground where the obstacles are located, so that if they are knocked over, they can be accurately repositioned. This is a smart idea and will be implemented into the proposed agility tester when positioning markers.

Surprisingly, the AFL agility run does not incorporate a football within the test, so it does not completely represent in-game performance. The test is considered a planned agility test, as the movement pattern is known, such that an athlete can repeat the test in an identical manner for each effort (Tanner et al., 2013). As a result, this test does not evaluate perceptual and decision-making factors and most-likely involves only closed-skills due to the pre-planned nature of the test. However, a modification of the test exists where a reactive component is added, where the player must react to an external stimulus (light, video or human) (Tanner et al., 2013).

2.3.3. 5-0-5 Agility Test

The 5-0-5 Agility Test was developed by Draper and Lancaster (1985) to be used as a way of measuring agility in the horizontal plane. It is a way of determining CODS, as well as strength and technique during accelerating and decelerating (Dawes and Roozen, 2012). The 5-0-5 Agility Test is a highly recognised test used in many sports and research applications. In the traditional test, a start line is made with two cones; another two cones are placed 10 m away and another two placed 15 m away from the start cones and 5m from the second set of cones (Figure 6) (Dawes and Roozen, 2012).

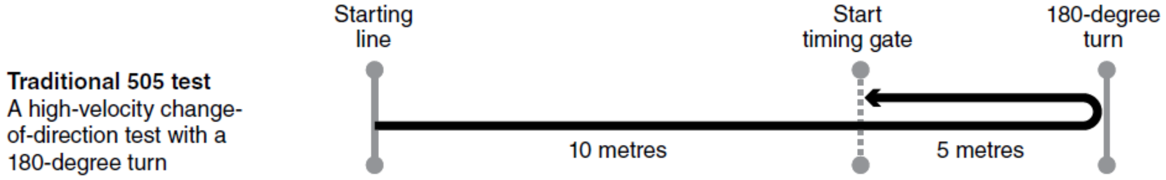


Figure 6: Traditional 5-0-5 Test (Nimphius, 2014)

The athlete begins at the start-line in a standing split start position (Dawes and Roozen, 2012). On command, the athlete must sprint as fast as possible to the turning line 15 m away. The athlete must make a 180-degree change-of-direction at this line and then sprint back 5 m, accelerating past the timing line. The time is recorded from the line 10 m from the starting line shown in Figure

6 and stopped as the athlete returns to this same line; thus, the 10 m distance is purely to gain speed and acceleration for a flying start. The test is repeated with legs alternating each time, for a minimum of three efforts per leg.

A modified version of the test exists, which removes the 10 m sprint section (Figure 7). As a result, the athlete starts the test still with no flying start, so there is a low-velocity entry before change-of-direction.

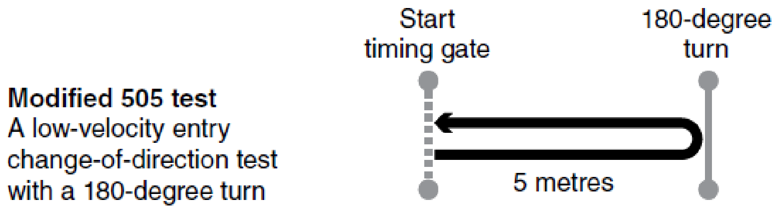


Figure 7: Modified 5-0-5 Test (Nimphius, 2014)

The 5-0-5 Agility Test can be used to distinguish between left and right leg performance differences (dominant vs. non-dominant leg) since it only contains one single 180-degree change-of-direction (Nimphius et al., 2012). Therefore, this attribute of the test is a major differentiator between other pre-planned agility tests like the Illinois, AFL Agility Test, 5-10-5 or T-Test since they require alternation between right and left legs for change-of-direction during the same effort performance. This test is very simple to set up and has a good measure of single sided and 180-degree CODS. However, this test does only test pre-planned agility and CODS, as there is no reactive component; it is not testing all components of agility. Trying to incorporate sporting equipment (such as dribbling a soccer ball, hockey puck or bouncing a basketball) into the test would be difficult to standardise.

2.3.4. T-Test or T-Drill

The T-test is one of the most common agility tests conducted extensively in a variety of sporting applications including NCAA testing (Coulson and Archer, 2009; Hoffman, 2006). The test was originally developed by Semenick (1990) to measure the ability an athlete has to change directions rapidly without losing CODS or balance. The pre-planned agility test is also used to determine the ability the athlete has to adjust their strides for accelerating and decelerating, controlling the body carefully with CODS whilst moving forward, backward and laterally (Dawes and Roozen, 2012). The test consists of four cones placed in a “T” shape (Figure 8).

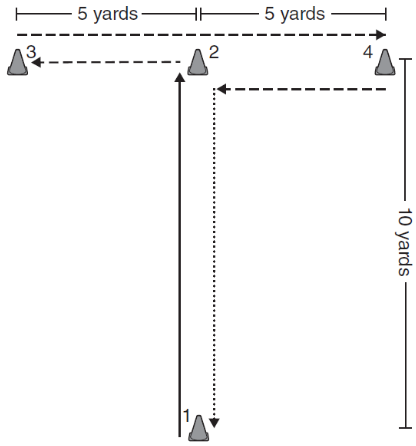


Figure 8: The T-Test (Dawes and Roozen, 2012)

The athlete must have an appropriate stance for their sport at cone 1, facing towards cone 2 (Dawes and Roozen, 2012; Reiman and Manske, 2009). Upon command, the athlete must sprint to cone 2 and touch the base with their right hand. Whilst remaining facing forwards always throughout the test, the athlete then shuffles sideways to the left, without crossing their feet, and touches the base of 3 with their left hand. They then shuffle sideways to their right to touch the base of cone 4 with their right hand. They then shuffle to their left to touch the base of 2 again with their left hand. Finally, they must run backwards to cone 1. The timer is stopped once they pass the starting line at cone 1.

The T-Test is easy to set-up, with minimal equipment required (only four cones). Research has shown a high test-retest consistency, proving high reliability. A major drawback to this test is that it, like the previously mentioned tests involves mere pre-planned agility since it is a predefined test. There does not exist any reactive component in response to a stimulus, as well as no implementation of sporting equipment with the test.

Pauole et al. (2000) found the ICC reliability values of the T-test to be between 0.94-0.98 for a total of 304 college-aged participants, showing it is highly reliable.

2.3.5. Pro-Agility Test (5-10-5 Shuttle or 20-Yard Shuttle Test)

The Pro-Agility Test is a very popular testing protocol, used as part of the performance testing battery for the NFL combine (Hoffman, 2006; McGee and Burkett, 2003). The test can assess CODS, leg strength, power and technique (Dawes and Roozen, 2012). Three cones are set up, 5 yards (4.6 m) apart from each other (Figure 9), covering a total of 10 yards (9.1 m).

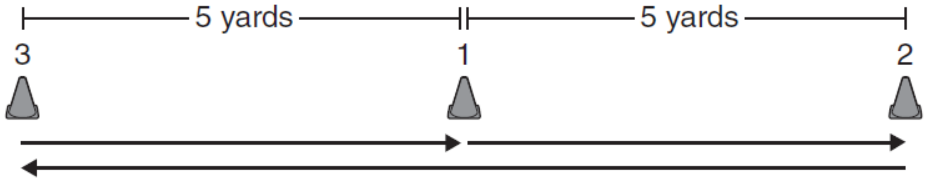


Figure 9: Pro-Agility Test (Dawes and Roozen, 2012)

The athlete assumes a three-point position by placing one hand on the ground of the centre start-line (cone 1) with feet shoulder width apart and placed equally either side of the line (Dawes and Roozen, 2012). The hand touching the line determines which way the athlete is going to travel (left hand means athlete must first go left, with same notion for right hand). The athlete must sprint to either cone 2 or 3, depending on the hand touching the ground. They must then change-direct and sprint to the opposite side cone and touch the ground with the opposite hand. Finally, they must sprint back past the centre-line (cone 1). As soon as the athlete breaks their three-point position, timing begins; it stops as they pass the centre-line at the end of the test. A minimum of three efforts is required.

The Pro-Agility Test has high test-retest consistency and is simple to set-up (requires only three cones) and administer. Limitations to the test is that it again, involves pre-planned agility and does not test any reactive ability.

2.3.6. 3-Cone Drill or L-Run Test

The 3-Cone drill is another popular agility test conducted, such as in the US NFL combine (Hoffman, 2006; McGee and Burkett, 2003). It is primarily aimed to assess CODS, acceleration, deceleration and technique (Dawes and Roozen, 2012). Three cones are placed 5 yards (4.6 m) from one another in an “L” shape (Figure 10).

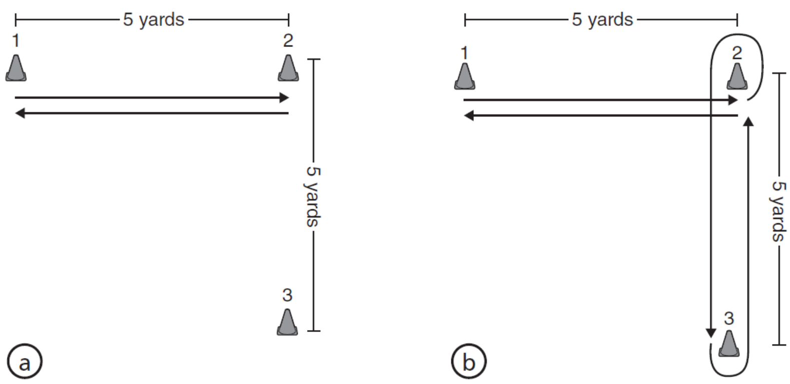


Figure 10: The 3-Cone Drill (Dawes and Roozen, 2012)

The athlete begins in a stance suitable for their sport at cone 1 and facing cone 2 (Dawes and Roozen, 2012). On command, the athlete sprints as fast as they can to cone 2, touching the ground with their right hand. They do a 180-degree change-of-direction and sprint back to cone 1, touching the ground with the right hand once again (Figure 10a). The athlete then sprints back to cone 2, turns around the outside of it, doing a 270-degree spin so that they can then sprint to cone 3. They must run around the outside of cone 3 and sprint back to cone 2 (Figure 10b). The athlete must then make a sudden change-of-direction left, sprinting back past cone 1. The timing starts on command and is stopped once the athlete has passed cone 1.

The 3-Cone Drill is very simple to set-up (just three cones) and implements a multitude of extremely physically demanding CODS, testing the physical capabilities of agility. However, there does not contain any reactive component to this test, so the cognitive factors of agility are not tested.

2.3.7. Hexagon Test

The hexagon agility test can assess body control, balance and coordination during production of high forces, as well as the ability of accelerating and decelerating rapidly, changing direction accurately (Dawes and Roozen, 2012). A hexagon is marked on the floor (Figure 11) using either chalk or tape with sides 2 feet apart (60.5 cm) with an angle of 120-degrees between each other.

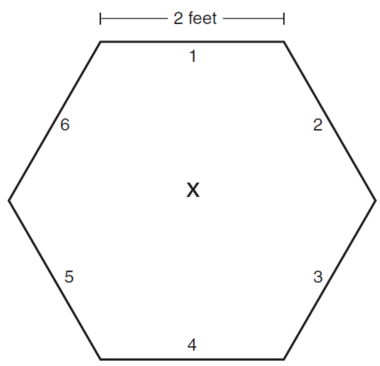


Figure 11: The hexagon test (Dawes and Roozen, 2012)

The athlete starts in the middle of the hexagon where they must always face forwards (in the starting position direction) (Dawes and Roozen, 2012). They must jump in and out of the hexagon with feet together in order from side 1 to side 6. The time is measured for the athlete to complete three full revolutions. Timing begins on command and stops at the end of the third revolution and back into the centre of the hexagon. The hexagon test should be performed both clockwise and counter-clockwise, so that both directions of travel are tested.

This test requires minimal equipment (just tape or chalk) and only a small space is required. The test is also conducted very quickly. Limitations to the test is that it is again, a pre-planned agility test such that it does not incorporate a stimulus, which can be reacted to by the athlete. This test may be more of a measure of jumping ability.

Pauole et al. (2000) found the Pearson product-moment correlation coefficient between the T-test and the hexagon test to be 0.42 and 0.48 for 152 males and 152 females, respectively. This implies that these two tests do not measure the same skills and attributes and are not correlated with one another.

2.3.8. Side Step and Edgren Side Step

The side-step has been thought of a method to measure agility, endurance and lateral speed (Barrow and McGee, 1979; Reiman and Manske, 2009). It involves two lines 12 ft (3.7 m) apart, marked every 3 ft (0.9 m) with a cone or tape (Figure 12), where a subject must assume a starting position with one foot over one of the lines (Barrow and McGee, 1979; Lacy, 2011). Upon command, they must side step with their leading foot towards the other line until the right foot crosses or touches the line. They then side step to the other line again and repeat with their opposing foot. The athlete must remain facing the same direction throughout the exercise, nor should their feet cross. The athlete must continue side stepping left and right as many times as possible for 30 seconds, with the number of line crosses equalling their score (Barrow and McGee, 1979; Lacy, 2011).

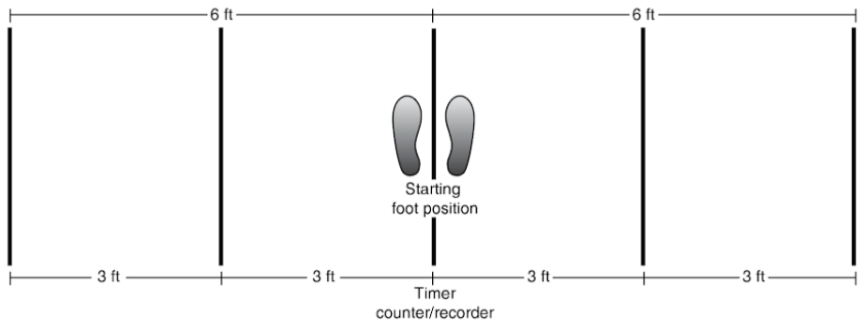


Figure 12: Side step test (Reiman and Manske, 2009)

The Edgren side step is a slight variation of the side-step, which is somewhat longer due to the use of metres instead of feet. In this version of the side-step, there are five cones placed one metre apart from one another (five metres total distance). The athlete must continue side stepping left and right as many times as possible for 10 seconds as opposed to the traditional 30 seconds (Johnson and Nelson, 1986; Reiman and Manske, 2009).

2.3.9. Reactive Agility Test

A reactive agility test (RAT) is a test that incorporates a stimulus, which the subject needs to react to; this could be in the form of a generic stimulus (visual light, arrow or audio), a video of a player exhibiting some kind of game specific movement, or a real-life human stimulus (Nimphius, 2014).

2.3.9.1. Traditional Y-Shaped Reactive Agility Test

In traditional RATs, the test is conducted by the subject running towards the stimulus, where they must change direction by running either left or right (at an angle of 45 degrees) based on the rules of the test. For example, the athlete may need to run towards the direction of the visual stimulus or direction the arrow is pointing (offensive play), or alternatively, they might need to run in the opposite direction (defensive play). This type of test can measure total test time, as well as usually measuring decision-making time too with a high-speed camera (Nimphius, 2014).

Haff and Triplett (2016) describe this test as the agility drill (Y-shaped agility); since the athlete must travel towards one of the cones either side placed at 45-degree angle, 2.7 m away; with the test forming a "Y" shape (Figure 13). The example configuration shows the stimulus will begin moving when the athlete reaches the starting timing gate, however the distance between the athlete and stimulus before it is presented is variable.

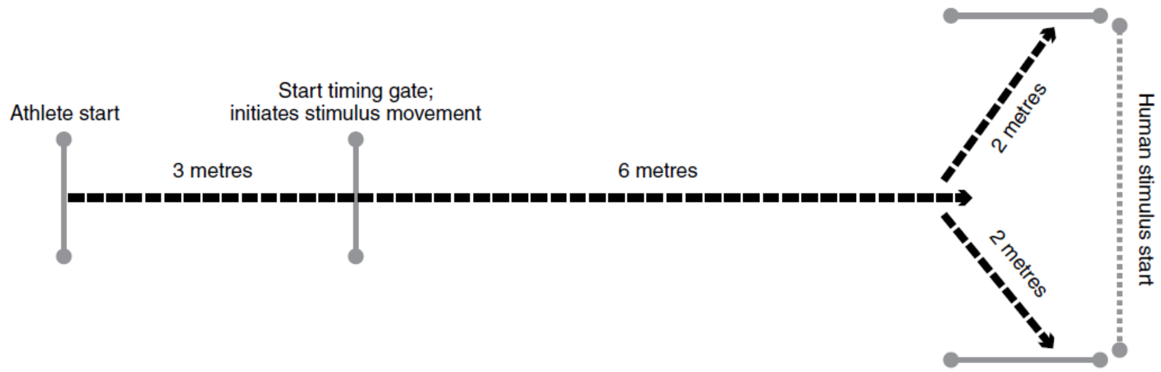


Figure 13: Example reactive agility test with a human stimulus (Nimphius, 2014)

A major benefit of this type of performance test is that it involves the perceptual and decision-making factors of a response to a stimulus, as well as the physical components of agility, where the athlete needs to use strength, power and balance to change direction quickly in response to a stimulus displaying CODS. Therefore, this test is a good measure of all aspects of agility performance. A limitation to this test is that it requires quite a lot of equipment to be accurate and reliable, as well as the need for a stimulus (visual/audio, projector for a video or human).

2.3.9.1.1. Reliability and Validity

Lockie et al. (2014) found a Y-shaped RAT differentiated semi-professional and amateur basketball players, while a pre-planned agility test did not; emphasising that planned and reactive agility are separate qualities and that agility tests should involve a perceptual and decision-making factor. Research has shown that highly-skilled athletes are better able to pick up and extract anticipatory cues of information, therefore are able to react faster in situations compared with lesser-skilled counterparts (Abernethy et al., 2001; Abernethy and Russell, 1987; Farrow et al., 2005). Therefore, the use of a reactive stimulus such as in the RATs should differentiate higher and lower skilled athletes.

2.3.9.2. Modified Reactive Agility Tests

Among the literature, it was found several studies had introduced some form of modified RAT (Farrow et al., 2005; Henry et al., 2011; Serpell et al., 2010; Spiteri et al., 2014; Veale et al., 2010). The variations in the tests that seemed to exist was implementing more reactive components (Spiteri et al., 2014), adding a side-stepping motion (Farrow et al., 2005), adding an additional running element at the end (Veale et al., 2010) or altering running distance (consistent in all).

Spiteri et al. (2014) implemented a modified RAT designed for basketball players, which involved two decision-making components (Figure 14).

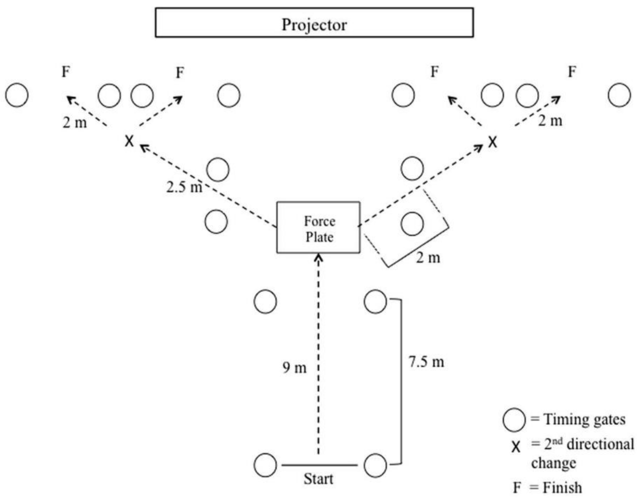


Figure 14: Modified RAT with two decision-making components (Spiteri et al., 2014)

The player needed to run towards the projector whilst simultaneously dribbling a ball. Upon arriving at a force plate, a random video (1 of 8 possibilities) displaying the perspective of a defensive player in an indoor basketball court presented a stimulus, causing the player to need to either make a fake, or change in direction left or right (45-degrees). After the initial COD, another random video stimulus was presented, where they then needed to either maintain possession of the ball, or pass it left or right.

Figure 15 shows how some of the other modified RAT tests varied from one another (Farrow et al., 2005; Veale et al., 2010). Veale et al. added a further running element and change-of-direction in their test designed for Australian football (Figure 15a). Farrow et al. added a side-stepping motion before approaching the reactive element in their RATs designed for netball (Figure 15b). This meant there was an added CODS involved in this test, which could possibly improve the validity that the test tests more components of agility.

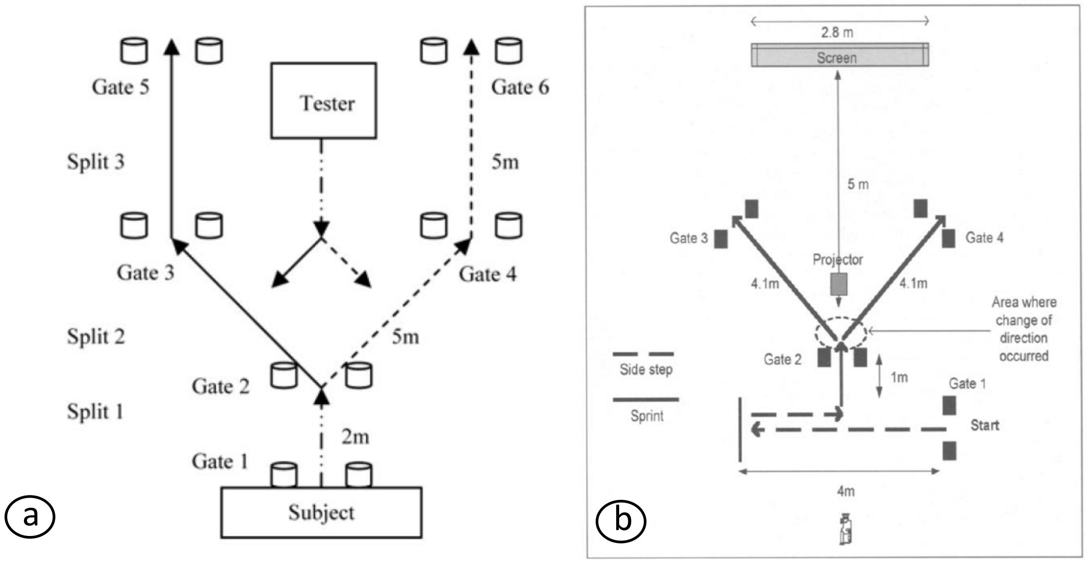


Figure 15: Modified RAT for (a) Australian football and for (b) netball (Farrow et al., 2005; Veale et al., 2010)

2.3.9.2.1. Reliability and Validity

Of the modified RATs conducted, each of the tests seemed to show relatively high correlation coefficients, with Spiteri et al. (2014) showing an ICC of 0.81, Farrow et al. (2005) showing an ICC of 0.83, Serpell et al. (2010) showing an ICC of 0.87 and Veale et al. (2010) showing a correlation of 0.91. Henry et al. (2011) showed the correlation coefficient between two RATs was 0.75, while the correlation between a planned and RAT was only 0.41- 0.68.

2.3.9.3. A Unique Reactive Visual Stimuli Agility Field Test

A study conducted by Benvenuti et al. (2010) aimed to assess the reliability of a reactive visual stimuli agility field test (RVS-T) and to compare agility performance of female soccer and futsal players with both planned visual stimuli agility field tests (PVS-T) and the RVS-T. The test consisted of four spheres, which were placed in a rectangle (Figure 16), lighting up when activated. The athlete would begin at sphere 1 and need to run to the sphere with a visual stimulus, touching it with their foot to turn it off and repeating for the remaining spheres. In the PSV-T, the sequence in which the spheres activated was known, while in the RSV-T, the sequence was unknown.

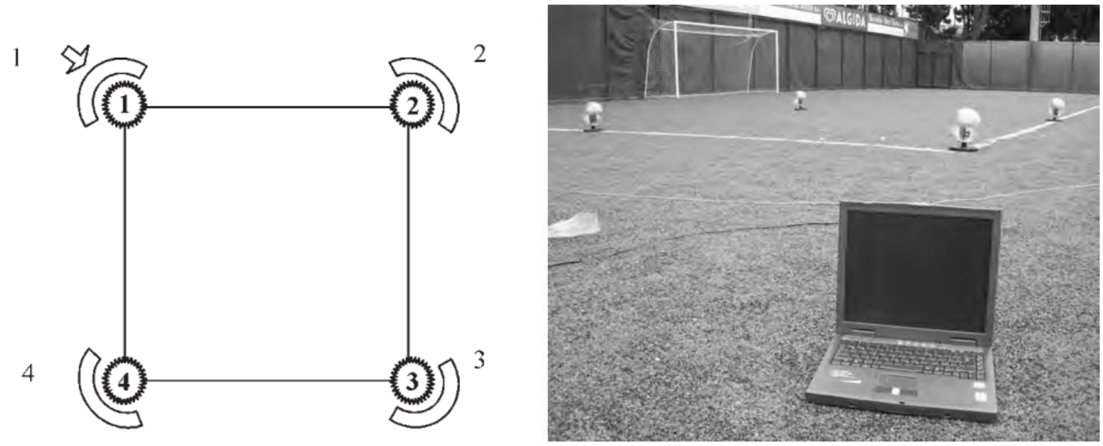


Figure 16: Diagram of sphere placement and experimental apparatus for the agility test (Benvenuti et al., 2010)

The study proved the RVS-T to be a reliable tool in evaluating agility in field conditions (ICC = 0.80), where significant differences in RVS-T performance was determined, with the futsal players outperforming their soccer counterparts. The study found the two groups performed similarly in the PVS-T, confirming agility performance is strongly influenced and altered in planned and reactive conditions. The proposed agility test in this thesis would use a similar concept, but would be further expanded, using six stimuli markers as opposed to four. Additionally, the visual cues would be anticipated to display a character, as well as present an auditory stimulus simultaneously. The athlete wouldbe expected to start in the centre, instead of at one marker, meaning there would be six possible locations to run to instead of three; increasing the element of randomness.

2.3.9.4. A Unique Stop'n'Go Reactive Agility Test

Sekulic et al. (2014) developed a new stop'n'go RAT (Figure 17). The way it worked is that an athlete would start at the start line, run forwards and this would trigger an infrared (IR) detector to randomly light up one of four LED lights on some cones at A, B C or D. The cone that was lit up had to be ran to and touched to deactivate it. The athlete would need to run back to the start line, touch it with their hand or foot and in doing this, they will have passed the IR to trigger the next LED light to turn on. They would then need to run towards this one and turn it off.

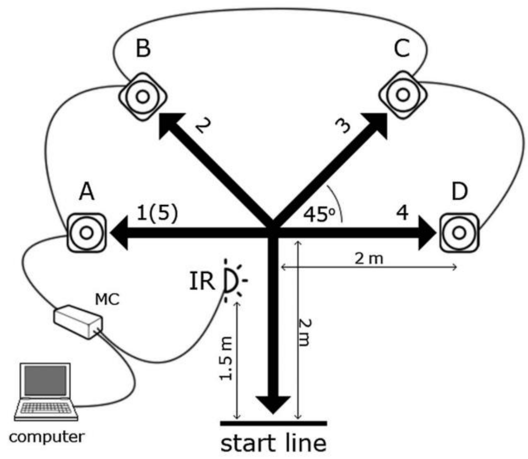


Figure 17: A Stop'n'Go reactive agility test (Sekulic et al., 2014)

There existed two version of the test presented by the authors; one involved a stop’n’go reactive component (SNG-RAT) while the other aimed to tests stop’n’go CODS (SNG-CODS). The SNG-RAT was found to have an ICC of 0.81 and 0.86 for men and women, respectively; the SNG-CODS had ICC values of 0.87 and 0.92 for men and women, respectively. This study made use of comparing both reactive and CODS components; this could be employed into the proposed thesis if time permitted. If not, once the test is complete, there is always the possibility of investigating this further.

This stop’n’go test described is similar to the proposed agility tester in some respects, such as the random activation of LED lights and implementing both CODS and a reactive component. An obvious limitation to the stop’n’go test is that it consists of a lot of wires (not wireless) and required the LED lights to be positioned on top of a cone. The positioning of the proposed agility test markers may be significantly different. They could be dispersed in a different pattern and may include more than four markers as employed in the stop’n’go test. It is likely that the proposed agility test will have the athlete start in the centre of the markers so that they are required to visually scan in all directions of view, instead of constantly beginning at a start line in the stop’n’go test (Figure 17).

2.3.9.5. Reliability and Validity

A detailed systematic review of the testing, training and factors affecting performance in agility in team sports was conducted by Paul et al. (2016). Their findings showed agility tests involving a light, video or human stimuli had high reliability and were generally considered valid (with ICC values of 0.80-0.91, 0.1-0.81 and 0.81-0.99, respectively); perceptual and decision-making factors were found to be the primary discriminant factor distinguishing highly-skilled athletes to lesser-skilled athletes. This was apart from a video stimulus in youth athletes, which offered a low-level reliability (ICC = 0.1-0.3).

2.3.9.6. Mimicking Sport-Specific Stimuli

Paul et al. (2016) recognised that an athlete processing complex motion during dynamic play in team sports differs from the requirements of reacting to a stationary light that can only be on or off. They concluded that agility tests involving both a physical and cognitive stimulus are likely to provide larger improvements. Suggestions were made for future development of new agility tests, to incorporate sport-specific agility scenarios, which employ the complex movements and decision-making aspects; inclusion of a ball or other sporting equipment, a variety of views, multiple players, differing movements (defending and attacking) and deceptive actions.

Ideally, the proposed agility test in this thesis would incorporate a range of these components to deliver the most dynamic test which may mimic as close as possible, the holistic whole-body sports-specific movements encompassed during play. However, realistically, it would be extremely difficult to design an agility test that could incorporate all or most of these features and remain valid, reliable and able to be standardised across a range of unique sports. This is exceedingly challenging as the sport-specific physical and cognitive-perceptual requirements across sports could vary significantly. In addition, the time constraints imposed by the project and inherent technological limitations due to funding and knowledge base would interfere with development of such test. Thus, to aim to design a test which can remain both valid and reliable, as well as be able to become standardised across a variety of sports, the proposed agility test intends on an inclusion of sport-specific equipment (such as a ball) which will be able to be integrated with the devices themselves. This integration of sporting equipment within the test, whilst remaining standardised, will boast a system which is one step ahead of a majority of tests out there, which normally only use a few a set of cones or cannot remain standardised with the use of sporting equipment.

The test will be implemented so the athlete will need to use strength, speed and accuracy to produce rapid, whole-body movements in response to the random stimuili presented. The athlete will need to be aware of their surroundings as well as excel in anticipation, visual scanning and reaction time. The beginning of the test has the possibility that an activated marker could occur at any location in a complete 360^{o} of view of the athlete, meaning a variety of views will be required.

The test could also be developed to employ deceptive stimuli to further assist in an athlete mimicking the whole-body movements likely performed in their sport as well as allowing integration of another form of perceptual and decision-making factors. For example, a marker could activate, but then self-deactivate shortly after, with the activation of a new stimuli from a different marker. This could occur once or more times in each test. It is unlikely this feature would be incorporated into this iteration of the system, however, is something which could be considered in future work.

Recall earlier, that Paul et al. (2016) found the ICCs of RATs incorporating a visual stimulus were 0.80-0.91, compared with ICCs of 0.1-0.81 and 0.81-0.99 for video and human stimuli, respectively. This study found that RATs with a human stimulus were most optimal and would be most valid and reliable for an agility test, with highested ICCs. A light stimulus in the RATs contained the next best ICCs, with some values within the range of a human stimulus. This shows that the use of a light stimuli is still significantly justified, as inclusion of human stimuli into the test would make it very difficult to remain valid, reliable and standardised across various sports and differing sporting clubs and institutions. The author does acknowledge that this study was conducted on RATs, whereas the proposed test will differ quite significantly in terms of test layout, structure, how it is performed. This could cause difficulty in translating the findings from Paul et al. (2016) to the new test.

Although the proposed agility test may not entirely mimic sport-specific stimuli through the physical and cognitive-perceptual requirements imposed on the athlete, research has shown some form of stimulus is still practically appropriate (Lockie et al., 2014). Given the circumstances and nature of the project, the choice of using a light and audio stimulus has been justified in the proposed test.

2.3.10. Agility Test Classification

Based on the above analysis for each of these "agility" tests, they can be classified as a test for either CODS, maneuverability, perceptual-cognitive ability or a combination of them (Table 7). The tests have been classified based on Haff and Triplett (2016) and Nimphius (2014). A large majority of these so-called agility tests are measures of CODS or manoeuvrability and it is only the RATs that contain a perceptual-cognitive component.

Table 7: Classifications of CODS and Agility Tests (Modified from Haff and Triplett, 2016; Nimphius, 2014)

Test	Change-of-Direction Speed (CODS)	Manoeuvrability	Perceptual-Cognitive Ability
Illinois Agility Test	✓	✓	
AFL Agility Test	✓	✓	
5-0-5 Agility	✓		
T-Test	✓	✓	
Pro Agility	✓		
3-Cone Drill	✓	✓	
Hexagon	✓		
Side Step	✓		
RAT (Light or Arrow)	✓		✓
RAT (Video)	✓		✓
RAT (Human Stimulus)	✓		✓

The current literature has shown a large number of tests claiming to measure agility performance in the field and on the court, however a large majority of "gold standard" tests do not require the athlete to use any perceptual or decision-making factors; they only are a measure of CODS since the tests are pre-planned and the athlete knows the direction to travel before beginning (Coulson and Archer, 2009; Haff and Triplett, 2016; Lockie et al., 2014; Nimphius, 2014). As a result, some authors are mentioning that these kind of tests should not be called agility tests, but referred to as CODS tests since they involve closed skills only (Haff and Triplett, 2016). A major limitation in these types of tests is that the athlete can practise them continuously, such that they can master the required movements and repeat it with automation without needing to use any perceptual-component. Although many of the tests have been proven to be able to differentiate players of different ability levels (Stewart et al., 2014; Till et al., 2015), they are not a true measure of agility performance and it is the RATs or other tests involving perceptual and decision-making factors that should be the only tests considered to tests all components of agility. Nimphius (2014) mentions how an agility-testing battery should include both CODS test and RATs, as RATs only require a 45-degree turn and a more physically demanding CODS needs to be implemented. The proposed agility tester will likely implement all kinds of angled turns, causing a variety of CODS, which could potentially fulfil the needs mentioned by Nimphius. As shown in this section, it is possible to test both CODS and cognitive components of agility simultaneously by RATs (Haff and Triplett, 2016; Nimphius, 2014). Inglis and Bird (2016) discussed how RATs are a more reliable and valid in assessing agility compared with traditional pre-planned and light agility drills. Tanner et al. (2013) also mentioned how RATs are thought to be much more game-specific and most likely correlate with actual game performance better. Therefore, it is important to implement a test that tests CODS, as well as the perceptual-cognitive aspects of agility performance using a reactive component. The proposed agility tester intends on incorporating both CODS and cognitive factors, which will make it differ from majority of these pre-planned agility tests.

2.4. Competitive Market Analysis

As with all new products, a competitive market analysis was undertaken to determine currently available agility testing devices that may be considered a threat to the proposed product. The primary goal of this section of the literature review is to recognize component features presented by current industry products and to establish the gap in the market of these devices. These systems are introduced individually, then further analysed and compared later in this section.

2.4.1. The Competing Products

After undertaking thorough research, the devices that appeared to be of the highest threat included the FITLIGHT Trainer™ (FITLIGHT Sports Corp., Ontario, Canada), Wireless Training Timer SEM (WittySEM) (Microgate Corporation, Bolzano, Italy), SMARTfit® Strike Pods (Smartfit Inc., Camarillo, California) and photocell timing gate incorporated systems; SmartSpeed Pro (Fusion Sport, Brisbane, Australia) and SpeedLight (Swift Performance, Brisbane, Australia). The Freelap Timing System (Freelap SA, Fleurier, Switzerland) was also a considerable threat, as well as traditional photocell timing gate systems such as the TC Timing System (Brower Timing Systems, Draper, UT), Wireless Timing Network (ALGE-TIMING GmbH, Vienna, Austria) or the Wireless Race Timing System (TAG Heuer, La-Chaux-de-Fonds, Switzerland). The Powerdash and Agility Timers (Zybek Sports, Broomfield, CO) were considered low threats, being significantly less than the other timing gates for reasons discussed in Section 2.4.5 when comparing and analysing the products in detail.

2.4.2. Reactive Component Focused Products

The FITLIGHT Trainer™ (Figure 18), Witty SEM (Figure 19) and SMARTfit® (Figure 20) are all systems designed primarily to be used in reactive training.

Figure 18: FITLIGHT Trainer™ (FITLIGHT Corp., 2018)

Figure 19: Witty SEM (Microgate, 2015)

Figure 20: SMARTfit® Strike Pods (Smartfit Inc., 2018)

Each system has the capability of presenting a visual stimulus using a light-emitting diode (LED) display with all LEDs able to emit red-green-blue (RGB) light. SMARTfit® and Witty SEM have a LED matrix capable of displaying any character (numbers, letters symbols), while the FITLIGHT Trainer™ is a simple circular LED array. Each device incorporates a reactive test, where the devices will activate randomly, requiring the athlete to deactivate it as quickly as possible; thereby reactivating another device. The individual components can be spread out and set-up to the liking of the test administrator, allowing high customisation and adaptability of tests. Extra accessories would be required to use the Witty SEM and FITLIGHT Trainer^{TM} for pre-planned agility tests, where the SMARTfit® does not have any capability to run pre-planned agility tests.

With SMARTfit® and Witty SEM able to display a multitude of characters, this means further customisation of tests to be conducted (e.g. find a letter amongst other alphabet letters) could be used in a rehabilitation sense. However, both products only work best when mounted to a tripod, otherwise the display is not easily viewable. The SMARTfit contains a touch sensor, the Witty SEM a proximity sensor and FITLIGHT Trainer^{TM} contains both a touch and proximity sensor, which could be useful in further testing applications. As the devices are so small, they easily fit into a carry case or bag. These products seem to display characteristics similarly to what has been proposed for the agility testing being designed and developed in this thesis

2.4.3. Proximity Timing Gate

The Freelap Timing System (Figure 21) uses proximity for athletes to be timed. It works via electromagnetic fields being emitted by the TX Junior Pro transmitter (a cone-like structure) (Figure 21a), received by an FxChip (Figure 21b), which detects this field. By the athlete wearing the FxChip, a timing measurement can be recorded as the athlete with the chip comes into proximity to the TX Junior Pro. The TX Junior Pro can be configured as either the starting cone, finishing cone or a lap cone. When a timed recording is complete by an athlete passing the final cone, an audible sound is heard. A relay (Relay Coach BLE) collects the data wirelessly and transmits it to a smartphone or tablet using Bluetooth. This system does not contain any sort of reactive component, so can only be used for pre-planned agility tests. All equipment is capable of being contained within a custom bag (Figure 21c).

Figure 21: Freelap Timing System. (a) TX Junior Pro. (b) FxChip. (c) Entire system within containment (Freelap SA, 2018)

2.4.4. Photocell Timing Gates

2.4.4.1. Basic Theory

Electronic photocell timing systems used in traditional timing gates remain the gold standard for the accurate and reliable assessment of athlete performance (Earp and Newton, 2012; Haugen et al., 2014; Stanton et al., 2016). Put simply, they generally work via emitting an infrared light beam from a photocell emitter, which is received on a reflector, reflecting the beam so that it can be detected by the photocell sensor; the time is recorded when there is an interference of the beam (e.g. by an athlete passing) (Yeadon et al., 1999).

2.4.4.2. Products

SmartSpeed (Figure 22a) and SpeedLight (Figure 22b) are two photocell timing gates which contain an incorporated reactive component. These systems use the traditional photocell timing system, so require an emitting, receiving and reflecting component (two components). These systems can provide a visual stimulus that can be viewed a full 360 degrees, as well as provide an audio stimulus.

a SmartSpeed **(Fusion Sport, Brisbane, Australia)**	b SpeedLight **(Swift Performance, Brisbane, Australia)**

Figure 22: (a) SmartSpeed. (b) SpeedLight (Fusion Sport, 2018; Swift Performance, 2018)

Traditional photocell timing gates that do no incorporate a reactive component analysed in this literature review are the TC Timing System (Figure 23a), Wireless Timing Network (Figure 23b), Wireless Race Timing System (Figure 24a) and Powerdash/Agility Timer (Figure 24b). These systems are excellent when it comes to recording time measurements, however, without the use of additional accessories or equipment, they are unable to incorporate a reactive component.

a TC Timing System **(Brower Timing Systems, Draper, UT)**	b Wireless Timing Network (WTN) **(ALGE-TIMING GmbH, Vienna, Austria)**

Figure 23: (a) TC Timing System. (b) Wireless Timing Network (ALGE-Timing GmbH, 2018; Brower Timing Systems, 2018)

a **Wireless Race Timing System** (TAG Heuer, La-Chaux-de-Fonds, Switzerland)	b **Powerdash/Agility Timer** (Zybek Sports, Broomfield, CO)

Figure 24: (a) Wireless Race Timing System. (b) Powerdash/Agility Timer (TAG Heuer, 2018; Zybek Sports, 2018)

2.4.4.3. Reliability

Haugen et al. (2014) and (Yeadon et al., 1999) demonstrated greater accuracy in dual-beam systems, as opposed to single-beam systems (±0.06 seconds), since they can be falsely triggered by swinging arms or lifted knees. Both authors recommended the use of dual-beam for scientists and practitioners who need accurate and reliable results. The Swift Performance SpeedLight implements dual-beam technology, which shows its measurements are highly accurate and reliable. Fusion Sport's SmartSpeed range uses a single-beam system, however, implementation of an error correction processing algorithm allows the elimination of false triggering errors. D'Auria et al. (2006) showed the system complies with Australia's National Sport Science Quality Assurance standards (maximum typical error 0.05 s over 30 m) by showing a typical error ≤0.03 s for 5, 10 and 20 m. Earp and Newton (2012) found that single-beam systems with signal processing had higher accuracy than dual-beam systems with no signal processing, supporting the reliability of use of the SmartSpeed Pro. All other photocell timing gates mentioned (TC Timing System, WTN, Race Timing System and Powerdash/Agility Timers) implement a single beam photocell, but it is unknown if signal processing is used to minimise false triggering error. These systems are used in competitive sporting and research applications, which suggests a high degree of reliability and accuracy in their measurements.

2.4.5. Device Comparison and Analysis

A detailed market analysis comparison table was developed (Table 8) containing key features offered by each product, so comparisons could be made easily through visual ticks or crosses. Analysing and comparing simple set-up was originally included, however this component of the system is quite subjective and dependent on the user experience, so was subsequently removed. Zybek Sports' Powerdash/Agility Timers were not included in the competitive market analysis table, since there were found to be a significant number of features not provided by the system, including not being wireless, presenting a visual or audio stimulus, no assisted set-up, phone/laptop compatibility, reactive agility test capabilities, unique athlete tagging, integration of sporting equipment, multiple testing options/difficulties or use in a rehabilitation setting. It should also be noted the TC Timing System, WTN and Race Timing System were grouped together as their features were consistent with one another.

Table 8: Competitive market analysis feature comparison

Product	FITLIGHT Trainer	Witty SEM	SMARTfit Strike Pods	SmartSpeed Pro	SpeedLight	Freelap Timing System	TC / Wireless / Race Timing System	The Proposed Agility Tester Developed in this Project
Company	FITLIGHT Corp.	Microgate	Smartfit Inc.	Fusion Sport	Swift Performance	Freelap SA	Brower Timing Systems/ ALGE-Timing/ TAG Heuer	Flinders University/ SHAPE Research Centre
Portable/lightweight	✓	✓	✓	✓	✗	✓	✓	✓
Usable indoors and outdoors	✓	✓	✓	✓	✓	✓	✓	✓ ^
Wireless	✓	✓	✓	✓	✓	✓	✓	✓
Measures time	✓	✓	✓	✓	✓	✓	✓	✓
Visual stimulus	✓	✓	✓	✓	✓	✗	✗	✓
Audio stimulus	✓	✗	✓ *	✓	✓	✗	✗	✓
Assisted set-up	✗	✗	✗	✓	✓	✗	✓	✓
Assisted test layout set-up	✗	✗	✗	✗	✗	✗	✗	✓
Phone or computer compatible	✓	✗	✓	✓	✓	✓	✗	✓
Best use independent of other equipment	✓	✗	✗	✓	✓	✓	✓	✓

Product	FITLIGHT Trainer	Witty SEM	SMARTfit Strike Pods	SmartSpeed Pro	SpeedLight	Freelap Timing System	TC / Wireless / Race Timing System	The Proposed Agility Tester Developed in this Project
Reactive agility tests	✓	✓	✓	✓	✓	✗	✗	✓
Does not require timing gate set-up	✓	✓	✓	✗	✗	✓	✗	✓
Real-time data feedback	✓	✓	✓	✓	✓	✓	✓	✓
Usable in variety of sports	✓	✓	✓	✓	✓	✓	✓	✓
Integration of sporting equipment with the device itself	✗	✗	✗	✗	✗	✗	✗	✓
Self-initiated/ automatic start	✗	✗	✗	✓	✓	✓	✓	✓
Pre-planned agility tests	✓ *	✓ *	✗	✓	✓	✓	✓	To be implemented/trialled in future work
Unique athlete ID tagging	✗	✓ *	✗	✓ *	✓ *	✓	✗	To be implemented in future work
Customisable test options	✓	✓	✓	✓	✓	✓	✓	To be implemented in future work
Impact resistant	✓	*Unknown*	✓	✓	✓	✓	✓	To be implemented in future work
Use in rehabilitation setting	✓	✓	✓	✓	✓	✗	✗	To be implemented in future work
Accuracy/Precision (s)	0.001	0.001	0.01	0.001	0.01	0.02	0.001 (TC) / 0.0001 (Wireless & Race)	*To be determined in future work*

** When integrated with equipment accessories*

^ Not applicable in all circumstances

2.4.5.1. Common Feature Considerations

Upon analysis of the competing market, it seems that all products posing as a significant threat are usable indoors and outdoors, are wireless, measure time, provide real-time data feedback, are usable in a variety of sports as well as have tests that are customisable. In a sporting application, it is vital that the product is capable of operating both indoors and outdoors, where field and court sports occur; such that an athlete is tested in the sport's natural environment. Having a wireless system is important because it provides ease of access, minimises trip hazards and is more aesthetically pleasing to the user. Measuring time is a given necessity, as it is the most logical method for distinguishing the results obtained between athletes when performing the test. Real-time data feedback provides ease of use, while using the system with a variety of sports is of great importance too because it opens the market right up to those who can use the product. Additionally, being able to run through a test that contains some form of element of the actual sport is very beneficial. It is important to be able to have a product that is adaptable such that the user or administrator can change how the system is set-up or how a test is run to change their testing strategies and incorporate new training regimes. All products that implement a reactive component had this possibility

2.4.5.2. Portable/Lightweight

The need for a portable system is essential with sporting applications, as the product needs to be taken onto the field or court to test athletes in an environment representative of their sport. All systems were portable, in such a way that they can be packed away into either a bag or carry case as seen in the images shown, except for Swift Performance's SpeedLight. Although the system is transportable, it does not come with either bag or carry case, meaning transport may be difficult.

2.4.5.3. Visual & Audio Stimulus

Most systems implemented at least a visual stimulus, with exception of the Freelap Timing System and the traditional timing gates. The visual stimulus is a key component for reactive agility testing, which is incorporated in all systems (with a visual stimulus) using an LED display. The FITLIGHT Trainer™, SMARTfit® Strike Pods SmartSpeed and SpeedLight all have the capability of coupling the visual stimulus with an audio stimulus too for reactive applications. However, the SMARTfit® Strike Pods audio stimulus comes from the main controller, so the individual pods do not make a noise; this could argue that this system does not have an audio stimulus. Some systems contained an audible sound such as when assisting set-up or completion of a test, but this is not the context being analysed. The simultaneous use of both visual and audio stimulus is important to implement, so that two of the athlete's sensory pathways are being used.

2.4.5.4. Assisted Set-Up

The SmartSpeed, SpeedLight and traditional timing gates all had assisted set-up in-built into their systems. This may be in the form of an LED indicating the correct positioning (Fusion Sports' SmartSpeed, ALGE-Timing's WTN, TAG Heuer's Race Timing System), laser aligning pointer (Swift Performance's SpeedLight), or an audible sound (Brower Timing Systems' TC Timing System). Assisted set-up is vital for ease of use for the test administrator. When the equipment assists in set-up, it also minimises the chance of human error and thus creates further standardisation.

2.4.5.5. Phone or Computer Compatible

With the modern use of smartphones, implementing a system compatible of interacting with a phone makes ease of use for the user better. The Witty SEM is not compatible with phones or computer, as it contains its own console (Witty Timer), while the FITLIGHT Trainer™ is only compatible with Android smartphones or tablets. All the traditional timing gates are not phone or computer compatible, as they contain their own equipment to see the test results. With a requirement to use the specified equipment provides a restriction to the usability of the system, as every user may have their own preferences which may only be achieved through a smartphone or computer.

2.4.5.6. Best Use Independent of Other Equipment

All competing products seemed to have their best use independent of other equipment or accessories that are not supplied with their basic packages. However, this was not the case for the Witty SEM or SMARTfit® Strike Targets. The Witty SEM only has its best use when implemented with other equipment such as its Witty GATE (photocell timing gate). Without the SMARTfit Strike Target accessories, they must lay flat on the ground, which does not have an optimal view of the LED display, especially when trying to observe characters. Even with their height mounting system, they still need a weight placed on this to stop it from falling over.

2.4.5.7. Reactive Agility Tests

A reactive component of the system is of great importance, as to truly measure an athlete's ability, there must be some form of perceptual and decision-making factors as already emphasised throughout this literature review. As photocell timing gates are only able to measure time, this means they can only be used in tests that required planned CODS when used independently; no reactive component exists and thus there are no perceptual and decision-making factors. The Freelap Timing System also did not contain any form of reactive component and is therefore only able to complete CODS tests. Apart from these systems, all other competing products seemed to have some form of reactive component implemented, with a minimum using a visual stimulus, but some with both visual and audio stimulus.

2.4.5.8. Integration of Sporting Equipment with the Device Itself

As noted, it was consistent among the devices that they could run tests for a variety of sports, where many of the systems needed extra accessories to do so. Although it is possible to incorporate sporting equipment or balls when testing the devices (e.g. dribbling a basketball or football), none of the devices incorporated the sporting equipment into the device itself. The proposed agility tester plans on requiring the athlete to remove a ball from the devices which will be the mechanism for deactivating a marker. This is a distinguishing feature of the proposed agility tester, which makes it differ from all analysed products. Furthermore, as with the devices analysed, it will also have simple incorporation of sporting equipment such as the athlete dribbling a ball if desired during training regimes.

2.4.5.9. Self-Initiated/Automatic Start

A self-initiating/automatically starting test provides ease of use for the administrator and for the athlete. It was found the FITLIGHT Trainer™, Witty SEM and SMARTfit® system are not capable of self-initiating their tests and require the test to be manually started. All products that use a timing gate have the capability of an automatic start, as it can detect when the athlete has ceased interference of the photocell emitter/receiver beam.

2.4.5.10. Precision/Accuracy

Of the competitive products, the WTN and Race Timing System were found to have the highest degree of accuracy, being 0.0001 seconds. A common accuracy of 0.001 was found in four of the devices; the FITLIGHT Trainer™, Witty SEM, SmartSpeed and the TC Timing System. SMARTfit® and SpeedLight had accuracy of 0.01 seconds, while the worst accuracy was from the Freelap Timing System being 0.02 seconds. With 0.01 and 0.001 seconds being the consistent precision accuracy values, the proposed agility test needs to have this accuracy or better to compete against current products.

2.4.5.11. To be Implemented in Future Work Features

2.4.5.11.1. Pre-Planned Agility Tests

It was found all systems except for the SMARTfit Strike Pods were able to conduct pre-planned agility tests such as the Illinois Agility Test, 5-0-5 Drill, T-Test or Pro Agility. Designing the proposed agility tester to be able to run pre-planned agility tests was not a part of the scope of the project nor an initial desirable feature, since it was established agility involves some form of perceptual and decision-making factor. However, after analysis of existing market products, it should be considered implementing this feature into the system down the track, because it would keep the market open to those who still believe that pre-planned agility tests are a valid and reliable form of measurement for agility.

2.4.5.11.2. Unique Athlete ID Tagging

Although the proposed agility tester does not incorporate unique athlete ID tagging (e.g. using radio-frequency identification - RFID), this was not a feature as part of the scope of this project. Upon analysis of competing products, it should be noted that only the Freelap Timing System can implement a unique athlete tagging feature without the use of additional equipment; Witty SEM requires Witty RFID and SmartSpeed requires SmartScan, SpeedLight requires SpeedReader. Therefore, the proposed agility tester will not contain the feature of unique athlete tagging, which can be substantiated in the fact that majority of competing products do not contain this characteristic independent of extra accessories. Additionally, since the test is intended for one athlete to be tested at a given time, there is not really a need for athlete ID tagging. However, considerations for future work of the proposed agility tester may end up incorporating unique athlete tagging when it is desirable to increase the set-up procedure time.

2.4.5.12. Impact Resistant

Performance testing of athletes is very physically demanding, where athletes are going to use high forces of strength and power to get from one position to the next. As a result, maximum effort could cause error in anticipating movement and as a result could potentially knock over the equipment. At high speeds, a lot of momentum could really impact the equipment, damaging the housing or internal electronic hardware. As a result, it is important that the equipment contains some form of impact resistance. It is unlikely that this would be able to be achieved in this project, however, should certainly be considered in future work. All existing market products were found to be impact resistant, however this had to be assumed with the Witty SEM being used in research and sporting applications.

2.4.5.13. Use in Rehabilitation Setting

All products that incorporate a reactive component have the capability of being used in a rehabilitation setting. Although not part of the scope of the proposed agility tester, this is a feature that could certainly be implemented into the future.

2.4.5.14. Other Considerations

After testing the Freelap Timing System first-hand, there were several major drawbacks observed. One aspect is that audio feedback is only heard once the athlete has finished their lap. This is a major problem, especially coupled with other problems where standing too close to the first signal can trigger the timer to start unknowingly or standing too far away may not trigger the timer to start. This means that an athlete may use maximal effort on a test run that had not recorded or may have had the timer start too early. This means the athlete has wasted energy and would need to wait to recuperate their energy, otherwise they may not be able to perform at their best if the test were to be completed again too soon.

From careful analysis of the competing products in the current market, there is no systems that can successfully implement all features described, let alone complete most without the use of additional equipment or accessories. Although the devices were able to be used with a variety of sports, in a way where athletes could dribble a ball, or hold some equipment, there was no form of standardisation if implemented. Another major gap in the market of all systems was that they could not incorporate sporting equipment or balls into the device itself. The proposed agility tester plans to force the user to remove the equipment as they would in their profession (e.g. pick up and pass like in basketball or netball or a handpass in AFL), which would subsequently initiate the activation of the next stimulus.

2.5. Literature Review Conclusions

Agility has been an ambiguous term used in sports science, with no consistent definition or classification among the current literature. Sheppard and Young (2006) suggested that agility involves a multitude of components involving both physical and perceptual and decision-making factors. Emerging literature is now in agreeance that agility involves a cognitive component in conjunction with physical qualities. Athlete testing is a significantly useful method to evaluate the performance of athletes; acting to support improvement by allowing goals to be set and progress to be evaluated as well as differentiating higher-performing athletes from their lower-performing counterparts. As agility involves both physical and perceptual and decision-making factors, many standardised, well-established “agility” tests such as the Illinois agility, AFL agility, 5-0-5, T-test, Pro-agility, 3-cone drill and hexagon used in athlete testing combines and batteries clearly involve pre-planned agility only. This implies they are change-of-direction tests only; the missing cognitive component of the tests therefore creates uncertainty of the validity of these tests to evaluate agility. RATs have been developed, which incorporate both change-of direction speed as well as the response to a stimulus, providing a true evaluation of agility as it addresses both physical and cognitive components. However, there is no consistency in application of the RATs (multiple varying factors), with no current standardisation to the implementation of the test.

A competitive market analysis found products to be of significant threat included the FITLIGHT Trainer™ (FITLIGHT Sports Corp., Ontario, Canada), Wireless Training Timer SEM (WittySEM) (Microgate Corporation, Bolzano, Italy), SMARTfit® Strike Pods (Smartfit Inc., Camarillo, California) and photocell timing gate incorporated systems; SmartSpeed Pro (Fusion Sport, Brisbane, Australia) and SpeedLight (Swift Performance, Brisbane, Australia). It was found the major gap in the market was incorporating sporting equipment or balls with the system itself. Additionally, if using sport-specific equipment alongside the system, there was no form of standardisation to be used between clubs and institutions. Another limitation was that each system needed some form of accessory, such as a tripod or mount, which usually needed to be bought separately. The proposed agility tester is planned to evaluate all components of agility (physical and perceptual and decision-making factors), as well as have components that are stand-alone and provide best use without an accessory or extra equipment. The test will be capable of integrating the sport-specific equipment or balls with the test such that the player must remove it to deactivate it and thereby activate the next random stimulus; providing a novel technology to the market. Although this thesis will not have an industry-ready system by the end of the project, a proof-of-concept prototype will be developed, showing that it is able to incorporate all of these features; acting to pave the way towards developing a test that could possibly become a new standardised way of evaluating athlete agility performance across sporting clubs for use with a variety of field and court sports and athletic abilities and to be used in test batteries and combines.

3. Engineering Design Process

This project had the primary aim of designing and developing a new, novel system capable of testing an athlete's agility. It started at the roots of a simple idea and developed into a physical product; thus, this thesis aims to provide a detailed description in the design processes executed to implement such a system. At the start of the project, it was known the project would involve mechanical, electrical and software engineering; naturally, it made sense to follow a design process within the field of engineering.

Before selecting an appropriate design process model to follow, it was important to take a step backward by investigating some existing models developed and the similarities between them. This ensured that the best design process was to be followed for this project. Many design process models have been established in publications where Pahl et al. (2007) lists over one hundred since 1953. Some of the most notable design models are from Roth (1965), French (1971), Verein Deutscher Ingenieure (1993), Pahl et al. (2007) and Haik and Shahin (2011) and (Dym et al., 2014).

Roth simply divided the process into a problem formation phase, a functional phase and an embodiment phase. Similarly, (Dym et al., 2014) also sectioned their design process to include a task formulation phase and a functional phase. When expanded, the model by Dym et al. presented a prescriptive design process model that entailed a problem definition, conceptual design, preliminary design or embodiment of schemes, detailed design and design communication. This aligns with both French (1985) and Pahl et al. (2007), which base the stages of their models on analysing the problem and clarifying the task (involving planning and specifying information), conceptual design (specification of a principle solution), embodiment of designs (specification of layout) and design detailing (specification for production). Haik and Shahin (2011) also presented a similar design process of requirements (market analysis, needs and requirement development), product concept (functions and specification), solution concept (conceptualisation and evaluating alternatives), embodiment and design detail (analysis, simulation and experimentation). Verein Deutscher Ingenieure (VDI), a professional engineers' society from Germany have produced various guidelines, such as the VDI 2221: Systematic Approach to the Design of Technical Systems and Products, where this guideline suggests a systematic approach is to be followed, first by analysing and understanding the problem, breaking it down into sub-problems, finding suitable sub-solutions and then integrating these into an overall solution (Verein Deutscher Ingenieure, 1993). From these examples, there is a clear logical flow of stages; Thompson and Lordan (1999) summarised that design process models contain the same basic elements, including an analysis of need, generation of ideas, evaluation, schematic design and detail design.

These examples are only just a mere few of the many design processes out there. However, it is clear to see that each one contains a clear, logical and systematic flow of elements that entails recognising the problem or need at hand, clarifying the task, completing market research, determining the design requirements and specifications, conceptualising, refinement and evaluation of ideas, design embodiment, followed by detailing the design for a final product. This compiled design process structure has been illustrated for a better visual representation (Figure 25). This general structure was followed when undertaking this project.

It should be noted that an engineering design process is often applied through an iterative process as represented in the figure, where past stages are returned to. Through this, design definitions, conceptualisation, development and implementation is often repeated until the best solution is presented so that optimisation of the design is achieved, with needs and specifications met (Cross, 2000; Matthews and Institution of Mechanical Engineers, 2012).

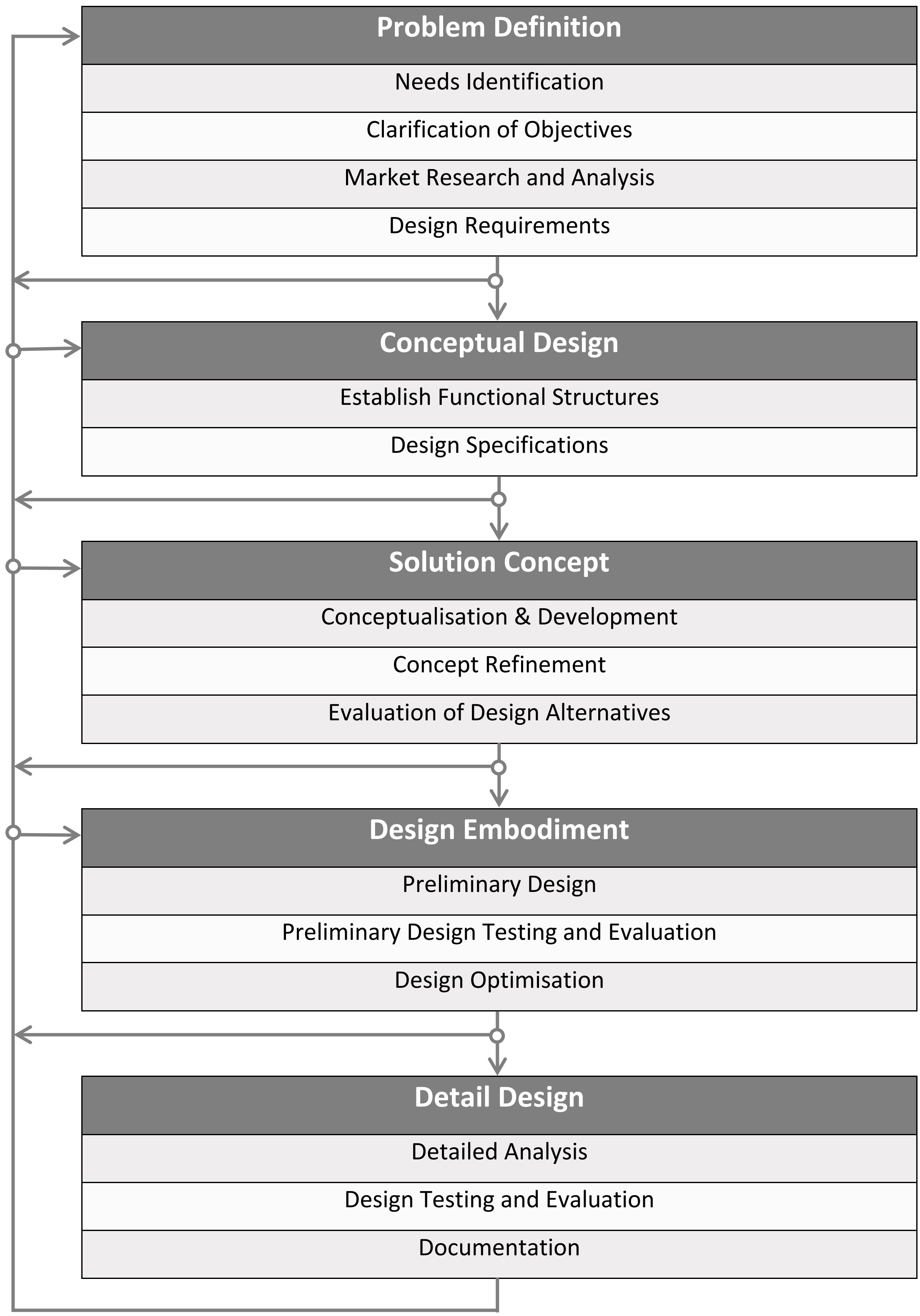


Figure 25: Design process followed for this project

3.1. Problem Definition

To be able to design and develop a product design solution, one must first establish what the problem is to be solved and the need behind why it needs to be solved. By understanding these fundamental factors, a successful design can be developed to solve the problem according to the functional needs. It is this very reason why this is one of the most important stages of the design process, as anyone can make a new design, but for it to be a success in contemporary society, the solution must be able to solve an existing problem with a true need for the design to exist.

3.1.1. Market and Research Analysis

A thorough market and research analysis was conducted over the course of the first stage of the project. This was through the literature review and was discussed in-depth in Section 2.

3.1.2. Needs Identification

The literature showed that performance testing protocols are a widely accepted way of evaluating the ability and skills athletes possess in sports clubs, sports science and physiology by coaches, trainers, physical educators and conditioning specialists (McGee and Burkett, 2003; Pauole et al., 2000). Through performance testing, an athlete's performance and skill capabilities can be monitored and evaluated (Kutlu et al., 2017), as well as use for talent identification (Bidaurrazaga-Letona et al., 2015; Chu and Vermeil, 1983; Hermassi et al., 2011; Reilly et al., 2000) , differentiation between higher-skilled players from their lesser-skilled counterparts (Farrow et al., 2005; Gabbett and Benton, 2009; Henry et al., 2011; Lockie et al., 2014; Mooney et al., 2011; Morland et al., 2013; Pauole et al., 2000; Veale et al., 2010), and even usefulness in deciding best positions of play (McGee and Burkett, 2003).

Among sports performance testing, agility has become an increasingly popular skill to evaluate in an athlete in a notable number of field and court sports. However, the literature shows that the standardised agility tests conducted in sports institution and club testing batteries and combines do not truly reflect the construct; they are not capable of testing all components of agility. With tests such as the Illinois Agility, AFL Agility Run, 5-0-5, T-Test (T-Drill), Pro-Agility (5-10-5 Shuttle) and the 3-Cone Drill (L-Run) involving pre-planned agility only, they lack the perceptual and decision-making factors of agility, thus are only measures of the physical components.

In an endeavour to develop a testing protocol capable of testing both perceptual and decision-making factors of agility, as well as the physical factors, RATs were developed, which incorporates a reactive component to the test. These have proved a reliable and valid method of evaluating agility performance (Farrow et al., 2005; Inglis and Bird, 2016; Serpell et al., 2010; Veale et al., 2010) through differentiation between higher-skilled athletes from their lesser-skilled counterparts (Farrow et al., 2005; Gabbett and Benton, 2009; Henry et al., 2011; Lockie et al., 2014; Morland et al., 2013; Veale et al., 2010; Young et al., 2011). However, the two major limitations associated with RATs included not being standardised and that they require a lot of expensive equipment to perform the test. The traditional RATs also did not include sport-specific equipment (such as a game-ball) and the ones that did varied significantly in testing protocols.

Similarly, a competitive market analysis also concluded that a gap in commercially available products was that they were not able to incorporate sports-specific equipment into the test and remain standardised, such that the results could be compared across sporting institutions or clubs. Another significant finding was that no system was able to assist in setting up the layout of the components for the test, so a user could know exactly where they needed to position the devices to produce a consistent test set-up. The closest feature to this was that timing gates had the ability to assist in alignment of the transmitter and reflector. Furthermore, it seemed that for best use, most systems needed some form of accessory, in the form of a tripod or mount, which was required to be bought separately.

3.1.3. Problem Statement

By collating all the information presented in the market and research analysis, the main issues arisen regarding the current situation in agility testing could be provided through a succinct problem statement:

"Current agility tests do not incorporate the perceptual and decision-making factors associated with agility and therefore do not truly represent the construct. Emerging reactive agility tests have the capacity to test these perceptual and decision-making factors but are not standardised as they vary in many aspects between studies conducted in literature. To conduct these tests, they also require a large amount of expensive equipment. Industry products are unable to incorporate sport-specific equipment within their tests and remain standardised, as well as perform their primary function at the greatest capacity, without the use of additional equipment."

3.1.4. Needs Statement

With a clear understanding of the current situation regarding agility testing. A needs statement was developed, which served as the building blocks for the design and development of the proposed system.

"To design and develop a portable, lightweight, wireless and easy to use novel system capable of testing the physical, perceptual and decision-making factors of an athlete's agility. The system must incorporate some form of sport-specific equipment and a reactive component that introduces randomness to the test. The test conducted by the system must also not be able to be anticipated or predicted by the athlete."

3.2. Device Requirements

By identifying the current problem and need for a new agility testing system, both the problem and needs statements were able to be used as the framework for designing and developing the proposed system. The next stage involved identifying the needs of the system, such that it could be broken down into specific requirements. This process was completed by all parties associated with this project, where the requirements were governed by the literature, market research and analysis, problem and needs statements. Therefore, this stage of the project contributed to providing a tool for translation between broad ideas and needs, into the specific device requirements essential to accomplish the primary aim.

3.2.1. Requirement Identification

After confirmation of the project, an initial kick-off meeting was held between Dr. Elliot, Dr. Hobbs, Prof. Taylor and the author to get a clear understanding and clarification of the system objectives and requirements. Dr. Elliott was able to provide a clear explanation of his ideas and thoughts behind what he proposed the system needed to have market appeal. All corresponding bodies provided feedback and discussed furthermore, expanding on ideas and including anything more that seemed relevant to make the system as successful as possible. Through this and the initial proposal document delivered by the SHAPE Research Centre (Figure A- 1 in Appendix A: A.1), alongside the literature review and market research and analysis, a set of specific design requirements were identified (Table 9).

Table 9: Design requirements for the proposed agility testing system

Design Requirement	Design Description
Safe to use/ minimum risk profile	Safety is always the number one design requirement for any system, as the safety of the user is uttermost important. The system must not harm the user or anyone in direct contact with it.
Low-cost	Aiming to develop a system that is of low-cost will give it an edge over its competing products as it will attract customers to purchase the product.
Portable	The system will need to be taken to areas of varying terrain, where athlete training occurs such as on the field, track or gymnasium. As a result, it is important the system can be compacted together for easy transportation. Current agility systems are portable such that they can be carried around in a suitcase, bag or carry-case; therefore, the proposed system needs to be able to provide the same option. Therefore, for the system to be portable, each component needs to occupy as little space as possible.
Lightweight	With the system being transported around location to location, it is vital the system is lightweight so that it makes transportation simpler and easier for the user.
Wireless	Seven of the eight competing products analysed were found to support wireless communication, so to meet the current market standard, it is imperative that the system is wireless, such that communication between devices and the user is wireless.
Adjustable test difficulty	The system needs to be able to have adjustability such that there are multiple testing difficulties able to be set. A harder test would mean a great testing zone area, resulting in more distance covered by the athlete. By incorporating variable test difficulty, the system is open to a wider target market of various fitness levels and age groups. A minimum of two different difficulties is required, so that a test conducted on league players could differ compared with entry-level players.
Easy to set-up	The literature review found currently implemented pre-planned agility tests are simple to set-up, involving nothing but a few cones and a measuring tape or wheel. Many high-end industry timing gates provide assisted componentry set-up to align the emitter and detector together, which makes set-up far simpler for the user.
Assisted test layout set-up	The review of literature found that no commercial product currently can assist the user with test set-up in such a way that helps them determine exactly where to position the device when placed in a specific sequence. For example, if a user wants to place their system in the form of a shape, they must manually measure the distances between them.
Compatible with Australian rules football	Dr Elliott had a firm belief this system would be very well suited in the Australian rules football profession. Thus, the test needs to be compatible with Australian rules football at a minimum.

Compatible with various sports	It is desirable to be able to be compatible with other sports including netball, basketball or rugby.
Sufficient area occupied by testing zone	The testing zone should ideally cover an area greater or equal to the AFL Agility Run. The testing zone of the AFL Agility Run occupies approximately 25m^2 (Tanner et al., 2013). Therefore, Dr. Elliot emphasised the proposed agility test needs to occupy this area at a minimum to have a competitive edge
Simple test initiation	To initiate the test, it must be a simple process, where either the system can automatically detect an athlete within the testing zone, or the athlete initiates the test when they are ready.
Functions indoors and outdoors	It is important that the test can be conducted in an environment that the athlete is normally accustomed to whilst playing their sport; for example, an AFL player should complete the test outdoors on the field, while a basketball player should complete it on the court. The test should function the same no matter what environment it is put in.
Waterproof design	Although it is unlikely the test would be conducted in extreme weather conditions, it is highly desirable to be able to still function even when raining.
Standardisation	This requirement is extremely important if the device is to become a breakthrough in the market. The test needs to be conducted across various sporting clubs and sports, so a test that is standardised means that data can be compared confidently.
Integration of sport-specific equipment	Another key requirement that will make this product stand out significantly from other products is the incorporation of sport-specific equipment whilst remaining standardised. The literature and competitive market analysis found that no tests can implement sporting equipment while remaining standardised. It is desired that a game ball can be placed on top of each marker.
Marker activation and deactivation	A marker needs to become activated, such that it is the location where the user needs to navigate to. The marker needs a method to detect when a ball has been removed from the athlete, thereby deactivating it. Deactivating the currently activated marker will subsequently cause activated of another random marker.
Visual and audio stimulus	When a marker is activated, it will present both an audio and visual stimulus to the athlete to indicate that the ball needs to be removed from that marker. In doing so, two of the athletes' sensory pathways (visual and auditory) are recruited. Only one random marker is activated at a time.
Visual stimulus presents random character	It is desirable that the visual stimulus is in the form of a character (letter, number or symbol), this way, a random set can be applied to different tests so that it brings uncertainty and unpredictability.

Incorporates physical factors of agility	The system needs to ensure the athlete uses physical components of strength, power, balance and coordination to sprint short-distances and make sudden change-of-directions to each stimulus to deactivate it.
Incorporates perceptual and decision-making factors of agility	A major differentiator the proposed system implements is incorporation of the reactive element, which causes the athlete to use perceptual and decision-making to complete the test.
Minimum of four (4) markers	It was established by Dr. Elliott that a minimum of four markers would need to be included in the system. By incorporating at least four markers into the test, it means the athlete has many possible locations they must run towards.
Measures time taken to complete test	All present agility tests measure time taken to complete the test, which is a good method of inferring agility as athletes with a greater ability can navigate and complete the tests faster than individuals with less ability.
Intra-system communication	The system needs to be able to have all components connected to one another and to be able to communicate wirelessly.
Communication with laptop or smart phone	As the user needs to be able to determine the results of each test, it must be able to communicate with either a laptop, a smart mobile phone or both.
Provide real-time data feedback	The competitive market analysis found that all existing products were able to provide real-time data feedback; the time taken to complete the test was able to be seen subsequently when the test was finished. It is vital that the proposed agility tester can provide the user with the test time upon completion to meet current market standards.
System feedback clear and concise	The feedback from the system needs to be clear enough that the user understands what it is referring to but is as concise as possible, so minimal time is spent having to read and decipher the feedback; enabling more time focused on the athlete.
User interface must be simple	It is important that the interface to communicate with the system is easy to use and simple to understand.
Provides feedback to commands	The system needs to have a way of providing feedback to the user so that they know when they have entered a command or pressed a button. This could be in the form of a sound or light.
Data storage	Must be able to store data and provide the user with basic test configuration values. For example, the user needs to know what the current test difficulty is and other various test modes. Additionally, it would be ideal for the user to be able to determine a time measurement from any test previously ran.
Maximised battery lifetime	A standard testing battery or combine would occur during normal business hours (9am – 5pm). Therefore, the system needs to be able to operate ideally for eight hours at a minimum.

Easy change/recharge of batteries	For added convenience to the user, it should be easy for to change or recharge the batteries (one or the other, depending on the final design). This means that the user should not need to use any tools to complete this process and it should be a simple procedure to do so.
Maximised service lifetime	It is desirable to have as long service lifetime as possible. This means that parts should not break easily and should not need to be replaced
Minimal learning curve	There needs to be a small learning curve required by an athlete to perform the test who is unfamiliar with it or has not completed it before. During testing combines and batteries, many athletes are completing tests, so there is not enough time to train the athlete to complete the test
Maximised load capacity	The marker needs to be able to withstand the force applied from the weight of the ball. It should also be able to withstand the force applied if an athlete happens to stumble onto it. However, due to the scope of the project, the latter will likely be unable to be complete and should be further developed in future work.
Shock absorbent/ resistant	Since the components may experience high impact forces applied to them during the test from possible accidental miss of the ball, they need to be able to withstand such occurrences.
Appropriate programming language	The programming language used must be well-known so that a wide range of technical personnel are able to read over it and understand the general outline and functions.
Bug-free	The program needs to be written such that it handles all possible scenarios and user inputs. Naturally, people are inquisitive and like to test the limits of systems. The best method here is to try to break the program so then the appropriate coding can be written to handle such instances.
Consistent	The system needs to be able to assist set-up and run the test repeatedly and consistently with minimal variance.
Reliable	The test needs to produce reliable results that can be confidently compared with results from other tests.
Valid	The test needs to be a valid way of assessing the agility construct.
Unpredictable	The system must not produce a test that athletes are able to recognise or anticipate patterns, so that decision-making and perceptual factors of agility are tested through generation of random marker activation.
Height adjustability	It is desirable that the markers can have height adjustability, which the user can set when training and not performing the standardised version.
Sport-specific equipment tracking	Monitoring the ball will provide further information from the test and thus allow extensive analysis in the results such as if the athlete has fumbled the ball or if they have competently pick it up off the marker.
Can be turned on and off	The system can be turned on and off at the desire of the user. This is important so that the system is not constantly running and wasting battery.

Can be put in a sleep mode	To further maximise battery lifetime, it is appropriate that the system can go into a sleep mode if left for a period of inactivity.
Responsive to user inputs	The system needs to be quick to respond to user inputs and apply the necessary changes or run the appropriate commands.
Customisable tests	It is desirable that a user can generate custom tests so that the system can be incorporated into standard training regimes.

3.2.2. Requirement Prioritisation

The above listed requirements were all deemed important factors in developing the proposed system, however, it was not realistic to be able to implement all of these into the project. It was understood that designing and developing the system would be very involved and require a lot of work right from the initial stages. Thus, it was recognised that significant time constraints existed to complete the project within the deadline, as well as restrictions on the hardware components to be purchased, so that the project would remain within budget.

To assist in developing a successful product that can achieve the primary aim by the end of the project, it was imperative to categorise the design requirements into first order and second order features. First order requirements were minimum design features needed to ensure that the system could complete the primary aim (essential features), while second order requirements were design factors found to be important but were not essential (desirable features) for this stage of the project to meet the aims and objectives. Therefore, if these requirements were not able to be achieved, they would be implemented where possible, but were expected to be completed in future work. Thus, the next stage of the design process entailed prioritising the requirements specified in the previous section, so that it could be determined which tasks should be focused on more to ensure successful completion of the project.

An absolute importance value was given to each requirement between 0 and 10, with a higher value indicating greater importance. Requirements that were deemed to have equal importance were given the same absolute importance. The design requirements were retabulated (Table 10), displaying their prioritisation (1^{st} or 2^{nd} order) and their corresponding absolute importance. The total values of absolute importance were used to determine the corresponding relative importance of each requirement. This was represented as a percentage, which helped establish the importance of each requirement in relation to all the other requirements. The table showed requirements that were deemed most important (with 1^{st} order prioritisation and an absolute importance value of 10) being; portable, lightweight, wireless, adjustable, easy to set-up, compatible with Australian rules football, functions indoors and outdoors, is able to be standardised, integrates sport-specific equipment, is able to have activated and deactivated markers with them displaying a visual and auditory stimulus, incorporates all factors of agility (physical, perceptual and decision-making), the number of markers (minimum four), measures time taken to complete the test, has communication with either a laptop or mobile smartphone, is safe to use, has a sufficiently sized testing zone area, is reliable, valid, unpredictable and can be turned on and off using a button.

Table 10: Design requirements and their corresponding prioritisation and importance

Design Requirement	Prioritisation (1st or 2nd order)	Absolute Importance (1-10)	Relative Importance (%)
Portable	1st	10	2.857
Lightweight	1st	10	2.857
Wireless	1st	10	2.857
Adjustable	1st	10	2.857
Easy to set-up	1st	10	2.857
Compatible with Australian rules football	1st	10	2.857
Functions indoors and outdoors	1st	10	2.857
Standardisation	1st	10	2.857
Integration of sport-specific equipment	1st	10	2.857
Marker deactivation and activation	1st	10	2.857
Visual and audio stimulus	1st	10	2.857
Visual stimulus presents random character	1st	10	2.857
Incorporates physical factors of agility	1st	10	2.857
Incorporates perceptual and decision-making factors of agility	1st	10	2.857
Minimum of four (4) markers	1st	10	2.857
Measures time taken to complete test	1st	10	2.857
Communication with laptop or smart phone	1st	10	2.857
Safe to use/minimum risk profile	1st	10	2.857
Sufficient area occupied by testing zone	1st	10	2.857
Reliable	1st	10	2.857
Valid	1st	10	2.857
Unpredictable	1st	10	2.857
Can be turned on and off using a button	1st	10	2.857
Simple test initiation	1st	9	2.571
Low-cost	1st	8	2.286
Assisted test layout set-up	1st	8	2.286
Intra-system communication	1st	8	2.286
Maximised battery lifetime	1st	8	2.286

Design Requirement	Prioritisation (1st or 2nd order)	Absolute Importance (1-10)	Relative Importance (%)
Appropriate programming language	1st	8	2.286
Waterproof design	1st	7	2.000
Provide real-time data feedback	1st	7	2.000
Minimal learning curve	2nd	6	1.714
Compatible with various sports	2nd	6	1.714
System feedback clear and concise	2nd	5	1.429
Responsive to user inputs	2nd	5	1.429
Easy change/recharge of batteries	2nd	4	1.143
Height adjustability	2nd	4	1.143
Provides feedback to commands	2nd	3	0.857
Maximised load capacity	2nd	3	0.857
Shock absorbent/ resistant	2nd	3	0.857
Data storage	2nd	3	0.857
Bug-free	2nd	3	0.857
Consistent	2nd	3	0.857
User interface must be simple	2nd	2	0.571
Maximised service lifetime	2nd	2	0.571
Can be put in a sleep mode	2nd	2	0.571
Customisable tests	2nd	2	0.571
Sport-specific equipment tracking	2nd	1	0.286

3.2.3. Objective Tree

Now that it was understood what requirements were deemed most important and critical in accomplishing the aims and objectives set out to achieve, an objective tree could be constructed, which is a tool developed by Cross (2000). When successfully implemented, it clearly defines the hierarchical relationships and interconnections between aims and functions, so that the means of achieving objectives and the direction of the device design is established (Cross, 2000; Haik and Shahin, 2011).

The design requirements were divided into four primary categorical sets; test features, mechanical design, electrical design and safety. The objective tree was then developed to show a hierarchical structure (Figure 26), such that the most important requirements are located at the top, with less important features located further down the bottom.

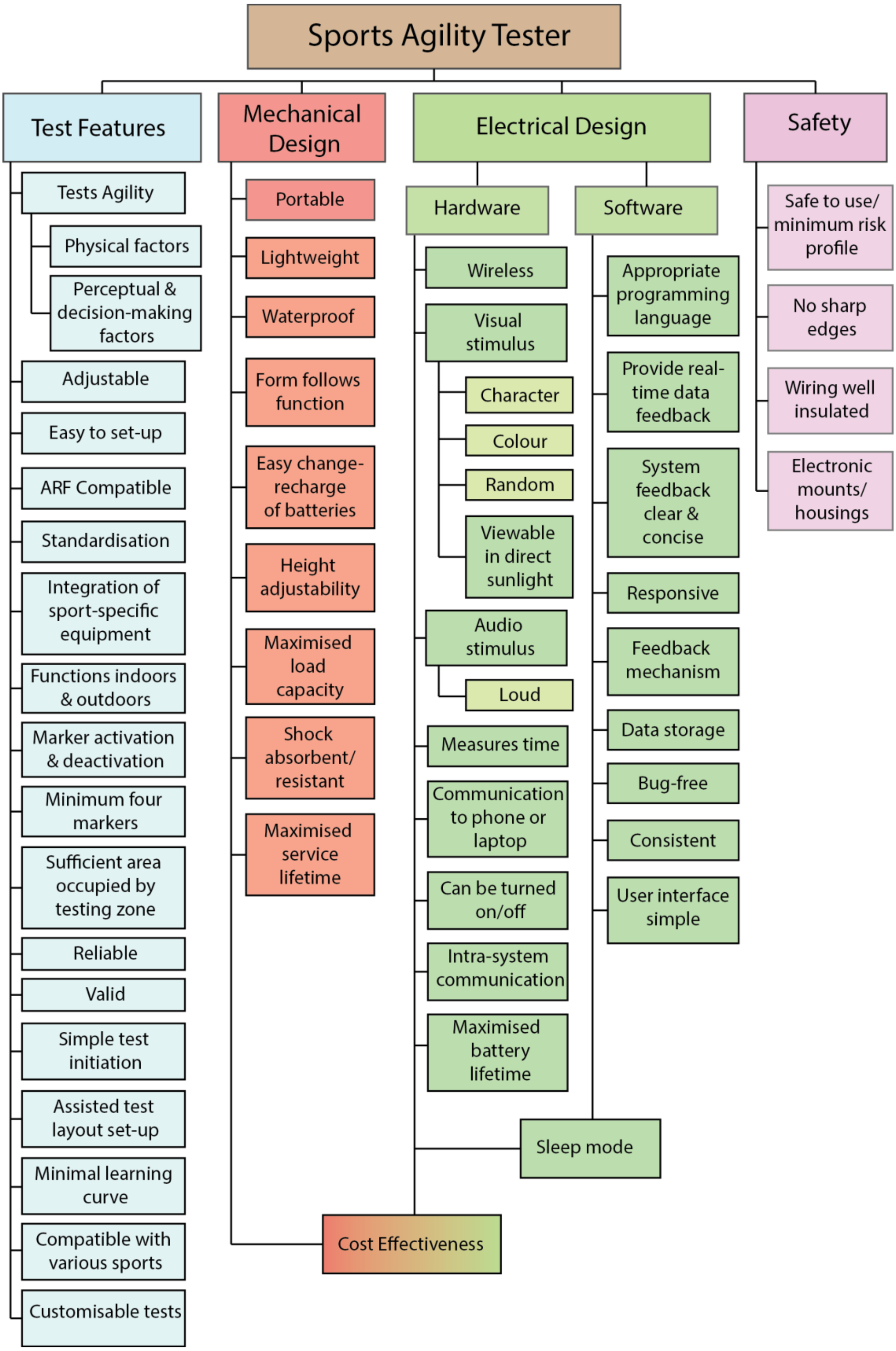


Figure 26: Objective tree for the project

3.3. Conceptual Design

With successful completion of the initial stage of the project (problem definition), a set of design requirements had been made that were clearly prioritised regarding importance. Thus, the next stage (conceptual design) could be undertaken, with the first step to establish a set of functional structures, which would be used to as a framework to develop a set of engineering design specifications that could be used to guide the design and development of the system.

3.3.1. Functional Analysis

In terms of the design process, a function is a solution-neutral action that allows a design to accomplish a specific task (Dym et al., 2014; Haik and Shahin, 2011). Thus, functional analysis is the process of establishing the functions that a design must perform.

3.3.2. Overall Function Structure

A design consists of an overall function, which is the relationship between a products inputs and its outputs (Haik and Shahin, 2011). To develop a set of solution-neutral functions, it was first imperative to establish the overall function of the proposed system, which could be presented succinctly as a *"system that tests an athlete's agility."* An overall function diagram was developed (Figure 27,) which clearly defines the inputs and outputs of the system; inputs including the user, the initial system settings, the subject (the athlete to be tested on), a power source, sport-specific equipment, environmental factors and a change in circumstance; outputs of a correct test configuration and user feedback.

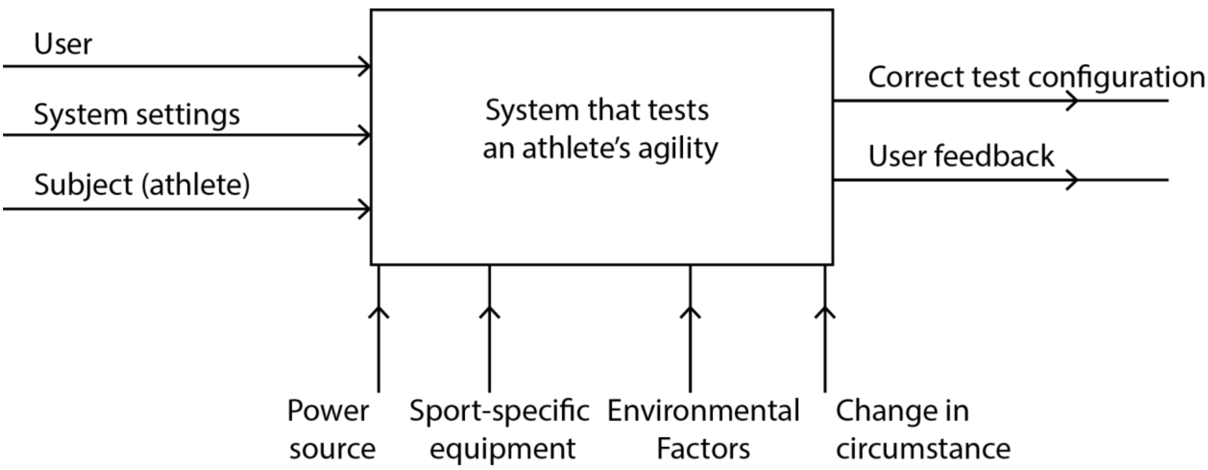


Figure 27: Overall function diagram for the proposed system

The overall function is represented as a "black box", which signifies that the current functions required to convert the inputs to outputs is not known. The next step of the functional analysis involved determining what goes within the black box to make it transparent.

3.3.3. Establishing Functional Structures

The overall function can be broken down into further subfunctions; this process of breaking down functions is known as function decomposition, and is employed by Haik and Shahin (2011) and Dym et al. (2014) and was implemented in this project.

It was understood that the conversion from the set of inputs into the set of outputs was going to be a very complex process at the main function level. Therefore, to determine what functions would go within the black box, the overall main function (a system that tests an athlete's agility) was separated into a set of essential functions; the user/system interaction, set-up test, intra-system processing and perform test. These functions were then assigned subfunctions, organised carefully into a systematic and logical order such that execution reflected the solution-neutral actions produced to achieve the main function. The subfunctions needed to complete each function were listed and were tabulated (Table 11), which formed what is known as a function tree. This was used as a guide for development of the final complete function structure.

Table 11: Functions encompassing the subfunctions for the proposed product

1. User/System Interaction	2. Set-up Test	3. Intra-System Processing	4. Perform Test
i. Receive input from the user	i. User positions marker	i. Sense the ball	i. System generates a character to display
ii. System processes input	ii. Measure marker distance	ii. Store the data and use it as the baseline values	ii. Wait for user to enter the testing zone
iii. System does the appropriate actions	iii. System indicates marker position status	iii. Generate the test sequence	iii. System initiate tests
iv. System indicates input has been received	iv. System checks if marker is in the correct position		iv. Athlete waits for stimulus
v. System provides user feedback	v. If it is, the user can position ball on marker		v. System present audio and visual stimulus

1. User/System Interaction	2. Set-up Test	3. Intra-System Processing	4. Perform Test
	vi. The process repeats for all markers		vi. System starts a timer
	vii. The system now has a correct test configuration		vii. System ignores sensor values from environmental factors
			viii. Athlete navigates to marker
			ix. Athlete removes the ball
			x. The sensor detects the change in circumstance by comparing values with the baseline sensor values
			xi. System records the elapsed time & repeats the process for the remaining markers
			xii. The system provides user feedback and waits for user to set-up the test again

3.3.4. Function Structure

A complete function structure was then able be developed, which incorporated the overall function structure diagram initially developed, the function tree and flow of materials (Figure 28). The overall function of the system was surrounded by a boundary indicated by a solid line to differentiate the system with its functions and subfunctions, from its inputs and outputs, so that an establishment of a feasible product could be made.

Through assessment of the function structure diagram (Figure 28), a clear set of essential functions could be identified within the overall function through bounded boxes. The interactions between subfunctions and functions could be observed, so that it was possible to determine exactly a logical order of solution-neutral actions needed to occur to convert the inputs into outputs.

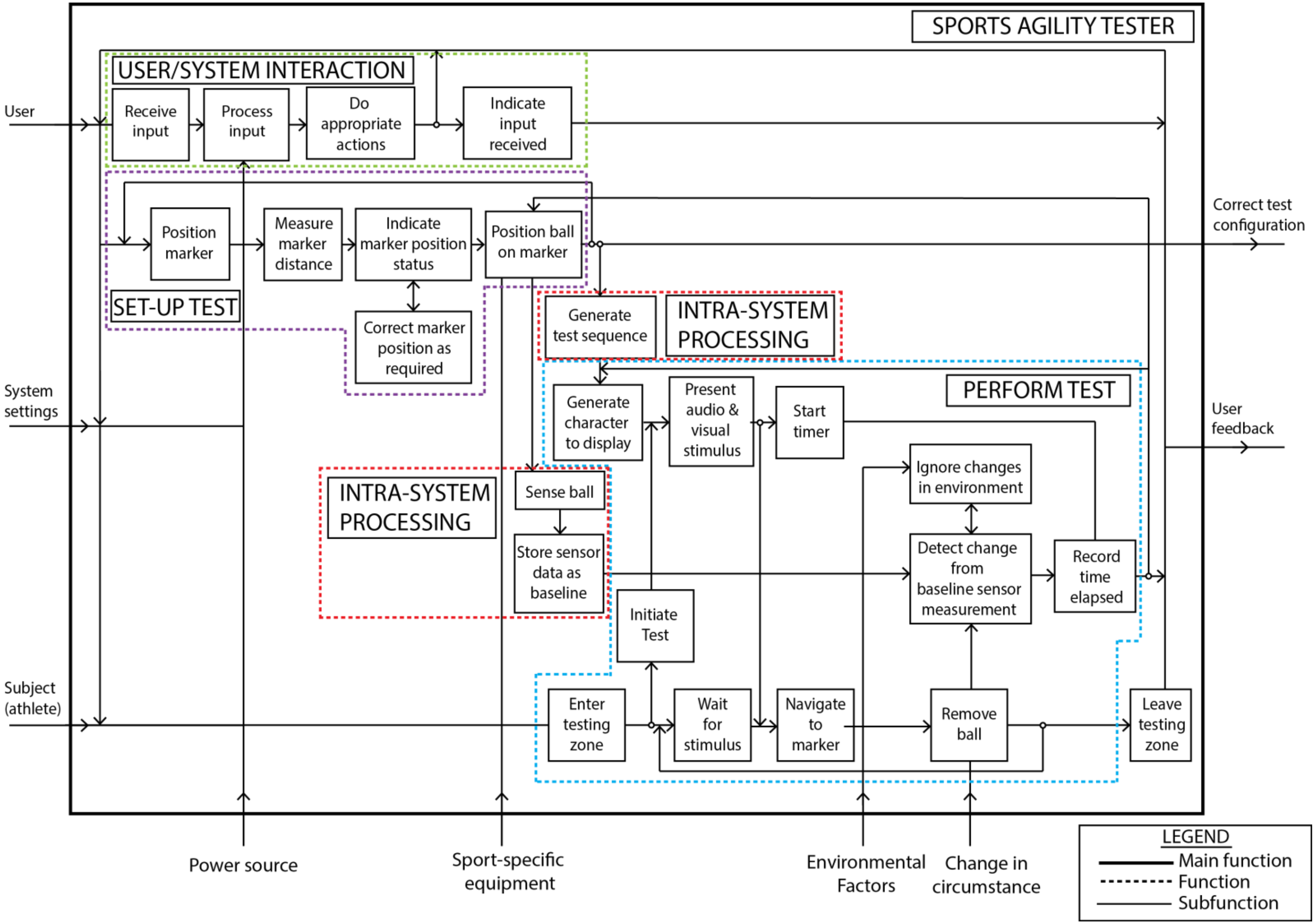


Figure 28: Complete function structure for the sports agility tester with subfunctions shown

3.4. Engineering Specifications

Up until this point of the project, all assessment had been conducted in a qualitive nature. This stage of the project entailed using quantitate assessment to set metrics and constraints associated with the design requirements and function structure. Through clearly defined design specifications, the subsequent conceptualisation and development of the product would have a definitive set of boundaries it needed to follow, as well as by providing an aim to achieve measurable targets. The design specifications would make it possible to evaluate the success of the final system at the end of the project by comparing actual metrics to target metrics. Thus, it was important that a set of design requirements were developed, which reflected the design requirements and function structure appropriately. Therefore, the development of the design specifications was imperative for achieving a successful design.

The performance-specification method was used to develop a set of design specifications, as employed by Haik and Shahin (2011). The performance attributes of the system were able to be devised through the objective tree and functional analysis. These performance attributes were then grouped into categories: functional, operation and performance requirements; workmanship and manufacturing; safety, regulatory and environmental requirements; human factors; budget; and schedule. Using the design requirements and further research and analysis through literature and standards, performance limits could be set for each attribute. A “delighted” target value and “disgusted” threshold value were recorded, to indicate the target value to achieve for, but also a value that would still be adequate to meet the design requirements and function structure.

Several specifications were developed, which correlated with 2nd order design requirements. As the requirements were deemed not essential to meet the aims and objectives for a basic design, the specifications that correlated were given target values that were desirable, but also threshold values, which signified that if they were not complete, then this was still acceptable.

3.4.1. Functional, Operational and Performance Requirements

3.4.1.1. User/System Interface Communication

It was established through the design requirements that the system needed to be able to connect to either a laptop or a mobile smartphone. It was extremely desirable to be able to connect to both devices, however it was understood it may not be achievable. Thus, the type of devices the system could connect to was aimed to be two devices, but one device was acceptable.

3.4.1.2. Physical Connections

An essential design requirement was that the system would be wireless. It was imperative that the system achieved this design requirement, since the competitive market analysis showed that all devices that were considered threats were wireless. Thus, the physical connections between devices and the user should zero.

3.4.1.3. Set-up Time & Steps Required to Set-up Test

One of the design requirements was that the test is simple to set-up. A simple to set-up test can correlate with how long it takes to set-up and the steps required to set-up the system. Therefore, it was imperative that these values were as low as possible. A reasonable two-minute set-up time was established as the target, with four minutes as the threshold.

The number of steps required to set-up the test would be dependent on the number of devices, therefore, instead of a whole number representing the total number of steps for the system, a value for the number of steps per device was established. To set-up the test, a user would need to, at minimum; place the marker down, place the sport-specific equipment on top of it and then press a button to indicate the ball is there. Therefore, three steps per device was defined as the target value. Accounting for other possibilities, a threshold value of eight steps per device was set.

3.4.1.4. Stimulus & Unpredictability

An increasing number of characters displayable by the visual stimulus would increase the unpredictability of the system, adding to the conceptual components of agility. Thus, it was desirable to be able to display all alphanumeric characters (A-Z and 0-9); totalling 36 characters. It was understood that this may be unable to be accomplished, so a threshold value of one character was established, so that at a minimum, the marker needed to display a simple visual stimulus. Similarly, the number of unique colours displayable on the marker would also increase the unpredictability of the system. It was decided that at a minimum, three colours were needed to be able to be displayed (red, green and blue), however, a target value was set for nine colours to be displayable.

The number of unique marker activation sequences corresponds to how predictable the test is. The test should have at least 25 unique sequences to ensure that the athlete is unable to make any predictions. With 25 unique sequences, there may be some athletes with exceptional memory to be able to anticipate the rest of the sequence after deactivating some markers. To ensure that this would not be possible, a target value was set at 50 unique sequences. Additionally, the number of markers is synonymous with this design specification, as an increasing number of markers would mean a greater number of marker activation sequences. After some careful consideration, six markers were deemed a suitable number to ensure a good balance between the maximum number of marker sequences and the complexity of the system. As significant budget restrictions existed, it was thought that developing six markers may not be possible and so the minimum number of markers was established to be four as per the design requirement.

3.4.1.5. Sensor Response Time

To ensure the test would be as responsive as possible to the athlete removing the ball, a reasonable sensor response time target was set at 0.001 seconds, but with a threshold value of 0.01 seconds.

3.4.1.6. Adjustable Test Difficulty

Dr. Elliott had mentioned that he believed the system should have at least two test difficulties, one for elite athletes and the other for amateur players. It was decided that it would be beneficial to have a third test difficulty for youth players too.

3.4.1.7. Sport Compatibility

By maximising the number of sports the system is compatible with, the target market increases, as well as ease of use across sporting institutions and sporting clubs. Being a prototype, it was understood not all sport compatibility could be accomplished, however it was decided to try to aim for compatibility with Australian rules football, netball, basketball and rugby. Soccer was not included, as it involves kicking the ball, which could be difficult to do without kicking the marker itself. An essential design requirement was that the system is at least compatible with Australian rules football, thus the system needed to be compatible with at least this sport.

3.4.1.8. Maximised Battery Lifetime

With a system that is hoped to be incorporated into testing batteries and combines, it needs to be able to run continuously for the period of a full working day, thus it was desirable that the system could run for eight hours. However, being a prototype, this may not be achievable at this stage, so a limit was established at three hours of run time.

To assist in saving power, it would be beneficial to implement a power-saving mode, for if the system is inactive for a length of time, or between tests. Being a prototype, it was accepted that the system being in full power mode for the entirety of it being on was acceptable.

3.4.1.9. Programming Language

It was desirable that one programming language was to be used to develop the system. However, being a prototype device, a target threshold of two programming languages would be accepted if it was required.

3.4.1.10. User Commands & Feedback

The number of feedback types would increase useability of the system for the user. A system that could provide a visual and auditory feedback response, as well as information for stored variable values such as the test time would provide an optimised system. Being a prototype device, it was established a threshold value of one feedback type would be enough. Similarly, the number of commands a user could provide the system, such that it provides a large amount of information about the test (one command to obtain marker activation sequence, one for test difficulty, etc.) would be exceptional. Other commands could include progressing to the next test or resetting the system. At a basic level, the system needed to know when to begin the test, proceed to set-up the next test and retrieve the most recent test time. In addition to this, the amount of time taken to respond to a command was another specification to consider. A system which couldrespond to a command in less than one second would be considered optimal, but a threshold target value would be under three seconds.

The competitive market research found that all tests were able to provide feedback on test results post-test. It would be extremely beneficial to see the elapsed time peri-test; however, this feature would likely be very difficult to implement into a prototype. Thus, post-test feedback was accepted as the threshold target.

3.4.1.11. Storing Previous Test Data

Having the system capable of storing tests would provide convenience to the user. If the test were to store previous test times, a minimum number of test times to store could be estimated. Estimating each test taking approximately one minute to set-up and perform, with each athlete performing three tests and one practise test, with an 8-hour system run time; it should be able to store approximately 480 tests; a target value was rounded up to 500. As this specification correlated with a 2nd order design specification, not being able to store any tests would be acceptable.

3.4.1.12. Device Boot-up Time

How long each device takes to boot up and be ready for any user input is essential in the contemporary age lived in today of instant gratification. A system that could be ready for user input in under three seconds would be considered exceptional and thus was set as the target value. A threshold target was set for device boot up time under 10 seconds.

3.4.1.13. Binary Specifications

Several specifications were unable to be provided with a scalable metric, however, were still deemed important specifications that needed to be evaluated. Each device within the system needed to be able to communicate with one another and it needed to function indoors and outdoors (including in direct sunlight). It was desirable that the test would initiate automatically upon detecting the athlete, however with this specification being correlated to a 2nd order requirement, it was acceptable for this not to be implemented. Similarly, the user creating custom tests and sport-specific equipment tracking specifications were developed with the same thought in mind.

3.4.2. Physical Requirements

3.4.2.1. Area Occupied by Testing Zone

A specification set by Dr. Elliot was that the testing zone needed to cover at minimum the area of the AFL Agility Run, which was found to be 25 m^2 through the literature review (Tanner et al., 2013). Therefore, this was set as the threshold target. An arbitrary 100 m^2 was set as the target, since it would be desired that the test could cover as large area as possible if required, for say a custom created test.

3.4.2.2. Sport-Specific Game Ball Sizing

With each marker needing to be able to have sport-specific equipment (most commonly a game ball) placed on it in some way, the marker therefore needed to be able to have a load capacity that reflected the weight of these game balls. Additionally, it needed to be large enough to house the ball, with it also remaining positioned until the athlete removeed it. Thus, research was conducted into various sport-specific game balls, which corresponded to common field and court sports to be made compatible with the test; Australian rules football, netball, basketball and rugby. It was important to establish the standard sizing for typical adult league play.

The game ball sizing varies depending on the sport, as well as the age and sex of the individuals playing. For example, the Australian football ranges from size mini, junior, youth, 1, 2, 3, 4, 4.5

and 5; while a netball only has two sizes (4 and 5) (NSW Netball Association, 2018). The reason for various sizing is so the ball dimensions meet the anthropometrics of the target players; smaller balls would be designed for players who are younger as they will fit within their hands far easier, while larger balls would be suited for older players with larger hands (HART Sport, 2018). Similarly, it is accepted that generally, males have larger hands than females (Comparison of the 5th and 95th percentile), so in some sports, the male and female game ball varies in size too (Garrett, 1971).

3.4.2.2.1. Australian Rules Football

The official AFL game ball is the Sherrin Kangaroo Brand Australian football (Russell Corporation, Scoresby, Melbourne, Australia), which is size 5 for males and size 4 for females (Russell Corporation, 2018). The shape is in the form of a symmetrical oval shape, with the size 5 game ball conforming to dimensions 720 – 730 mm and 545 – 555 mm for the circumference and transverse circumferences, respectively (Australian Football League, 2017). The ball weighs between 450 and 480 grams (Nauright, 2018). The size 4 game ball is approximately 690 mm and 530 mm, respectively (Russell Corporation, 2018). No length of the football is documented in the AFL official rulebook, however the United States Australian Football League (USAFL) does state the ball is 270 – 280 mm in diameter (USAFL, 2018).

3.4.2.2.2. Netball

As mentioned, a netball contains two sizes; 4 and 5. The official size netball used in game is size 5, or a size 5 Association Football (Netball Australia, 2018). An official match ball measures 690 – 710 mm in circumference and weighs 400 – 450 grams (INF, 2018).

3.4.2.2.3. Basketball

The official basketball game ball for the Basketball Australia and the National Basketball Association (NBA) is the Spalding basketball (Spalding, Scoresby, Melbourne, Australia; a division of Russell Corporation), while the official basketball for the National Collegiate Athletic Association (NCAA) Championships and the National Basketball League (NBL) is by Wilson (Wilson Sporting Goods, Chicago, Illinois, USA). The official basketball is size 7 for males and size 6 for females (Spalding, 2018; Wilson Sporting Goods, 2018). The circumferences are 29.5 and 28.5 inches (≈749 and 724 mm) for size 7 and 6, respectively. According to the official NCAA rulebooks, the official weight of a basketball is to be 20 to 22 ounces (567 to 623.7 grams) for Men's Basketball and 18 to 20 ounces (510.3 to 567 grams) for Women's Basketball (NCAA, 2018a, 2018b).

3.4.2.2.4. Rugby

The official NRL game ball is the Steeden rugby football (Gray-Nicolls Sports, Cheltenham, Victoria, Australia), while for World Rugby Union it is the Gilbert rugby ball (Gilbert, Grays of Cambridge (Int), East Sussex, United Kingdom). The official game ball is classified as a size 5 for males (Gray-Nicolls Sports, 2018). It has a length of 280 – 300 mm, with 740 – 770 mm circumference and 580 – 620 mm transverse circumference, with a weight 410 – 460 grams (World Rugby, 2018).

3.4.2.2.5. Game Ball Sizing Summary

The information gathered was collated and tabulated to show the ball sizing, circumference and weight of the official game balls (Table 12).

Table 12: Official game ball sizing for various sports

Sport	Official Game Ball Size	Circumference (mm)	Length (mm)	Weight (grams)
Australian Rules Football	M: 5 F: 4	M: 720 – 730, 545 – 555 F: 690, 530 (C, TC)	270 – 280	450 – 480
Netball	5	690 – 710	219.6 – 226^	400 – 450
Basketball	7	M: 749 F: 724	M: 238.4^ F: 230.5^	M: 567 – 623.7 F: 510.3 to 567
Rugby	5	M: 740 – 770, 580 – 620 (C, TC)	280 – 300	410 – 460

* C = circumference, TC = transverse circumference

* M = males, F = females

^ Diameter, d, calculated using the circumference, C, with formula: $d = C/\pi$

3.4.2.3. System Containment

With the system being designed and developed to be in the form of a proof-of-concept prototype, it was decided the carry-case would be in the form of a suitcase; so that one less item needed to be designed and manufactured. With such a large variety of sized suitcases available, there was quite a lot of flexibility in terms of containment volume. So that the prototype could be tested against a containment, but also save costs of purchasing a new suitcase, two specific suitcases were considered for the carry-case as they were already owned by the author; one large and one medium. The large suitcase was the HS MV+ Deluxe Expandable Hardside Spinner Case (American Tourister, Samsonite, Rhode Island, USA), which has dimensions 790 x 540 x 320/ (360 expanded) mm, weighing 4.7 kg and capacity of 107/125 L (0.107/0.125 m^3). The medium suitcase was the Bon Air Deluxe Expandable Hardside Spinner (American Tourister, Samsonite, Rhode Island, USA), which has dimensions 660 x 460 x 280/ (310 expanded) mm with weight 3.8 kg and 64/75 L capacity (0.064/0.075 m^3). To allow for some tolerance for padding, the target values were established at 95% of the expandable capacity volumes. Thus, the total system target volume to achieve for was 0.07125 m^3 while the threshold value was 0.11875 mm^3.

3.4.2.4. Physical Dimensions

Several factors attributed to the target values specified for the physical dimensions of the devices encompassed within the system; housing of electronic componentry, the sport-specific game balls and containment within a carry-case.

It was desirable to have a smaller design, such that it would become more portable and easier to store within a carry-case. However, a trade-off existed; since all electronic componentry needed to fit within the device, as well as it being able to have a sport-specific game ball placed on top of it and remain secured until it was removed by the athlete. With electronic componentry unknown at this point in the project, it made it very difficult to know what a big enough enclosure would be. Being a prototype, future work could be implemented to scale down the device enclosure, but a prototype enclosure which would be too small and not fit electronics would be a major issue. A reasonable set of target dimensions was set at 150 mm for the length and width, and 250 mm for the height. The threshold values were set to be 200 mm for length and width and 300 mm for the height.

To ensure that a ball positioned on the marker would remain stable until removed by the athlete, the length and width were compared against the maximum length value for the sport-specific game balls (Table 12). The rugby ball was found to have the largest possible length of 300 mm. Therefore, the established 150 mm and 200 mm length and width values were found to be 50% and 66.7% of the diameter. With the marker covering at least 50% of the rugby ball, one could assume that the design would be able to reliably hold the ball until an athlete removed it.

The minimum suitcase length was 280 mm for the Bon Air Deluxe. Allowing for 5% padding tolerance, 95% of this value is 266 mm. Comparing the established height targets of the marker with this tolerance allowable suitcase length value, the 250 mm target would fit, but the 300 mm target value would not. However, laying the marker in either of the other directions (627 and 437 mm with 5% padding tolerances), the marker height would fit. Thus, the established target values for the length, width and height dimensions were justified.

3.4.2.5. Load Capacity

At a minimum, each marker needed to be able comfortably handle the load distributed by any type of ball placed on top of it. The heaviest ball was the basketball with a maximum weight of 623.7 grams (Table 12). Using this value, the maximum load imposed due to the weight of a ball could be determined using Newton's Second Law of Motion (Equation 4).

Equation 4: Newton's Second Law of Motion

$$F = ma$$

$$\therefore F = (0.6237\ kg) \times \left(\frac{9.81m}{s}\right) = 6.1185\ N$$

Rounding up to a whole number, the marker needed to be able to be capable of carrying a 7 N load at the minimum to be able to withstand the load imposed by the weight of an Australian rules football, netball, basketball or rugby.

This force accounted for the weight of the ball only and was not accounting for possible forces produced by the athlete who may momentarily push down to get a grasp to pick up the ball. Accounting for this situation, the load capacity needed to be higher. Another scenario that would produce more force would be someone pushing down on the marker. However, it was difficult to provide a value to the force produced in these scenarios. The Eastman Kodak Company (2004) recommends the upper limit of force for a push down movement at elbow height to be 29 kg, which is equivalent to 284.39 N. Being a prototype device, the aim was to provide the essential functionality and providing a device that could handle a 290 N load would come at a great manufacture cost, thus as a result, this was set as a delighted target goal.

3.4.2.6. System Mass

Although there is no limit for the maximum allowable mass an individual can lift, the National Code of Practice for Manual Handling 1990 suggests as a general guide, the mass of an object should be below or within the range of 16 – 20 kg; as the weight increases from 16 kg onwards, the risk of back injury significantly increases (Safe Work Australia, 2005). Therefore, the total mass of the system within its containment needed to aim to be no more than 16 kg, but with a target of 12 kg, which was 25% below this value. However, since the mass of a carry-case could vary significantly, it was decided this value would represent the total mass of the system not including the mass of the carry-case.

3.4.2.7. Ingress Protection

The Ingress Protection (IP) Code (or IP Rating) defines the international standards for classification of protection against the intrusion of solid bodies or water into an electrical enclosure (IEC, 2013). The IP Rating is represented as two numerals, the first identifies the level of protection the enclosure has against the ingression of solid objects, while the second numeral identifies the protection the equipment inside the enclosure has against the harmful ingress of water.

Referring to the code, it would be desirable that the system has an IP Rating target of IP56. That is, in terms of ingression of solid objects: the product would be completely protected against contact from any solid body (even objects less than 1 mm) and should be dust protected, such that the ingress of dust does not enter in such a quantity that interferes with the satisfactory operation of the design (IEC, 2013). It would be important that objects would not be able to ingress into the device while it in use and dust would not ingress in a quantity such that there could be harmful effects to the device or the user. In terms of harmful ingress of water: the product should withstand powerful water jets from any direction and have no harmful effects (IEC, 2013). It would be desirable to be able to use the device in all weather conditions, even in heavy pouring rain since many athletes would continue training regimes no matter the weather. Additionally, if the product was accidentally left outside, it would not be detrimental; so, this IP Rating reflects this.

The threshold IP rating for the system was established to be IP44. In terms of ingression of solid objects: no object greater than 1 mm shall enter the device, meaning objects as small as screws and most wires would not penetrate the device (IEC, 2013). In terms of harmful ingress of water: the product would withstand splashing water from any direction and have no harmful effects (IEC, 2013). This IP Rating would reflect standard use out in the field or on the court on a normal day and if there was light rain. If used for the standardised test, the product would not be expected to be used in heavy rains since it would likely affect the athlete's results.

3.4.2.8. Outdoor Environments

One of the key requirements provided was that the system needed to be able to function both indoors and outdoors. Ingress protection was already discussed which covered specifications regarding protection from dust and water. Another significant area to consider was exposure to direct sunlight and the affect it might have on the system performance or on the enclosure itself. In very hot conditions, the electronics may not perform optimally and the exposure to direct sunlight could cause breakdown of the housing material. It was assumed that breakdown of the material would be due to long-term exposure. Ambient temperature conditions were also another factor to consider. For example, if the device was left outdoors overnight, temperatures could become very low and as a result, could affect the electronics of the system. Due to the time constraints imposed, it was deemed out of the scope of this project to optimise the system so that it would be resistant to both breakdown of housing material due to extreme direct sunlight exposure or in conditions where it would be left outside overnight. These factors should be considered and addressed in future work.

3.4.2.9. Adjustable Height and Impact Resistance

Adjustable height and impact resistance were specifications which correlated with 2nd order aims, so the threshold target values were set accordingly. It was desirable to have an adjustable height device, but it was not essential. Due to time constraints, it was addressed early on that this proof-of-concept prototype system would not have adjustable height and that this could be considered in future work. Impact resistance would have been optimal to implement, however, being a proof-of-concept prototype, it was deemed out of the scope of the project and thus a target value was set to 0 N.

3.4.3. Workmanship and Manufacturing

3.4.3.1. Number of Parts

A reasonable limit for the number of parts the system would contain was set to be less than 150 as a threshold and less than 75 for a target value.

3.4.3.2. Internal Parts Enclosed

It was desirable that 100% of parts would be enclosed internally, however being a prototype, it was determined this would not be possible, so a threshold value of 90% was established.

3.4.3.3. Tools Required for Battery Recharge/Change

For ease of use for the user, it was established that having no mechanical tools to change or recharge the battery would be beneficial, however it was understood being a prototype, then a tool may be required to accomplish this.

3.4.4. Safety, Regulatory and Environmental Requirements

3.4.4.1. Risk Measure

Safety is a number one priority, thus is was imperative that a prototype would be developed that is safe to use. A way of assessing the risk level is through a risk assessment and a risk matrix. The risk matrix used in this project was based on the one employed for all risk assessments conducted at Flinders University (Figure 29).

Likelihood						
	Very Likely	Medium	High	High	High	Extreme
	Likely	Medium	Medium	High	High	High
	Possible	Medium	Medium	Medium	Medium	High
	Unlikely	Low	Low	Medium	Medium	Medium
	Highly Unlikely	Low	Low	Low	Medium	Medium
		Negligible	**First aid**	**Minor injury**	**Major injury**	**Fatality**
		Consequence				

Figure 29: Risk matrix used when assessing risks at Flinders University

The risk assessment was conducted such that each potential hazard associated was identified, assessing the consequence and the likelihood of occurrence to provide a risk measure. Risk controls were evaluated to determine a residual risk level, with the worst case-scenario identifying the risk level of the system. The target would be to have a prototype that is of a low risk level to use, but with a threshold of medium risk level. A risk level any higher would not be acceptable for use.

3.4.4.2. Extra-low Voltage System

The Installation Requirements for Customer Cabling (Wiring Rules) 2006 Australian Standard [AS/NZS 60950.1:2003] defines an extra-low voltage (ELV) that does not exceed 42.4V peak or 60V DC (Australian Communications Industry Forum, 2006). It was deemed necessary to develop an extra-low voltage system to minimise the risk involved in using it. However, being a prototype, it was decided that an even lower maximum voltage was to be set as the limit to ensure safety. Thus, a maximum voltage the system would be allowed to use is 12V, with a desirable maximum voltage of less than 10V.

3.4.4.3. Noise Produced by the System

The noise produced by the system could be distinguished between the sound made from the mechanical and electrical components and the noise produced by the stimulus.

Sound intensity is measured in decibels (dB), which is expressed as a logarithmic function of the acoustic power squared (Glover, 1993). Having a logarithmic scale means that the energy produced by a source is doubled for every 3 dB shift increase (Glover, 1993; Safe Work Australia, 2015).

The various sound intensity levels corresponding to several devices and environments were tabulated (Table 13). It should be noted that pain can be felt at the 130 dB threshold.

Table 13: Sound intensity levels (Modified from Glover, 1993; Safe Work Australia, 2015)

Level (dB)	Qualitive Description	Source/Environment
10	Very Faint	Hearing Threshold; Anechoic Chamber
20	Very Faint	Whisper; Empty Theatre
30	Faint	Quiet Conversation
40	Faint	Normal Private Office
50	Moderate	Normal Office Background Noise
60	Moderate	Normal Conversation
70	Loud	Radio, Loud Conversation
80	Loud	Heavy Traffic, Noisy Office
90	Very Loud	Lawn Mower, Unmuffled Truck
100	Very Loud	Sheet-metal Workshop; Boiler Factory
110	Very Loud	Chainsaw
120	Extremely Loud	Rock drill
130	Extremely Loud	Rivet Hammer
140	Extremely Loud	Jet engine at 30m

From the table, it is possible to provide a reasonable set of sound intensity values for the proposed system to be developed. It would be extremely desirable that the system produces minimal sound as possible, so a value less than 20 dB is a good target to achieve for, which corresponds to a very faint sound of a whisper or empty theatre. However, it would still be acceptable for a system that produces less than 60 dB, corresponding to a normal conversation.

The audio stimulus presented by the marker must be loud enough such that it could easily be heard by the athlete in an outdoor setting on the field or court. However, it would also need to be within a safe range such that no damage to hearing occurs and no pain is felt. It was already been established that it must not exceed 130 dB, which is the threshold where pain is felt. The Managing Noise and Preventing Hearing Loss at Work Code of Practice 2015 specifies the length of time an individual can be exposed to various noise levels with no hearing protection (Table 14).

Table 14: Length of time an unprotected person is to be exposed for corresponding to various noise levels (Modified from Safe Work Australia, 2015, p. 7)

Noise Level (dB)	Exposure Time
80	16 hours
82	12 hours
85	8 hours
91	2 hours
97	30 minutes
100	15 minutes
103	7.5 minutes
106	3.8 minutes
109	1.9 minutes
112	57 seconds
115	28.8 seconds
118	14.4 seconds
121	7.2 seconds
124	3.6 seconds
127	1.8 seconds
130	0.9 seconds

As it is known that the stimulus would not be presented for very long (< 1 second), which corresponds to 130 dB, however it was already established this value was not acceptable. To account for those who may have exceptional hearing, a maximum sound intensity level of 110 dB was selected, which corresponds to a chainsaw and allows for 57 continuous seconds. Although the stimulus would be presented for less than a second, this value considered the administrator who may monitor the test that may be continuously conducted for an entire day.

3.4.4.4. Number of Loose Parts

With a possibility that an athlete may knock over a marker, there could not be any loose parts in a final industry product. However, being a proof-of-concept, this iteration of the design was to show a working design, producing a tangible product from theory. Thus, it was established that future development and optimisation could aim to ensure no parts are loose. Thus, a target value was established to have no loose parts, but threshold target was established at two parts.

3.4.5. Human Factors

3.4.5.1. Sensory Pathways

An essential design requirement was that each marker would need to present two distinct sensory pathways; auditory and visual. Thus, target and threshold values for the number of sensory pathways the marker would use was set to two sensory pathways.

3.4.5.2. Incorporated Agility Components

Through the literature, two essential components of agility were found; a physical and a cognitive component (CODS and perceptual-cognitive speed, respectively). To recognise the subcomponents that make up these two essential components, the deterministic model of agility components proposed by Young et al. (2002), modified by Sheppard and Young (2006) and further modified by Nimphius (2014) can be referred to (Figure 3). The model shows several subcomponents can be defined, with CODS having anthropometrics of body position and technique regarding foot placement, stride adjustment and body lean and posture; straight line sprint speed; and leg qualities specific to COD step including concentric, isometric and eccentric strength, power and rate of force development and reactive strength. Perceptual-cognitive speed subcomponents include visual scanning, anticipation, pattern recognition, knowledge of situations and reaction time. Thus, 14 individual subcomponents of agility can be defined using this deterministic model. Ideally, the target would be that the system is able to test all components of agility, however in reality, several of these may not be able to be met. Thus, a target of all 14 components of agility would be required to be tested on, but with a threshold target of 11 components.

3.4.5.3. Familiarity

It would be essential that the test is simple enough that an athlete can perform it confidently and as quickly as possible. With many athletes being tested in a single testing battery or combine, to get through all of them, there is not enough time for multiple practise runs. Therefore, the target would be to have the athlete complete one practise run and feel confident in the test procedure. The threshold target was established to be two practise runs.

3.4.6. Reliability, Maintainability and Supportability

3.4.6.1. Accuracy of Time Measurement

Through the competitive market analysis, the two most general precision accuracy of time measurements was found to be 0.01 seconds (SMARTfit Strike Pods, SpeedLight) and 0.001 seconds (FITLIGHT Trainer, Witty SEM, SmartSpeed Pro, TC Timing System). Thus, it was desirable to have a system that could produce a precision accuracy at 0.01 seconds at least, but a target of 0.001 seconds was most desired.

3.4.6.2. Number of Known Bugs in Program

The number of bugs in any system is never desirable, but being a prototype, it would be expected that some known bugs will exist that could be resolved in future work. Therefore, a target of zero bugs was set, with a threshold of three bugs.

3.4.7. Budget

No official budget was agreed upon for the development of the system, however, the amount of contributions was limited. Development of the system would need to be implemented through contributions $600 from the College of Science and Engineering Thesis Budget and $400 from Dr. Elliott's staff account. It was desirable to have a target budget under $1,000; however, the author was willing to make $600 worth of contributions out of his personal account if required. Thus, a total budget of $1,600 was to be allocated to the design and development of the product as a threshold value.

3.4.8. Schedule

The schedule of the project was essentially set in relation to the due dates of the thesis. To allow enough time in writing the thesis, prototype development ideally needed to be complete by the 1st of October 2018. However, if required, any last-minute finalisations needed to be completed by the 10th of October 2018, so that there would be an opportunity to meet one final time with Dr. Elliot, Dr. Hobbs and Prof. Taylor to discuss the results and then finalise the thesis.

3.4.9. Finalised Specification Table

By utilising the tools used in the previous stages of the design process as well as further research as required, a detailed list of engineering specifications was determined that clearly defined the engineering specifications (Table 15). These engineering specifications were then able to act as a framework for design and development of the proposed sports agility tester.

Table 15: Engineering specifications and their target metrics

Engineering Specification	Target (Delighted)	Threshold (Disgusted)
Functional, Operational and Performance Requirements		
Intra-system communication	Yes	Yes
Physical connections between devices and the user	0 physical connections	0 physical connections
Set-up time	2 minutes	4 minutes
Steps required to set-up test	3 steps / device	8 steps / device
Type of device system can connect to	2 types	1 type
Number of characters displayable on marker	36 characters	1 character
Number of unique colours displayable on marker	9 colours	3 colours
Functions indoors and outdoors	Yes	Yes
Sensor response time	0.001 seconds	0.01 seconds
Adjustable tests difficulty	3 test difficulties	2 test difficulties
Number of markers	6 markers	4 markers
Number of sporting professions compatible with	4 sports	1 sport
Number of unique marker activation sequences	50 unique sequences	25 unique sequences
Maximised battery lifetime	8 hours	5 hours
Number of programming languages used	1 language	2 languages
When user feedback is received	Peri-test	Post-test
Number of feedback types	3 feedback type	1 feedback type
Number of user commands	5 commands	3 commands
Response time after user input command	< 1 second	< 3 seconds
Device boot-up time	< 3 seconds	< 10 seconds
Automatic test initiation upon athlete detection	Yes	No

Number of previous tests stored at any given time	500 tests	1 test
Device power modes	2 modes	1 mode
User can create custom tests	Yes	No
Sport-specific equipment tracking	Yes	No
Physical Requirements		
Area occupied by testing zone	100 m^2	25 m^2
Length and width of each marker	150 x 150 mm	200 x 200 mm
Height of each marker	250 mm	300 mm
Fit system inside a carry-case	0.07125 mm^3	0.11875 mm^3
Weight of total system	12 kg	16 kg
Marker load capacity	50 N	7 N
Ingress Protection	IP56	IP44
Adjustable height	2 settings	1 setting
Impact resistance	500 N	0 N
Workmanship and Manufacturing		
Number of parts	< 75 parts	< 150 parts
Internal parts enclosed	100%	90%
Mechanical tools required to change/recharge battery	0 tools	1 tool
Safety, Regulatory and Environmental Requirements		
Safe to use	Low risk profile	Medium risk profile
Extra-low voltage system	< 10 V	< 12 V
Noise produced by system	< 20 dB	< 60 dB
Noise produced by audio stimulus	100 –110 dB	80 – 99 dB
Number of loose parts	0 parts	2 parts
Human Factors		
Number of sensory pathways marker stimulus uses	2 sensory pathways	2 sensory pathways
Number of incorporated agility components	14 components	11 components
Number of tests before athlete is familiar with test	1 test	2 tests
Reliability, Maintainability and Supportability		
Accuracy of distance measurement sensor	< 2.5 cm	< 10 cm
Accuracy of time measurement	0.001 seconds	0.01 seconds

Number of known bugs in program	0 bugs	3 bugs
Variance in test results for a given athlete	< 0.15 seconds	< 0.5 seconds
Budget		
Manufacturing cost (excluding sundries)	< $1,000	< $1,600
Schedule		
System designed and built by due date	1 October 2018	10 October 2018

Legend

(blue)	Project area
(yellow)	2nd order specification

3.4.10. Quality-Function-Deployment Method

With a set of engineering design specifications developed and justified to act as a set of constraints for designing and developing the proposed sports agility tester, the quality-function-deployment (QFD) method was then utilised to compare the design specifications against the design requirements using a House of Quality (HOQ) chart (Figure 30).

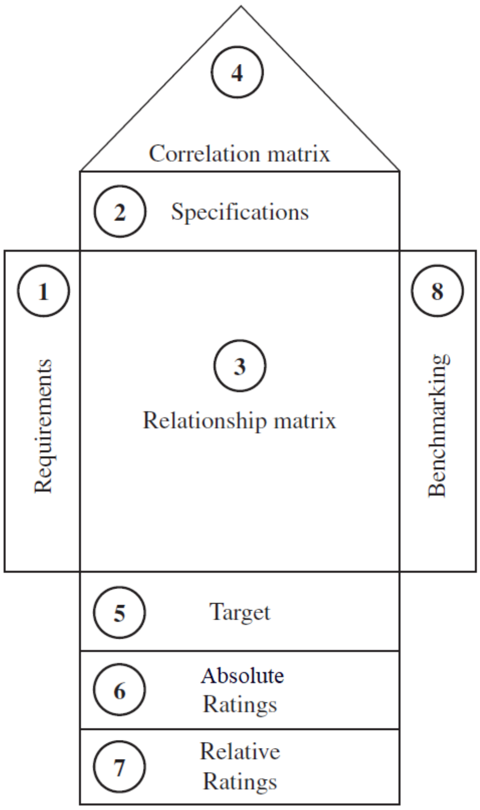


Figure 30: Basic structure of the House of Quality chart (Haik and Shahin, 2011)

The House of Quality is a useful tool that provides significant information: established requirements and specifications (Region 1 and 2, respectively); an establishment of correlations between the initially set design requirements and the set of design specifications (Region 3); establishment of correlations and conflicts between the design specifications (Region 4); specification target value metrics (Region 5); and absolute and relative importance ratings (Region 6 and 7, respectively) (Haik and Shahin, 2011). The HOQ is also capable of comparing the design specifications with competing products on the market (Region 8), but due to the complexity imposed by the significant amount of design requirements and specifications, as well as the specificity in novelty of the proposed device, it was decided that this region of the HOQ would not be implemented.

The House of Quality (Appendix B: B.1-B.4) had to be separated into multiple tables in order to fit into an A4 page size for printing purposes.

3.4.10.1. Correlated and Conflicting Specifications

Through the correlation matrix developed through the House of Quality (Table B- 1 in Appendix B: B.1), design specifications that were correlated were able to be identified. Specifications that contained a weak correlation were assigned a '1', a medium correlation a '3' and a high correlation a '9'. More importantly, specifications that were conflicting could also be established by assigning a '–'. This meant that to better achieve one (a more positive outcome), a trade-off needed to occur such that the other was affected negatively. Specifications found to have significant conflicts were the manufacturing cost, the system being designed and built by the due date, the number of bugs in the program, the total number of parts and a maximised battery lifetime. This emphasised the importance of carefully monitoring these specifications throughout the design process to ensure that they were met in the prototype delivery.

3.4.10.2. Requirement-Specification Relationships

The requirement-specification relationship matrix employed through the House of Quality (Table B- 2 in Appendix B: B.2) determined the correlations between each specification against each requirement. Similarly, to the specification correlation matrix, requirements and specifications that were deemed to have a strong correlation were given a value of '9', a medium correlation of '3' and a weak correlation of '1'. This step of the HOQ was used to determine whether all requirements had been properly addressed by the specifications. The requirement-specification relationship matrix showed that the design specification set appropriately reflected the design requirements.

3.4.10.3. Absolute and Relative Importance

Using the values from the requirement-specification matrix, the absolute and relative importance values could be determined for each specification based on the correlation scores it received. To calculate the absolute importance, the sum of correlation ratings multiplied by the requirement importance factor were determined as defined by Equation 5.

Equation 5: Calculating absolute importance

$$Absolute\ Importance = \sum(Correlation\ Rating \times Importance\ Factor)$$

An example calculation completed for the "set-up time" specification is presented (Appendix B: B.3). The same process was followed for all other specifications.

Using the total absolute importance values, the relative importance corresponding to each specification could be determined. Thus, all design specifications, corresponding threshold targets, absolute importance and relative importance values were tabulated (Table 16).

It was interesting to find that the system being *"safe to use"* was not presented as the most important specification; when this is generally considered the number one priority. Thus, the relative importance of this specification was considered quite unexpected. Another specification, the "*number of unique marker activation sequences"* was only considered to have a medium relative importance, which was deemed quite unexpected. One of the requirements was that the system should not be predictable, and the athlete should not be able to anticipate the next activated marker. Thus, with an increasing number of unique marker activation sequences, the probability that the athlete can predict the next marker activation becomes very small. It was presumed this specification would have a high relative importance, but it was not found to be the case. The specification, *"number of sensory pathways marker stimulus uses"*, had a low relative importance, which was unexpected. This was because it was considered vital that the stimulus used at a minimum two sensory pathways; ideally, an auditory and visual stimulus. It would not be acceptable for the system to present only one sensory pathway, so it was thought this specification would have a higher relative importance than it did.

Table 16: Engineering design specifications in order of greatest importance determined by the quality-function-deployment House of Quality

Design Specification	Threshold Target	Absolute Importance	Relative Importance (%)
No. of incorporated agility components	10 components	1874	**4.70**
Manufacturing cost	< $1,600	1814	**4.55**
System designed and built by due date	10-Oct-18	1752	**4.40**
Maximised battery lifetime	5 hours	1701	**4.27**
Intra-system communication	Yes	1686	**4.23**
No. of parts	< 150 parts	1636	**4.10**
Number of markers	4 markers	1554	**3.90**
Functions indoors and outdoors	Yes	1532	**3.84**
Safe to use	Medium risk profile	1498	**3.76**
Number of sporting professions compatible with	1 sport	1399	**3.51**
Number of known bugs in program	3 bugs	1199	**3.01**
Area occupied by testing zone	25 m^2	1176	**2.95**
Weight of total system	16 kg	1056	**2.65**
No. of unique marker activation sequences	25 unique sequences	1015	**2.55**
Physical connections between devices and the user	0 physical connections	987	**2.48**
Set-up time	4 minutes	907	**2.28**
Fit system inside a carry-case	< 0.11875mm^3	813	**2.04**
Steps required to set-up test	8 steps / device	802	**2.01**
Accuracy of distance measurement sensor	< 10 cm	795	**1.99**
No. of feedback types	1 feedback type	785	**1.97**
Automatic test initiation upon athlete detection	No	781	**1.96**
Impact resistance	0 N	763	**1.91**

Design Specification	Threshold Target	Absolute Importance	Relative Importance (%)
Variance in test results for a given athlete	< 0.5 seconds	736	**1.85**
Sport-specific equipment tracking	No	729	**1.83**
Adjustable height	1 setting	713	**1.79**
No. of displayable characters on marker	1 character	662	**1.66**
No. of unique colours displayable on marker	3 colours	662	**1.66**
User can create custom tests	No	651	**1.63**
Sensor response time	0.01 seconds	650	**1.63**
User feedback	Post-test	637	**1.60**
Marker load capacity	7 N	615	**1.54**
Ingress Protection	IP44	613	**1.54**
Adjustable tests difficulty	2 test difficulties	575	**1.44**
Length x width of each marker	200 x 200 mm	528	**1.32**
Internal parts enclosed	90%	490	**1.23**
Noise produced by audio stimulus	80 – 99 dB	469	**1.18**
Accuracy of time measurement	0.01 seconds	453	**1.14**
No. of tests before athlete is familiar with test	2 tests	385	**0.97**
No. of sensory pathways marker stimulus uses	2 sensory pathways	378	**0.95**
Number of user commands	3 commands	326	**0.82**
Height of each marker	300 mm	308	**0.77**
Response time after user input command	3 seconds	273	**0.68**
Type of device system can connect to	1 type	264	**0.66**
Tools required to change/recharge battery	1 tool	222	**0.56**

Design Specification	Threshold Target	Absolute Importance	Relative Importance (%)
No. of previous tests stored at any given time	1 test	213	**0.53**
Device power modes	1 mode	180	**0.45**
Extra-low voltage system	< 12 V	172	**0.43**
Number of loose parts	2 parts	138	**0.35**
Noise produced by system	< 60 dB	114	**0.29**
Device boot-up time	10 seconds	98	**0.25**
Number of programming languages used	2 languages	82	**0.21**

By employing the House of Quality through utilisation of the quality-function-deployment process, the design specifications that were deemed to have the most significant importance were able to be determined (Table 17). These specifications were found to have the most design requirements correlated with them, so they were found to be the specifications to monitor carefully.

Table 17: Most important specifications determined through the House of Quality

Specification	Relative Importance (%)	Analysis
Number of incorporated agility components	**4.70**	One of the very reasons as to why this project is being undertaken. The system needs to be able to incorporate as many components of agility as possible, with a focus in the physical, perceptual and decision-making factors.
Manufacturing cost	**4.55**	Cost of manufacturing the prototype was deemed significantly important. The budget meant that components would need to be researched and compared against one another to decide carefully the most suitable at the lowest cost possible.
System designed and built by due date	**4.40**	The project was identified to be substantial in the objectives to complete. To have a successful system by the end of it, care would need to be taken to manage time wisely and focus on the specifications which are most important.
Maximised battery lifetime	**4.27**	With the system being wireless and portable, a maximised battery life is important. With the increasingly large number of functions to accomplish, the amount of electronic hardware grows, thus, careful selection needs to be made to ensure battery life is not adversely affected.
Intra-system communication	**4.23**	Communication between devices is important as the markers need to be able to communicate when a ball has been removed and when the next marker needs to be activated. Thus, the intra-system communication is vital to ensure randomness of the test.
Number of parts	**4.10**	An increasing number of functions means an increasing number of parts to achieve the aims and objectives.

Specification	Relative Importance (%)	Analysis
Number of markers	**3.90**	The number of markers will establish the total number of sequences possible to be implemented, as well as total testing area zone (which will dictate the total time taken to set-up the test), the complexity of setting up and even the time taken to build the system prototype. There are significant trade-offs when selecting an appropriate number of markers.
Functions indoors and outdoors	**3.84**	Designing and developing a prototype that would be able to function indoors and outdoors would be a significant challenge. The mechanical design would need to be developed appropriately, as well as close attention to selection of electronic hardware, which is resistant to environmental factors such as direct sunlight.
Safe to use	**3.76**	Safety is always an important specification to follow. Care would be taken to ensure the prototype is safe to use by the administrator and the athlete.
Number of sporting professions compatible with	**3.51**	It would be important to ensure that the test could function with at minimum, Australian rules football. Time permitting, a design should be developed, which is compatible with other sports too, such that the target market is increased significantly.

3.5. Concept Development

Up until this point, the project had progressed through the design process in a systematic and logical manner. The problem definition stage established identifying needs, clarifying objectives, conducting market research and setting a set of clearly defined design requirements. This brought forth the conceptual design stage, establishing functional structures and developing an appropriate set of design specifications with target values that reflected the previous stages of the project. Using the quality-function-deployment method, the House of Quality was able to establish correlations between requirements and specifications, such that the most important specifications could be determined. These specifications would now be the forefront in conceptualisation and development of the prototype as the project entered the solution concept stage of the design process.

It is in this stage that multiple solutions would be generated and evaluated against one another to determine the best possible concept solution for the project. The aim was to develop a concept solution, which was optimised through thorough analysis and evaluation of the alternatives.

3.5.1. Preliminary Conceptualisation

Osborn (2013) found that by following two simple principles produced better ideas; “deferment of judgement” and “quantity breeds quality”. Thus, in order to stimulate creative thinking, the well-known creative method of brainstorming (developed by Osborn) was first conducted, which enabled generation of a broad scope and large number of ideas, without criticism to avoid inhibition of a creative flow (Cross, 2000; Thompson and Lordan, 1999). Osborn presented four basic rules to be followed; criticism should be avoided (deferred judgement), freewheeling to be welcomed (which may stimulate originality), quantity is wanted and combinations and improvements to be sought out. Therefore, these principles were followed whilst brainstorming in this project (Appendix C: C.1). The first brainstorming session was quite broad, delving into areas such as initiation of the test, signal deactivation, audio and visual stimuli, measurement protocol, safety and programming (Figure C- 1). The next brainstorming session specifically was for the mechanical design, which investigated areas such as height adjustment, shape, material, inner design and how to position the ball on the marker (Figure C- 2). These brainstorming sessions aimed to get ideas out there and begin the flow of creativity.

3.5.2. Generation of Design Alternatives

To develop a set of design alternatives, a morphological chart could be developed, which is a tool for generating a set of means to accomplish primary functions of a design. By breaking down the design into a primary function established through the functional decomposition conducted when determining the function structure, the design was able to be undertaken in a more manageable way. The morphological chart was used as a basis for identifying possible solution alternatives to meet the functions specified, with further additions made through iteration, as more ideas were presented that established further functions needing to be achieved.

Morphological charts were developed for the electrical and software components of the design (Figure 31) and for the mechanical components of the design (Figure 32). The morphological charts show the final charts that were developed, which include functions and their corresponding means that were discovered to be needed later in the project when more in depth component selection was conducted. Therefore, to maintain the flow of this thesis, there are some functions discussed later in Section 3.6: Concept Evaluation.

Function	Means							
Electrical Design								
Single-board Microcontroller	Arduino	Raspberry Pi	Teensy 3.6	Mbed	PIC32	BeagleBone Black	Pololu A-Star	Flinduino
Measure marker distance	Ultrasonic Sensor	Infrared Sensor	LiDAR	Laser Distance Measurer	Tape Measure	Measuring Wheel	Laser Range Finder	
Sense position in space	Gyro Sensor	IMU	Accelerometer					
Rotate sensor	Stepper Motor	DC Motor	Servo Motor	Manually				
Motor Encoding	Transmissive Photo-interrupter	Absolute Encoder						
Resist tangled wires	Slip-ring	Mount Processor with Sensor						
Display visual stimulus	LED	LCD Screen	LED Matrix	14-Segment Display				
Character Display	Letters	Numbers	Symbols					
Feedback mechanism	LED	LCD Screen	LED Matrix	Segment Display	Speaker	Laser Diode		

Function	Means							
Sense ball	Force Sensitive Resistor	Touch Capacitive Sensor	LiDAR	Infrared Sensor	Proximity/ Ambient Light Sensor	Ultrasonic Sensor	Mechanical Switch	
Auditory Stimulus	Piezo Buzzer	Small Speaker	Standard Car Speaker					
Power source	Ni-MH	Li-Po	Li-Ion	Ni-Cad	Lead Acid	Alkaline	Lithium Coin Cell	LiFePo4
User/System interface/ input	Serial USB	Ethernet	Bluetooth	Wi-Fi	Momentary Button	Rotary switch		
Intra-system communication	Bluetooth	Wi-Fi	Nordic	General RF	Cellular			
Initiate test	Self-initiated (Button)	Automatic Sensor)	Through the User/System Interface					
Software Design								
Programming Language	C/C++	C#	Python	Java	JavaScript	Ruby		

Figure 31: Electrical and software design morphological chart

Function	Means							
Mechanical Design								
Assist in Detection of Ball	Spring-Loaded	Shaped Mechanical Design						
Enclosure	Manually Build	Purchase Commercial Product						
Enclosure Material	PVC/PC Sheet	Vacuum moulding	Bucket	PVC Pipe	3D Printing	Wood	Metal	Laser-cut Acrylic
Sensor Rotation	Laser-cut Acrylic Gears	Commercial Metal Gears	Belt and Pulley	Chain Drive	Drive Belt	Worm Drive		
Shield Sensor from Environment	Acrylic	Glass	Polycarbonate	Clear CCTV Dome	Environment Resistant Sensor			
Anti-slip	Rubber Mat	Adhesive Bumper Strip	Adhesive Pad Protector					
Hardware Mounting	Metal Brackets	3D-Print	Double-sided Tape	Velcro	Cable Ties			

Figure 32: Mechanical design morphological chart

3.5.2.1. *Evaluating Design Space*

From the morphological charts developed, the corresponding design space was able to be determined, which is the total number of design combination possibilities (Dym et al., 2014). The design space was determined by multiplying each number of means corresponding to each function by one another, as shown in the calculation below:

$$8 \times 7 \times 3 \times 4 \times 2 \ldots \times 5 = 1.69 \times 10^{14} \text{ combinations}$$

A complete table that represents the total number of means corresponding to each function was developed (Table D- 1 in Appendix D: D.1) alongside a full calculation conducted. The above value is a very approximate value, since there are some functions that could use more than one mean listed, but also some functions could use the same mean as one used in another function, thus the total number of possible combinations may vary. Furthermore, it is also important to note that not all combinations are feasible, as the combinatorial arithmetic does not account for the need for parts and hardware to synchronise into a common solution. Nevertheless, it is clear to see an overwhelmingly large number of possible combinations.

Therefore, instead of using the traditional method of developing a set of design ideas that used a mixture of means, it made more sense in this project to address each function separately by assessing the best possible mean that corresponded to each discrete function. Through this, the most optimal mean could be selected for each function, which could then be integrated together into a final concept solution. In some cases, multiple functions were addressed together when it was deemed appropriate to do so.

3.5.3. Evaluation of Design Alternatives

Through the morphological charts developed for the electrical and software components of the design (Figure 31) and the mechanical components of the design (Figure 32), the means for achieving specific functions had been listed. Now a more in-depth approach was able to be taken, to evaluate the design alternatives and to select the best approach to take in achieving each function. An evaluation of design alternatives was completed through a decision matrix, which is a tool for comparing alternatives via a numerical approach. The process was followed from the procedure outlined by Haik and Shahin (2011).

Each decision matrix had a set of criteria customised to best evaluate the alternatives to achieve the optimum solution for the given function. The set of criteria were given a weighting factor based on the design specifications developed in the previous stage; the weighting factors needed to add up to 100% as stated by Haik and Shahin (2011). Each alternative was then evaluated as to how well it met the criteria, receiving a rating factor between 0 – 10. Similarly, to how the House of Quality was employed through the quality-function-deployment method, the weighting factor and rating factor was multiplied together and summed across all the criteria for each design alternative. The alternative with the greatest score was deemed the best possible mean for achieving the given function.

3.5.3.1. Single-Board Microcontrollers

Whilst still early in the conceptualisation stage of the project, it was unclear the amount of electronic hardware components that would be required to provide a solution to the design problem. It was also unclear of the number of connections which would be made to the microcontroller and what type of connections would be made (I^2C, SPI, UART, TX/RX, Analogue, Digital or PWM to name a few). It was also unknown how large the program would become (number of variables, program length), so memory requirements were also unknown. Therefore, it was essential to choose a microcontroller that had a larger capability (large memory, RAM, input/output pins, analogue pins, all communication methods) so that there was flexibility and allowance in componentry selection. Having a microcontroller with a larger capacity of features reduced the risk of not having the minimum requirements to produce a solution that provided the essential features. This would also help make the prototyping stage a lot easier, as there would be more flexibility in where components could be connected to and if peripherals needed to be changed for alternatives. Furthermore, it was essential to choose the microcontroller first to ensure compatibility of peripheral hardware, since some components may only work for some microcontrollers (for example, it may work with Raspberry Pi but not Arduino). As multiple design iterations would be required before the commercially available product, minimisation and reduction of non-essential features could be removed later.

Several types of microcontrollers were investigated into the feasibility for this project; Arduino, Raspberry Pi, Teensy, Mbed, PIC32, Pololu A-Star, BeagleBoard or Flinduino. It should be noted that the Raspberry Pi is not a single-board microcontroller, but really a single-board computer, but will be referred to as a microcontroller from here for convenience in discussion in this thesis (Little Bird Company, 2018a). It was known that multiple variants existed between companies, so it was essential to choose an appropriate model when comparing. The product with the largest capability with most reasonable cost was considered. As a result, the following microcontrollers were selected as alternatives to evaluate; Arduino Mega 2560 (Arduino, Ivrea, Italy), Raspberry Pi

Model B+ (Raspberry Pi Foundation, Cambridge, United Kingdom), Teensy 3.6 (PJRC, Sherwood, Oregon, USA), PIC32 microcontrollers in general (Microchip, Chandler, Arizona, USA), A-Star (Pololu Corporation, Las Vegas, Nevada, USA), BeagleBone Black (BeagleBoard, Michigan, USA) and the Flinduino (Flinders University, Adelaide, Australia).

To assist in the evaluation process, a detailed comparison was made between the various microcontrollers for the most important features (Table E- 1 in Appendix E: E.1). The table compares cost, input/outputs (I/O), analogue input, programming language, memory, RAM, CPU clock speed and operating/input voltage.

3.5.3.1.1. Microcontroller Alternatives Decision Criteria

The criteria that was deemed most important in evaluating the microcontroller features was that it was low-cost, the number of I/O pins it contained, the ease of programming, if it could directly interface with hardware, the clock speed, RAM, memory and the experience the author had with the product (Table 18).

Low-cost was considered significantly important as it aligned with keeping the *"manufacturing cost"* specification down, which had a high relative importance of 4.55% (Table 16). As a result, this criterion was given a weighting of 40%. The number of I/O ports was significantly important, as mentioned earlier that there needed to be an allowance for adaptability of multiple types of connections for the peripheral hardware, thus was given an importance weighting of 20%. Ease of programming and use experience both corresponded with *"system designed and built by the due date"*. Although this specification had a high relative importance (4.40%) (Table 16), the weighting factors for these criteria were low (both 5%), as the author was confident in their programming skills, so the learning curve for adapting to a new microcontroller would not be as high as for someone just beginning to program. Directly interfacing with hardware, clock speed, RAM and memory were considered equally important in the decision process, so were each given a weighting of 10%.

The criteria that was deemed the most important features of the microcontroller and their corresponding weightings were then tabulated (Table 18).

Table 18: Microcontroller decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Low-cost	40
Number of inputs/outputs	20
Ease of programming	5
Directly interface with hardware	10
Clock Speed	10
RAM & Memory	10
Use Experience	5
Total	**100**

3.5.3.1.2. Microcontroller Alternatives Evaluation

Using the criteria and corresponding weighting factors, the decision matrix for evaluating which microcontroller alternative to use could be developed (Figure 33). The decision matrix found the Arduino Mega 2560 to be a suitable candidate with a total score of 7.1. Factors that pushed this alternative to have the highest score was cost, the number of I/O pins, the ease of programming and that it could directly interface with hardware. Arduino single-board microcontrollers can be bought at an extremely low cost when purchasing a "clone" version, which is a copy of the original made by a company other than Arduino. This is possible as Arduino is an open-source hardware and software company. With multiple libraries available, Arduino microcontrollers are very easy to program, with the benefit of using C/C++, which the author is very familiar with. Electronic hardware can easily interface with the Arduino, which would make connecting sensors and other peripherals far easier.

Microcontroller Alternatives \ Weighting \ Design Criteria	Low-cost	Number of inputs/outputs	Ease of programming	Directly interface with hardware	Clock Speed	RAM & Memory	Use Experience	Total
	0.4	0.2	0.05	0.1	0.1	0.1	0.05	1
Arduino Mega 2560	10 / 4	7 / 1.4	10 / 0.5	10 / 1.0	1 / 0.1	1 / 0.1	0 / 0	7.1
Raspberry Pi 3 Model B+	5 / 2.0	5 / 1.0	8 / 0.4	4 / 0.4	10 / 1.0	5 / 0.5	10 / 0.5	5.8
Teensy 3.6	6 / 2.4	6 / 1.2	10 / 0.5	10 / 1	6 / 0.6	5 / 0.5	0 / 0	6.2
FRDM-K64F Mbed	6 / 2.4	2 / 0.4	8 / 0.4	10 / 1	5 / 0.5	5 / 0.5	10 / 0.5	5.7
PIC32[5]	2 / 0.8	10 / 2	8 / 0.4	10 / 1	4 / 0.4	4 / 0.4	4 / 0.2	5.2
BeagleBone Black	4 / 1.6	8 / 1.6	9 / 0.45	10 / 1	10 / 1	10 / 1	0 / 0	6.65
Pololu A-Star	7 / 2.8	3 / 0.6	8 / 0.4	10 / 1	1 / 0.1	5 / 0.5	0 / 0	5.4
Flinduino	9 / 3.6	3 / 0.6	10 / 0.5	10 / 1	3 / 0.3	2 / 0.2	2 / 0.1	6.3

Figure 33: Decision matrix for microcontroller alternatives

Legend

x	Rating factor (0 - 10)
y	Relative importance

3.5.3.2. Measure Marker Distance

Measuring the distance between markers was deemed a very important characteristic of the system as it would be required to maintain standardisation of the test. The various means identified in achieving this was using an ultrasonic sensor, infrared sensor, light detection and ranging scanner (LiDAR), laser distance measurer, tape measure, measuring wheel or laser range finder. These means could be separated into two categories; hardware based and human based. A system that can measure marker distance using hardware would be far more convenient and faster than one where a human must do the measuring; however, it is also costlier and time consuming as programming, calibration and validation needs to be completed.

Referring to the design specifications, the *"steps required to set-up the test"* and *"time taken to set-up the test"* had relative importance ratings of 2.01% and 2.08% (Table 16), respectively. Additionally, the correlation matrix developed from the House of Quality (Table B- 1 in Appendix B: B.1) showed that there was a negative correlation between these specifications and the *"number of markers"*, which had a relative importance of 3.9% (Table 16). This meant that it was desirable to increase the number of markers, but this in turn would increase the complexity of the set-up process. Therefore, having a system that is self-aware and able to assist in measuring the markers would be ideal to meet the test set-up specifications and assist in making the system simpler to use and easing the set-up process complexity. Using a system to measure the distances between markers would remove human-errors and help make the test more consistent and able to be standardised when used across various sporting clubs or institutions. Therefore, it was deemed necessary to attempt to develop a system that could assist the user by automatically detecting the distances of the markers and determining the location required; this would remove human-associated means. A detailed comparison chart that compared features of each device type regarding total cost, typical range value, if it is usable outdoors, the resolution, field-of-view, if it is Arduino compatible and any further notes were tabulated (Table E- 2 in Appendix E: E.2).

3.5.3.2.1. LiDAR Scanner

Generally, LiDAR scanners are quite expensive with prices in the thousands. However, there were found to be a few that were of low-cost, between $229 and $595 including the Scanse Sweep (Scanse LLC, San Leandro, California, USA), LiDAR-Lite (Garmin Corporation, New Taipei City, Taiwan), RPLIDAR (Shanghai Slamtec Co, Shanghai, China) and the TeraRanger Evo (Terabee, St Genis-Pouilly, France). With these lower prices that would meet the budget restrictions, it meant that the LiDAR scanner was a feasible hardware option for measuring distance within the system.

3.5.3.2.2. Distance Measurement Alternatives Decision Criteria

To select an appropriate distance measuring device, a set of criteria was developed to assess each alternative mean against. These were found to be low-cost, maximum distance measurement, usable outdoors, resolution (accuracy), field-of-view (how wide angle the sensor detects), compatibility with Arduino and if it could be made autonomous (no need for human interference) (Table 19). As with the microcontroller criteria, low-cost was considered most important, with a weighting factor of 25%. The specification *"functions indoors and outdoors"* had a very high relative importance (3.84%) (Table 16), so it was imperative the device could be used outdoors, combating the natural environment such as sunlight, wind and rain. The specification *"area occupied by testing zone"* also had a high relative importance (2.95%) (Table 16). Thus, it was important for the sensor to be able to detect distances far away. As a result, these two criteria had importance weightings of 20%. It was important that the hardware was compatible with Arduino, so this criterion received a weighting of 15%. The specification *"set-up time"* and *"steps required to set-up test"* had higher importance factors (2.28% and 2.01%, respectively) than the importance factor for the *"accuracy of the distance measurement sensor"* (1.99%) (Table 16). Therefore, it was deemed more important to ensure the sensor could be used autonomously (weighting factor 10%), without the need for human interference to make measurements of the markers, than have a sensor that had a higher resolution (5% weighting factor). The criteria and corresponding weighting factors were then tabulated (Table 19).

Table 19: Distance measurement decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Low-cost	25
Max distance measurement	20
Usable outdoors	20
Resolution	5
Field-of-View	5
Arduino Compatible	15
Autonomous	10
Total	**100**

3.5.3.2.3. Distance Measurement Alternatives Evaluation

Using the criteria and corresponding weighting factors, the decision matrix for evaluating which distance measurement sensor alternative to use could be developed Figure 34). The decision matrix found the use of a LiDAR scanner to be most suitable with a total score of 8.6. Factors that pushed the LiDAR scanner to have such a high score was that it was deemed a great all-rounded device that met most of the criteria well. With a variety of cheaper LIDAR scanners available, the device could be low-cost when compared against the laser distance measurer, laser range finder and the amount of ultrasonic and IR sensors required to produce a working solution concept. LiDAR scanners are generally not affected significantly by outdoor environments due to the use of a laser, the field-of-view is very small, they can be programmed directly using and Arduino and they can be made to measure autonomously.

Criteria / Weighting / Alternatives	Weighting	Ultrasonic Sensor	Infrared Sensor	LiDAR Scanner	Laser Distance Measurer	SF02 Laser Range Finder
Low-cost	0.25	9 / 2.25	6 / 1.5	8 / 2	5 / 1.25	3 / 0.75
Max distance measurement	0.2	3 / 0.6	3 / 0.6	7 / 1.4	10 / 2	8 / 1.6
Usable outdoors	0.2	5 / 1	0 / 0	10 / 2	10 / 2	10 / 2
Resolution	0.05	8 / 0.4	4 / 0.2	4 / 0.2	10 / 0.5	4 / 0.2
Field-of-View	0.05	3 / 0.15	2 / 0.1	10 / 0.5	10 / 0.5	10 / 0.5
Arduino Compatible	0.15	10 / 1.5	10 / 1.5	10 / 1.5	0 / 0	10 / 1.5
Autonomous	0.1	10 / 1	10 / 1	10 / 1	5 / 0.5	10 / 1
Totals	1	6.9	4.9	8.6	6.75	7.55

Figure 34: Decision matrix for distance measurement sensor

Legend

x	Rating factor (0 - 10)
y	Relative importance

3.5.3.3. *Sense Position in Space*

Sensing position in space was only relative to design solutions that required the marker to be able to know its position and angle of rotation. If using an ultrasonic or IR sensor, then the idea of manually rotating the marker to detect the two markers next to it was thought up. Thus, having evaluated the means for distance measurement and finding the LiDAR sensor to be most suitable; the need for sensing position in space was not necessary to meet the engineering specifications and design requirements.

3.5.3.4. *Rotate Sensor, Motor Encoding & Resisting Tangled Wires*

At this point in the project, it was unknown that sensor rotation would be required; it was not until the LiDAR scanner was selected that this function was found needing to be solved. As a result, evaluating a sensor rotation mechanism, subsequent motor encoding and resistance of tangled wires is discussed in Section 3.6.2: Rotating the LiDAR and Section 3.6.3: Motor Encoding.

3.5.3.5. *Display Visual Stimulus, Auditory Stimulus & Character Display*

One of the non-negotiable design requirements was to display a visual and audio stimulus when a marker becomes activated, so that an athlete can react and run towards the marker. Incorporating a stimulus into the test would increase the *"number of incorporated components of agility"* in an athlete performing the test, meeting the most important specification with importance of 4.7% (Table 16). Having both an audio and visual stimulus would mean that two sensory pathways are activated for the athlete, increasing the number of sensory pathways the marker uses (relative importance of 0.95% from Table 16). When visually scanning for the next marker to activate, the auditory cue would help them know which direction the marker is in, then the visual stimulus would help to confirm which marker to run towards. Additionally, having two sensory pathways activated means that the system accounts for any possible disabilities an athlete may possess, which may impair them from "normal" conditions in relation to seeing or hearing.

Choosing the type of visual stimulus was quite straightforward. It was known that it needed to be able to display a character, so this ruled out using one single LED. The remaining possible alternatives to use as the visual stimulus was an LCD screen, LED matrix or a segmented display.

3.5.3.5.1. LCD Screen

Liquid-crystal displays (LCDs) were an alternative investigated into. There are a few types of different LCD screens; the ones researched were numeric serial enabled and graphics. One is more for displaying lines of text (numeric serial enabled), while the another can be used to display images or graphics. Some various LCD screens that are compatible with Arduino microcontrollers are shown in Figure 35.

Figure 35: Various Arduino compatible LCD screens (Little Bird Company, 2018b)

3.5.3.5.2. LED Dot Matrix

The LED dot matrix was considered a viable option as the visual stimulus. It is made up of multiple LEDs (referred to as pixels in this context) arranged in a matrix. It could be constructed in a way that allows a multitude of characters to be displayed. It was determined that the matrix would need to be a minimum of 7 x 5 pixels (35 pixels) to display all alphanumeric characters (Figure 36). This was confirmed through Kodak's Ergonomic Design for People at Work (The Eastman Kodak Company, 2004).

Figure 36: Alphanumeric characters displayed on an LED matrix (Hengpattanapong, 2018)

LED matrix displays generally consist of a red, green and blue (RGB) LED. The matrix can display a wide variety of colours using additive colour through pulse width modulation (PWM) output, which controls the duty cycle of each LED light (the ratio of time the LED is on to the time it is off), thus adjusting the brightness intensity. Some examples of LED matrix displays are shown in Figure 37.

Figure 37: Various LED matrix displays (Adafruit, 2018)

3.5.3.5.3. Segment Display

Segment displays were investigated, which presented the possibility of 7-, 9-, 14- or 16-segments. The most common segment displays are the 7-segment display (Figure 38a and Figure 38b) and 16-segment display (Figure 38c and Figure 38d).

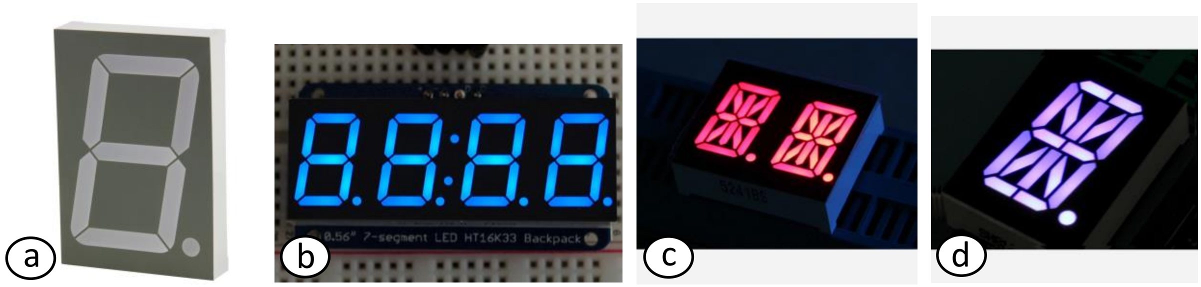


Figure 38: Various segment displays (Little Bird Company, 2018b)

A 7-segment was not able to display all characters clearly; for example, it is not possible to differentiate a ‘B’ from an ‘8’. Additionally, a 9-segment display cannot represent characters such as ‘N’, ‘M’, ‘R’ or ‘Y’. Thus, the minimum number of segments to clearly differentiate characters was found to be a 14-segment display (FSD), however, a 16-segment display (SISD) would present a larger number of possible characters (Figure 39).

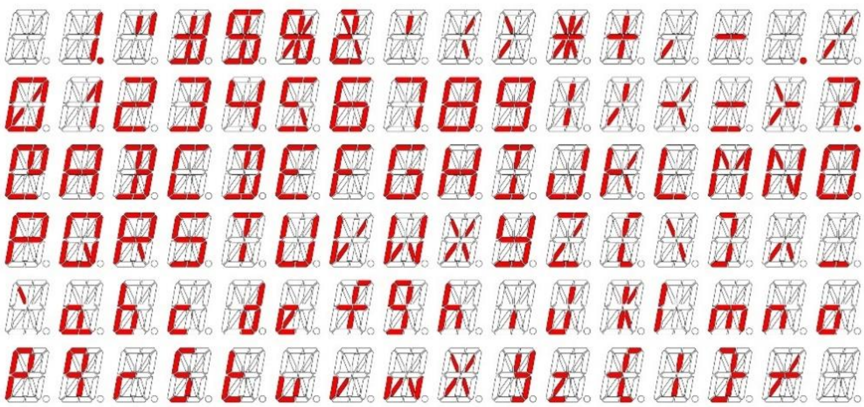


Figure 39: Displaying characters on a segment display (Parts Not Included, 2017)

3.5.3.5.4. Visual Display Alternatives Decision Criteria

A set of criteria was developed to evaluate the visual display alternatives against. The aspects that were deemed important to compare against was the number of displayable characters, multiple colour capability, size, low-cost and minimised current draw (Table 20). Aligning with both the microcontroller selection and distance measurement sensor, low-cost had the highest weighting (35%), which aligned with the important specification of keeping *“manufacturing cost”* down. Additionally, ensuring a maximised battery was of high importance (4.40%) (Table 16). With a lot of hardware requirements, minimised current draw was needed for all components, thus this criterion was given a 25% weighting to reflect this. The number of displayable characters and unique colours had medium relative importance values (1.66%) (Table 16), which was important to maintain the unpredictability and randomness of the test; so, these two criteria received a 15% weighting. The specification for length and width of each marker and height had low relative importance values of 1.32% and 0.77% (Table 16), respectively. Thus, the size of the display was only given a weighting of 10%. The criteria and respective weighting factors were then tabulated (Table 20).

Table 20: Distance measurement decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Number of Displayable Characters	15
Can Display Multiple Colours	15
Large Size	10
Low-Cost	35
Minimised Current Draw	25
Total	**100**

3.5.3.5.5. Visual Display Alternatives Evaluation

Using the criteria and corresponding weighting factors, the decision matrix for evaluating the visual display hardware alternatives was developed (Figure 40). Through the matrix, it was found that the LED matrix received the highest score. There were many benefits with this component; there was an extremely large amount of characters displayable (including all alphanumeric), they can display a wide variety of colours through PWM duty cycle control, they can be of quite a large size and similarly to “clone” versions of the Arduino, there were cheap clones available.

The LCD screen was very expensive, used a lot of power and parts suitable for the Arduino were not very large. It was found most segment displays were either very small, which was an issue as it needed to be viewed from distance out in direct sunlight. Alternatively, they were extremely large displays; there was no in between, which meant a trade-off in the mechanical design size. The major consideration that eliminated the segment display was that it was not able to display more than one colour.

Criteria	Weighting	LCD Screen	LED Matrix	Segment Display
Number of Displayable Characters	0.15	10 / 1.5	9 / 1.35	6 / 0.9
Can Display Multiple Colours	0.15	10 / 1.5	10 / 1.5	4 / 0.6
Large Size	0.1	6 / 0.6	10 / 1	4 / 0.4
Low-Cost	0.35	3 / 1.05	10 / 3.5	10 / 3.5
Minimised Current Draw	0.25	3 / 0.75	5 / 1.25	7 / 1.75
Totals	1	5.4	8.6	7.15

Legend

x	Rating factor (0 - 10)
y	Relative importance

Figure 40: Decision matrix for visual stimulus display

3.5.3.5.6. Test Layout Conceptualisation

The way the markers would be set-up in the test layout was an important factor to consider. Originally, it was thought that with four markers, the test would be set-up in a square, with a marker at each corner. Other ideas were presented on the use of perhaps six markers instead of four, which could be positioned to form a regular hexagon (Figure F- 1 in Appendix F: F.1). An increasing number of markers would be ideal to increase the number of possible locations the athlete would need to run to. A hexagon was decided as the layout of choice if six markers were to be used, because the distance between each marker was consistent and would assist in allowing the test to remain as standardised as possible. Other possible configurations could be investigated; however, this would involve in depth research and analysis. Now that it was known that LED matrix displays would be used as the visual stimulus, further conceptualisation was developed with the possibility of using multiple displays per marker (Figure F- 1 and Figure F- 2 in Appendix F: F.1).

3.5.3.5.7. Auditory Stimulus

Selecting an appropriate audio stimulus was trivial; all it would need was to be loud enough to be heard when outdoors. The choices were either a piezo buzzer, a small speaker or a standard car speaker. Standard car speakers can range anywhere from 4” to 6.5” in diameter. As the design would likely need a lot of components, care had to be taken to ensure that there would be enough room within the enclosure. Thus, the size factor ruled out selecting a car speaker. Additionally, it was unknown how they would perform with the prototype.

With a small speaker or piezo buzzer left to decide, the problem was knowing what would be loud enough to order the correct part. At the point in time for deciding between the two, there was not a lot of time left, as lead times needed to be accounted for as well as development. It was decided that both a speaker and a piezo buzzer would be purchased, since a piezo buzzer was found to be very cheap (< $2) and it could be used as a feedback mechanism anyway.

3.5.3.6. Feedback Mechanism

User feedback received a medium importance rating of 1.66% (Table 16). A feedback mechanism was required for the system so that a user is made aware of when input has been received and processed, when the test has begun and when it has finished. This was deemed important when conducting the literature review, through experimentation with the Freelap Timing System and observing that this system did not provide feedback when a test had initiated, which provided uncertainty if the test was recording the time or not.

The specification *“number of feedback types”* had a medium importance rating of 1.97% (Table 16). As a result, it was decided that the system would incorporate both a visual and auditory feedback mechanism. As described above, the piezo buzzer was very cheap and was considered for the auditory stimulus, thus was purchased with the backup plan of using it as a feedback mechanism. As it was determined that LED matrix displays would be used as the visual stimulus, these could also be used as a feedback mechanism too. Thus, selecting the auditory and visual stimulus provided accomplishment of meeting other specification needs.

3.5.3.7. Sense Ball and Assistance in Detection using Mechanical Design

One of the essential design requirements was the integration of sport-specific equipment (such as a game ball) with the system. It was established that the ball would be placed on the marker, where the athlete would be required to remove it. By removing the ball, it would deactivate that marker, thereby reactivating another one. For this to occur, the marker needed a method of sensing that the ball was positioned on it and sensing when it has been removed. This could be accomplished with a purely electrical solution using just hardware, or a mechanical-electrical combinatory design could be implemented.

There were thoughts on using a spring-loaded system in combination with a mechanical switch or force-sensitive resistor (a type of resistance sensor). As a ball would be placed on the marker, it would push a platform down that would be spring-loaded (Figure F- 3 in Appendix F: F.1). The force-sensitive resistor would sense the force imposed through the weight of the ball; a sense of force would mean the ball is there and no force would mean it has been removed. Similarly, the same concept could be used but with a mechanical switch replacing the force-sensitive resistor.

The idea of using a touch capacitive sensor was investigated, but it was found that this sensor required a conductive object to touch it (the object needs to hold an electrical charge) to sense a change in capacitance. Thus, this would be a sensor that is more inclined to be used with the touch of an individual’s finger (which is conductive). As a result, this design alternative was not further considered in the design making process.

Purely electrical design solutions considered was the use of a LiDAR sensor, IR sensor or ultrasonic sensor. As the sensor would only need to detect the sport-specific equipment a short distance away, another type of distance measurement sensor was considered, the proximity sensor, which is able to detect objects at close range.

3.5.3.7.1. Proximity Sensor Integrated with Ambient Light Sensor

There are several principles that can be applied for the application of a proximity sensor; infrared, capacitive, doppler effect, inductive, magnetic, optical, radar, sonar or photocell. The principle method considered in this application was the use of an infrared. Normally, an infrared sensor is unable to be used in very bright ambient light conditions, such as in direct sunlight outdoors, however through research and investigation, it was determined that when integrated with an ambient light sensor (ALS), filtering techniques could be used for ambient light cancellation to correctly measure the distance in direct sunlight (Silicon Laboratories, 2013; Vishay Intertechnology, 2015).

3.5.3.7.2. Sport-Equipment Sensing Alternatives Decision Criteria

The hardware to sense the sport-specific equipment was deemed one of the most essential components of the device, thus, extra care needed to be taken when evaluating the alternatives available. The criteria deemed most suitable to compare against was low-cost, resistance to sunlight, waterproof design, minimised complexity and the confidence in developing a design that would be applicable in this context (Table 21).

A low-cost component was essential, as mentioned previously, so this criterion received a weighting factor of 30%. In line with meeting the project due dates, the component that would result in the least complex design would be preferred, to minimise possible risks of failure, as well as simplicity in design. As a result, minimised complexity received a weighting factor of 25%.

Having a system that could detect the ball both indoors and outdoors was deemed extremely important, thus being resistant to sunlight received a weighting of 20%, in line with integration of incorporating a waterproof design with the component, also receiving a 20% weighting factor. Confidence in developing a design that can integrate the hardware into a feasible solution had a small, yet importance in comparing the alternatives and as a result, received a weighting factor of 5%. The criteria and corresponding weighting factors for evaluating the sensor to detect the sport-specific equipment were then tabulated (Table 21).

Table 21: Sport-specific equipment sensor decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Low-cost	30
Resistant to Sunlight	20
Waterproof Design	20
Confidence of Application in this Context	5
Minimised Complexity	25
Total	**100**

3.5.3.7.3. Sport-Specific Equipment Sensor Alternatives Evaluation

Using the criteria and corresponding weighting factors, the decision matrix for evaluating the alternative sensors to use for sensing when sport-specific equipment is positioned on the marker was developed (Figure 41). The decision matrix determined the proximity sensor with integrated ambient light sensor the most suitable component for this application. This hardware performed well across all criteria, such as being resistant to sunlight with minimised complexity. Sketching possible concept solutions using this sensor (Figure F- 4 in Appendix F: F.1), it could be placed behind a section of laser-cut acrylic (Perspex) embedded into the mechanical design. This meant that implementing a water-proof design using this would be very simple. There was a high confidence that this hardware would work as intended, whereas a sensor like the ultrasonic would not be able to be used behind any kind of solid object (including Perspex). The LiDAR scanner was very expensive compared to the other components and using it in this context was questionable. The electrical-mechanical design solutions involving a force sensitive resistor or mechanical switch were ultimately determined to have too much complexity, as well as uncertainty around developing a product that could have a waterproofed prototype if using two platforms.

Weighting / Alternatives	Low-cost	Resistant to Sunlight	Waterproof Design	Confidence of Application in this Context	Minimised Complexity	Total
	0.3	0.2	0.2	0.05	0.25	1
Force Sensitive Resistor	8 / 2.4	10 / 2	5 / 1	5 / 0.25	0 / 0	5.65
LiDAR	0 / 0	10 / 2	10 / 2	5 / 0.25	6 / 1.5	5.75
Infrared Sensor	8 / 2.4	0 / 0	10 / 2	0 / 0	10 / 2.5	6.9
Proximity/Ambient Light Sensor	7 / 2.1	10 / 2	10 / 2	9 / 0.45	10 / 2.5	9.05
Ultrasonic Sensor	5 / 1.5	10 / 2	0 / 0	0 / 0	10 / 2.5	6
Mechanical Switch	10 / 3	10 / 2	5 / 1	5 / 0.25	7 / 1.75	8

Legend

x	Rating factor (0 - 10)
y	Relative importance

Figure 41: Decision matrix for sensor to detect sport-specific equipment such as a game ball

3.5.3.8. Development of a Master and Slave Component

At this point, it was determined that it would be most suitable to separate the system so that it contained the markers and a separate primary device which would do the distance measuring, communication between the markers and user interaction handling and processing. Thus, this primary device could be referred to as the master device, while the markers could then be referred to as slave devices. As the LiDAR scanner was found to be the most suitable distance measurement sensor, the master device could be positioned in the centre of the slave components and perform the distance measurement for setting up the test. Some early conceptualisation for the marker was developed (Figure F- 5 and Figure F- 6 in Appendix F: F.1).

3.5.3.9. Power Source

It was decided that determining a suitable power source would be determined later in the development stage of the project when it was known how much current draw each device would consume and thus the power requirements of the system. Therefore, battery alternatives evaluation and selection are described in Section 3.7.12.

3.5.3.10. User/System Interface & Intra-system Communication

An essential first order specification was that the system needed to be wireless, which was deemed necessary through the competitive market analysis and by the proposal made by Dr. Elliott. It was decided to evaluate the intra-system communication as well as the user/system interface together, since the hardware used to achieve one, may very well be able to achieve the other function. The options for system communication included Bluetooth technology, Wi-Fi, Nordic radio-frequency (RF), general RF and cellular. A detailed analysis was conducted, comparing various specifications of each type of communication, including cost, the typical range, data rate transfer speed, frequency, power consumption, if it can support multiple connections simultaneously, if it has mobile phone or computer compatibility and any important further notes (Table E- 3, presented in Appendix E: E.3).

3.5.3.10.1. System Communication Alternatives Criteria

The criteria deemed most suitable for comparing the system communication alternatives was low-cost, the number of connections possible, phone and computer compatibility, the maximum range achieved, if it could work independent of other hardware and the power consumption of the hardware (Table 22).

As with all other component selection, low-cost was deemed the most criteria with weighting factor of 30%. This aligned with reducing the manufacturing cost with relative importance of 4.55% (Table 16). Additionally, maximised battery lifetime had a high relative importance (4.27%) (Table 16), thus power consumption was established to have a weighting factor of 25%. An increased maximum range of the device would increase the opportunity for increasing the testing zone area, which had a medium importance (2.95%) (Table 16), thus, this criterion received a 20% weighting factor. Working independent of other hardware would mean a minimisation in the number of parts in the system, which had a high relative importance (4.1%) (Table 16), as well as reducing the manufacturing cost. Therefore, this criterion received a weighting factor of 10%. Aligning with this weighting factor, connection with a phone or computer was as important, since it would be very beneficial if the system used to communicate between devices could also connect to the user. The number of connections the device could make was of importance, but compared with the other criteria, not as high, so it was given a 5% weighting factor. The criteria and corresponding weighting factors were then tabulated (Table 22).

Table 22: System communication decision matrix criteria and corresponding weighting factors

Criteria	Weighting (%)
Phone/Computer Connection	10
Low Cost	30
Number of Connections	5
Maximum Range	20
Independent of Other Hardware	10
Power Consumption	25
Total	**100**

3.5.3.10.2. System Communication Alternatives Evaluation

Using the criteria and corresponding weighting factors, the decision matrix for evaluating the communication hardware alternatives was developed (

Figure 42). This matrix was evaluated in a different way, compared with the other evaluation techniques. The reason for this was because the highest scoring alternative was the Nordic RF, which is not compatible with mobile phones or computers. Similarly, the second highest scoring alternative was general RF, which again is not compatible with either mobile phones or computers. The hardware that would be able to support either a mobile phone or computer compatibility (or both) was cellular, Bluetooth and Wi-Fi. Bluetooth received the highest score of these three alternatives, as it had compatibility with both mobile phone and a laptop with inbuilt Bluetooth or a computer with a Bluetooth dongle. It also had a reasonable cost compared with the other user device alternatives. It also worked independent of other hardware, which was something Wi-Fi and cellular both could not achieve, as Wi-Fi needed a router to connect to and cellular needed an antenna and a SIM card. Thus, Bluetooth communication was selected as the communication type for communicating with the mobile phone or computer.

For intra-system communication, the matrix determined the Nordic RF to be best suited for the application. They were found to have an extremely good maximum range, worked independent of other hardware and were deemed very good cost. The number of connections that could be connected was slightly limited, but still was acceptable for the number of markers to be used in the system.

Criteria \ Weighting \ Alternatives	Weighting	Bluetooth	Wi-Fi	Nordic RF	General RF	Cellular
Phone/Computer Connection	0.1	10 / 1	10 / 1	0 / 0	0 / 0	10 / 1
Low Cost	0.3	7 / 2.1	5 / 1.5	9 / 2.7	8 / 2.4	3 / 0.9
Number of Connections	0.05	2 / 0.1	10 / 0.5	6 / 0.3	6 / 0.3	10 / 0.5
Maximum Range	0.2	3 / 0.6	2 / 0.4	10 / 2	8 / 1.6	6 / 1.2
Independent of Other Hardware	0.1	10 / 1	0 / 0	10 / 1	10 / 1	0 / 0
Power Consumption	0.25	4 / 1	7 / 1.75	9 / 2.25	10 / 2.5	5 / 1.25
Totals	1	5.8	5.15	8.25	7.8	4.85

Legend

x	Rating factor (0 - 10)
y	Relative importance

Figure 42: Decision matrix for system communication

3.5.3.11. Initiate Test

In terms of initiating the test, three options were presented; the user self-initiation the test using a button, automatic detection using a sensor or through the user/system communication interface using Bluetooth as determined in the communication alternatives evaluation. Ideally, a system that could automatically detect when to initiate the test would be optimal. However, due to time constraints presented, this would not likely be able to be implemented in this project. It was deemed most suitable that the user could indicate when the test should initiate using a command through Bluetooth. This would be simplest to implement and would work with the hardware already selected.

3.5.3.12. Programming Language

Now that it was known that the Arduino would be used as the microcontroller for this project, it was now known that C/C++ would be the language used since this is the native language for Arduino (Arduino, 2018). This meant C#, Python, Java, JavaScript and Ruby could all be removed as means for this function.

3.5.3.13. Enclosure Material

The enclosure that the electronic components would be contained in needed to be determined. A decision needed to be made whether to purchase a pre-fabricated enclosure or to custom design one. There was benefits and limitations for each option. Purchasing a pre-fabricated enclosure would be time-saving and trustworthy to use as an enclosure; however, it would cost money to buy the enclosures, as well as the time to customise it so that it could house the electronics. Custom designing one would take more time, however it would generally cost less money to do so and give the ability to customise the enclosure so that the electronic hardware could be mounted appropriately. It was therefore decided to design and develop a customised enclosure. It was important to select a material that would achieve the best outcome when compared against a set of design criteria. The enclosure material and method alternatives were polyvinyl chloride (PVC)/polycarbonate (PC) sheet, vacuum moulding, a bucket, PVC pipe, 3D printing, wood, metal or laser-cut acrylic.

3.5.3.13.1. Enclosure Material Alternatives Decision Criteria

The criteria used to evaluate the enclosure material alternatives was low-cost, high-strength, being usable outdoors, minimised construction time and customisable (Table 22). Low-cost was given a weighting factor of 30% to assist in reducing the manufacturing cost. To ensure that the system would be designed and built by the due date (4.4% importance) (Table 16), minimised construction time was deemed essential during the selection process, establishing a 30% weighting factor. One of the primary reasons for deciding to design and develop the enclosure was to provide customisation, therefore this criterion was given a weighting factor of 25%. It was important that the design would be able to be used outdoors (3.84% importance) (Table 16), however this criterion was not deemed as important as the other criteria mentioned, therefore receiving a 10% weighting factor. The weight of the system had medium importance (2.65%) (Table 16), so the weight of the material did have some importance, so it was established to have a weighting factor of 10%. Having a material that would be of high-strength was important enough to compare the alternatives against, as the device would need to be able to have a maximised load capacity (1.54% importance) (Table 16) to hold the sport-specific equipment, as well as be able to withstand impacts from the athlete (1.91% importance) (Table 16); therefore, was given a 5% weighting factor. The criteria and corresponding weighting factors for enclosure material alternative selection is shown in Table 22.

Table 23: Decision matrix criteria and corresponding weighting factors for enclosure material alternatives

Criteria	Weighting (%)
Low-cost	30
High-strength	5
Usable Outdoors	10
Minimised Construction Time	25
Customisable	20
Weight	10
Total	**100**

3.5.3.13.2. Enclosure Material Alternatives Evaluation

The decision matrix for evaluating the enclosure material alternatives (Figure 43) found vacuum moulding to be deemed unsuitable, as it would not be able to be used outdoors as it would not be possible to make it waterproof or sun resistant. Additionally, Flinders University was only capable of producing a thickness of 1 mm, therefore it would have incredibly low physical strength, which would likely not even be able to hold any of the sport-specific balls. A bucket was very cheap and would require minimal construction time but had low strength and was not able to be customised at all, as well as not looking aesthetically pleasing at all. PVC piping would have been a good option; it could be purchased in 75 mm, 90 mm, 150 mm, 225 mm, 300 mm or 375 mm diameters; the most suitable would have been 150 mm or 225 mm. However, this alternative was also not customisable and had specific minimum lengths able to be purchased, which meant the cost was significantly high. Wood and metal did not meet the requirements for minimised construction time and weight. Additionally, metal was not completely usable outdoors, because it would get very hot in direct sunlight; this would be very bad for the hardware as well as create a safety hazard if touched.

Enclosure Alternatives \ Weighting \ Design Criteria	Low-cost	High-strength	Usable Outdoors	Minimised Construction Time	Customisable	Weight	Total
Weighting	0.3	0.05	0.1	0.25	0.2	0.1	1
PVC/PC Sheet	5 / 1.5	8 / 0.4	10 / 1	2 / 0.5	8 / 1.6	8 / 0.8	5.8
Vacuum Moulding	10 / 3	1 / 0.05	3 / 0.3	6 / 1.5	7 / 1.4	10 / 1	7.25
Bucket	9 / 2.7	2 / 0.1	10 / 1	10 / 2.5	0 / 0	7 / 0.7	7
PVC Pipe	2 / 0.6	8 / 0.4	10 / 1	8 / 2	0 / 0	7 / 0.7	4.7
3D Printing	10 / 3	4 / 0.2	1 / 0.1	6 / 1.5	10 / 2	10 / 1	7.8
Wood	6 / 1.8	8 / 0.4	8 / 0.8	2 / 0.5	8 / 1.6	5 / 0.5	5.6
Metal	4 / 1.2	10 / 0.5	4 / 0.4	2 / 0.5	7 / 1.4	3 / 0.3	4.3
Laser-Cut Acrylic	10 / 3	6 / 0.3	7 / 0.7	4 / 1	8 / 1.6	6 / 0.6	7.2

Figure 43: Decision matrix for enclosure material alternatives

Legend

x	Rating factor (0 - 10)
y	Relative importance

3D printing was found to be the best candidate for material selection, which is a process of additive manufacturing. Using 3D printing would mean flexibility in the design, being able to be heavily customised exactly as required. The use of 3D printing would mean that design iterations could easily be conducted to make changes and adjustments as necessary. It was noted that Flinders University had in possession approximately fourteen 3D printers (Ultimaker 2+ and Ultimaker 2+ Extended, Ultimaker, Gelderland, Netherlands), which meant they would be easily accessible. A major benefit in using 3D printing would be that Flinders University allows free access for students to use these printers. It would be essential to ensure that the cost of development remained within budget. Thus, it was concluded that the time spent on designing the CAD models heavily outweighed the cost of developing the enclosures using any of the other design alternatives within the decision matrix (Figure 43). Therefore, the enclosures could essentially be developed at no manufacturing cost using this design method in this project.

3.5.3.14. Sensor Rotation

At this point in the project, it was unknown that sensor rotation would be required; it was not until the LiDAR scanner was selected that this function was found needing to be solved. As a result, evaluating a sensor rotation mechanism and subsequent motor encoding is discussed in Section 3.6.2: Rotating the LiDAR.

3.5.3.15. Shield Sensor from Environment

Two sensors needed to be shielded from the environment (particularly due to rain). Several options were presented to protect the sensors, including acrylic, glass, polycarbonate, a clear CCTV dome or developing a customised 3D print that would protect it. Glass or acrylic could have been used to shield the proximity sensor with integrated ambient light sensor. These two alternatives could be implemented easily with the mechanical design. As Flinders University offered acrylic laser-cutting, this option was selected since it was readily available, and it could be cut to specification. For the LiDAR scanner, thoughts around enclosing it specially with 3D printing was investigated. The author had experience with LiDAR scanner technology through a work placement internship and knew that the scanner could detect objects through glass windows. Window glass has 83-90% transparency (The Gale Group, 1979). Knowing the LiDAR scanner could work with this transparency, evaluation of using a clear CCTV dome could be determined. One company selling one claimed transparency of 97% and low distortion factor of 0.16%. This deemed the part to be appropriate for use with the LiDAR scanner and thus was selected for shielding it against the environment.

3.5.3.16. Anti-slip

Ensuring the components would not slip was important for positioning and maintaining position. Therefore, a rubber mat, adhesive bumper strip and adhesive pad protectors were considered. The adhesive pad protectors were selected as they were very small and could easily be fixed to the mechanical design towards the end of the project when everything had been finalised.

3.5.3.17. Hardware Mounting

To mount the various hardware components, metal brackets, custom designed 3D prints and double-sided tape was considered. Ideally, metal brackets would be best suited. Although not desirable, double sided-tape, Velcro and cable ties were considered options too. Using these were considered acceptable as the system being developed was a proof-of-concept prototype. These alternatives would be used as a very last resort. With the many different hardware components, it was deemed necessary to aim to develop custom designed 3D printed parts for each component.

3.6. Concept Evaluation

So far, the development and analysis of the morphological chart had provided a set of means for accomplishing various function required to achieve the aims and objectives for the project solution. Through careful analysis, non-viable means were able to be eliminated during the selection process. The remaining alternatives were evaluated against a set of criteria, which were weighted alongside a rating factor to establish the most optimal mean of the selection set. It was then required to make further refinements in the design solution, through more in-depth investigation and evaluation. This would assist in finalising development of a suitable concept solution.

3.6.1. LiDAR Scanner Comparison

It was established that a LiDAR scanner would be the most suited sensor to measure distance of the markers to assist in set-up of the system. Among the many LiDAR scanners on the market, suitable candidates that were within the budget range were found; Scanse Sweep, LiDAR Lite v3HP, RPLIDAR A1 and A2 and the TeraRanger Evo 60 m. A detailed analysis of the devices was made, comparing cost, distance range, scan rate, sample rate, distance resolution accuracy, current consumption and if it was usable outdoors, if it was Arduino compatible and if it had integrated system 360^{o} rotation (Table E- 4 in Appendix E: E.4).

3.6.1.1. LiDAR Scanner Alternatives Decision Criteria

Through the analysis made of the various devices, a further refined set of criteria were established to evaluate the alternatives based on the features that were deemed most important; low-cost, maximum distance measurement, scan rate, sample rate, current consumption and if the scanner could automatically rotate 360°. The corresponding weighting factors for these criteria can be observed in Table 24. Low-cost and maximum distance measurable were the criteria that were deemed most important during the selection process (40% and 30% weighting factors, respectively). The amount of current consumption was the next significant criteria with weighting factor of 15%, to help minimise battery consumption. Finally, other features deemed important were sample rate, scan rate and 360° rotation with 5% weighting factors.

Table 24: LiDAR scanner decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Low-Cost	40
Max Distance	30
360° Rotation	5
Scan Rate	5
Sample Rate	5
Current Consumption	15
Total	**100**

3.6.1.2. LiDAR Scanner Alternatives Evaluation

The decision matrix for evaluating the LiDAR scanner alternatives (Figure 44) established the LiDAR Lite v3HP to be the best alternative, with lowest cost, maximal distance measurement and lowest current consumption. Although it did not have integrated 360° rotation, a custom-built mechanism could be developed to provide it with rotation.

In terms of the alternatives, the Scanse Sweep had the worst current consumption with a high cost and a medium sample rate. The RPLIDAR A1 was low-cost but had a low maximal distance and scan rate. Conversely, the RPLIDAR A2 had a high sample rate and a better maximal distance but was very expensive. Both RPLIDAR scanners had high current consumption and low scan rates. The TeraRanger Evo was very low-cost, however it claimed 60 m maximal distance was reduced when in direct sunlight. The sample rate was not specified so this criterion was given a zero.

Criteria	Weighting	Scanse Sweep	LiDAR Lite v3HP	RPLIDAR A1	RPLIDAR A2	TeraRanger Evo 60m
Low-Cost	0.4	3 / 1.2	10 / 4	9.5 / 3.8	1 / 0.4	10 / 4
Max Distance	0.3	10 / 3	10 / 3	3 / 0.9	4 / 1.2	8 / 2.4
360° Rotation	0.05	10 / 0.5	0 / 0	10 / 0.5	10 / 0.5	0 / 0
Scan Rate	0.05	10 / 0.5	6 / 0.3	1 / 0.05	1.5 / 0.075	4 / 0.2
Sample Rate	0.05	5 / 0.25	5 / 0.25	6 / 0.3	10 / 0.5	0 / 0
Current Consumption	0.15	1 / 0.15	10 / 1.5	2 / 0.3	2 / 0.3	7 / 1.05
Totals	1	5.6	9.05	5.85	2.975	6.6

Legend

x	Rating factor (0 - 10)
y	Relative importance

Figure 44: Decision matrix for LiDAR scanner alternatives

3.6.2. Rotating the LiDAR Scanner

With the LiDAR Lite v3HP being the LiDAR scanner of choice, it was found to be important to have a means of rotating the sensor. Various conceptualisation ideas were developed in an attempt to incorporate a 3D printed enclosure surrounding an assembly mechanism which would be capable of rotating the LiDAR scanner (Figure F- 7 and Figure F- 8 in Appendix F: F.1).

The most logical and viable means of achieving the function of rotating the LiDAR scanner was to use a stepper motor, DC motor, servo motor, or by manually rotating the whole device. To set the test up in a way that could become standardised meant that the exact positions of the markers would be needed to be known for assisting the set-up. Thus, manually rotating the whole device was deemed far to imprecise and inaccurate. This left selection of the three different motors. A brief overview of these motors is described in the next sections to understand which motor would be best suited for this application.

3.6.2.1. Stepper Motor

A stepper motor has the capability to rotate at precise angles, as well as rotate at accurately controlled speeds (Muñiz et al., 2008). Simply put, this is achieved as the stepper motor can move in discrete steps through powering a set of inner coils (grouped into phases) in a specialised sequence. The magnetic field generated by the coils causes a magnet attached to the shaft to move. Reversing the sequence causes rotation in the opposite direction. Various types of stepper motors exist in the market (Figure 45).

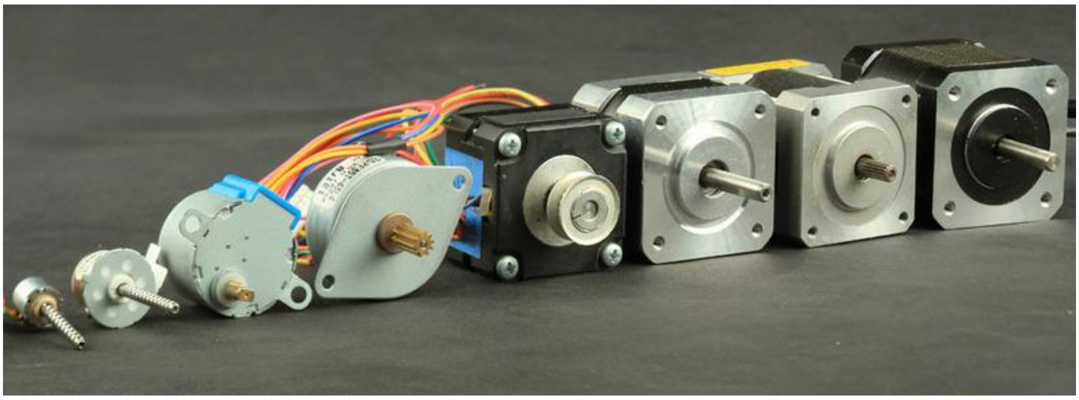


Figure 45: Various stepper motors (Earl, 2015)

3.6.2.2. DC Motor

DC motors can rotate a shaft by converting direct current (DC) power into mechanical energy. The shaft will continue to rotate while DC power is supplied to the motor; thus, they are continuous. They are the most common type of motor and have two forms; brushed and brushless. Brushed DC motors are cheap, easy to run and come in a wide variety of shapes and sizes (Figure 46). Brushless DC motors are generally better, having high efficiency, noiseless operation, higher speed ranges, better speed versus torque characteristics and have a long operating life (Microchip Technology, 2003).

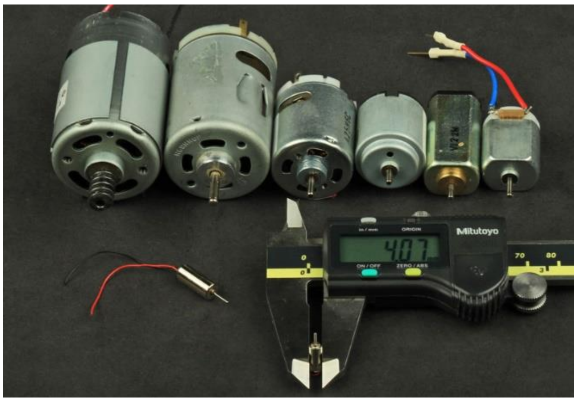


Figure 46: Various types of DC motors (Earl, 2015)

3.6.2.3. Servo Motor

Two variations of the rotary servo exist; a positional servo and a continuous rotation servo. A positional servo has precise control of its angular position using carefully-timed pulses, however it is limited in the amount of rotation it can achieve (Earl, 2015; ISL Products International, 2017). A positional servo is unable to make a complete revolution; generally it has an angular rotation range of motion of 180^{o} (ISL Products International, 2017). Conversely, the continuous rotation servo can complete a full revolution, spinning continuously with control over speed and direction; this type of servo loses its ability to precisely control the position (Earl, 2015). There exist various types types of servo motors (Figure 47).

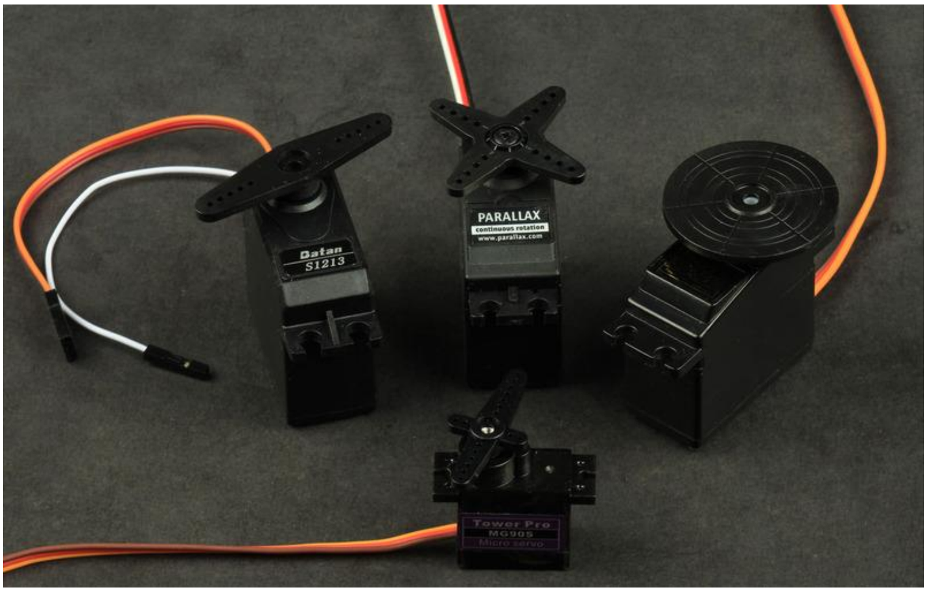


Figure 47: Various types of servo motors (Earl, 2015)

3.6.2.4. Motor Alternatives Decision Criteria

When evaluating the motor alternatives, the decision criteria found most important to compare was if the motor could achieve precise angular and speed control, had low current consumption and was able to have continuous rotation (Table 25). Precise angular positioning was deemed most important, with a weighting factor of 50%, as it would be required to position the markers exactly in the correct position. Continuous rotation was also very important with 30% weighting factor, as it would be intended to put the distance measuring master component in the centre of the slave components and rotate the LiDAR scanner around 360^{o} to correctly measure the positions of the devices. The other two criteria were given a 10% weighting factor as they were still deemed important, but not as important as precise angular positioning or continuous rotation. The criteria and corresponding weighting factors can be seen in Table 25.

Table 25: Motor alternatives decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Precise Angular Positioning	50
Precise Speed Control	10
Low Current Consumption	10
Continuous Rotation	30
Total	**100**

3.6.2.5. Motor Alternatives Evaluation

The decision matrix for evaluating which motor alternative to select (Figure 48) determined the stepper motor to be the most suitable motor to use for set-up of the markers. This was because it was able to achieve high precision angular positioning, as well as precise speed control with continuous 360° rotation.

Criteria	Weighting	Stepper Motor (rating / relative importance)	DC Motor (rating / relative importance)	Servo Motor (rating / relative importance)
Precise Angular Positioning	0.5	10 / 5	3 / 1.5	10 / 5
Precise Speed Control	0.1	10 / 1	7 / 0.7	10 / 1
Low Current Consumption	0.1	2 / 0.2	6 / 0.6	9 / 0.9
Continuous Rotation	0.3	10 / 3	10 / 3	0 / 0
Totals	1	9.2	5.8	6.9

Legend

x	Rating factor (0 - 10)
y	Relative importance

Figure 48: Decision matrix for motor alternatives for rotating the LiDAR scanner

3.6.2.6. Coupling the Motor and LiDAR Scanner

To ensure that wires would not get tangled, a slip ring was investigated to be used in the design, which would maintain wired connections whilst allowing continuous rotation without getting the wires tangled and damaged. The problem with using a slip ring was that the motor could not be directly connected to the LiDAR scanner to rotate it. Thus, it was important to establish a coupling method such that the LiDAR scanner could be rotated using the motor; which would be offset at a distance. Alternatives that were deemed possible means for accomplishing this function was laser-cut acrylic gears, commercial metal gears, a chain drive, drive belt or worm drive.

3.6.2.6.1. Motor-LiDAR Scanner Coupling Decision Criteria

The criteria used in the decision-making process involved it being low-cost, having a high efficiency, being able to maintain a constant velocity ratio, the noise produced, the maintenance required and the amount of slipping that could occur (Table 26). Low-cost was deemed most important, with weighting factor 30%, followed by minimal maintenance required with weighting factor 25%. It was essential for the alternative to make as little noise as possible, so this received a 15% weighting factor. It was not desirable for slipping to occur as well as to maintain a constant velocity ratio, so these were given weighting factors of 15% and 10%, respectively. As the load would be small, the torque required would also remain low, so efficiency was not of a big concern, thus this was given a weighting factor of 5%. The criteria and corresponding weighting factors can be seen in Table 26.

Table 26: Decision matrix criteria for alternatives for coupling motor and LiDAR scanner

Criteria	Weighting (%)
Low-Cost	30
Efficiency	5
Constant Velocity Ratio	10
Minimal Noise Produced	15
Minimal Maintenance Required	25
Minimal Slipping	15
Total	**100**

3.6.2.6.2. Motor-LiDAR Scanner Coupling Alternatives Evaluation

The decision matrix for determining the coupling method to use for the motor and LiDAR scanner (Figure 49) found that using laser-cut acrylic gears was deemed most suitable, as it this could be done in-house at Flinders University without consuming any of the budget delegated. The use of laser-cut gears would mean a constant velocity ratio would be maintained, with no slipping and no maintenance required. Some conceptualisation sketching to incorporate this coupling of the motor and LiDAR scanner was developed (Figure F- 9 and Figure F- 10 in Appendix F: F.1). The limitation in using a geared mechanism would be that it would produce a bit of noise, but this trade-off was justified.

Criteria / Weighting / Alternatives	Weighting	Laser-Cut Acrylic Gears	Commercial Metal Gears	Chain Drive	Drive Belt	Worm Drive
Low-Cost	0.3	10 / 3	5 / 1.5	2 / 0.6	8 / 2.4	2 / 0.6
Efficiency	0.05	6 / 0.3	6 / 0.3	8 / 0.4	10 / 0.5	3 / 0.15
Constant Velocity Ratio	0.1	10 / 1	10 / 1	9 / 0.9	2 / 0.2	10 / 1
Minimal Noise Produced	0.15	2 / 0.3	1 / 0.15	4 / 0.6	10 / 1.5	8 / 1.2
Maintenance Required	0.25	10 / 2.5	2 / 0.5	2 / 0.5	4 / 1	2 / 0.5
Minimal Slipping	0.15	10 / 1.5	10 / 1.5	10 / 1.5	3 / 0.45	2 / 0
Totals	1	8.6	4.95	4.5	6.05	3.45

Legend

x	Rating factor (0 - 10)
y	Relative importance

Figure 49: Decision matrix for alternatives involving coupling the motor and LiDAR scanner

3.6.3. Motor Encoding

Only after having ordered a stepper motor was it found out that it is unable to know its angular position in space. This meant that the motor would be able to control precise angular positioning, however it would not be able to know which way the LiDAR scanner would be pointing. Thus, it was imperative to determine a way of encoding the stepper motor so that there could be a reference point to refer to. Two options were presented to encode the stepper to provide a datum reference point, which could be calibrated to; using a transmissive photo-interrupter or by an absolute optical encoder. It was found the absolute optical encoders were extremely expensive, ranging from $128-$250 (Digi-Key Electronics, 2018). Conversely, a transmissive photo-interrupter could be purchased for a significantly lower price of $2.58 (element14, 2018). As a result, the transmissive photo-interrupter was selected as the method for encoding the stepper motor.

Through some research and investigation, the Vishay TCST2103 (Vishay Intertechnology, Malvern, Pennsylvania, USA) was found to be the perfect component to use as the datasheet specified that it contained a daylight blocking filter. Sketches were created which assisted in ideation and development towards implementation of the Vishay TCST2103 as a motor encoder (Figure F- 11 in Appendix F: F.1).

3.6.4. Finalisation of Electronic Componentry

Through assistance from the Engineering Technical Services team at Flinders University, it was determined that DC-DC step down buck converters would be required to regulate the voltage from the battery for various components. At this stage, it was thought that bi-directional logic level converters would also be required to reduce the logic level input into some components, however it was later found in the embodiment stage during electronic component development and integration that the existing logic level inputs were safe for all the hardware components.

A preliminary functional block diagram was generated for both the master and slave component (Figure 50 and Figure 51, respectively), which aimed to determine the connections between the various electronic components, as well as confirm if they required a logic level converter or DC-DC step down buck converter. The functional block diagram allowed visualisation of key component details and connections to confirm compatibility. It should be noted that after development of the prototype, many errors were determined with these functional block diagrams. After the hardware connection interactions were confirmed, correct functional block diagrams associated with the master and slave component were developed (Figure H- 1 and Figure H- 2 in Appendix H: H.1, respectively).

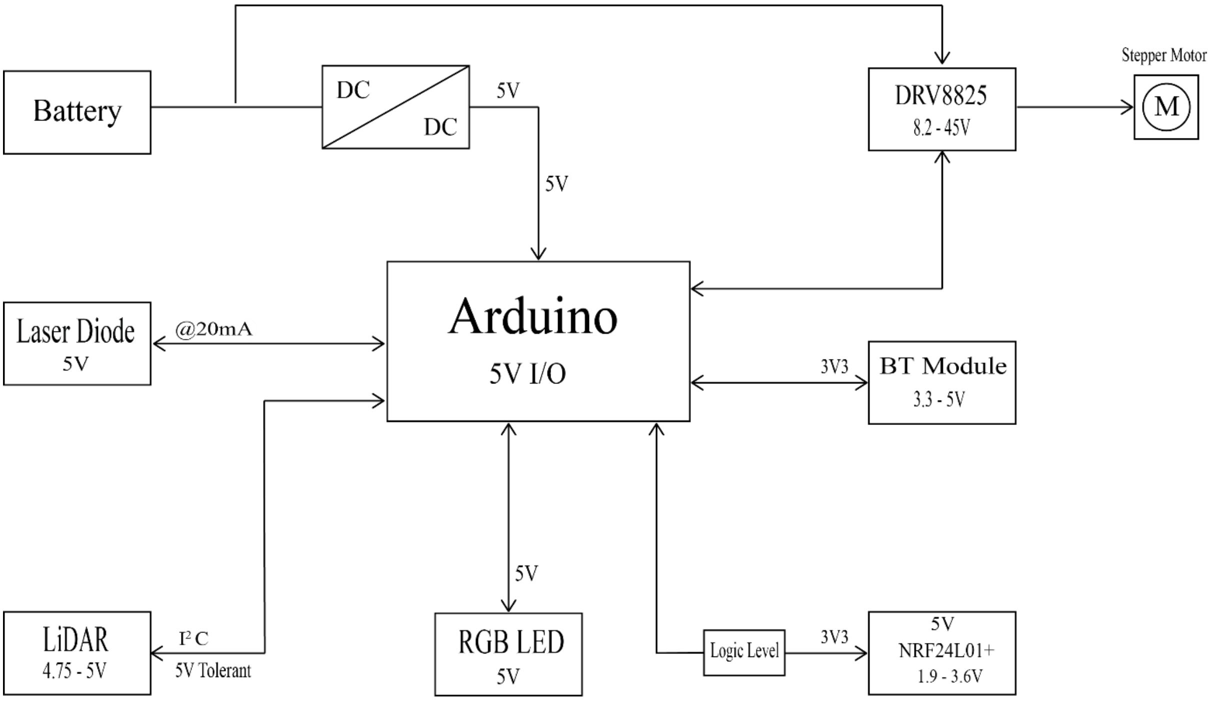


Figure 50: Preliminary functional block diagram for the master component

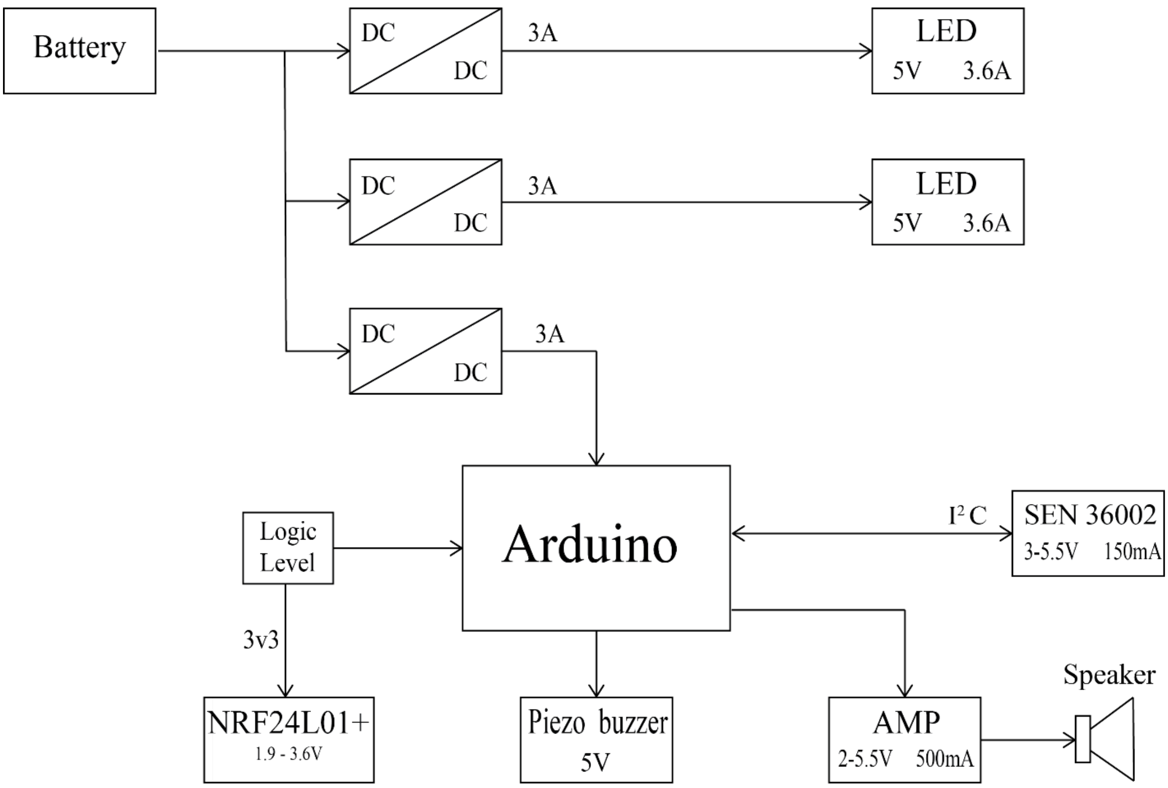


Figure 51: Preliminary functional block diagram for the slave component

3.6.5. Summary of Concept Evaluation

Through investigation, research and analysis, a set of select design alternatives were selected as possible means for meeting specific design functions. By evaluating the means against a set of criteria customised for the function through a decision matrix, the most optimal alternative could be established for integrating into the solution product. By repeating the process for each function, the best performing alternative for each design function was able to be determined. The solution concept stage had therefore formulated a set of mechanical, electrical and software design features, which when integrated, would provide the best possible outcome for the proposed agility testing system. Through further analysis and refinement, a set of suitable components were selected corresponding to each mean. This selection process involved careful selection of the best suited components using further research, investigation and engineering common sense. Therefore, the functions, corresponding means and selected components for the final concept solution were tabulated (Table 27 and Table 28). It should be noted that some of the components selected and presented in this section were deemed either inadequate or unsuitable by progressing through the embodiment stage and integrating the solution together. Justifications and reasoning to why components were exchanged for another is discussed further in the embodiment stage of this thesis in Section 3.7. Additionally, details on the final components selected including the part name, supplier, catalogue number, link, quantity purchased, the unit price, shipping and the subtotal were also tabulated (Table K- 1 in Appendix K: K.1).

Table 27: Most optimally established means for electrical design functions

Function	Mean	Component(s)
Electrical Design		
Single-board Microcontroller	Arduino	Arduino Mega 2560 + Prototype Shield
Measure marker distance	LiDAR Scanner	Garmin LiDAR-Lite v3HP
Rotate sensor	Stepper Motor	Bipolar Stepper, 200 Steps/Rev, 2.8V 1.7A/Phase (Using DRV8825 Driver)
Motor Encoding	Transmissive Photo-interrupter	Vishay TCST2103
Resist tangled wires	Slip-ring	6-wire Slip Ring with Flange (22mm diameter)
Display visual stimulus	LED Matrix	CJMCU-64 RGB LED 8x8 Matrix
Feedback mechanism	LED Matrix, Piezo Buzzer and Laser Diode	CJMCU-64 RGB LED 8x8 Matrix, Piezo Speaker 2.048kHz, Red Laser Diode Module, Single RGB LED Module
Sense ball	Proximity/ Ambient Light Sensor	Si1145 UV / Ambient Light / Proximity sensor (SEN-36002)
Character Display	Letters, Numbers and Symbols	Letters, Numbers and Symbols
Auditory Stimulus	Small Speaker	Adafruit 3” Speaker (4Ω, 3W) amplified using Mono 2.5W Class D Audio Amplifier (PAM8302)
User/System interface/ input	Bluetooth	Sunfounder Bluetooth 4.0 HM-10 Master Slave Module, Button Switch module
Intra-system communication	Nordic RF	nRF24L01+ Transceiver with Socket Adaptor
Initiate test	Through the User/System Interface	Using command via Bluetooth Low Energy (BLE)
Regulate Voltage	Step-down Buck Converter	LM2596 DC-DC Step-down Adjustable Power Supply Module

Table 28: Most optimally established means for mechanical and software design functions

Function	Means	Components
Mechanical Design		
Enclosure	Manually Built	Manually Built
Enclosure Material	3D Printing	3D Printing
Sensor Rotation	Laser-cut Acrylic Gears	Laser-cut Acrylic Gears
Shield Sensor from Environment	Laser-cut Acrylic and Clear CCTV Dome	Transparent 4” CCTV Acrylic Clear Camera Dome
Anti-slip	Adhesive Pad Protector	Self-Adhesive Black Anti Slip Silicone Bumper Pad Shock Absorbers
Hardware Mounting	3D-Printing	Primarily 3D-Printing
Software Design		
Programming Language	C/C++	C/C++ (written in the Arduino software program and Microsoft Visual Studio)

3.7. Design Embodiment

The term embodiment design was first introduced in literature by French, which describes the stage of a design process where the layout design is established (component configurations) and form design (design shape and individual component materials) (French, 1971; Pahl et al., 2007). Through conceptualisation, generation of alternatives and evaluation of these alternatives for determination of suitable design means and components, the solution concept could now be used to produce a definitive layout and form of the proposed solution for design embodiment.

Using concept sketching (Figure F- 1 to Figure F- 11 in Appendix F: F.1) as a guide, the master and slave device designs were modelled using three-dimensional CAD software (Autodesk Inventor Professional 2018, Autodesk, San Rafael, California, USA) to visualise the solution and to ensure that integration of parts would be successful. With this project entailing the design and development of a proof-of-concept prototype that had not been developed before, it was deemed most suitable for designing and manufacturing custom components using 3D printing opposed to purchasing commercially available parts, which could be further investigated in future work. The major benefits for using 3D printing was established in Section 3.5.3.13.2. To summarise the findings from this section; 3D printing would provide means of rapid prototyping custom shaped designs as well as testing integration of component assemblies. At the time of working on this thesis, Flinders University had many 3D printers available as well as supplying complimentary filament for students to print, which would reduce budget significantly compared with other manufacturing processes.

Prototype development followed CAD modelling, which entailed assembling the mechanical design, testing electronic componentry by wiring together components and writing preliminary software code for individual components, integrating electronic components together and finally integrating the mechanical, electrical and software design layouts together. Thus, this section aimed to provide a comprehensive description and explanation for the processes in developing a final design embodiment from the initial solution concept.

3.7.1. Computer-Aided Design Assemblies

A completed CAD assembly was developed for both master and slave components of the system to visualise the interaction between the various components. Using datasheets, the hardware components that were not going to be soldered onto the Arduino prototype board were also CAD modelled so that integration between the customised 3D printed parts could be determined for better alignment and for mounting.

It should be noted that the final CAD assemblies presented in this section, were a result of careful design optimisation and refinement through multiple iterations of various stages of the design process. By implementing customised 3D printed parts, it meant that the components could be adjusted and optimised when tested for integration with the current design at the time. A lot of adjustments were made when the electronic hardware arrived, which were reverse engineered into CAD software using Vernier callipers and a ruler. Thus, many of the components seen in the following rendered CAD assembly images were a result of optimisation achieved later in the design embodiment stage, however, to maintain flow of this thesis, the final assemblies have been shown here.

Another important thing to note was that due to significant time constraints, the CAD assembly had to be developed whilst many of the electronic components were still in transit due to long shipping or lead times. It was imperative to begin 3D printing as soon as possible, since it was known that six slave components and one master component needed to be developed. As a result, the slave enclosure was developed to be as large as possible within the engineering dimensional specification constraint. This was done so that there was enough room to house all electronics as a design that would be too small, which could not house all electronics would be detrimental to the project. Conversely, a design that would be slightly larger would leave room for design refinement and optimisation for future work.

Another major drawback to this situation was that many electronic component dimensions were unknown as datasheets were either not supplied or were not descriptive enough to generate a CAD model replica, such as the speaker or the LiDAR-Lite. Additionally, as it was decided to evaluate battery alternatives closer towards the end of the project, the battery size and shape was also unknown. As a result, no mount was able to be integrated into the slave enclosure design for either the speaker or battery. Velcro mounts were included in the master enclosure, which was developed later. However, due to time constraints, mounts were not able to be developed or printed to attach to the slave enclosure, thus development of such mounts was deemed necessary for future work.

3.7.1.1. Master Component CAD Assembly

Various views of the master component CAD assembly are shown in this section (Figure 52 - Figure 58). A labelled isometric view has been shown to provide a perspective seen from the outside when viewed in real-life (Figure 52). The enclosure was made from three components; the base, lid and wall. The lid and base of the enclosure had been developed to incorporate a snap-fit feature to avoid use of tools when changing or recharging the battery. Additionally, the enclosure was designed this way so simpler mounting of inner components could be achieved as well as for replacing parts if required. In the image, some other outer components can be seen, such as the clear CCTV 4” dome on the top, which was mounted to the enclosure lid, a latching ON/OFF button for isolating the battery when the system would be desired to be off, a momentary button module and small laser cut-out acrylic to position over a small hole in the enclosure wall, such that the RGB LED module can be seen from the outside. A similiarly angled isometric view has been shown with hidden visibility of the enclosure lid, walls and 4” CCTV dome (Figure 53), so the inner componentry can be inspected.

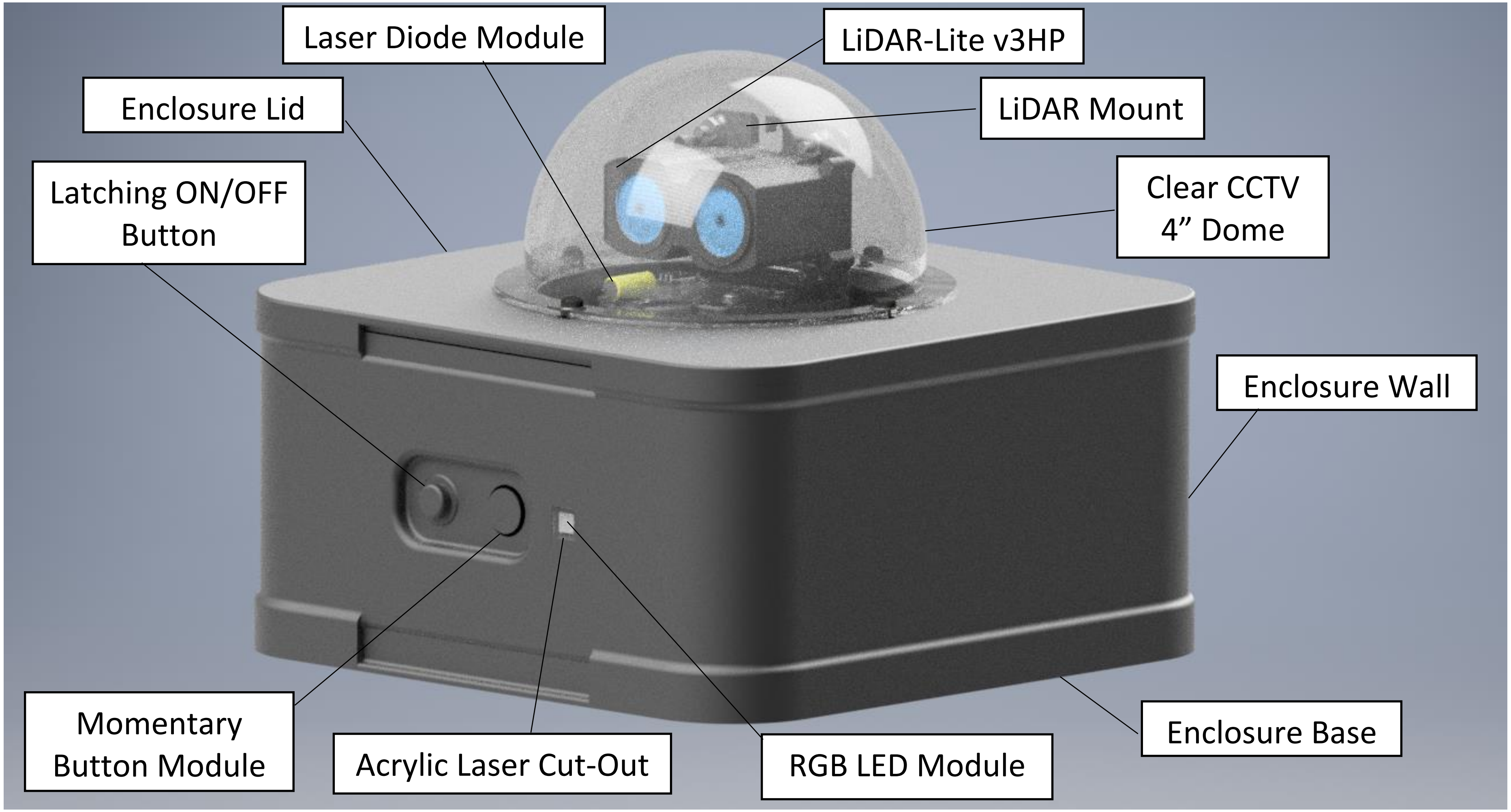


Figure 52: Isometric view of the master component CAD assembly

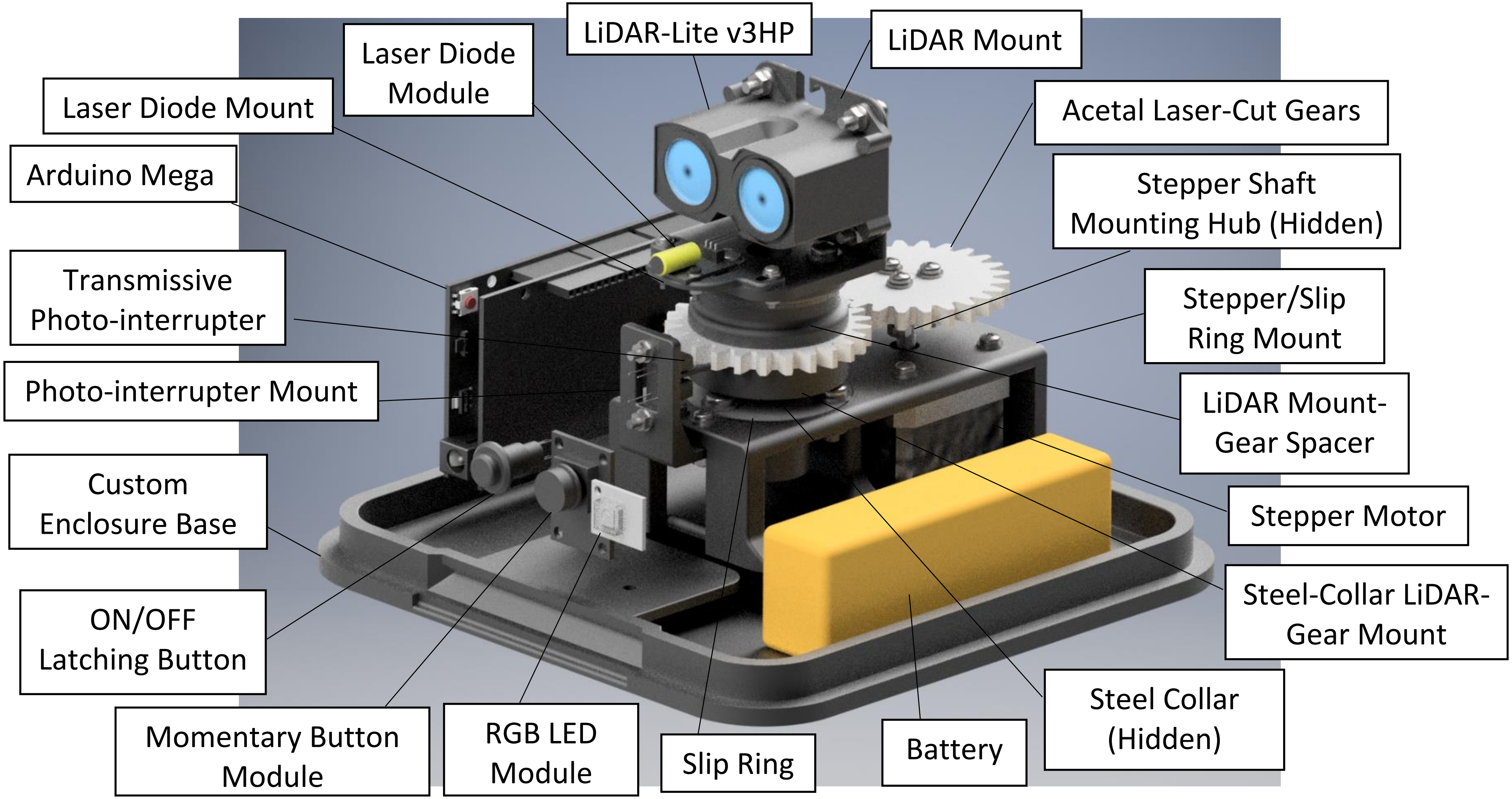


Figure 53: Isometric view of the master component CAD assembly with removed walls, lid and dome covering

The master component inner componentry is quite complex in nature, primarily for the rotation of the LiDAR scanner. This is achieved by coupling the stepper motor's angular rotation to the LiDAR scanner assembly through the set of laser-cut acetal spur gears. One gear is mounted to the shaft of the stepper motor, by fastening it to a commercial Pololu shaft hub mount. The coupled gear is mounted between a spacer on top and a mount below it, which has a circular cut-out beneath it to slide onto a steel collar that is tightened onto the top section of the slip ring using an Allen key. The LiDAR scanner is then mounted on the top of this assembly, with three long fasteners screwed through the four components (in descending order from the top: LiDAR scanner mount, spacer, acetal gear and steel collar mount). Mounted to the LiDAR scanner mount is a laser diode, which would be used to assist the user when setting up the markers. Within this gear-LiDAR scanner assembly, a small protrusion has been designed such that it would pass through the slit of the transmissive photo-interrupter, such that a reference origin position would be known.

Another important feature to note in this image is the small slit at the top of the LiDAR scanner mount. This was incorporated as originally the cord from the LiDAR scanner interfered with the 4” dome when closing the lid. Thus, the slit allowed the cord to remain restrained within the mount so that it avoided contact with the dome.

A top view of the inner componentry of the master device (Figure 54) further assists visualisation of the assembly and where components are relative to one another. A small protrusion with a slit is shown on the right so that Velcro could be used to fix the battery in place.

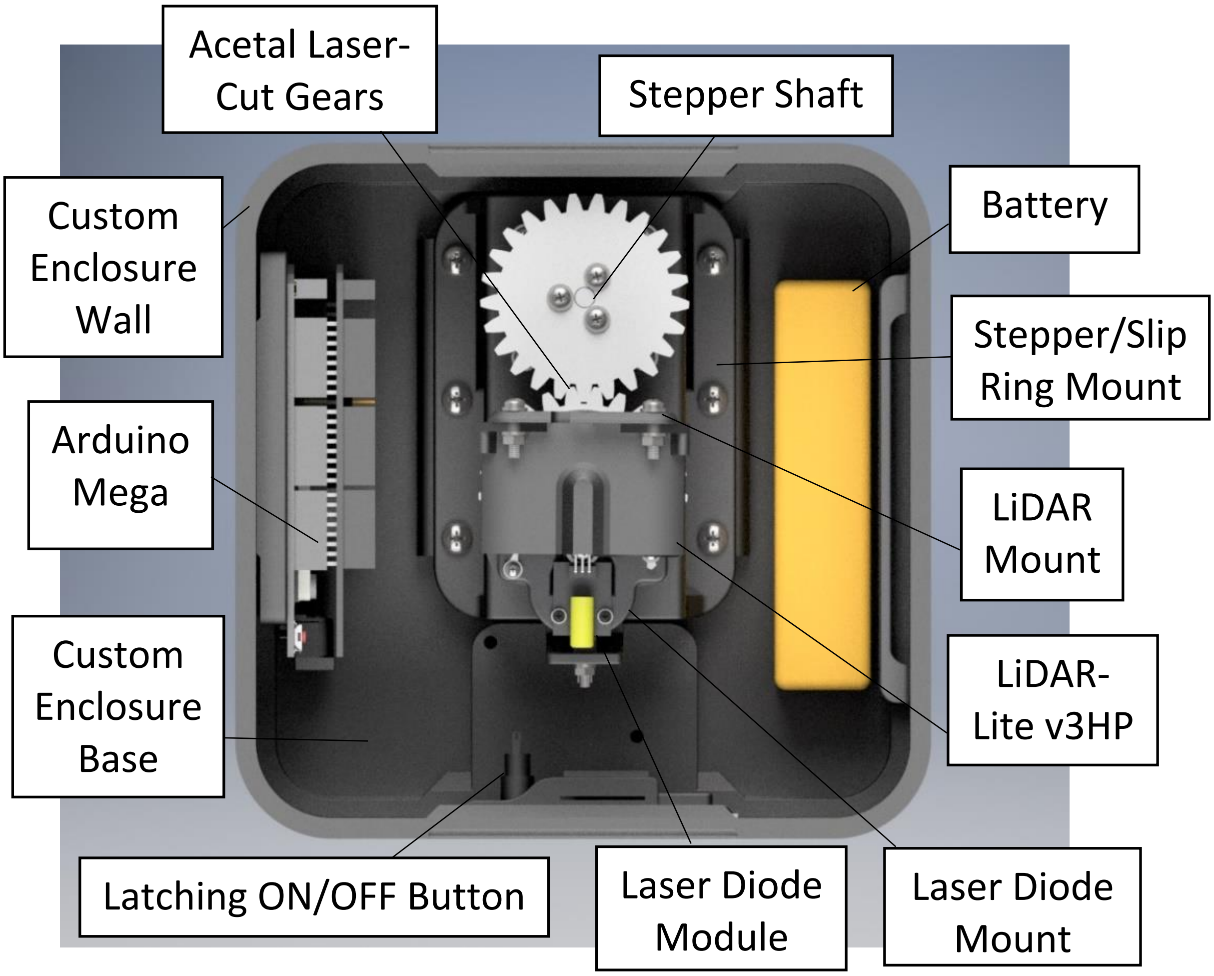


Figure 54: Top view of the master component CAD assembly

Figure 55 shows a side view of the inner componentry of the master device, with Figure 56 showing the same view but with a cross-section of all components so even the enclosure and dome can be observed. These images further assist in visualising the complex integration of parts of the master component.

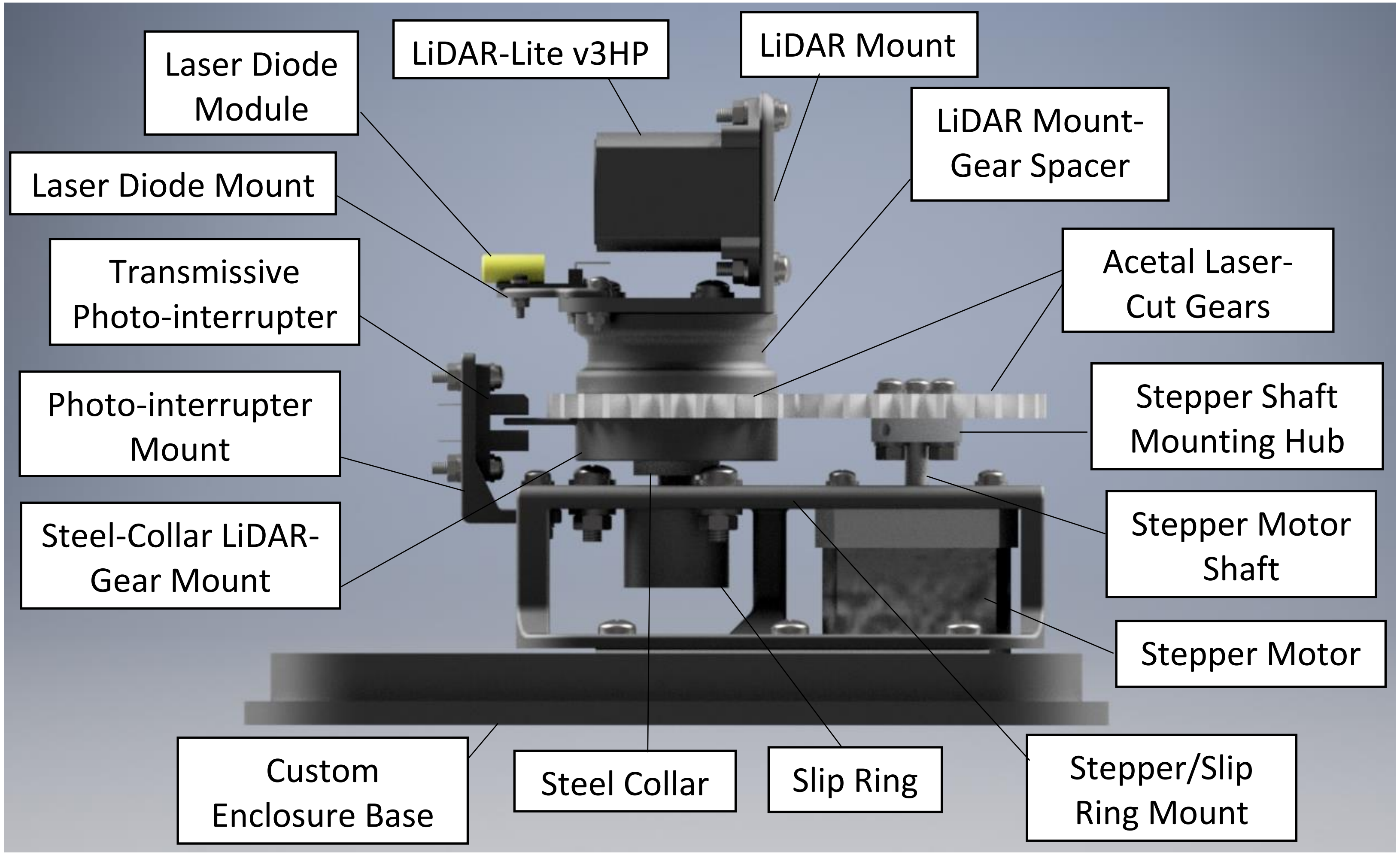


Figure 55: Side view of the master component CAD assembly

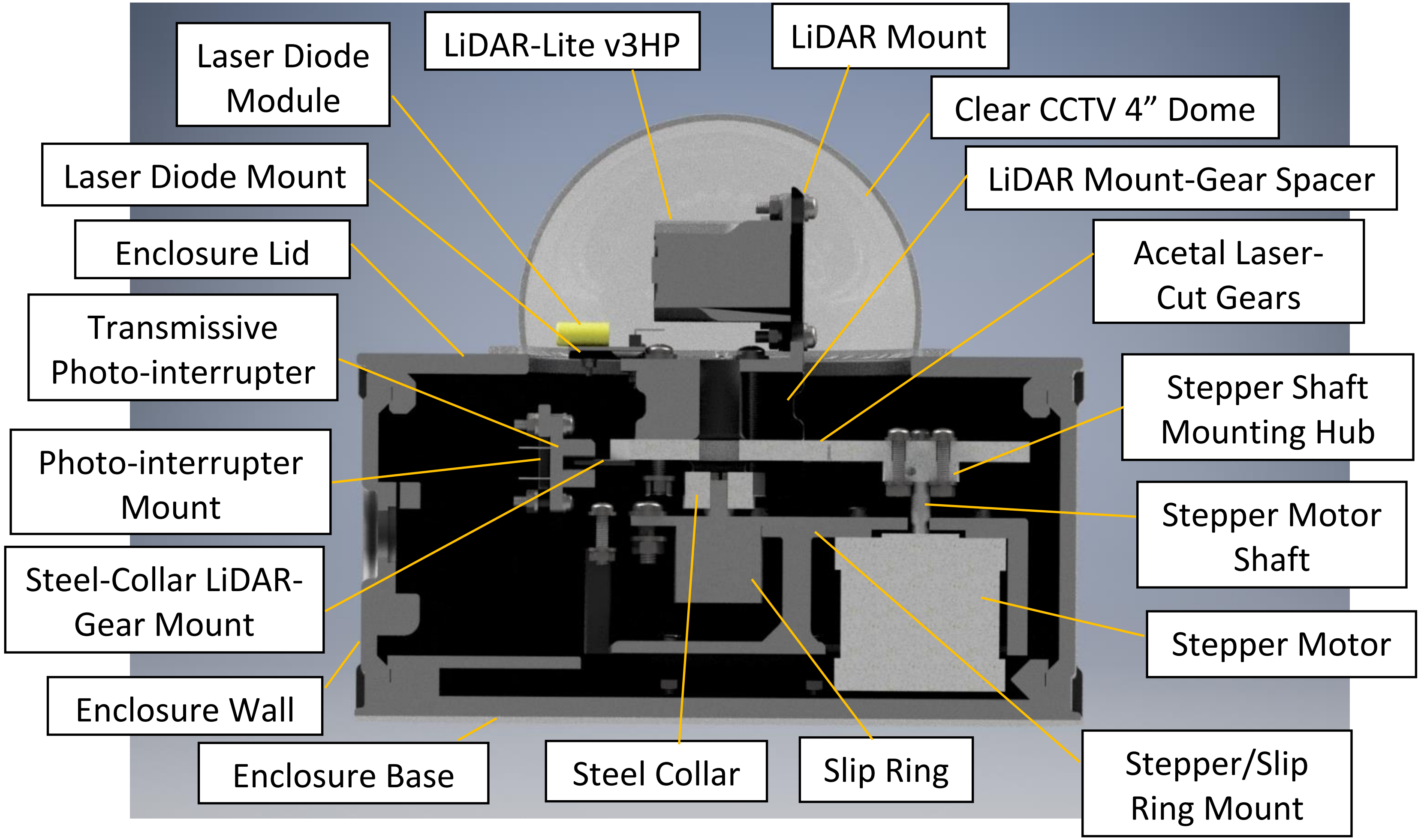


Figure 56: Cross-sectional side view of the master component CAD assembly

A close-up view of the interaction between the steel-collar LiDAR-gear mount protrusion and the transmissive photo-interrupter (Figure 57) shows how the LiDAR scanner rotates through the coupled gearing mechanism. The steel-collar LiDAR-gear mount contains a protrusion which moves with the direction of the LiDAR scanner and thus the direction of the LiDAR scanner can be given a reference point. The stepper motor can control angular positioning precisely through steps, thus the calibration ensures the system is consistent each time it is used.

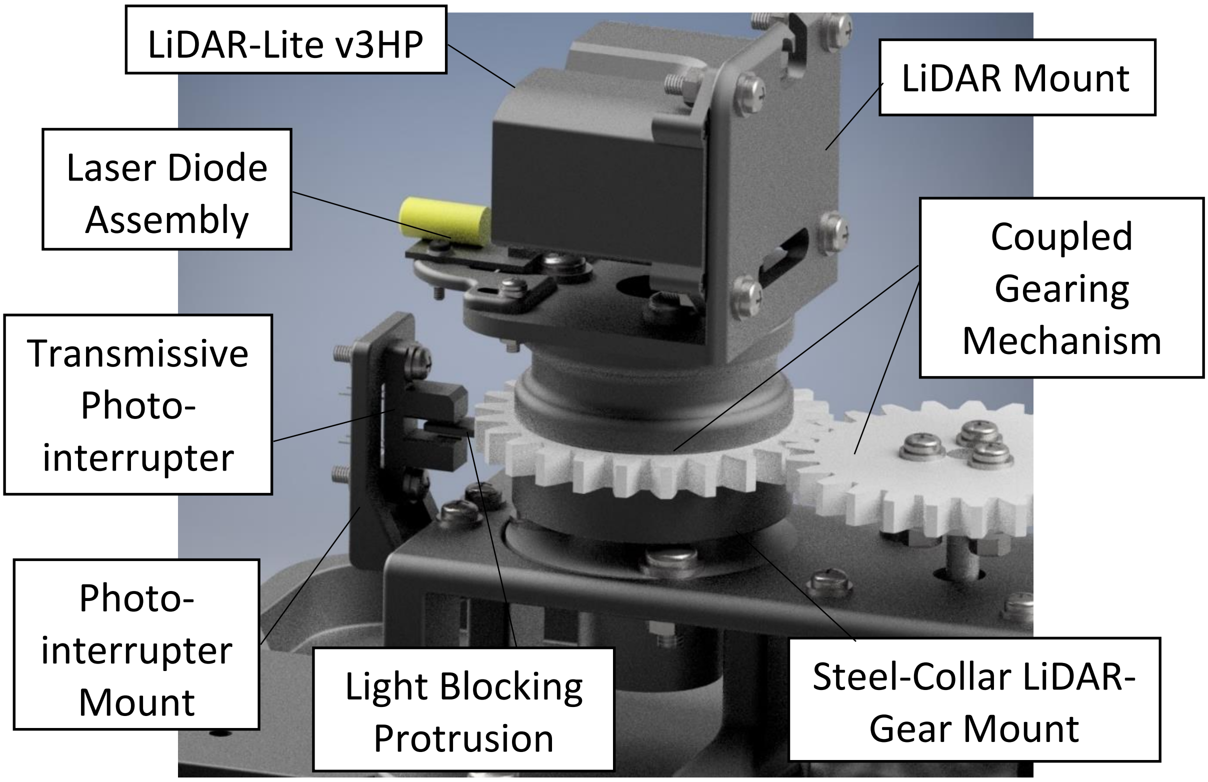


Figure 57: Close-up view of the photo-interrupter used as a reference point for the LiDAR scanner angular position

A cross-sectional view of the enclosure lid and wall for the master component final CAD assembly (Figure 58) shows the enclosure has been designed to employ a snap-fit design so that no tools are required for recharging or changing the battery. The small indentation on the left is so an individual can get a grip on the lid using their fingers to pull it off.

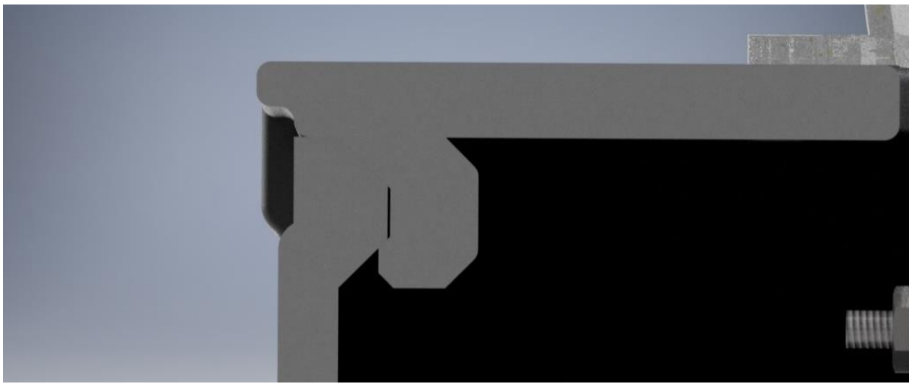


Figure 58: Cross-sectional close-up view of the master component snap-fit design

3.7.1.2. Slave Component CAD Assembly

The slave component is responsible for presenting the athlete with a visual and audio stimulus. Various views of the slave component are presented in this section (Figure 59 - Figure 74). An isometric view of the outer componentry of the slave device (Figure 59) shows that it contains two LED matrix displays on the outside, which are positioned at 120^{o}. This provides the athlete with a larger viewing angle to see the visual stimulus with, since the possibility of seeing it from far angles is possible. One notable feature of the slave component is the two protrusions on the top. These have been specially designed such that sport-specific equipment, including Australian rules football, netball, basketball and rugby can be positioned on top of it, whilst also allowing the user to position their hand underneath it for quickly picking the ball up from the marker. The ball is sensed via the IR proximity sensor with integrated ALS. This is shielded behind a laser-cut square of clear acrylic, which is fixated into the enclosure.

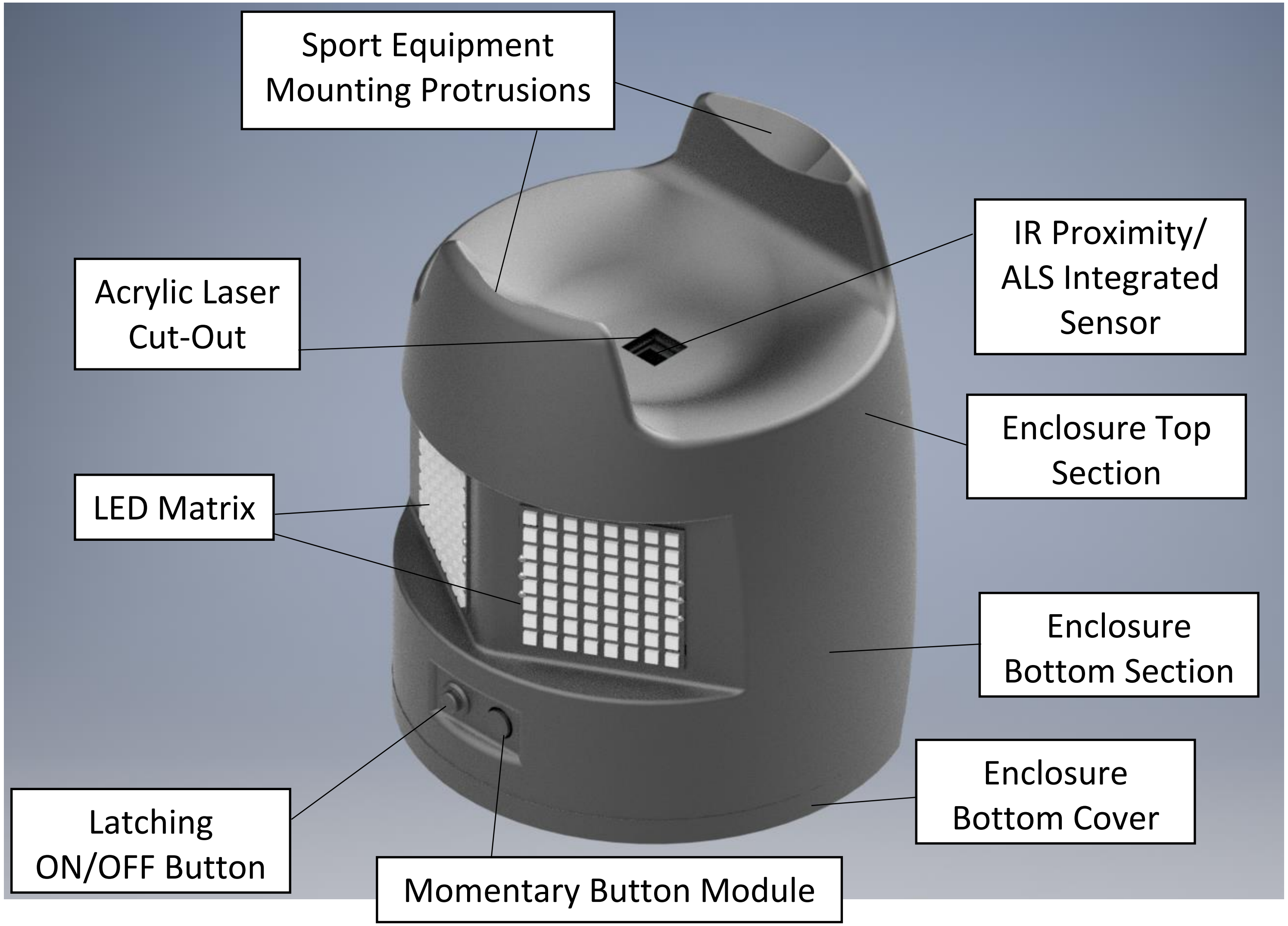


Figure 59: Isometric view of the slave component CAD assembly

A front view of the slave component (Figure 60) shows that both LED matrix displays are easily visible. The latching ON/OFF button can also be seen with the momentary button module too.

Figure 60: Front view of the slave component CAD assembly

Similarly, to the master component, the slave component is also composed of three major components for the enclosure; it contains a top half section, the primary bottom section and a base. The slave component was designed this way for three reasons; the Ultimaker 2+ 3D printer did not have a build height that was tall enough for the design; simpler mounting of electronic componentry with more openness; and significantly decreased print times. Printing as one component would have taken over four days, while printing in two parts reduced print time to approximately less than two days (0.15mm layer height and 10% infill), likely due to minimising the need for supports.

Figure 61 shows a labelled back and front view (Figure 61a and Figure 61b, respectively) of the inner componentry of the primary bottom section. In these images, the DC-DC step down converter can be seen, which will regulate the voltage to usable values for the hardware. The Arduino and prototype shield are mounted at the back, with the two buttons at the front. There is an empty section which was originally for another DC-DC step down converter (Figure 61a), so that there would be one for each LED matrix since they have a current limit of 3A. However, upon testing the matrix displays, it was determined that they could be chained together and still use less than 3A. This primary bottom half of the enclosure had snap-fit protrusions designed so that they could connect with an opposing counterpart on the top enclosure half.

Originally, there was an idea to create some small openings or holes at the front of the design to allow for speaker sound to pass through the enclosure housing. However, to try to minimise ingression of water or dust into the enclosure, this design idea was disregarded. It was also established that high volume speakers were being purchased, which should allow passage of sound through the enclosure medium.

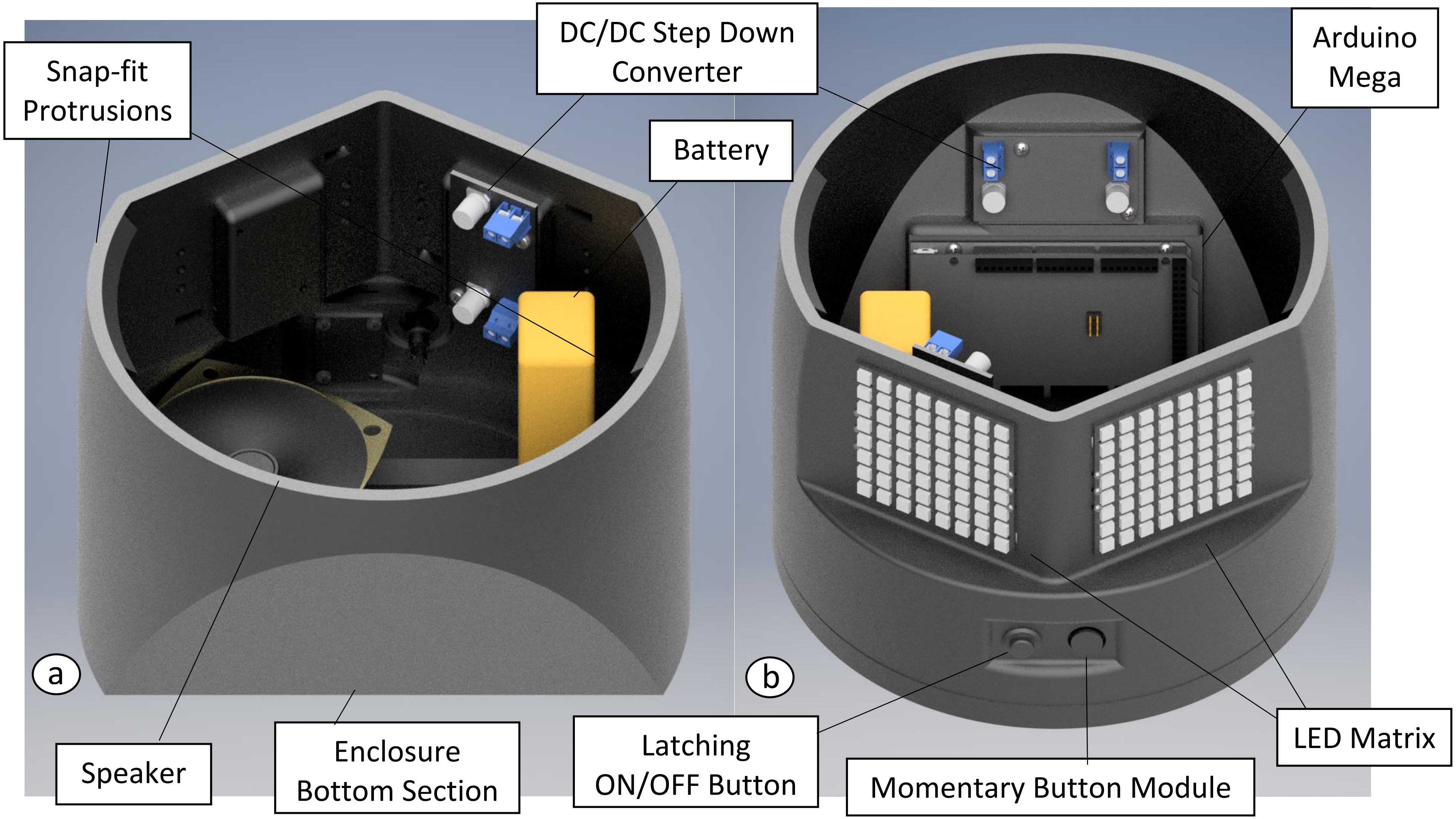


Figure 61: Close-up views of the inner componentry of the slave component bottom section

The top half section of the slave component (Figure 62) shows the snap-fit counterparts that house the snap-fit protrusions from the primary bottom section. In this image, the mounting method for the proximity/ALS is shown, with fastening to the top half section of the slave component.

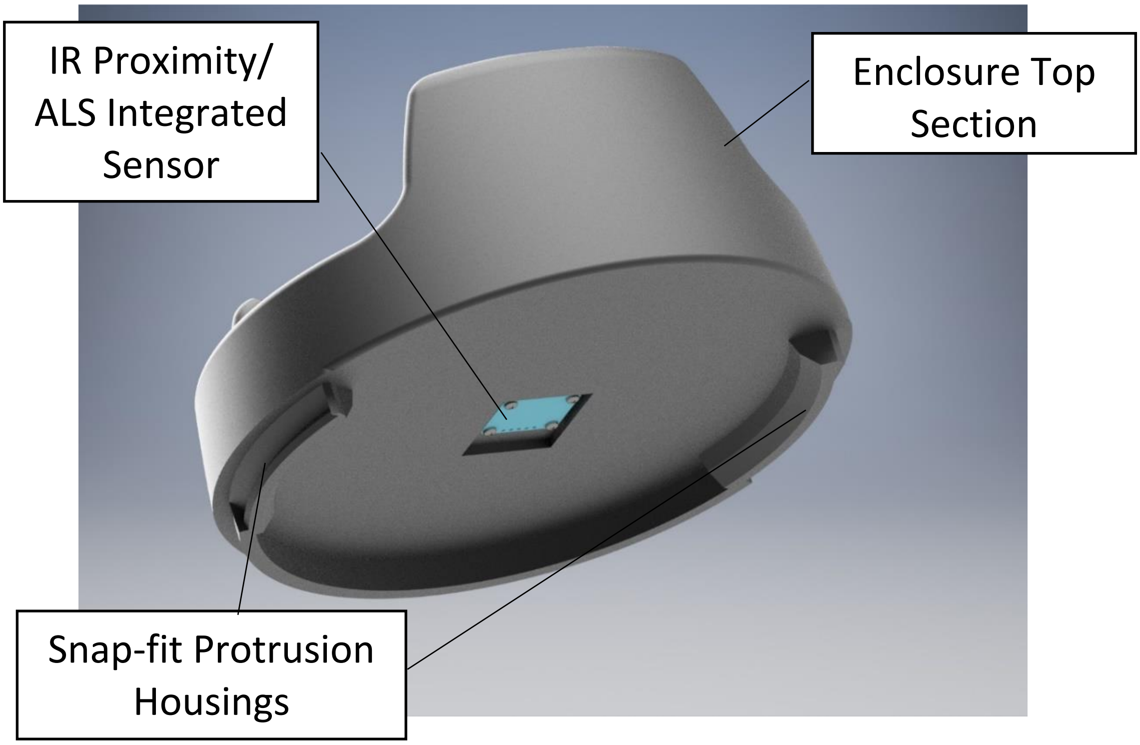


Figure 62: Top half section of the slave component enclosure with sensor to detect a ball

A cross sectional view of the top half of the slave component (Figure 63) shows the snap-fit assembly mechanism interaction, as well as the hole for the proximity/ALS to sense through.

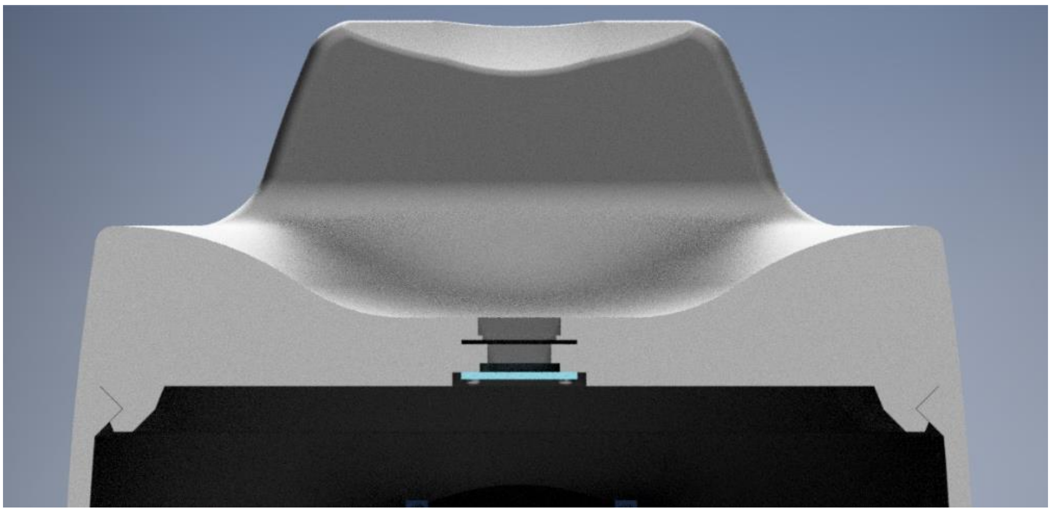


Figure 63: Cross-sectional close-up front view showing the snap-fit assembly and ball sensing slit

The base of the slave component (Figure 64) shows where the screws connect to the primary bottom half, with holes that are indented so that the surface can remain flat. Additionally, a snap-fit mechanism was developed, where the author received inspiration from a simple TV remote (Figure 65), which employed a snap-fit cover so that batteries could be replaced easily with no screws. The snap-fit developed for the slave component base used the TV remote as a guide for development, but it was customised to suit the design of the slave component.

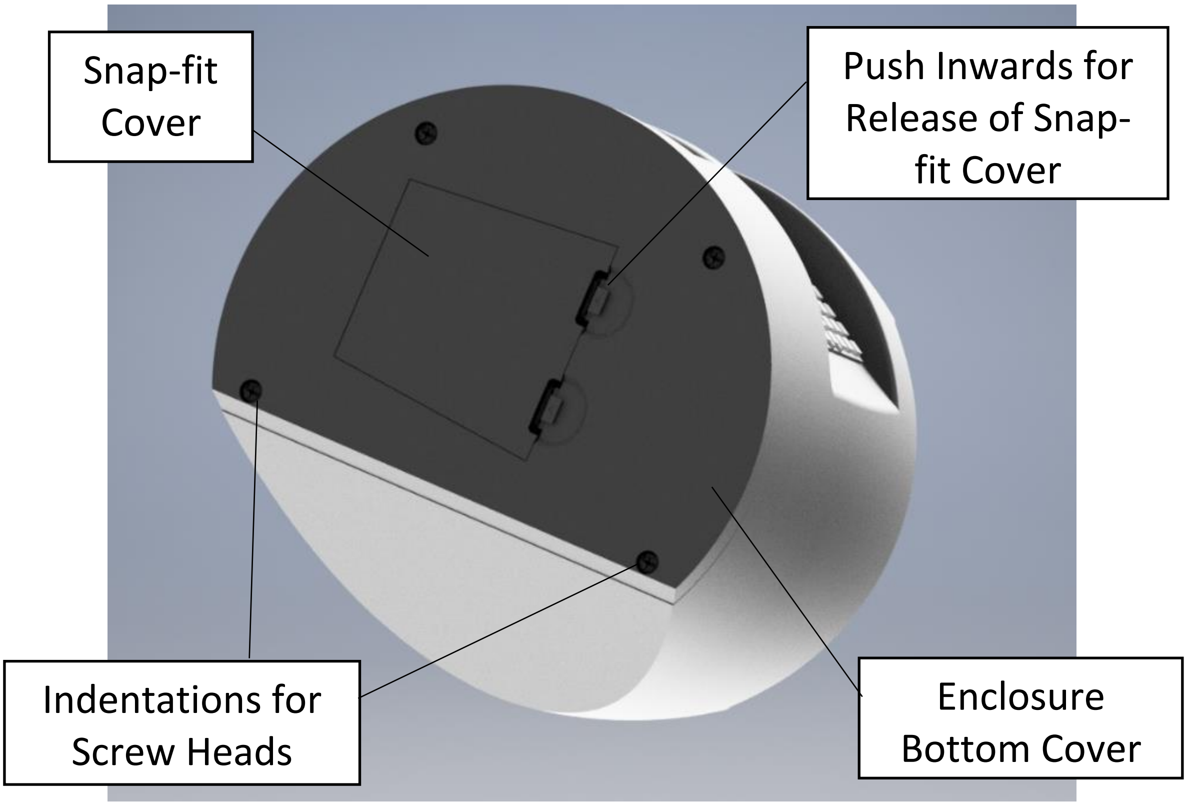


Figure 64: Isometric bottom view of the slave component CAD assembly showing the snap-fit design

Figure 65: Inspiration from a simple TV remote for developing a snap-fit case

A cross sectional view of the snap-fit assembly for the slave component base (Figure 66) shows the interaction between the cover and the base. The simple snap-fit mechanism was employed so that no tools would be required to be used to change/recharge the battery.

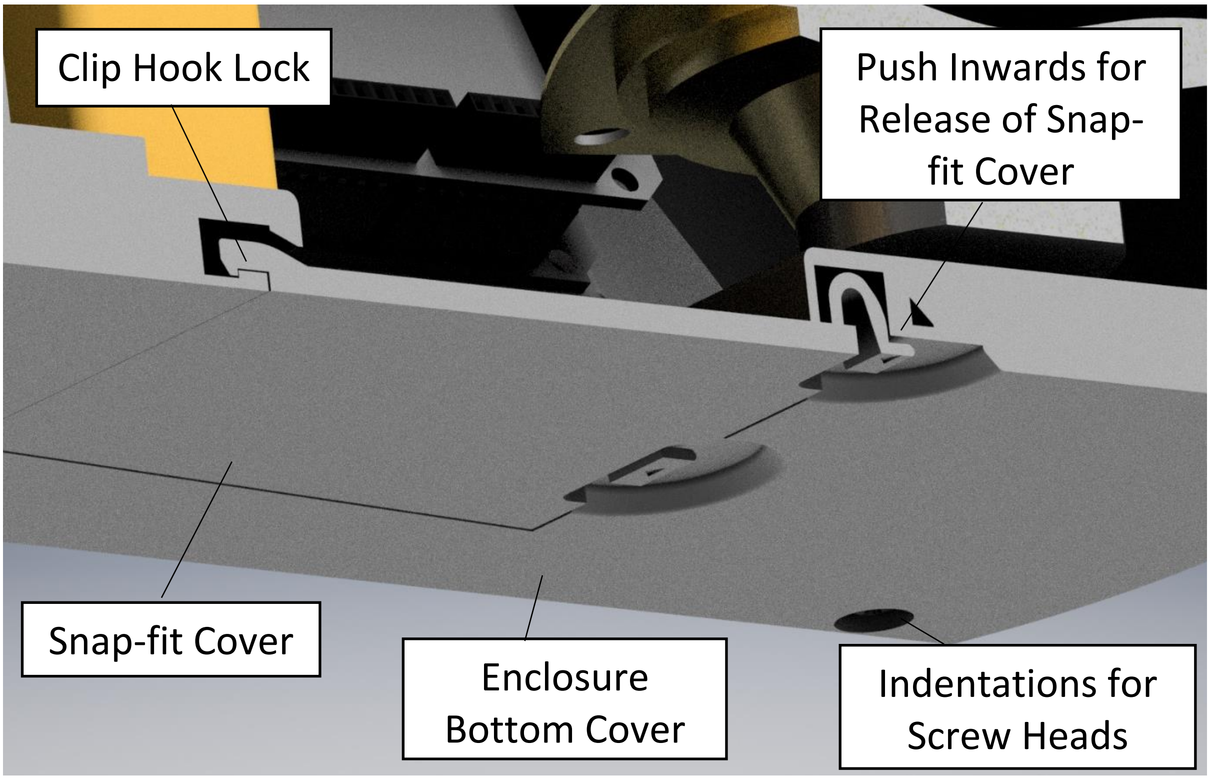


Figure 66: Close-up cross-sectional view of the snap-fit design

A cross-sectional view taken at the centre, dividing the front and back sections of the slave component are shown (Figure 67). The inner front section (Figure 67a) shows positioning of the IR proximity and ALS integrated sensor, one of the DC/DC step down converters, latching ON/OFF button and momentary button module and LED matrix mounting locations. The inner back section (Figure 67b) shows positioning of the Arduino Mega and one of the DC/DC step down converters. Both images show positioning of the battery, snap-fit cover and snap-fit locking assembly. These images have been included to further visualise the inner componentry layout. As already previously discussed, no mount was developed for the speaker or battery due to time constraints.

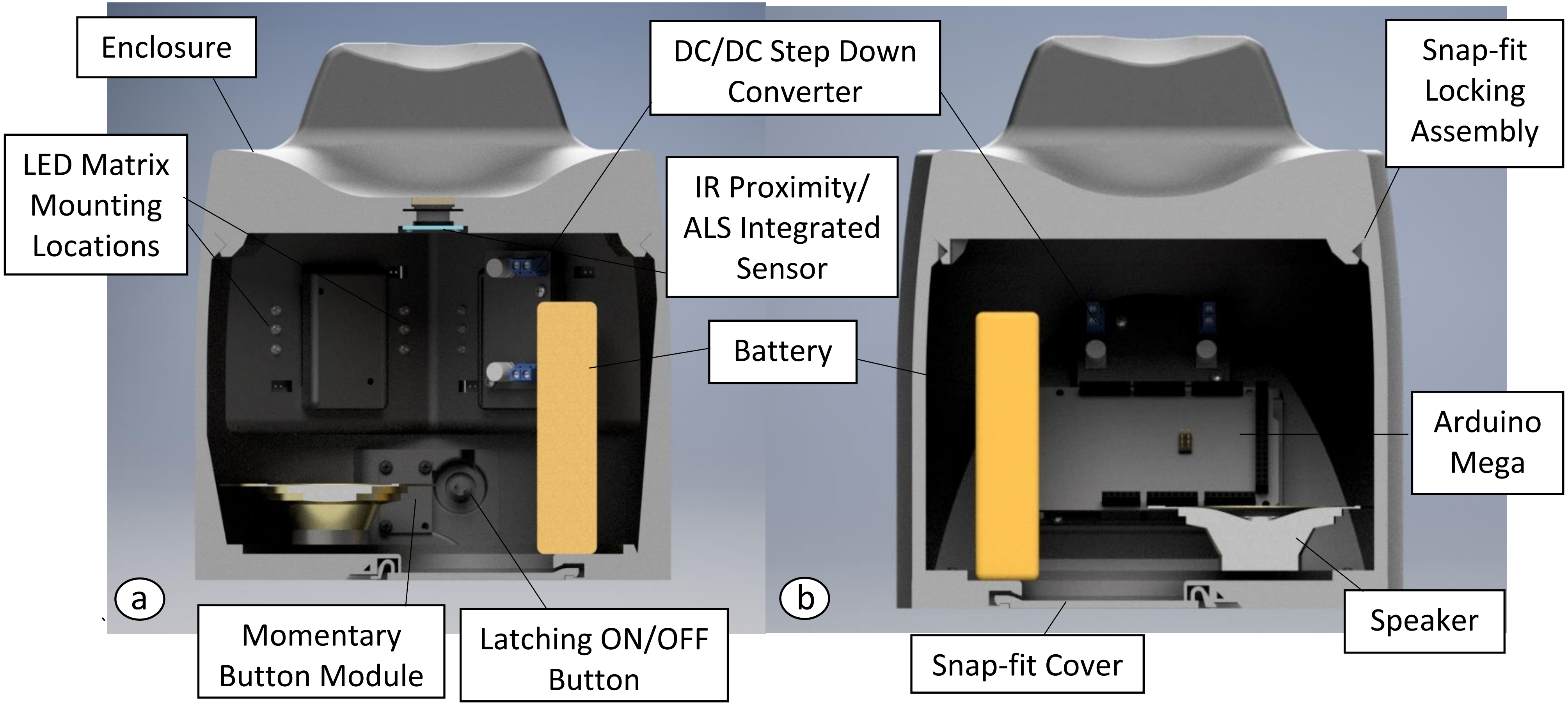


Figure 67: Cross-sectional views of the slave component CAD assembly showing the inner front section (a) and back section (b)

A side view of the marker (Figure 68) presents the slanted flat backing. The reason for this was found during the competitive market analysis, which found that majority of the competing products were able to perform pre-planned agility tests as well as reactive ones. As a result, it was decided that the marker could be positioned with the flat part on the ground (Figure 69), such that the proximity/ALS sensor is facing slightly to the side. Two markers could be positioned facing one another so that a timing gate is made (Figure 69). Therefore, when positioned in this way, the athlete could run straight through them and be detected.

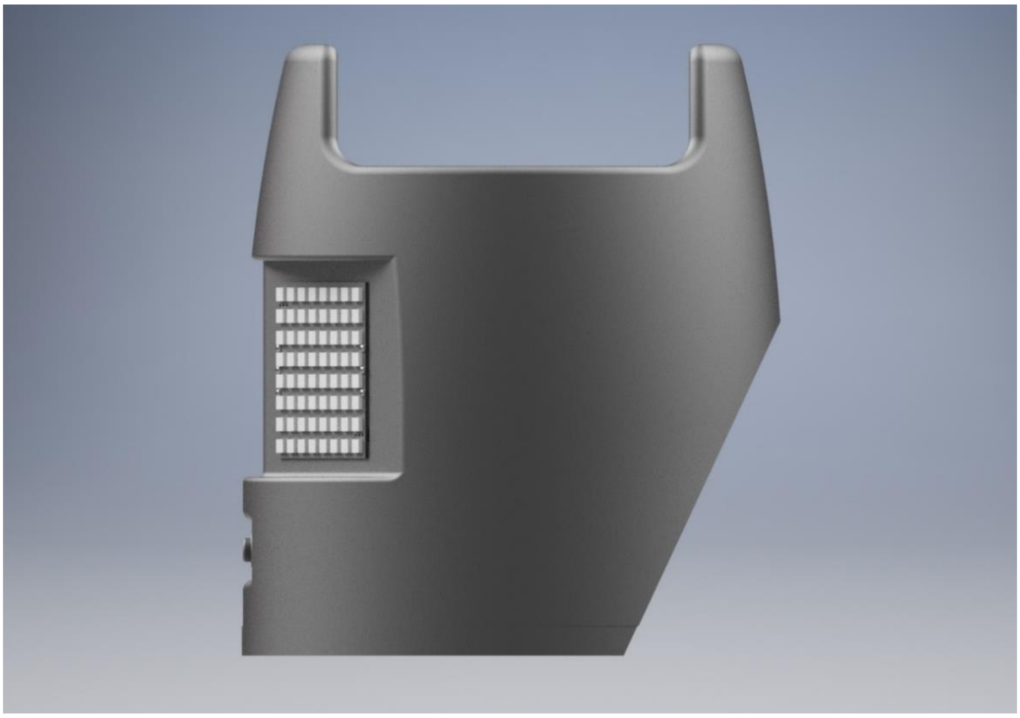


Figure 68: Side view of the slave component CAD assembly

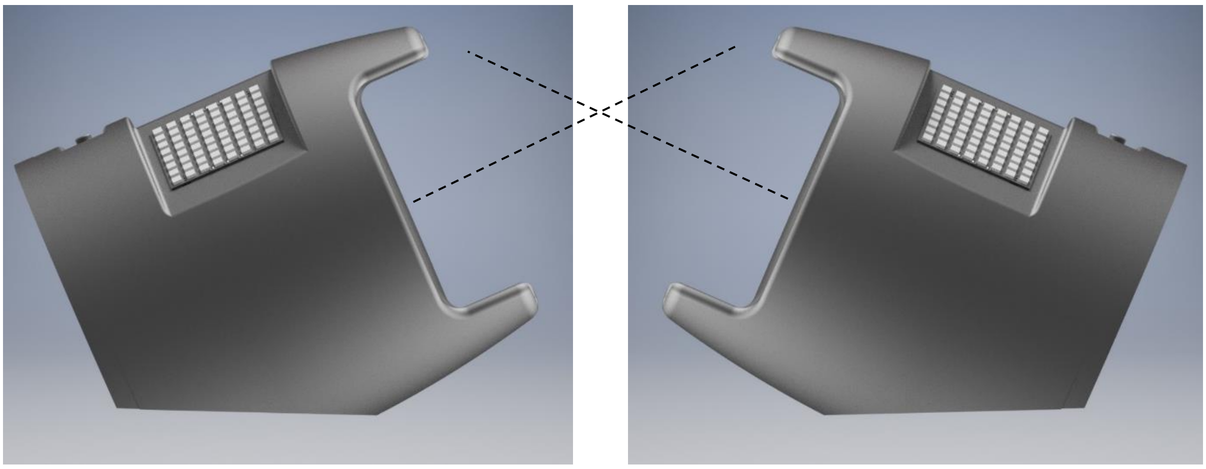


Figure 69: Representation of how two slave components can be positioned on the flattened backing to form a timing gate

Various sport specific game balls were positioned on top of the CAD modelled markers (Figure 71 - Figure 74). These balls were CAD modelled with specifications determined through the research conducted in the conceptual design phase. The models could then be used to sculpt out the enclosure top half so that they could be positioned perfectly to remain positioned until an athlete removes them. Figure 71 to Figure 72 shows various views of an Australian football positioned on the marker. Figure 73 and Figure 74 show a netball and basketball positioned on the marker, respectively.

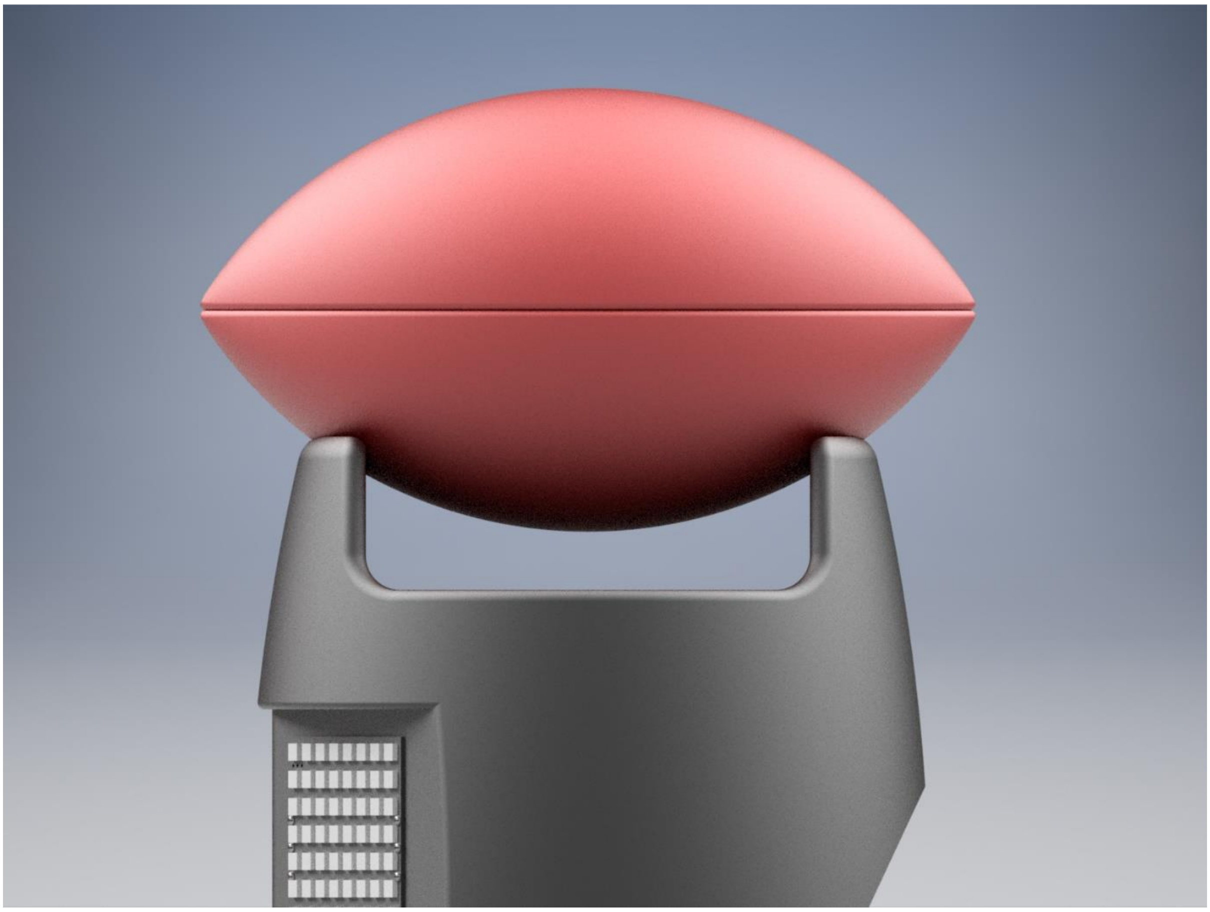


Figure 70: Close-up view of the Australian football positioned on the slave component CAD assembly

Figure 71: Isometric views of the slave CAD assembly with an Australian football positioned on top

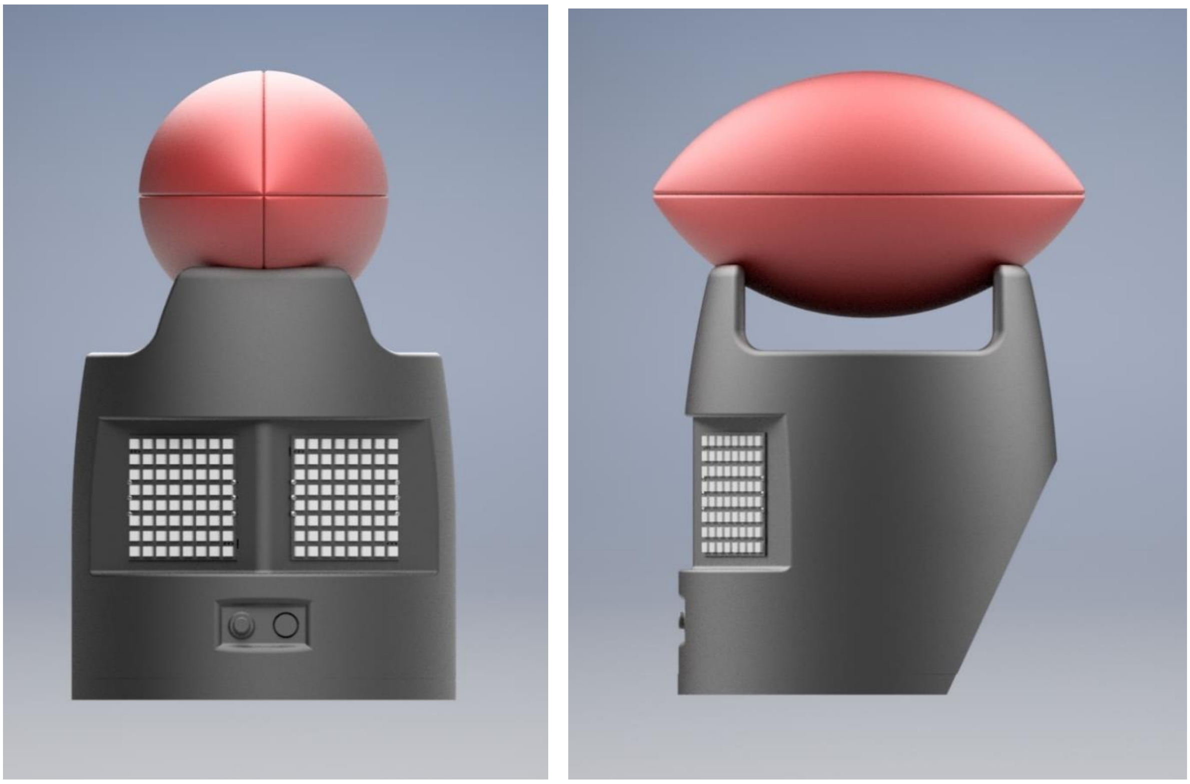


Figure 72: Front and side view of the slave CAD assembly with an Australian football positioned on top

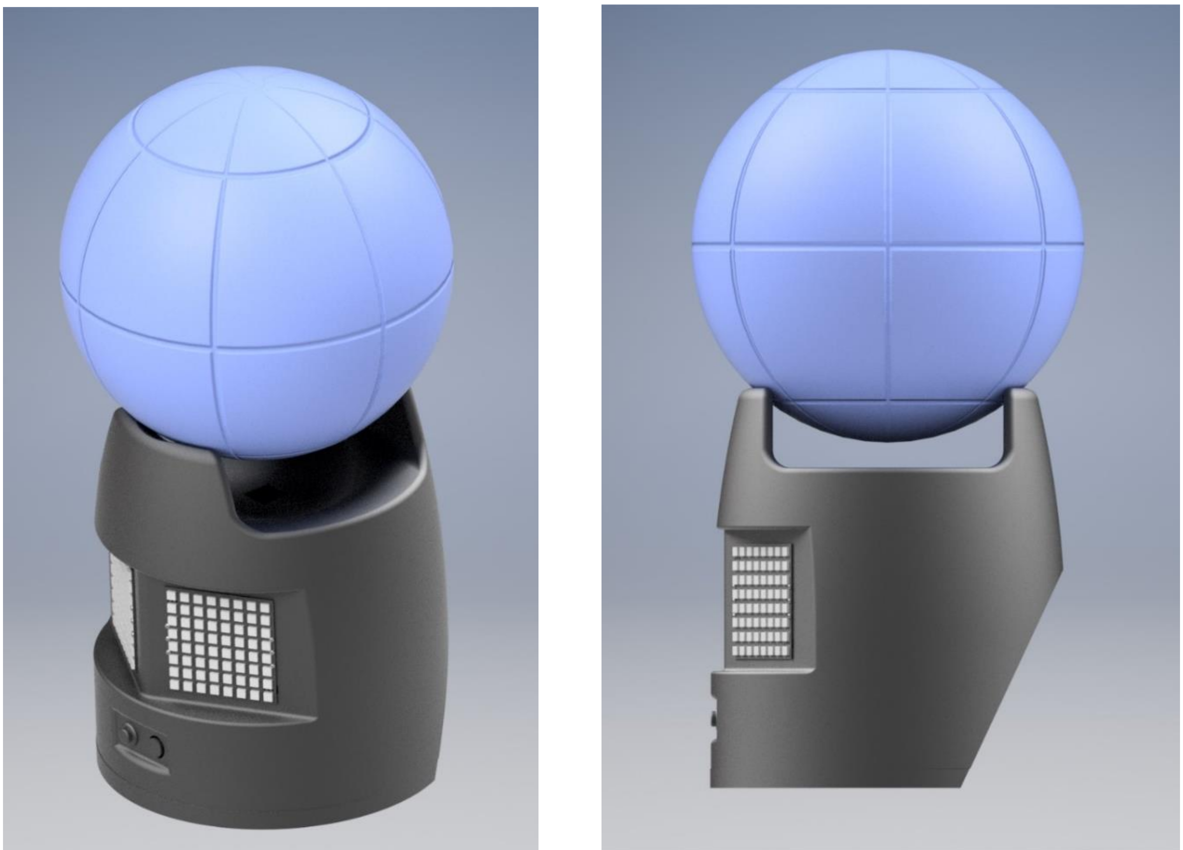


Figure 73: Isometric and side view of the slave CAD assembly with a netball positioned on top

Figure 74: Isometric and side view of the slave CAD assembly with a basketball positioned on top

3.7.2. Summary of Final Solution Concept Embodiment

The engineering design process had chartered through a logical and systematic methodology that enabled the development of a solution concept for both the master and slave components. Through these design layouts visualised in three-dimensional CAD software, the embodiment for the solution concept of the system had been established. This now could provide the framework for physical embodiment of the system by developing and constructing a proof-of-concept prototype. With a preliminary mechanical design, the parts required to be 3D printed were processed for manufacturing whilst electronic component testing was in progress. All electrical hardware was tested in isolation using solderless prototype breadboards and jumper wires initially to ensure the part worked and software program was written appropriately. Wiring the electronics in the correct circuit and programming the hardware was achieved by following the respective datasheets, using Arduino and third party library "Example" code (within the library package), as well as through various community forums; primarily from Arduino (https://forum.arduino.cc), StackExchange (https://arduino.stackexchange.com) and cplusplus (http://www.cplusplus.com/). All coding was completed in the Arduino IDE (Arduino, Ivrea, Italy) for all Arduino based programming and Visual Studio Community (Microsoft, Redmond, Washington, USA) for all computer program-based programming. When all components were confirmed working, integration of one component at a time to the main circuit was accomplished, ensuring compatibility at each stage. The integration of some hardware components caused complications to arise, as well as integration of some mechanical and electrical components bringing forth attention issues with the design. These challenges faced and the methods to overcoming them are discussed throughout the following sections.

The design process for development, construction and assembly of the master and slave devices was completed through an iterative process of concurrent mechanical, electrical and software design. Thus, this thesis provides a logical description of development of all components of design as opposed to describing these separately. The developmental progression for the master and slave component prototype embodiment is discussed separately to maintain flow of this thesis, however these components were also developed simultaneously.

3.7.3. Slave Component Development

With anticipation of developing six slave components, it was imperative to work through the development stage of the marker as quickly as possible. The proposed strategy for developing all devices, was to first focus on integrating all components of one marker together and confirm that the device was working correctly. Once all hardware components were confirmed working, including after developing the prototype shield, then the remaining marker devices would be assembled. Thus, this section shows development of one marker, but final successful developmental iterations were repeated for all other markers.

3.7.3.1. Enclosure Design

As mentioned in Section 3.7.1, many design iterations occurred to get to the final CAD assembly shown. Thus, there are many components in this section that were not shown in the CAD assembly, which were optimised in later stages of the design process as required. The preliminary enclosure design for the marker was 3D printed and is shown in two separate parts (Figure 75). The reason for printing in two parts was already mentioned in Section 3.7.1.2. To reiterate, the enclosure was printed in separation for three primary reasons; one was due to a significant reduction in printing time as this minimised the need for printing supports throughout the centre (which take longer to print than the design itself); another because the Ultimaker 2+ 3D printer does not have a build volume that has a large enough height (The Ultimaker 2+ Extended did, but only two were available for use with restricted access); the last reason was for more accessibility when mounting electronic components. This can be seen in Figure 76, with a top view showing how no top section improved accessibility significantly.

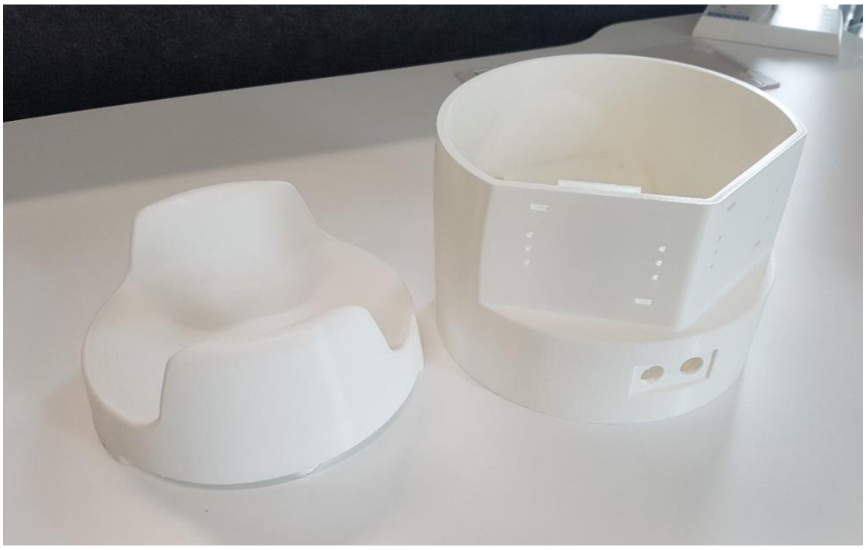


Figure 75: Preliminary marker enclosure printed in two pieces

Figure 76: Top view of the preliminary marker enclosure 3D print showing no top half provides improved accessibility for electronic mounting

One of the key features of the marker was mounting of the sport-specific equipment. The two protrusions were designed carefully so that a match ball could be positioned on top of it. A discussion with Dr. Elliot established the importance of a small gap to allow an athlete to be able to position their hand underneath the ball such that they could swiftly "scoop" it up. Initial conceptualisation was sketched to achieve this (Figure F- 6 in Appendix F: F.1). Adjustments were made to appropriately allow an athlete to successfully position their hand underneath the ball. The design allowed a hand to be positioned within the protrusion gap (Figure 77) for this mentioned feature.

Figure 77: Testing the gap where an athlete can pick it up

The various dimensions of the enclosure were of no accident; the two protrusions on the top were designed specially that they would fit within the gaps if stacked on top of another marker (Figure 78). This was purposefully designed this way so that the system could be more compacted together to fit within a confinement shown from the front view (Figure 78a) and side view (Figure 78b). By designing the marker enclosure in this way mean that the marker height could be increased without sacrificing total volume occupied by the system.

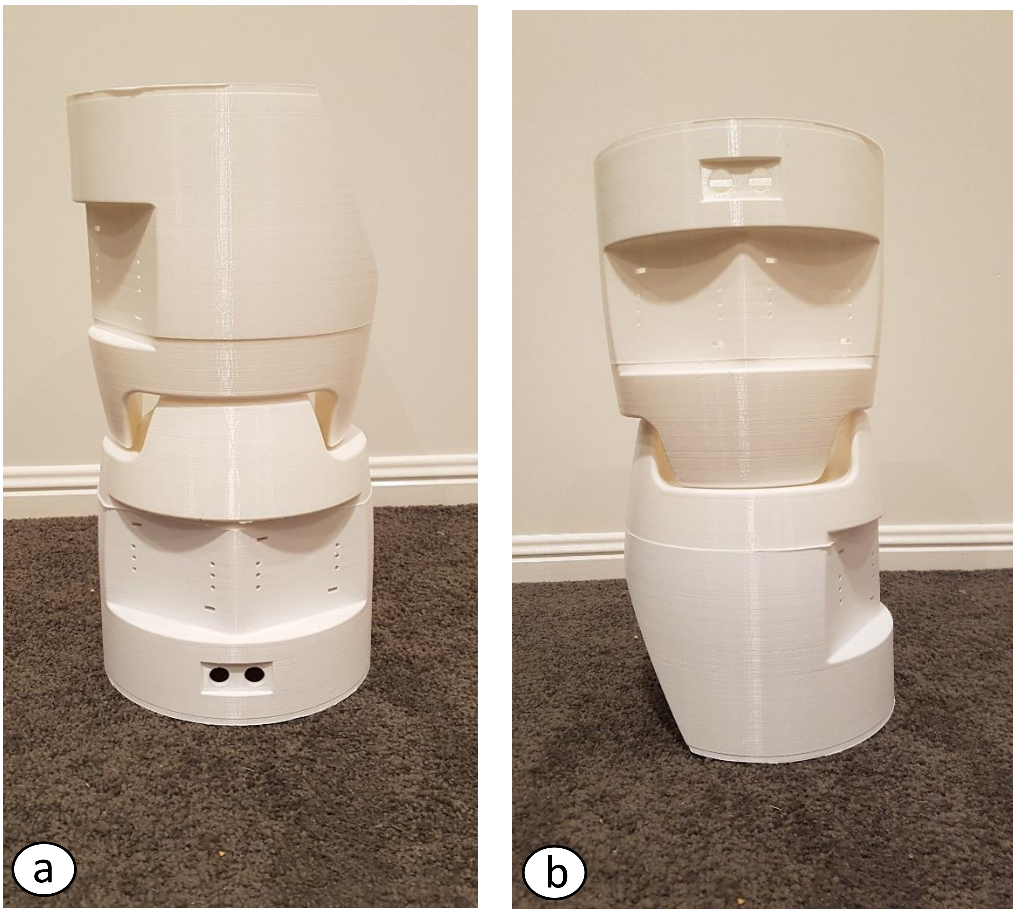


Figure 78: Ball mounting protrusions slot into the neighbouring gaps when the markers are stack.

The mounts for the various electronic hardware was tested with the DC-DC step down buck converter (Figure 79a and Figure 79b) and buttons (Figure 79b). It was quite difficult to get the hardware mounted, due to having to put a nut within small gaps (Figure 79a). Thus, the next iteration of the enclosure allowed the electronics to be fastened into the enclosure without the need for nuts. The only problem with this technique, is that the plastic can become de-threaded by the metal fasteners if screwed and unscrewed too many times; with foresight of mounting electronics once and not removing them, this design change was justified.

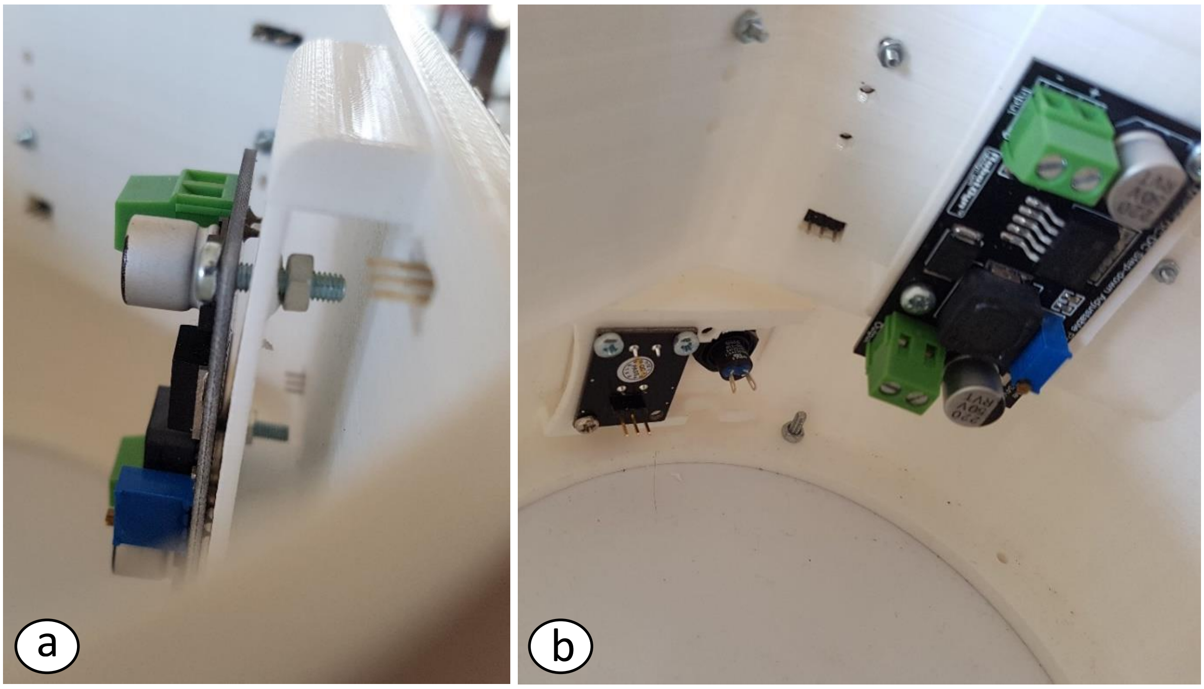


Figure 79: Some of the hardware mounted to the slave component enclosure

An Australian football was positioned on the marker to determine if it was stable (Figure 80). A netball and basketball were also positioned on the marker to determine their stability (Figure 81a and Figure 81b, respectively). The protrusions mounted the match balls quite well, however there was slight instability, which could be a problem with strong winds outdoors. As a result, the protrusions were altered such that the design was sculpted specifically for the various balls. This design change is shown with CAD images for simpler comparison (Figure 82). This image also shows another design change in respect to the ball sensor; a small ledge was added to the design for the laser-cut acrylic square to sit on, for simplicity when affixing it to the enclosure in the future.

Figure 80: Testing the preliminary marker enclosure with an Australian football

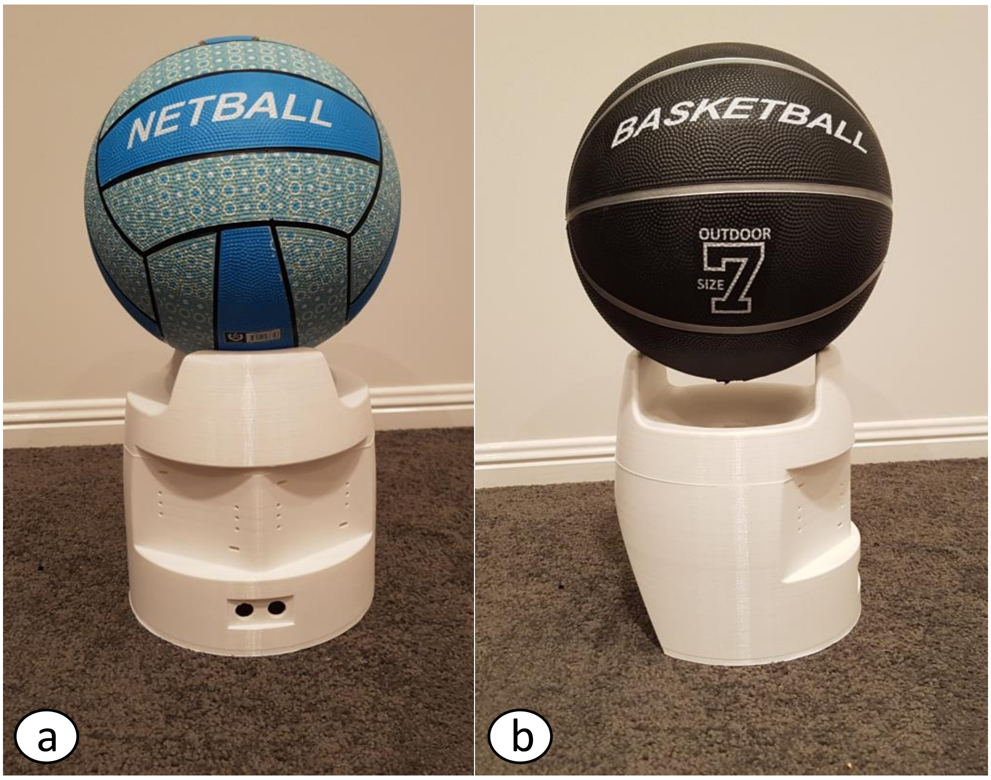


Figure 81: Testing the preliminary marker enclosure with a netball and basketball

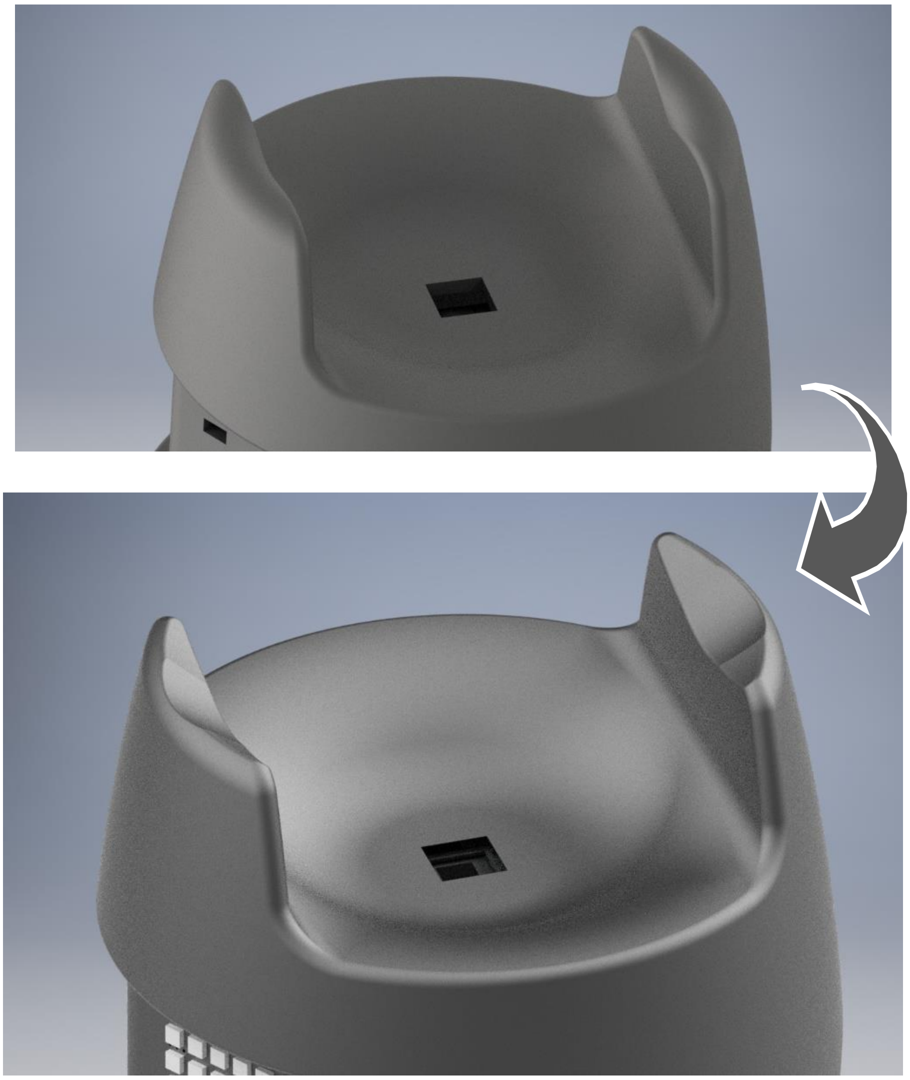


Figure 82: Design change to ball mounting protrusions so the balls will be more stable when positioned on the marker

3.7.3.2. LED Matrix Visual Display

This section describes the development towards displaying characters on the LED matrix display. Initially, the LED matrix was connected to the Arduino Mega to receive digital input and a DC power supply to provide power to the matrix. To familiarise with the matrix, several Adafruit open-source libraries were initially used; NeoMatrix (https://github.com/adafruit/Adafruit_NeoMatrix), NeoPixel (https://github.com/adafruit/Adafruit_NeoPixel) and GFX Library (https://github.com/adafruit/Adafruit-GFX-Library). Example code was uploaded to the Arduino and the matrix was able to successfully display a series of colours (Figure 83).

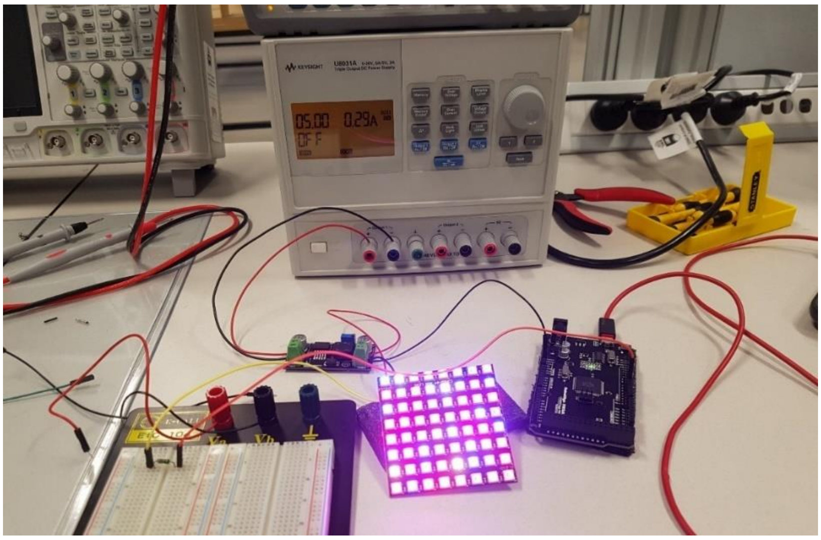


Figure 83: Testing a single LED matrix display

The LED matrix displays used were able to be chained, which is the process of connecting the last pixel (LED) of the matrix with the first pixel of the next. The main concern for the LED matrix was that they were capable of a lot of current consumption. Therefore, the aim of chaining the LED matrix displays was to determine the average current draw. The DC-DC step down buck converters were only able to produce an output of 3A so to chain the matrix displays, it was import that the current draw was less than this. The chained matrices were set with medium brightness (Figure 84), which produced about 0.56A displaying bright purple. The current draw was highly dependent on the colour produced and the number of pixels being used, however each combination produced far less than 3A at full brightness.

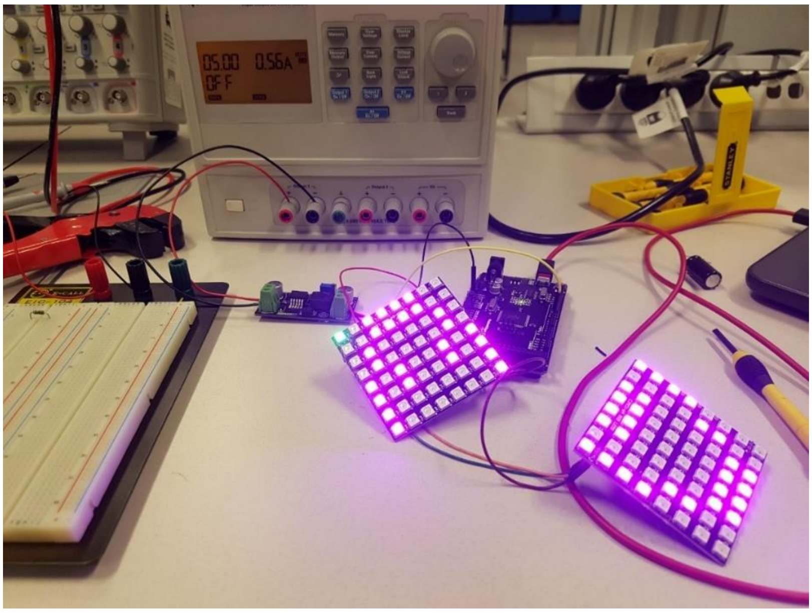


Figure 84: Chaining two LED matrix displays together

The LED matrix displays were mounted to the marker, along with the buttons to test the integration of the hardware and mechanical design (Figure 85). With the LED matrix displays mounted to the marker, it made it far easier to visualise how the characters would appear in the test setting as well as having the opportunity to view from afar.

Figure 85: Mounting the LED matrix to the marker enclosure

A testing process was conducted to visualise what the LED matrix displays would look like lit up on the marker (Figure 86). In the image, the brightness was set to maximum and through observation of the image, it was clear to see the LED displays were extremely bright.

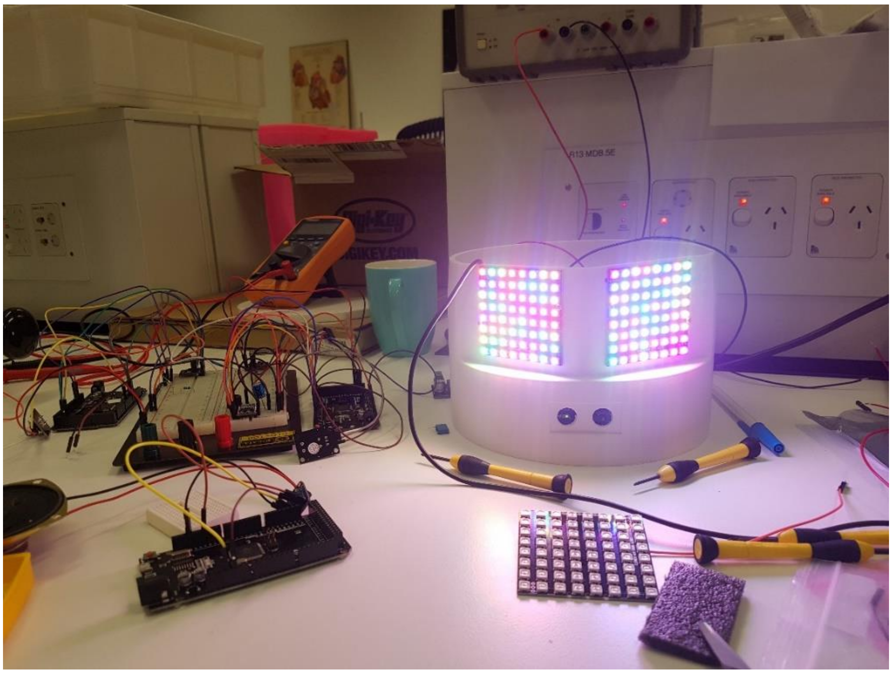


Figure 86: Testing the LED display at full brightness

Through experimentation, a clear understanding of how the LED matrix displays functioned. A more advanced library was interchanged for the Adafruit libraries; that being FastLED (https://github.com/FastLED/FastLED). This library was used because it contained more features more suitable to the project, such as adjusting brightness non-destructively (the Adafruit library brightness control was destructive in nature to the LEDs), easier control of colour using a full HSV (hue, saturation, value) colour scheme, faster memory and math functions and high efficiency (Garcia, 2018).

The next step in this development cycle required determining how to display a character on the LED matrices. Through careful conceptualisation and planning, a method for displaying alphanumeric characters on the marker was established for development. Through testing, it was known that each LED pixel could be addressed individually through an array with the FastLED library, where element 0 represented the first LED, element 1 represented the sequential LED, and so forth, with the last LED being represented by element 128. Thus, it was identified that some code could be written such that it cycled through each element in an array of 64 values, with each element having a value of either ‘1’ or a ‘0’, which would be used to determine whether an LED needed to be on or off. The idea was that binary numbers (numerals of base-2), could be used in conjunction with the code as they are represented as a series of 1’s and 0’s.

An online LED matrix editor was used (https://xantorohara.github.io/led-matrix-editor/) to generate a set of alphanumeric characters as well as some miscellaneous characters, such as a tick, cross and various shapes. Uppercase, lowercase, numeric and miscellaneous characters were developed using the online generator (Figure 87, Figure 88, Figure 89 and Figure 90, respectively).

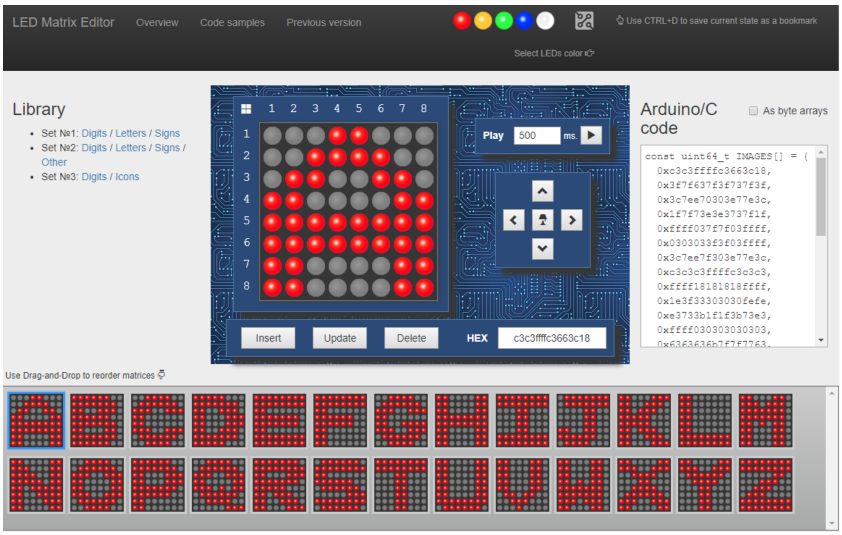


Figure 87: Uppercase characters were custom designed using a LED matrix editing software

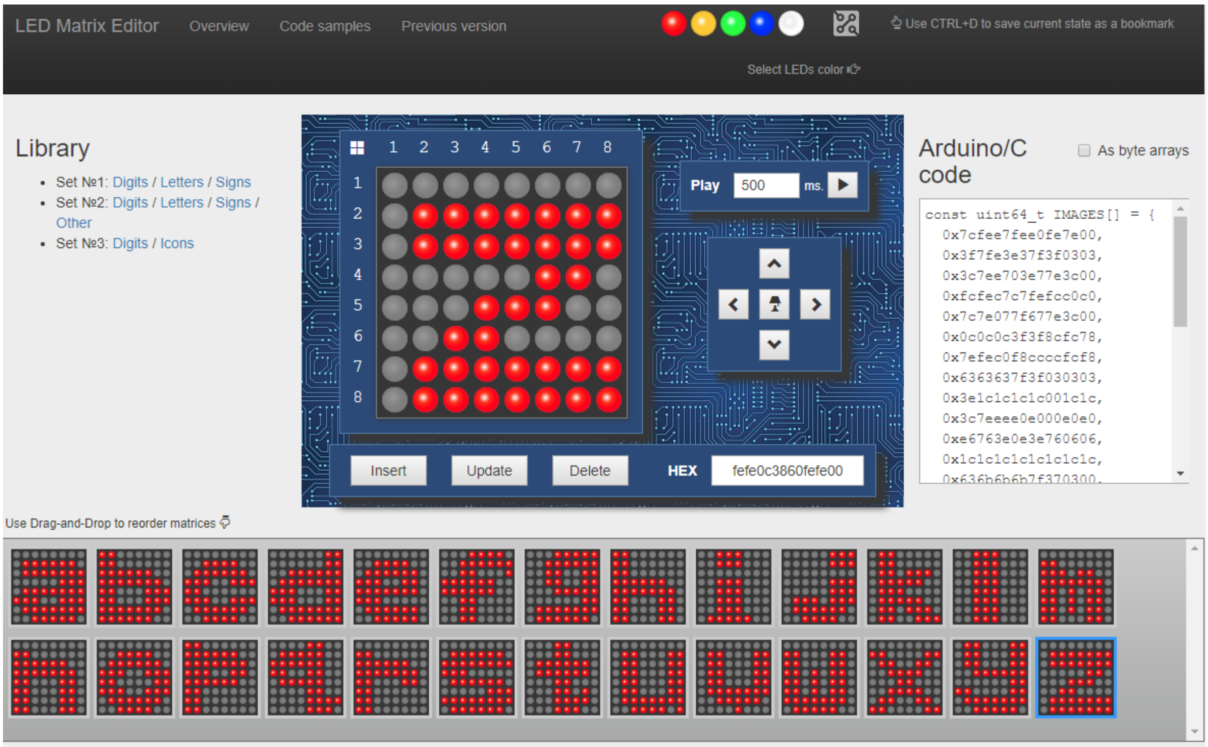


Figure 88: Lower characters were custom designed using a LED matrix editing software

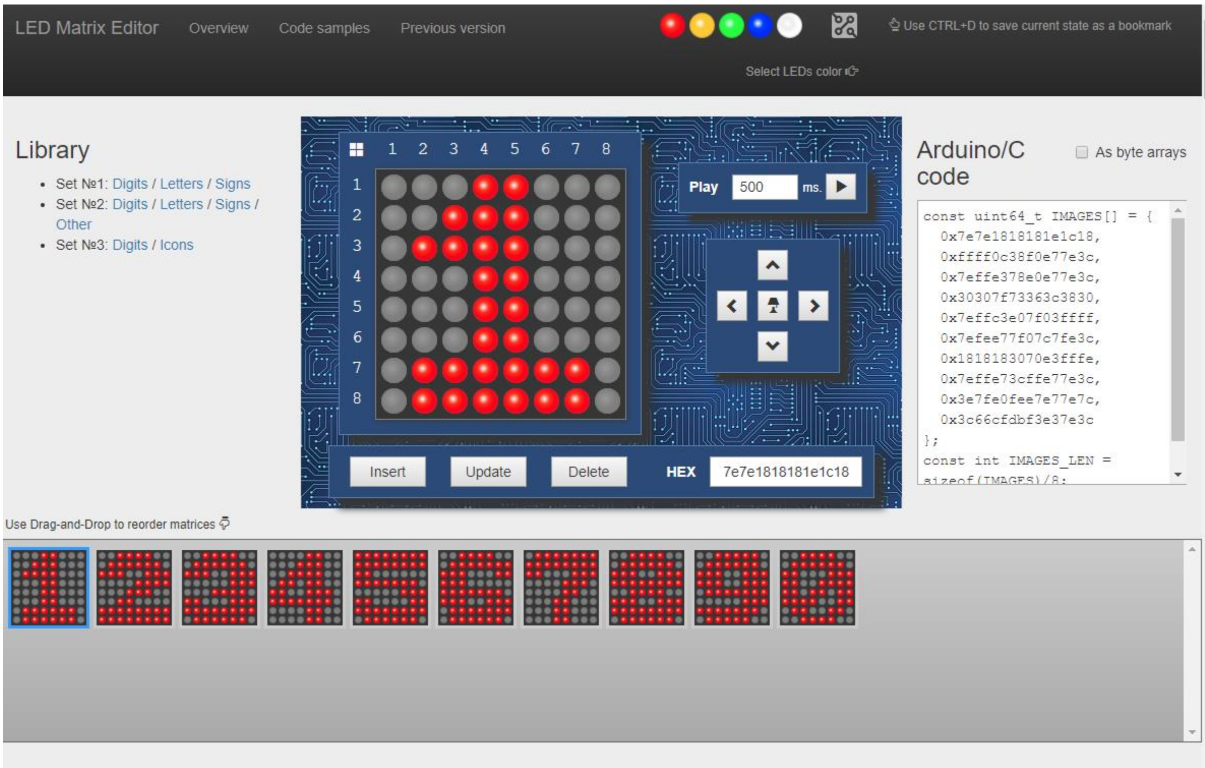


Figure 89: Numeric characters were custom designed using a LED matrix editing software

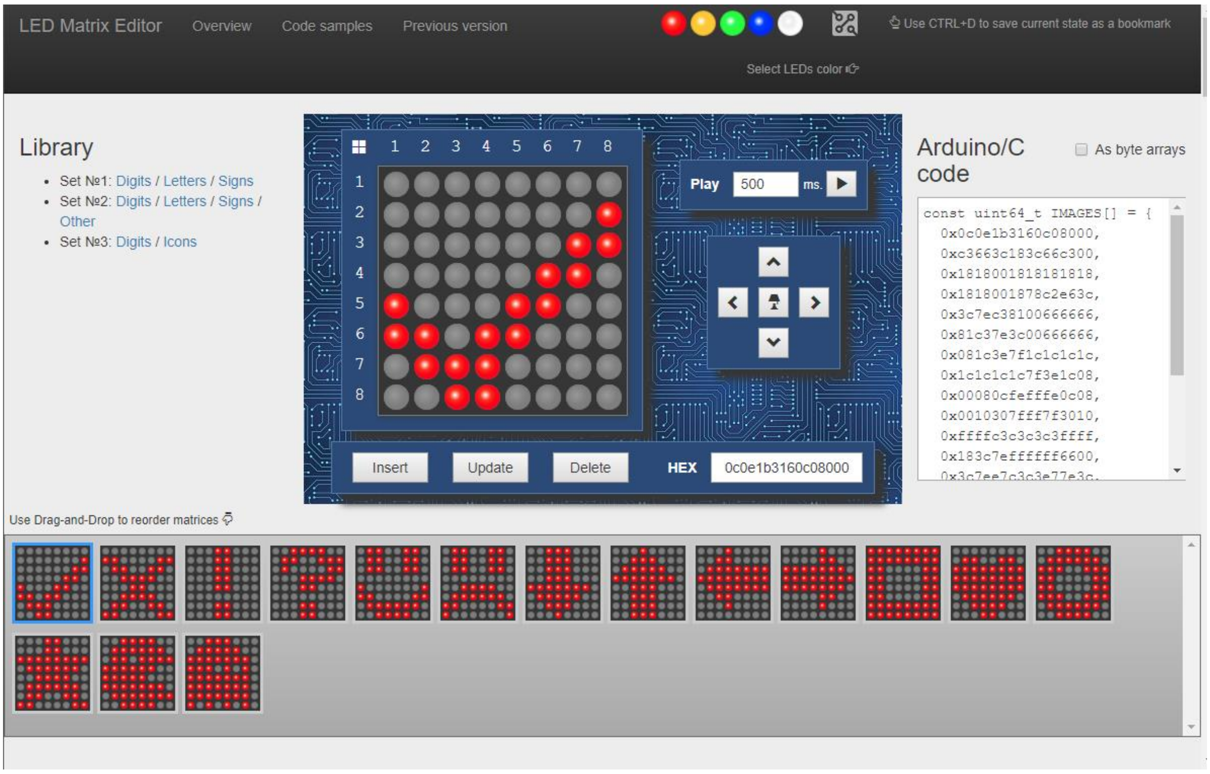


Figure 90: Miscellaneous characters were custom designed using a LED matrix editing software

From the images shown, the code for the LED matrix characters were displayed as hexadecimal value on the right, which are numerical values in base-16. The major problem with the current state of the code, was that each character was in the form of a uint64_t (an unsigned long long integer), so it would not be possible to cycle through each element to determine which LED needed to be on or off without doing some form of conversion. An essential for each device in the proposed system was that it should be fast at processing and quick to run commands and functions. If the program needed to make conversions each time a character was to be used, it would be time consuming and reduce the response time of the system. Additionally, making the conversions at start-up would increase the boot-up time, which one of the specifications was to reduce this as much as possible. A better solution was proposed, which did involve slightly more work initially for the programmer, however, would reduce processing time of the program itself. Thus, a program was designed and developed that would allow a user to be able to simply copy and paste the code from the website into the program and it would generate the appropriate code for the Arduino main slave program (Appendix J: J.1).

To summarise how the program works, a data structure “hexData” was created, which holds a character name value and the hexadecimal value that the user supplies to the program. It then converts the hexadecimal character value into a string in binary form with just one line of code. However, upon analysis, it was determined that the hexadecimal number provided by the program produced a binary array that was represented as a flipped and mirrored character. Thus, appropriate adjustments were made to rearrange the characters in the correct order for the Arduino program. The program prints to output a multilayered array (Figure 91), which contains an array of the characters, which are in the form of a 64-element character array for each character.

```
const char LETTERS_UPPER[][65] PROGMEM = { // Store in PROGMEM to save SRAM
  // [0] A:
  { "0001100000111100011001101100001111111111111111111100001111000011"},

  // [1] B:
  { "1111110011111110110011101111111001111111011000110111111101111100"},

  // [2] C:
  { "0011110001111110111001111100000011000000111001110111111000111100"},

  // [3] D:
  { "1111100011111110110011101100011111000111110011101111111011111000"},

  // [4] E:
  { "1111111111111111110000001111110111111101100000011111111111111111"},

  // [5] F:
  { "1111111111111111110000001111100111111001100000011000000110000000"},

  // [6] G:
  { "0011110001111110111001111100000011001111110011101111111000111100"},

  // [7] H:
  { "1100001111000011110000111111111111111111110000111100001111000011"},

  // [8] I:
  { "1111111111111111000110000001100000011000000110001111111111111111"},
```

Figure 91: Output of the program that generates the LED matrix friendly character code

Integrating the code into the slave program successfully produced characters that were able to be displayed on both LED matrix displays. The characters ‘A’ and ‘X’ are shown (Figure 92a and Figure 92b, respectively), with ‘Y’ and ‘Z’ also shown (Figure 93a and Figure 93b, respectively).

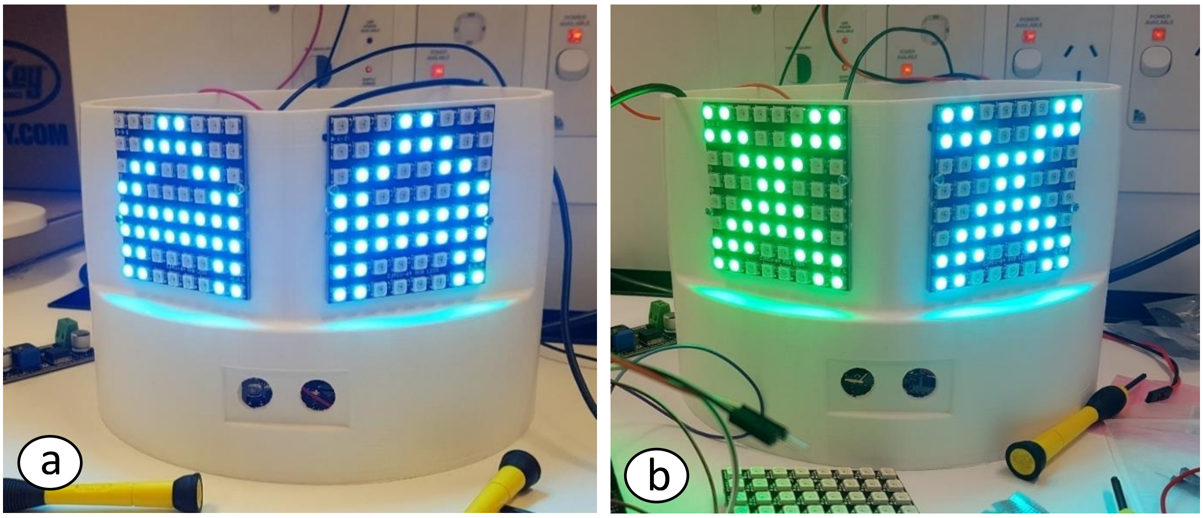


Figure 92: Displaying an 'A' and 'X' on the marker LED matrices

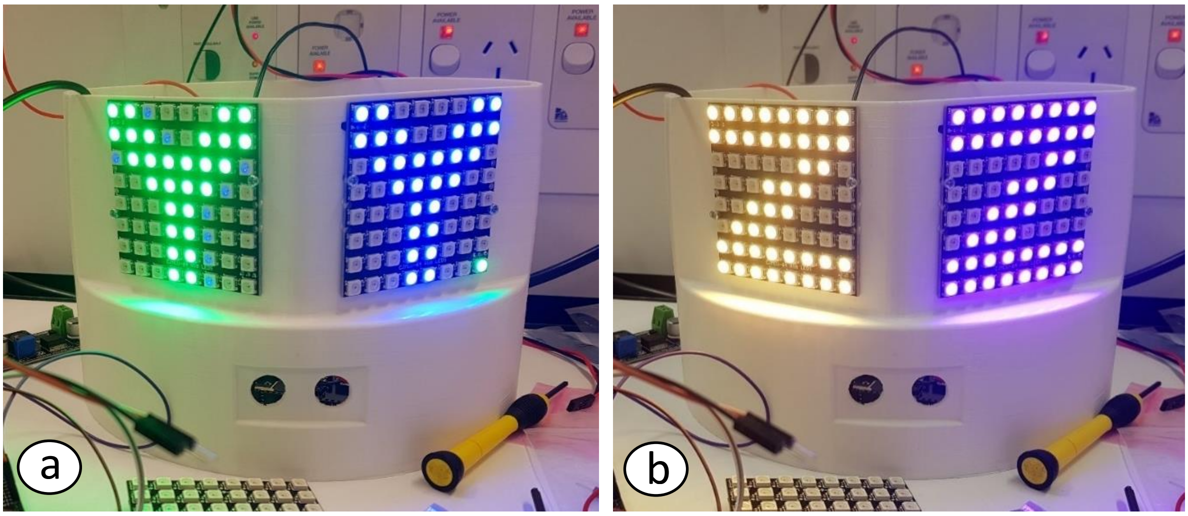


Figure 93: Displaying a 'Y' and 'Z' on the marker LED matrices

3.7.3.3. Snap-fit Enclosure Base

A snap-fit enclosure was developed for the slave component, to meet the target specification of not using tools for recharging or changing the battery. The cross-sectional image of the snap-fit assembly was shown in Figure 63 in Section 3.7.1.2. After printing the components and trialling, the snap-fit cover was able to fit within the cover (Figure 94a) but failed with the part having a brittle fracture when trying to push it out (Figure 94b). Upon inspection, the brittle fracture occurrence made sense as the 3D printing orientation was parallel to the fracture orientation. This meant that with the current part, to release the snap-fit cover, an applied force needed to occur, which acted against the weakest point in the design. Therefore, a trial was completed, printing the cover in an upright position that is generally not considered. Testing the snap-fit assembly again produced a successful design, with the component not failing. This was because the 3D printing orientation was now perpendicular to the applied force, giving it greater strength in this direction. The two printing orientations are shown in Figure 95; the orientation that produced a failed design (Figure 95a) and the one that produced a successful design (Figure 95b).

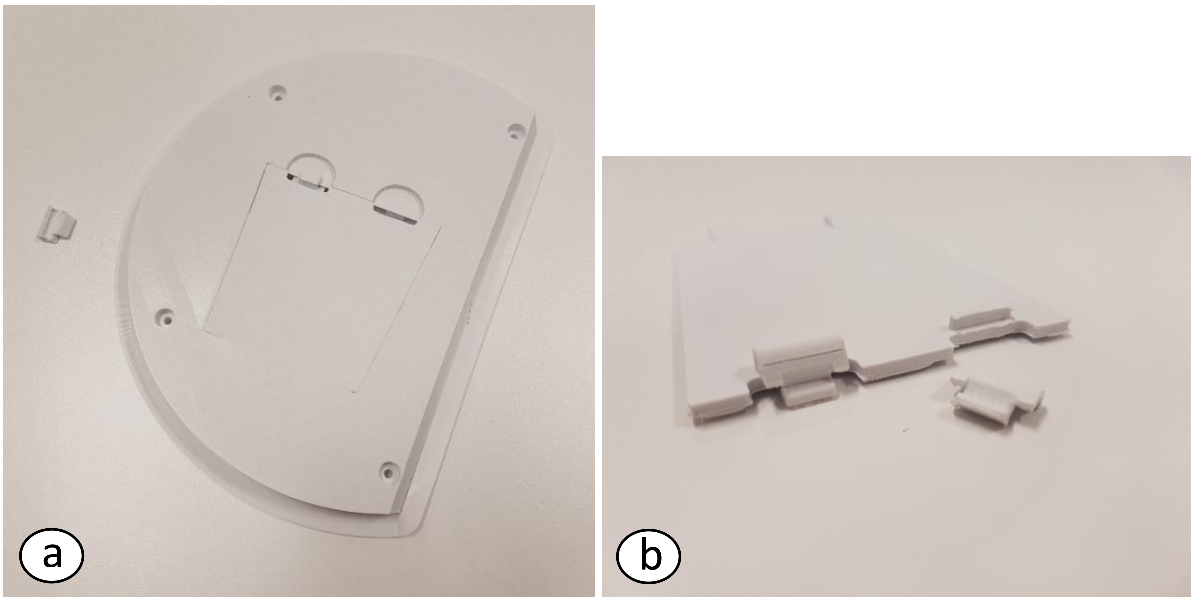


Figure 94: Failed snap-fit component cover

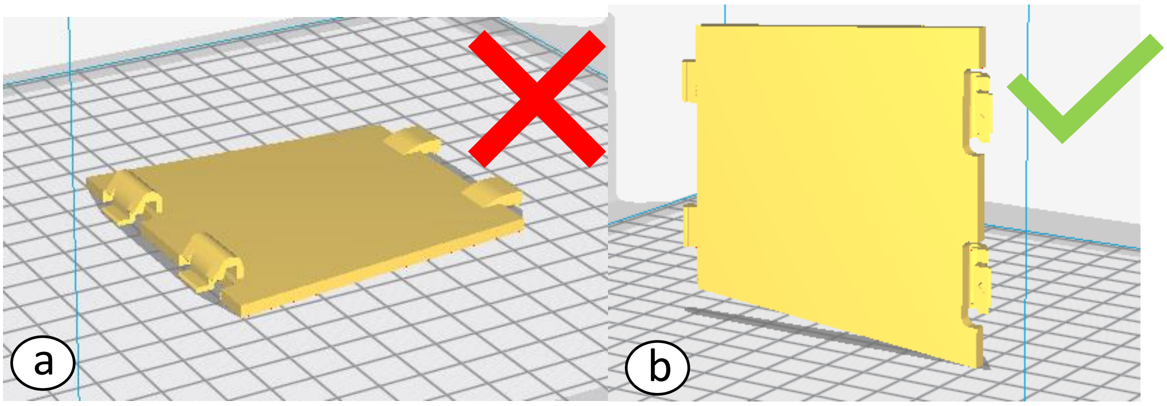


Figure 95: Printing orientation was vital to ensure printing of a high-strength component that would not break

The successful snap-fit components can be seen within the slave base (Figure 96a) and stand-alone (Figure 96b). Upon testing, it was found that it was very difficult to release the snap-fit cover using fingers only. This could be because of two reasons; the printing material (PLA) may be too rigid, or the snap-fit locking section is too thick. It was found that using any flat object was able to release the cover, such as a key (Figure 97). The specification for not to use tools to change/recharge the battery was implying not having to use a tool that may not be available in all situations. However, most people will always carry a pair of keys with them, so the chances of the user or one of the athletes having a set of keys is high. As a result, this specification was established as being met regarding the slave component, as keys are not considered a tool that is not readily available.

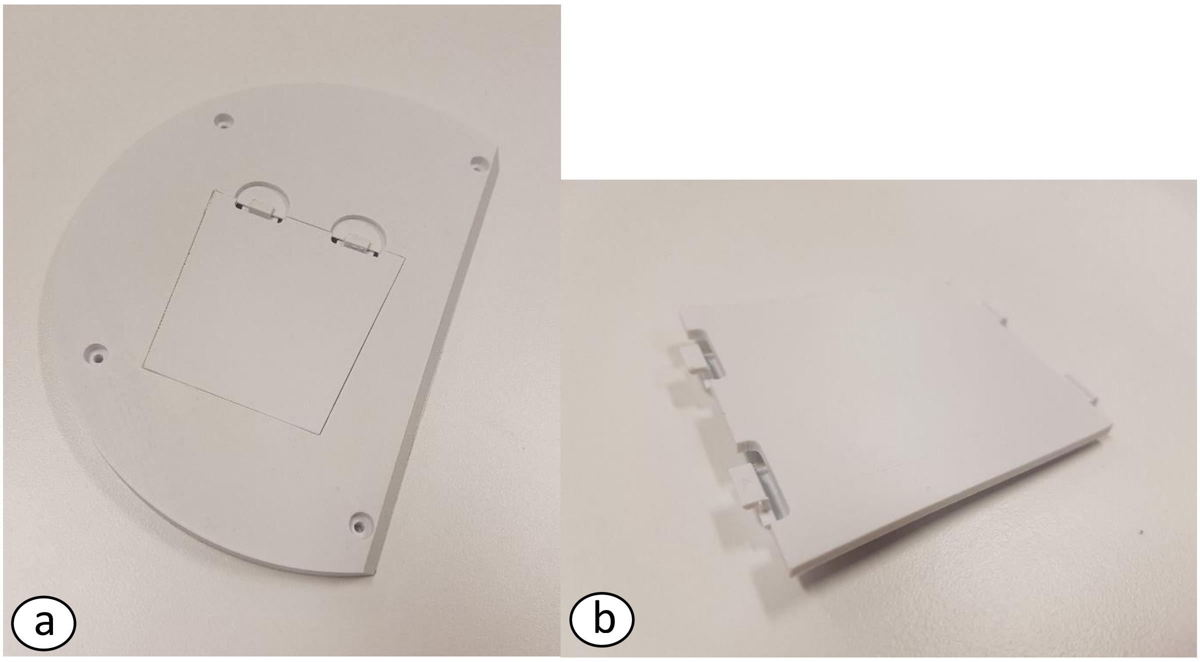


Figure 96: Final snap-fit components for the marker base

Figure 97: A simple set of keys can unhinge the snap-fit cover

3.7.4. Master Component Development

3.7.4.1. Hardware Testing

Various hardware components of the master were tested in isolation to ensure that they were working correctly. This also gave the opportunity to produce code that could allow the hardware to complete its desired function.

3.7.4.1.1. LiDAR-Lite v3HP

Various stages of a test cycle was conducted for determining the accuracy of the LiDAR scanner (Figure 98 and Figure 99). The LiDAR-Lite v3HP was connected to the Arduino Mega according to the datasheet (Garmin Corporation, 2016), mounted such that it was parallel with the ground surface. The LiDAR scanner was programmed using the library supplied by Garmin; LIDARLite Arduino Library (https://github.com/garmin/LIDARLite_Arduino_Library). An 8 m tape measure was extended from the LiDAR scanner, with a medium sized box placed at various distances in the centre of the tape. When using the medium sized box, the accuracy was found to be within approximately ±2.5 cm. This met the delighted target engineering specification for distance measurement accuracy.

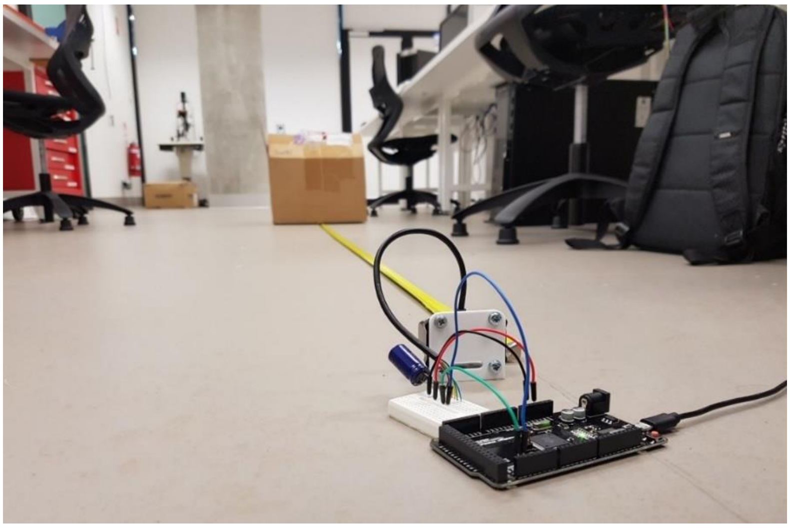
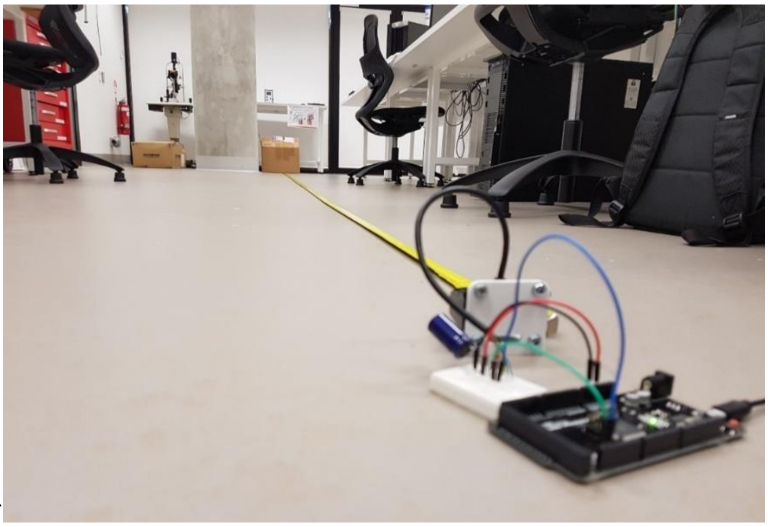


Figure 98: Testing LiDAR accuracy using a box



`

Figure 99: Testing LiDAR accuracy using a box at 8m away

3.7.4.1.2. Laser Diode Module

A major problem experienced during this test phase, was that it was incredibly difficult to align the LiDAR scanner with the box as the laser was invisible to the human eye; thus, differentiating a measurement from the neighbouring pole (Figure 99) was troublesome. The need for a visible laser for feedback where the LiDAR scanner is measuring was imperative for ease of use of the system whilst setting up the markers. This had been considered in the conceptualisation stage of the project. A laser diode module with low power (< 1 mW) and 650 nm wavelength (red) was purchased from a local hobby shop that was configured to be compatible with Arduino. The laser was tested indoors at up-close distances (Figure 100a) and extended distances (Figure 100b). The laser diode module had surprisingly good results, considering the laser was of low power; it was easily viewable at large distances over 10 m away (not shown in the figure). The laser diode module was eventually tested outdoors (Section 3.8.2.1) and was observed to be viewable in direct sunlight.

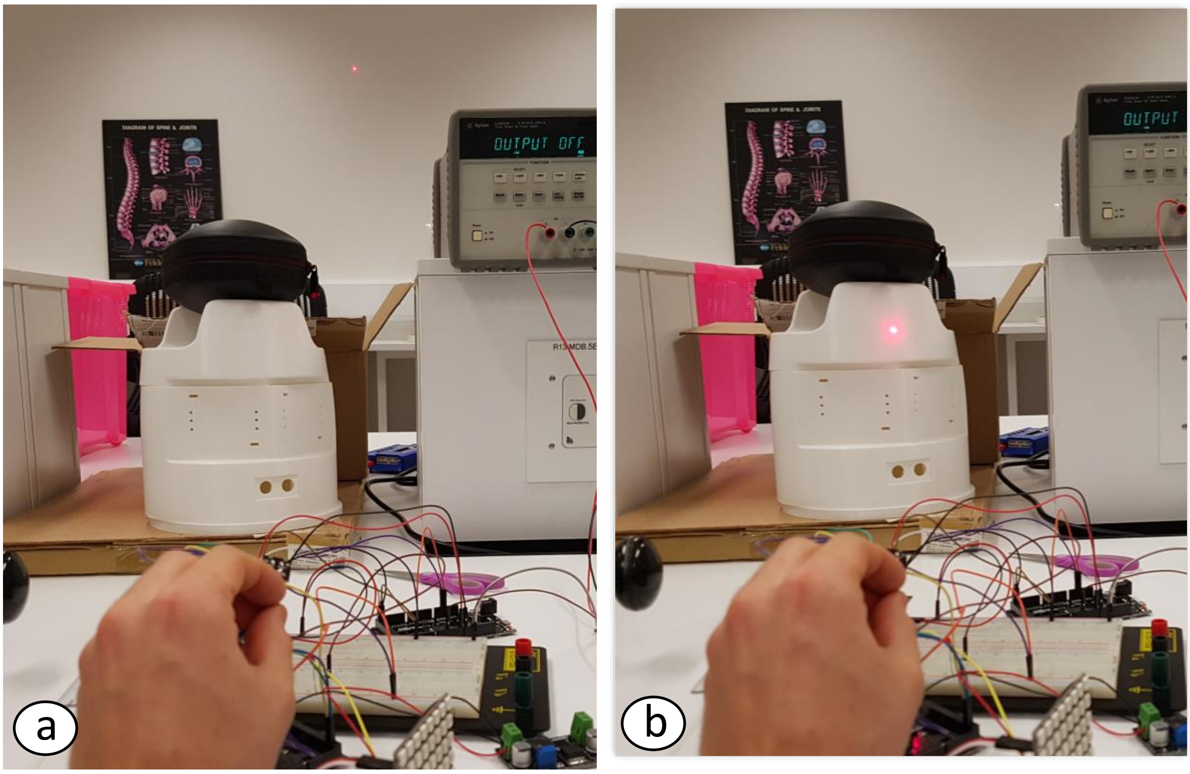


Figure 100: Testing the laser diode module

3.7.4.1.3. RGB LED Module

Another form of feedback mechanism implemented was a visual RGB LED module (Figure 101). The module is made of three LEDs positioned very close to each other, similarly to the LEDs on the matrix displays on the marker. The red, green and blue LEDs are controlled using three digital

PWM pins. The brightness of each LED could individually be controlled using PWM duty cycle output control, which is a value between 0 and 255, with 0 being off and 255 with maximum brightness. Adjusting various brightness intensities of the LEDs produces different colours. This module could be used for feedback when pressing a button, when various stages of the program are complete, or simply to use it as an indicator to show when the device is on.

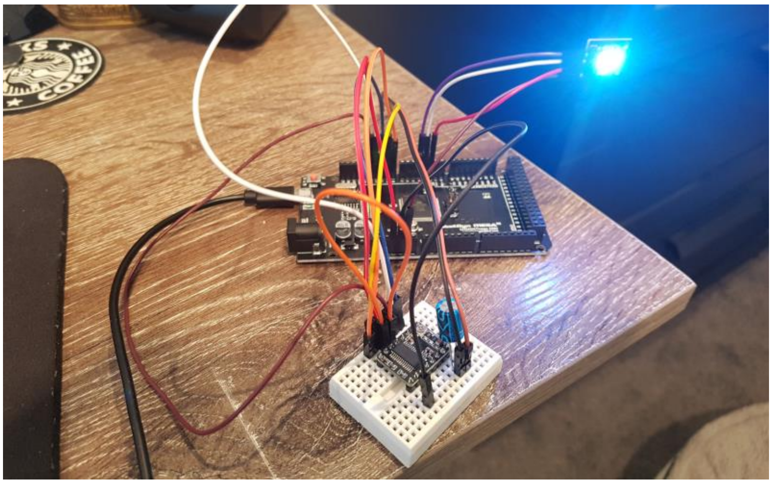


Figure 101: Testing the RGB LED module

3.7.4.1.4. Stepper Motor

To use and control a stepper, either a stepper driver carrier could be used or a prototype shield. As it was intended to use the Arduino Mega prototype shield with components that would likely protrude from it, restricting any other shields to be positioned on top, the stepper driver carrier was selected as the alternative to use. Research during the solution concept stage of the project found the DRV8825 breakout board (Pololu Corporation, Las Vegas, Nevada, USA) a suitable candidate to control the stepper motor. The driver was connected according to the minimal wiring diagram (Figure 102) specified on the Pololu product page.

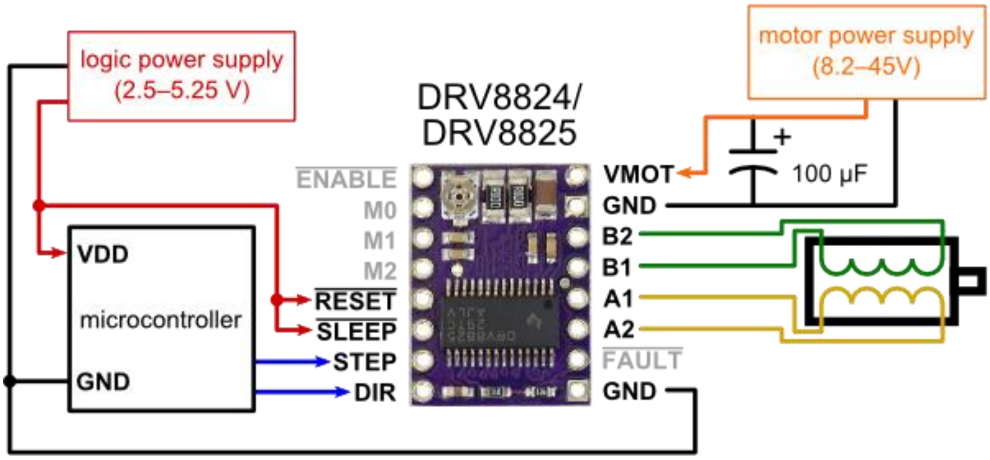


Figure 102: Minimal wiring diagram followed for connecting the DRV8825 to the stepper motor and Arduino (Pololu Corporation, 2018a)

Initially, it was unknown which stepper motor wires were classified as A1, A2, B1 and B2. Referring to the product page for the stepper motor purchased for this project (also a product of Pololu Corporation), the wiring diagram mentioned leads A, C, B and D (Figure 103).

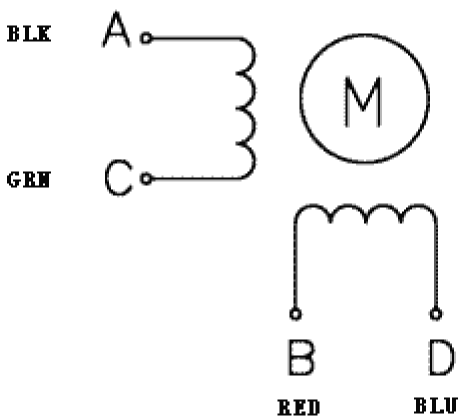


Figure 103: Motor wiring diagram for the stepper used in this project (Pololu Corporation, 2018b)

Referring to the FAQ section of the product page for the DRV8825, Pololu described stepper leads A and C were to be connected to the stepper motor driver board outputs A1 and A2, respectively; stepper leads B and D to board outputs B1 and B2, respectively. This information, along with the corresponding colours for the stepper motor used in this project, was tabulated for a visual representation of the connections (Table 29).

Table 29: Stepper motor lead connections with motor driver carrier

Stepper Motor Lead	Corresponding Stepper Motor Colour	Stepper Motor Driver Carrier Board Output
A	Black	A1
C	Green	A2
B	Red	B1
D	Blue	B2

Before testing the stepper, it was important to ensure the stepper motor driver current limit was set correctly so that the driver and stepper were not damaged. The stepper motor was rated at 1.7 A/phase; thus, this was the recommended current limit of the motor. The maximum holding torque of the motor was 3.7 kg-cm, which was more than required since the system was of low-load. Therefore, it was deemed acceptable to reduce the current limit to 1 A/phase, which would consume less current and allow the system to run cooler. The corresponding VREF voltage to set the DRV8825 to could be determined (Equation 6).

Equation 6: Calculating VREF voltage for limiting the current limit of the stepper driver

$$\text{Current Limit} = \text{VREF} \times 2$$

$$\therefore 1\ \text{A} = \text{VREF} \times 2$$

$$\therefore \text{VREF} = 0.5\ \text{V}$$

Therefore, a VREF voltage of 0.5V was required to be set on the motor driver carrier. When setting this normally, multimeter probes are positioned on the GND pin (for the negative probe) and the VREF via (a small metal circle to place the positive probe). However, the DRV8825 used was a clone of the official Pololu product; most clones do not have the VREF via. Therefore, to measure the VREF voltage in this instance, the multimeter probe was placed on the top of the potentiometer. However, after many attempts, no voltage reading could be obtained from the DRV8825. Flinders University happened to have some spare DRV8825 motor driver carriers, but the voltage could not be measured on these either; coincidentally, the parts obtained from Flinders University were also clones. With options now limited, a DRV8834 was available to be tested which is a low-voltage driver, almost identical to the DRV8825 regarding the breakout board. The motor driver carrier was connected according to the minimal wiring diagram (Figure 104).

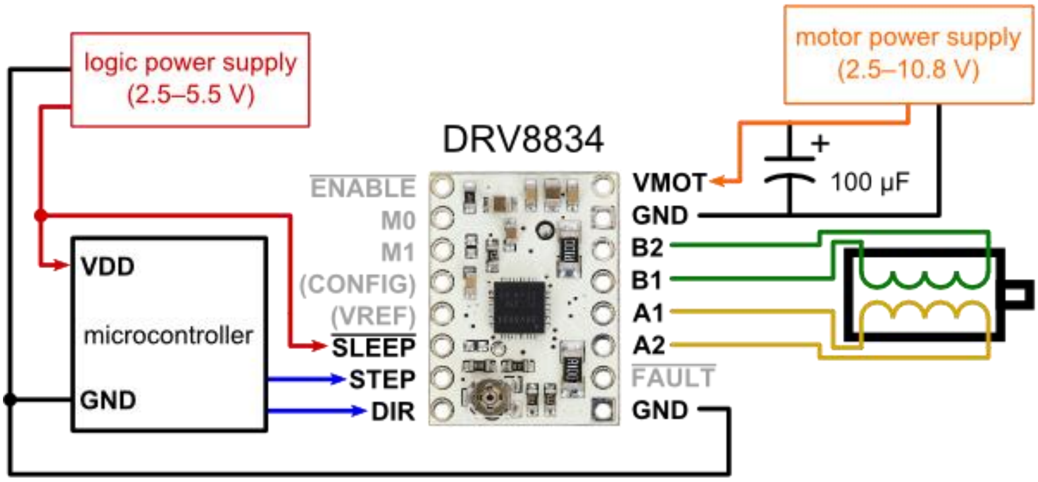


Figure 104: Minimal wiring diagram followed for connecting the DRV8834 to the stepper motor and Arduino (Pololu Corporation, 2018c)

The VREF voltage was able to be successfully measured using the DRV8825 stepper driver carrier. Thus, with the current limit set, the stepper motor was able to be successfully tested. The driver was programmed using an Arduino library, StepperDriver (https://github.com/laurb9/StepperDriver), which was written for the A4988, DRV8825, DRV8834 and DRV8880 stepper motor driver carriers. The DRV8834 stepper motor driver has capacity to microstep up to 1/32 of a step. With the stepper producing a 1.8^{o} step angle (200 steps/revolution), a 1/32 microstep could reduce the step angle down to 0.05625^{o}. Therefore, the stepper motor could be moved in increments of 0.05625^{o}, which woud be extremely precise. This meant that when determining the angular positioning of the markers relative to the master component, the maximum absolute error would be ± 0.05625^{o}.

3.7.4.1.5. nRF24L01+ Module

The nRF24L01+ modules were programmed using an Arduino library, RF24 (https://github.com/nRF24/RF24). Communication was successful between two Arduino Mega boards upon programming and testing (Figure 105). In this testing cycle, the nRF24L01+ modules were directly next to one another. It was not until later in the development of the system, that complications arose when the modules were not directly neighbouring one another. This was deemed a significant problem; with it occurring further in the development, a more detailed description of this complication is discussed later in Section 3.7.10.

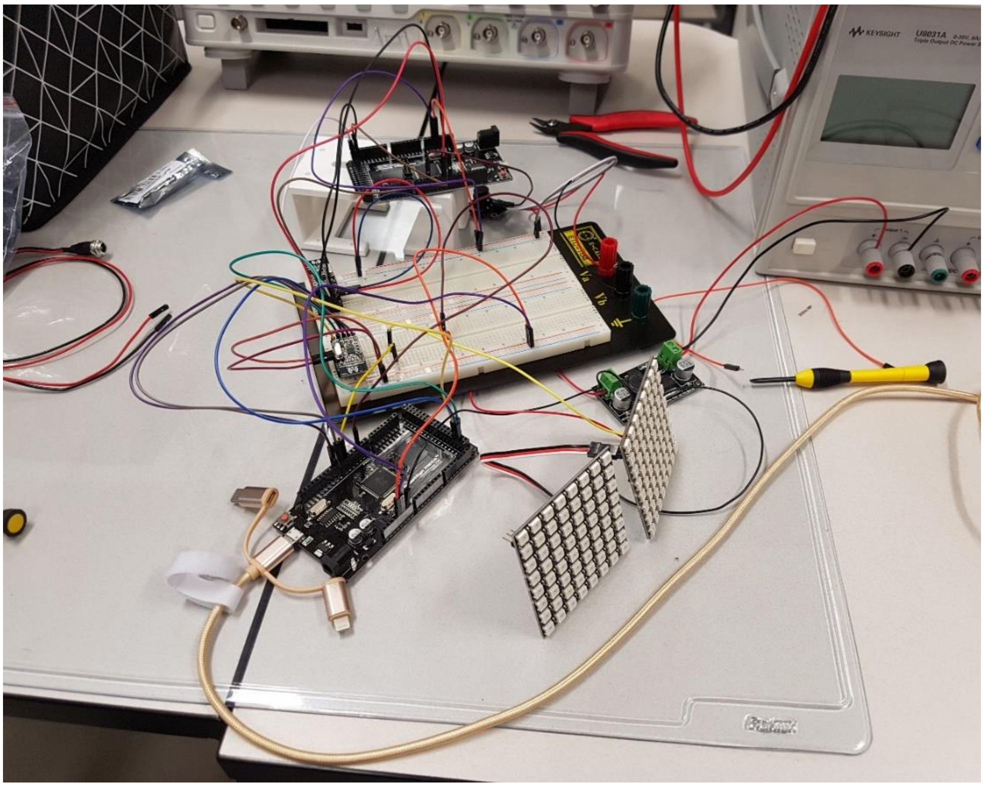


Figure 105: Testing the nRF24L01+ modules

3.7.4.2. Stepper Angular Positioning Coupled Gearing Mechanism

It was established during alternatives evaluation in the solution concept design stage that laser-cut acrylic gears would be used. However, upon inspection of available materials, it was found that Flinders University also had Acetal (polyoxymethylene, POM). Acetal is a common material used for gears in low load and torque mechanisms as it has a low slip ratio and low wear rate (Kukureka et al., 1995). Additionally, these gears can be used unlubricated and thus require minimal maintenance. Thus, laser-cut acetal was used to manufacture the spur gears instead of acrylic (Figure 106).

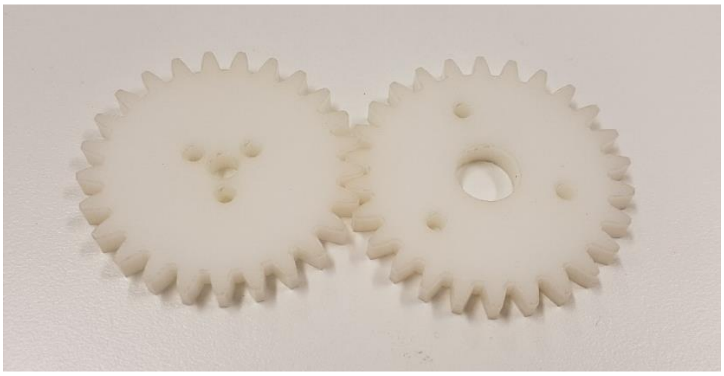


Figure 106: Laser-cut acetal spur gears for coupling the stepper angular rotation to the LiDAR scanner

The gears were able to be mounted, alongside the other various 3D printed components (Figure 107). The coupling mechanism was tested, and rotation of the LiDAR scanner was successful.

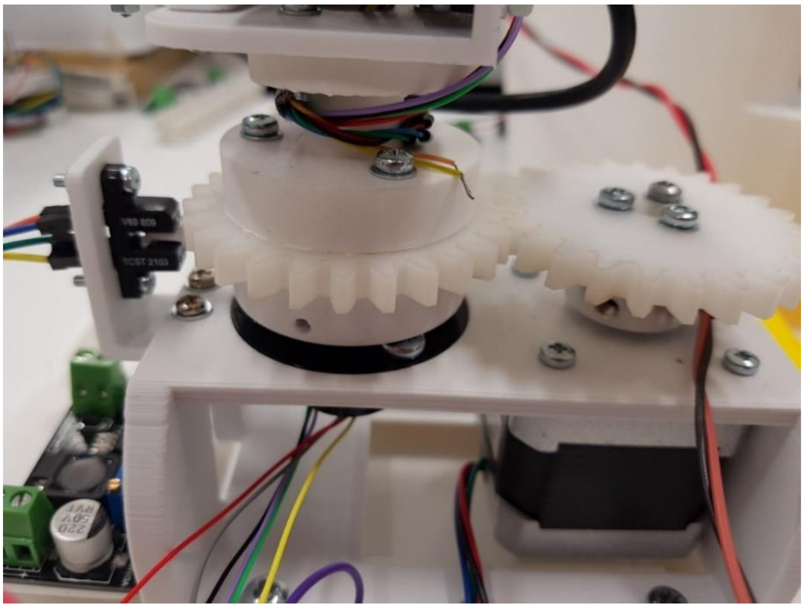


Figure 107: Coupled angular positioning gearing mechanism assembly

The final electronic hardware and circuitry for the master component was completed, with all components integrated together using multiple breadboards and prototype jumper leads (Figure 108). The top view (Figure 109) allows observation of the multiple hardware components, showing the complex circuitry, with many different connections and jumper leads scattered.

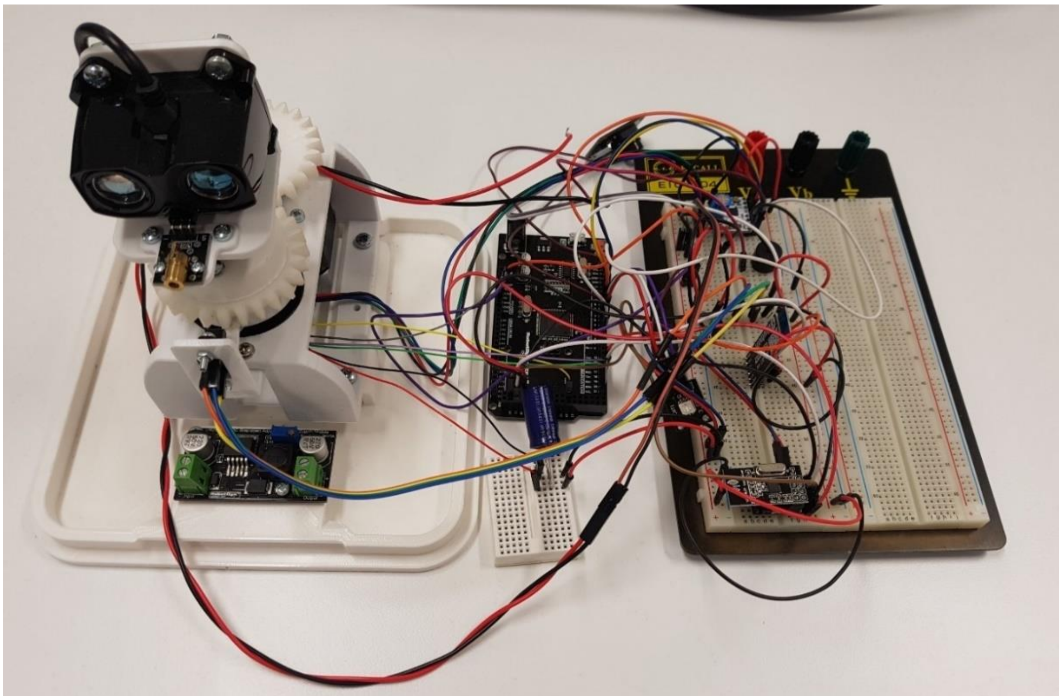


Figure 108: Electronic componentry circuit connects for the master component

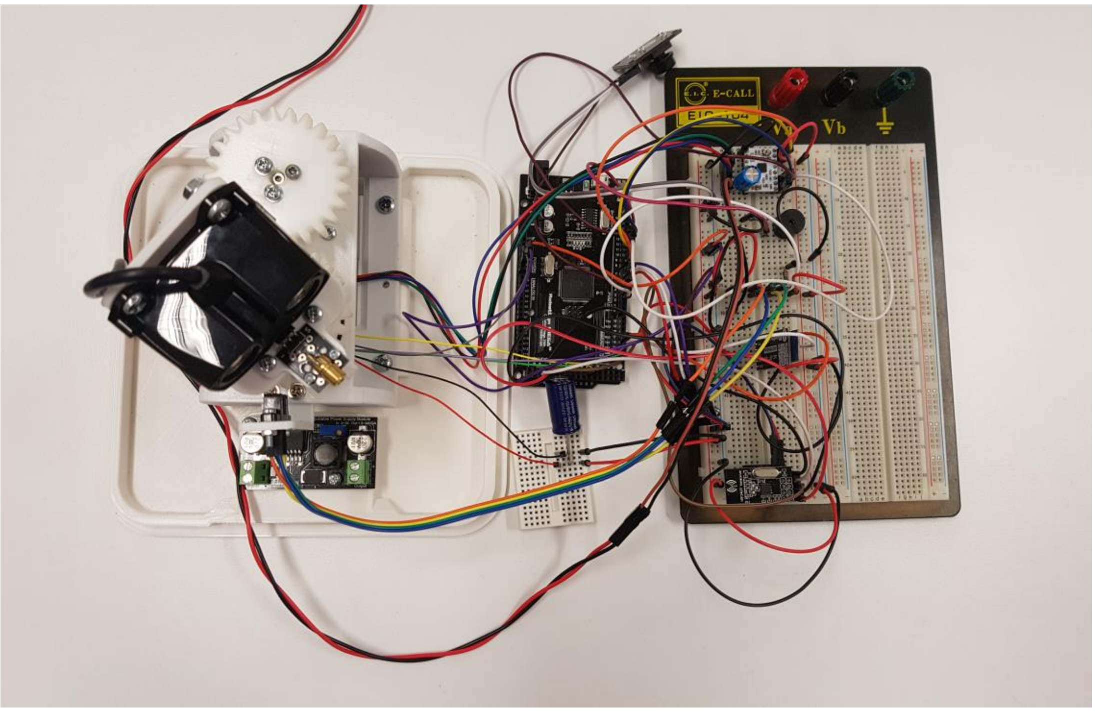


Figure 109: Top view of the final master electronic circuitry of hardware

As testing was conducted with compatibility of 3D printed parts, slight design modifications and refinement occurred through an iterative process. An example of these design refinements and optimisations is shown through refinement for the stepper motor and slip ring mount (Figure 110). One more refinement was completed after this image later in the embodiment when a complication arose during properly wiring the system, which is discussed in Section 3.7.7.1.

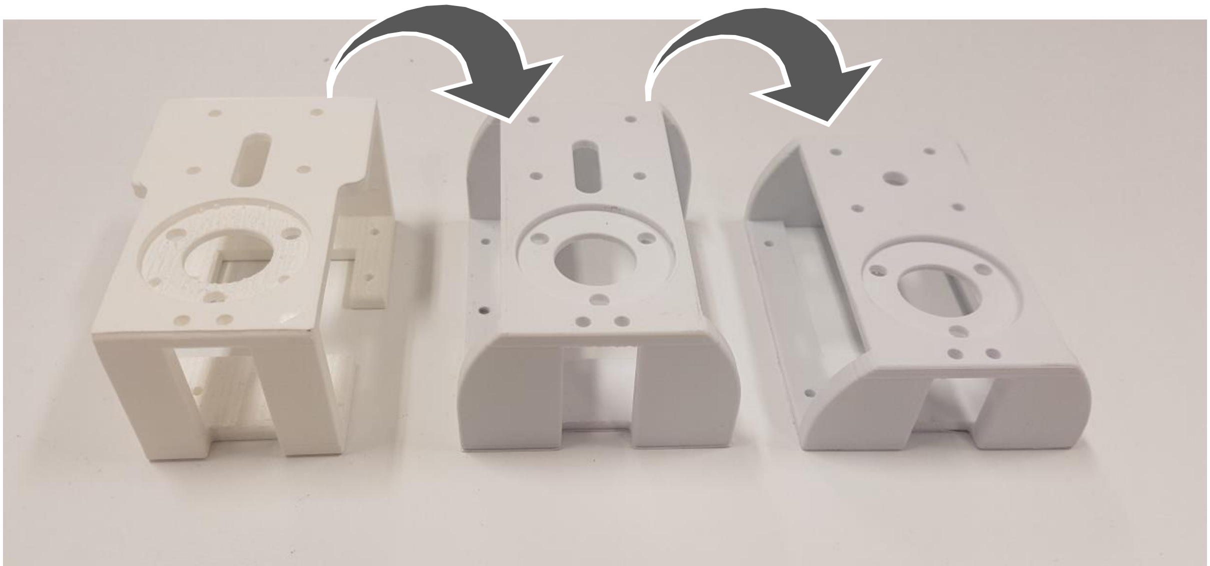


Figure 110: Example of the iterative design process for optimisation and minimisation

Distance measurement testing was performed again with the LiDAR scanner mounted to the master assembly (Figure 111). During this test, it was found the LiDAR scanner was successful at detecting the marker at distances of about 5 m, but at 8 m was only able to detect the wall above it. An important thing to note was that the slip ring was quite unstable, causing the whole LiDAR scanner mount assembly to have a slight offset angle away from the ground. Thus, as the assembly was rotated, the LiDAR was not directly parallel with the ground throughout its rotation cycle. When a slight force was applied to the LiDAR scanner to angle it down slightly, it was able to detect the marker at 8m. As a result, to try to reduce this problem, the master assembly was adjusted to reduce the height as much as possible; unfortunately, this did not fix the problem.

Another important complication found during this test was that the laser diode module was not directly parallel with the beam of the LiDAR scanner, thus did not represent the true position where the LiDAR scanner was measuring. As a result, slight modifications were made to the laser diode module mount and LiDAR scanner mount, to include small slots so that the mounting angle could be adjusted (Figure 112), such that the laser angle could be calibrated with the LiDAR scanner beam. After printing the new parts, the laser was able to be calibrated correctly with the position of the LiDAR scanner sensing direction.

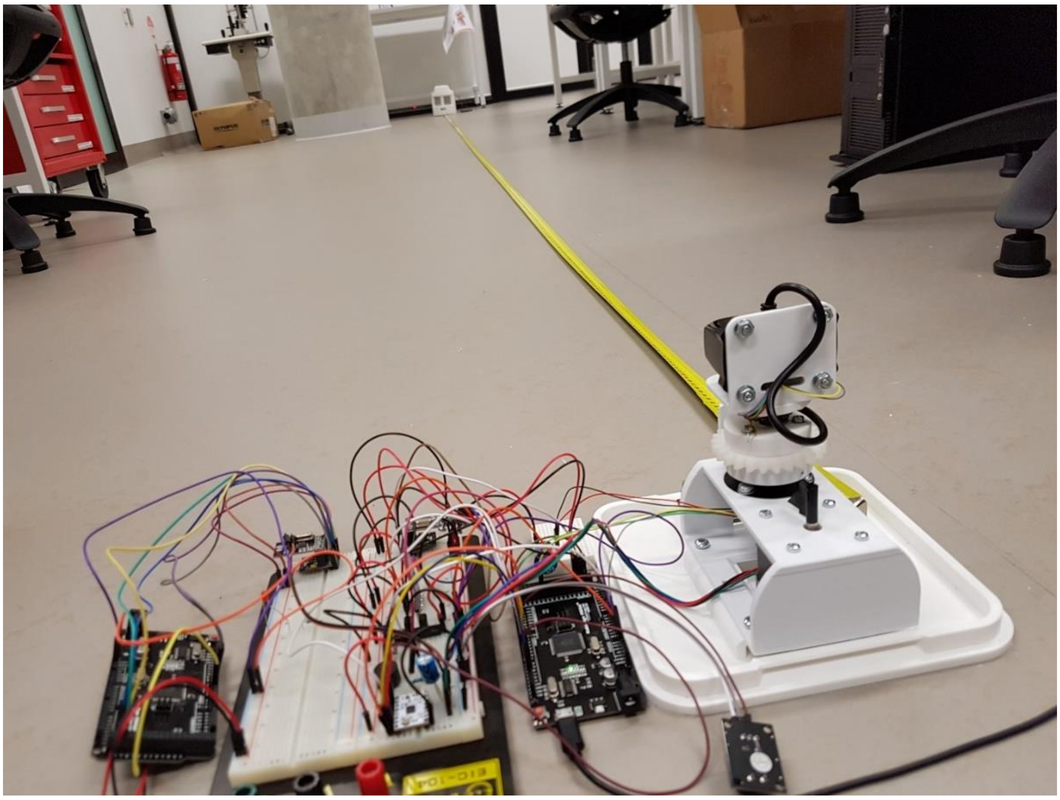


Figure 111: Testing distance measurement whilst LiDAR mounted to master assembly

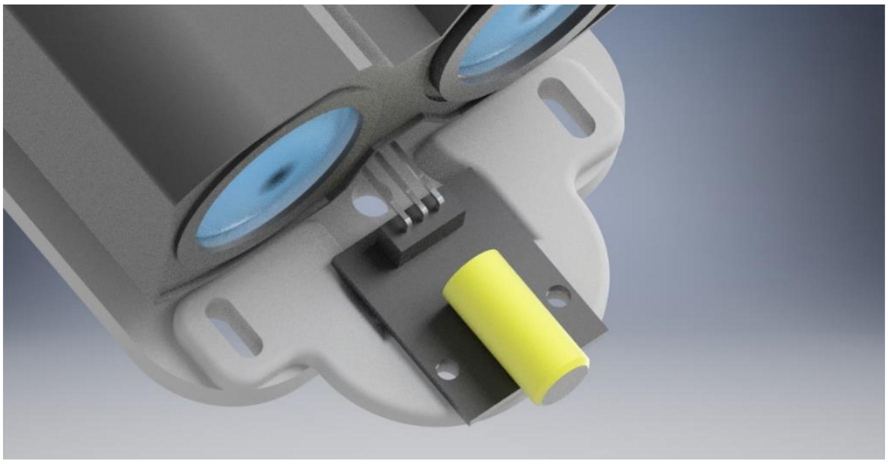


Figure 112: Slotted mount for the laser diode module such that the beam could be calibrated with the LiDAR scanner

Initial testing of the snap-fit enclosure lid deemed unsuccessful. The the initial design for the lid (Figure 113a) was intended to have the LiDAR scanner assembly separated and fastened once the lid was in place. The process was too complex with the little room available and would require wasted time removing and fastening the assembly each time the lid was to be taken off. As a result, modifications were made to increase the centre hole (Figure 113b) to allow the lid to be simply snapped into place without having to dissemble the LiDAR scanner assembly. Further optimisations were completed after this design to increase the hole diameter furthermore, as it was found that pulling the lid off would sometimes knock the laser diode module. Through repetition of knocking the laser diode module, it consequently caused it to break off the PCB breakout. Luckily it was easy enough to resolder back into place.

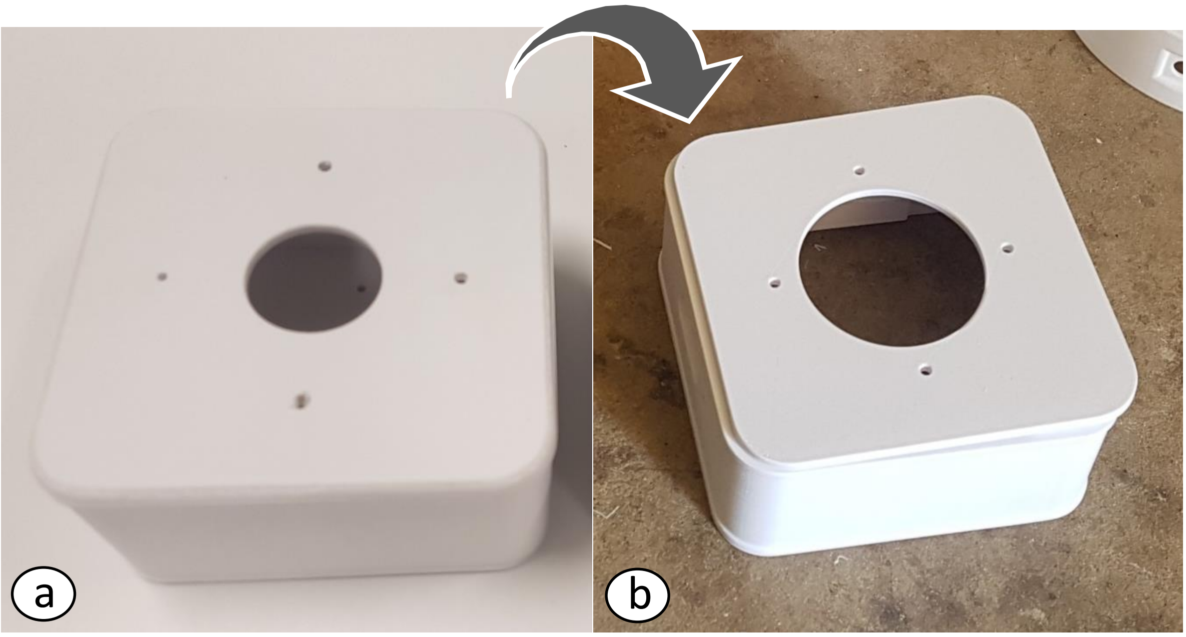


Figure 113: Increasing enclosure lid opening for LiDAR scanner

The last test for the master enclosure was to ensure that the clear CCTV dome was able to be mounted to the enclosure lid and that there was no interference with the LiDAR scanner assembly. The dome was mounted to the master component lid (Figure 114). Initially, it was found that if the outer fasteners on the LiDAR scanner mount faced outwards, there was interference due to the nut; inversing the fastener resolved this. However, the main issue was the LiDAR scanner wiring protruding from the top interfering with the dome. To resolve this issue, the LiDAR scanner mount was modified, such that the wiring could be restrained within the 3D printed part Figure 115.

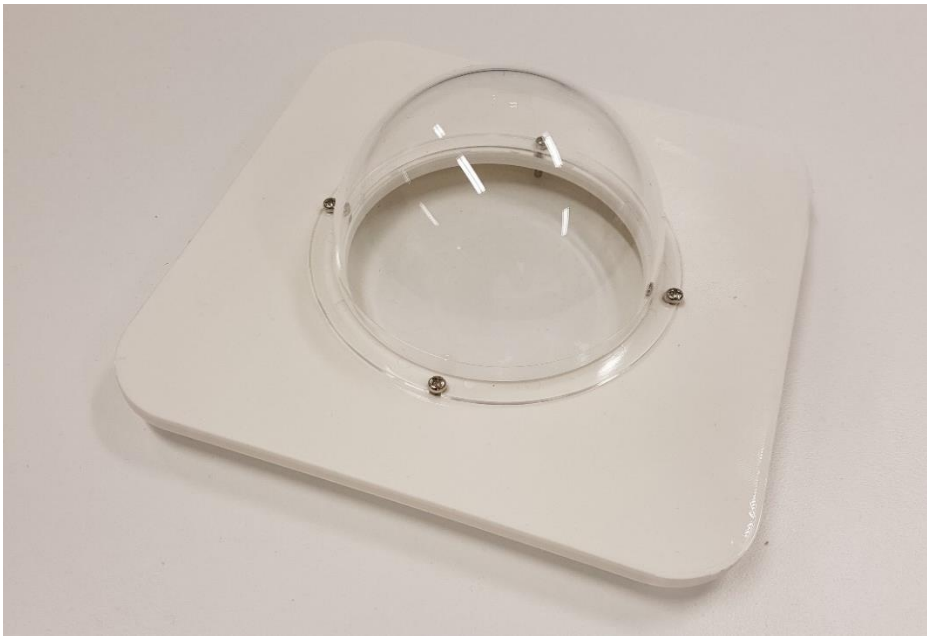


Figure 114: Clear CCTV dome mounted to the master enclosure lid

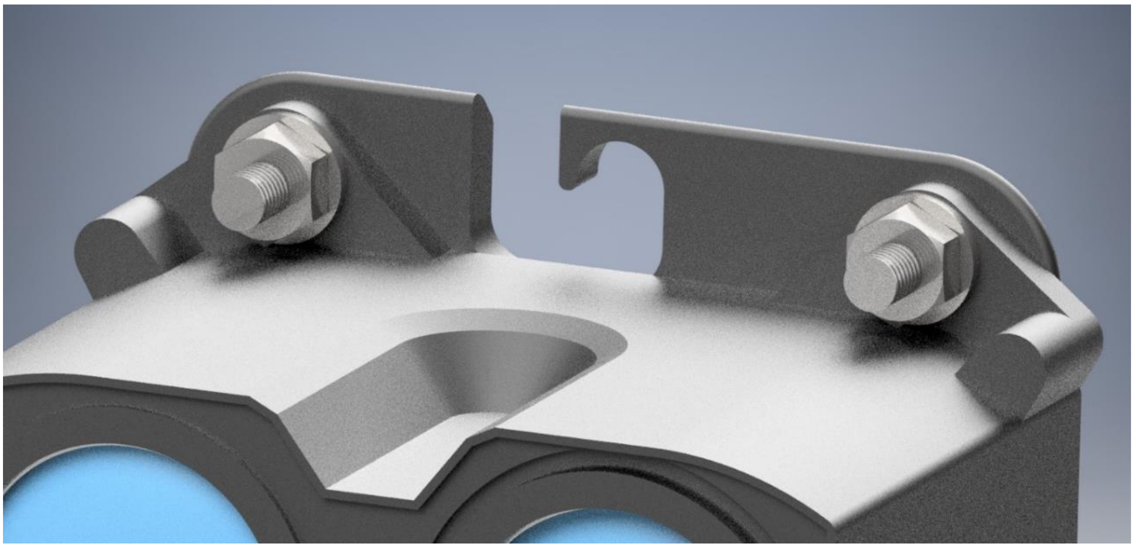


Figure 115: LiDAR scanner mount modified to restrain the wiring from interfering with the CCTV dome

3.7.5. System Integration

It was imperative to begin testing interactions between master and slave as whole assemblies. The initial phase of this stage of development consisted of partial integration of the systems with not all components connected (Figure 116).

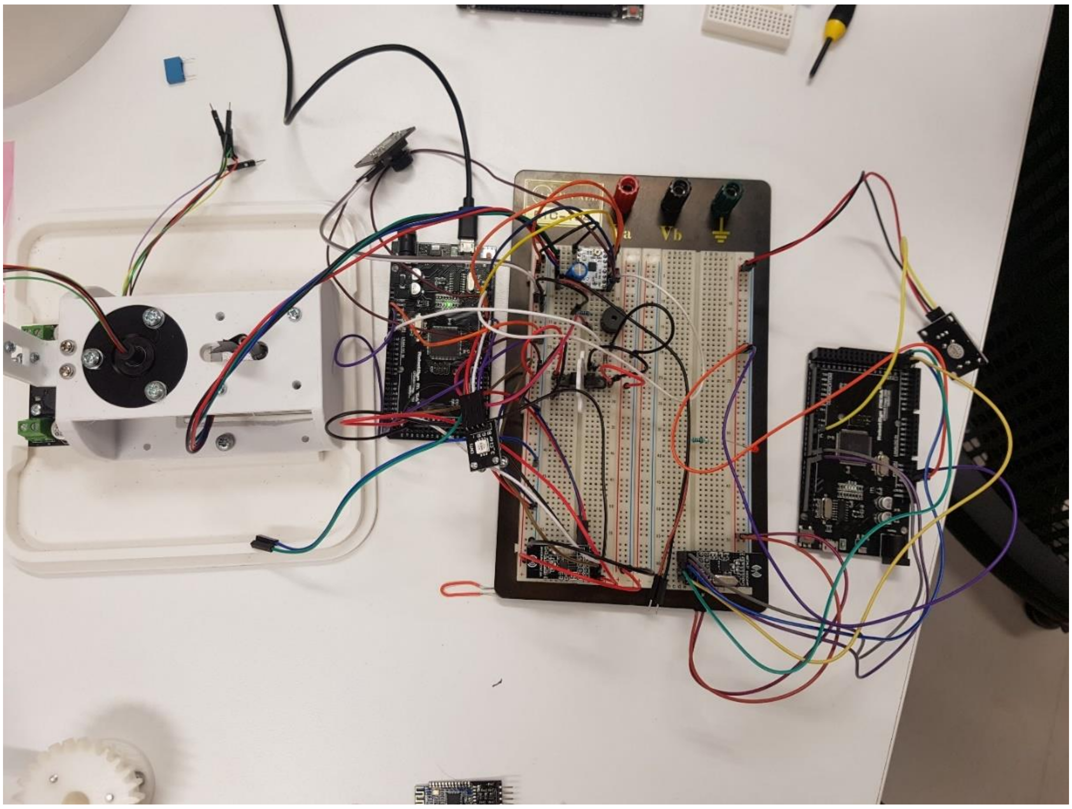


Figure 116: Initial integrated system interactions between master and slave

Progressive development and testing finally delivered whole system integration and interaction between master and slave component. The final wiring and component assemblies are shown in Figure 117. The primary objective was to ensure all components were successfully working properly, with communication between master and slave component. At this point, all hardware was working correctly. However, the nRF24L01+ modules were not completely reliable, as many packets of data would fail sending during testing. To try to resolve this issue, some high-powered modules were ordered, which contained a large antenna. A more detailed description of the complications of the nRF24L01+ module is discussed in Section 3.7.10.

With wiring of the master and slave component complete, it was important to draw up an electronic schematic, which could be referred to in the instance any wires would come loose or for when wiring the remaining slave components. The wiring schematics for both master and slave devices are presented in Figure I- 1 and Figure I- 2 in Appendix I: I.1, respectively.

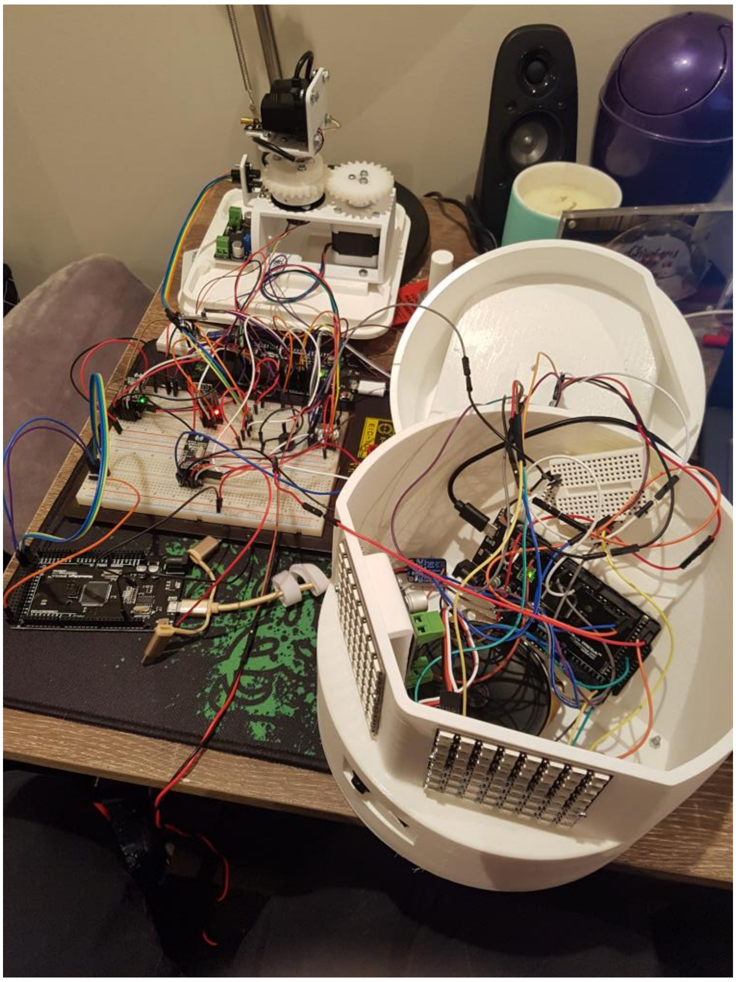


Figure 117: Integration of slave and master components

3.7.6. Optimising Design Aesthetics

With the most-part of the integrated slave and master components working thus far (nRF24L01+ module complication was to be resolved later), the next stage of development could be undertaken. With this proof-of-concept prototype having commercial appeal, it was deemed appropriate to paint the device enclosures, as the white made the devices seem incomplete and less appealing. The process of removing all the 3D printed supports on all devices was a long and tedious process. After this was complete, the parts were sanded back using course, medium and then fine sandpaper (Figure 118). Ultra-cover 2X satin spray paint in canyon black (Rust-Oleum, Vernon Hills, Illinois, USA) was used to paint the parts. It was not until after painting that it was realised that a satin clear-coat was available from the same company. As a result, no clear-coat was used when spray painting the slave components. The parts were then positioned for painting, with an initial light coat (Figure 119). Three successive coats were applied in 20-minute intervals for the top half, bottom half (after being flipped) and the inner components. Figure 120 shows an image during progression of painting.

Figure 118: Preparing the 3D printed parts for painting by through removal of supports and sanding

Figure 119: Painting set-up and application of initial light coat

Figure 120: Progressive painting of the enclosures

The final painted 3D printed parts (Figure 121) were left to cure for three days to allow the paint to adhere to the plastic correctly.

Figure 121: Final painted 3D printed components

One slave component was left so that testing could still be completed during the painting and curing process. This also allowed a visual comparison of the painted and non-painted marker to be made (Figure 122). The black painted marker was deemed far more visually aesthetic.

Figure 122: Comparison of a black painted slave enclosure next to an unpainted white one

3.7.7. Prototype Board and Wiring Development/Optimisation

The previous stages of the development of the master and component had entailed using solderless breadboards and jumper leads to connect electronic hardware together. This was a useful technique when establishing connections and testing hardware integration. The next stage of development entailed incorporating the electronic components with the prototype board. In doing this, the unorganised and tangled wires (Figure 117) were optimised by minimising wired connections, as well as providing more secure connections to the Arduino microcontroller.

A preliminary layout for positioning of the master hardware to the prototype board was completed (Figure 123). The process to fit all electronic hardware onto the small prototype board was difficult, as several lines of holes were interconnected in the centre of the prototype, as well as a VCC and GND interconnected line across the top and bottom (Figure 123). To further increase the difficulty, the components needed to be positioned such that there would be no interference with mechanical parts, such as the Bluetooth module interfering with the rotating gear assembly or the large capacitor interfering with the stepper and slip ring mount.

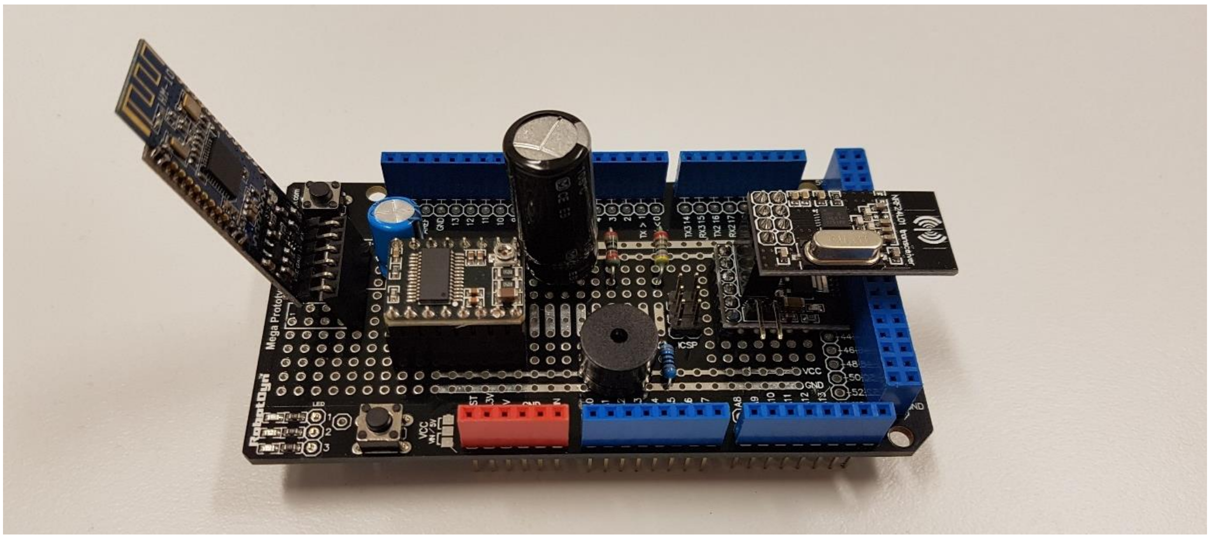


Figure 123: Preliminary hardware positioning for the master prototype board

At this point in time, it was determined that the current nRF24L01+ modules were either faulty or did not have enough power to produce reliable and consistent transmission between devices (further discussed in Section 3.7.10). A high-powered module was ordered, but whilst in transit, the prototype board needed to be completed. Therefore, using an image obtained online (Figure 124), an approximation had to be made, where it could be estimated that the new module PCB up to the base of the antenna would be approximately 2.5 times larger.

Figure 124: Current nRF24L01+ module (bottom) compared against a high powered module (top) (Tangient LLC, 2018)

Thus, the nRF24L01+ socket adaptor was rotated and capacitor for the LiDAR scanner shifted to a new location (Figure 125). It was imperative that foresight was used to estimate where the antenna would be positioned in the master enclosure such that it would not be interfering with any other components. With this configuration, the antenna would be able to be positioned between the transmissive photo-interrupter and enclosure wall near the buttons.

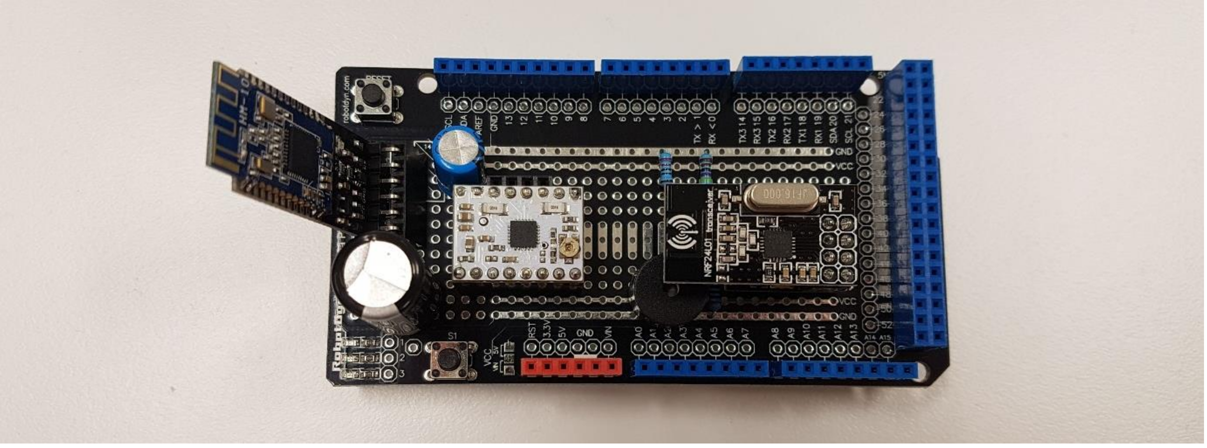


Figure 125: Final hardware positioning for the master prototype board

A similar iterative process was completed for establishing hardware positioning for the slave component with preliminary positioning first completed (Figure 126). An image was not taken of the final assembly before soldering the components and wires to the board. Therefore, the final hardware configuration of the slave component prototype board with soldered components has been shown (Figure 127).

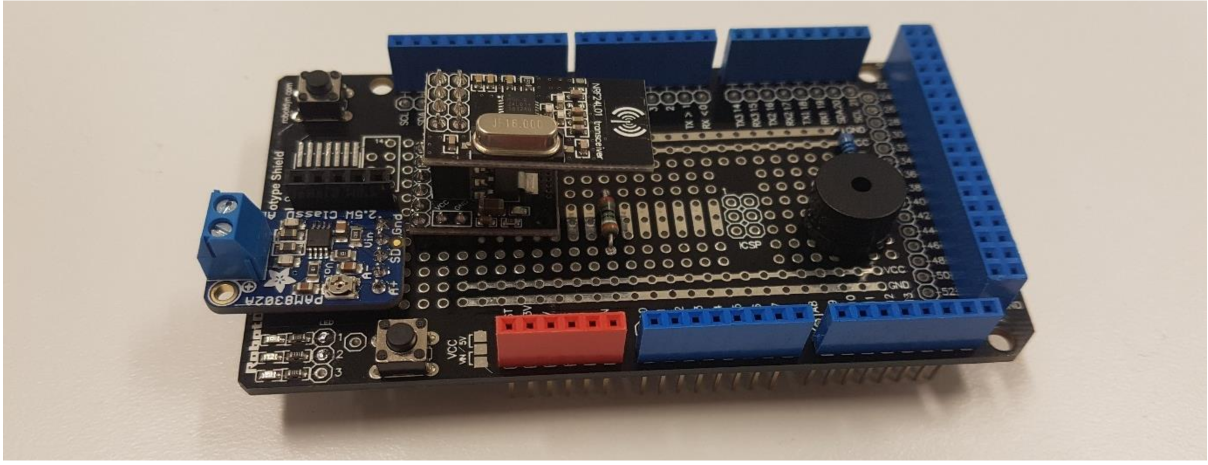


Figure 126: Preliminary hardware configuration for the slave prototype board

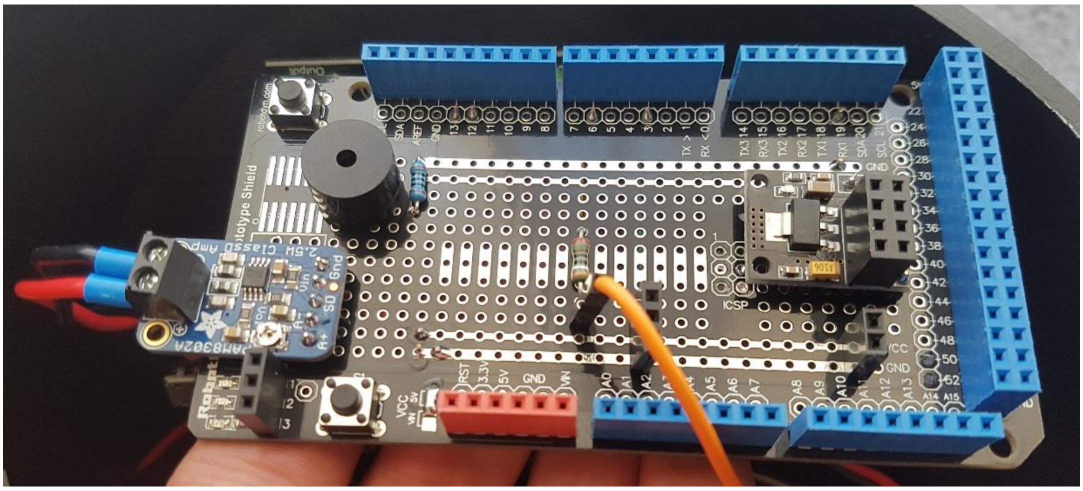


Figure 127: Final hardware configuration for the slave prototype board

With the hardware component positioning having been established, it was important to develop a schematic that could be used as a guide when soldering wires and components to the board. This was imperative so that wires were not incorrectly connected, and components not positioned incorrectly. The prototype board wiring schematics generated and followed for master and slave component are presented in Figure I- 3 and Figure I- 4 in Appendix I: I.2, respectively.

Single-core wires were used to connect the pins together as recommended by Engineering Technical Services at Flinders University. The stepwise process of soldering the wires for the master component prototype board (Figure 128) was difficult and very time consuming. Some pins for the stepper motor driver carrier and capacitors was first soldered (Figure 128a), followed with the remaining pins and the nRF24L01+ module (Figure 128b) and then the Bluetooth module (Figure 128c). Further additions had to be made for altering the positioning for the laser diode module and transmissive photointerrupter wires (Figure 128d), which was implemented after determining some complications when integrating wiring with the master component mechanical design assembly.

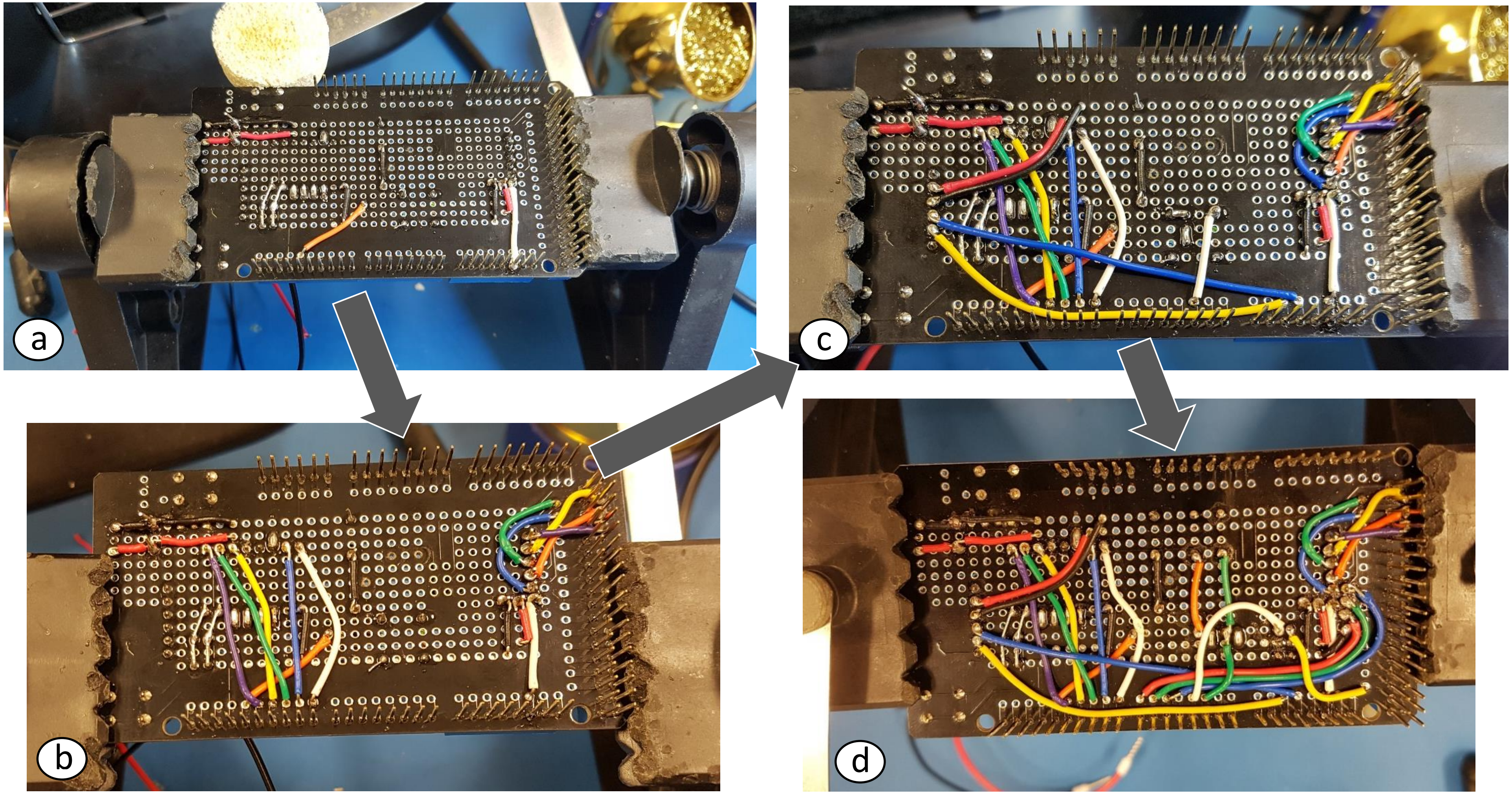


Figure 128: Stepwise process for wiring the master

With less hardware components integrated into the slave component, the resulting prototype board was simpler than the master (Figure 129).

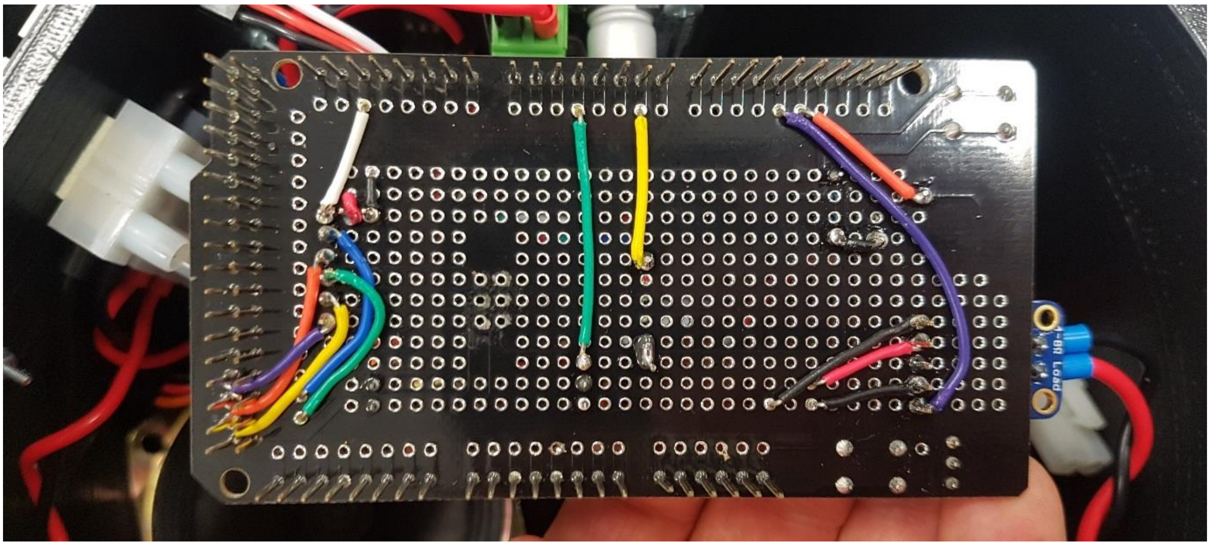


Figure 129: Slave component prototype wiring configuration

The connections between Arduino and hardware was accomplished by cutting wire and crimping the corresponding headers. Similarly, larger connections such as into the DC-DC step down buck converters, audio amplifiers and terminal blocks were crimped using crimping wire Ferrules to minimise the possibility of short-circuiting positive and negative wires together. The use of terminal blocks allowed separation of the circuit, so the Arduino and LED matrix display could be powered separately. Wires were also grouped together to clean the wiring configuration. Figure 130 shows the slave component with solderless breadboard and jumper wires (Figure 130b), compared with another slave with prototype shield board and managed wires (Figure 130a). The wiring was clearly improved and optimised using a prototype shield and managing wires. The master component after integrating the prototype shield and wires is also shown in Figure 131.

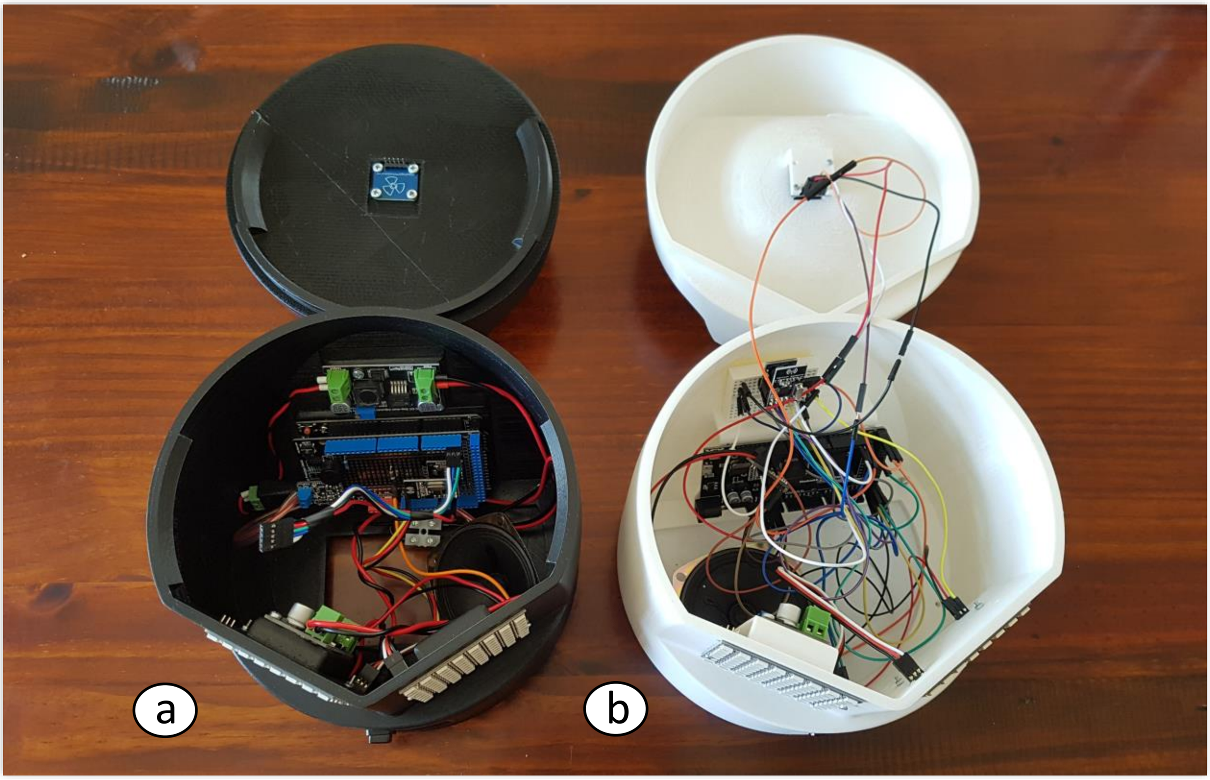


Figure 130: Optimised and clean wiring configuration by using a prototype shield board and grouping wires together

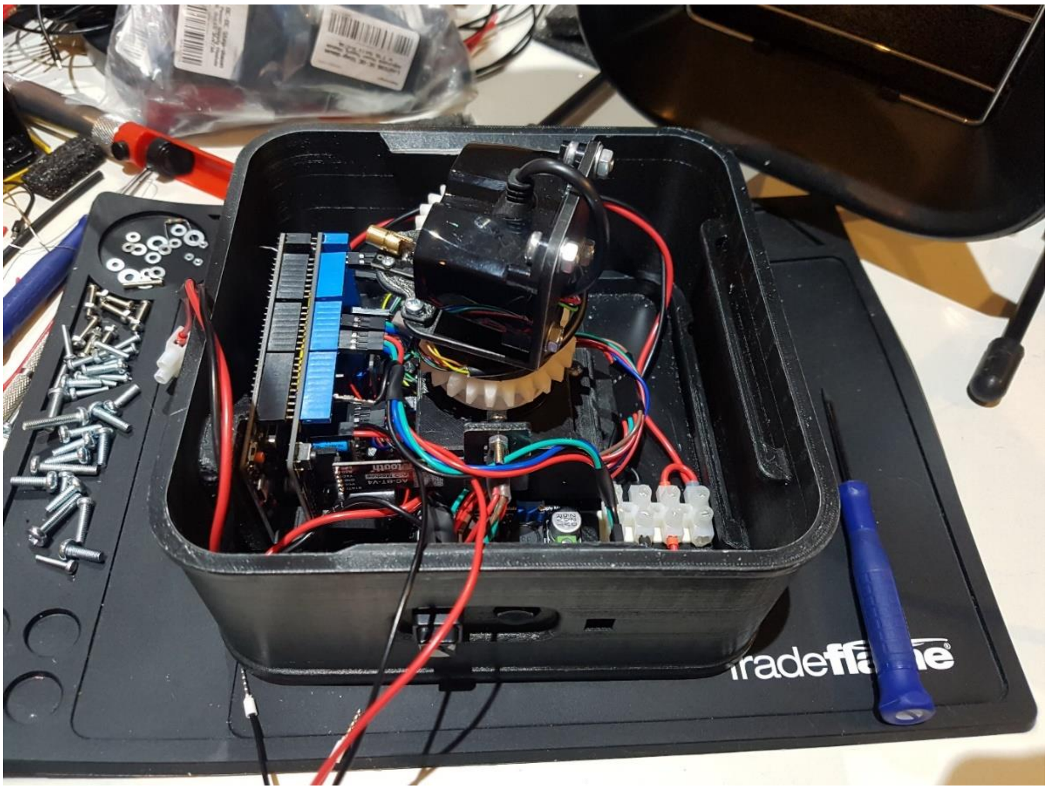


Figure 131: Master component after integrating the prototype shield and wires

3.7.7.1. Master Wiring Complications

Several complications arose regarding the master component after implementing the prototype shield onto the Arduino Mega and connecting all wires together.

3.7.7.1.1. Transmissive Photo-Interrupter Blocking Protrusion

The first notable complication was that the wires restricted the rotation of the LiDAR scanner assembly, as the protrusion that interrupts the photo-interrupter was blocking movement (Figure 132).

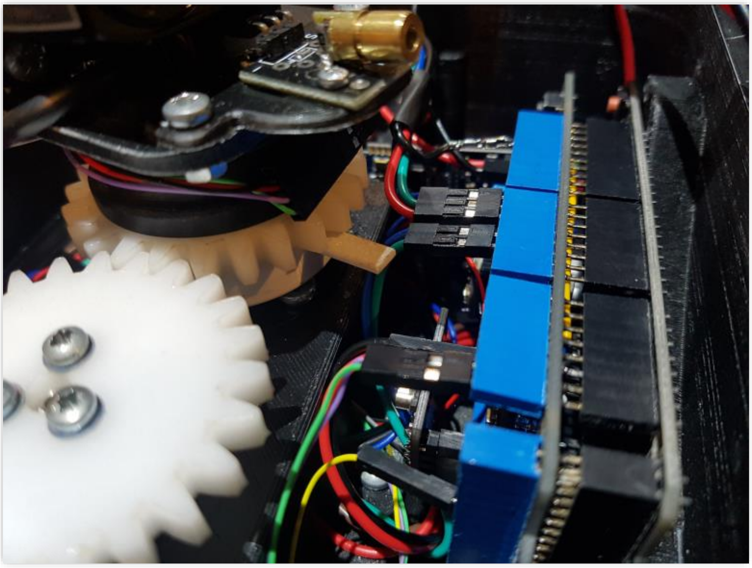


Figure 132: Wiring blocking rotation of the LiDAR scanner assembly

To resolve this issue, some headers were integrated into the centre of the prototype board, soldering wires between these and the Arduino pin headers. The wiring after completing this was shown in Figure 128d. After this refinement, the LiDAR scanner assembly was able to rotate freely, with no wires restricting movement (Figure 133).

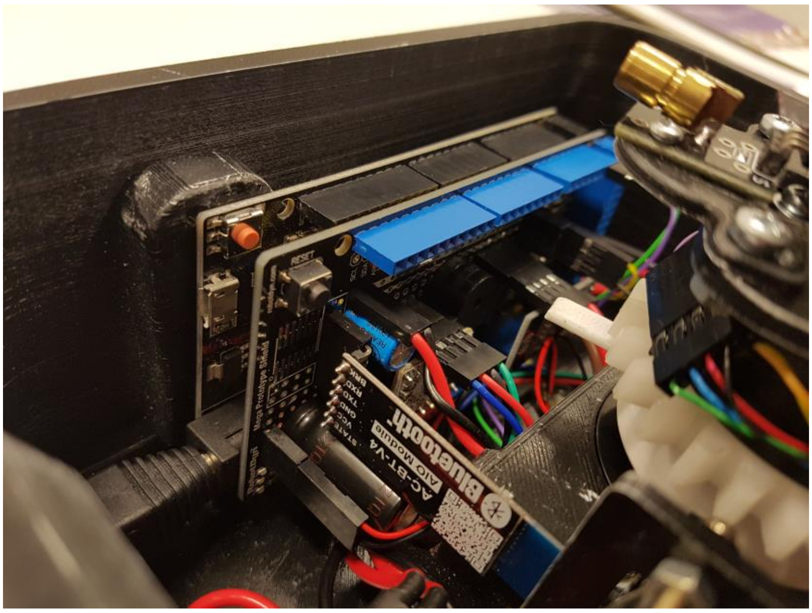


Figure 133: Integrating headers within the centre of the prototype board provided free rotation of the LiDAR scanner assembly

3.7.7.1.2. Male Type DC Power Barrel Integration Complications

Another major complication occured when inserting the male DC power barrel into the Arduino to provide it with power. The component was too long for the current mounted position of the Arduino. Even after unmounting and shifting as far as possible to the right, the DC power barrel still did not have enough room for wires to protrude the barrel and not interfere with the enclosure. The Bluetooth module and the large capacitor for the LiDAR scanner interfered with the stepper and slip ring 3D printed mount, so the Arduino could not be shifted any further to the right (Figure 134). A temporary fix was accomplished by cutting out a section of the 3D printed mount (Figure 135). This allowed movement of the Arduino to allow the wires to be connected to the DC barrel without interference of the enclosure. This problem may have been avoided if the DC power barrel had been CAD modelled into the master assembly; identifying the problem before 3D printing the components. However, it was not determined that the DC power barrel could be used to power the Arduino until later in the development process.

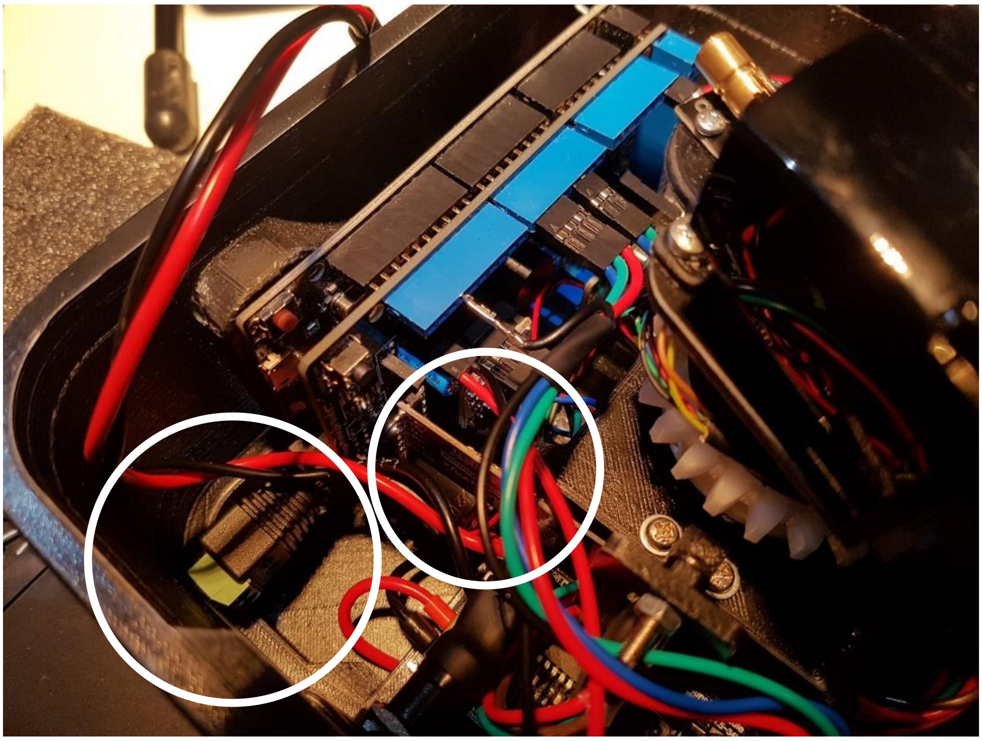


Figure 134: Several interference complications with the master component

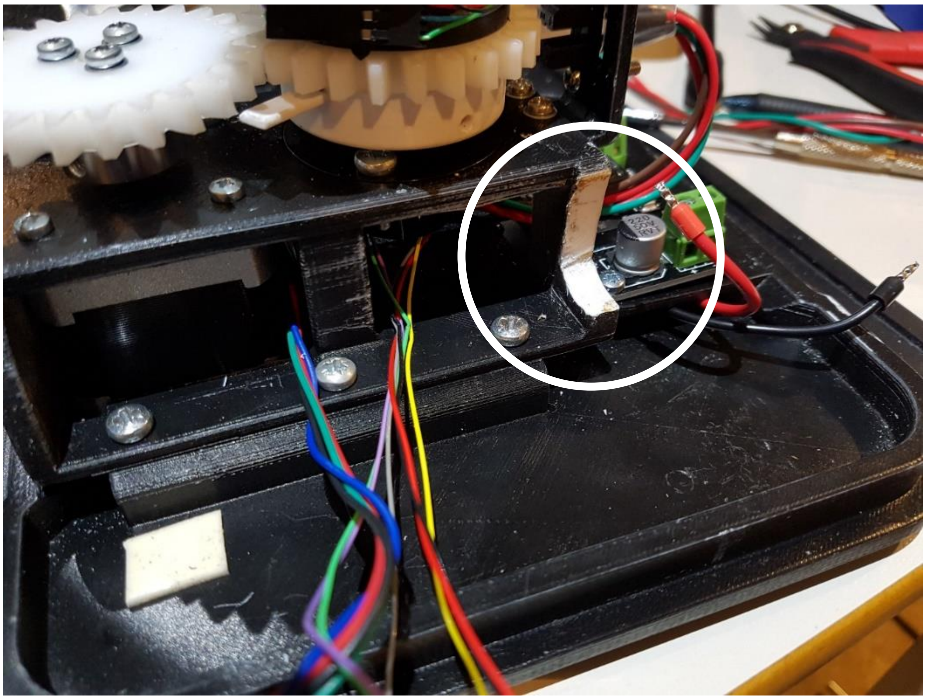


Figure 135: Temporary fix to resolve the interference complications

Some changes were made to the master design to accommodate for the DC power barrel being integrated into the device. The first design change was shifting the Arduino mount to the right to allow for more room (Figure 136). By completing this first design change, it therefore provided room for the Bluetooth module to be flat, parallel with the prototype board inside of perpendicular, so the headers on this module were changed. The final design change was adjusting the stepper and slip ring mount to provide room for the large capacitor (Figure 137).

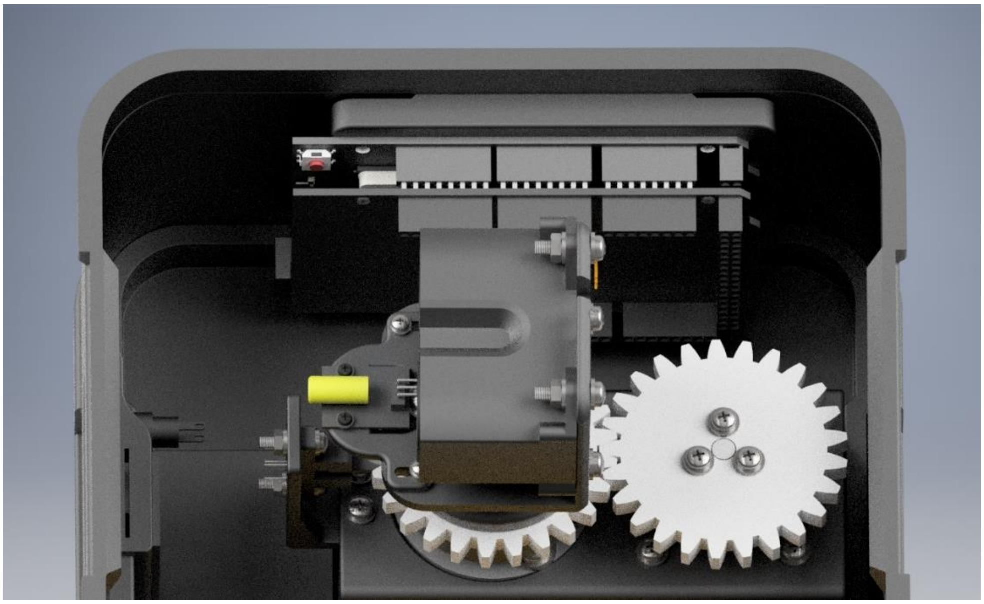


Figure 136: Shifting the Arduino mount to accommodate for the male DC power barrel

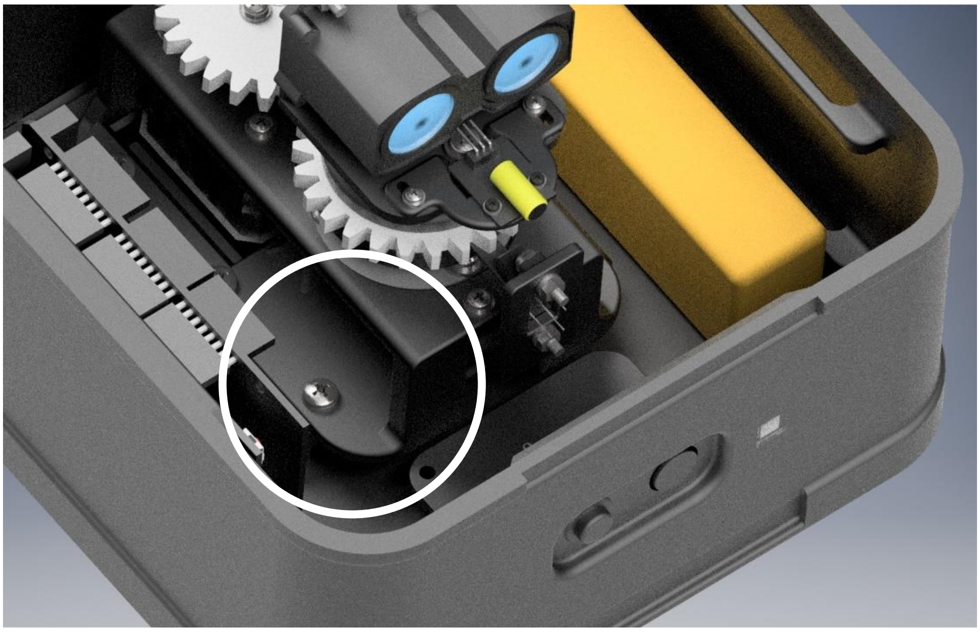


Figure 137: Stepper and slip ring mount design change to accommodate for the large capacitor

3.7.8. Additional Complications & Optimisations

3.7.8.1. Snap-fit Complications

Complications arose with the snap-fit design of the master component. The idea was for a user to be able to remove the lid without tools, so the battery could be changed/recharged with simplicity and ease. Initially, the snap-fit design was a 45° tapered protrusion and counterpart. The snap-fit design was found to be difficult to remove, so it was filed away. This significantly improved ease-of-use, but after multiple uses of snapping the parts together and removing, the part eventually broke, with the lid (Figure 138) and the wall (Figure 139). This was because the 3D printer had printed a shell, so the inner part was hollow. Therefore, filing back the part removed the shell and expose the hollow inside, significantly reducing the strength at this point.

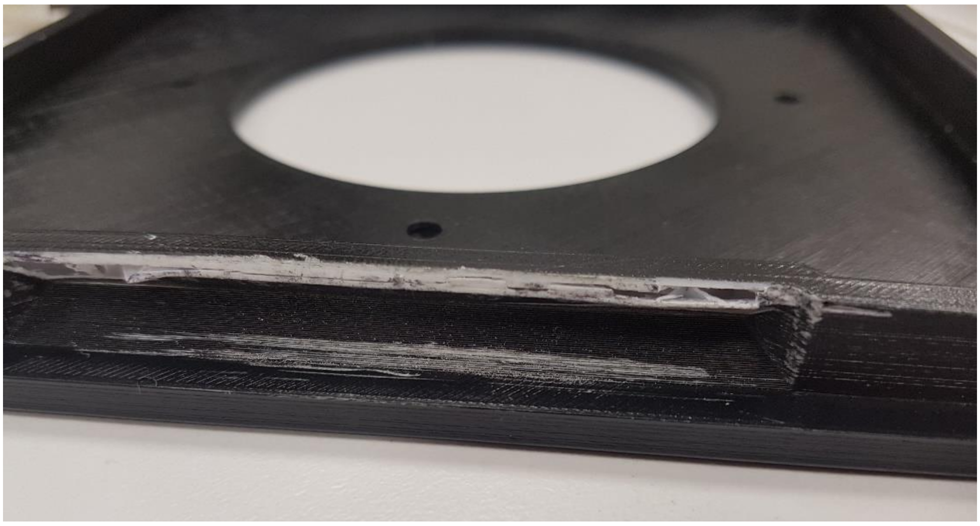


Figure 138: Snap-fit lid failure after many uses

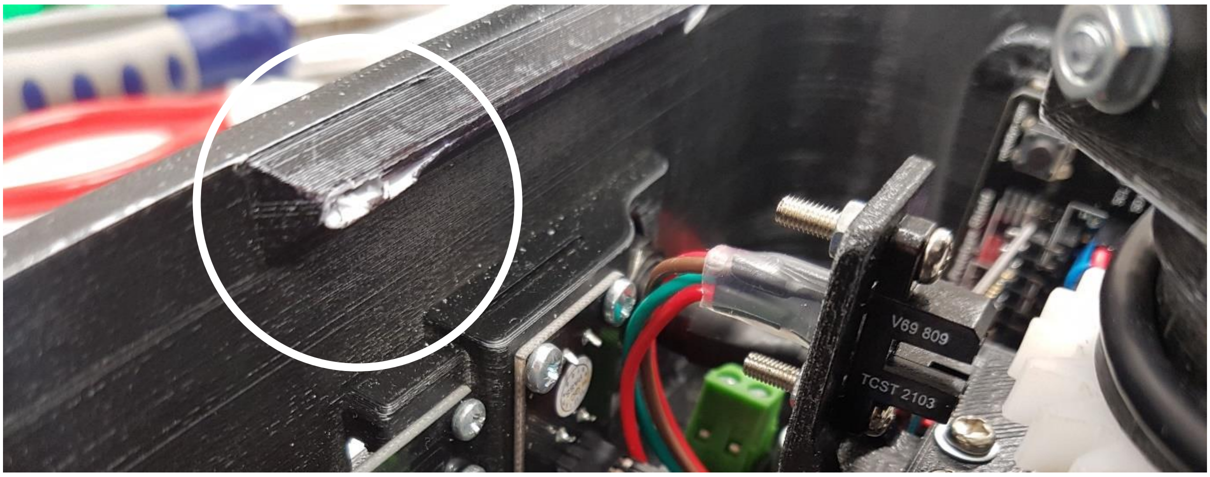


Figure 139: Snap-fit enclosure wall failure after many uses

To resolve this complication, the old snap-fit assembly design (Figure 140a) was optimised to reduce the amount of protrusion into its counterpart (Figure 140b). Testing the new design, the snap-fit was far easier to remove. However, after painting, the process of removing the lid was quite difficult again, which could have been due to an increased friction coefficient. The only way of removing the lid is by using a flat-head screwdriver. Using one tool to change/recharge the battery was within the specification, but it was most ideal to not have to use a tool. This was unfortunate, but due to time constraints, further optimisation could not be accomplished.

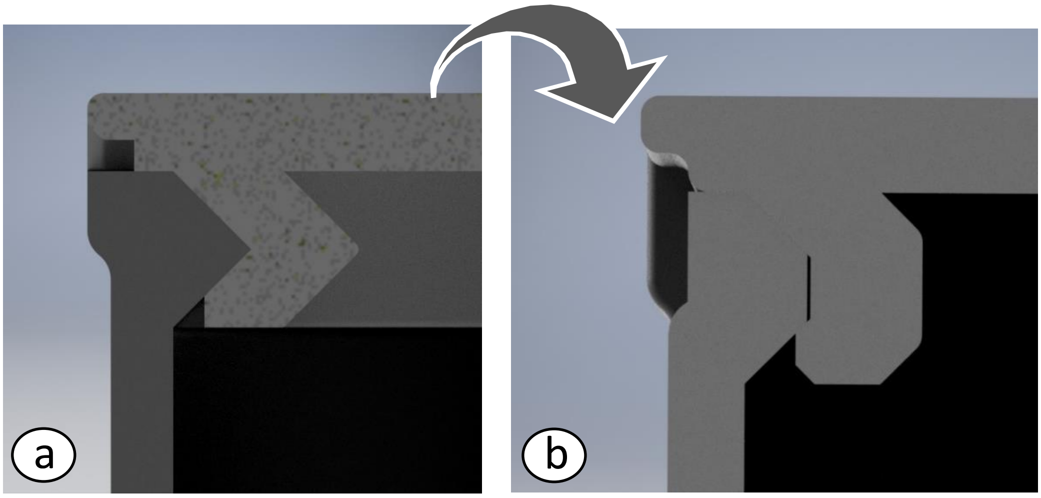


Figure 140: Snap-fit design refinement and optimisation

3.7.8.1.1. Brittle Fracture of Components

Whilst removing the master enclosure lid, the transmissive photo-interrupter mount was knocked and subsequently snapped. A temporary fix was to super-glue the piece back together. To avoid the occurrence from happening again, the part was optimised slightly to include some chamfers on either side of the edges to provide more structural support, with the two designs shown in Figure 141.

Figure 141: Transmissive photo-interrupter mount design optimisation

3.7.9. 3D Printing Complications

3D printing provided a useful manufacturing method for rapid prototyping, with ease of redesigning and optimising parts in CAD. Some limitations to this method was that there were instances where the 3D printer would not print correctly. For example, warping occurred at the edges of the slave enclosure base (Figure 142), master enclosure base (Figure 143) and master enclosure wall (Figure 144). There were many more failures that occurred, but these are just a few of the instances. Parts that were larger seemed to have a higher chance of failure. The only resolution to this was to remove the print and try again.

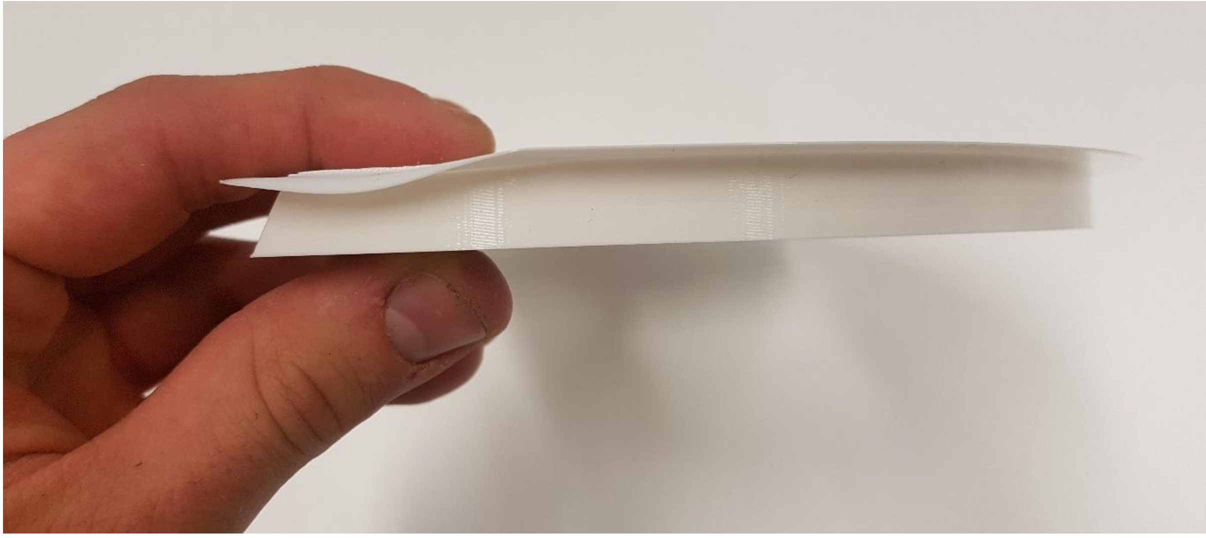


Figure 142: 3D printing failure for the slave enclosure base

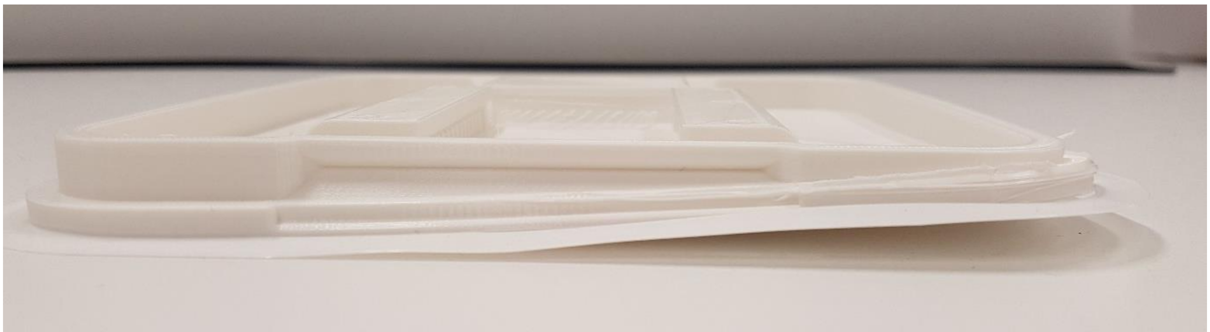


Figure 143: 3D printing failure for the master enclosure base

Figure 144: 3D printing failure for the master enclosure

3.7.10. Intra-System Communication Complications

A large portion of hardware testing consisted of electronics mounted to a breadboard, with the nRF24L01+ modules positioned directly next to one another. Sending and receiving messages in these tests provided reliable and consistent results, with fast transmission times and no packet loss of data or failures in sending. Integration of hardware components into the corresponding master and slave enclosures found that the module was not as reliable and consistent as it had been in the previous tests. With the module inside 3D printed enclosures, confined next to neighbouring hardware componentry, packet loss had increased significantly, with many failures trying to send data to the receiving device shown in the serial output from the Arduino IDE (Figure 145).

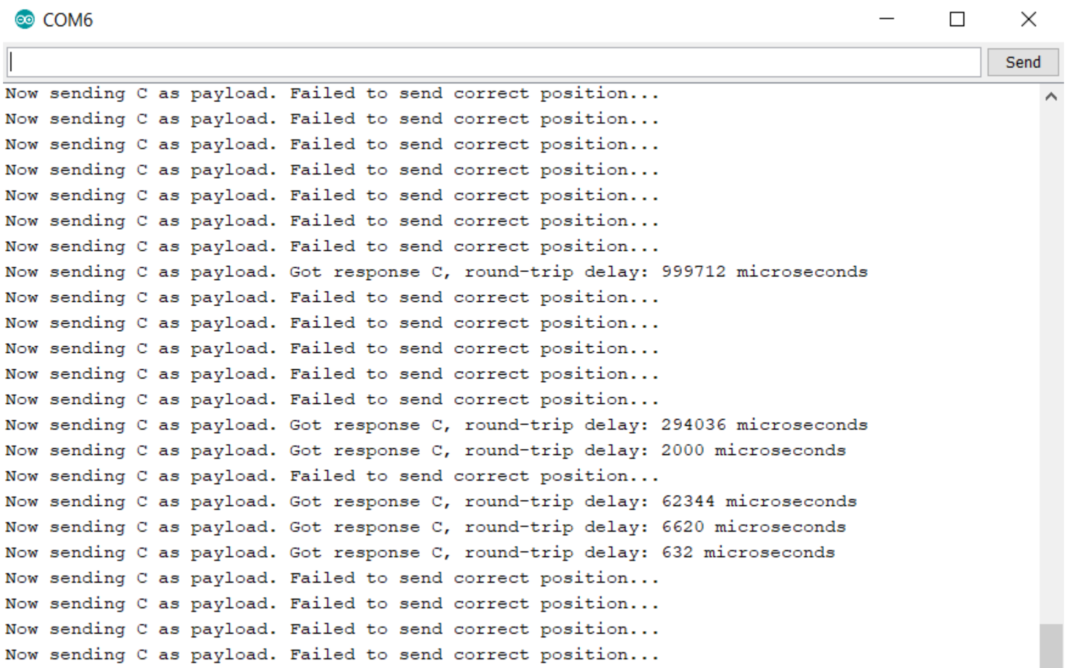


Figure 145: nRF24L01+ data packet loss when integrated inside the enclosures

The inconsistency would be a major problem with a test that needed to be standardised; a marker trying to communicate to another marker to tell it to activate but failing to send the message would add time to the test as well as inconsistency of receiving the message. Thus, some high power nRF24L01+ modules were ordered, which contained a large antenna, power amplifier (PA) for transmission and low noise amplifier (LNA) for receiving. There were significant problems regarding shipping and transit; the package never arrived. By the time it was realised that the package had been lost, a significant amount of time had passed. Another order was made with another company, but this company took a very long time to process the order; as a result, this order had to be cancelled and the parts were ordered from another company. The parts were eventually received, with the new module shown in isolation (Figure 146a) and next to the normal module (Figure 146b). The normal module has the antenna embedded in the PCB at the end, which is significantly smaller than the antenna on the new module; therefore, it was expected the high-power module should outperform the regular one.

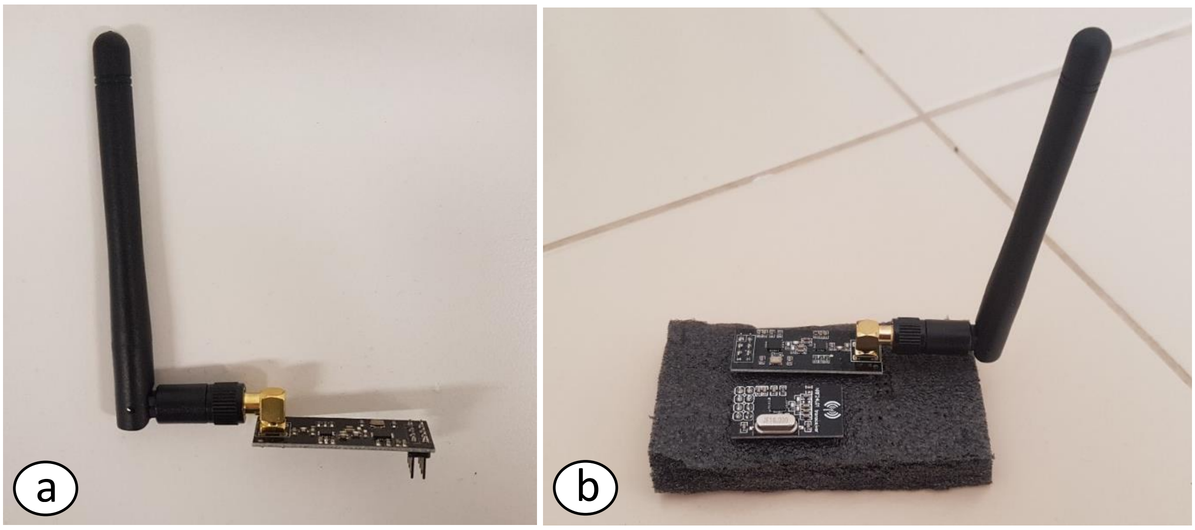


Figure 146: High power nRF24L01+ module with power amplifier and low noise amplifier and comparison with the normal module (right)

The same code was tested again with the new modules; every message was being received successfully, with a miniscule 0.00054 second round-trip delay (Figure 147).

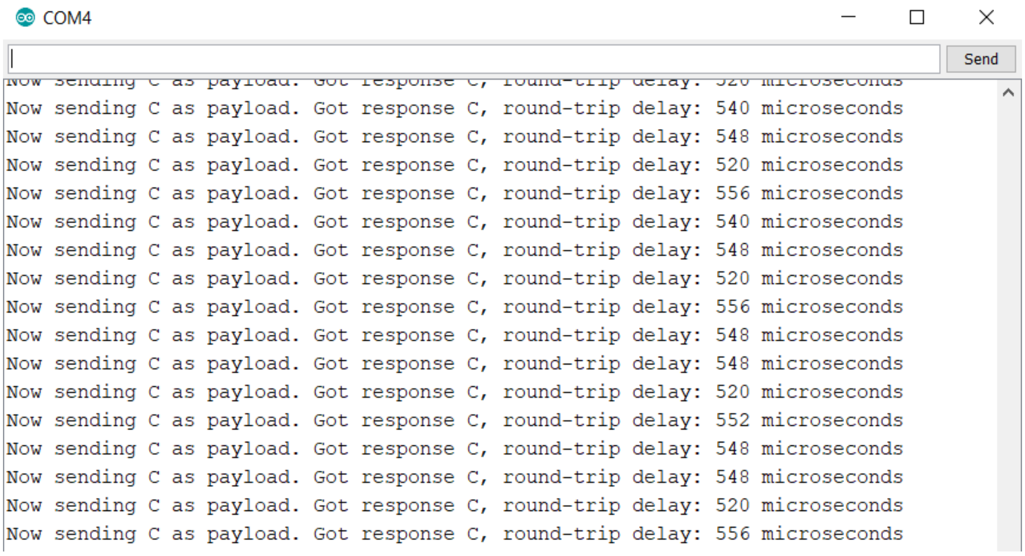


Figure 147: Successfully sending and receiving data consistently via the high power nRF24L01+ module

Unfortunately, the significant complications in shipping and transit of the parts, due to shear bad luck with the companies selected, meant that the parts were received with only one week remaining before thesis submission. This time had been dedicated towards thesis write-up; as a result, only one day was able to be put aside to try to integrate the intra-system communication and debug the program. Prior to receiving the modules, the program had to be written such that the code was based off the tests conducted with the low-power modules.

3.7.11. Software Development

The software development was conducted throughout the entirety of the embodiment stage of the project. Programming had to be completed to test various hardware components and write the main programs for the system. The general methodology followed for developing any form of program in this project first consisted of understanding the problem and what needed to be achieved. Once this was established, pseudocode was generally written as a form of program design, which would present the functions and steps required to be completed to achieve the aims of the program; this was written for human reading. Following this, main function bodies were established and then the specific coding lines in between were added and implemented. The code would then be refined and optimised using an iterative process. This section aims to provide the development of primary key functions required to be implemented for the successful development of the system.

3.7.11.1. nRF24L01+ Module Data Pipe Addresses

According to the datasheet, the nRF24L01+ module is able to connect up to six other nRF24L01+ modules simultaneously, a feature known as MultiCeiver™ (Nordic Semiconductor Inc, 2008). To accomplish this, each nRF24L01+ module needed its own data pipe address. Nordic Semiconductor recommended a 5-byte (equivalent to 40-bit) address, with the primary receiver (PRX) having a unique 5-byte address (data pipe 0) and the remaining transmitters sharing the four most significant bytes in their data pipe address, with the lowest significant byte (LSB) being unique. An example for a data pipe configuration set is shown in Figure 148.

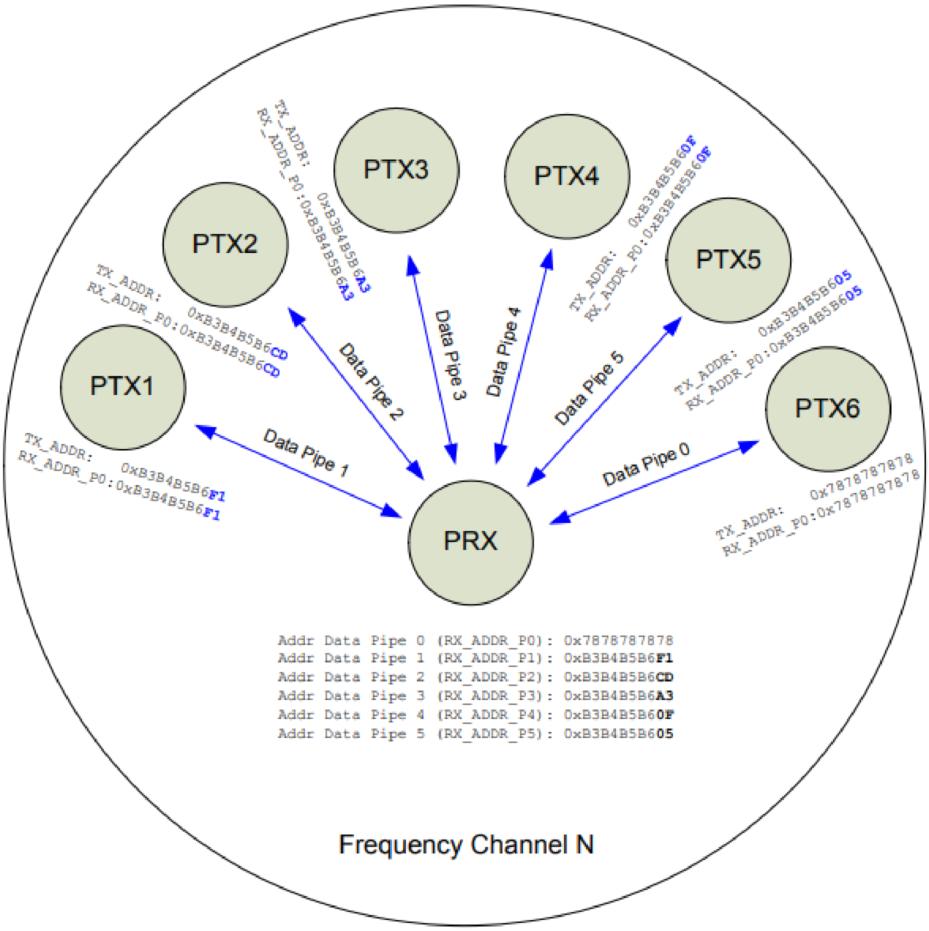


Figure 148: Pipe addressing example using MultiCeiver™ and Enhanced ShockBurst™ (Nordic Semiconductor Inc, 2008)

An arbitrary 40-bit pipe address was given to the master component and first slave. All slaves shared the four most significant bytes of the first slaves address and were given a unique LSB c1, c2, c3, c4, c5 and c6, respectively (Table 30). The '0x' at the start of the value specifies that the value is given as a hexadecimal.

Table 30: Configured addresses for each nRF24L01+ module associated with the components in the system

Device	40-bit Pipe Address
Master	0x3e64b727ff
Slave #1	0x714e2b65c1
Slave #2	0x714e2b65c2
Slave #3	0x714e2b65c3
Slave #4	0x714e2b65c4
Slave #5	0x714e2b65c5
Slave #6	0x714e2b65c6

The 5-byte addresses could be broken down into their respective byte values (Figure 149). This figure helps visualise the concept that the LSB is the last two values; each value represents 4-bits, so two values equals 8-bits and there are 8-bits in one byte.

Device	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Master	0x3e	0x64	0xb7	0x27	0xff
Slave #1	0x71	0x4e	0x2b	0x65	**0xc1**
Slave #2	0x71	0x4e	0x2b	0x65	**0xc2**
Slave #3	0x71	0x4e	0x2b	0x65	**0xc3**
Slave #4	0x71	0x4e	0x2b	0x65	**0xc4**
Slave #5	0x71	0x4e	0x2b	0x65	**0xc5**
Slave #6	0x71	0x4e	0x2b	0x65	**0xc6**

Figure 149: Data pipe addresses for each component in the system

The datasheet specified that data from the RX data pipe 0 is shared with TX data pipe 0. Thus, as the master component needed to connect to all six markers, but also transmit data too, the method to ensuring that this successfully worked correctly was to open a writing pipe with the master data pipe address each time it needed to transmit data. Following this, a reading pipe on the slave #1 data pipe address was immediately opened, to receive any data transmitted from it. Figure 150The serial output of the nRF24L01+ configured values after setting the data address pipes (Figure 150) showed how the TX address (TX_ADDR) is shared with the RX address 0 (RX_ADDR_P0).

```
--------------- nRF24L01 Details ---------------
STATUS           = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P0-1     = 0x3e64b727ff 0x714e2b65c2
RX_ADDR_P2-5     = 0xc3 0xc4 0xc5 0xc6
TX_ADDR          = 0x3e64b727ff
RX_PW_P0-6       = 0x07 0x07 0x00 0x00 0x00 0x00
EN_AA            = 0x3f
EN_RXADDR        = 0x03
RF_CH            = 0x6c
RF_SETUP         = 0x07
CONFIG           = 0x0e
DYNPD/FEATURE    = 0x03 0x06
Data Rate        = 1MBPS
Model            = nRF24L01+
CRC Length       = 16 bits
PA Power         = PA_MAX
```

Figure 150: Confirming the data pipe addresses of the nRF24L01+ module

3.7.11.2. Dynamic Slave Data Pipe Addresses

The final program for master and slave was written such that the address for the slave component is dynamic in nature. That is, the same code is used for all slave devices without changing any values, such as specifying which slave number it is. The program was written so that no matter what order the slave components are turned on, their data pipe address will correspond to the order in which the devices were turned on. This was achieved by first turning on the master, which continues to broadcast the first data pipe address until a slave has connected. As the slave device is turned on, it listens on the master data pipe address and will see exactly which address to configure itself as, thus determining if it is Slave #1, Slave #2, Slave #3… etc. It then transmits this address back to the master so that it is aware that a slave device has connected to that pipe address and the connection process is complete.

3.7.11.3. Button Debouncing

Button bounce (or switch bounce) is a phenomenon experienced when pressing (activating it) or releasing (deactivating it). There is a period where the contacts of the button "bounce" as they do not make perfect contact, causing the state of the button to erratically alter between HIGH and LOW (1 and 0, respectively) (Neuroscience and Robotics Laboratory, 2006). The phenomenon was drawn in Adobe Illustrator for a visual representation of some example events that could occur (Figure 151).

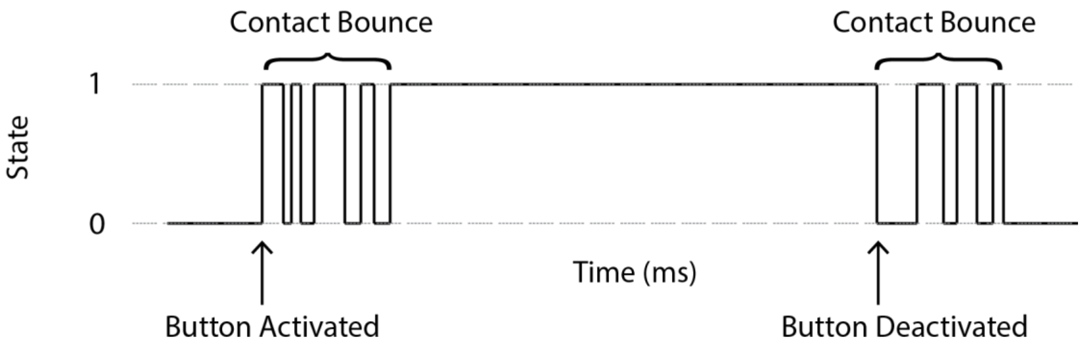


Figure 151: Button bouncing showing a period of contact bounce, which causes the button state to change erratically

Ideally, the button should read HIGH when activated and LOW when deactivated, providing a clean edge. Therefore, button debounce can be employed; the process of removing the bouncing effect. Button bounce can be handled in two ways; through software and through hardware. The software method was employed initially, with the thought that if it failed, then the hardware method could be used. It should be noted that the button was configured to read HIGH when pressed and LOW when released, however these states were reversed so that an interrupt could be used to wake the Arduino when it was put to sleep (this is discussed further in Section 3.7.11.8).

An interrupt can be used to make an Arduino complete a specific function when an event has occurred on a digital pin. This specific function is known as an interrupt service routine (ISR) (Arduino, 2018). In this instance, the ISR can be programmed to run successively after a change in state of the digital pin, so it knows when the button has been pressed and released. Referring to the Arduino reference page for attachInterrupt(), it was determined that the Arduino Mega is able to use digital pins 2, 3, 18, 19, 20 and 21 as an interrupt pin.

The button was connected to digital pin 3, attaching an interrupt to trigger when the pin changes value. Within the ISR, when the state of the button was LOW, a timer would be started. During testing, the time elapsed was printed to serial output every time the button state was HIGH. Through testing, it was determined that a real button press was generally above 50ms; any lower was a button debounce (Figure 152).

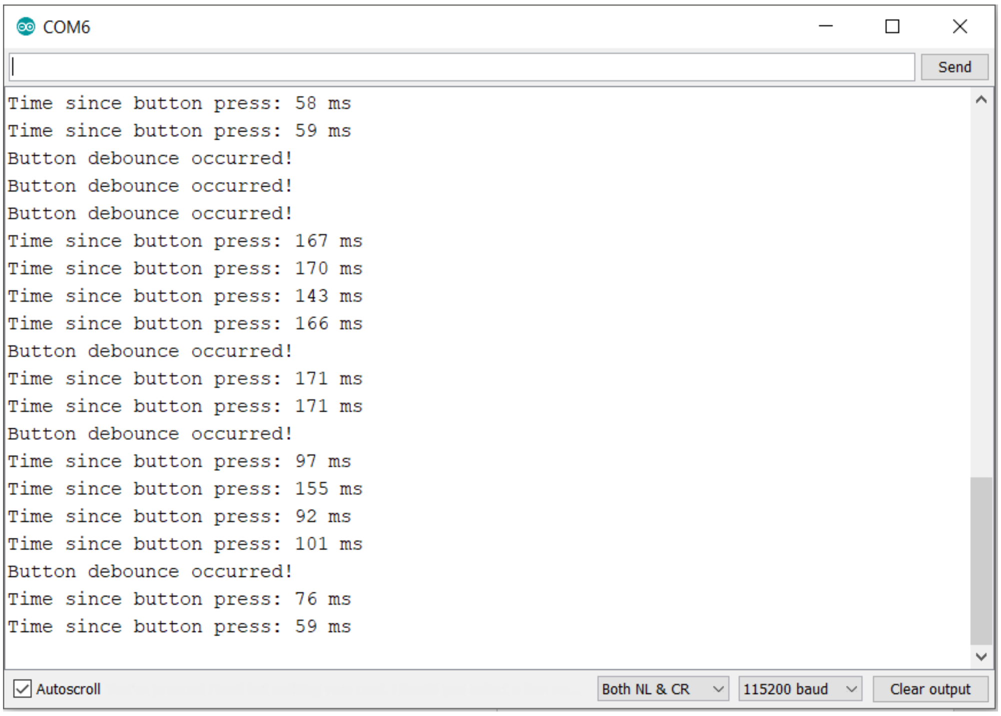


Figure 152: Testing for button bouncing

Thus, when the button state was HIGH, it checked if the time elapsed was greater than 50ms; if it was, then a global flag (in the form of a Boolean) was set to indicate that the button had been pressed. If the flag was true, then further code would be processed in the main program, sequentially resetting the flag once complete. By incorporating the global flag, it minimised the time spent in the ISR, which was recommended (Arduino, 2018). Any global variables used in an ISR should always be declared as volatile, so that the main program updates the variable correctly (Arduino, 2018). Thus, the flag was declared as volatile. Testing the program, the software method for debouncing the button worked perfectly as it needed to, thus, the hardware method was not attempted to be employed.

3.7.11.4. Programming the Bluetooth Module

Programming the Bluetooth module was simpler than expected. It was able to be programmed similarly to how the console serial was coded, because simply put, it was just another serial communication device. The system was able to communicate with a Dell Inspiron 15 5000 Series laptop (Dell Technologies, Austin, Texas, USA) that contained inbuilt Bluetooth, using a program called Bluetooth LE Lab by Ian Savchenko, which was accessible in the Microsoft store (https://www.microsoft.com/en-au/p/bluetooth-le-lab/9n6jd37gwzc8#activetab=pivot:overviewtab) or from GitHub (https://github.com/IanSavchenko/BleLab). The communication was successfully being transmitted and received by both ends (Figure 153). However, this software was slightly difficult to navigate.

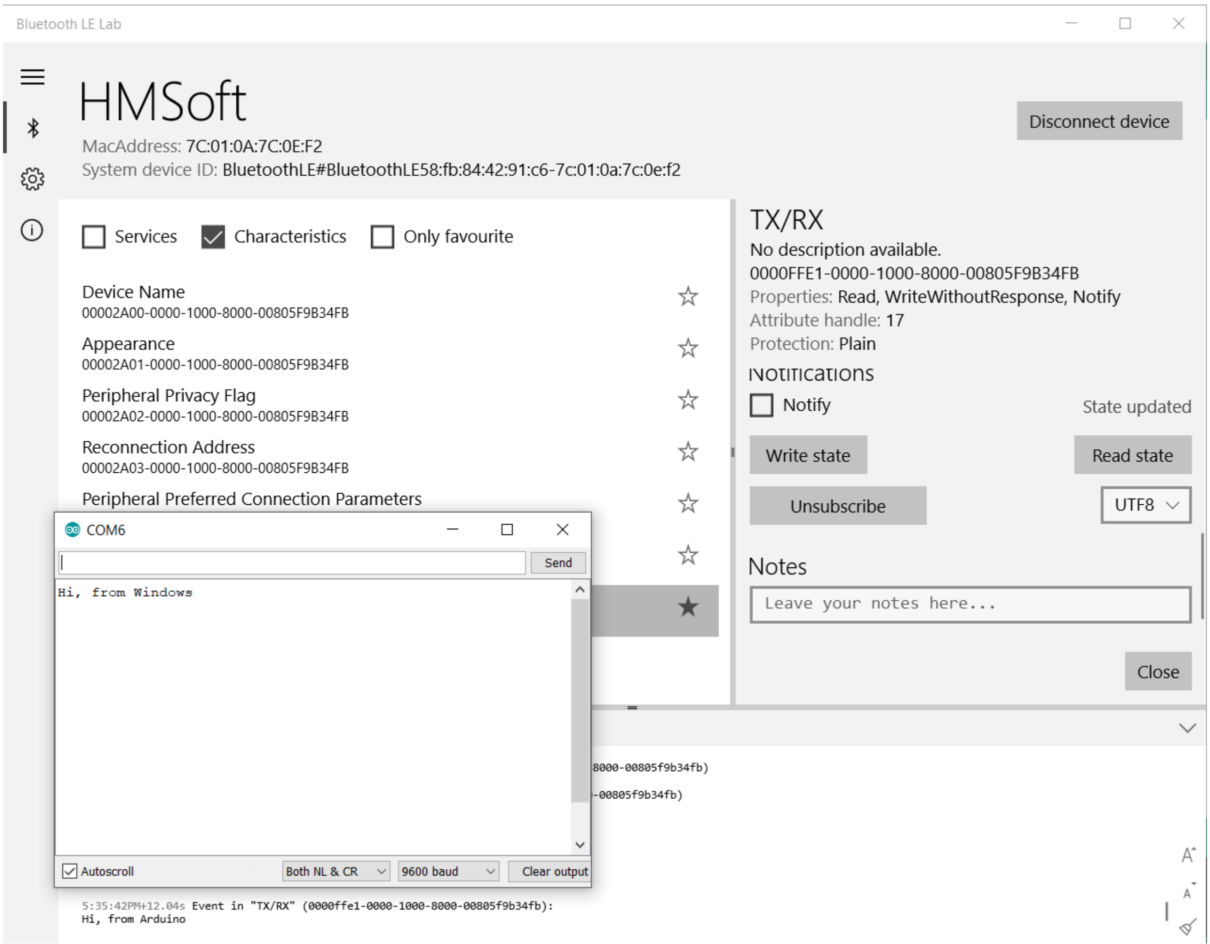


Figure 153: User/system communication through computer software

The system was also able to connect to a mobile phone. An Android Galaxy S7 (Google, Mountain View, California, USA) was used to pair with the system. Various applications were experimented with, but the author found Serial Bluetooth Terminal (Kai Morich, Hockenheim, Germany) an excellent application with a simple, yet effective user interface. This application can be downloaded from the Google Play Store for free. The system for now is called "HMSoft" (Figure 154a); but was to be changed later when a passcode can be successfully implemented. The reason for this was because a device named "Sports Agility Tester" may tempt nearby people to try to connect to the device and interfere with the system. In the options, it was imperative the CR+LF (carriage return + line feed) was enabled (Figure 154b) for receive and send.

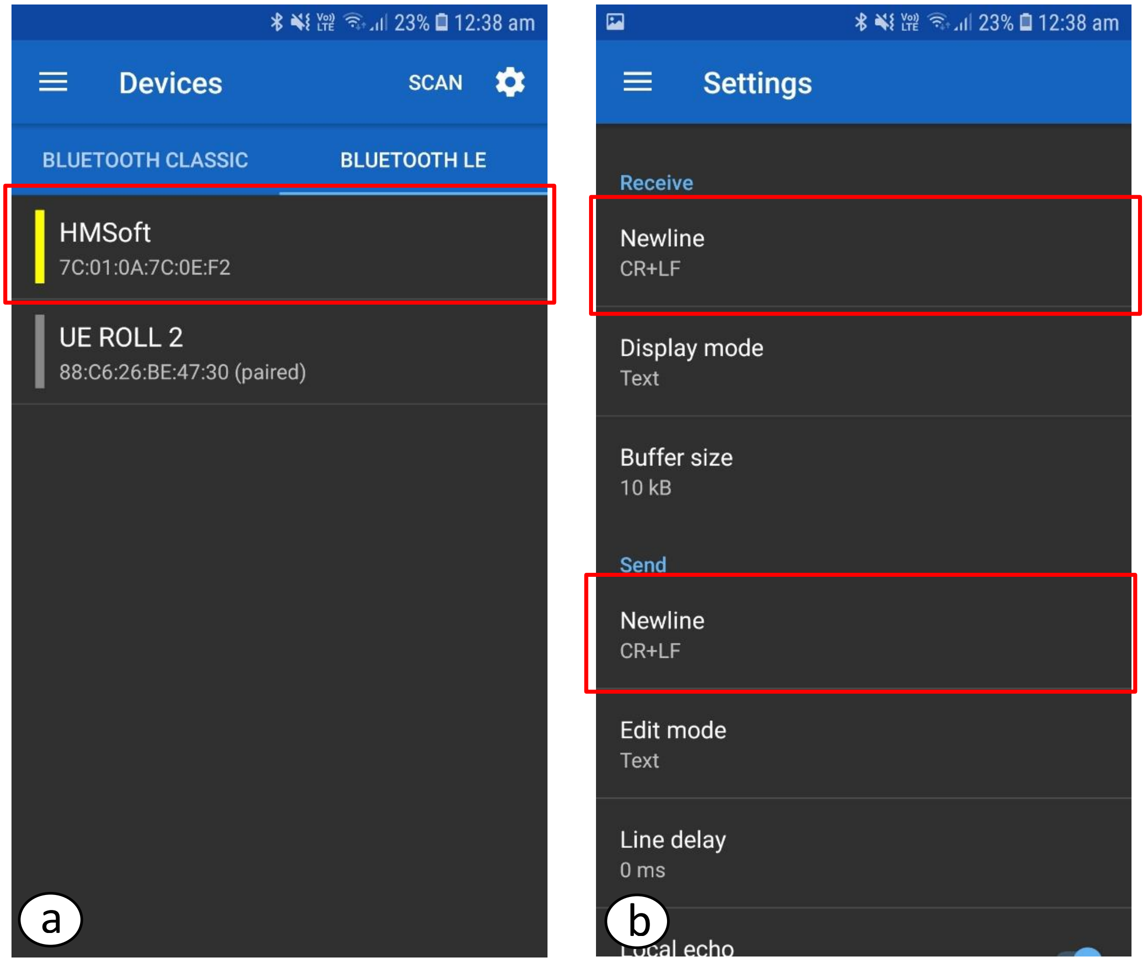


Figure 154: Configuring the settings for Serial Bluetooth Terminal application on Android smartphone

After writing up some code so that the system could receive data from the connected Bluetooth device and send information back, various commands were developed. Some examples included indicating when the next test was to begin, changing test difficulty, resetting the system, printing the results of the last test or indicating what number test the system was up to. Using the Android smartphone, commands were sent to the system which processed the commands and then sent back the corresponding information or confirmation message as feedback (Figure 155)

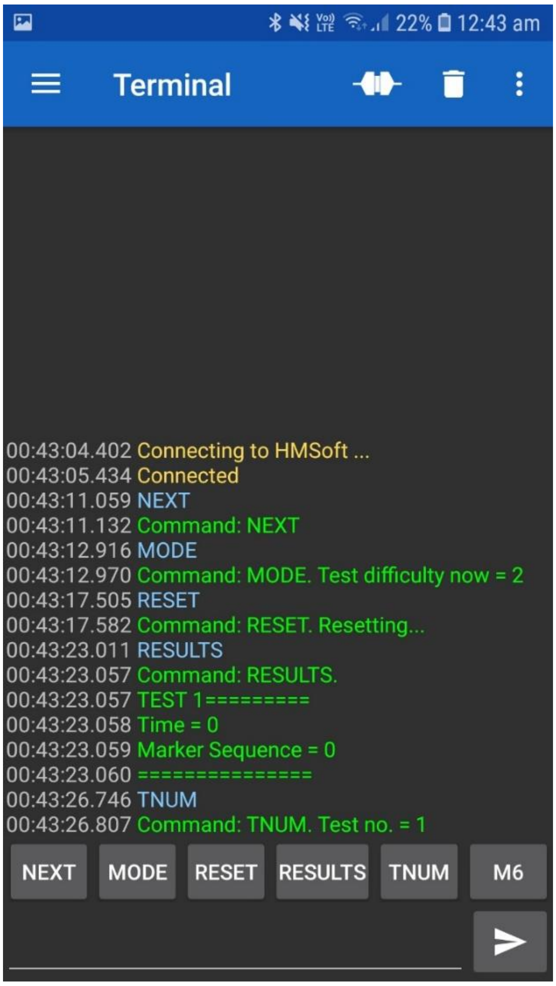


Figure 155: Various commands provided from the user to the system

3.7.11.5. Generating a Set of Possible Test Sequences

The term permutation has strong relevance in this project, which can be defined as a selection of elements where the order is important. It is important that the term permutation is not mixed up with the term combination, where order is not important.

The number of permutations can be determined for a given set of elements, n and total number of possibilities, r (Equation 7).

Equation 7: Calculating permutations

$$P(n, r) = \frac{n!}{(n-r)!}$$

In terms of this project, there are six markers (n = 6); there are six possible routes that can be ran (r = 6). Thus, filling the equation with the values:

$$P(6,6) = \frac{6!}{(6-6)!} = \mathbf{720}$$

Thus, with six markers, there is 720 unique sequence permutations. However, not all permutations have an equal running distance. To ensure fairness and standardisation of the test, it would be important that for each test completed, the same distance is completed. Thus, a user-friendly program was written to produce code that provides a set of possible test sequences (all with the same distances) that can be directly copy and pasted into the Arduino Master. The primary aim was that any one of these tests could be implemented in any given test. The code for the program is presented in Appendix J: J.2. It was determined early in the project that the markers would be set-up in a hexagon shape for the standardised version of the test (Figure F- 1 and Figure F- 2 in Appendix F: F.1). Thus, the program calculates all possible distances using basic geometry of a regular hexagon (Figure 156).

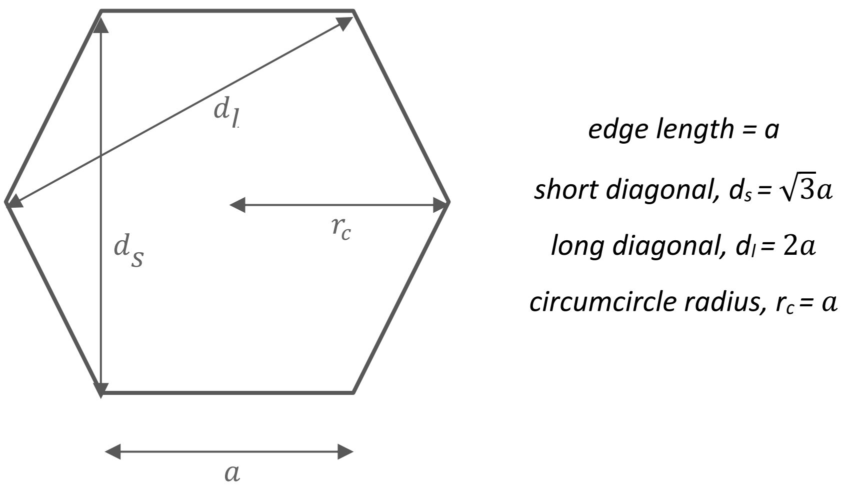


Figure 156: Basic geometry of a regular hexagon required to calculate test distance

In all sequence permutations, the run from the master to any of the slave devices is the same distance. Therefore, no matter what permutation, the edge length would always be added to the distance value.

The program was written so that a user specifies the possible marker values (e.g. 123456). For all user-input sections in the code, it ensures that the user enters the correct input before proceeding to the next step to stop the program from crashing. Once receiving the possible marker values, it then calculates all possible permutations and outputs the possible permutations to the console.

The program then asks the distance from the centre to any marker (the edge length). No matter what distance value is input into the program, the number of unique distance values is always going to be the same, with the same corresponding counts for each. The distance allows tangible calculations to be made, as well as the user can see respective distances for various test difficulties. When a distance input is received, a set of calculations are performed to determine the distance for each consecutive run within the permutation. This is achieved by separating the number value and determining where each marker is located in the sequence. The relative distances between each marker is stored and depending on the run, the corresponding distance is added to the total distance. The value is rounded to four decimal points so that counting the unique distances is achieved correctly. All distance values are output to the console.

The program then determines the corresponding unique distances using a set, which are containers that can store unique elements in a specific order. A counts vector is used to store the corresponding counts for each unique distance. The program then outputs the unique value counts for the unique distance values with an index next to each unique distance. The user enters the corresponding index value for the distance they desire, and the program then produces output for an array of uint32_t values (unsigned int) with all possible permutations corresponding to that unique distance. This array can be directly copied into the Arduino master program.

Upon inspection of the possible permutations, although having the same distances, there were a few with two successive instances of running to the adjacent marker. For example, the permutation 612354 contains 123, which contains two successive instances of running to an adjacent marker. Essentially, the athlete would have the marker to deactivate directly in front of them two times in a row without having to change direction. Thus, it was decided to adjust the program to ensure that any sequence that had two successive adjacent runs was to be removed. This was achieved by checking if the previous marker was a neighbouring marker, and if it was, then a counting variable would be incremented. If the subsequent marker was a neighbouring marker, then the count would be incremented again. This would then trigger the program to delete that permutation. Initially, the program was using arrays, however manipulation was deemed far easier to accomplish when replacing the arrays with vectors. After filtering the 720 permutations for any permutations that had two successive adjacent markers, 600 permutations remained.

Initially, it was decided that the test would finish when there was one ball remaining. This was because the athlete would know where they would need to run to automatically when realising there is one ball left; therefore, they would only run to five of the six markers. The reason for thinking the test should be conducted in this way, was that it was desirable to make the test as random as possible. However, upon further analysis of the literature review, it would be more beneficial to ensure that the athlete runs to all six markers. The reason for this, is because the main aim was that the system has the capability of testing all components of an athlete's agility. The literature showed that within the perceptual and decision-making factors of agility, there exists subcomponents of pattern recognition and knowledge of situations. An athlete who is truly agile, would remember key information, such as knowing exactly which markers they have already run to and what markers would be remaining to be deactivated. Thus, it made sense that a truly agile athlete would know exactly where to run when one marker is remaining to be deactivated, compared with someone who is less agile and may need to visually scan the area to find the last marker. This concept also comes under the knowledge of the situation. Thus, it was decided that the athlete must deactivate all markers.

Example output from the program is shown in Figure 157.

```
 -------------------
  Unique value counts
 -------------------
1  | 67.3205   = 12
2  | 74.641    = 48
3  | 77.3205   = 48
4  | 80        = 24
5  | 81.9615   = 72
6  | 84.641    = 120
7  | 87.3205   = 24
8  | 89.282    = 48
9  | 90        = 12
10 | 91.9615   = 48
11 | 94.641    = 60
12 | 97.3205   = 24
13 | 99.282    = 24
14 | 101.962   = 24
15 | 104.641   = 12
-------------------------------------------
Enter the index value (1 - 15) to determine all possible corresponding sequences -> 6
-------------------------------------------

 ------------------------------------
  All possible sequences for 84.641
 ------------------------------------
const uint32_t permutations[] {
  124536, 125346, 126354, 132546, 132564, 134256, 134652, 136542, 142356, 143562, 145326, 146532, 152346, 154236, 154632
,
  156324, 156342, 162534, 163542, 164352, 213645, 214653, 215463, 231465, 235641, 236451, 241653, 243615, 243651, 245163
,
  245361, 251643, 253461, 254613, 256431, 261435, 261453, 263451, 265143, 265341, 312546, 312564, 314562, 316254, 316452
,
  324156, 325164, 326514, 341562, 342516, 346152, 352164, 354126, 354162, 356214, 356412, 361542, 362154, 364512, 365124
,
  412653, 413265, 415623, 416235, 421365, 421563, 423615, 423651, 425613, 431625, 435261, 436215, 451263, 452613, 453621
,
  461325, 461523, 463215, 465213, 465231, 512436, 512634, 514326, 516324, 516342, 521346, 523164, 524316, 526134, 532416
,
  532614, 534126, 534162, 536124, 541326, 542136, 546312, 562314, 563124, 564132, 613425, 614235, 615243, 621435, 621453
,
  623145, 623541, 625431, 631245, 632451, 634215, 635421, 641235, 643125, 643521, 645213, 645231, 651423, 652431, 653241


  };
------------------------------------
```

Figure 157: Example output from the permutation generating program

The program has the option to output the values in a form that can be directly copied and pasted into Microsoft Excel. This was so that the sequence counts could be verified using the COUNTIF function to check the counts of the unique distances. Excel verified that the program had produced the correct values (Figure 158) which match the output values calculated from the program (Figure 157).

Unique Distance	Number of Unique Sequences
67.3205	12
74.641	48
77.3205	48
80	24
81.9615	72
84.641	120
87.3205	24
89.282	48
90	12
91.9615	48
94.641	60
97.3205	24
99.282	24
101.962	24
104.641	12

Figure 158: Checking the program has counted the unique distances correctly using the Excel COUNTIF function

Using Excel, a bar chart was developed that represented the unique distance value counts for arbitrary distances d1, d2, d3… through to d15 (Figure 159). From the bar chart, d6 produced 120 unique permutations.

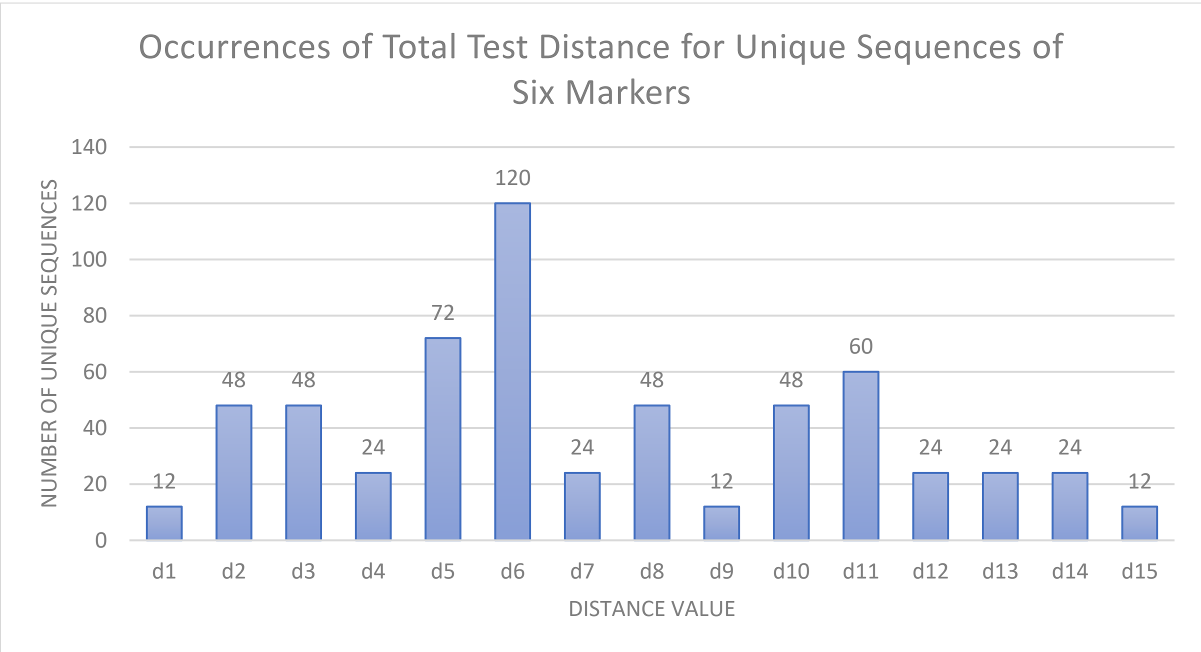


Figure 159: Occurrences of total test distance for unique sequences of six markers

Plugging a value in for the edge distance, such as a = 10 m, provided the bar chart with tangible values. The unique distance corresponding to 120 permutations was now 84.641 m.

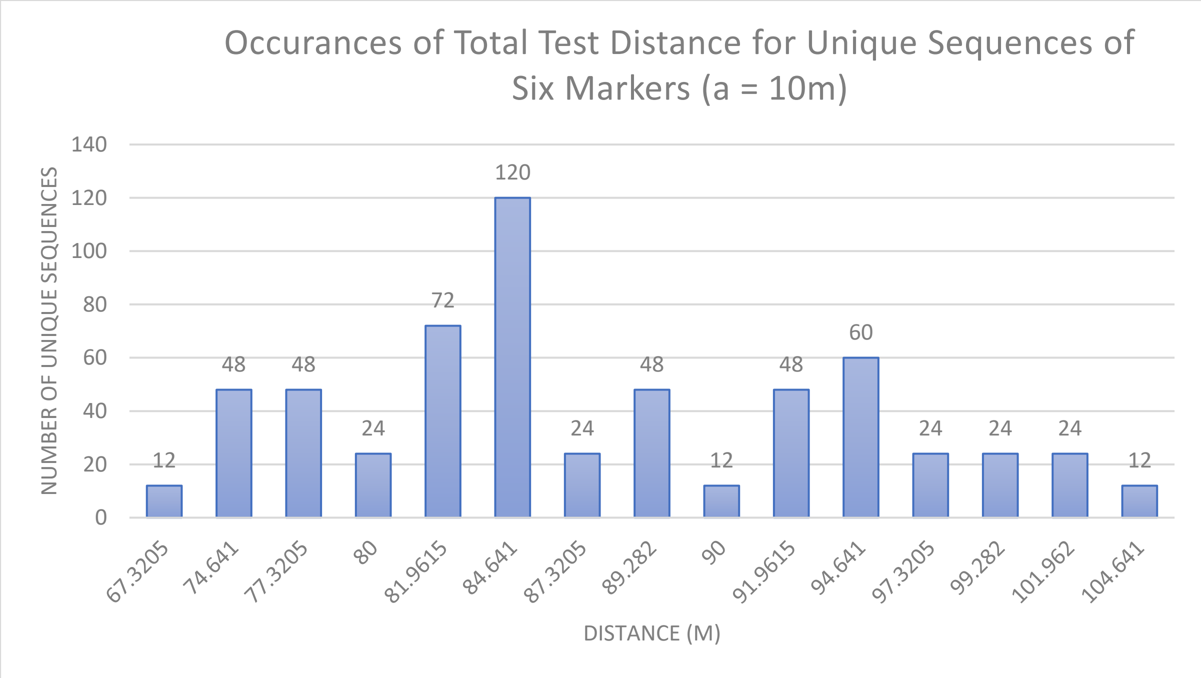


Figure 160: Occurrences of total test distance for unique sequences of six markers (d = 10 m)

3.7.11.6. Programming the Proximity/Ambient Light Sensor

The proximity/ambient light sensor was one of the most important hardware components of the system. It was through this sensor, that the system would differentiate from its competitors, as it would integrate sport-specific equipment with the system. It should be noted that the Si1145 sensor can measure the UV index, which is a number that correlates linearly to the intensity of sunlight that reaches the earth (Silicon Laboratories, 2013). This value was not used in the final program; however, in future work, it could be deemed useful for automatically detecting if the marker needs to adjust the LED matrix brightness to adapt to various light intensities such as in direct sunlight where it needs a very high brightness.

To program the Si1145 sensor, the Adafruit SI1145 library was used (https://github.com/adafruit/Adafruit_SI1145_Library). After experimenting with the library and referring to the Si1145 datasheet (Silicon Laboratories, 2013), it was noticed that the Si1145 produced two sensor measurement values for the proximity, infrared and ambient light; the Adafruit library only obtains one of these values. Referring to the datasheet, PS1_DATA0 and PS1_DATA1 corresponded to the proximity values, ALSIRDATA0 and ALSIRDATA1 corresponded to the infrared proximity values, ALSVISDATA0 and ALSVISDATA1 corresponded to the ambient light data values, and UVINDEX0 and UVINDEX1 corresponded to the (ultraviolet) UV index values. Upon inspection of the Adafruit_SI1145.cpp file, only '0' data values were obtained; thus, the source-code was modified to include functions that obtained the '1' data values too. These changes were made to the Adafruit_SI1145.cpp and Adafruit_SI1145.h files (Figure 161 and Figure 162, respectively).

```
    // returns the UV index * 100 (divide by 100 to get the index)
uint16_t Adafruit_SI1145::readUV(void) {
    return read16(SI1145_REG_UVINDEX0);
}

uint16_t Adafruit_SI1145::readUVOther(void) {
    return read16(SI1145_REG_UVINDEX1);
}

    // returns visible+IR light levels
uint16_t Adafruit_SI1145::readVisible(void) {
return read16(SI1145_REG_ALSVISDATA0);
}

uint16_t Adafruit_SI1145::readVisibleOther(void) {
    return read16(SI1145_REG_ALSVISDATA1);
}

    // returns IR light levels
uint16_t Adafruit_SI1145::readIR(void) {
return read16(SI1145_REG_ALSIRDATA0);
}

uint16_t Adafruit_SI1145::readIROther(void) {
    return read16(SI1145_REG_ALSIRDATA1);
}

    // returns "Proximity" - assumes an IR LED is attached to LED
uint16_t Adafruit_SI1145::readProx(void) {
    return read16(SI1145_REG_PS1DATA0);
}

uint16_t Adafruit_SI1145::readProxOther(void) {
    return read16(SI1145_REG_PS1DATA1);
}
```

Figure 161: Changes made to the Adafruit_SI1145.cpp file (Adafruit_SI1145 Arduino library)

```
uint16_t readUV();
uint16_t readUVOther();
uint16_t readIR();
uint16_t readIROther();
uint16_t readVisible();
uint16_t readVisibleOther();
uint16_t readProx();
uint16_t readProxOther();
```

Figure 162: Changes made to the Adafruit_SI1145.h file (Adafruit_SI1145 Arduino library)

With the changes made, the program was able to retrieve both sets of data. Upon investigation, the '0' data values appeared to be a lot more stable, but sometimes it was difficult to observe the difference with the ball on the marker and the ball off. The '1' data values were a lot higher than the '0' values, by orders of magnitude. These values were generally very easy to distinguish between the ball on and the ball off. This preliminary test was conducted indoors at night, therefore could not reliably be accepted as true for during the day or out in direct sunlight. Initially, code had been written in the slave program to measure all data values, but it caused the program to become slow in response to changes in light. Changes were made to calibrate the marker, to determine which sensor data produced the biggest difference in ball on and ball off values. However, this method was not a fair way to determine which values to use, since the '1' values almost always had a huge difference between values (in an indoor setting). It was more important to use values that had consistency and reliability, such that no 'false' ball removals would be sensed.

Thus, it was important to establish the best possible values to use; therefore, a series of tests were conducted both indoors and outdoors. The aim was to determine which sensor values could reliability detect the ball on the marker and the ball off the marker with consistency. An Australian rules football was used in all the following tests.

3.7.11.6.1. Outdoor Test

The outdoor test involved testing the sensor data values in direct sunlight. The test was based on observations seen in the data over a 20-minute time period (Table 31), but only a few values were recorded; one from intense direct sunlight and another when less sunlight was present (due to clouds).

Table 31: Si1145 sensor outdoor test to determine which values to use for detecting the ball

	Ball on the Marker	Nothing on the Marker
PS1_DATA0	• Very stable (≈776) for all instances of sunlight	• Less stable, however there was a clear differentiation between values on and off (733 difference)
PS1_DATA1	• Incredibly unstable (no baseline with a very large range of values)	• Incredibly unstable (no baseline with a very large range of values)
ALSIRDATA0	• Relatively stable baseline values (≈587 in intense direct sunlight, ≈328 with medium sunlight)	• Less stable, but a clear differentiation between values on and off (≈1650 difference)
ALSIRDATA1	• Incredibly unstable (no baseline with a very large range of values)	• Incredibly unstable (no baseline with a very large range of values)

ALSVISDATA0	• Relatively stable baseline values (≈289 in intense direct sunlight, ≈266 with medium sunlight)	• Less stable, but a clear differentiation between values on and off (≈425 difference)
ALSVISDATA1	• Incredibly unstable (no baseline with a very large range of values)	• Incredibly unstable (no baseline with a very large range of values)

3.7.11.6.2. Indoor Test

The indoor test was conducted during the day, in a room that was very well-lit through natural lighting (no lights were on) and in another room that was a lot darker and not as well-lit. The observations made for each data type was made over a 20-minute time period for both very well-lit and dark room (Table 32).

Table 32: Si1145 sensor indoor test to determine which values to use for detecting the ball

	Ball on the Marker	Nothing on the Marker
PS1_DATA0	• Very stable (well-lit ≈774, not well-lit ≈762)	• Very stable (well-lit ≈708, not well-lit ≈709) • Clear differentiation between ball on and off values in all cases
PS1_DATA1	• Relatively stable baseline. • Angle of the ball altered the baseline value (from ≈1750 to ≈63800), but the baseline remained stable	• Relatively stable, however every now and again it changed to a new baseline (anywhere from ≈12200 to ≈17500) • Clear differentiation in ball on and off values • Every so often the value would change erratically (ranging from ≈8000 to ≈62000) • When left to obtain data for a while, values may change up to 7000 from the baseline.
ALSIRDATA0	• Very stable (well-lit ≈259, not well-lit ≈254)	• Very stable (well-lit ≈285, not well-lit ≈259)

		• Clear differentiation between ball on and off values in well-lit room • Unable to detect changes in IR in the not well-lit room
ALSIRDATA1	• Relatively stable baseline. • Angle of the ball altered the baseline value (from ≈1750 to ≈63800), but the baseline remained stable in either case	• Quite stable • When left to obtain data for a while, values may change up to 7000 from the baseline.
ALSVISDATA0	• Very stable (well-lit ≈262, not well-lit ≈260)	• Very stable (well-lit ≈260, not well-lit ≈260) • The ALS could not reliably detect changes between the ball on or off in any circumstance
ALSVISDATA1	• No clear stable baseline, which changed anywhere from ≈190 to ≈5500	• Clear to see momentary ball on and ball off data, but as the baseline is not stable, the ball on values sometimes exceeded the ball off values

3.7.11.6.3. Test Conclusions

From the observations made that were presented (Table 31 and Table 32), the following conclusions were made for the various sensor data values:

- PS1_DATA0, ALSIRDATA0 and ALS0 sensor values were good for outside in direct sunlight comparing BALL ON and BALL OFF.
- PS1_DATA0 sensor values were good for indoors comparing BALL ON and BALL OFF.
- In some instances, ALSIRDATA0 was only able to detect small changes in infrared inside.
- ALSVISDATA0 was not suitable to detect changes indoors
- PS1_DATA1, ALSIRDATA1 could be suitable to detect changes indoors, BUT would need quite a large upper and lower limit difference value to accommodate for the large changes in baseline values
- ALSVISDATA1 was not suitable to detect differences in BALL ON and BALL OFF accurately and reliably indoors or outside.

From the conclusions made, it was decided that in all applications, the proximity data PS1_DATA0 could be used reliably in all instances. However, future development could prove the other data useful at some point. Thus, the code that obtains all the Si1145 sensor values was kept within the slave program, except a series of Booleans were created, which could be set to false so that the program would not measure these sensor values. For now, only the proximity data PS1_DATA0 would be used for this prototype system.

Implementing just the PS1_DATA0 value into the program, the slave component was able to detect when the ball had been removed in direct sunlight reliably with no false removals sensed. The output of the test can be seen in Figure 163.

```
------- Si1145 Sensor Values [NO BALL] -------
[AVG] Prox: 65535
[AVG] IR: 11690
[AVG] ALS: 1571

Putting Arduino to sleep...
Arduino woke up!

------- Si1145 Sensor Values [WITH BALL] -------
[AVG] Prox: 761
[AVG] IR: 518
[AVG] ALS: 284

[PS1_DATA0] Using Prox0 Sensor Data.
[LOWER] Prox: 745
[UPPER] Prox: 776
[LOWER] IR: 60047
[UPPER] IR: 63174
[LOWER] ALS: 60698
[UPPER] ALS: 30968

------- Test Marker & Character Sequence -------
[RX] Received MARKER sequence: : 634215
Converted the char sequence to a number: 634215
[RX] Received CHARACTER sequence: : kuahsf
[CHARACTER] Character to display on LED Matrix: s
[CHARACTER] LOWER character set to display character index: 18
[COLOUR] Hue to display: 173

Waiting for ball to be removed...
BALL REMOVED!
```

Figure 163: Successfully detecting a ball being removed from the marker in direct sunlight

3.7.11.6.4. Allowing Sensitivity for Swiping Hand Over

The system was tested to see if it could detect a quick swipe of a hand past the sensor. This would allow customisation of tests for various training regimes. The values in printed to the Arduino serial output (Figure 164) suggested that it was possible to do so, as it was outputting different values when the hand was swiftly moved across. The program which was produced for testing the various sensor data values described earlier in this section was used to determine if it was possible to detect a quick swipe of the hand. The marker was successfully able to detect the swipe, even in direct sunlight.

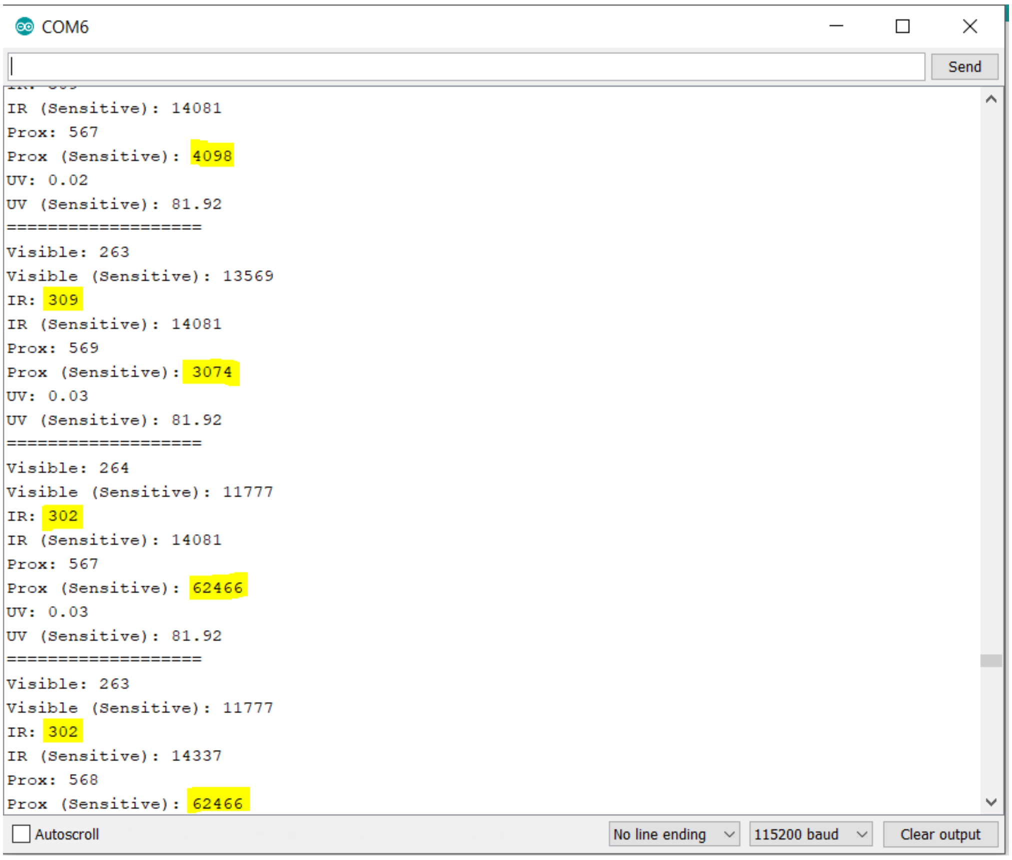


Figure 164: Fast sweep of hand across the proximity/ambient light sensor

3.7.11.7. Generating Random Permutations for Each Test

The Arduino library Entropy (https://sites.google.com/site/astudyofentropy/) was used to generate random numbers. This was because Arduino's random function was not particularly good at generating "random" values. A random index value would be generated between 0 and the length of the permutation array length which would be dependent on the number of markers used in the test. Each permutation for each test would then be stored, thus, the program had been written to check all past permutations to make sure that the one selected did not match one previously used (an acceptable number of tests before a repeat could be specified). The slave would then receive the marker sequence from the master component to determine which position it corresponded to in the test.

Similarly, a character set would be randomly chosen (either lower-case, upper-case or numeric values). As there are only three possible sets, it was more common for these values to repeat. Therefore, code was written to ensure that if a character set was repeated too many times (this could be adjusted), then it would try to generate a new character set. Once an appropriate character set is generated, the program could generate a character sequence for the markers to display. A random set of values would be selected within the appropriate character set. The slave component would receive this character set and as it already knows which position it would be in within the sequence, it would only extract the character it would need to display. Since the characters could be represented as integer values, the correct character set could be determined just by the character integer value. The character set would need to be known because of the different arrays, so the marker would need to know which array to access. Thus, if the value would be within a certain set of limits, it could be established which character set to use.

3.7.11.8. Enabling Arduino Sleep Power Mode

It was desirable to be able to put the Arduino into a sleep mode, so that the device uses less power consumption when it is not required; this would maximise the battery lifetime. The only way of waking an Arduino after it has been put to sleep is via an interrupt. The idea was to use the momentary button to wake the Arduino. Thus, the Arduino could sleep whenever there is no ball on the marker, since during these periods, nothing else is required from the marker.

After following many Arduino sleep tutorials, pressing the button was not waking the Arduino. The button ISR worked perfectly fine when the Arduino was awake, but when put to sleep, it became unresponsive, no matter if HIGH, CHANGE, LOW, FALLING, RISING was set as the interrupt trigger. Some example code from Gammon (2012) presented implementation of a sleep mode by using a pull-up resistor (a resistor between the button signal and VCC) so that the button was HIGH when deactivated and LOW when activated. By incorporating the button in this way, an ISR would be triggered from a button press, which would change the state to become LOW.

The button modules from RobotDyn had a pull-down resistor (resistor between button signal and GND). Therefore, the VCC and GND wires were swapped around and an interrupt was attached so that it triggered an ISR when the button was LOW. This now successfully woke the Arduino when the button was pressed.

Through further experimentation, it was found possible to attach this interrupt just before putting the Arduino to sleep and detaching it as soon as one ISR had been completed, so that the ISR was only called once when the button was pressed.

When attempting to integrate the code into the main program, the Arduino would wake up immediately after being put to sleep. Through a lot of debugging, it was found that the Entropy library was the reason for the Arduino waking up immediately (commenting out "Entropy.initialize();" would allow the Arduino to remain in a state of sleep). Thus, the source code was analysed; it was found the reason for the Arduino waking up, was because the Entropy library used a Watch Dog Timer (WDT), which interrupted the program every 16ms. An issue was that once Entropy was initialised, it could not be uninitialized. However, it was possible to disable all WDTs using header #include <avr/wdt.h>, and the function wdt_disable(). This function was called just before the Arduino is put to sleep. Then, as soon as the Arduino was woken up using the button interrupt, the WDTs were reset using wdt_reset(). The program was now able to successfully put the Arduino to sleep, wake it up with a button interrupt and then use the button as normal whilst the Arduino was awake and in normal use. Output from testing this functionality is shown in Figure 165.

```
Putting Arduino to sleep...
just woke up!
Arduino woke up!
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
Putting Arduino to sleep...
just woke up!
Arduino woke up!
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
BUTTON PRESSED
Putting Arduino to sleep...
```

Figure 165: Successfully implementing a sleep mode in the Arduino while maintaining normal button press whilst awake

3.7.11.9. LED Brightness

Setting the correct brightness of the LED matrix was very important, as they are one of the most battery draining hardware components. By minimising the brightness where possible, this therefore would reduce current draw and increase the maximum battery lifetime, which was a high importance specification.

It was already mentioned in 3.5.3.5.2 during evaluating design alternatives (solution concept stage), that the term pulse width modulation (PWM) was mentioned. An LED brightness could be adjusted by altering the duty cycle (the ratio of time on to time off) using PWM. Various duty cycles are shown in Table 36, with the corresponding PWM values.

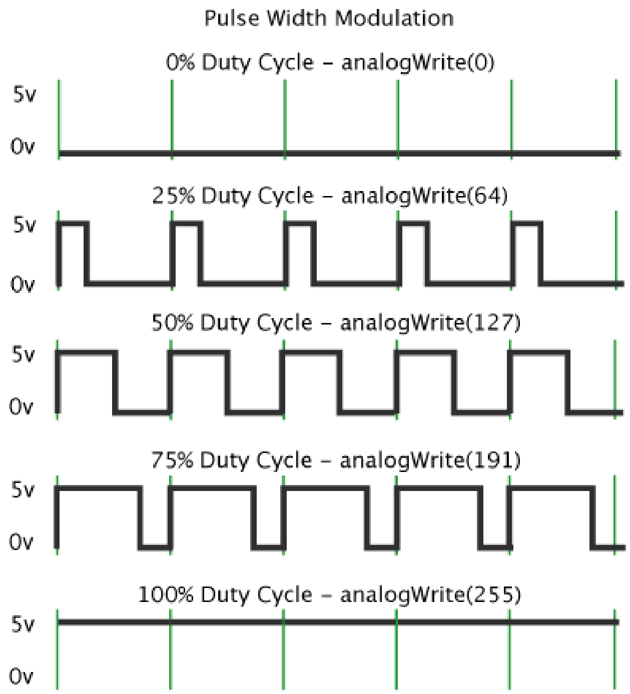


Figure 166: Various duty cycles using PWM (Arduino, 2018)

The maximum PWM value is 255, so outputting a value half of this, gives a 50% duty cycle. It would be a reasonable guess to estimate that the LED brightness would also be 50% of the maximum brightness. However, due to the human eye's non-linear response to light, adjusting the duty cycle of the LEDS using PWM (and thus the brightness) is not perceived in a linear fashion as one would think. This was considered as an advantage, to give the illusion that the marker LED matrix display is bright, but really it is using a low duty cycle and thus less power.

Through investigation, a suggestion was made on the Arduino forum to use f-numbers to generate a linear response for the LEDs. The theory of a full-stop f-number scale was researched and found to be a concept in photography, which represents the ratio of an optical system's focal length to that of the diameter of the entrance pupil (Smith, 2008). The f-number (also known as the f-ratio, f-stop or focal ratio) scale could be represented as a sequence of numbers over the power of the square root of two (Equation 8) with the conventional f-number and the corresponding equation for it.

Equation 8: Conventional f-numbers and corresponding equations

$$\frac{f}{1} = \frac{f}{(\sqrt{2})^0}, \qquad \frac{f}{1.4} = \frac{f}{(\sqrt{2})^1}, \qquad \frac{f}{2} = \frac{f}{(\sqrt{2})^2} \quad \cdots \quad \frac{f}{N}$$

Each neighbouring value is either one f-stop greater or one f-stop lower than the neighbouring value. This was the key for incrementing the LED brightness incrementally using a linear scale. Now, it was desirable to modify the LED using the concept of brightness, represented as a percentage of the maximum value displayable. The brightness could therefore be plotted against the N value of the f-numbers, which could later be used to determine an appropriate PWM value. With a maximum PWM value of 255, the maximum f-number used was f/16, which corresponded to a value of 256. The brightness values were thus incremented in values of 6.25% (100/16 = 6.25%) and these brightness values were put alongside the corresponding N values of the f-numbers. As the conventional f-numbers were not represented by the true calculation, the actual calculated N value were determined, and the data was tabulated (Table 33).

Table 33: Corresponding brightness values for f-numbers

Brightness	Calculated N value
0	1
6.25	1.414
12.5	2.000
18.75	2.828
25	4.000
31.25	5.657
37.5	8.000
43.75	11.314
50	16.000
56.25	22.627
62.5	32.000
68.75	45.255
75	64.000
81.25	90.510
87.5	128.000
93.75	181.019
100	256.000

A graph containing the f-number N values versus brightness was developed (Figure 167).

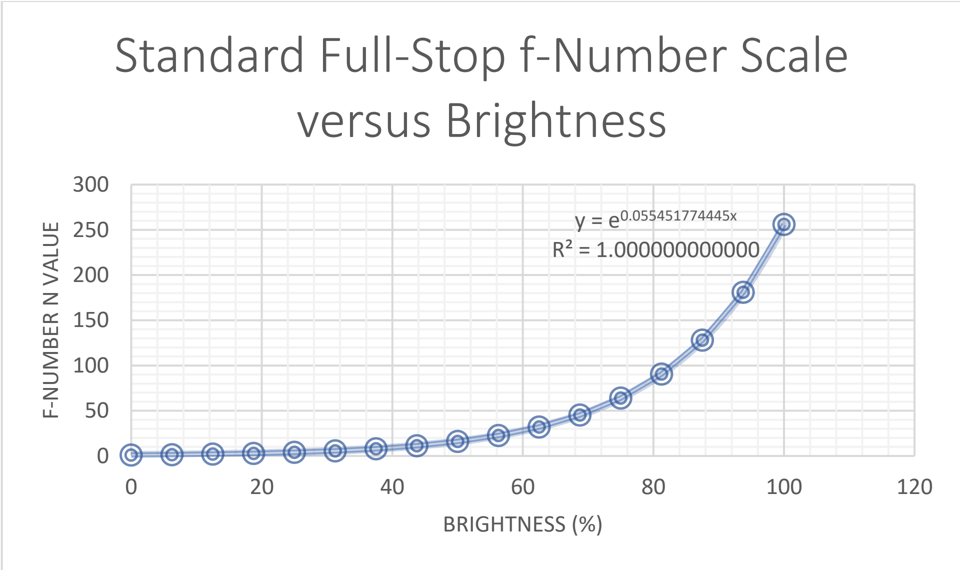


Figure 167: Non-linear graph of f-number N values versus brightness

An exponential trendline was generated with equation: $y = e^{0.055451774445x}$ ($R^2 = 1$). Now it was known that PWM values must be a maximum of 255, so at 100% brightness, PWM should be 255; however, at x = 100, y = 256. Additionally, when brightness would be 0%, the PWM value should be 0; however, at x=0, y=1. As a result, subtracting 1 from the trendline equation provided the desirable PWM values. Therefore, the equation $y = e^{0.055451774445x}-1$ could be used to determine the corresponding PWM value from a set brightness value. As PWM values needed to be in the form of an integer, the resulting value was simply rounded. Testing the equation within the program, incrementing brightness values by 10% appeared to increment linearly.

3.7.12. Selecting an Appropriate Battery

3.7.12.1. Voltage Requirements

Referring to the Arduino Mega technical specifications (Arduino, 2018), when using an external power supply, the recommended input voltage should be 7-12 V.

3.7.12.2. Determining Current Consumption

Up until this point in the project, powering of each device had been performed using a DC power supply. It was deemed necessary to determine an appropriate battery when all electronic circuitry was complete, and all hardware components integrated together. Thus, when all electronic circuitry had been completed and all that was left to do was to connect a battery, the circuit could be connected to a digital display DC power supply to determine the exact values of current draw for each device. Thus, to achieve this, both marker and slave were connected to a digital DC power supply (Figure 168).

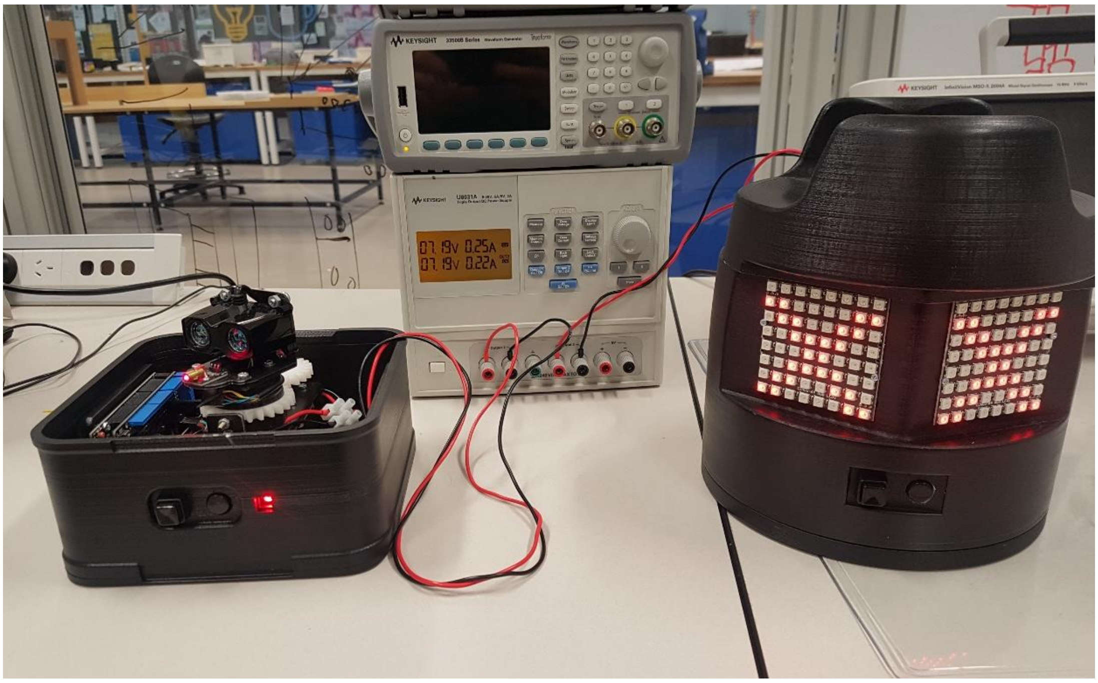


Figure 168: Set-up for evaluating current consumption of master and slave component

It was desirable to determine the peak current consumption and the average for each component. Various tests were conducted, altering the code appropriately, such as adjusting brightness from an indoor value to the maximum outdoor brightness value. Additionally, the current draw was observed when the visual and auditory stimulus was presented, which was presumed to be the moment when the most current draw occurred from the slave. The average current consumption values (Figure 169) shows the master values displayed on the top (OUTPUT 1) and the slave on the bottom (OUTPUT 2). The worst-case scenario (both audio and visual stimulus active on the slave had a high 1.83 A current draw (Figure 170).

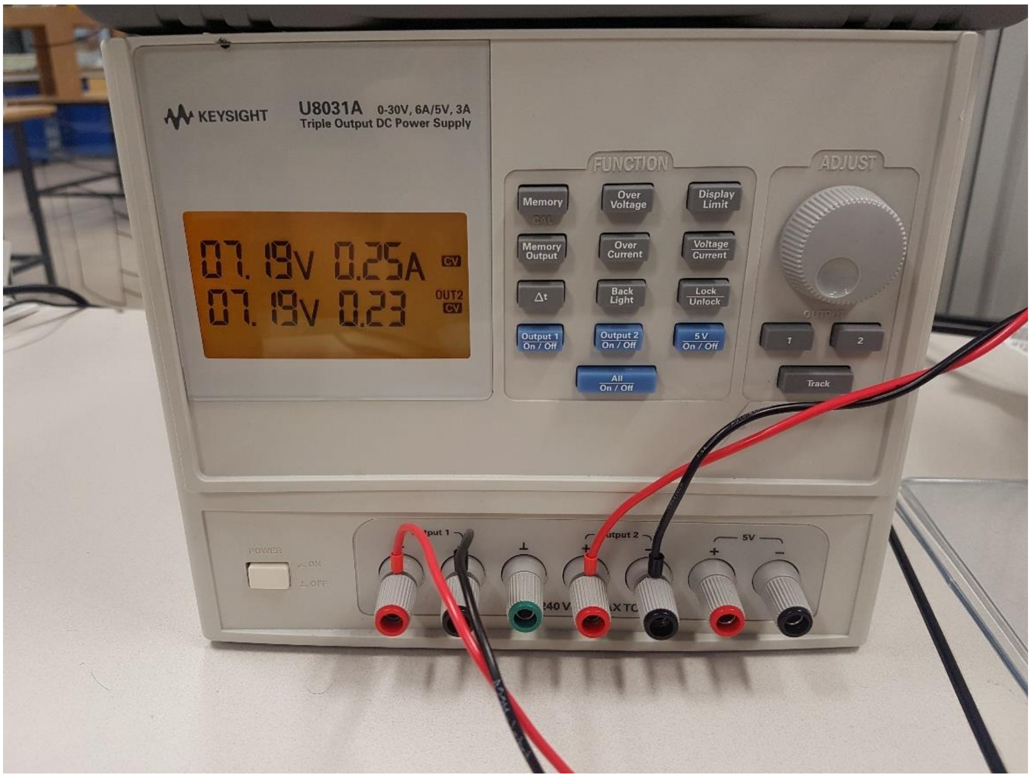


Figure 169: Average current consumption values for master (top values) and slave (bottom values) components

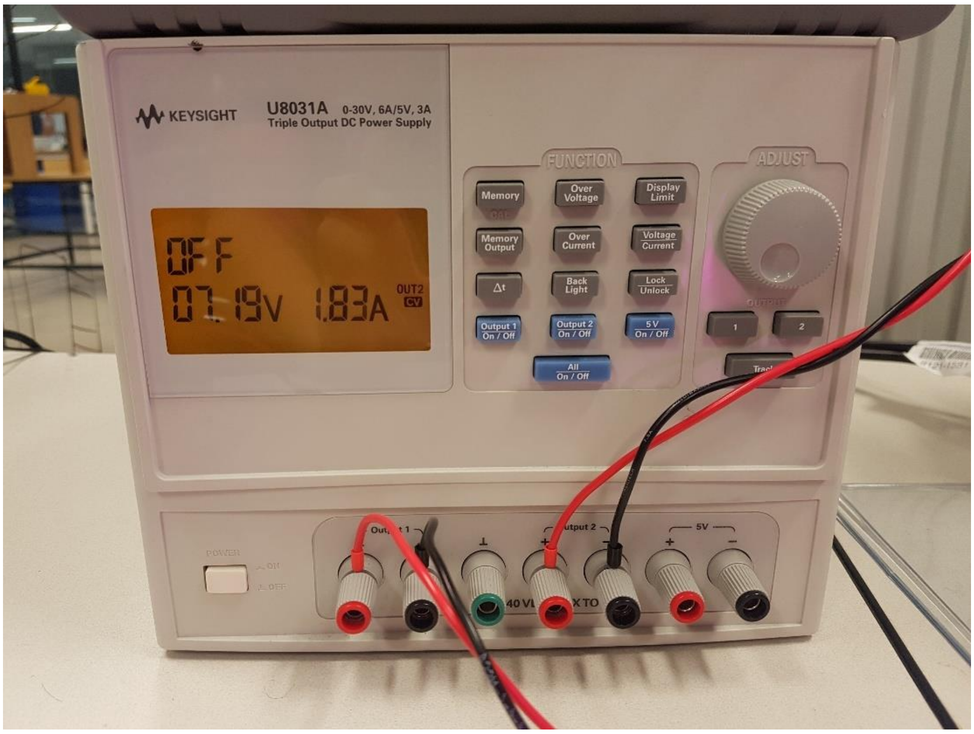


Figure 170: Worst-case current consumption for slave with LED matrix at full brightness and speaker presenting audio

The various current consumptions were recorded along with corresponding notes regarding the hardware components on at the time (Table 34).

Table 34: Current consumption of the master and slave components

Consumption Type	Current Consumption	Notes
Master		
Peak	0.49 A	Stepper motor rotating, LED on, all other components in idle.
Average	0.18 – 0.25 A	All components running except stepper motor (LED, nRF24L01, LiDAR etc.)
Slave		
Inside Brightness Peak	1.07 A	LED matrix on and speaker active
Inside Brightness Average	0.21 A	LED matrix on, nRF24L01+ and even during acquisition of the ball sensor
Outside Brightness Peak	1.83 A	Maximum brightness possible. LED matrix on and speaker active
Outside Brightness Average	0.46 – 0.76 A (variable)	LED matrix on + all other components except speaker
Power Saving Mode	0.15 A	Recorded when Arduino was in sleep mode

3.7.12.3. Battery Alternatives Decision Criteria

Similarly to the solution concept phase of the project, a set of decision criteria needed to be developed to evaluate the battery alternatives. Several decision criteria were deemed important when evaluating the alternatives; it needed to be low-risk, have a large power capacity, low-cost, as small as possible physical size, has minimised weight, has a high current draw capability and can be recharged with minimal or no memory effect (Table 35). Having a low-risk battery was imperative, as the safety of the user, athlete and even during the development stages of creating the prototype needed to be accomplished. It is this reason why this criterion had a significant weighting factor of 40%. The physical size was of great importance at this stage of the project, since a battery too large to fit within the enclosure would not be acceptable. Thus, this criterion received a weighting factor of 15%. Power capacity, low-cost, minimised weight and current draw capabilities were all deemed equal in importance, with a weighting factor of 10% Although low-cost had generally been the highest rated decision criteria in other selection processes, power capacity, minimised weight and current draw were also important. It was important the battery had a good power capacity, to increase the maximised battery lifetime, as well as having a good capability of drawing current quickly, such as in the instance of the maximum current draw of 1.83 A seen in the test described in this section. A rechargeable battery would be most optimal, but was not an essential, thus this criterion received a 5% weighting factor. The criteria and their corresponding weighting factors are presented (Table 35).

Table 35: LiDAR scanner decision matrix criteria and corresponding weighting factor

Criteria	Weighting (%)
Low-Risk	40
Power Capacity	10
Low Cost	10
Physical Size	15
Minimised Weight	10
Current Draw Capability	10
Rechargeable and Memory Effect	5
Total	**100**

3.7.12.4. Battery Alternatives Evaluation

The decision matrix for evaluating the battery alternatives (Figure 171) shows the possible alternatives could have been nickel-metal hydride (Ni-MH), lithium polymer (Li-Po), lithium-ion (Li-Ion), nickel-cadmium (Ni-Cd), Lead Acid, Alkaline, Lithium Coin Cell, lithium iron phosphate (LiFePo4). The decision matrix found the Ni-MH battery to be the most suitable candidate for the project. Although the many benefits of Li-Po and Li-Ion, these batteries were deemed too dangerous, since there was a possibility of large explosions, especially with a prototype device that would not have the appropriate safety precaution circuitry. LiFePo4 was found to safer than the Li-Po and Li-Ion, however, again, there were still risks associated with using this type of battery. A lead-acid battery has excellent power capacity, is extremely cheap and can be recharged. However, it is not very safe, as well as being extremely large and heavy. A lithium coin cell was found to be low-risk, low-cost, extremely small and have minimised weight, however its current draw capability was very low. Ni-Cd had good physical size and minimised weight, however the power capacity was not the greatest and there are memory effects associated with charging these batteries incorrectly. The decision matrix found alkaline batteries just falling short. These are very low-risk, reasonably low-cost, very small size and minimised weight, however their current draw capacity is not the best, they are not rechargeable, and their power capacity was also not the greatest. Ni-MH were deemed very safe, had a good power capacity, were quite low-cost and physical size and are optimal for recharging. They are slightly heavier, but this was a reasonable trade-off.

Criteria / Weighting / Design — Battery Alternatives	Low Risk	Power Capacity	Low Cost	Physical Size	Minimised Weight	Current Draw Capability	Rechargeable and Memory Effect	Total
	0.4	0.1	0.1	0.15	0.1	0.1	0.05	1
Ni-MH	9 / 3.6	8 / 0.8	7 / 0.7	7 / 1.05	6 / 0.6	6 / 0.6	10 / 0.5	7.85
Li-Po	0 / 0	8 / 0.8	2 / 0.2	7 / 1.05	10 / 1	7 / 0.7	10 / 0.5	4.25
Li-Ion	0 / 0	5 / 0.5	2 / 0.2	8 / 1.2	10 / 1	7 / 0.7	10 / 0.5	4.1
Ni-Cd	4 / 1.6	1 / 0.1	9 / 0.9	10 / 1.5	10 / 1	4 / 0.4	5 / 0.25	5.75
Lead Acid	2 / 0.8	10 / 1	10 / 1	1 / 0.15	1 / 0.1	10 / 1	10 / 0.5	4.55
Alkaline	10 / 4	3 / 0.3	8 / 0.8	10 / 1.5	10 / 1	1 / 0.1	0 / 0	7.7
Lithium Coin Cell	10 / 4	0 / 0	10 / 1	10 / 1.5	10 / 1	0 / 0	0 / 0	7.5
LiFePo4	5 /	8 / 2	1 / 0.8	7 / 0.1	10 / 1.05	7 / 1	10 / 0.7	6.15

Figure 171: Decision matrix for battery type alternatives

Legend

x	Rating factor (0 - 10)
y	Relative importance

Sources: (Fried, 2013; Hua and Syue, 2010)

As the due date for the project was fast approaching, the batteries needed to be purchased locally to save time spent waiting on shipping and transit. Jaycar Electronics were offering a 7.2 V, 3300 mAh stick pack battery for $59.95. However, Hobby Habit were offering a 7.2 V, 5000 mAh stick pack battery for just $49.95, which was cheaper and had a higher power capacity than the one offered by Jaycar Electronics. Hobby Habit was also selling a 7.2 V, 2400 mAh stick pack for $35. For $15 more per battery, the maximum battery lifetime would be more than doubled. As a result, the 5000 mAh battery was chosen to be purchased from Hobby Habit. It was unfortunate that time constraints were present, as HobbyKing were offering a 5000 mAh, 7.2 V stick pack for just $28.71, which had to be shipped from NSW, meaning they would not arrive in time before the end of the project date. The batteries obtained from Hobby Habit were able to be connected to all the components successfully.

3.7.13. Final Prototype Assembly

3.7.13.1. Master Component

The final embodiment of the design layout, encompassing mechanical, electrical and software design established a final prototype. An isometric view (Figure 172), front view (Figure 173) and side view (Figure 174) of the master component in its complete form shows the integration of the whole assembly.

Figure 172: Isometric view of the final master component prototype

Figure 173: Front view of the final master component prototype

Figure 174: Side view of the final master component prototype

The master component inner componentry can be observed with images containing the assembly with the enclosure lid removed for both isometric view (Figure 175) and top view (Figure 176).

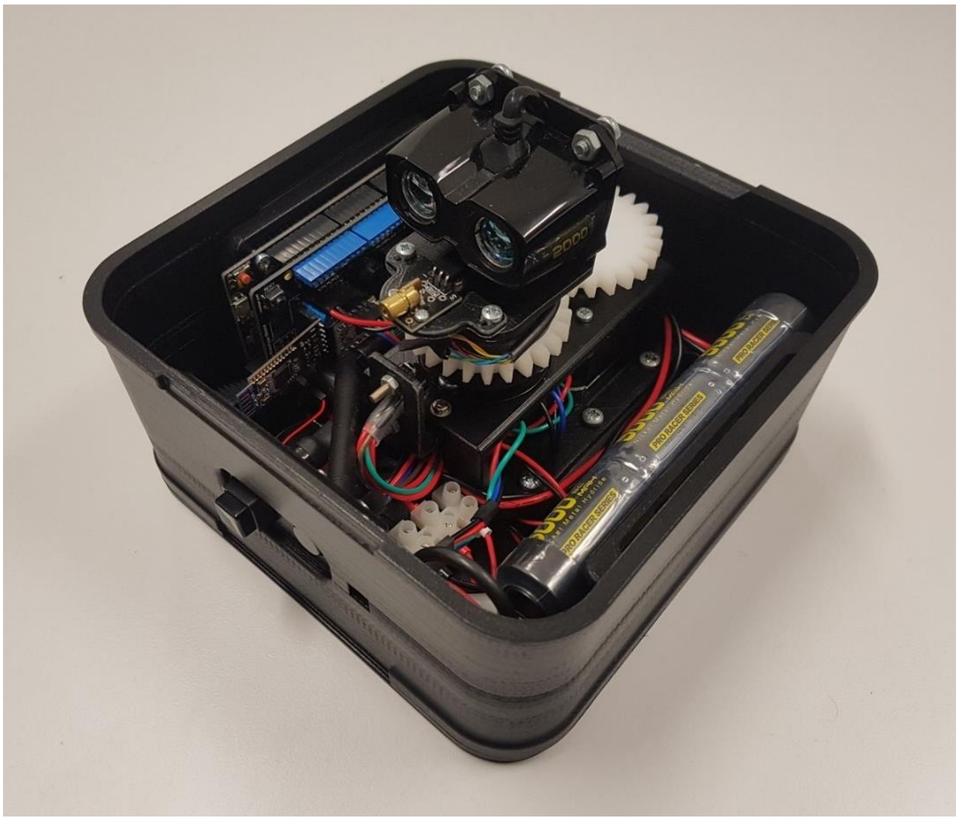


Figure 175: Isometric view of the master prototype without enclosure lid to see inner componentry

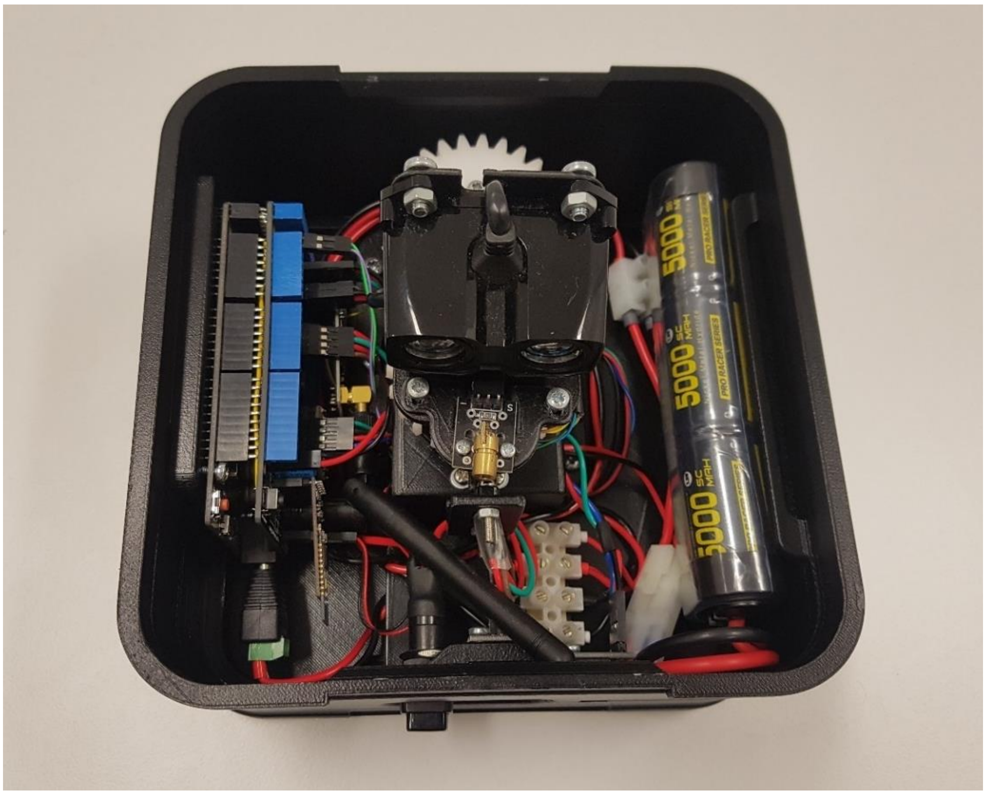


Figure 176: Top view of the master prototype without enclosure lid to see inner componentry

3.7.13.2. Slave Component

An isometric and side view of the marker is shown (Figure 177 and Figure 178, respectively).

Figure 177: Isometric view of the slave component

Figure 178: Side view of the slave component

The marker is shown on its side (Figure 179), which can be used for pre-planned agility tests, to use the marker as a timing gate. The marker was abel to successfully incorporate sporting equipment with the device itself, including integration of an Australian football (Figure 180), netball (Figure 181), rugby ball (Figure 182) and basketball (Figure 183).

Figure 179: Slave component lying on flat backing

Figure 180: Slave component with Australian football positioned on it

Figure 181: Slave component with netball positioned on it

Figure 182: Slave component with rugby positioned on it

Figure 183: Slave component with basketball positioned on it

From the various images presented, the marker had compatibility with a variety of sports. Each of the balls could be positioned easily on it and were very stable. The marker without the top section allowed observation of the inner componentry (Figure 184). The top half section could remain connected by its corresponding wires while also still viewing the inner componentry (Figure 185).

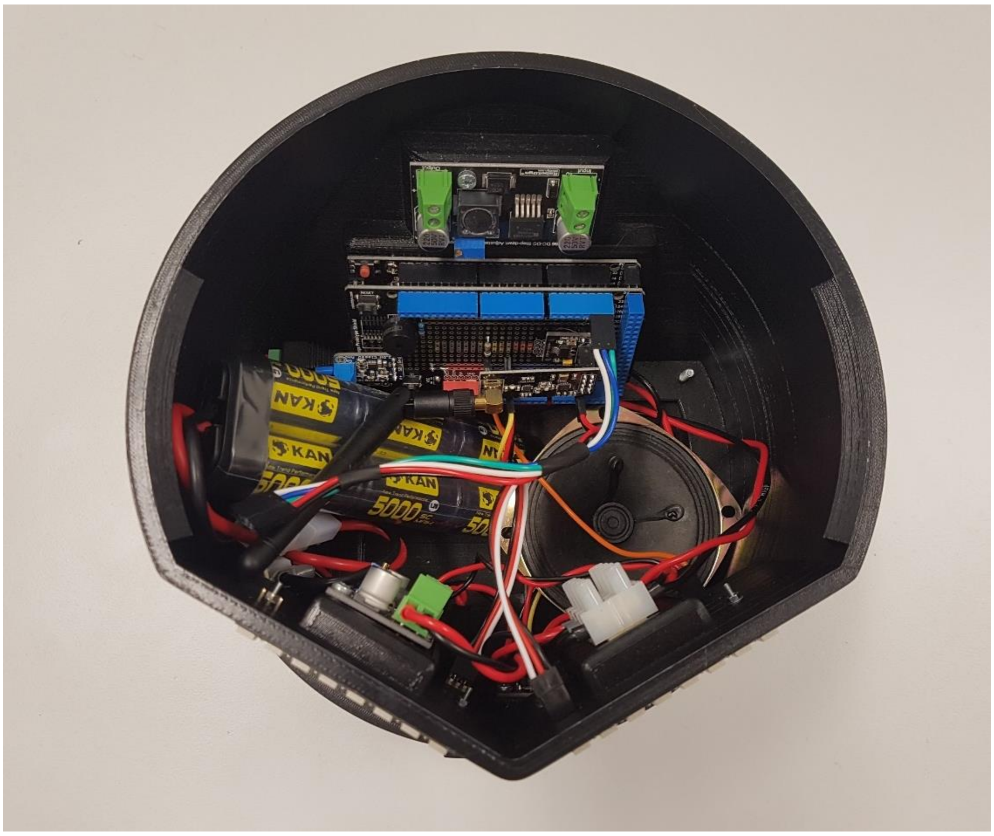


Figure 184: Top-view of the slave component with no top section to see the inner componentry

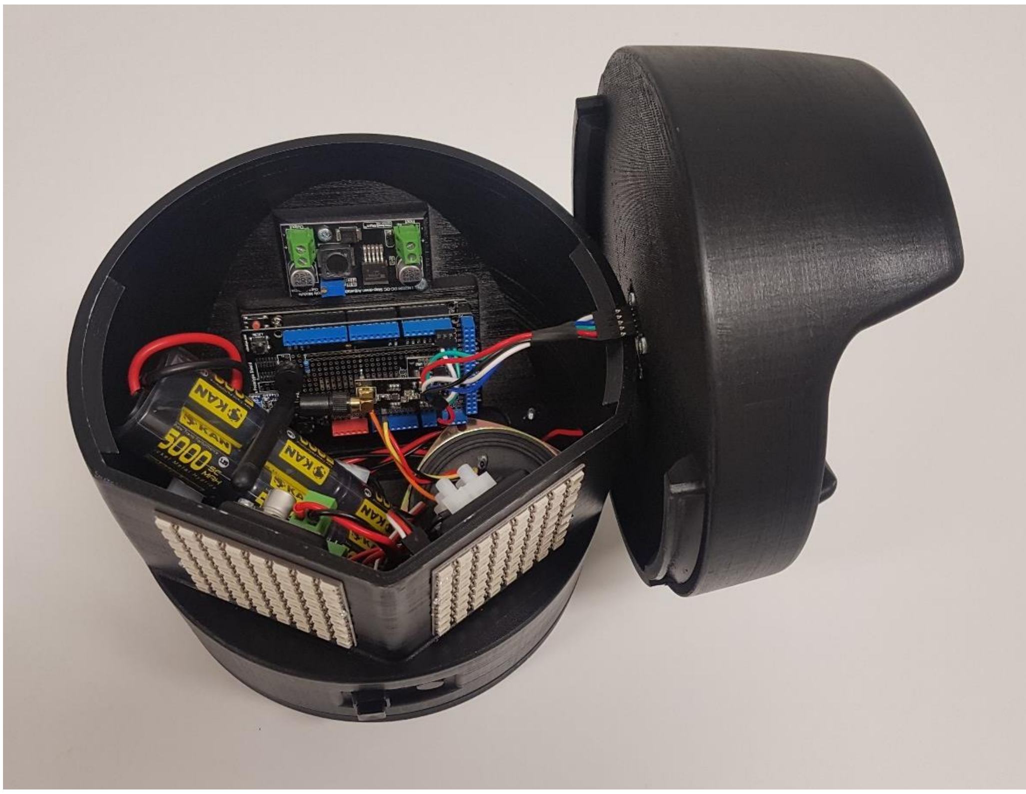


Figure 185: Top-view of the slave component with top section attached

The master component and all slave markers with a various type of balls mounted on each (Figure 186) demonstrates the integration of all components to form one complete system.

Figure 186: Final prototype devices that encompass the sports agility testing system

3.8. Design Detail

3.8.1. Detailed Analysis

3.8.1.1. Bill of Materials

A complete Bill of Materials for the final components incorporated into the prototype (Table K- 1 in Appendix K: K.1) contains information including the part name, supplier, catalogue number, link, quantity purchased, the unit price, shipping and subtotal. The total cost for all components, including the sports specific game balls was $1,521.75. This total was within the budget of $1,600, meeting this engineering specification constraint.

3.8.2. Design Testing

Design testing was conducted to measure the actual prototype metrics from the target and threshold target values, as well as determine qualitative data too. This section aimed to provide a detailed analysis of the various tests conducted for the system and to discuss the results.

3.8.2.1. System Interactions with the Environment and Testing

Testing was completed to see the interaction the marker had with the outside environment with speaker sounding and LED matrix displaying (Figure 187). It was found difficult to take a photo of the marker in the sunlight and be able to see the LED matrix display even though it was clear to see in real life.

Figure 187: Testing ball removal outdoors

Testing was conducted to ensure that the marker could be assisted with positioning using the LiDAR scanner outdoors in direct sunlight, even with the clear CCTV dome on it (Figure 188). Looking closely at the figure, the red dot from the laser diode module can be seen between the two LED matrix displays on the slave device, which assisted in setting it up in the correct position.

Figure 188: Testing marker set-up outdoors in direct sunlight

The test could be configured in a regular hexagon outdoors (Figure 189), with various balls being positioned on the markers (Figure 190).

Figure 189: Testing marker set-up outdoors

Figure 190: Test set-up with balls on markers

3.8.2.1.1. Knocked Over Markers

Although no run-through of the test was able to be completed, it is important to discuss the consequence to the system if one of the slave devices was knocked over due to an athlete knocking it whilst picking up the sporting equipment. As the system has never been designed before, there are many consequences which could result, which should be investigated further to establish a set of rules in which the test must follow. One possible idea for such a scenario, could be that that if a marker is knocked over, then the test is stopped and becomes invalid; the balls must be placed on the markers again and the test must be reset. Another possible idea could be a knocked marker would result in an addition of time to the athlete's run. For example, for each knocked over marker, an additional one second would be added to the total time. For either of these scenarios, it would be ideal to introduce some sort of hardware which could recognise position in space or sense acceleration so that the marker is aware that it has fallen over. This is something which could be investigated further in future work. An alternative idea is that no consequence occurs due to a marker being knocked over and the test continues to run as normal.

It is assumed that it will likely be a common occurence for markers to be knocked over when conducting the test. Obviously, this could cause damage to the internal electronic hardware or mechanical enclosure of the prototype system. This is because this project aimed to provide a proof-of-concept system which met the minimum requirements to perform the desired test. Being quite fragile, it would not be designed to be knocked around. However, in future design iterations, it would be extremely desirable to implement devices which are impact resistance and would not cause damage to the devices if they were knocked over.

3.8.2.1.2. Marker Stability

A "dummy" run was conducted with the markers set up as if they were ready for a real test run (Figure 190) which were able to present a successive set of stimuli, to see the interactions between the system and the environment. It was found that the stability of the slave markers was slightly compromised due to an uneven grassy surface. It is not appropriate to assume that all fields would contain well-kept even grass, especially for lower level athletes and sporting clubs so the mechanical design should be modified so that it is more stable on uneven surfaces. Additionally, even with a flat surface, it was possible for the markers to tip slightly backwards due to the mechanical design of the slave device. There was slight unstability as majority of the marker's base was located towards the front of the device (due to the flat surface on the back), therefore, most of its mass was located at the centre or at the front of the device. Thus, it would be imperative in future work to establish a mechanical design which may allow the slave devices to have a more stable base so that it is difficult for the marker to be knocked over.

3.8.2.2. Reliability

The literature review of this thesis discussed the concepts of reliability and validity, and how these corresponded with evaluating test quality. In the early stages of the project, it was desirable to test the reliability and validity of the test when it was complete. The reliability of the test was going to be completed through analysis of variance (ANOVA) via the intraclass correlation coefficient by integration with the Spearman-Brown Prophecy Formula. The reliability was also going to be confirmed through test-retest via the Pearson Product-Moment Correlation equation. Initially, it was thought that the prototype would be able to be completed on real athletes to determine the reliability of the test. However, it was established that designing and developing the test was already enough work as it was. Through an agreement between the author and supervisors, Dr. Elliott, Dr. Hobbs and Prof. Taylor, the shift was focused on how well the prototype met the needs of the test itself, if it could perform all the functions that were desired, as well as ensuring that it met a set of established engineering design specifications. If time permitted, the test was hoped to be conducted on the author and supervisors to see how the test performed and what the test results would be like; to be able to make preliminary judgements on comparisons between other standardised agility tests. Unfortunately, due to the significant time constraints, the system was unable to be fully completed such that a full test could be conducted.

3.8.2.3. Validity

For a test to be valid, it needs to measure what it has specified it is to measure, as accurately as possible (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Bishop, 2008; Haff and Triplett, 2016; Lacy, 2011). Several types of validity were discussed in the literature review; content validity, construct validity, criterion-related validity; concurrent and predictive validity.

To re-iterate, the content validity of a test refers to the degree at which a test can satisfy whether it is truly measuring a skill or ability (Barrow and McGee, 1979; Baumgartner and Jackson, 1983; Lacy, 2011). The content validity of the proposed sports agility tester was going to be established by ensuring that the test did indeed test the components of agility, which were established within the literature review. Although the development of the test was unable to be completed, it was possible to anticipate the components of agility that would be used within the test. Using the deterministic model by Young et al. (2002), Sheppard and Young (2006) and (Nimphius, 2014), the components of agility could be established to compare the test against. With what was achieved in terms of product development and what the system was able to achieve, it was possible to emphasise that the test would implement both perceptual and decision-making factors, using the randomised visual and auditory stimuli presented by the markers. The athlete would require anticipation for the next activated marker, visually scanning each one in an attempt to see/hear the stimulus as quickly as possible. The athlete would then need to react as quickly as possible to each activated stimulus. Pattern recognition would be used, as highly agile athletes should be able to know exactly where they need to run when only the last marker is remaining, showing a knowledge of the situation. Change-of-direction speeds would also be employed, with the athlete requiring careful positioning of the body through foot placement, stride adjustment and body lean and posture. They would need to sprint as quickly as possible to a marker, through initial reactive strength and power and rate of force development when knowing where to run to. The athlete would employ various strength properties of concentric, isometric and eccentric strength as they make sudden change-of-directions.

The construct validity refers to the degree at which a test measurement can accurately measure an underlying construct (Haff and Triplett, 2016). The construct validity of the test was going to be tested by evaluating performance of the test conducted by highly-skilled athletes and lower-skilled players if time permitted. Even during the literature review, it was emphasised that it would likely not be achievable to determine the construct validity of the test.

The concurrent validity of a test is the degree to which it scores against another established test, which measures the same ability. Thus, the test results were going to be compared against well-established tests such as the Illinois Agility or the AFL Agility Run. Unfortunately, not having a complete test running, this validity could not be determined.

It was established that the predictive validity of the test would not be determined, since it refers to the degree at which the measurement may predict some future form of measure or performance (Barrow and McGee, 1979; Haff and Triplett, 2016).

3.8.2.4. LiDAR Sensor Accuracy

The lid of the master component had to be removed so that that a micro-USB could be connected to a computer to receive the serial output of the measurement values. Figure 191 and Figure 192 shows the test layout conducted to determine the actual measurement distance from the LiDAR scanner distance output. It was found that the measurement values were not very stable, as they had been in other tests. Additionally, it was found that the accuracy was not as precise anymore, with up to ±15 cm of incorrect results. This was a bit shocking as the other test results had shown that the accuracy worst-case was ±10 cm. The tests had produced more reliable results when the markers were not painted or sanded. Thus, painting the markers may have caused some kind of problem with the reflection of the laser of the LiDAR scanner, causing inconsistency with the results of the distance measurement.

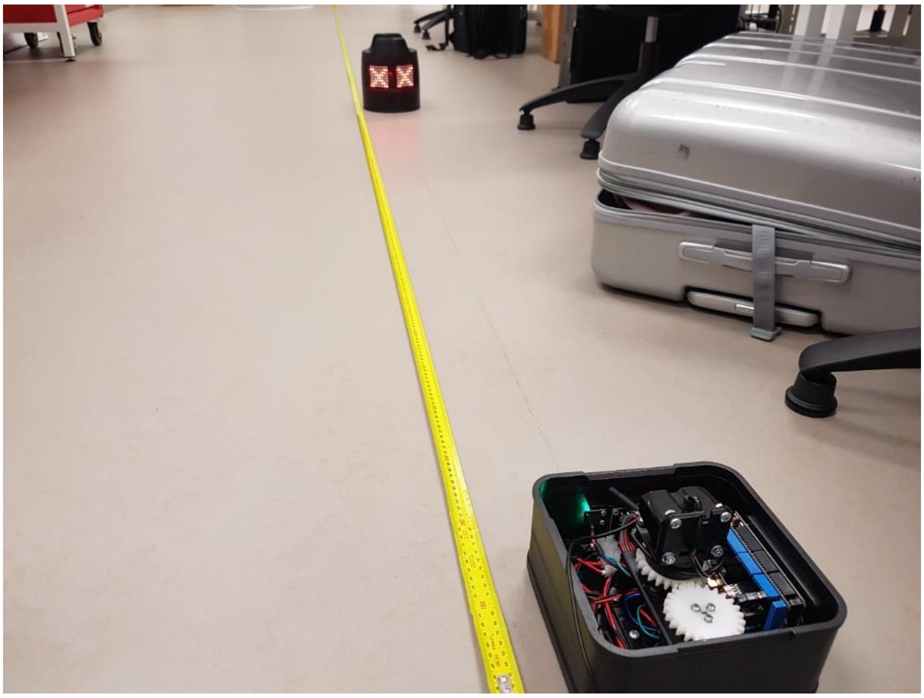


Figure 191: Testing LiDAR scanner accuracy using a tape measure

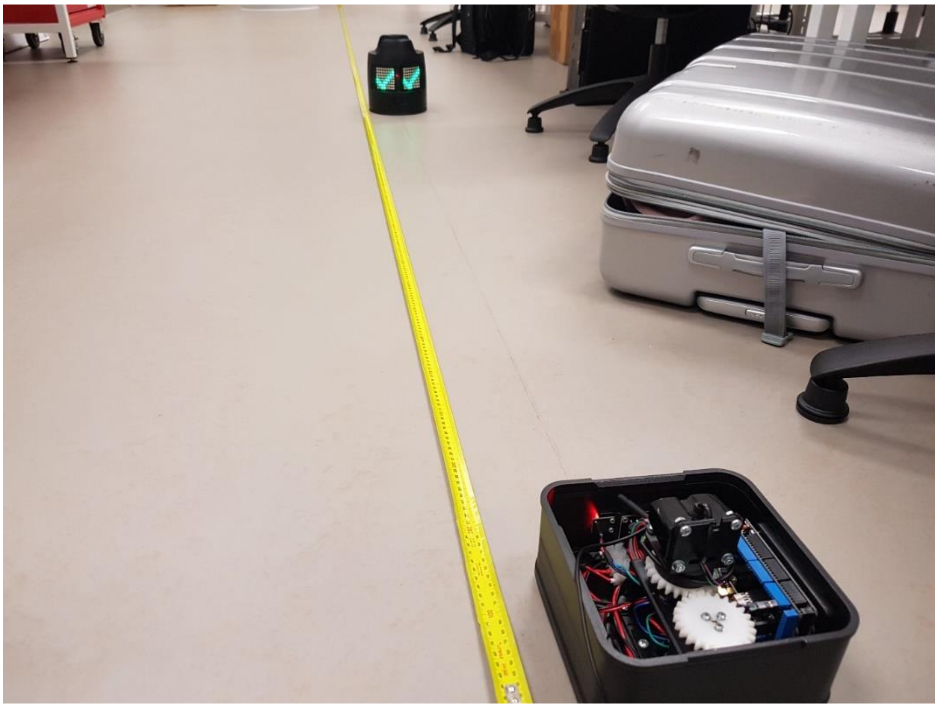


Figure 192: Testing LiDAR accuracy and response of the marker display

3.8.2.5. Ball Sensor Response Time

To determine the ball sensor response time, the marker was calibrated with the ball on it. The ball was then quickly removed, before the marker had time to reach the program where it measured the sensor data. As a result, the marker would immediately detect the ball being removed, so the response time for this could be determined. Ten tests were conducted, with the various response times (Figure 193) showing the average sensor response time being 0.0036427 seconds.

```
Putting Arduino to sleep...
Arduino woke up!

------- Si1145 Sensor Values [WITH BALL] -------
[AVG] Prox: 794
[AVG] IR: 254
[AVG] ALS: 259

[AVG] _Prox: 55923
[AVG] _IR: 6707
[AVG] _ALS: 57201

[PS1_DATA1] Using Prox1 Sensor Data.
[ALS_IR_DATA1] Using IR1 Sensor Data.
[ALS_VIS_DATA1] Using ALS1 Sensor Data.

[LOWER] Prox: 54804
[UPPER] Prox: 57041
[LOWER] IR: 6572
[UPPER] IR: 6841
[LOWER] ALS: 56056
[UPPER] ALS: 58345

------- Test Marker & Character Sequence -------
[RX] Received MARKER sequence: : 634215
Converted the char sequence to a number: 634215
[RX] Received CHARACTER sequence: : kuahsf
[CHARACTER] Character to display on LED Matrix: s
[CHARACTER] LOWER character set to display character index: 18
[COLOUR] Hue to display: 232

Waiting for ball to be removed...
Time taken to detect: 1856 microseconds
```

```
[COLOUR] Hue to display: 31

Waiting for ball to be removed...
Time taken to detect: 1840 microseconds
```

```
Waiting for ball to be removed...
Time taken to detect: 7232 microseconds
```

Figure 193: Various ball sensor response times

3.8.2.6. User/System Communication Response Time

The benefit to using the Serial Bluetooth Terminal application on the smartphone, was that it provided the time with 0.001 second resolution. This meant that commands could be provided to the system and the response time could be calculated by subtracting the sending time from the receiving time. Various commands were sent to the system to determine the feedback response time (Figure 194).

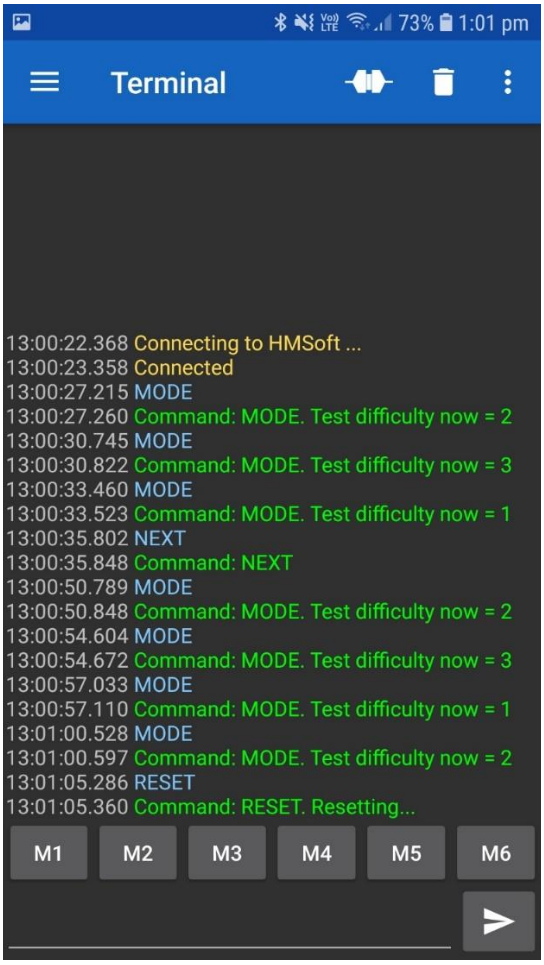


Figure 194: Extremely fast response times from user commands

The transmit and receive times were recorded and the difference calculated. The values were tabulated (Table 36) with the average transmit-receive time being 67.7 ms (0.0677 seconds)

Table 36: Bluetooth transmit (TX) and receive (RX) times

TX Time (ms)	RX Time (ms)	ΔTime (ms)
215	260	45
745	822	77
460	523	63
802	848	46
749	848	99
789	848	59
604	672	68
33	110	77
528	597	69
286	360	74
	AVERAGE:	**67.7 ms**

3.8.2.7. Calculating Battery Life-Time

Due to significant time constraints, a whole day was not able to be dedicated to testing the system maximum battery lifetime. However, this value could be estimated using the test conducted for determining average and peak current consumption of the slave and master components. Referring to Table 34, some reasonable estimates could be determined.

The average current consumption of the master was 0.18 – 0.25 A. Peak current consumption occured when the motor was on (0.49 A), which occured only when setting up the markers. The stepper would run for a maximum of 10 seconds, then the remaining time it is on it is consuming at worst, 0.25 A of current. 10 seconds is equivalent to 0.278% of an hour. Therefore, 99.722% of the time, 0.25 A is being drawn and the remaining 0.278% is during peak current consumption. This brings into account the possibility of the user recalibrating the test markers every now and again. Therefore, it could be assumed the master draws on average, 0.2506672 A per hour. With a 5000 mAh capacity, in theory and in an ideal world, the master should run for 19.95 hours. Obviously, this value would be reduced in real-world application.

Similiar calculations could be completed on the slave component. With inside brightness, average current consumption was 0.21 A, with peak current consumption at 1.07 A when the sound stimulus was simultaneously presented with the LED matrix displays. It could be assumed the test would be conducted once every minute on average, with peak current consumption approximately occurring 1 second every minute (when the sound stimulus is presented). Ignoring power saving mode, it could be estimated peak current occurs 1.67% of the time and so 98.33% of the time average current consumption occurs. Thus, 0.224362 A would be drawn every hour. According to these calculations, the slave should last for 22.29 hours indoors.

Using similar estimates for outside brightness, with peak current consumption at 1.83 A and average at 0.76 A; a total of 0.777869 A would be drawn every hour. Thus, outdoors, the marker should last for approximately 6.43 hours.

It should be noted that allowing the devices to enter a sleep mode after a period of inactivity or between tests was considered and was a desirable characteristic of the system. This requirement was found to be a 2nd order requirement with low relative importance of 0.571 (Table 10). As a result of the low relative importance of the requirement, the corresponding specification related to the number of power modes also received a low relative importance of 0.45 (Table 16). Despite these low values, the author considered the use of a sleep mode important to the overall system, as it would contribute to maximise the battery lifetime. Therefore, these low relative importance values did not truly represent the needs of the system. Thus, a sleep mode for the slave devices was integrated into the system to be entered between tests. In future work, it would be desirable for these devices to enter this sleep mode after a period of inactivity too. Development of a sleep mode for the slave devices was successfully implemented and is described in Section 3.7.11.8. Unfortunately, whilst in sleep mode, the slave devices still seemed to draw the same amount of current as if they were in normal mode. This is something which should be investigated further in future work.

3.8.2.8. Marker Load Capacity

A rough estimate of the marker load capacity was determined. 20.6 kg of mass was applied to the marker (as measured by a set of bathroom weights); the marker remained undamaged. No more mass was applied in case of breaking the component, which would have been detrimental. Thus, using the acceleration due to gravity (9.81 m/s^2), the corresponding force on the marker could be determined:

$$F = ma$$

$$\therefore F = 20.6\ kg \times 9.81\ m/s^2$$

$$F \approx 202\ N$$

3.8.2.9. Containment of the System

The whole system was able to fit inside the larger suitcase mentioned during development of the engineering specifications. Padding was achieved using two medium woollen blanket throws to protect the slave components and one medium blanket throw to protect the master component; to stop them from getting damaged during transit. The master component also had foam placed over the clear 4" CCTV dome to protect it from getting scratched. All the components of the system could be contained within the suitcase (Figure 195). The components were able to be wrapped up in the blankets and strapped into the suitcase (Figure 196), thus showing that they met the requirement that the whole system needed to fit within a containment.

Figure 195: All prototype devices of the system contained within a suitcase

Figure 196: Prototype devices wrapped in blankets to protect them during transit within the suitcase

The mass of the whole system was measured with everything contained within the suitcase (six slave components, the master component and three medium sized blankets). The total mass was found to be 15.4 kg including the weight of the suitcase, which met the threshold engineering specification target of being under 16 kg. As the suitcase had wheels, the only time the user woud have to lift the suitcase is when putting it in and out of a car. The suitcase was tested rolling along gravel and grass and although a bit more difficult than smooth ground, it was still achievable.

3.8.2.10. Testing Noise Level of the System

Due to budget constraints, a real sound meter could not be used to measure the sound level produced by the system. However, a study by Murphy and King (2016), which tested the accuracy of various iOS and Android applications found the most accurate application for Android was Sound Meter (Smart Tools co., Daegu, South Korea), which was found to within ±2 dB(A) of true noise levels. Using this application, the master component produced a peak noise level of approximately 54 dB, while the noise stimulus from the slave component produced a peak noise level of approximately 82 dB. Although now repealed, the *Managing Noise and Preventing Hearing Loss at Work Code of Practice* (*Work Health and Safety Codes of Practise 2011)* suggested testing noise levels at a measurement distance position of 1 m (Australian Government, 2011). Thus, he mobile phone which contained the sound meter was positioned 1 m away from the devices in an enclosed room. Both these noise levels were within the required sound intensity levels from the design specifications

3.8.3. Documentation

3.8.3.1. Engineering Drawings

All engineering drawings for all assemblies are presented in Figure G- 1 to Figure G- 3 (Appendix G: G.1). All engineering part drawings are presented in Figure G- 4 to Figure G- 16 (Appendix G: G.2).

3.8.3.2. Electronic Schematics

The electronic wiring schematics for the master and slave components are presented in Figure I- 1 and Figure I- 2 (Appendix I: I.1), respectively.

3.8.3.3. Program Code

All program code has been presented in Appendix J. The program that converts hexadecimal values from the LED matrix editor to Arduino friendly code for the slave program is presented in J.1, the permutation generator program that calculates the corresponding distances is presented in J.2 and final master and slave programs are presented in J.3 and J.4, respectively.

4. Results & Discussion

The final design embodiment of the master and slave devices was presented in the previous section. Through a logical and systematic design process followed throughout this project, tangible devices were able to be developed that were presented as means to achieve the aim of developing a system that has the capability of testing an athlete's agility. A set of engineering specifications were established with quantitative metric targets that needed to be met. These engineering specifications were designed according to the design requirements established earlier on in the project design stage. Therefore, a prototype that met the targets set through the design specifications could confidently agree that it therefore met the design requirements. The design specifications developed in the conceptual design stage of the project have been listed in this section, showing a comparison between actual produced metrics compared with the target and threshold values (Table 37).

Table 37: Comparison of actual design specification metrics and target values

Engineering Specification	Target (Delighted)	Threshold (Disgusted)	Prototype Metric (Actual)
Functional, Operational and Performance Requirements			
Intra-system communication	Yes	Yes	Yes
Physical connections between devices and the user	0 physical connections	0 physical connections	0 physical connections
Set-up time	2 minutes	4 minutes	≈ 4 minutes/user dependent & difficulty dependent
Steps required to set-up test	3 steps / device	8 steps / device	4 steps / device
Type of device system can connect to	2 types	1 type	2 types
Number of characters displayable on marker	36 characters	1 character	78 characters
Number of unique colours displayable on marker	9 colours	3 colours	255 colours
Functions indoors and outdoors	Yes	Yes	Yes*

Sensor response time	0.001 seconds	0.01 seconds	0.0036427 seconds
Adjustable tests difficulty	3 test difficulties	2 test difficulties	3 test difficulties
Number of markers	6 markers	4 markers	6 markers
Number of sporting professions compatible with	4 sports	1 sport	4 sports
Number of unique marker activation sequences	50 unique sequences	25 unique sequences	120 unique sequence
Maximised battery lifetime	8 hours	5 hours	> 6.43 hours
Number of programming languages used	1 language	2 languages	1 language
When user feedback is received	Peri-test	Post-test	Post-test
Number of feedback types	3 feedback type	1 feedback type	3 feedback types
Number of user commands	5 commands	3 commands	5 commands
Response time after user input command	< 1 second	< 3 seconds	0.0677 seconds
Device boot-up time	< 3 seconds	< 10 seconds	Master: ≈2 sec Slave: ≈3.7 sec
Automatic test initiation upon athlete detection	Yes	No	No
Number of previous tests stored at any given time	500 tests	1 test	1000 tests
Device power modes	2 modes	1 mode	2 modes
User can create custom tests	Yes	No	No
Sport-specific equipment tracking	Yes	No	No
Physical Requirements			
Maximum area occupied by testing zone	100 m^2	25 m^2	41.57 m^2
Length and width of each marker	150 x 150 mm	200 x 200 mm	200 x 200 mm
Height of each marker	250 mm	300 mm	250 mm
Fit system inside a carry-case	0.07125 mm^3	0.11875 mm^3	< 0.11875 mm^3
Weight of total system	12 kg	16 kg	15.4 kg
Marker load capacity	50 N	7 N	202 N
Ingress Protection	IP56	IP44	IP40

Adjustable height	2 settings	1 setting	1 setting
Impact resistance	500 N	0 N	To be assessed in future work
Workmanship and Manufacturing			
Number of parts	< 75 parts	< 150 parts	141
Internal parts enclosed	100%	90%	100%
Mechanical tools required to change/recharge battery	0 tools	1 tool	Master: 1 tool Slave: 0 tools
Safety, Regulatory and Environmental Requirements			
Safe to use	Low risk profile	Medium risk profile	Low risk profile
Extra-low voltage system	< 10 V	< 12 V	7.2 V
Noise produced by system	< 20 dB	< 60 dB	54 dB
Noise produced by audio stimulus	100 –110 dB	80 – 99 dB	82 dB
Number of loose parts	0 parts	2 parts	1 part
Human Factors			
Number of sensory pathways marker stimulus uses	2 sensory pathways	2 sensory pathways	2 sensory pathways
Number of incorporated agility components	14 components	11 components	To be assessed in trials/user dependent
Number of tests before athlete is familiar with test	1 test	2 tests	To be assessed in trials/user dependent
Reliability, Maintainability and Supportability			
Accuracy of distance measurement sensor	< 2.5 cm	< 10 cm	< 15 cm
Accuracy of time measurement	0.001 seconds	0.01 seconds	To be assessed in future work
Number of known bugs in program	0 bugs	3 bugs	To be determined in future work
Variance in test results for a given athlete	< 0.15 seconds	< 0.5 seconds	To be assessed in trials
Budget			

Manufacturing cost (excluding sundries)	< $1,000	< $1,600	$1,521.75
Schedule			
System designed and built by due date	8 October 2018^	17 October 2018^	Unfinished test development

*Able to function outdoors only when not raining

^ Specification targets for *"system designed and built by due date"* changed due to unforeseen circumstances and granted extension of thesis.

Legend

Specification met

Specification not met

Specification to be assessed in future work

A total of 51 specifications were developed for this project. However, 12 of these were considered second order specifications that were hoped to be achieved but were not expected to be completed in this project (these were likely to be implemented in future work). 39 specifications were essential requirements, required to be met in this project. Of the 39 essential first order specifications, 36 were met; 8 of the 12 second order specifications were met. Thus, a total of 44 out of the 51 design specifications were met. Therefore, 92.3% of first order specifications were met; 66.7% of second order specifications were met; and thus, 86.3% of all established design specifications were met. Although the system was unable to provide a complete test by the end of the project, a lot of the design specifications were still able to be met, which showed how successful the design really was.

The system was able to achieve intra-system communication between the various slave devices and the master. This was achieved through the high-powered nRF24L01+ modules, which were very consistent and reliable. With use of these modules and a Bluetooth Low Energy (BLE) module that could connect to either a computer or smartphone (at least two types of devices), the entire system was portable and wireless. The system was made to handle five commands, with possibility of even more once the final test is completed. It was able to respond to user commands in 0.0677 seconds, which incredibly surpassed the one second target value. The system was able to successfully implement an assisted, automated set-up procedure that could be completed in approximately four minutes, but was dependent on the user and the difficulty, as this would increase the distance and thus the amount of walking required to do. The steps required to set-up the test was measured in the number of steps per device. The prototype was able to accomplish set-up with just four steps per device.

A significant 78 characters were displayable on the LED matrix, which could have been increased even higher, however even this number surpassed the desired 36 characters. With the use of the FastLED library, 255 colours could be presented on the markers due to the HSV colour format.

In terms of functioning outdoors, the system was able to function outdoors just fine in terms of hardware components. The hardware had been specially chosen to ensure that it would work in direct sunlight. However, the design itself would not be able to be used if it was raining, as the slave device was deemed only IP40 rated. Thus, this specification was not met, which had the target of at least IP44. Each slave device was able to prevent wires from penetrating inside, but as the LED matrix was mounted on the outside with no protection, they would not even be able to have even the slightest bit of water on them. The main reason for not building an enclosing cover for the matrices was primarily due to time constraints presented. To develop a device that would be IP44 or IP56 as targeted, would be through careful design iterations and permanent designs. As further development was expected, no permanent, un-reversing design refinements were undertaken, such as the process of sealing a design so that it was waterproof.

Three test difficulties were implemented into the design, with the distances easily changeable. The number of markers used was six, so that the maximum number of permutations could be generated, with a value of 120 possible unique permutations. By using six markers, they were able to be positioned in a regular hexagon to maintain standardisation of the test. Four different balls were tested for compatibility with the markers; an Australian rules football, netball, basketball and a rugby. Thus, the test was at least compatible with four sports.

The sensor response time was found to be 0.0036427 seconds, which met the threshold target, but just fell short of the target value. The boot-up time for master device was two seconds and 3.7 seconds for the slave. This met the threshold target value but could easily be reduced by reducing the delays at the beginning when displaying the introductory stars and circles. The accuracy of the time measurement was not able to be measured, as no time was able to be determined since the test development had not been fully completed. The accuracy of the LiDAR scanner was at worst, approximately 15 cm, which seemed to have worsened after painting the markers. There could be the possibility that the paint had caused the markers to become more of a specular surface, which is not ideal for LiDAR scanner laser reflection.

A theoretical maximum battery life was able to be determined using the current consumption values tested when determining which battery to purchase. In the worst-case scenario, with maximum brightness outdoors, the slave component would last for 6.43 hours. However, during indoor use, the maximum battery for both master and slave devices was almost 20 hours.

Just one programming language was used, which was C/C++, the native language for Arduino. The number of feedback types was three; one piezo buzzer presenting an auditory feedback mechanism, an LED on the master and LED matrix on the slave, and the laser diode module during assisted set-up. A speaker and the LED matrix were used as the visual stimuli, providing two sensory pathways to the athlete.

As no test was able to be performed, it was not possible to specify the number of incorporated agility components the test would cause the athelete to use, which is a shame because this was the specification with highest importance. However, as already mentioned when discussing content validity in Section 3.8.2.3, the incorporated agility components that would be implemented when running the test could be estimated based on the milestones on what the prototype devices had been able to accomplish so far.Manufacturing cost was deemed the second most important design specification. This was met, with a total spending of $1,521.75, which was below the threshold target of $1,600.

Completing the project on time was the third highest important specification. This specification was given such a high importance because it was known that the project had a lot of work to be completed from the very beginning. Thus, it was known that it would be difficult to get everything completed on time. One of the main reasons as to why the prototype likely did not get to be finished was due to receiving the high-powered nRF24L01+ modules so late in the project (with exactly one week remaining for thesis submission), due to significant problems with shipping and transit of lost packages and companies that did not process the order fast enough. A few more days working on the prototype and the test would likely have been up and running successfully. It was already confirmed that the slave component was able to process all communication data and run the test as required. Tests had been performed of inputting arbitrary values in place of messages sent by the master component, where the slave was able to pinpoint where its position in the permutation sequence and what character it would need to display on the LED matrix. The program had been written so that the user received feedback post-test, thus this specification would be met when the rest of the test would be resolved.

The maximum area occupied by the testing zone was not as large as desired (actual 41.57m^2), but still was larger than the AFL Agility run. The major limitation was that the LiDAR scanner was not able to reliably detect the markers further than 2m away. This could have been due to multiple factors, such as the unstable slip ring, the paint choice for the slave, the odd geometry of the slave or the surface area of the slave. The slave met the specifications for length, width and height, as well as fitting within the suitcase containment with padding around all components. The weight of the entire system was quite heavy, being 15.4 kg, but 4.7 kg was the weight of the suitcase alone. It was confirmed the marker load capacity was capable of withstanding at least an approximate 202 N load.

A total of 141 parts were used to develop the system; this value was not including sundries such as fasteners, wires, or terminal connector strips. The master component had 33 parts, while each slave component had 18 parts. This specification had a high importance, so it was excellent that it was able to be met. 100% of internal parts were enclosed; the LED matrices were not considered "internal parts".

The total system was deemed to have a low-risk profile. The system was of extra-low voltage, using a 7.2 V battery. The noise produced by the system was 54 dB, which was under the 60 dB threshold limit. The noise produced by the stimulus only just met specification, being 82 dB; although, indoors this sound was extremely loud

In terms of second order specifications that met the metric targets, the master was able to store 1000 test values and had two device power modes; a normal functioning mode and a sleep mode. No automatic test initiation was implemented from detecting the athlete, no custom tests could be developed, and no sport-specific equipment tracking was implemented. Only one height setting was implemented (the height as is), and impact resistance was a specification to look at integrating further in the future. The master required at least one tool to open the enclosure lid (a flat-head screwdriver), because after painting, it became a lot more difficult to remove it. The slave component only required a key to remove the snap-fit cover for recharging the battery. The total number of loose parts was just the battery in the slave component. Although not ideal, double-sided tape had to be used to stick the speakers down into place as there was no time to develop mounts for them. The number of tests before an athlete is familiar with the test is something that needs to be assessed in trials, with a similar story for the variance in test results for a given athlete. The number of bugs in the program was to be determined in future work, as it was difficult to determine exactly how many bugs were present since the test could not be conducted.

4.1. Project Significance

This project had major significance as it started as a simple idea and developed into a tangible prototype system. Although the test development was not complete yet, it was able to incorporate sporting equipment within the test, with hope to have a standardised test that could be used across sporting institutions and sporting clubs. At the time of writing, no agility test uses integration of sport-specific equipment within the test and can remain standardised. Additionally, no test can assist a user lay their test out in a configuration such as how this prototype had successfully implemented. Although there were some issues with the LiDAR scanner accuracy, the author is sure that this is something that can be resolved in future work.

Currently implemented standardised tests such as the Illinois Agility, AFL Agility Run, 5-0-5, T-Test (T-Drill), Pro-Agility (5-10-5 Shuttle) and the 3-Cone Drill (L-Run) all only use physical components of agility, as they involve just pre-planned agility. The proposed agility test will, when complete, test all components of agility and thus provide a means for truly measuring agility performance in athletes. It is hoped that this test will one day become incorporated into testing batteries, combines or simple training regimes.

The project also has significance that it could be used in a rehabilitation setting. Through a customisable interface for the user, it could be programmed to conduct tests that are suited for rehabilitation of people who may suffer from some form of disability. The test may also provide usefulness in academic or research studies that require the testing of athletes.

4.2. Limitations of the Project

4.2.1. Uncomplete Test

Unfortunately, due to time constraints, a working test was unable to be achieved. This is the most significant limitation of this project by far. With a few more days of testing the prototypes, it is believed that the system will be able to perform a running test.

4.2.2. Accuracy of the LiDAR Scanner

It was determined that there existed some accuracy issues with the LiDAR scanner (±15 cm or so). During initial testing phases the results looked promising, however, after several changes made to the designs, the accuracy significantly decreased. The LiDAR scanner values were not as stable as they had been during earlier testing phases; this could be due to the painted surface of the markers. Accuracy issues with the device that assists in setting up the layout of the system is not good at all, as the aim is to try and make the test as standardised as possible. The LiDAR scanner will need to produce more reliable and accurate results for this to occur.

4.2.3. Mechanical Design

The mechanical design had several flaws. One of these flaws was that the slip ring was not stable and as a result, the LiDAR assembly rocked side to side slightly. This was most likely the reason as to why the LiDAR scanner was only able to detect the markers approximately 2 m away; pushing forwards on the assembly, directing it downwards caused the LiDAR scanner to detect the marker further away. Thus, it would be more beneficial to use a part that would remain parallel with the ground, such as a Lazy Susan bearing. During development, this option was not known. The rocking of the LiDAR scanner assembly could be one of the reasons why the LiDAR scanner values were not as stable as they had been previously.

Another important thing to note about the mechanical design was that assembling the devices, especially the master device, was a difficult process as there were many small gaps needed to fit fingers through or awkwardly reach around to fasten a component in.

4.2.4. Wiring

The use of header wiring pins meant that the wiring connections between hardware components and the Arduino were not optimal, in that they were very loose and easy to come off. This could be a significant problem when testing, as there would be the chance of markers being knocked and thus wires becoming loosened. Being a prototype, things were constantly changing, which was why headers were very useful, as they are parts which allow electronic components to detach and reattach as needed. However, a more permanent design should be employed so that the wiring is not loose, and components will not come off easily.

4.2.5. Ingress Protection

The prototype was unable to have the ingress protection desired (IP56 target, or IP44 threshold target). As a result, the prototype could not be used in the rain, which was a major limitation.

Further refinement and optimisation of the design could be completed so that it could become IP56 rated.

4.2.6. Limited Patent Search

The author recognises that a thorough patent search was not conducted during the market analysis and research stage of the project. This meant that there is the possibility that a novel system capable of testing an athletes physical, perceptual and decision-making factors of agility is under works for development. However, the author wished to evaluate products that were currently on the market and knew that a patent search could be completed after this stage of the project.

4.2.7. 3D Printing

A lot of the parts are constructed from 3D Printing. Minimal commercially available products were used in terms of the mechanical design, this therefore increases the complexity of manufacturing. Additionally, the 3D printing material used was PLA, which is known to have a low glass transition temperature. As a result, there is a chance that if left in direct sunlight for too long, the 3D print may melt or warp. A more optimal printing material would be Polyethylene Terephthalate Glycol (PETG) filament, which has a higher glass transition temperature than PLA. Unfortunately, PLA was the only material that Flinders University was loading their printers with. If the prototype were to get commercial appeal, it would be likely that a whole different manufacturing method would be used, which would mean that the mechanical design would need to be totally redesigned.

5. Future Work

5.1. Mechanical Design Component

5.1.1. Refinement and Optimisation

As the specification for ingress-protection was not met, refinements and optimisations should be made to the mechanical design of both slave and master component such that it has protection against water and dust ingression. By incorporating these features, it would allow the design to function outdoors even in rain, which would be optimal for athletes who may be performing a training regime.

Design minimisation is another factor to consider regarding the mechanical design component. The author is aware that the slave components are quite large, and the master component is relatively large. Reducing the size of the designs would allow them to fit within a smaller containment, providing better portability. Smaller designs would normally correlate with a lighter design too, which would reduce the total weight of the system.

To reduce the size of the master component, a flat stepper motor could be used. This would significantly reduce the weight as well as the height of the device.

5.2. Electronic Component

5.2.1. Implementation of Printed Circuit Boards (PCBs)

The prototype shield boards were good to use for a proof-of-concept prototype. However, future work should be conducted to employ developing PCBs for a design that is more aesthetic and reliable.

5.2.2. Introduction of Radio Frequency Identification (RFID)

Future work could provide the possibility of implementing RFID scanning. That way, athletes could wear a wristband or a clip around their shorts so that the system could identify athletes performing the test.

5.2.3. Reducing Current Draw

The master used approximately 0.251 A per hour on average during use. The slave component used approximately 0.224 A per hour on average while used inside, and 0.778 A on average during outside use. Although the inside use was not too much of an issue in terms of power consumption, the outside use current draw is quite high. A method for reducing the current draw could be to reduce the LED matrix pixel size from 8x8 down to 7x5, which was determined to be the minimum to display all alphanumeric characters. A lot less LEDs would be used doing this, thereby saving power consumption.

5.3. Software Component

5.3.1. Program Optimisation

Significant optimisation and refinement will need to be completed regarding the master and slave programs. Of course, the program needs to be finished before this is conducted. But once a run has successfully been implemented, then the program should be tested for bugs.

More commands could be added to the master component through the Bluetooth module, such as changing the brightness of the LED matrices (by communicating with the slave components).

5.3.2. Development of Mobile Phone Application

Development of a mobile phone application for the system would be extremely ideal. The literature review found that many of the competing products had software options that allowed the user to simply download the application and run the hardware from their smartphone. A mobile application would be able to provide a sleek and easy to use user interface, which could have a lot of commands through the simple press of a button.

5.4. Test Component

5.4.1. Test Validity and Reliability

Due to time constraints and the load of the project through design and development, the test was unable to be tested on real athletes to determine the reliability and validity of the test.

Once the program has been finished and once it has been optimised and bugs have been removed, it would be highly desirable to test the validity and reliability of the proposed agility test by comparing the results between highly-skilled athletes, their lesser skilled counter-parts and those who do not have a professional sporting background. Additionally, the results should be compared with well-established tests such as the Illinois Agility or the AFL Agility Run to see if there are any correlations between the values.

5.4.2. Including a Target

In the current test implemented by the system, it requires the athlete to remove the sport-specific game ball from each marker and hand-pass it as they would in their sporting profession. The problem here is that the athlete will need to hand-pass the ball to either a team member, or they would simply hand-pass it to a random location in space. In the instance of hand-passing the ball to a team member, each test would require six additional available players to stand around, which could be time better utilised. In the instance where a hand-pass of the ball is random in space, this means that once the ball is removed, there is no decision making, or real test of skill of agility, since they could pass the ball in any direction and not have to think about its consequences. However, to further incorporate additional aspects of agility and skill in their sporting profession, targets could be set up where the athlete needs to aim to hand-pass into or on these targets. In the instance of AFL, pop-up hand-ball targets (Figure 197) (Sherrin, Scoresby, Australia; HART Sport, Melbourne, Australia) are excellent examples of such targets. The position at which the ball hits the target could be used as a point base, for example, hitting the 10 could give 10 points, hitting the 7 could give seven points and the 5 would give five points. This would mean the scoring would have to be altered slightly, so instead of the output as a measure of time, the time factor could be multiplied by a weighting factor to obtain a relative point base. A limitation to this idea, is that it would make it very hard to then determine the correlation of the test between other standardised tests since they all infer agility through the measurement of time. Alternatively, specific number values could negate time from the athlete's test; for example, hitting a 10 could negate one second from the test, a 7 could negate 0.4 seconds and a 5 could negate 0.2 seconds.

Figure 197: Pop-up hand ball targets (HART Sport, 2018; Russell Corporation, 2018)

Of course, introducing a new variable into the system, the element of accuracy of hand-passing the ball to the specified target, could bring potential challenges. For example, the test may need to ensure the ball is hand-passed behind a line and the target is positioned at the same distance for each test, so that an athlete is not too close and to ensure standardisation. Another major issue is that it would make more sense for the type of target to vary depending on the sport, where the targets displayed in Figure 197 may be applicable for Australian football and rugby, but not for basketball or netball, so it would make it hard to compare the test between different sports and this would therefore reduce standardisation. There could be the possibility of designing a target that is compatible with all sports and remains fair to each sporting profession.

5.4.3. Test Customisation

It would be beneficial to allow customised tests to be programmed into the system so that it can also be used in training sessions, which a coach may think could work well with their current regime. Further optimisation would need to be completed on the program to implement this feature.

5.4.4. Use in Rehabilitation

The structure of the system allows it to have significance in multiple areas within the field of rehabilitation. As the system can present a random stimulus in the form of auditory and visual pathways, this could help with tests or exercises involving a patient needing to react to a stimulus. As the LED matrix can display random characters of many different colours, a test could be developed that requires a patient to find a specific character or specific coloured character within a set of other characters. It could also be used as a form of memory game.

6. Conclusion

The primary aim of this project was to design and develop a novel system capable of testing all components of an athlete's agility, with a primary focus for capability with Australian rules football; but with the possibility of also incorporating other sports such as netball, basketball or rugby. The initial idea was presented through the SHAPE Research Centre and associate supervisor, Dr. Elliott, who has a wealth of knowledge within the field of Sport, Health and Physical Education. Through experiencing firsthand the testing protocols implemented on athletes, he was able to identify a gap within the market, that no currently implemented standardised test can incorporate all components of agility whilst also incorporating the use of sport-specific equipment such as a game ball. A thorough literature review found that even reactive agility tests, which have proven a reliable and valid method for measuring agility performance, incorporating most components of agility, have their limitations, including non-standardisation and expensive equipment required to perform the test. Even a competitive market analysis found that products currently on the market can perform reactive agility tests but are unable to incorporate a sport-specific ball and remain standardised. The analysis also found that no product was able to assist a user in setting up the test in a layout, such as the system informing the user exactly where they need to position the corresponding components. The closest to assisted set-up regarding competitive products was timing gates, which assist the user in aligning the receiver with the emitter. Thus, through identification of several needs through Dr. Elliott, the literature review and the competitive market analysis, a new, novel sports agility testing system was proposed for design and development.

An engineering design process was followed through the design and development of the proposed product. The clear methodology followed had enabled a logical and systematic progression through a problem definition stage, which established the needs for the system, clarified objectives and determined design requirements through guidance by Dr. Elliott, the literature review conducted and the competitive market analysis. The design requirements were categorised as either first or second order, which assisted in differentiating the essential requirements from desirable ones. Through the conceptual design stage, a function structure was established through functional diagrams, development of a function objective tree and functional decomposition. The analysis of all work completed until this point allowed the establishment of a set of design specifications through the performance-specification method that correlated with the design requirements. Metric delighted and disgusted (threshold) target values were assigned to each specification, which was then used in the quality-function deployment process to develop a House of Quality chart. Correlations and conflicts between specifications were found, as well as the correlations between requirements and specifications. Through this, it could be determined the most significantly important specifications, such that these would be ideally focused on during the conceptualisation and developmental stages of the project. By initial brainstorming sessions, the flow of creativity began, with ideation and conceptualising successively following.

Using a morphological chart, design means were established that correlated to a set of specific functions to achieve in terms of mechanical, electrical and software design. These alternatives could be evaluated by weighing them against a set of criteria correlated to the required function to achieve, comparing assigned values in a decision matrix; which presented a numerical approach for decision making during the selection procedure. With the most optimal alternatives determined, these could be integrated into a final solution concept. The final components to purchase, corresponding to the solution means were established and this served as a vital stage for further development and design embodiment into 3D CAD model assemblies. The CAD models were deemed essential for visualisation of the master and slave layout solution concept, to ensure integration between parts would be successful.

The final stage of development entailed mechanical, electrical and software embodiment, to produce a tangible product solution that would accomplish the functions required to meet the aims and objectives of the project. This stage of the project was deemed most difficult, as it incorporated many new concepts and skills not learnt before, especially regarding electronic componentry. The process was iterative in nature, where designs were optimised and refined as required. Through integration of the various components, a final prototype system was developed, which incorporated one master device and six slave devices; it is these components that made up the test to evaluate the agility performance in an athlete.

Unfortunately, due to significant time constraints with the amount of work that needed to be completed, a final test was not able to be implemented; the programs developed for master and slave devices were extremely close to achieving the desired objective but just fell short. The original nRF24L01+ communication modules did not work as they should and so some high-powered modules were ordered. Due to significant issues involving shipping and transit, the modules arrived one week before thesis submissions. As a result, there was some issues with communication between the master and slave components that were not able to be resolved due to insignificant time remaining. Provided with more time, this issue would have been able to be resolved and a working system would likely have been achieved. Nevertheless, the design methodology followed up until this point, had produced an impressive system as is. The system was able to successfully assist a user in automated set-up of configuring the slave components in a specific layout (the system incorporates the use of a regular hexagon), a feature that no other competing product can achieve. Additionally, the system incorporates the implementation of sport-specific equipment within the test, with the hope to maintain standardisation; another feature that other products on the market have not incorporated. The system had been tested without the communication section and it had successfully detected removal of a ball in direct sunlight, as well as a hand swiftly moved across the sensor without any false detections.

The proposed agility testing system achieved 92.3% of first order specifications and a total of 86.3% of both first and second order specifications. Although the test was not completely fully-functional by the end of the project, it was still identified that it still had a large amount of capabilities and features in which it could perform; this was reflected through a qualitative comparison with the engineer design specifications. It was through this; the design had been confirmed to meet many of the design requirements specified and identified during the initial stages of the project, thus meeting almost all objectives and aims. Therefore, this project has had great significance, where a proof-of-concept prototype has been deemed highly successful, and will act to pave the way towards developing a test that could possibly become a new standardised way of evaluating an athlete’s agility performance across a range of sporting institutions and clubs, for use with a variety of field and court sports, athletic abilities and incorporation within testing batteries, combines and training regimes.

7. References

Abernethy, B., Gill, D.P., Parks, S.L. and Packer, S.T. (2001), “Expertise and the perception of kinematic and situational probability information”, *Perception*, Vol. 30 No. 2, pp. 233–252.

Abernethy, B. and Russell, D.G. (1987), “The Relationship Between Expertise and Visual Search Strategy in a Racquet Sport”, *Human Movement Science*, Vol. 6 No. 4, pp. 283–319.

Adafruit. (2018), “Store”, available at: https://www.adafruit.com/ (accessed 20 May 2018).

ALGE-Timing GmbH. (2018), “ALGE-Timing Wireless Training Network”, Vienna, available at: https://alge-timing.com/AlgeTiming_V2/en/home.html (accessed 26 July 2018).

American Educational Research Association, American Psychological Association, and National Council on Measurement in Evaluation. (2014), *Standards for Educational and Psychological Testing*, American Educational Research Association, Washington, DC.

Arduino. (2018), “Arduino”, Ivrea, Italy, available at: https://www.arduino.cc/ (accessed 24 May 2018).

Arm Limited. (2018), “Mbed”, available at: https://www.mbed.com/en/ (accessed 24 May 2018).

Australian Communications Industry Forum. (2006), “Installation Requirements for Customer Cabling (Wiring Rules) [AS/ACIF S009:2006]”, Standards Australia, NSW.

Australian Football League. (2017), *Laws of Australian Football 2017*, Melbourne.

Australian Government. (2011), “Work Health and Safety Act Code of Practice 2011: Managing Noise and Preventing Hearing Loss at Work (Repealed)”, ACT, available at: https://www.legislation.gov.au/Details/F2011L02804/Html/Volume_2 (accessed 11 October 2018).

Baechle, T.R. (1994), *Essentials of Strength Training and Conditioning*, 3rd ed., Human Kinetics, Champaign, IL.

Barber, O.R., Thomas, C., Jones, P.A., Mcmahon, J.J. and Comfort, P. (2016), “Reliability of the 505 change-of-direction test in netball players”, *International Journal of Sports*

Physiology and Performance, Vol. 11 No. 3, pp. 377–380.

Barrow, H.M. and McGee, R. (1979), *A Practical Approach to Measurement in Physical Education*, 3rd ed., Henry Kimpton Publishers, London.

Baumgartner, T.A. (1968), "The applicability of the spearman-brown prophecy formula when applied to physical performance tests", *Research Quarterly of the American Association for Health, Physical Education and Recreation*, Vol. 39 No. 4, pp. 847–856.

Baumgartner, T.A. and Jackson, A.S. (1983), *Measurement for Evaluation in Physical Education*, 2nd ed., Brown Company Publishers, Dubuque.

Benet, G., Blanes, F., Simó, J.E. and Pérez, P. (2002), "Using infrared sensors for distance measurement in mobile robots", *Robotics and Autonomous Systems*, Vol. 40 No. 4, pp. 255–266.

Benvenuti, C., Minganti, C., Condello, G., Capranica, L. and Tessitore, A. (2010), "Agility assessment in female futsal and soccer players", *Medicina (Kaunas)*, Vol. 46 No. 6, pp. 415–420.

Bidaurrazaga-Letona, I., Carvalho, H.M., Lekue, J.A., Badiola, A., Figueiredo, A.J. and Gil, S.M. (2015), "Applicability of an agility test in young players in the soccer field", *Revista Brasileira de Medicina Do Esporte*, Vol. 21 No. 2, pp. 133–138.

Bishop, P.A. (2008), *Measurement and Evaluation in Physical Activity Applications: Exercise Science, Physical Education, Coaching, Athletic Training & Health*, Routledge, New York.

Born, D.P., Kunz, P. and Sperlich, B. (2017), "Reliability and validity of an agility-like incremental exercise test with multidirectional change-of-direction movements in response to a visual stimulus", *Physiological Reports*, Vol. 5 No. 9, available at:https://doi.org/10.14814/phy2.13275.

Born, D.P., Zinner, C., Düking, P. and Sperlich, B. (2016), "Multi-directional sprint training improves change-of-direction speed and reactive agility in young highly trained soccer players", *Journal of Sports Science and Medicine*, Vol. 15 No. 2, pp. 314–319.

Brahim, M. Ben, Bougatfa, R. and Amri, M. (2013), "Reliability , Validity and Minimal Detectable Change of a New Multi-Change of Directionagility Test for Soccer Players", Vol. 3 No. 4, pp. 190–196.

Brower Timing Systems. (2018), “TC Timing System”, Draper, available at: http://browertiming.com/index.php (accessed 26 July 2018).

Chaalali, A., Rouissi, M., Chtara, M., Owen, A., Bragazzi, N.L., Moalla, W., Chaouachi, A., et al. (2016), “Agility training in young elite soccer players: Promising results compared to change of direction drills”, *Biology of Sport*, Vol. 33 No. 4, pp. 345–351.

Chalmers, S. and Magarey, M. (2016), “Annual improvement in fitness test performance for elite junior Australian football cohorts.”, *Journal of Science and Medicine in Sport / Sports Medicine Australia*, Sports Medicine Australia, Vol. 19 No. 10, pp. 843–7.

Chaouachi, A., Brughelli, M., Chamari, K., Levin, G.T., Abdelkrim, N. Ben, Aurencelle, L. and Astagna, C. (2009), “Lower limb maximal dynamic strength and agility determinants in elite basketbal players.”, *The Journal of Strength and Conditionning Research*, Vol. 23 No. 5, pp. 1570–1577.

Chelladurai, P. (1976), “Manifestations of agility”, *Canadian Association of Health, Physical Education, and Recreation*, Vol. 2, pp. 36–41.

Chu, D.C. and Vermeil, A. (1983), “The Rationale for Field Testing”, *NSCA Journal*, Vol. 5 No. 2, pp. 35–36.

Čoh, M., Vodičar, J., Žvan, M., Šimenko, J., Stodolka, J., Rauter, S. and Maćkala, K. (2018), “Are Change-of-Direction Speed and Reactive Agility Independent Skills Even When Using the Same Movement Pattern?”, *Journal of Strength and Conditioning Research*, Vol. 32 No. 7, pp. 1929–1936.

Cooke, K., Quinn, A. and Sibte, N. (2011), “Testing speed and agility in elite tennis players”, *Strength and Conditioning Journal*, Vol. 33 No. 4, pp. 69–72.

Core Electronics. (2018), “Store”, Newcastle, available at: https://core-electronics.com.au/ (accessed 17 June 2018).

Coulson, M. and Archer, D. (2009), *Practical Fitness Testing: Analysis in Exercise and Sport*, A&C Black, London.

Cox, R.H. (2011), *Sport Psychology: Concepts and Applications*, 7th ed., McGraw Hill

Companies, New York, available at:https://doi.org/10.1093/oxfordhb/9780195342314.013.0030.

Cross, N. (2000), *Engineering Design Methods: Strategies for Product Design*, edited by 3, Wiley, Chichester.

D'Auria, S., Tanner, R., Sheppard, J. and Manning, J. (2006), "Evaluation of various methodologies used to assess sprint performance.", *Proceedings of the Australian Institute of Sport Applied Physiology Conference.*, ACT.

Dawes, J. and Roozen, M. (2012), *Developing Agility and Quickness*, Human Kinetics, Champaign, IL.

Delextrat, A. and Cohen, D. (2009), "Strength, power, speed, and agility of women basketball players according to playing position", *Journal of Strength and Conditioning Research*, Vol. 23 No. 7, pp. 1974–1981.

Digi-Key Electronics. (2018), "Store", Thief River Falls, available at: https://www.digikey.com.au/ (accessed 15 August 2018).

Digilent Inc. (2018), "chipKIT Max32", available at: https://store.digilentinc.com/ (accessed 24 May 2018).

Draper, J.A. and Lancaster, M.G. (1985), "The 505 test: A test for agility in the horizontal plane", *Aust J Sci Med Sport*, Vol. 17 No. 1, pp. 15–18.

Dym, C.L., Little, P. and Orwin, E.J. (2014), *Engineering Design: A Project-Based Introduction*, 4th ed., Wiley, Hoboken, available at:https://doi.org/10.1017/CBO9781107415324.004.

Earl, B. (2015), "Adafruit Motor Selection Guide", Adafruit, available at: https://learn.adafruit.com/adafruit-motor-selection-guide/types-of-motors (accessed 4 August 2018).

Earp, J.E. and Newton, R.U. (2012), "Advances in electronic timing systems: Considerations for selecting an appropriate timing system", *Journal of Strength and Conditioning Research*, Vol. 26 No. 5, pp. 1245–1248.

element14. (2018), "Store", Premier Farnell Limited, NSW, available at: https://au.element14.com/.

Farrow, D. (2010), “A Multi-Factorial Examination of the Development of Skill Expertise in High Performance Netball”, *Talent Development and Excellence*, Vol. 2 No. 2, pp. 123–135.

Farrow, D. and Abernethy, B. (2002), “Can anticipatory skills be learned through implicit video-based perceptual training?”, *Journal of Sports Sciences*, Vol. 20 No. 6, pp. 471–485.

Farrow, D., Young, W. and Bruce, L. (2005), “The development of a test of reactive agillity for netball: a new methodology”, *Journal of Science and Medicine in Sport*, Vol. 8 No. 1, pp. 52–60.

Fessi, M.S., Makni, E., Jemni, M., Elloumi, M., Chamari, K., Nabli, M.A., Padulo, J., et al. (2016), “Reliability and criterion-related validity of a new repeated agility test”, *Biology of Sport*, Vol. 33 No. 2, pp. 159–164.

Fiorilli, G., Iuliano, E., Mitrotasios, M., Pistone, E.M., Aquino, G., Di Costanzo, A., Calcagno, G., et al. (2017), “Are change of direction speed and reactive agility useful for determining the optimal field position for young soccer players?”, *Journal of Sports Science and Medicine*, Vol. 16 No. 2, pp. 247–253.

FITLIGHT Corp. (2018), “FITLIGHT Trainer”, Ontario, available at: https://www.fitlighttraining.com/ (accessed 26 July 2018).

Freelap SA. (2018), “Freelap Timing System”, Fleurier, available at: http://www.freelap.ch/ (accessed 26 July 2018).

French, M.J. (1971), *Engineering Design: The Conceptual Stage*, Heinemann Educational, London.

Fried, L. (2013), “All About Batteries”, Adafruit, New York, available at: https://learn.adafruit.com/all-about-batteries/ (accessed 12 September 2018).

Fusion Sport. (2018), “SmartSpeed”, Brisbane, available at: https://www.fusionsport.com/ (accessed 26 July 2018).

Gabbett, T. and Benton, D. (2009), “Reactive agility of rugby league players”, *Journal of Science and Medicine in Sport*, Vol. 12 No. 1, pp. 212–214.

Gabbett, T., Kelly, J., Ralph, S. and Driscoll, D. (2009), “Physiological and anthropometric characteristics of junior elite and sub-elite rugby league players, with special reference to starters and non-starters", *Journal of Science and Medicine in Sport*, Vol. 12 No. 1, pp. 215–222.

Gabbett, T.J., Kelly, J.N. and Sheppard, J.M. (2008), "Speed, change of direction speed, and reactive agility of rugby league players", *Journal of Strength and Conditioning Research*, Vol. 22 No. 1, pp. 174–181.

Gammon, N. (2012), "Power saving techniques for microprocessors", available at: https://www.gammon.com.au/forum/?id=11497 (accessed 29 September 2018).

Garcia, D. (2018), "FastLED Arduino Library", available at: http://fastled.io/ (accessed 7 October 2018).

Garmin Corporation. (2016), "Lidar Lite v3 Operatiion Manual and Technical Specifications", New Taipei City, available at: https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf (accessed 7 June 2018).

Garrett, J.W. (1971), "The adult human hand: some anthropometric and biomechanical considerations.", *Human Factors*, Vol. 13 No. 2, pp. 117–131.

Glover, T.J. (1993), *Pocket Ref*, Sequoia Publishing, Littletoin, CO.

Gratton, C. and Jones, I. (2004), *Research Methods for Sports Studies*, Routledge, London.

Gray-Nicolls Sports. (2018), "Steeden NRL Match Ball", Cheltenham, available at: https://steeden.gnsports.com/ (accessed 6 October 2018).

Hachana, Y., Chaabène, H., Ben Rajeb, G., Khlifa, R., Aouadi, R., Chamari, K. and Gabbett, T.J. (2014), "Validity and reliability of new agility test among elite and subelite under 14-soccer players", *PLoS ONE*, Vol. 9 No. 4, pp. 1–6.

Haff, G.G. and Triplett, T.N. (2016), *Essentials of Strength Training and Conditioning*, 4th ed., Human Kinetics, Champaign, IL.

Haik, Y. and Shahin, T.M. (2011), *Engineering Design Process*, 2nd ed., Cengage Learning, Stamford.

HART Sport. (2018), "HART Sport Target", Brisbane, available at:

https://www.hartsport.com.au (accessed 6 October 2018).

Haugen, T.A., Tønnessen, E., Svendsen, I.S. and Seiler, S. (2014), “Sprint time differences between single- and dual-beam timing systems”, *Journal of Strength and Conditioning Research*, Vol. 28 No. 8, pp. 2376–2379.

Hengpattanapong, T. (2018), “Red led uppercase, lowercase English alphabet and number [Image]”, 123RF Limited, available at: https://www.123rf.com/photo_60300623_stock-vector-red-led-uppercase-lowercase-english-alphabet-and-number.html (accessed 6 August 2018).

Henry, G., Dawson, B., Lay, B. and Young, W. (2011), “Validity of a reactive agility test for Australian football”, *International Journal of Sports Physiology and Performance*, Vol. 6 No. 4, pp. 534–545.

Henry, G., Dawson, B., Lay, B. and Young, W. (2012), “Effects of a feint on reactive agility performance”, *Journal of Sports Sciences*, Vol. 30 No. 8, pp. 787–795.

Henry, G.J., Dawson, B., Lay, B.S. and Young, W.B. (2013), “Decision-making accuracy in reactive agility: Quantifying the cost of poor decisions”, *Journal of Strength and Conditioning Research*, Vol. 27 No. 11, pp. 3190–3196.

Hermassi, S., Fadhloun, M., Chelly, M.S. and Bensbaa, A. (2011), “Relationship between agility T-test and physical fitness measures as indicators of performance in elite adolescent handball players”, *PEDAGOGICS, PSYCHOLOGY, MEDICAL-BIOLOGICAL PROBLEMS OF PHYSICAL TRAINING AND SPORTS*, Vol. 5, pp. 125–131.

Hoffman, J. (2006), *Norms for Fitness, Performance, and Health*, Human Kinetics, Champaign, IL.

Hopkins, W.G. (2000), “Measures of Reliability in Sports Medicine and Science Measurement Error & Reliability”, Vol. 30 No. 4, pp. 1–15.

Hua, A.C.C. and Syue, B.Z.W. (2010), “Charge and discharge characteristics of lead-acid battery and LiFePO4 battery”, *2010 International Power Electronics Conference - ECCE Asia -, IPEC 2010*, available at:https://doi.org/10.1109/IPEC.2010.5544506.

IEC. (2013), *IEC 60529:1989+A1:1999+A2:2013 - Degrees of Protection Provided by Enclosures*

(IP Code).

INF. (2018), *Rules of Netball 2018 Edition*, Manchester, available at: https://s3-ap-southeast-2.amazonaws.com/netball-wp-assets/wp-content/uploads/2013/09/11102630/INF-Rules-of-Netball-2018-Edition.pdf.

Inglis, P. and Bird, S.P. (2016), "Reactive Agility Tests - Review and Practical Applications.", *Journal of Australian Strength and Conditioning*, Vol. 24 No. 5, pp. 62–69.

ISL Products International. (2017), "Servo Motor Fundamentals", available at: https://www.islproducts.com/designnote/servo-motor-fundamentals/ (accessed 4 August 2018).

Jeffreys, I. (2011), "A Task-Based Approach to Developing Context- Specific Agility", *Strength and Conditioning Journal*, Vol. 33 No. 4, pp. 52–59.

Johnson, B.L. and Nelson, J.K. (1986), *Practical Measurements for Evaluation in Physical Education*, 3rd ed., Burgess Publishing Company, Minneapolis.

Kaplan, T., Erkmen, N. and Taskin, H. (2009), "The evaluation of the running speed and agility performance in professional and amateur soccer players", *Journal of Strength and Conditioning Research*, Vol. 23 No. 3, pp. 774–778.

Keogh, J.W.L., Weber, C.L. and Dalton, C.T. (2003), "Evaluation of anthropometric, physiological, and skill-related tests for talent identification in female field hockey.", *Canadian Journal of Applied Physiology = Revue Canadienne de Physiologie Appliquee*.

Kridner, J., Coley, G., Day, R.P.J. and Foundation, B. or. (2017), "BeagleBone Black System Reference Manual Rev C.x", available at: https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual (accessed 24 May 2018).

Kukureka, S.N., Chen, Y.K., Hooke, C.J. and Liao, P. (1995), "The wear mechanisms of acetal in unlubricated rolling-sliding contact", *Wear*, Vol. 185 No. 1–2, pp. 1–8.

Kutlu, M., Yapici, H. and Yilmaz, A. (2017), "Reliability and Validity of a New Test of Agility and Skill for Female Amateur Soccer Players", *Journal of Human Kinetics*, Vol. 56 No. 1, pp. 219–227.

Kutlu, M., Yapici, H., Yoncalik, O. and Çelik, S. (2012), "Comparison of a new test for agility and skill in soccer with other agility tests", *Journal of Human Kinetics*, Vol. 33 No. 1, pp. 143–150.

Lacy, A.C. (2011), *Measurement and Evaluation in Physical Education and Exercise Science*, 6th ed., Pearson Education, San Francisco.

Leon-Carlyle, C., Kivi, D., Przysucha, E. and Paterson, G. (2012), *Reliability and Validity of a New Test of Agility in Male Soccer Players*, Bibliothèque et Archives Canada. Lakehead University, Ontario, Canada, available at: https://knowledgecommons.lakeheadu.ca/handle/2453/203.

Liefeith, A., Kiely, J., Collins, D. and Richards, J. (2018), "Back to the Future – in support of a renewed emphasis on generic agility training within sports- specific developmental pathways within sports-specific developmental pathways", *Journal of Sports Sciences*, Routledge, Vol. 36 No. 19, pp. 2250–2255.

Lightware Optoeletronics. (2017), "SF02/F Laser Rangefinder Product Manual Rev12", available at: https://www.parallax.com/sites/default/files/downloads/28043-SF02-Laser-Rangefinder-Manual-Rev-12.pdf (accessed 7 June 2018).

Little Bird Company. (2018a), "Raspberry Pi: PiAustralia", available at: https://raspberry.piaustralia.com.au/ (accessed 24 May 2018).

Little Bird Company. (2018b), "Store", available at: https://www.littlebird.com.au/ (accessed 7 June 2018).

Little, T. and Williams, A.G. (2005), "Specificity of acceleration, maximum speed, and agility in professional soccer players", *Journal of Strength and Conditioning Research*, Vol. 19 No. 1, pp. 76–78.

Lloyd, D., Ackland, T.R. and Cochrane, J. (2009), "Balance and Agility", in Ackland, T.R., Elliott, B.C. and Bloomfield, J. (Eds.), *Applied Anatomy and Biomechanics in Sport.*, 2nd ed., Human Kinetics, Champaign, IL.

Lockie, R.G., Jeffriess, M.D., McGann, T.S., Callaghan, S.J. and Schultz, A.B. (2014), "Planned and reactive agility performance in semiprofessional and amateur basketball players", *International Journal of Sports Physiology and Performance*, Vol. 9 No. 5, pp. 766–771.

Mathews, D.K. (1978), *Measurement in Physical Education*, 5th ed., Saunders, Philadelphia.

Matlák, J., Tihanyi, J. and Rácz, L. (2016), “Relationship between Reactive Agility and Change of Direction Speed in Amateur Soccer Players”, *Journal of Strength and Conditioning Research*, Vol. 30 No. 6, pp. 1547–1552.

Matthews, C. and Institution of Mechanical Engineers. (2012), *Engineers’ Data Book*, 4th ed., Wiley, Chichester.

McGee, K.J. and Burkett, L.N. (2003), “The National Football League combine: A reliable predictor of draft status?”, *Journal of Strength and Conditioning Research*, Vol. 17 No. 1, pp. 6–11.

Meir, R., Newton, R., Curtis, E., Fardell, M. and Butler, B. (2001), “Physical Fitness Qualities of Professional Rugby League Football Players”, *Journal of Strength and Conditioning Research*, Vol. 15 No. 4, pp. 450–458.

Microchip Technology. (2003), “Brushless DC (BLDC) Motor Fundamentals AN885”, available at: http://ww1.microchip.com/downloads/en/appnotes/00885a.pdf (accessed 4 August 2018).

Microgate. (2015), “Wireless Training Timer User Manual V1.3.1”, Bolzano, available at: https://fccid.io/2ADEOWIT005/User-Manual/User-Manual-2912601.pdf (accessed 26 July 2018).

Mooney, M.G., Hunter, J.R., O’Brien, B.J., Berry, J.T. and Young, W.B. (2011), “Reliability and validity of a novel intermittent peak running speed test for australian football”, *Journal of Strength and Conditioning Research*, Vol. 25 No. 4, pp. 973–979.

Morland, B., Bottoms, L., Sinclair, J. and Bourne, N. (2013), “Can change of direction speed and reactive agility differentiate female hockey players?”, *International Journal of Performance Analysis in Sport*, Vol. 13 No. 2, pp. 510–521.

Muñiz, C., Levi, R., Benkrid, M., Rodríguez, F.B. and Varona, P. (2008), “Real-time control of stepper motors for mechano-sensory stimulation”, *Journal of Neuroscience Methods*, Vol. 172 No. 1, pp. 105–111.

Murphy, E. and King, E.A. (2016), “Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise", *Applied Acoustics*, Vol. 106, pp. 16–22.

Nauright, J. (2018), "Australian Rules Football", *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., available at: https://www.britannica.com/sports/Australian-rules-football.

Nawi, M. and Homoud, A. (2015), "Relationships between illinois agility test and reaction time in male athletes", *The Swedish Journal of Scientific Research*, Vol. 2 No. 3, pp. 28–33.

NCAA. (2018a), "2018-19 NCAA Men's Basketball Rules", National Collegiate Athletic Association, Indianapolis, available at: http://www.ncaapublications.com/productdownloads/BR19.pdf (accessed 2 August 2018).

NCAA. (2018b), "2017-18 NCAA Women's Basketball Rules", National Collegiate Athletic Association, Indianapolis, available at: http://www.ncaapublications.com/productdownloads/WBR19.pdf (accessed 2 August 2018).

Netball Australia. (2018), "Netball", available at: https://netball.com.au/ (accessed 26 July 2018).

Neuroscience and Robotics Laboratory. (2006), "Switch Debouncing", Northwestern University. McCormick School of Engineering., Evanston, available at: http://hades.mech.northwestern.edu/index.php/Switch_Debouncing (accessed 15 October 2018).

Nimphius, S. (2014), "Increasing Agility", in Joyce, D. and Lewindon, D. (Eds.), *High-Performance Training for Sports*, Human Kinetics, Champaign, IL, pp. 185–198.

Nimphius, S., McGulgan, M.R. and Newton, R.U. (2012), "Changes in muscle architecture and performance during a competitive season in female soccer players", *Journal of Strength & Conditioning Research*, Vol. 26 No. 10, pp. 2655–2666.

Nordic Semiconductor Inc. (2008), "nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0", Trondheim, Norway, available at: https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P (accessed 11

October 2018).

NSW Netball Association. (2018), “Netball FAQ”, New South Wales, available at: https://nsw.netball.com.au (accessed 26 July 2018).

Osborn, A.F. (2013), *Applied Imagination: Principles And Procedures Of Creative Thinking*, *Oxford*, available at:https://doi.org/citeulike-article-id:975273.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Engineering Design: A Systematic Approach*, edited by Wallace, K. and Blessing, L., 3rd ed., Springer, London.

Parsons, L.S. and Jones, M.T. (1998), “Development of speed, agility, and quickness for tennis athletes”, *Strength and Conditioning Journal*.

Parts Not Included. (2017), “Segmented LED Display ASCII Library [Image]”, available at: https://www.partsnotincluded.com/electronics/segmented-led-display-ascii-library/ (accessed 29 May 2018).

Paul, D.J., Gabbett, T.J. and Nassis, G.P. (2016), “Agility in Team Sports: Testing, Training and Factors Affecting Performance”, *Sports Medicine*, Springer International Publishing, Vol. 46 No. 3, pp. 421–442.

Pauole, K., Madole, K., Garhammer, J., Lacourse, M. and Rozenek, R. (2000), “Reliability and Validity of the T-Test as a Measure of Agility, Leg Power, and Leg Speed in College”, *Journal of Strength and Conditioning Research*, Vol. 14 No. 4, pp. 443–450.

Pitney, W.A. and Parker, J. (2001), “Qualitative Inquiry in Athletic Training: Principles, Possibilities, and Promises”, *Journal of Athletic Training*, Vol. 36 No. 2, pp. 185–189.

PJRC. (2018), “Teensy USB Development Board”, available at: https://www.pjrc.com/teensy (accessed 24 May 2018).

Pololu Corporation. (2018a), “DRV8825 Stepper Motor Driver Carrier, High Current”, available at: https://www.pololu.com/product/2133 (accessed 31 August 2018).

Pololu Corporation. (2018b), “Stepper Motor: Bipolar, 200 Steps/Rev, 42×38mm, 2.8V, 1.7 A/Phase”, available at: https://www.pololu.com/product/2267 (accessed 31 August 2018).

Pololu Corporation. (2018c), “DRV8834 Low-Voltage Stepper Motor Driver Carrier”, available at: https://www.pololu.com/product/2134 (accessed 31 August 2018).

Pololu Corporation. (2018d), “Pololu Products”, available at: https://www.pololu.com/ (accessed 24 May 2018).

Pyne, D.B., Gardner, A.S., Sheehan, K. and Hopkins, W.G. (2005), “Fitness testing and career progression in AFL football”, *Journal of Science and Medicine in Sport*, Vol. 8 No. 3, pp. 321–332.

Raspberry Pi Foundation. (2018), “Raspberry Pi 3 Model B+ Product Brief”, Cambridge, available at: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf (accessed 7 June 2018).

Raven, P.B., Gettman, L.R., Pollock, M.L. and Cooper, K.H. (1976), “A physiological evaluation of professional soccer players.”, *British Journal of Sports Medicine*, Vol. 10 No. 4, pp. 209–217.

Reilly, T., Williams, A.M., Nevill, A. and Franks, A. (2000), “A multidisciplinary approach to talent identication in soccer”, *Journal of Sports Sciences*, Vol. 18 No. 9, pp. 695–702.

Reiman, M.P. and Manske, R.C. (2009), *Functional Testing in Human Performance*, Human Kinetics, Champaign, IL.

Robertson, S., Woods, C. and Gastin, P. (2015), “Predicting higher selection in elite junior Australian Rules football: The influence of physical performance and anthropometric attributes”, *Journal of Science and Medicine in Sport*, Sports Medicine Australia, Vol. 18 No. 5, pp. 601–606.

Roozen, M. (2004), “Action-reaction: Illinois Agility Test”, *NSCA’s Performance Training Journal*, Vol. 3 No. 5, pp. 5–6.

Rouissi, M., Chtara, M., Berriri, A., Owen, A. and Chamari, K. (2016), “Asymmetry of the modified illinois change of direction test impacts young elite soccer players’ performance”, *Asian Journal of Sports Medicine*, Vol. 7 No. 2, available at:https://doi.org/10.5812/asjsm.33598.

Russell Corporation. (2018), “Sherrin Kangaroo Brand Football”, Scoresby, available at:

https://www.sherrin.com.au (accessed 6 October 2018).

Safe Work Australia. (2005), "National Code of Practice for Manual Handling [NOHSC:2005(1990)]", Canberra.

Safe Work Australia. (2015), "Managing Noise and Preventing Hearing Loss at Work Code of Practice", Canberra, available at: https://www.safeworkaustralia.gov.au/system/files/documents/1702/managing_noise_preventing_hearing_loss_work.pdf (accessed 5 August 2018).

Scanlan, A., Humphries, B., Tucker, P.S. and Dalbo, V. (2013), "The influence of physical and cognitive factirs one reactive agility performance in men basketball players", *Journal of Sports Sciences*, Vol. 32 No. 4, pp. 367–374.

Scanse LLC. (2017), "Scanse Sweep User's Manual and Technical Specifications", San Leandro, available at: https://s3.amazonaws.com/scanse/Sweep_user_manual.pdf (accessed 7 June 2018).

Schroeder, D. and Kleiss, D. (2018), "Flinders University Flinduino", available at: https://wiki.flinders.edu.au/display/FLINDUINO/Flinders+University+-+Flinduino (accessed 24 May 2018).

Sekulic, D., Krolo, A., Spasic, M., Uljevic, O. and Peric, M. (2014), "The development of a new stop'n'go reactive-agility test", *Journal of Strength and Conditioning Research*, Vol. 28 No. 11, pp. 3306–3312.

Sekulic, D., Pehar, M., Krolo, A., Spasic, M., Uljevic, O., Calleja-Gonzalez, J. and Sattler, T. (2017), "Evaluation of Basketball-Specific Agility: Applicability of Preplanned and Nonplanned Agility Performances for Differentiating Playing Positions and Playing Levels.", *Journal of Strength & Conditioning Research (Lippincott Williams & Wilkins)*, Vol. 31 No. 8, pp. 2278–2288.

Semenick, D. (1990), "The T-test", *Strength & Conditioning Journal*, Vol. 12 No. 1, pp. 36–37.

Serpell, B.G., Ford, M. and Young, W.B. (2010), "The Development of a New Test of Agility for Rugby League", *Journal of Strength & Conditioning Research*, Vol. 24 No. 12, pp. 3270–3277.

Serpell, B.G., Young, W.B. and Ford, M. (2011), “Are the perceptual and decision making components of agility trainable? A premlinary investigation”, *Journal of Strength and Conditioning Research*, Vol. 25 No. 5, pp. 1240–1248.

Shanghai Slamtec Co. (2016), “RPLIDAR A1 Introduction and Datasheet Rev1.0”, Shanghai, available at: http://bucket.download.slamtec.com/a8fbefc2a5112988723b0e15e8a3e17c3adeeb3c/rplidar_devkit_manual_en.pdf (accessed 7 June 2018).

Shanghai Slamtec Co. (2018), “RPLIDAR A2 Introduction and Datasheet Rev2.1”, Shanghai, available at: http://bucket.download.slamtec.com/f751e7bd56ca0a85eb9ec9c207f4836dc0ed2acc/LD208_SLAMTEC_rplidar_datasheet_A2M8_v2.1_en.pdf (accessed 7 June 2018).

SHAPE Research Centre. (2018), “SHAPE Research Centre”, Flinders University, Adelaide, available at: http://www.flinders.edu.au/ehl/shape/ (accessed 11 October 2018).

Sheppard, J. and Young, W. (2006), “Agility literature review: Classifications, training and testing”, *Journal of Sports Sciences*, Vol. 24 No. 9, pp. 919–932.

Sheppard, J.M., Young, W.B., Doyle, T.L.A., Sheppard, T.A. and Newton, R.U. (2006), “An evaluation of a new test of reactive agility and its relationship to sprint speed and change of direction speed”, *Journal of Science and Medicine in Sport*, Vol. 9 No. 4, pp. 342–349.

Sierer, S.P., Battaglini, C.L., Mihalik, J.P., Shields, E.W. and Tomasini, N.T. (2005), “The natonal footbal league combine: performance difference between deafted and nondreafted players entering the 2004 and 2005 drafts”, *Journal of Stength and Conditioning Research*, Vol. 22 No. 1, pp. 6–12.

Silicon Laboratories. (2013), “Si1145/46/47 Proximity/UV/Ambient Light Sensor IC with I2C Interface Rev 1.1”, Austin, available at: https://www.allaboutcircuits.com/uploads/articles/si1145-46-47.pdf (accessed 18 July 2018).

Šimonek, J., Horička, P. and Hianik, J. (2016), “Differences in pre-planned agility and reactive agility performance in sport games”, *Acta Gymnica*, Acta Gymnica, Vol. 46 No. 2, pp. 68–73.

Smartfit Inc. (2018), “SMARTfit Strike Pods”, Camarillo, available at: https://www.smartfit.rocks/ (accessed 26 July 2018).

Smith, W.J. (2008), *Optical Engineering: The Design of Optical Systems*, *Modern Optical Engineering*, 4th ed., McGraw Hill Companies, New York, available at:https://doi.org/10.1036/0071476873.

Spalding. (2018), “NBA Official Game Ball”, Scoresby, available at: https://www.spalding.com.au/ (accessed 2 August 2018).

SparkFun Electronics. (2018), “Wireless Buying Guide”, Niwot, available at: https://www.sparkfun.com/pages/wireless_guide (accessed 21 July 2018).

Spasic, M., Krolo, A., Zenic, N., Delextrat, A. and Sekulic, D. (2015), “Reactive agility performance in handball; development and evaluation of a sport-specific measurement protocol”, *Journal of Sports Science and Medicine*, Vol. 14 No. 3, pp. 501–506.

Spiteri, T., Nimphius, S., Hart, N.H., Specos, C., Sheppard, J.M. and Newton, R.U. (2014), “The contribution of strength characteristic to change of direction and agility performance in female basketball athletes”, *The Journal of Strength & Conditioning Research*, Vol. 28 No. 9, pp. 2415–2423.

Spiteri, T., Nimphius, S. and Wilkie, J. (2012), “Comparison of running times during reactive offensive and defensive agility protocols”, *Journal of Australian Strength & Conditioning*, Vol. 20 No. 1, pp. 73–78.

Sporis, G., Jukic, I., Milanovic, L. and Vucetic, V. (2010), “Reliability and Factorial Validity of Agility Tests for Soccer Players”, *Journal of Strength and Conditioning Research*, Vol. 24 No. 3, pp. 679–686.

Stanton, R., Hayman, M., Humphris, N., Borgelt, H., Fox, J., Del Vecchio, L. and Humphries, B. (2016), “Validity of a Smartphone-Based Application for Determining Sprinting Performance”, *Journal of Sports Medicine*, Hindawi Publishing Corporation, Vol. 2016, pp. 1–5.

Stewart, P.F., Turner, A.N. and Miller, S.C. (2014), “Reliability, factorial validity, and interrelationships of five commonly used change of direction speed tests”, *Scandinavian*

Journal of Medicine and Science in Sports, Vol. 24 No. 3, pp. 500–506.

Swift Performance. (2018), “SpeedLight”, Brisbane, available at: https://swiftperformance.com/ (accessed 26 July 2018).

TAG Heuer. (2018), “Photocell Solutions: HL2-31”, La-Chaux-de-Fonds, available at: http://www.tagheuer-timing.com/en/photocell-hl2-31-20m-photocell (accessed 26 July 2018).

Tangient LLC. (2018), “RF24L01 2.4GHz Radio/Wireless Transceivers How-To”, available at: https://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo (accessed 15 August 2018).

Tanner, R.K., Gore, C.J. and Australian Institute of Sport. (2013), *Physiological Tests for Elite Athletes*, edited by Tanner, R.K. and Gore, C.J., 2nd ed., Human Kinetics, Champaign, IL, available at:https://doi.org/10.1590/S1516-18462008000300012.

Technology, M. (2016), “PIC32MX5XX/6XX/7XX Datasheet”, available at: https://au.mouser.com/ (accessed 24 May 2018).

Tenenbaum, G., Levy-Kolker, N., Sade, S., Liebermann, D.G. and Lidor, R. (1996), “Anticipation and confidence of decisions related to skilled performance”, *International Journal of Sport Psychology*, Vol. 27 No. 3, pp. 293–307.

The Eastman Kodak Company. (2004), *Kodak’s Ergonomic Design for People at Work*, 2nd ed., Wiley, Hoboken, available at:https://doi.org/10.1002/9780470172469.

The Gale Group. (1979), “Glass”, *The Great Soviet Encyclopedia*, 3rd ed., Farlex, Inc, available at: https://encyclopedia2.thefreedictionary.com/glass.

Thompson, G. and Lordan, M. (1999), “A review of creativity principles applied to engineering design”, *Proc. Instn Mech. Engrs, Part E, J. Process Mechanical Engineering*, Vol. 213 No. E1, pp. 17–31.

Till, K., Cobley, S., O’Hara, J., Brightmore, A., Cooke, C. and Chapman, C. (2011), “Using anthropometric and performance characteristics to predict selection in junior UK Rugby League players”, *Journal of Science and Medicine in Sport*, Sports Medicine Australia, Vol. 14 No. 3, pp. 264–269.

Till, K., Cobley, S., O'Hara, J., Chapman, C. and Cooke, C. (2013), "A longitudinal evaluation of anthropometric and fitness characteristics in junior rugby league players considering playing position and selection level", *Journal of Science and Medicine in Sport*, Sports Medicine Australia, Vol. 16 No. 5, pp. 438–443.

Till, K., Cobley, S., O'Hara, J., Morley, D., Chapman, C. and Cooke, C. (2015), "Retrospective analysis of anthropometric and fitness characteristics associated with long-term career progression in Rugby League", *Journal of Science and Medicine in Sport*, Sports Medicine Australia, Vol. 18 No. 3, pp. 310–314.

Total Tools. (2018), "Laser Distance Measurers", available at: https://www.totaltools.com.au/ (accessed 7 June 2018).

Trecroci, A., Milanovi??, Z., Rossi, A., Broggi, M., Formenti, D. and Alberti, G. (2016), "Agility profile in sub-elite under-11 soccer players: is SAQ training adequate to improve sprint, change of direction speed and reactive agility performance?", *Research in Sports Medicine*, Vol. 24 No. 4, pp. 331–340.

USAFL. (2018), "Introduction to Australian Football", United States Australian Football League, Inc, available at: https://usafl.com/intro (accessed 4 August 2018).

Veale, J.P., Pearce, A.J. and Carlson, J.S. (2010), "Reliability and validity of a reactive agility test for Australian football.", *International Journal of …*, Vol. 5 No. 2, pp. 239–248.

Verein Deutscher Ingenieure. (1993), "VDI 2221 - Systematic Approach to the Design of Technical Systems and Products", VDI-Standard Handbook 2221, Düsseldorf, Germany.

Verstegen, M., Marcello, B., Verstegan, M. and Marcello, B. (2001), "Agility and Coordination", in Foran, B. (Ed.), *High-Performance Sports Conditioning*, Human Kinetics, Champaign, IL, pp. 139–165.

Vincent, W.J. and Weir, J.P. (2012), *Statistics in Kinesiology*, 4th ed., Human Kinetics, Champaign, IL.

Vishay Intertechnology. (2015), "VCNL4040 Fully Integrated Proximity and Ambient Light Sensor with Infrared Emitter, I2C Interface, and Interrupt Function Rev 1.4", Malvern, available at: https://www.vishay.com/docs/84274/vcnl4040.pdf (accessed 17 July

2018).

Wilson Sporting Goods. (2018), “NCAA Official Game Basketball”, Chicago, available at: https://www.wilson.com/ (accessed 2 August 2018).

Woods, C.T., Raynor, A.J., Bruce, L., McDonald, Z. and Collier, N. (2015), “Predicting playing status in junior Australian Football using physical and anthropometric parameters”, *Journal of Science and Medicine in Sport*, Sports Medicine Australia, Vol. 18 No. 2, pp. 225–229.

World Rugby. (2018), “Rugby Union: Laws of the Game”, Dublin, available at: https://laws.worldrugby.org/?law=2 (accessed 2 August 2018).

Yeadon, M.R., Kato, T. and Kerw In, D.G. (1999), “Measuring running speed using photocells”, *Journal of Sports Sciences*, Vol. 17 No. 3, pp. 249–257.

Young, W., Farrow, D., Pyne, D.B., McGregor, W. and Handke, T. (2011), “Validity and reliability of agility tests in junion Australian football players”, *Journal of Strength and Conditioning Research*, Vol. 25 No. 12, pp. 3399–3403.

Young, W. and Rogers, N. (2016), “Effects of small-sided game and change-of- direction training on reactive agility and change- of-direction speed”, *Journal of Sports Sciences*, Vol. 32 No. 4, pp. 307–314.

Young, W.B., Dawson, B. and Henry, G.J. (2015), “Agility and Change-of-Direction Speed are Independent Skills: Implications for Training for Agility in Invasion Sports”, *International Journal of Sports Science & Coaching*, Vol. 10 No. 1, pp. 159–169.

Young, W.B., James, R. and Montgomery, I. (2002), “Is muscle power related to running speed with change direction?”, *Journal of Sports Medicine and Physical Fitness*, Vol. 42 No. 3, pp. 282–288.

Young, W.B. and Murray, M.P. (2017), “Reliability of a field test of defending and attacking agility in australian football and relationships to reactive strength”, *Journal of Strength and Conditioning Research*, Vol. 31 No. 2, pp. 509–516.

Zybek Sports. (2018), “Zybek Performance Testing Equipment”, Broomfield, available at: https://www.zybeksports.com/ (accessed 26 July 2018).

8. Appendix

Appendix A

A.1. Proposal Brief

The following brief for the project was supplied by the SHAPE Research Centre:

FLINDERS UNIVERSITY: HIGH PERFORMANCE PROGRAMS (ELITE SPORT) PROJECT PROPOSAL

- **Overview**

The SHAPE Research Centre is a multidisciplinary cluster of academic researchers in Sport, Health, Performance, Exercise and Physical Education. SHAPE is particularly proud of our High Performance Programs (HPPs) which are delivered to a number of organisations including the SANFL Womens Talent Search, the South Adelaide Football Club's Development Academy, the West Adelaide Football Club's Development Academy, the Contax Netball Club and Surf Lifesaving South Australia. To enhance our program, we are seeking the development of a new technology that could be used to test an athlete's repeat agility.

- **The Objective**

To develop a cutting edge, wireless system that randomly signals (audio and visual) where to move on five consecutive occasions, each new signal activating once an athletes 'deactivates' the signal.

- **OUR PROPOSAL**

The image below is what is commonly used in elite sport environments to test an athletes reactions. Lights randomly appear and the athlete must touch the light as quickly as possible to initiate the next light. The technology we are seeking adopts this concept but within a broader testing scope that includes physically accelerating (running) to each light (or signal).

The signals would need to be random so each athlete cannot anticipate patterns.

If this project is of interest to your Honours or RHD students, we would be excited to use a prototype at our next HPP in September 2018.

Figure A- 1: Project brief presented by the SHAPE Research

Appendix B

B.1. House of Quality Correlation Matrix

Table B- 1: House of Quality specification correlation matrix

	Intra-system communication	Physical connections between devices and the user	Set-up time	Steps required to set-up test	Automatic test initiation upon athlete detection	Type of device system can connect to	Number of characters displayable on marker	Number of unique colours displayable on marker	Functions indoors and outdoors	Sensor response time	Adjustable tests difficulty	Number of markers	Number of sporting professions compatible with	Number of unique marker activation sequences	Maximised battery lifetime	Number of programming languages used	When user feedback is received
Intra-system communication	**9**	9	1		9				1			3		9	-		
Physical connections between devices and the user	9	**9**	1			9			1			1			1		
Set-up time	1	1	**9**	9	3		1	1	1	9	3	-					
Steps required to set-up test			9	**9**	9		1	1				-					
Automatic test initiation upon athlete detection	9		3	9	**9**					9							
Type of device system can connect to		9				**9**				9					-	3	1
Number of characters displayable on marker			1	1			**9**	1					1	3	-		
Number of unique colours displayable on marker			1	1			1	**9**						3	-		
Functions indoors and outdoors	1	1	1						**9**	1	3		9		-		

	Number of feedback types	Number of user commands	Response time after user input command	Device boot-up time	Number of previous tests stored at any given time	Device power modes	User can create custom tests	Sport-specific equipment tracking	Area occupied by testing zone	Length x width of each marker	Height of each marker	Fit system inside a carry-case	Weight of each device unit	Marker load capacity	Ingress Protection	Adjustable height	Impact resistance
Intra-system communication					1		3	3									
Physical connections between devices and the user	3		1										1		1		
Set-up time	1			9			1										
Steps required to set-up test							-									-	
Automatic test initiation upon athlete detection							-				3					-	
Type of device system can connect to	3	1															
Number of characters displayable on marker	3						9						-				
Number of unique colours displayable on marker	1						9										
Functions indoors and outdoors							9		1			9			9		

	Number of parts	Internal parts enclosed	Mechanical tools required to change/recharge battery	Safe to use	Extra-low voltage system	Noise produced by system	Noise produced by audio stimulus	Number of loose parts	Number of sensory pathways marker stimulus uses	Number of incorporated agility components	Number of tests before athlete is familiar with test	Accuracy of distance measurement sensor	Accuracy of time measurement	Number of known bugs in program	Variance in test results for a given athlete	Manufacturing cost	System designed and built by due date
Intra-system communication	9									3						-	-
Physical connections between devices and the user	-	1														-	
Set-up time	-															-	
Steps required to set-up test																-	
Automatic test initiation upon athlete detection	-					-						9		-		-	-
Type of device system can connect to	-															-	-
Number of characters displayable on marker	-				-				1	9				-	-	-	-
Number of unique colours displayable on marker									1	9				-		-	-
Functions indoors and outdoors	-	9		-			9			1		-			-	-	-

Sensor response time			9		9	9			1	**9**					-																				-											-		-		-	
Adjustable tests difficulty			3						3		**9**	9	9		-											9																				9	1				-
Number of markers	3	1	-	-							9	**9**	9	9										9		9			-						-			-						9					-	-	-
Number of sporting professions compatible with							1		9		9	9	**9**	1										9		3		9					9		-								9	9							-
Number of unique marker activation sequences	9						3	3				9	1	**9**	-							9		9		1			-																-					-	-
Maximised battery lifetime	-	1				-	-	-	-	-	-			-	**9**		-	-	-	-	-		9		-										-				9		-		-	-		-	-	-		-	-
Number of programming languages used						3										**9**		1						3	-																										
When user feedback is received						1									-		**9**	9		9																												-			
Number of feedback types		3	1			3	3	1							-	1	9	**9**	9	1	-			9	1					-					-															-	-
Number of user commands						1									-			9	**9**	-				9																											-
Response time after user input command		1													-		9	1	-	**9**																												-		-	
Device boot-up time			9												-			-			**9**														-								-					-		-	
Number of previous tests stored at any given time	1													9								**9**		9																								-		-	
Device power modes															9								**9**															9		9										-	
User can create custom tests	3		1	-	-		9	9	9			9	9	9		3		9	9			9		**9**	1					-	9		9	9	-								9					-	-	-	-
Sport-specific equipment tracking	3														-	-		1						1	**9**				-						-													-		-	-
Area occupied by testing zone									1		9	9	3	1												**9**			-						-											-			-	-	
Length x width of each marker																											**9**	3	-		1						1													-	
Height of each marker					3								9														3	**9**	-	-			9																	-	
Fit system inside a carry-case									9			-		-											-	-	-	-	**9**	1			-		-	9															-
Weight of each device unit		1					-											-						-				-	1	**9**	9			9				-												-	
Marker load capacity																								9			1			9	**9**	9	-	9				9												-	-
Ingress Protection		1							9																						9	**9**	-		-	9	-	9		9	-									-	-
Adjustable height				-	-								9											9				9	-		-	-	**9**		-															-	-
Impact resistance																								9						9	9			**9**	-	9		9				-								-	-
Number of parts	9	-	-		-	-	-		-	-		-	-		-			-			-			-	-	-			-			-	-	-	**9**					-								-		-	-
Internal parts enclosed		1							9																				9			9		9		**9**		9		9											-

Mechanical tools required to change/recharge battery																											1					-					**9**	9													-
Safe to use									-			-											9							-	9	9		9		9	9	**9**	9		-	9		-				-		-	-
Extra-low voltage system							-								9																							9	**9**												
Noise produced by system					-																		9									9			-	9				**9**		9								-	
Noise produced by audio stimulus									9						-																	-						-			**9**			9						-	
Number of loose parts																																		-				9		9		**9**									-
Number of sensory pathways marker stimulus uses							1	1					9		-						-			9																			**9**	9				-		-	-
Number of incorporated agility components	3						9	9	1			9	9		-																							-			9		9	**9**	-			-		-	-
Number of tests before athlete is familiar with test														-																														-	**9**						
Accuracy of distance measurement sensor					9				-	-	9				-											-																				**9**		-		-	-
Accuracy of time measurement											1				-																																**9**	-		-	
Number of known bugs in program					-		-	-		-					-		-			-	-	-		-	-										-			-					-	-		-	-	**9**	-		-
Variance in test results for a given athlete							-		-			-												-		-																						-	**9**	-	-
Manufacturing cost	-	-	-	-	-	-	-	-	-	-		-		-	-			-		-	-	-	-	-	-	-	-	-		-	-	-	-	-	-			-		-	-		-	-		-	-		-	**9**	
System designed and built by due date	-				-	-	-	-	-		-	-	-	-	-			-	-					-	-				-		-	-	-	-	-	-	-	-				-	-	-		-		-	-		**9**

B.2. House of Quality Requirement-Specification Relationship Matrix

Table B- 2: House of Quality requirement-specification relationship matrix

	Intra-system communication	Physical connections between devices and the user	Set-up time	Steps required to set-up test	Automatic test initiation upon athlete detection	Type of device system can connect to	Number of characters displayable on marker	Number of unique colours displayable on marker	Functions indoors and outdoors	Sensor response time	Adjustable tests difficulty	Number of markers	Number of sporting professions compatible with	Number of unique marker activation sequences	Maximised battery lifetime	Number of programming languages used	When user feedback is received
Portable		9	1						9			9	1	9			
Lightweight		1	3						1			3	1				
Wireless	9	9	3	1	9	9			3		1	1		1	9		3
Adjustable test difficulty		9	3	9					3		9	3	9		9		
Easy to set-up	9	3	9	9	9	1		3	3	9	3	9			9		
Compatible with Australian rules football									9	1			9	1	3		
Functions indoors and outdoors	9	9	3		9	1			9	9	3		9		9		
Standardisation	9		9	9	9	3	1	1	9	9	9	9	9	9	1	1	9
Integration of sport-specific equipment	3	1	9	9					9	1			9		9		
Marker deactivation and activation	9								9	9		3	1	9	9		
Visual and audio stimulus	3						9	9	9		1		1	1	9		9
Visual stimulus presents random character	3						9	9	9				1	9	9		9

	Number of feedback types	Number of user commands	Response time after user input command	Device boot-up time	Number of previous tests stored at any given time	Device power modes	User can create custom tests	Sport-specific equipment tracking	Area occupied by testing zone	Length x width of each marker	Height of each marker	Fit system inside a carry-case	Weight of each device unit	Marker load capacity	Ingress Protection	Adjustable height	Impact resistance
Portable									9		9	9	9	3	1	3	1
Lightweight								9	9		9	3	9	9	1	1	9
Wireless			1					9	3				1		9		
Adjustable test difficulty							9		9								
Easy to set-up	9						9	9	9		3		9			3	
Compatible with Australian rules football								1	1				1	9			9
Functions indoors and outdoors							9	1					3		9		1
Standardisation							9	9	9							9	
Integration of sport-specific equipment							9	9			1	9	9	9			9
Marker deactivation and activation							9		3							1	9
Visual and audio stimulus	9						9				1		9				
Visual stimulus presents random character	9						9										

	Number of parts	Internal parts enclosed	Mechanical tools required to change/recharge battery	Safe to use	Extra-low voltage system	Noise produced by system	Noise produced by audio stimulus	Number of loose parts	Number of sensory pathways marker stimulus uses	Number of incorporated agility components	Number of tests before athlete is familiar with test	Accuracy of distance measurement sensor	Accuracy of time measurement	Number of known bugs in program	Variance in test results for a given athlete	Manufacturing cost	System designed and built by due date
Portable	9	1						1	1	3						9	9
Lightweight	9		1	9					3	3						9	1
Wireless	9	9											1			3	1
Adjustable test difficulty	9									1	3	9		1	9	1	1
Easy to set-up	9			1							1	9		9		9	9
Compatible with Australian rules football	3									9						1	1
Functions indoors and outdoors	3	9		9						9					1	9	9
Standardisation	9							1				9	9	9	9	9	9
Integration of sport-specific equipment	3			1						9		1				1	9
Marker deactivation and activation	1						9	1	1					9			3
Visual and audio stimulus	9			9			9		9	9				3		9	3
Visual stimulus presents random character									3	9				3		3	1

Incorporates physical factors of agility	9											9	9	3										9		9				9			9	9	9			9						9	3					9	9
Incorporates perceptual and decision-making factors of agility	9						9	9		9		9	9	9	9									9		1				9				9	9			1			9		9	9	9					9	9
Minimum of four (4) markers	9	9									3	9	9	9										9		9		1	9						9			1						9	9					9	9
Measures time taken to complete test	9										9	9			9		9	9				9		9		9									1									9			9	9	9		1
Communication with laptop or smart phone		9	3			9									9		9	9	9	3				1	9										1													9		9	1
Safe to use/minimum risk profile		9			1				3															3						9	9	9	3	9	1	9	9	9	9	9	9	9								3	3
Sufficient area occupied by testing zone		9									9	9	9											9		9									1			1						9		9				9	1
Reliable	9		9	9	9		1	1	9	9	3	9	9	9										9	9	1							9		1									9		9	9	9	9		9
Valid	9						9	9	9	9	1	3	9	9										9	9	9									1								3	9		9	9	9	9		9
Unpredictable	9						9	9				9		9								3		9											9								1	9	3			3		3	9
Can be turned on and off using a button															9						9		9									9			9		1	9	3											3	1
Simple test initiation	3		9	9	9				1						9			3	9																1											9		1		3	3
Low-cost	9	9	9	9	9	3	9	9	9		1	9	9	9	9									9	9	3		3	9	9	9	9	9	9	9	9	1	9		3	9		9	9			1		3	9	3
Assisted test layout set-up	9		9	9	9				9			9	3	1	9			9		1				3		9			3	3		1	9		9											9		9	9	9	9
Intra-system communication	9	9		9	9				3		3	3			9		1	3		3				1	9	9				1					9	3												9		9	1
Maximised battery lifetime	9	1			3		9	9	9						9			3			1		9		9					9					9			9			3							1		9	3
Appropriate programming language	9								9							9		1	9			3																										9			1
Waterproof design		9							9				9											1	1				1	1		9	9	1	1	9	9	9	1		1									9	9
Provide real-time data feedback	3														3		9	9	1	9					9																						3	9			3
Minimal learning curve									1		1	3	9	9										9																			1		9				3		1

Compatible with various sports									9				9		3									9					3	3	9		3	3	1			1						9	1				9	1	9
System feedback clear and concise			9			1									3		1	3	1	9		1			1																										3
Responsive to user inputs	3														9			3	1	9		1																										9		1	9
Easy change/recharge of batteries			9																											1		9		1	3	3	9	9	9												9
Height adjustability			9						1				9											9				9	9	9	9	9	9	9	9	3		1						9		9			9	3	9
Provides feedback to commands	3					1									9		9	9	9	9		1																										9			1
Maximised load capacity													9											9				1		9	9		9	9				9												9	3
Shock absorbent/ resistant									3				9											9	9			1	1	3			9	9	3	9	1	9	3					1						9	9
Data storage	1														9				1	1		9																										9			3
Bug-free	9			9									1	3	9		9	9				9		9	9													9			1					9	9	9	9		9
Consistent	9		9						9		3	1			1		9	3							9																1			1	9	9	9	9	9		9
User interface must be simple			9			1									1			3	9	9		1																										1			3
Maximised service lifetime															3															3	9			9	9		1	9				9								9	3
Can be put in a sleep mode															9								9									9			1																1
Customisable tests	9	1		9				9	9		9	9	9	1				9	9					9		9		1			9		9					3						9	9	1		9	9		9
Sport-specific equipment tracking	9												1		3									9	9					3				9	9			9												9	9

B.3. Calculating House of Quality Specification Absolute Importance

Table B- 3: Set-up time specification weighted against each design requirement

Design Requirement	Importance (1-10)	Correlation (1,3 or 9)
Portable	10	1
Lightweight	10	3
Wireless	10	3
Adjustable test difficulty	10	3
Easy to set-up	10	9
Compatible with Australian rules football	10	
Functions indoors and outdoors	10	3
Standardisation	10	9
Integration of sport-specific equipment	10	9
Marker deactivation and activation	10	
Visual and audio stimulus	10	
Visual stimulus presents random character	10	
Incorporates physical factors of agility	10	
Incorporates perceptual and decision-making factors of agility	10	
Minimum of four (4) markers	10	
Measures time taken to complete test	10	
Communication with laptop or smart phone	10	3
Safe to use/minimum risk profile	10	
Sufficient area occupied by testing zone	10	
Reliable	10	9
Valid	10	
Unpredictable	10	
Can be turned on and off with a button	10	
Simple test initiation	9	9
Low-cost	8	9
Assisted test layout set-up	8	9
Intra-system communication	8	
Maximised battery lifetime	8	
Appropriate programming language	8	
Waterproof design	7	
Provide real-time data feedback	7	
Minimal learning curve	6	
Compatible with various sports	6	
System feedback clear and concise	5	9
Responsive to user inputs	5	
Easy change/recharge of batteries	4	9
Height adjustability	4	9

Provides feedback to commands	3	
Maximised load capacity	3	
Shock absorbent/ resistant	3	
Data storage	3	
Bug-free	3	
Consistent	3	9
User interface must be simple	2	9
Maximised service lifetime	2	
Can be put in a sleep mode with a button	2	
Customisable tests	2	
Sport-specific equipment tracking	1	

$$\text{Set-up Time Absolute importance} = \sum(Correlation\ Rating \times Importance\ Factor)$$

$$= (10 \times 1) + (10 \times 3) + (10 \times 3) + (10 \times 3) + (10 \times 9) + (10 \times 3)$$
$$+ (10 \times 9) + (10 \times 9) + (10 \times 3) + (10 \times 9) + (9 \times 9) + (8 \times 9) + (8 \times 9)$$
$$+ (5 \times 9) + (4 \times 9) + (4 \times 9) + (3 \times 9) + (2 \times 9) = 907$$

B.4. House of Quality Specification Targets, Absolute Importance and Relative Ratings

Table B- 4: House of Quality specification targets, absolute importance and relative importance

Design Specification	Threshold Target	Absolute Importance	Relative Importance (%)
Number of incorporated agility components	10 components	1874	**4.70**
Manufacturing cost	< $1,750	1814	**4.55**
System designed and built by due date	10-Oct-18	1752	**4.40**
Maximised battery lifetime	5 hours	1701	**4.27**
Intra-system communication	Yes	1686	**4.23**
Number of parts	< 100 parts	1636	**4.10**
Number of markers	4 markers	1554	**3.90**
Functions indoors and outdoors	Yes	1532	**3.84**
Safe to use	Medium risk profile	1498	**3.76**
Number of sporting professions compatible with	1 sport	1399	**3.51**
Number of known bugs in program	3 bugs	1199	**3.01**
Area occupied by testing zone	25m^2	1176	**2.95**
Weight of each device unit	1.2 kg	1056	**2.65**
Number of unique marker activation sequences	25 unique sequences	1015	**2.55**
Physical connections between devices and the user	0 physical connections	987	**2.48**
Set-up time	2 minutes	907	**2.28**
Fit system inside a carry-case	No	813	**2.04**
Steps required to set-up test	15 steps	802	**2.01**
Accuracy of distance measurement sensor	< 10 cm	795	**1.99**
Number of feedback types	1 feedback type	785	**1.97**
Automatic test initiation upon athlete detection	No	781	**1.96**
Impact resistance	5 N	763	**1.91**
Variance in test results for a given athlete	< 0.5 seconds	736	**1.85**
Sport-specific equipment tracking	No	729	**1.83**

Design Specification	Threshold Target	Absolute Importance	Relative Importance (%)
Adjustable height	1 setting	713	**1.79**
Number of displayable characters on marker	1 character	662	**1.66**
Number of unique colours displayable on marker	3 colours	662	**1.66**
User can create custom tests	No	651	**1.63**
Sensor response time	0.01 seconds	650	**1.63**
User feedback	Post-test	637	**1.60**
Marker load capacity	5 N	615	**1.54**
Ingress Protection	IP44	613	**1.54**
Adjustable tests difficulty	2 test difficulties	575	**1.44**
Length x width of each marker	200 x 200 mm	528	**1.32**
Internal parts enclosed	0.9	490	**1.23**
Noise produced by audio stimulus	80 – 100 dB	469	**1.18**
Accuracy of time measurement	0.01 seconds	453	**1.14**
Number of tests before athlete is familiar with test	2 tests	385	**0.97**
Number of sensory pathways marker stimulus uses	2 sensory pathways	378	**0.95**
Number of user commands	3 commands	326	**0.82**
Height of each marker	300 mm	308	**0.77**
Response time after user input command	3 seconds	273	**0.68**
Type of device system can connect to	1 type	264	**0.66**
Mechanical tools required to change/recharge battery	1 tool	222	**0.56**
Number of previous tests stored at any given time	250 tests	213	**0.53**
Device power modes	1 mode	180	**0.45**
Extra-low voltage system	< 12 V	172	**0.43**
Number of loose parts	2 parts	138	**0.35**
Noise produced by system	< 60 dB	114	**0.29**
Device boot-up time	10 seconds	98	**0.25**
Number of programming languages used	2 languages	82	**0.21**

Appendix C

C.1. Brainstorming

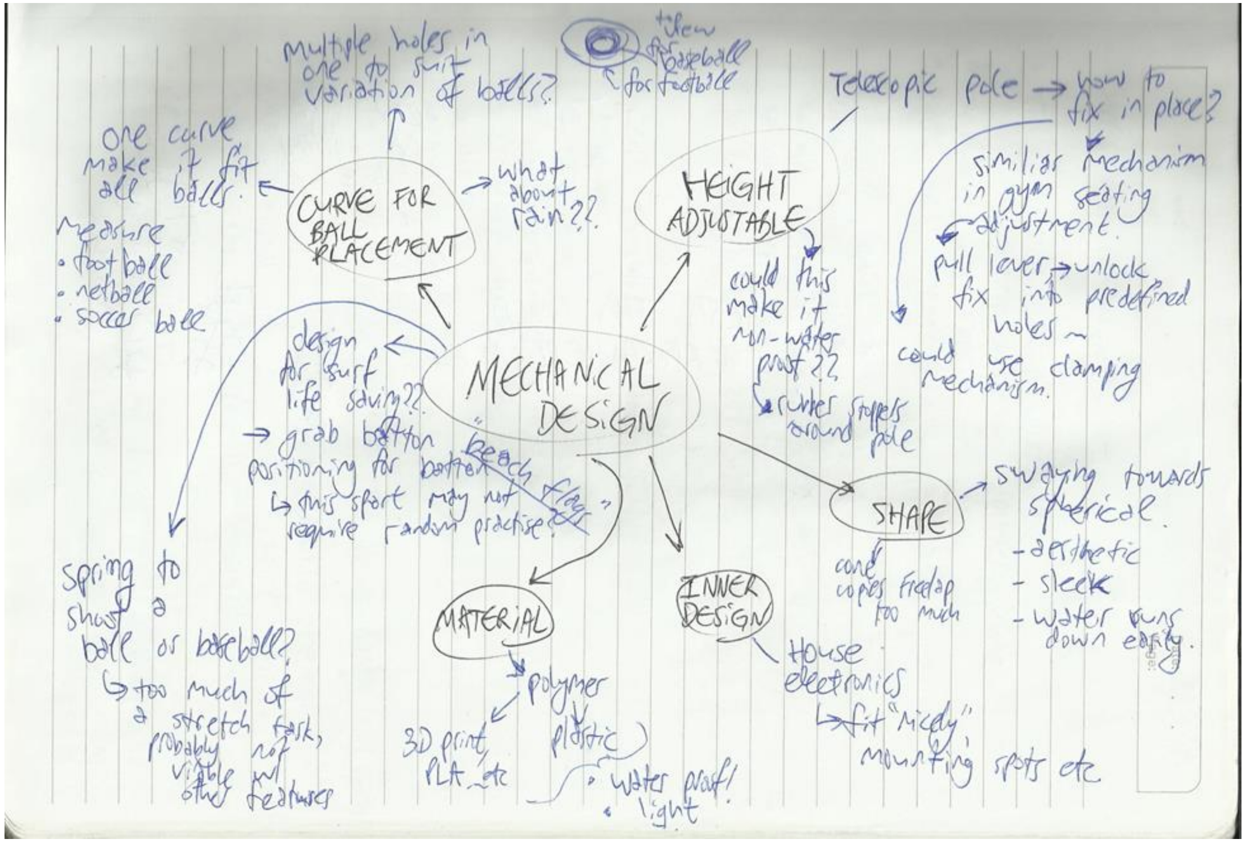


Figure C- 1: Brainstorming the mechanical design

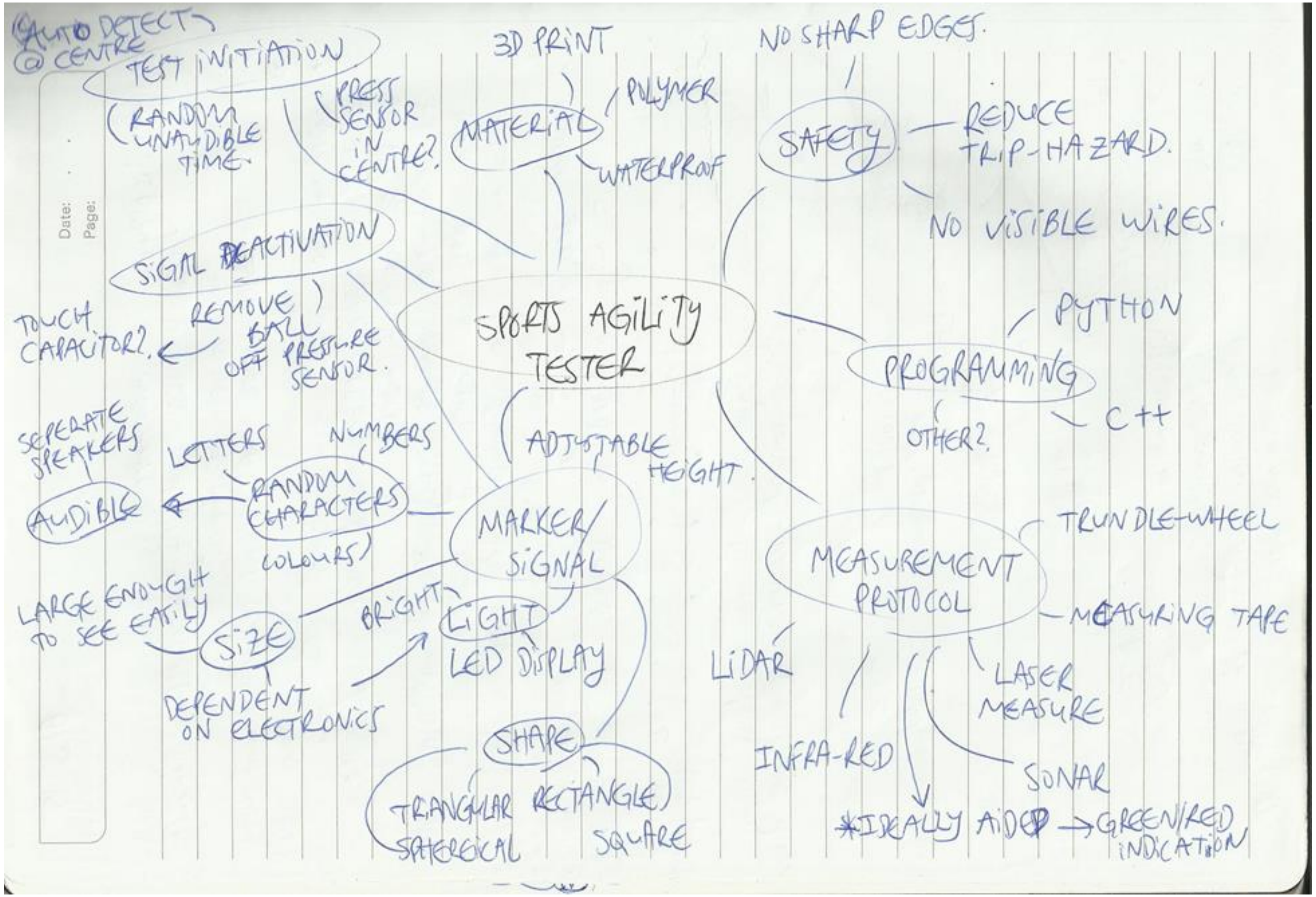


Figure C- 2: Brainstorming all factors of the sports agility tester

Appendix D

D.1. Calculating Design Space

Table D- 1: Determining design space using number of means associated with a design function

Function	Number of Means	Combinations Running Total
Single-board Microcontroller	8	8
Measure marker distance	7	56
Sense Position in Space	3	168
Rotate sensor	4	672
Motor Encoding	2	1344
Resist tangled wires	2	2688
Display visual stimulus	4	10752
Character display	3	32256
Feedback mechanism	6	193536
Sense ball	7	1354752
Auditory stimulus	3	4064256
Power source	8	32514048
User/System interface	6	195084288
Intra-system communication	5	975421440
Initiate test	2	1950842880
Programming Language	6	11705057280
Assist in Detection of Ball	2	23410114560
Enclosure	2	46820229120
Enclosure Material	8	3.74562^{13}
Sensor Rotation	6	2.24737^{13}
Shield Sensor from Environment	5	1.12369^{13}
Anti-slip	3	3.37106^{13}
Hardware Mounting	5	1.68553^{14}
Total Number of Combinations		$\underline{\mathbf{1.69 \times 10^{14}}}$

The total number of combinations for the morphological charts was determined by multiplying each number of means corresponding to each function by one another, as shown in the calculation below:

$$8 \times 7 \times 3 \times 4 \times 2 \times 2 \times 4 \times 3 \times 6 \times 7 \times 3 \times 8 \times 6 \times 5 \times 2 \times 6 \times 2 \times 2 \times 8 \times 6 \times 5 \times 3 \times 5$$
$$= 1.18 \times 10^{14} \text{ combinations}$$

Appendix E

E.1. Microcontroller Alternatives Comparison Chart

Table E- 1: Microcontroller alternatives comparison chart

Microcontroller	Features								
	Cost	I/O	Analog Input	Programming Language	Memory	RAM	CPU Clock Speed	Operating/ Input Voltage	Additional Notes
Arduino[1]	$11.70-$54.54	Up to 54 (15 PWM)	Up to 16	C/C++	Up to 256 KB	Up to 8 KB	Up to 16 MHz	5V / 7 – 12 V	I^2C, SPI, UART, shields available, Community support, Multiple variants available.
Raspberry Pi 3 Model B+[2]	$54.96	40 GPIO (2 PWM)	0	Python, C/C++, Java, Scratch, Ruby	External microSD card	1 GB	1.4 GHz	5V/ 4.75 – 5.25V	Bluetooth, Wi-Fi, Ethernet, audio jack, HDMI, USB ports, Linux OS, Community support
Teensy 3.6[3]	$49.00	42 (+20 SMD) (22 PWM)	25	C/C++	1 MB + microSD card slot	256 KB	180 MHz	3.3V/2.6 – 3.4V	Ethernet, I^2C, RTC
FRDM-K64F Mbed[4]	$49.56	16 (12 PWM)	6	C/C++	Up to 1024 KB + microSD card slot	256 KB	120MHz	5V/ 4.5 – 9V	I^2C, UART, Ethernet. Community support. Online compiler. Other variants available

Microcontroller	Features								
	Cost	I/O	Analog Input	Programming Language	Memory	RAM	CPU Clock Speed	Operating/ Input Voltage	Additional Notes
PIC32[5]	$48.02 - $70.79	Up to 83 (16 PWM)	Up to 16	C	Up to 512 KB	Up to 128 KB	Up to 80 MHz	3.3V / 7 – 15V	I^2C, SPI, UART. Multiple PIC32 variants available.
BeagleBone Black[6]	$62.38	65 (12 PWM)	7	C/C++	4 GB + microSD card slot	512 MB	1 GHz + 2x200 MHz	3.3V / 2.7 – 5.8 V	WiFi, Ethernet, microHDMI, UART, SPI, I^2C. Linux OS. Multiple software compatible. Community support.
Pololu A-Star[7]	$21.11 - $40.88	Up to 26 (7 PWM)	Up to 12	C/C++	32 KB	256 KB	16 MHz	5V / 2.7 – 36 V	Arduino compatible (shields and libraries). Multiple variants available.
Flinduino[8]	Not for Purchase (Prototyping Only)	24 (6 PWM)	13	C/C++	128 KB	32 KB	40 MHz	5V / 5 – 12 V	Flinders University made. Arduino Uno form factor. Arduino/chipKIT library compatible

Sources: [1] (Arduino, 2018); [2](Raspberry Pi Foundation, 2018); [3](PJRC, 2018); [4](Arm Limited, 2018); [5](Digilent Inc, 2018; Technology, 2016); [6](Kridner et al., 2017); [7](Pololu Corporation, 2018d); [8](Schroeder and Kleiss, 2018)

E.2. Distance Measurement Alternatives Comparison Charts

Table E- 2: Distance measurement alternatives comparison chart

Component	Features						
	Total Cost	**Typical Range**	**Usable Outdoors**	**Resolution**	**Field-of-View**	**Arduino Compatible**	**Additional notes**
Ultrasonic Sensor[1]	$60 - $640+ ($5.01 - $53.50 per unit)	Up to 6m	If not windy	0.3cm	Large 11º - 62º	Yes	May need to implement with sensors for position in space
IR Sensor[2]	$200 - $600+ ($16.86 - $48.35+ per unit)	Up to 6m	No	1cm typically	Large 30º – 70º	Yes	May need to implement with sensors for position in space
LiDAR[3]	$229 - $595+	Up to 40m	Yes	1cm typically	Very small	Yes	
Laser Distance Measurer[4]	$32 - $799+	Up to 250m	Yes	1.59 mm typically	Very small	No	Handheld, manual
SF02 Range Finder[5]	$504.51	Up to 50m	Yes	1cm	Very small (0.2º)	Yes	Generally, not Arduino compatible, but this component is

*Note: Components only compared if within budget

Sources: [1](Little Bird Company, 2018b) ;[2] (Benet et al., 2002; Little Bird Company, 2018b); [3](Garmin Corporation, 2016; Scanse LLC, 2017; Shanghai Slamtec Co, 2016, 2018);[4](Total Tools, 2018) [5](Lightware Optoeletronics, 2017)

E.3. System Communication Alternatives Comparison

Table E- 3: System communication alternatives comparison chart

Component	Features							
	Cost	**Typical Range**	**Data Rate**	**Frequency**	**Power Consumption**	**Multiple Connections**	**Mobile Phone or Computer Compatibility**	**Further notes**
Bluetooth	\$19.95- \$27.72+	10-100m	2MB/s	2.4 GHz	30 – 500mA	No	Yes	High power consumption generally. 1:1 connection unless master and multiple slaves with BLE.
Nordic	\$10.63 – \$29.19	10-1km	250KB - 2MB/s	2.4 GHz	13 – 115mA	Yes	No	Low power. Able to sleep.
Wi-Fi	\$9.22 - \$60.72+	Limited to Router Range	Up to 54MB/s	2.4 GHz	75-240mA	Yes	Yes	Requires connection to Wi-Fi router or access point (questionable feasibility on the field)
General RF	\$9.95 - \$64.51	Up to 600m	Up to 300kb/s	315 MHz – 915 MHz	8 – 80mA	Yes	No	Must comply with laws and regulations. Australia can use 915 MHz or 434 MHz.
Cellular	\$57.57 - \$79.92	Depends on network	Depends on network	GSM 850/ 900/ 1800/ 1900 MHz	240-350mA	Yes	Yes	Need SIM card and external antenna.

Sources: (Core Electronics, 2018; Little Bird Company, 2018b; SparkFun Electronics, 2018)

E.4. LiDAR Scanner Alternatives Comparison Charts

Table E- 4: LiDAR alternatives comparison chart

Component	Features								
	Cost	**Distance Range**	**Usable Outdoors**	**Arduino Compatible**	**360° Rotation**	**Scan Rate**	**Sample Rate**	**Distance Resolution Accuracy**	**Current Consumption**
Scanse Sweep[1]	$490	Up to 40m	Yes	Yes	Yes	Up to 1000 Hz	Up to 1075 Hz	±1%	Up to 650mA
LiDAR Lite v3HP[2]	$229	5cm - 40m	Yes	Yes	No	Up to 500 Hz	> 1000 Hz	±1%	Up to 85mA
RPLIDAR A1[3]	$239.25	0.15m – 6m	Yes	Yes	Yes	5.5 Hz	Up to 2000 Hz	±1%	Up to 600mA
RPLIDAR A2[4]	$595.49	0.15m - 18m	Yes	Yes	Yes	10 Hz	Up to 8000 Hz	±1%	Up to 600mA
TeraRanger Evo 60m	$241.63	0.5m – 60m	Yes - but range reduced from direct sunlight	Yes	No	240 Hz	Not specified	±1.5%	Up to 330mA

Datasheet Sources: [1](Scanse LLC, 2017); [2](Garmin Corporation, 2016); [3](Shanghai Slamtec Co, 2016); [4](Shanghai Slamtec Co, 2018)

Appendix F

F.1. Concept Sketching

Test Development

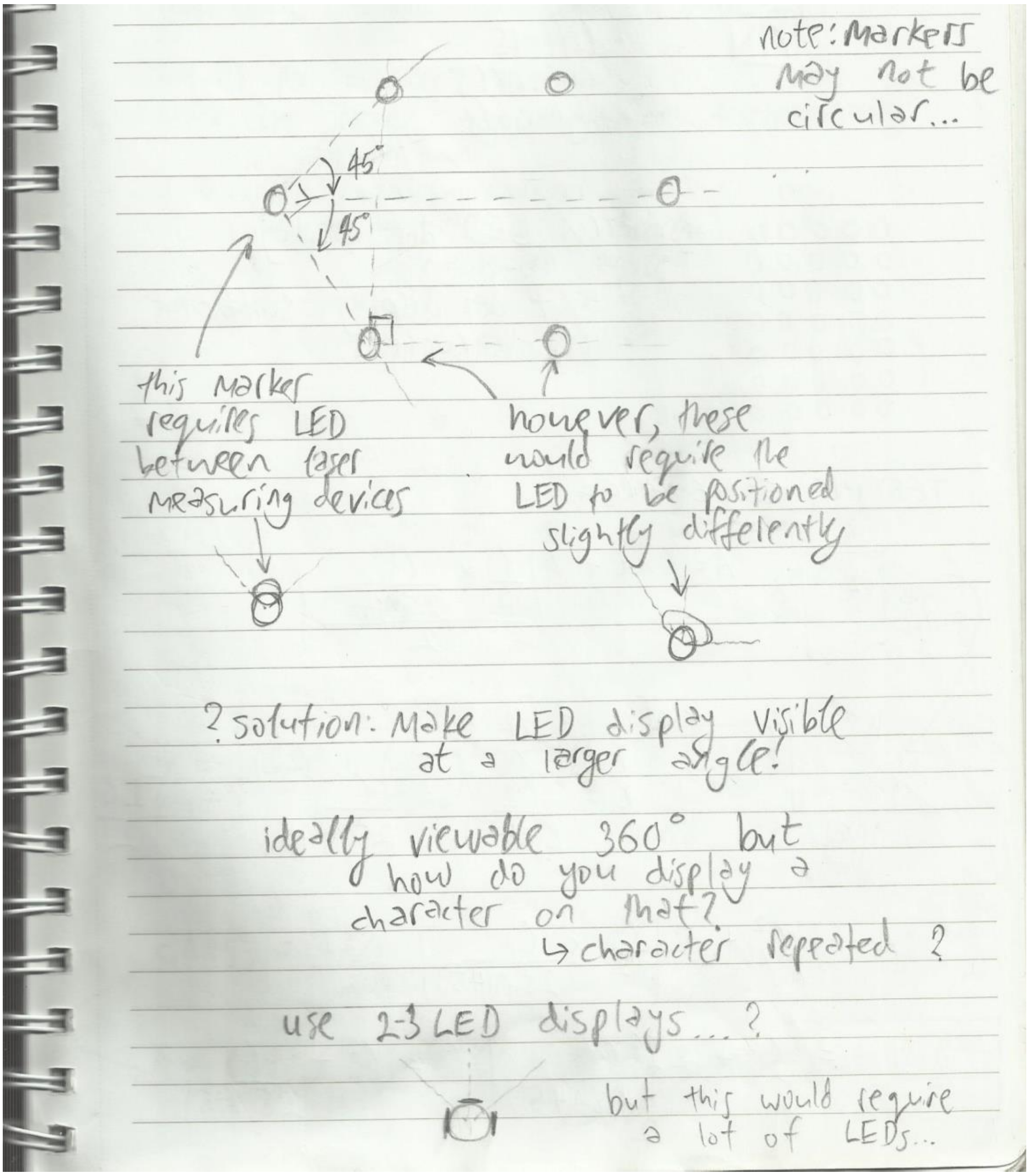


Figure F- 1: Test layout and set-up ideation

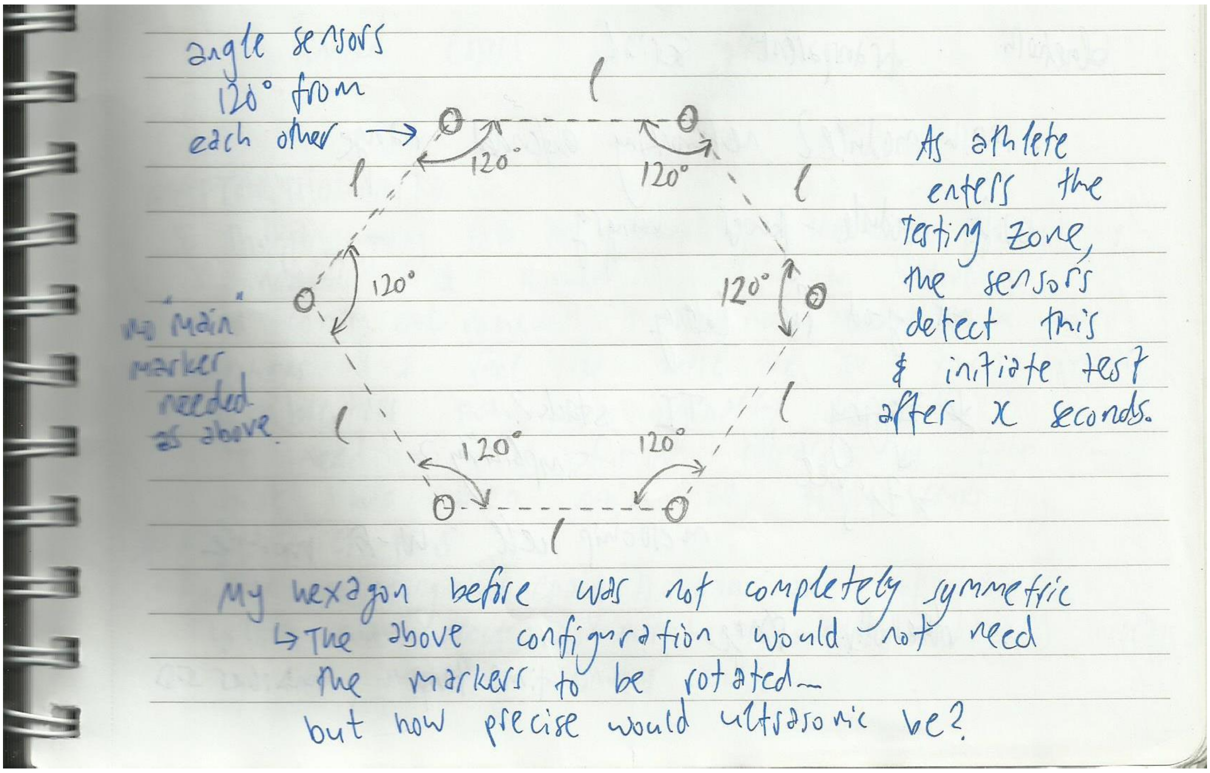


Figure F- 2: Test layout refinement

Slave

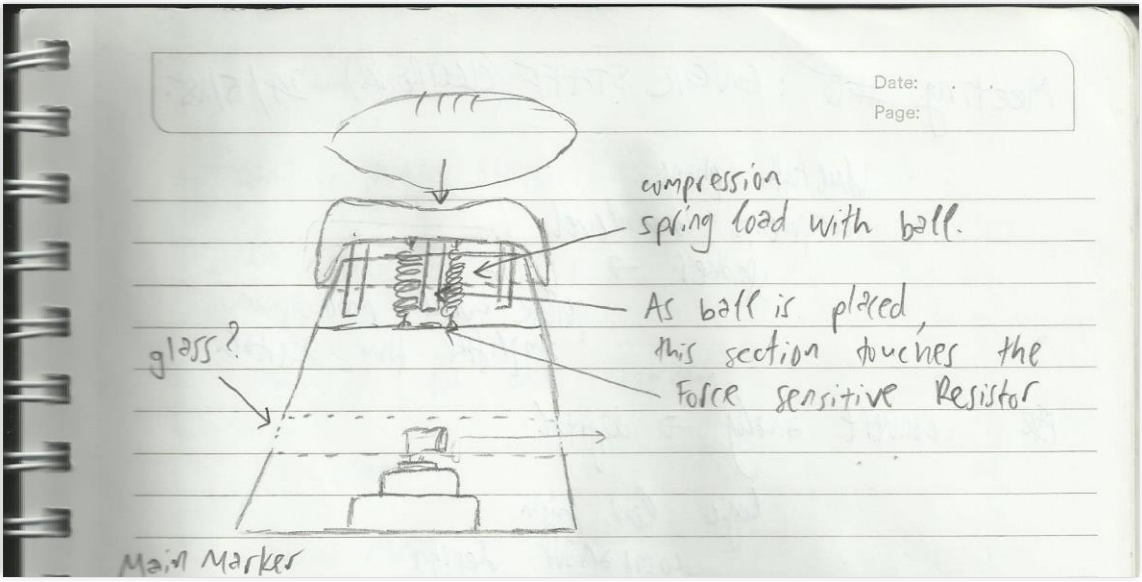


Figure F- 3: Marker ideation and conceptualisation before separation into a master and slave component

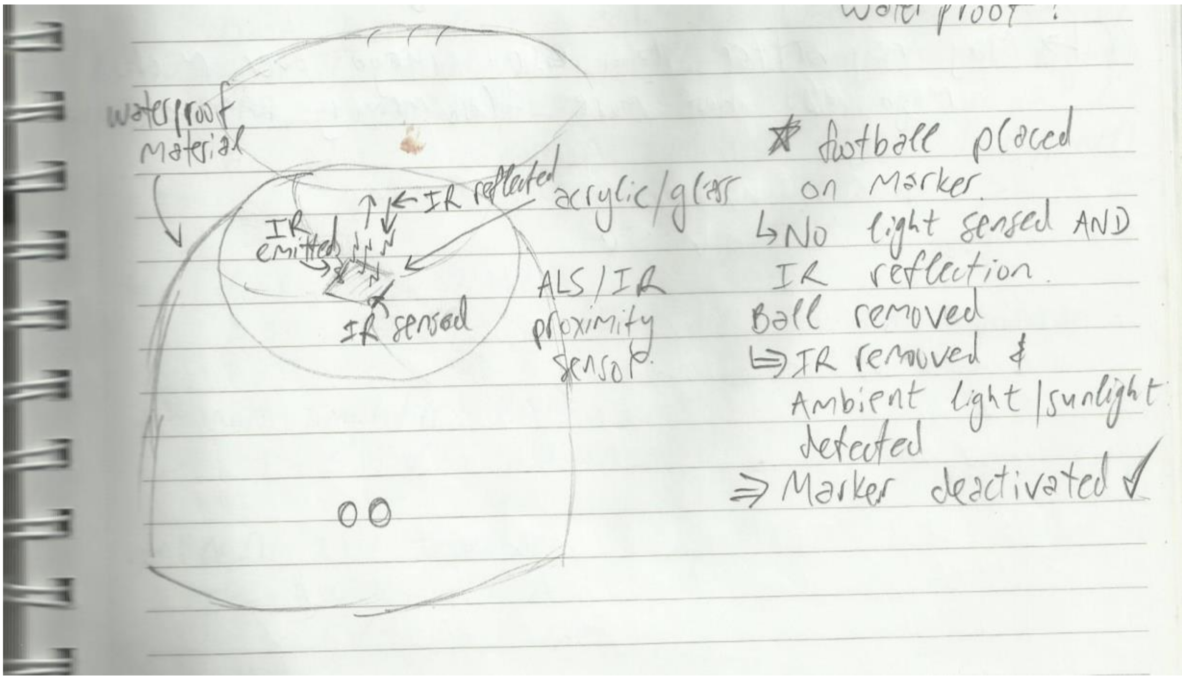


Figure F- 4: Sensor within marker to detect presence of a ball

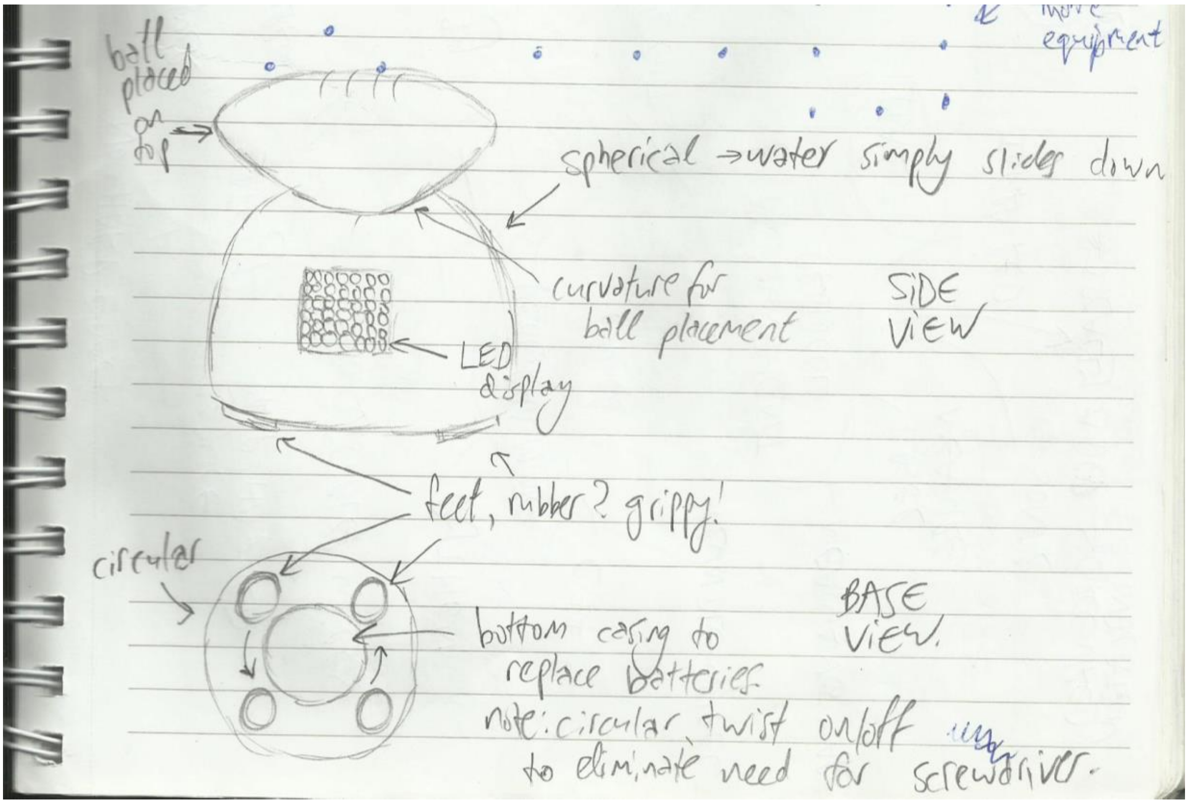


Figure F- 5: Slave component conceptualisation

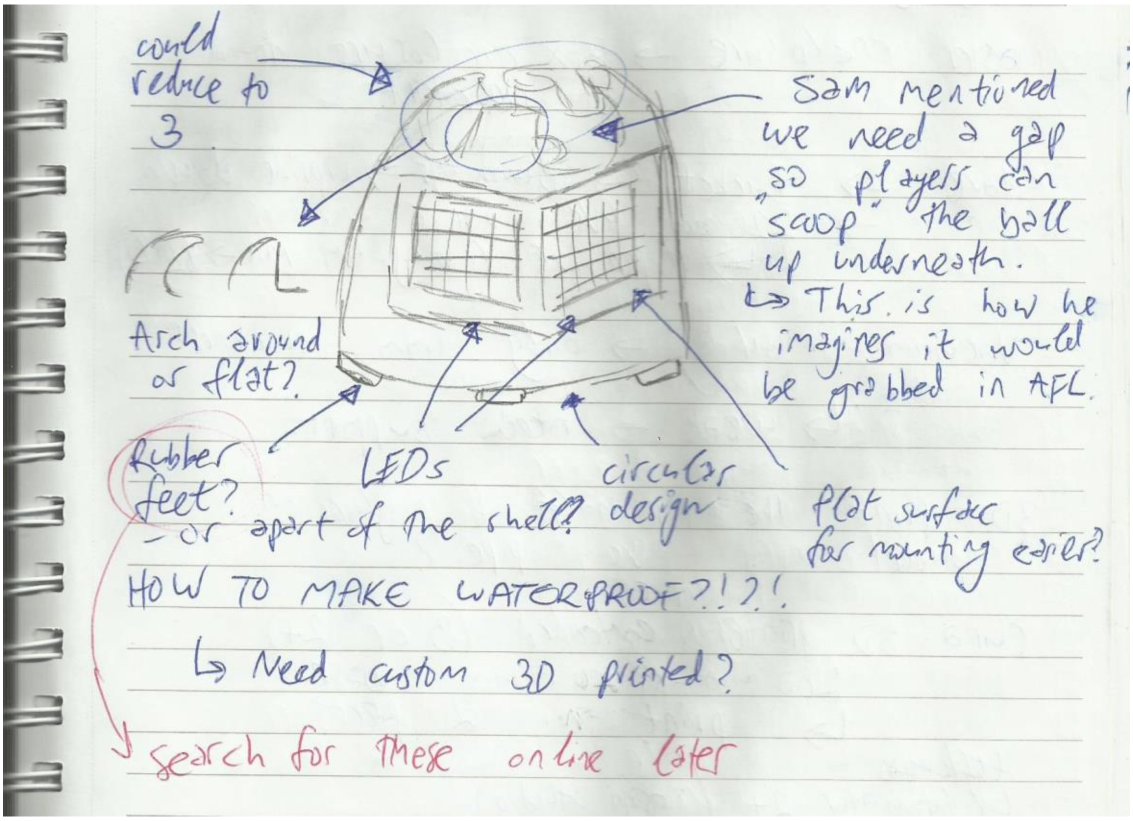


Figure F- 6: Raised ball mounting positions on the marker to allow for a gap to grab underneath the ball

Master

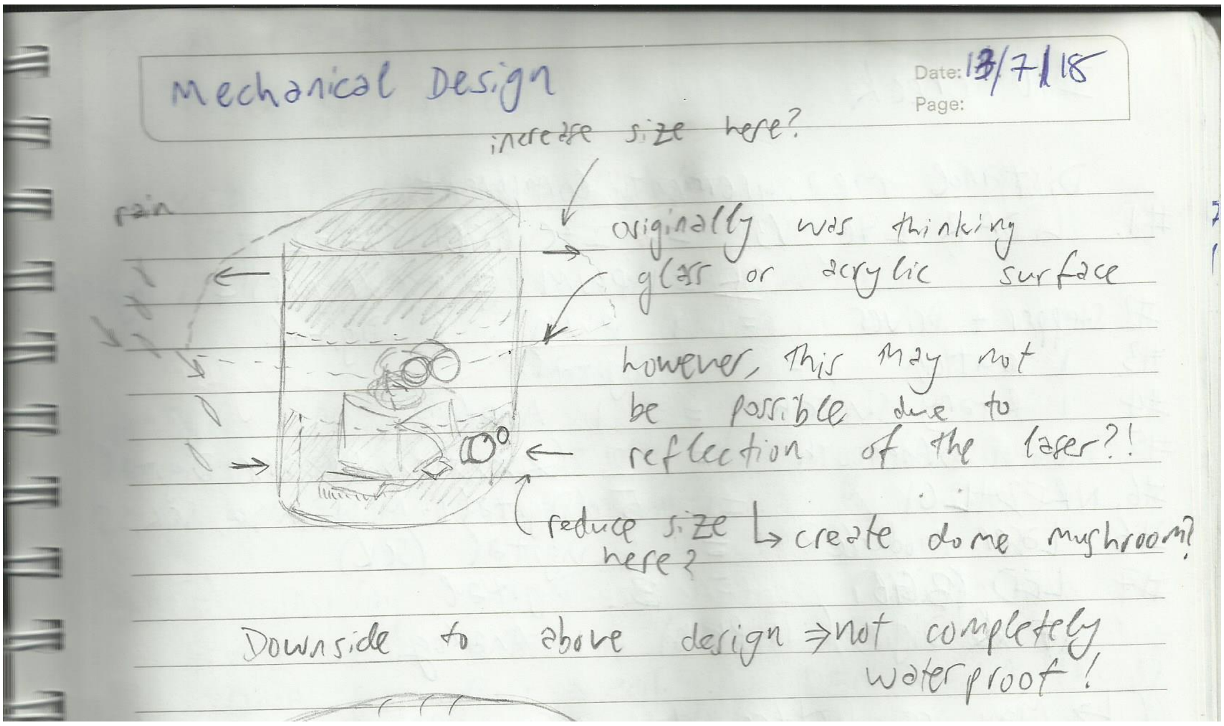


Figure F- 7: Master conceptualisation for waterproofing the design

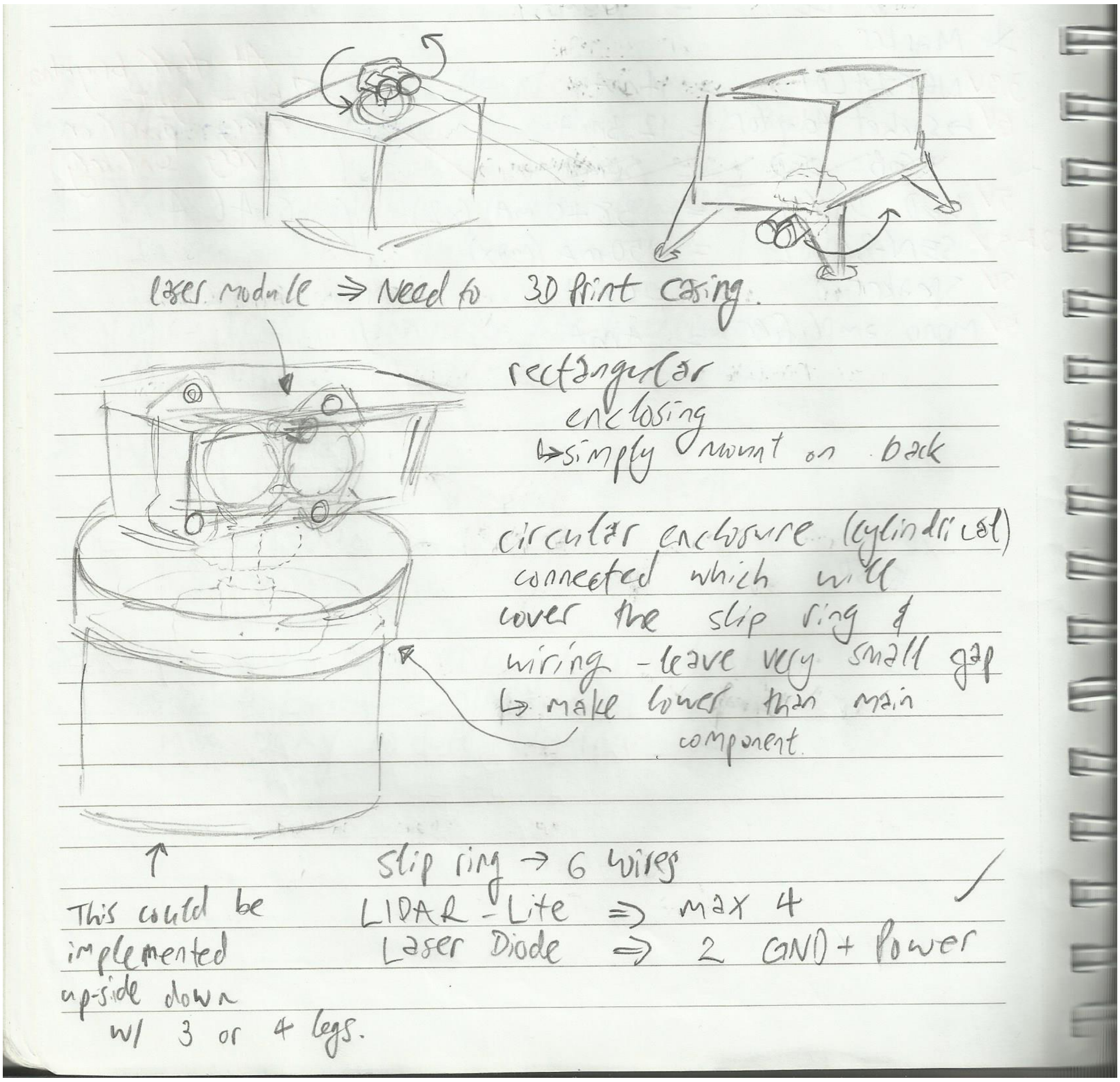
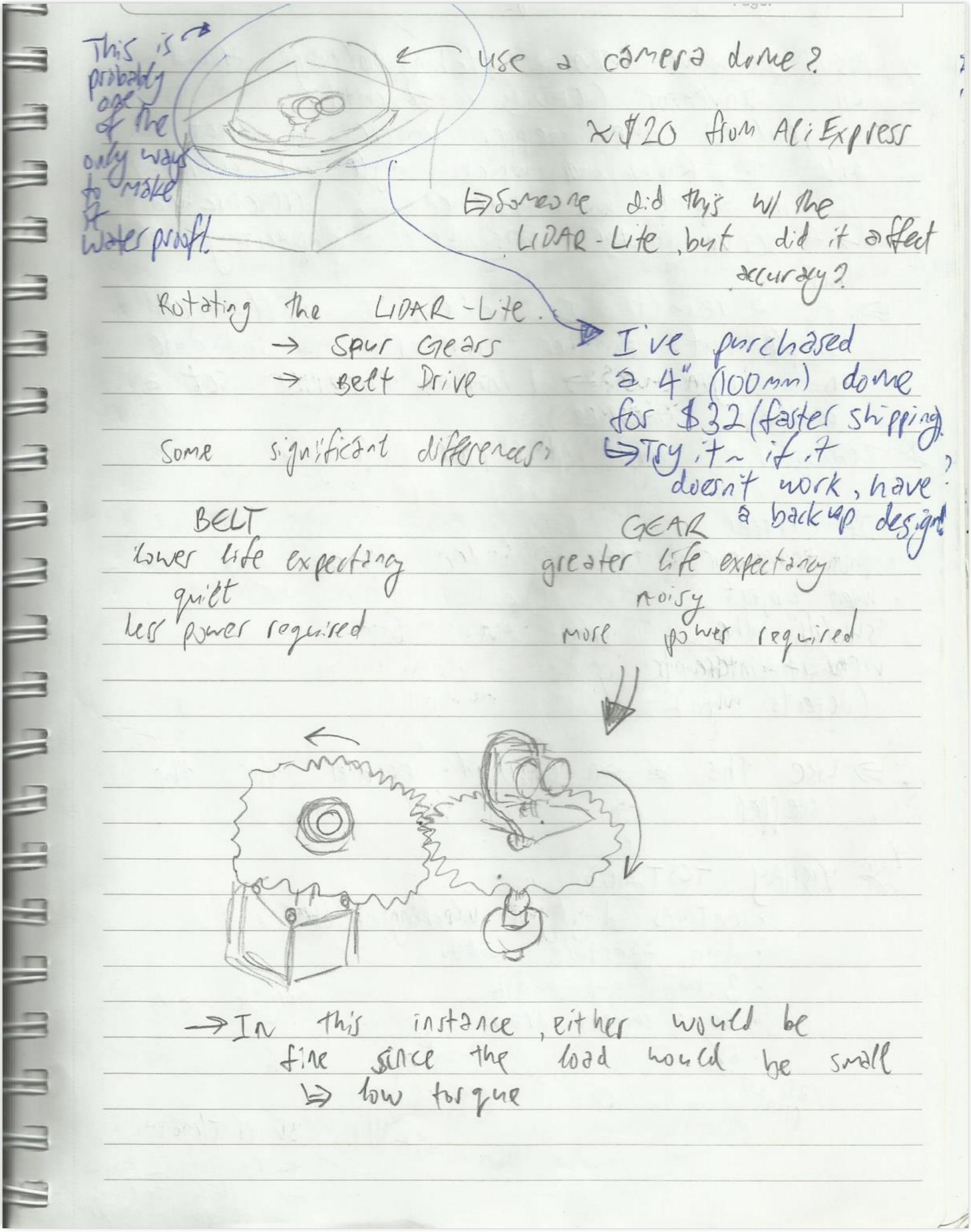


Figure F- 8: Master component enclosure conceptualisation



This is probably one of the only ways to make it waterproof.

← use a camera dome?

≈ $20 from AliExpress

⇒ Someone did this w/ the LiDAR-Lite, but did it affect accuracy?

Rotating the LiDAR-Lite.
→ Spur Gears
→ Belt Drive

⇒ I've purchased a 4" (100mm) dome for $32 (faster shipping)
⇒ Try it ~ if it doesn't work, have a backup design!

Some significant differences:

BELT	GEAR
lower life expectancy	greater life expectancy
quiet	noisy
less power required	more power required

→ In this instance, either would be fine since the load would be small
⇒ low torque

Figure F- 9: Waterproofing the master component and rotation of the LiDAR scanner

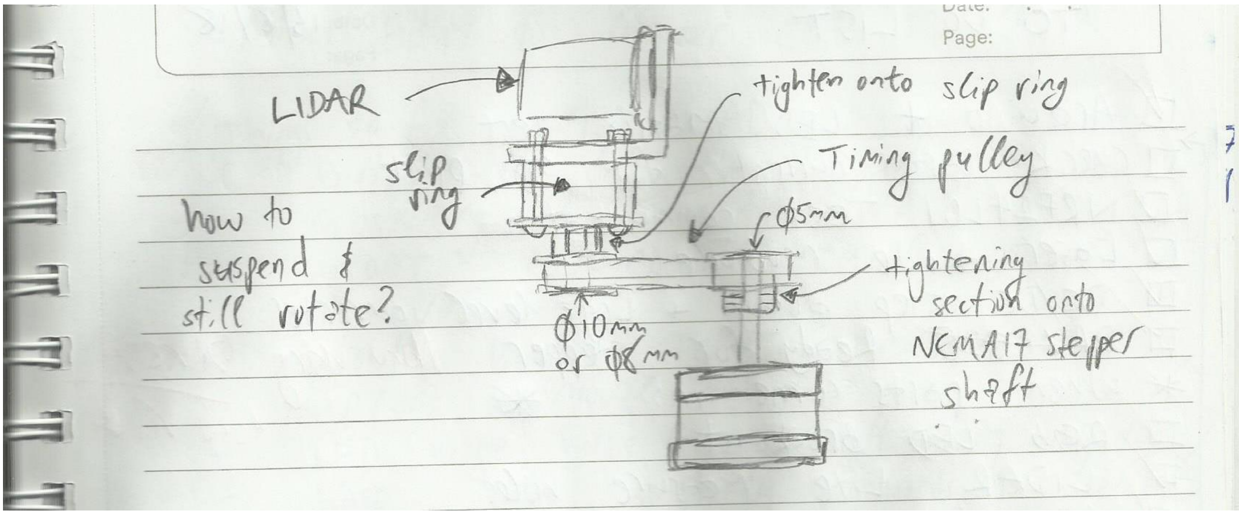


Figure F- 10: Conceptualising the master rotation mechanism

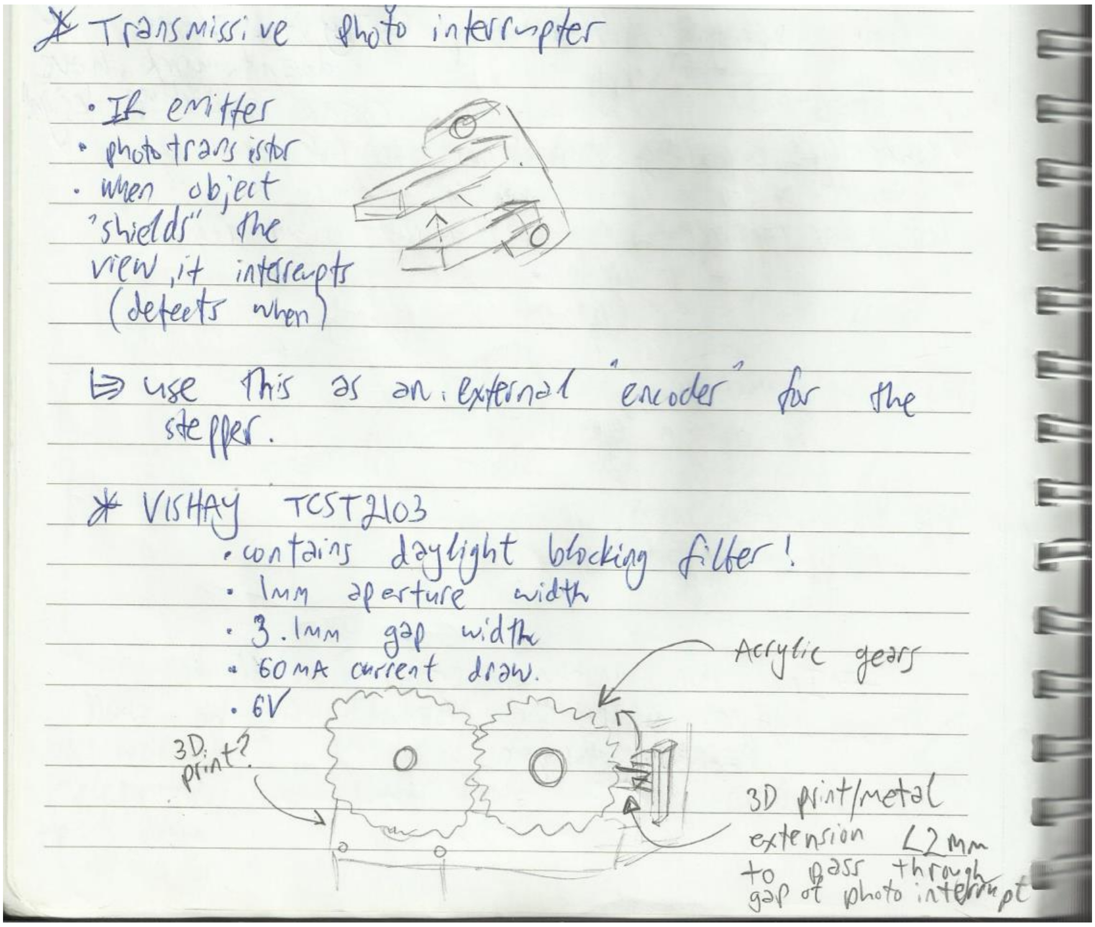


Figure F- 11: Master component rotation mechanism and LiDAR direction encoding

Appendix G

G.1. Engineering Assembly Drawings

All engineering assembly drawings were completed using Autodesk Inventor Professional 2018 (Student Edition)

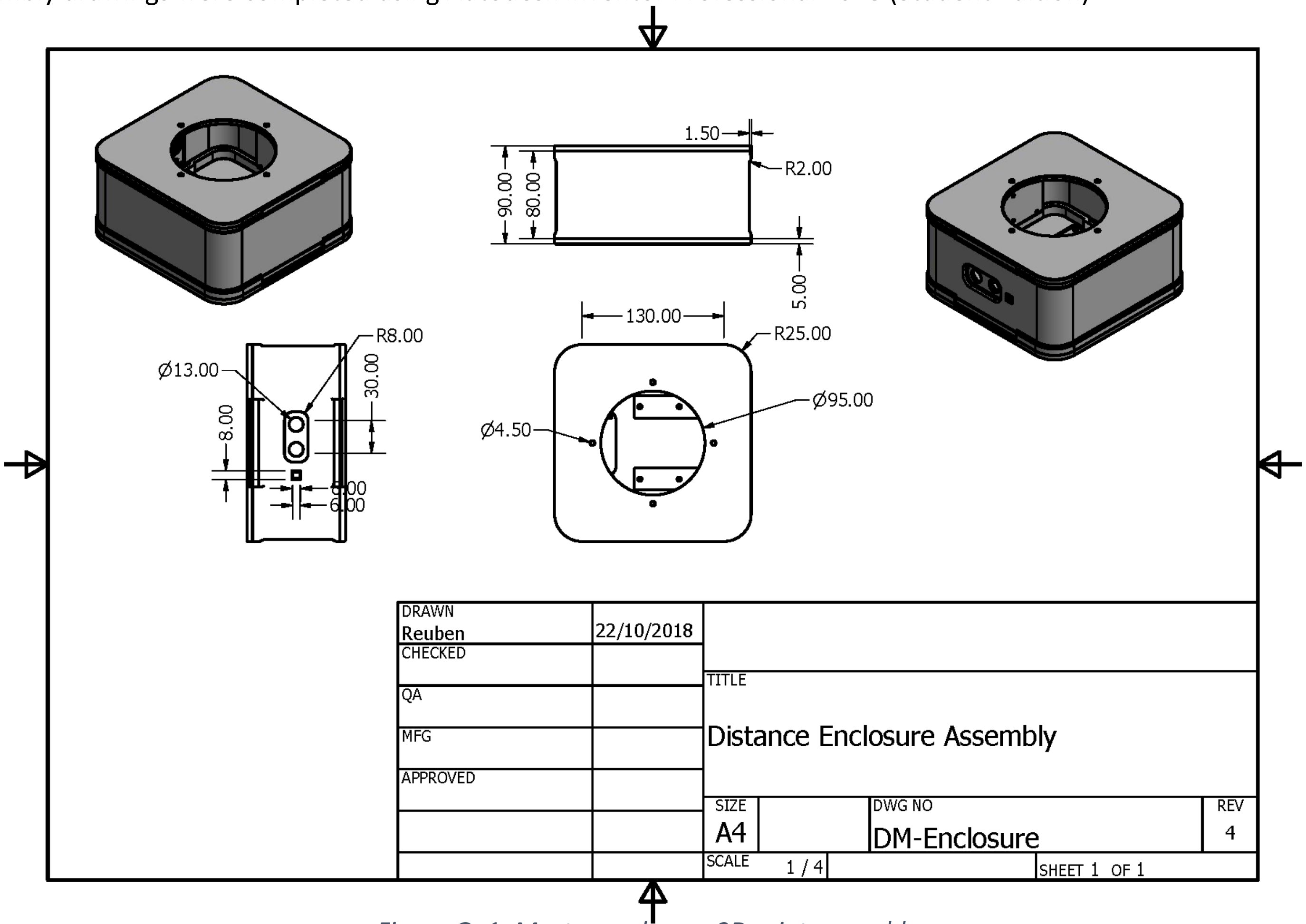


Figure G- 1: Master enclosure 3D print assembly

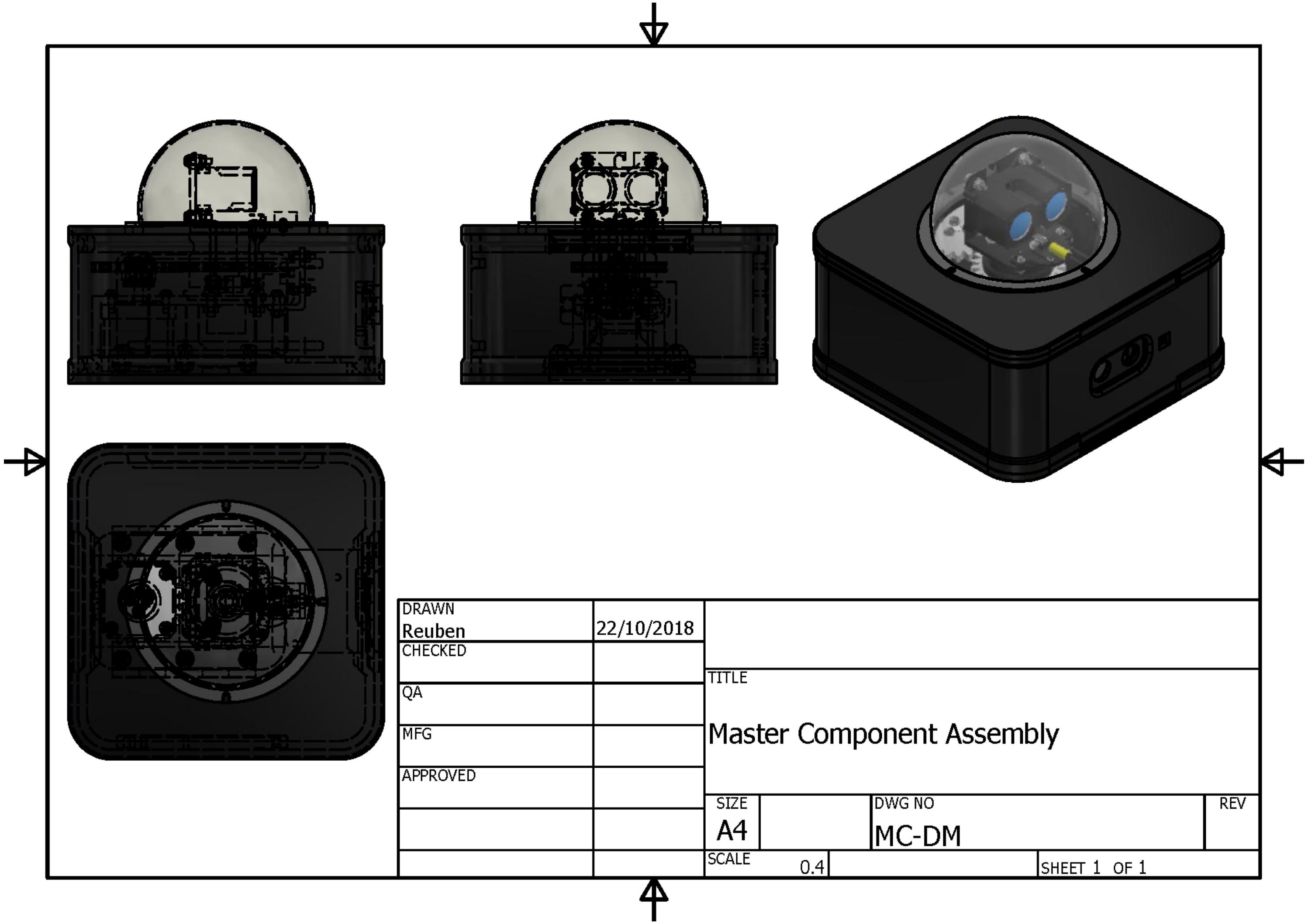


Figure G- 2: Master component assembly engineering drawing

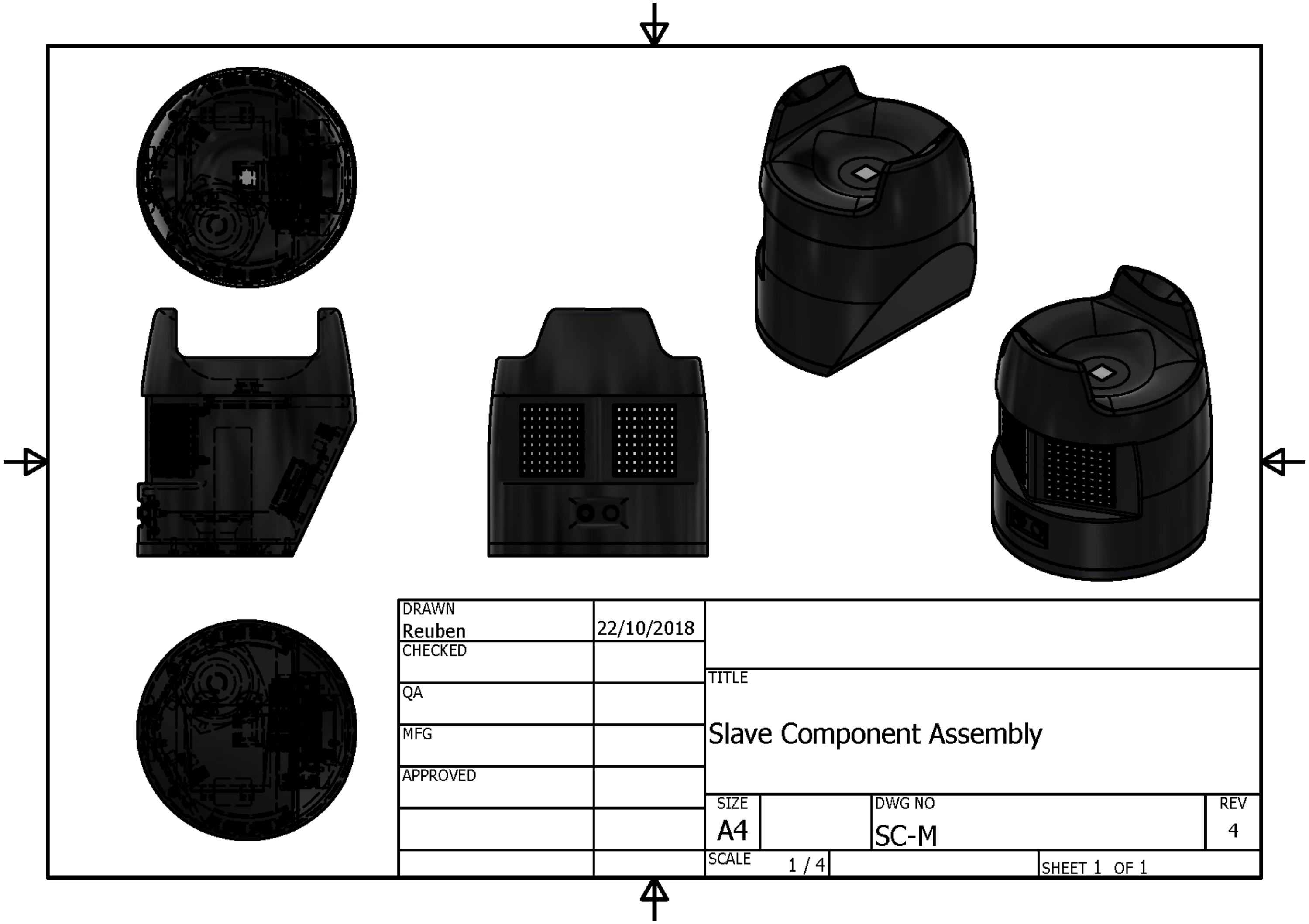


Figure G- 3: Slave component assembly engineering drawing

G.2. Engineering Part Drawings

All engineering part drawings were completed using Autodesk Inventor Professional 2018 (Student Edition).

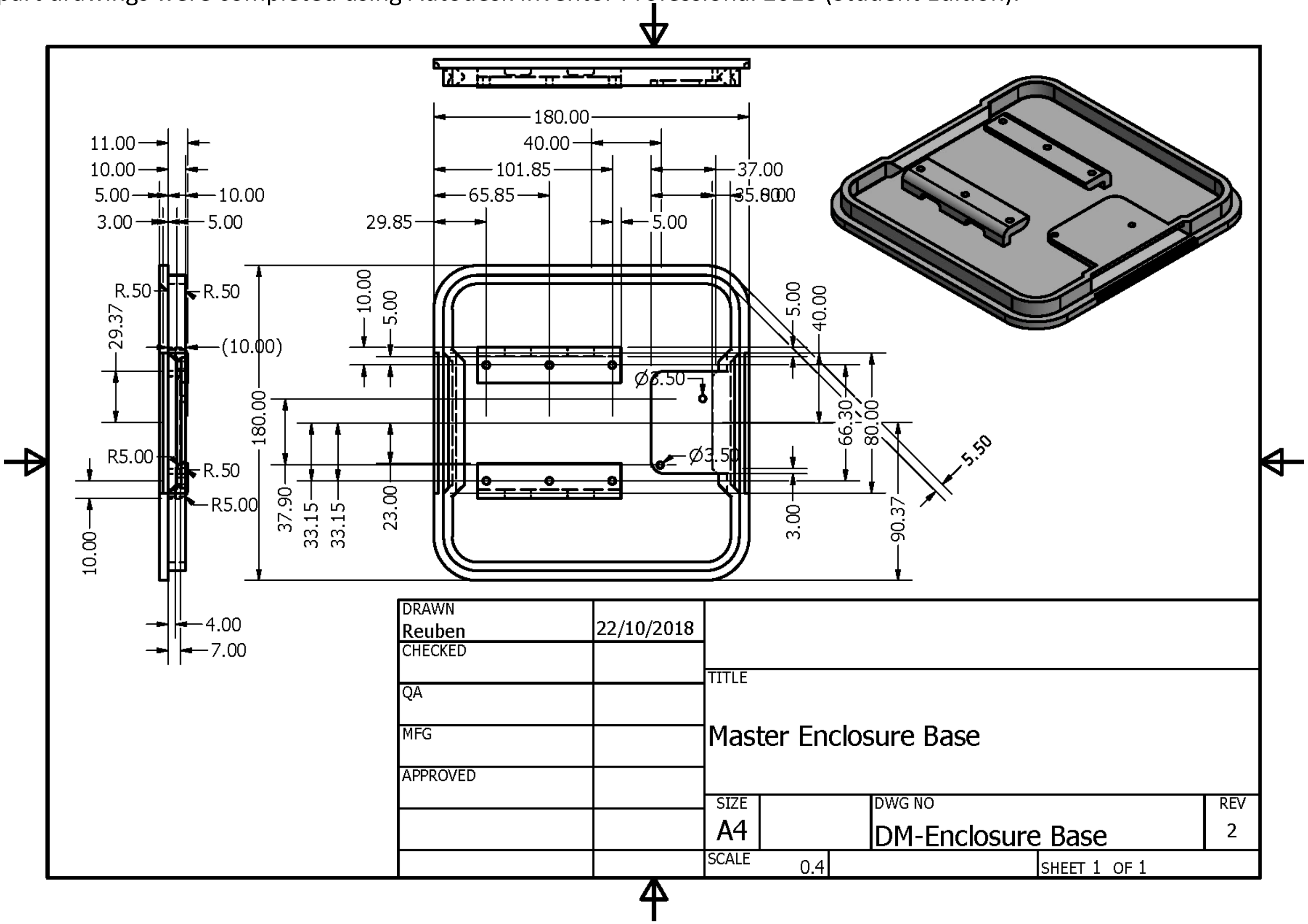


Figure G- 4: Master enclosure base engineering drawing

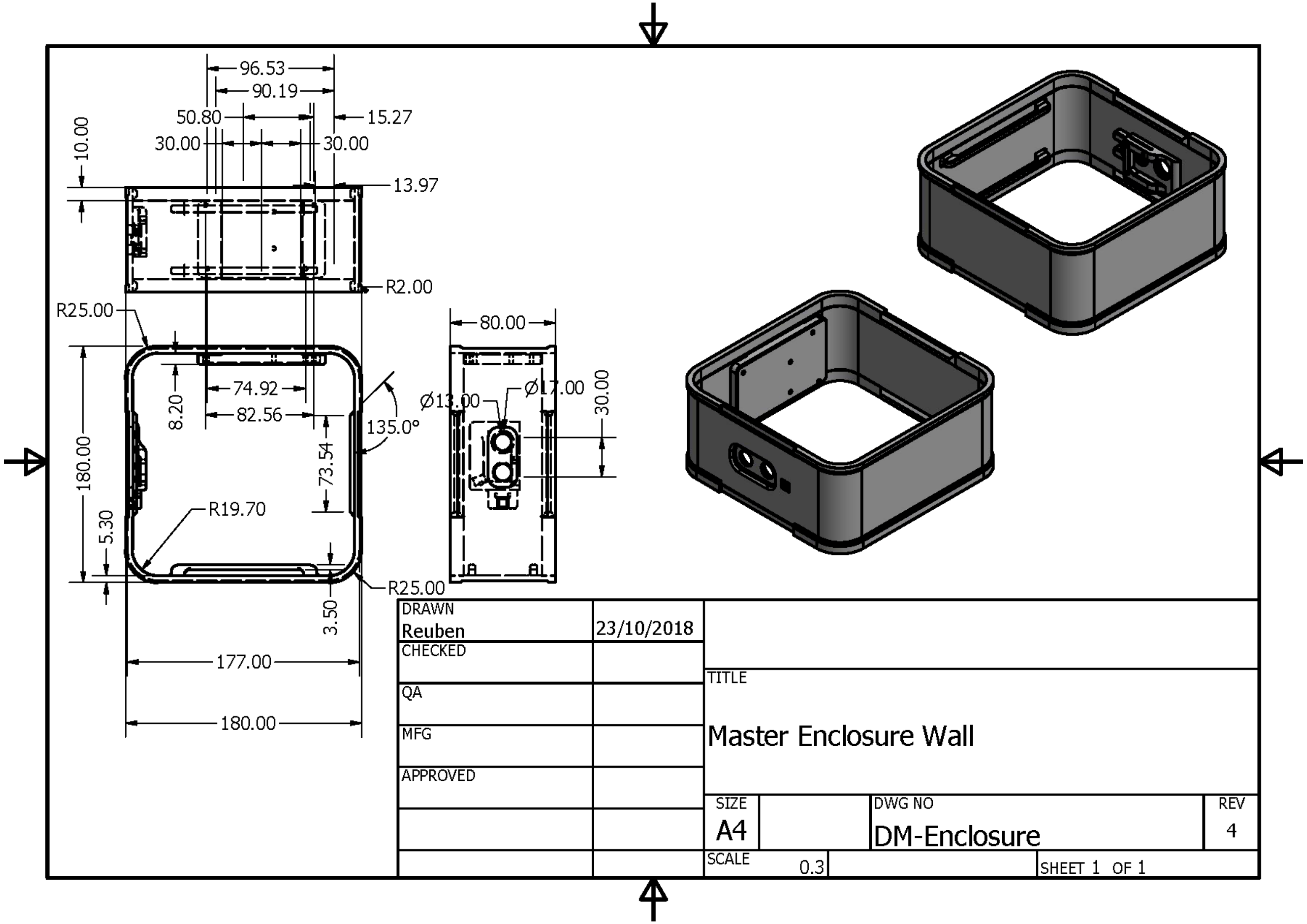


Figure G- 5: Master enclosure wall engineering drawing

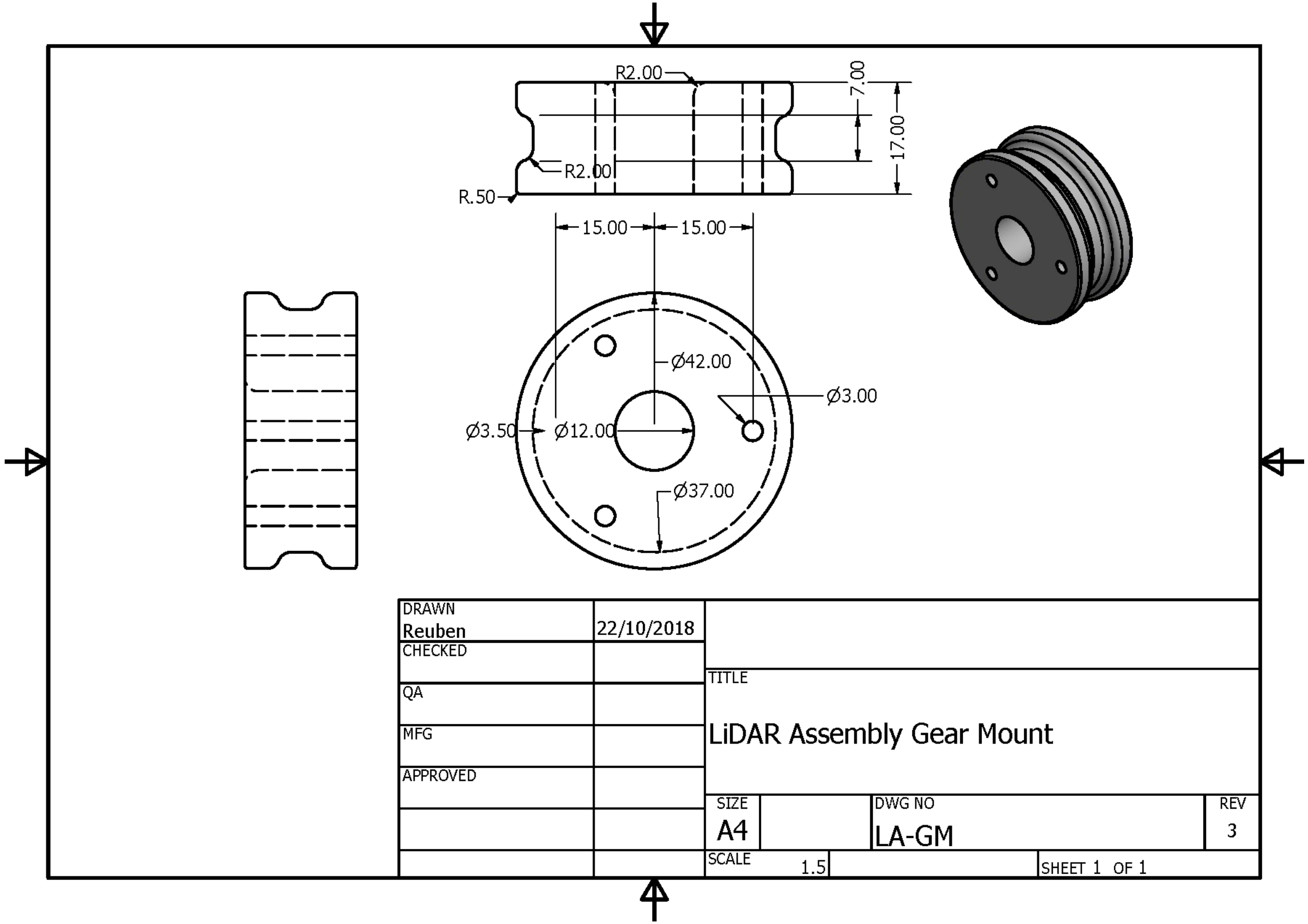


Figure G- 6: LiDAR assembly gear mount engineering drawing

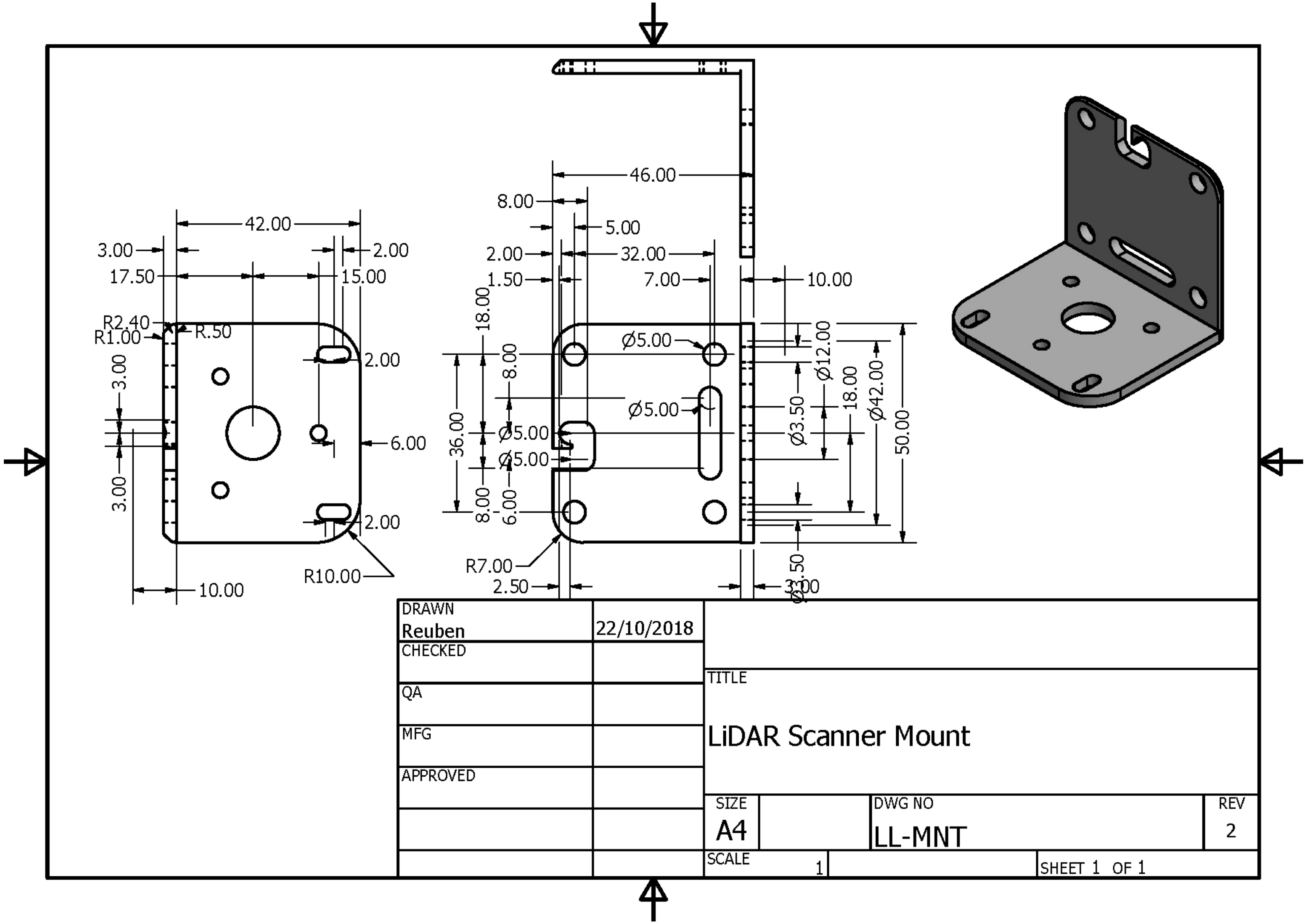


Figure G- 7: LiDAR mount engineering drawing

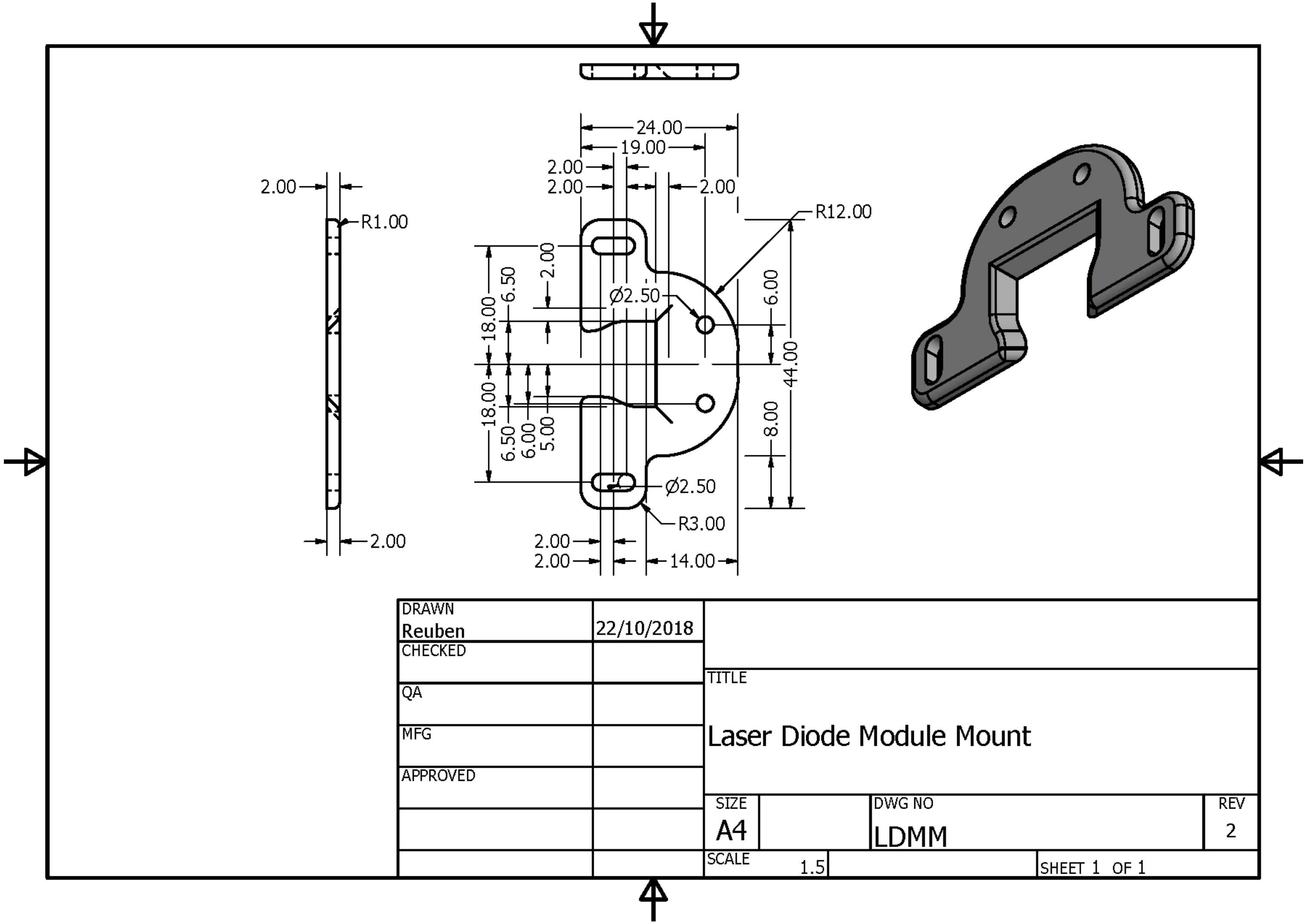


Figure G- 8: Laser diode mount engineering drawing

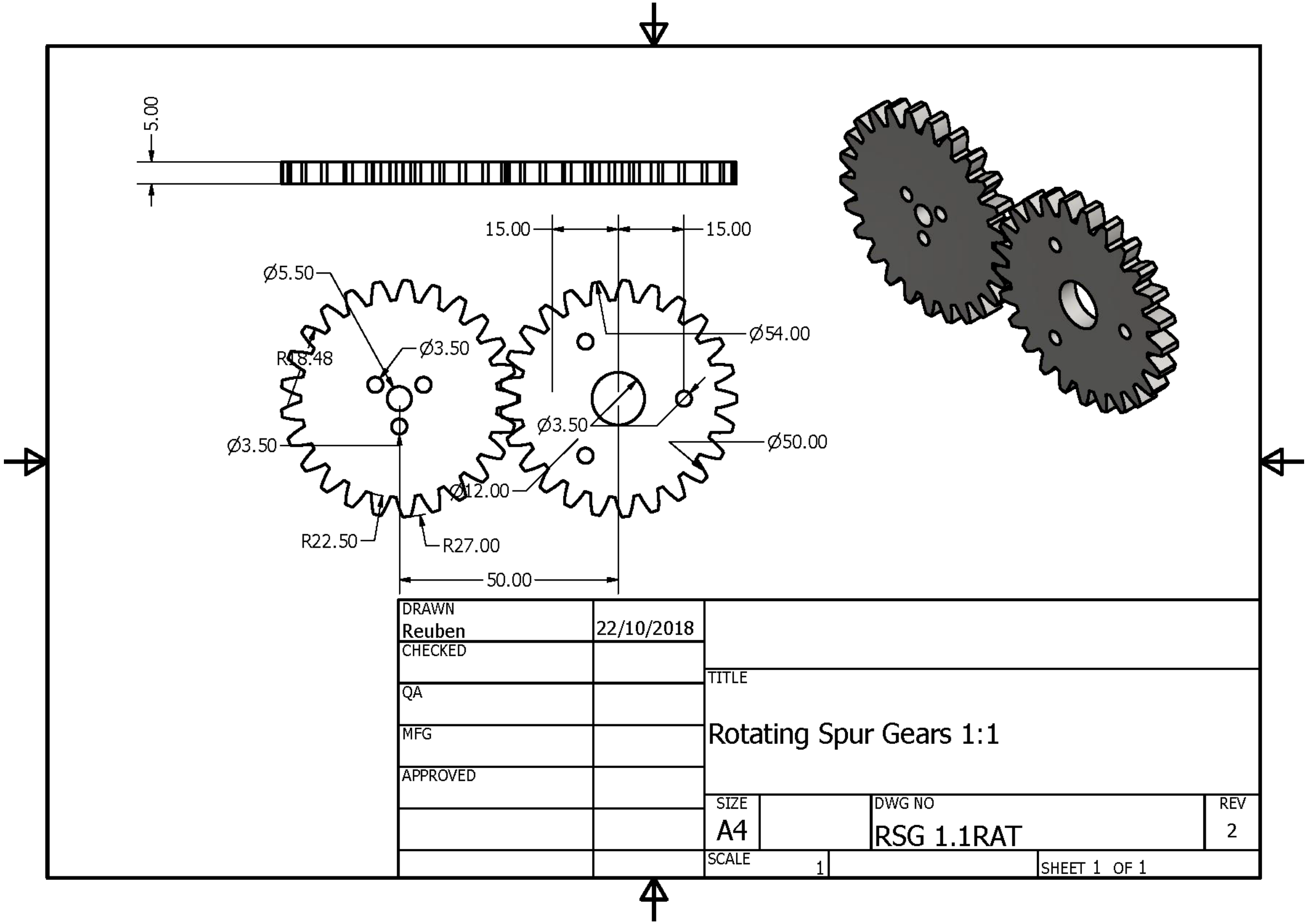


Figure G- 9: Rotating spur gears engineering drawing

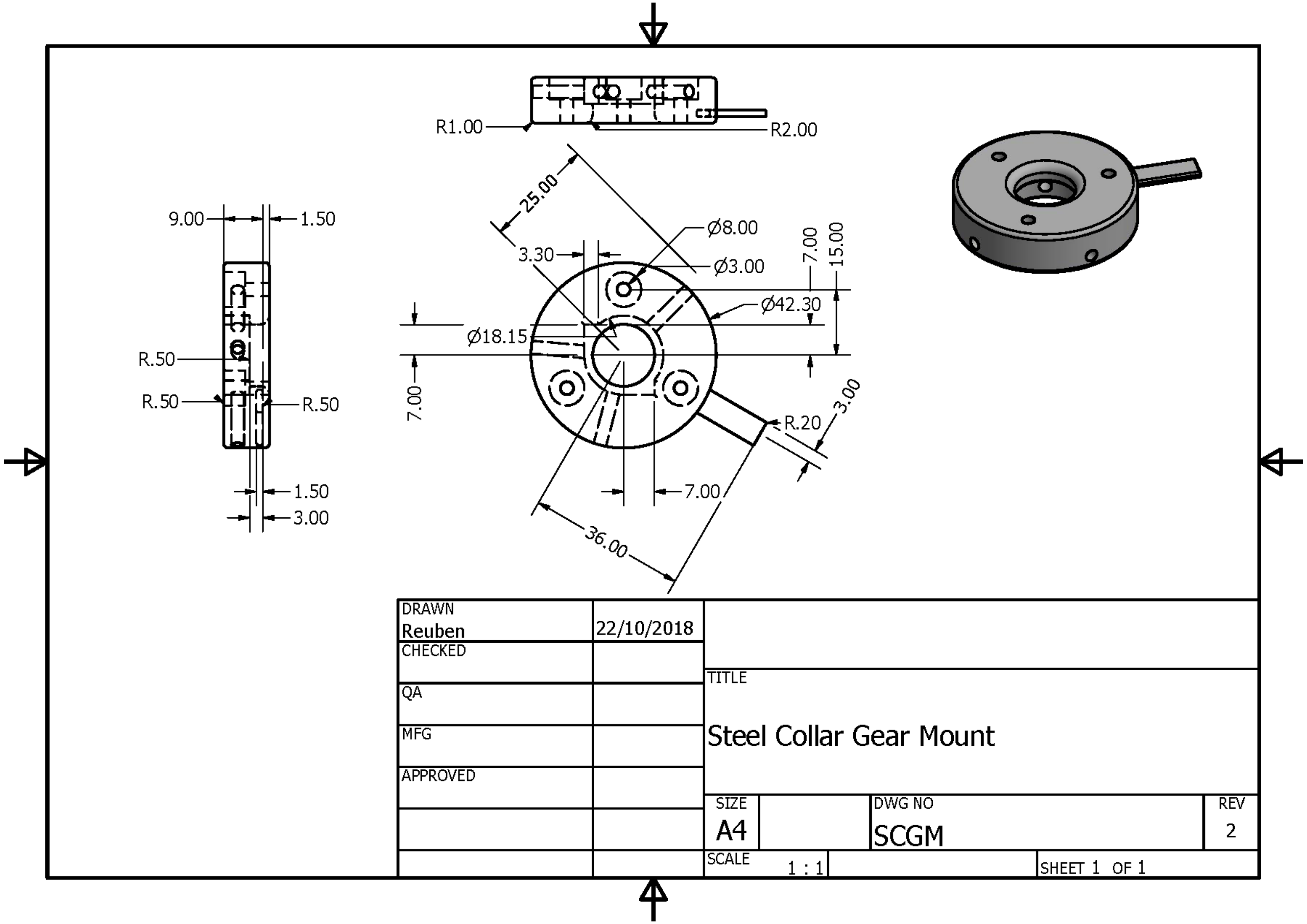


Figure G- 10: Steel collar gear mount engineering drawing

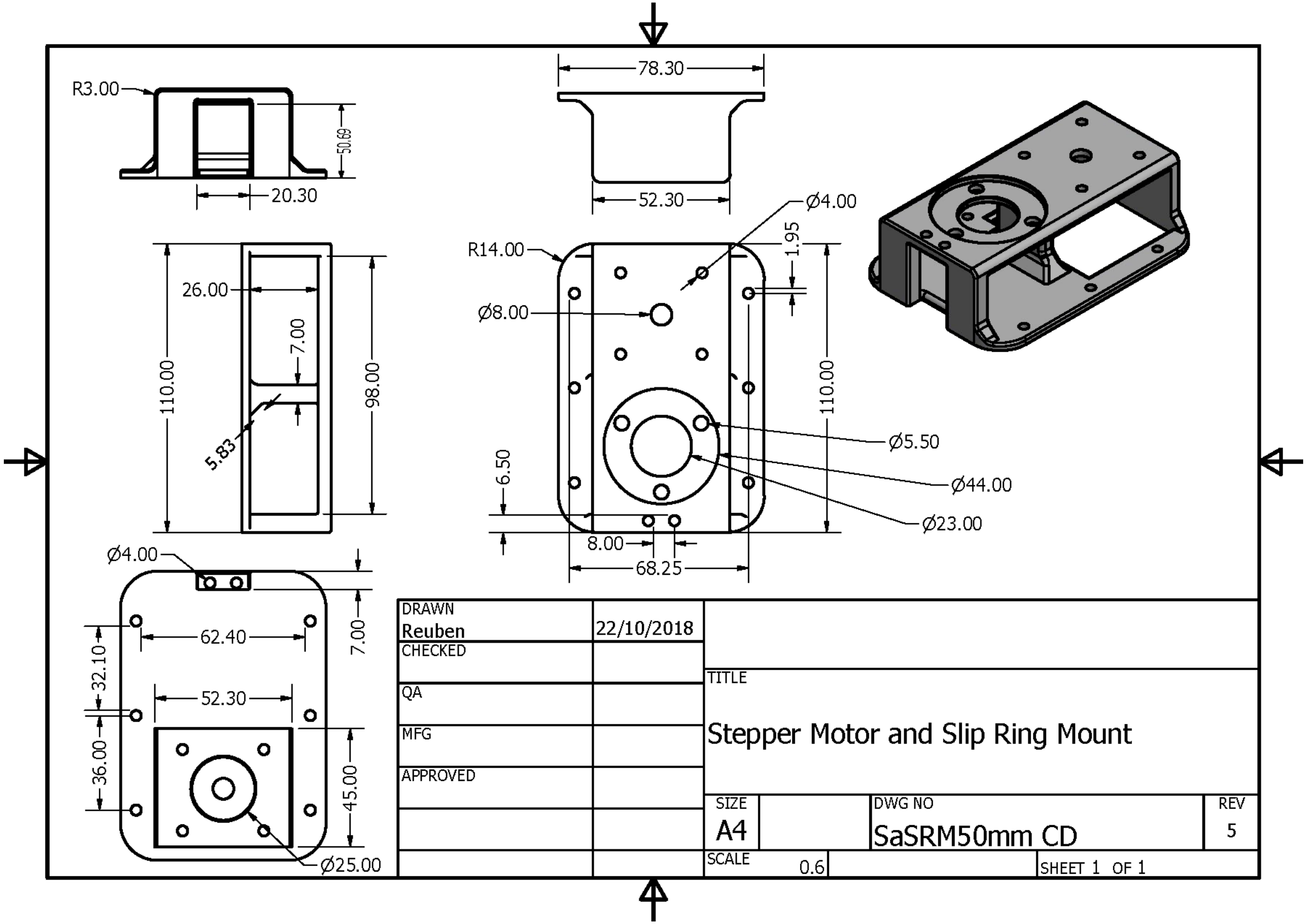


Figure G- 11: Stepper motor and slip ring mount engineering drawing

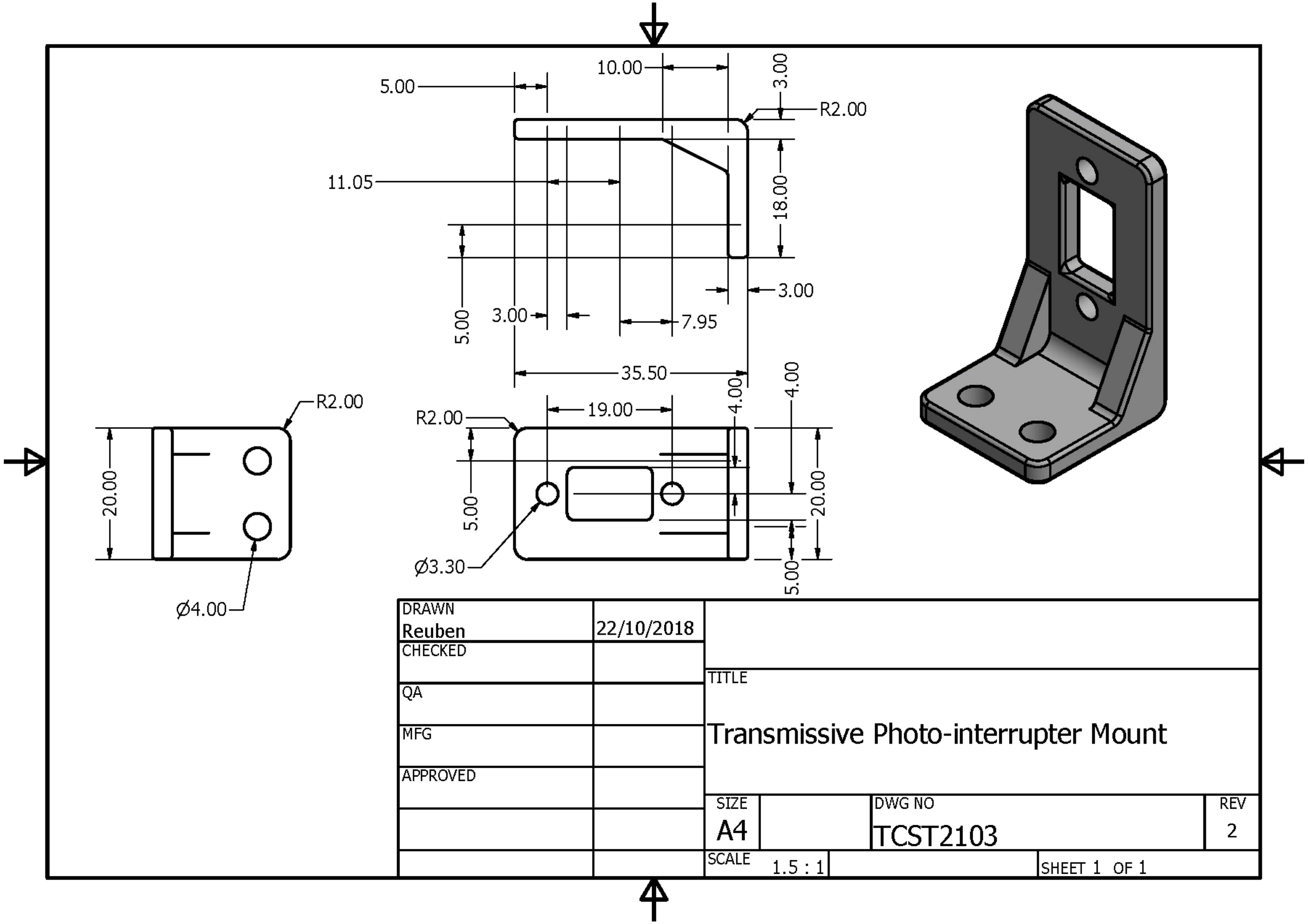


Figure G- 12: Transmissive photo-interrupter mount engineering drawing

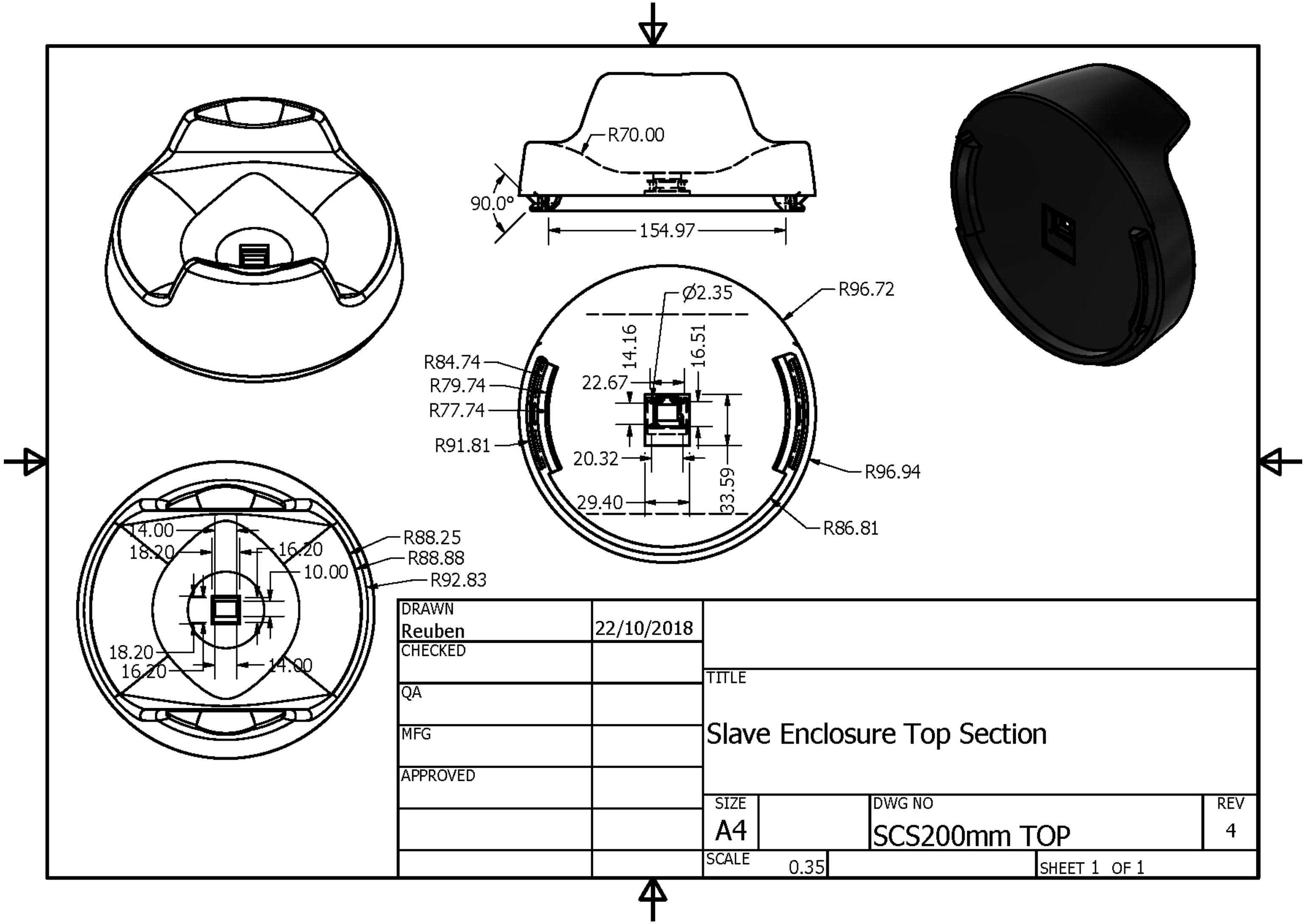


Figure G- 13: Slave enclosure top section engineering drawing

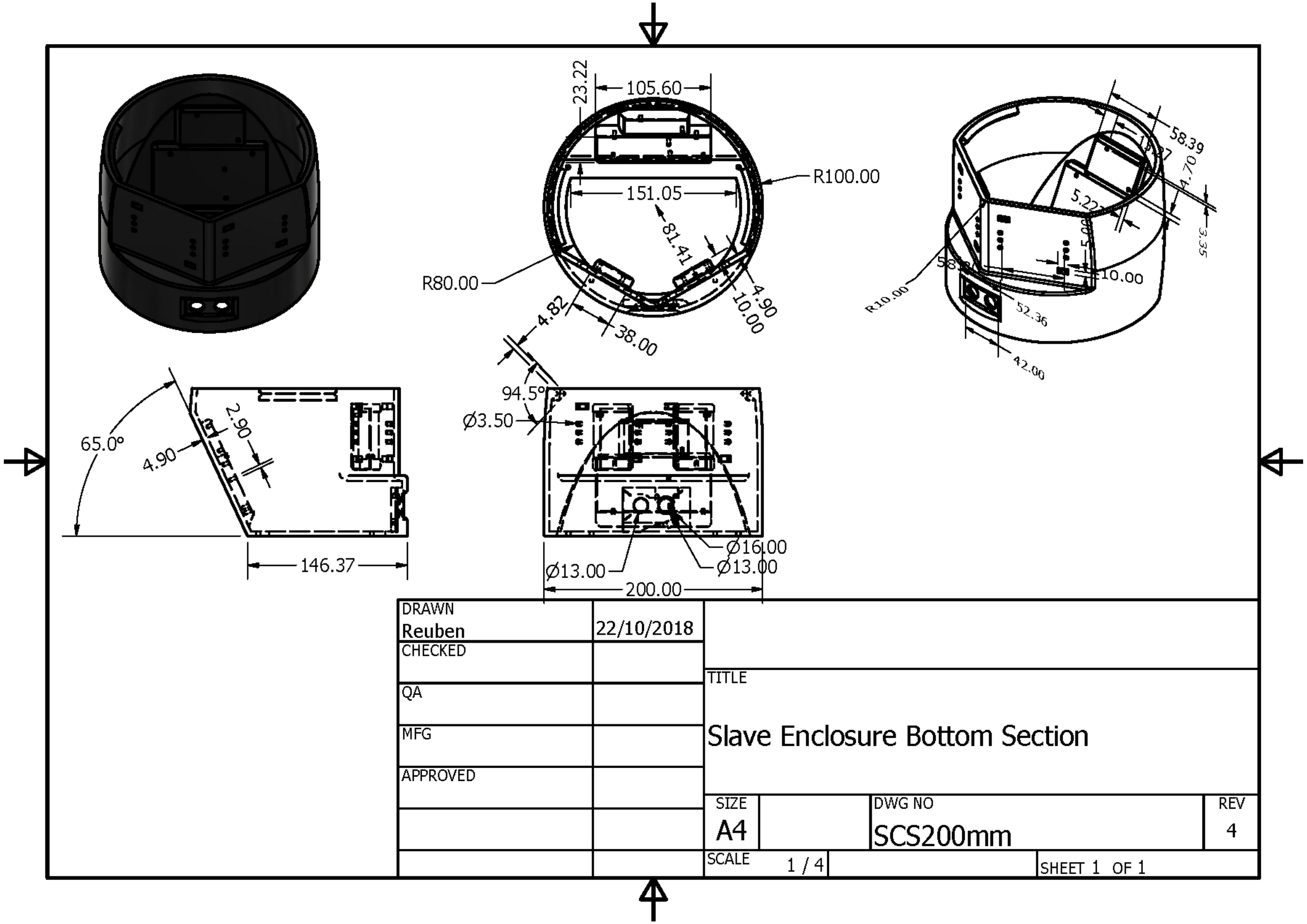


Figure G- 14: Slave enclosure bottom section engineering drawing

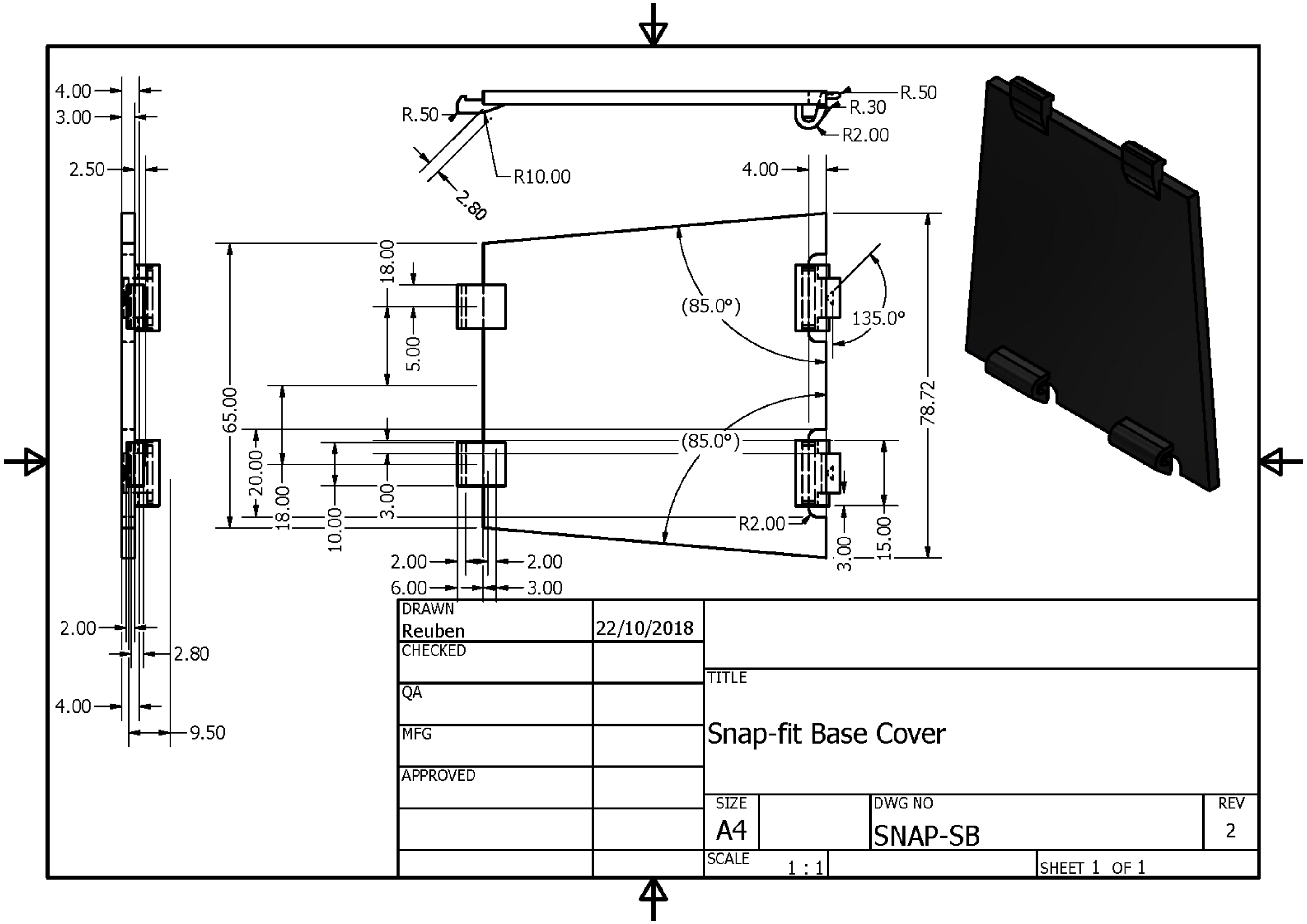


Figure G- 15: Snap-fit base cover engineering drawing

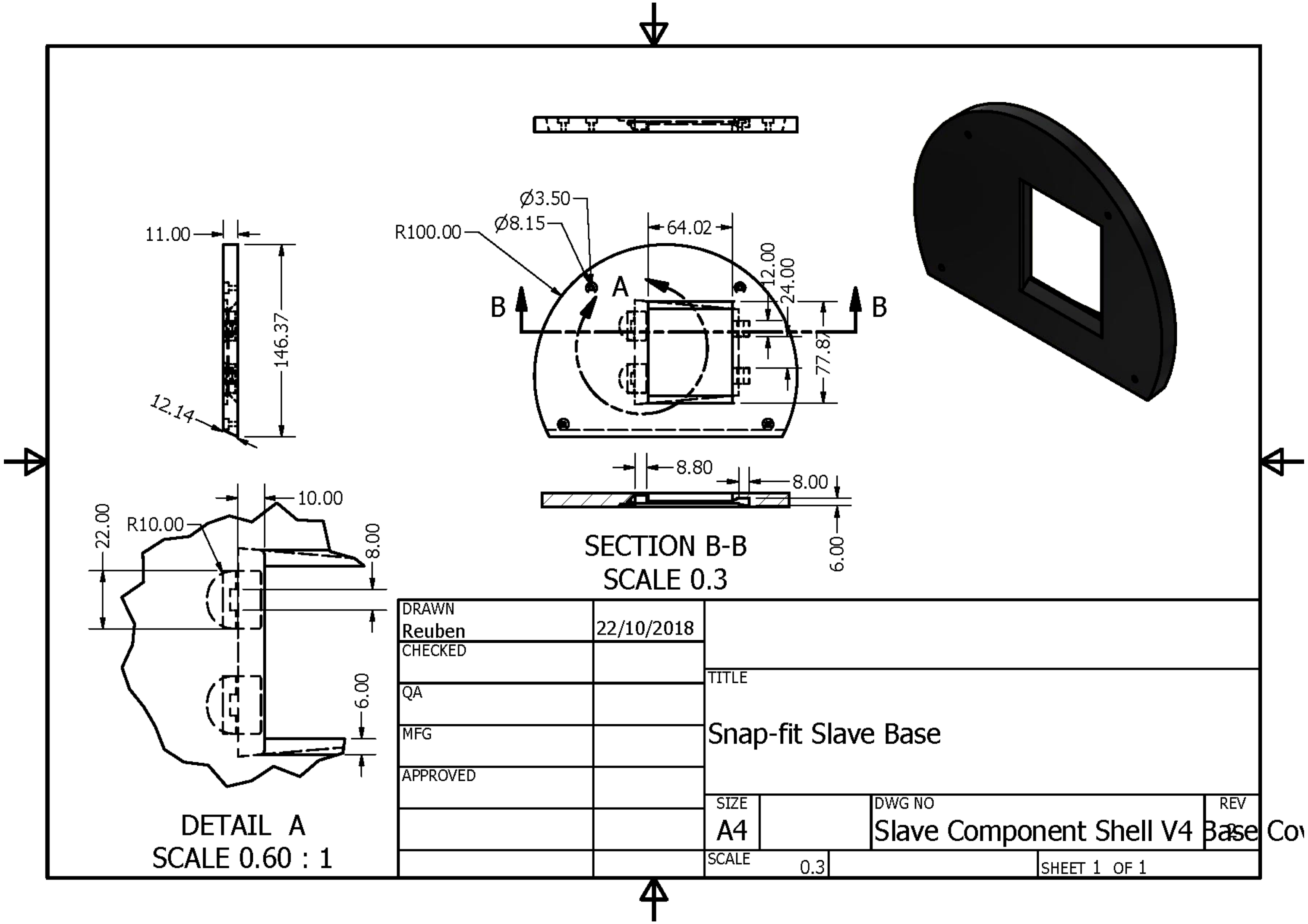


Figure G- 16: Snap-fit slave base engineering drawing

Appendix H

H.1. Functional Block Diagrams

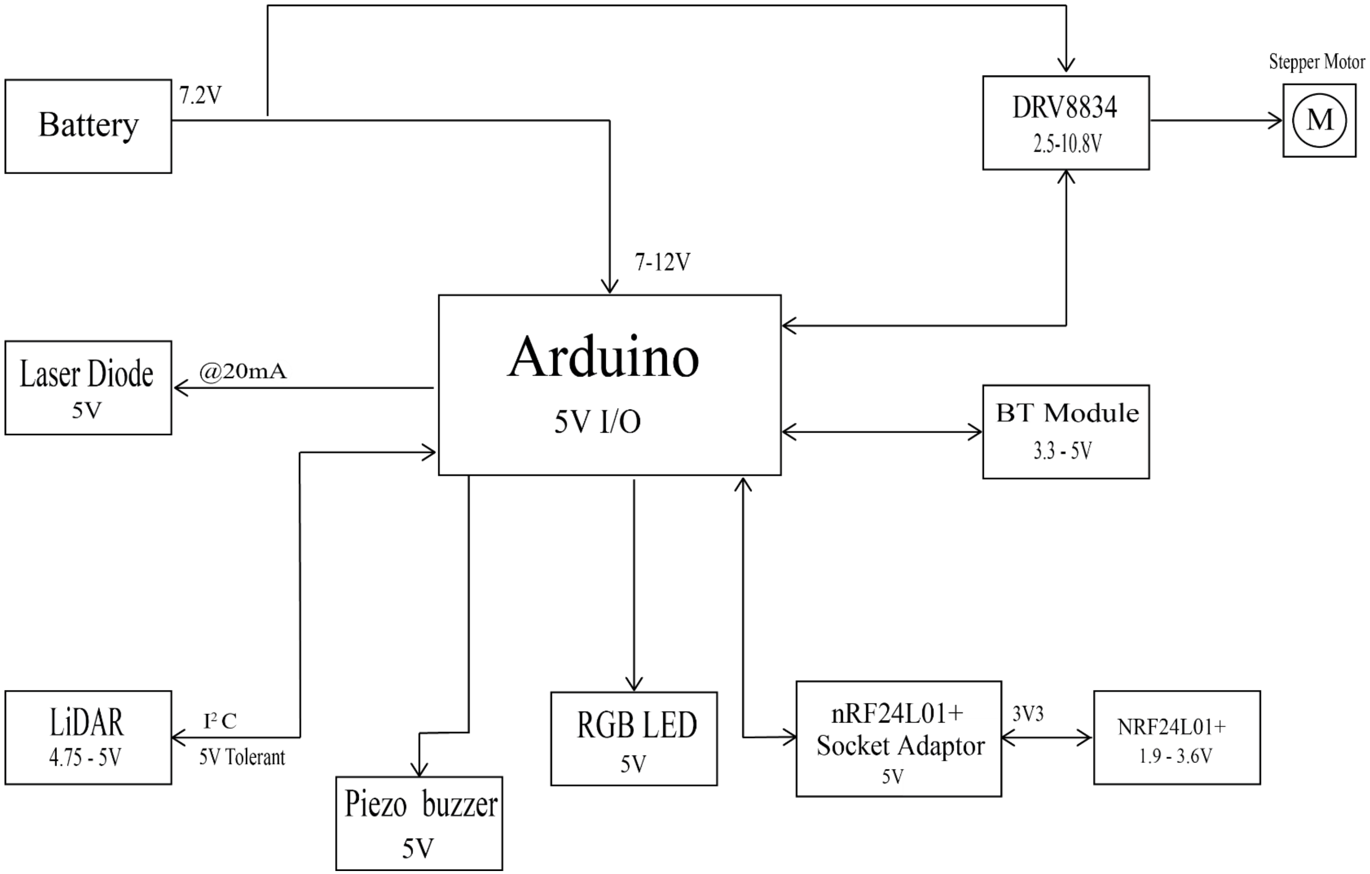


Figure H- 1: Master component functional body diagram

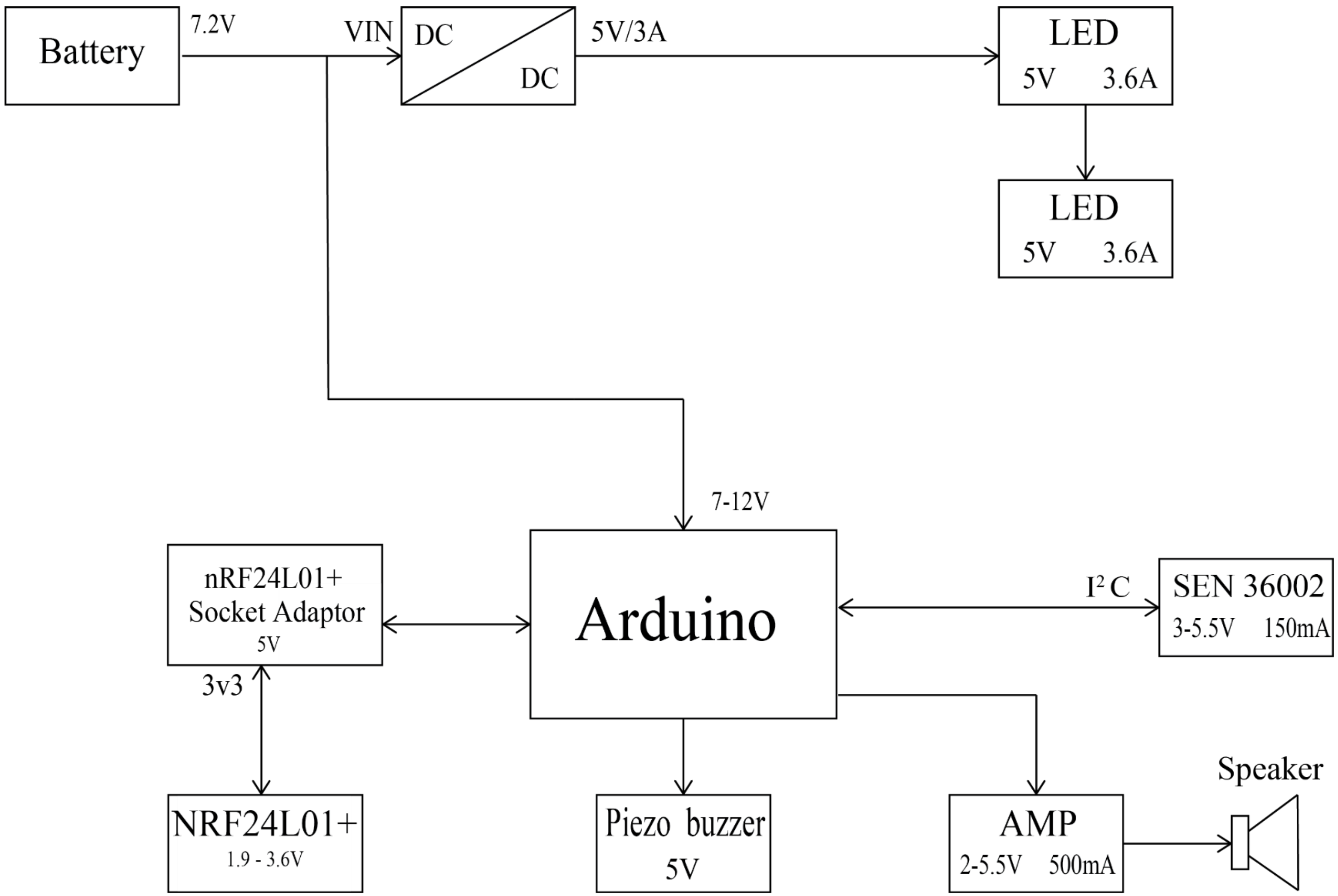


Figure H- 2: Slave component functional body diagram

Appendix I

I.1. Electronic Hardware Schematics

Master and slave electronic hardware schematics were completed using Adobe Illustrator CS6.

Both schematics have been scaled to fit an A4 page for printing purposes (actual size A3)

I.1.1. Master Schematic

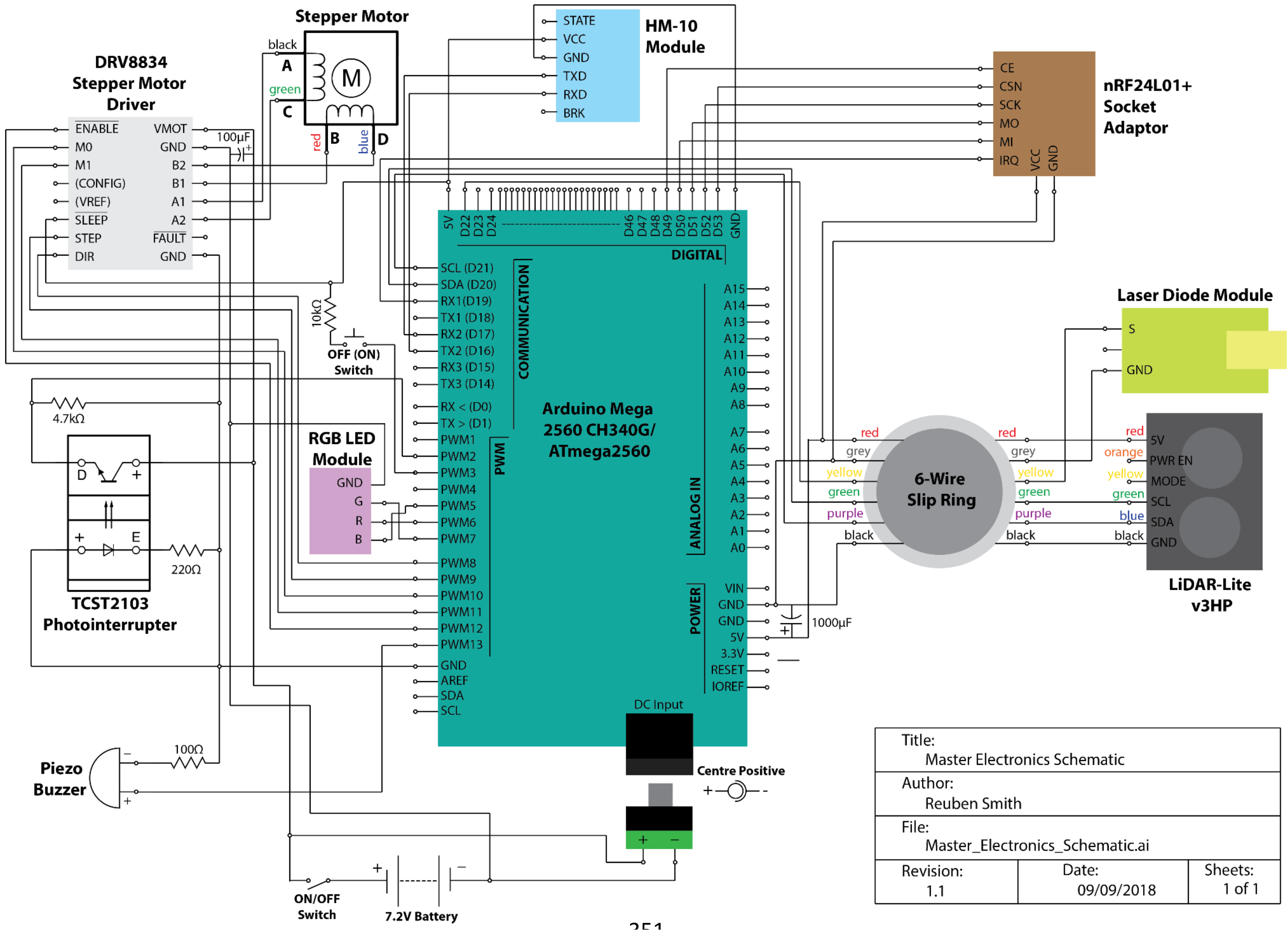
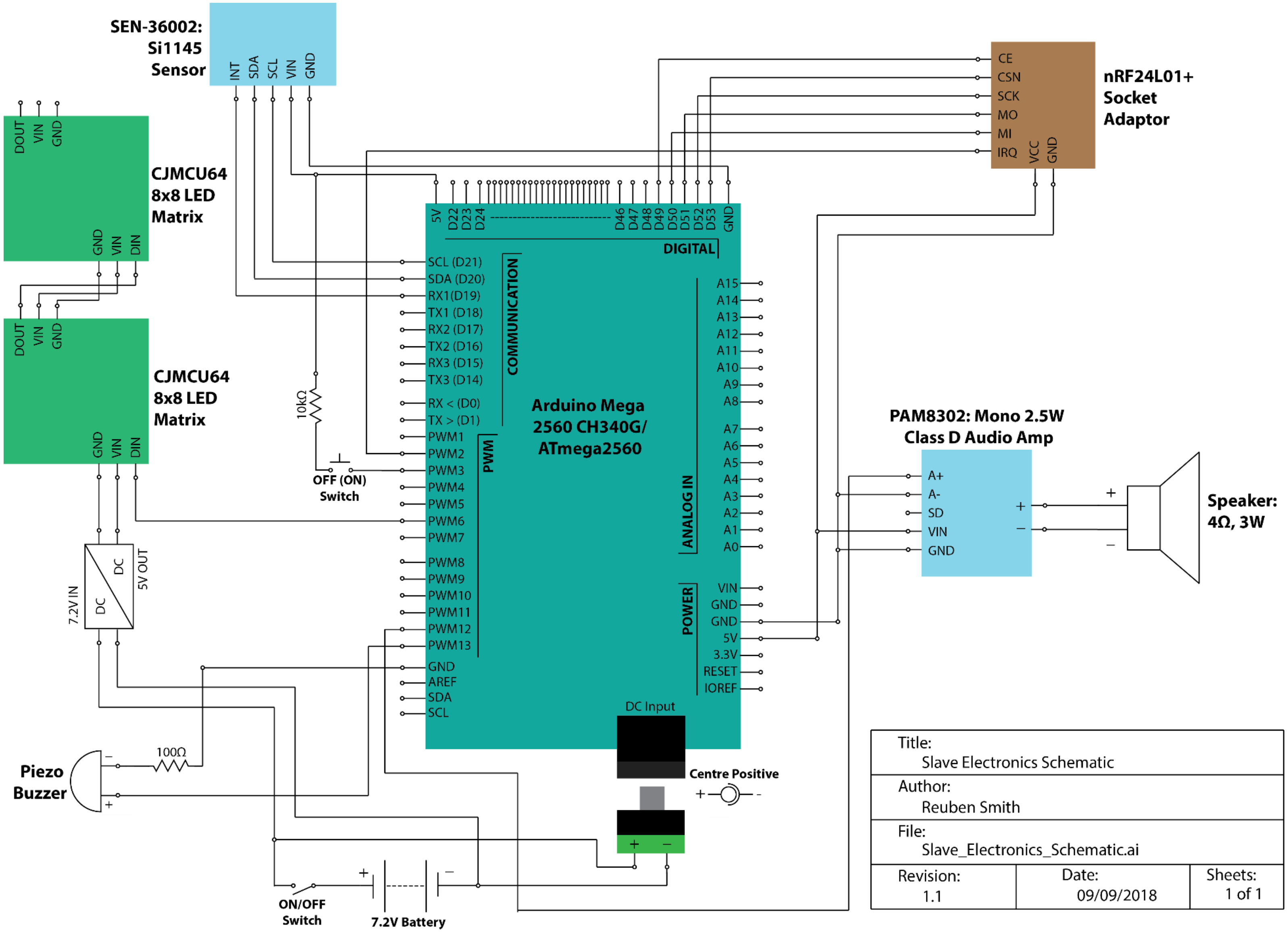


Figure I- 1:Master electronic schematic diagram

I.1.2. Slave Schematic



Title: Slave Electronics Schematic		
Author: Reuben Smith		
File: Slave_Electronics_Schematic.ai		
Revision: 1.1	Date: 09/09/2018	Sheets: 1 of 1

Figure I- 2: Slave electronic schematic diagram

I.2. Prototype Board Hardware Placement & Wiring Schematic

Master and slave prototype board hardware placement and wiring schematics were completed using Adobe Illustrator CS6.

Both schematics have been scaled to fit an A4 page for printing purposes (actual size A3).

I.2.1. Master Prototype Board

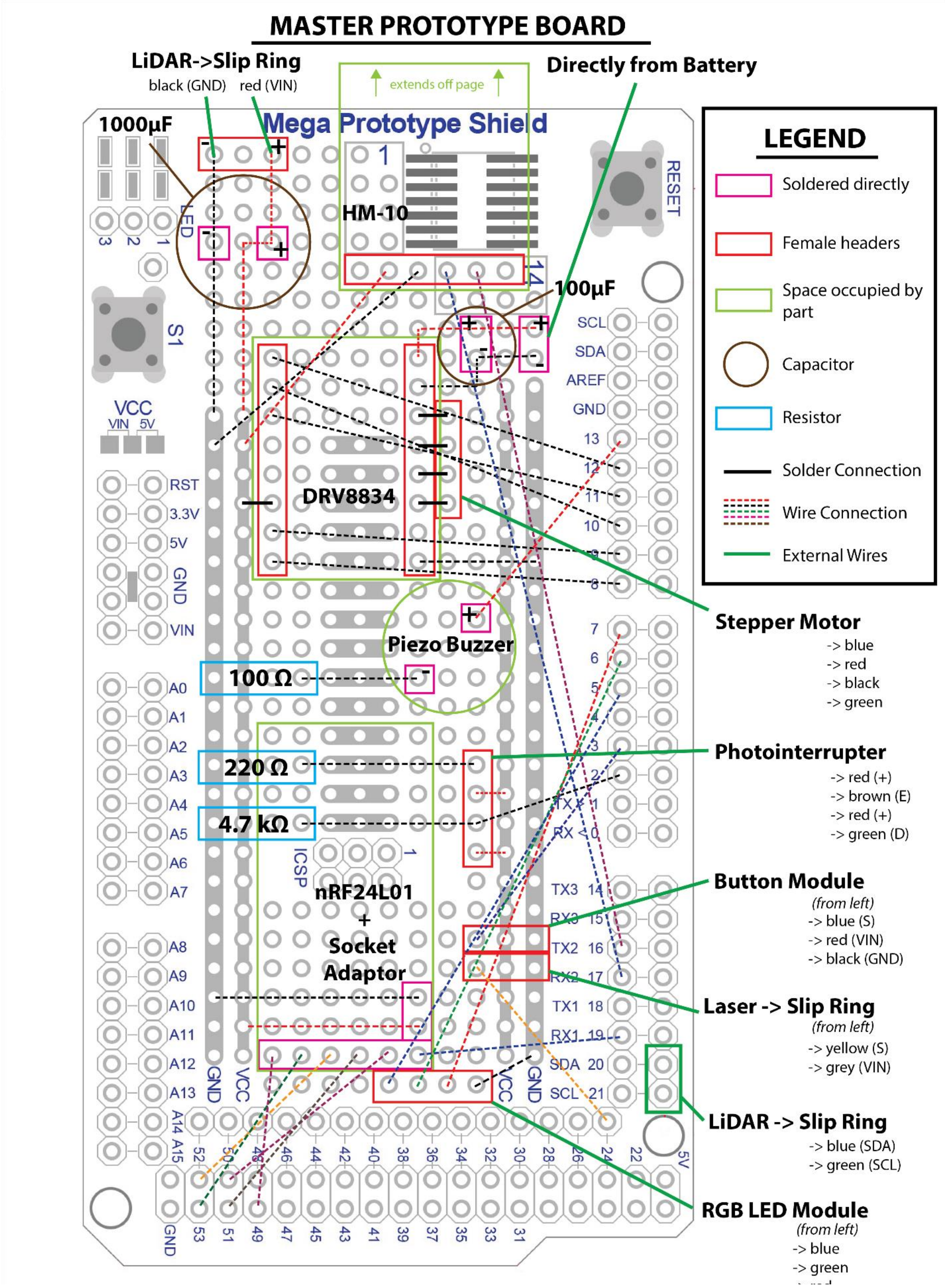


Figure I- 3: Master prototype board hardware and wiring configuration

I.2.2. Slave Prototype Board

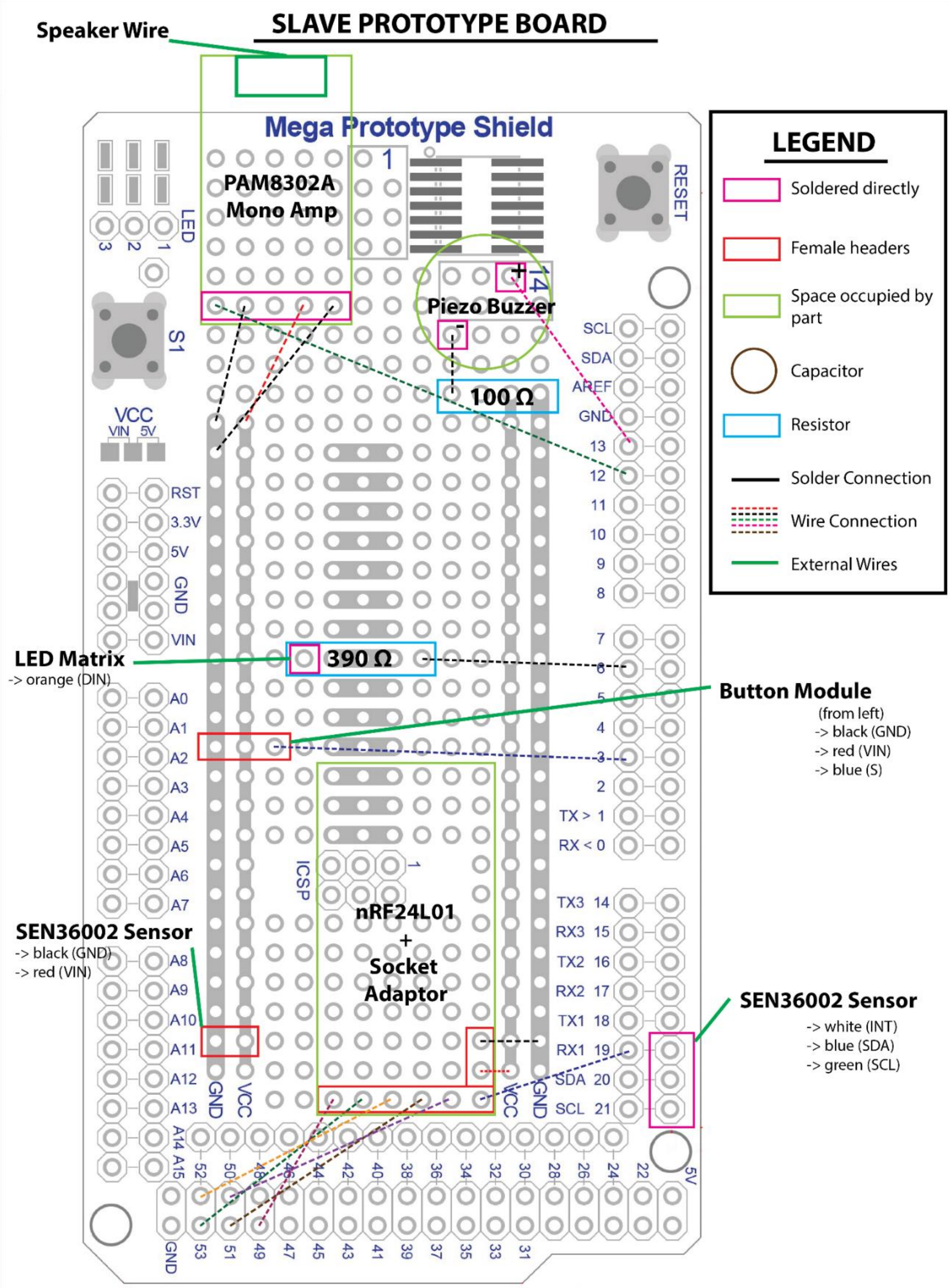


Figure I- 4: Slave prototype board hardware and wiring configuration

Appendix J

J.1. Program That Converts Hexadecimal Values from a LED Matrix Editor to Arduino Friendly Character Arrays for the LED Matrix Displays

```
#include <string>
#include <iostream>
#include <sstream>
#include <algorithm>
#include <bitset>

struct hexData {
	hexData() {}
	hexData(std::string c, std::string v) : character(c), hexValue(v) {}
	std::string character;
	std::string hexValue;
};

std::string stringVal[]{
	"LETTERS_UPPER",
	"LETTERS_LOWER",
	"DIGITS",
	"RANDOM_CHARACTERS"
};

const std::string charLettersUpper[][1]{
	"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T"
,"U","V","W","X","Y","Z"
};

const std::string charLettersLower[][1]{
	"a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t"
,"u","v","w","x","y","z"
};

const std::string charDigits[][1]{
	"0","1","2","3","4","5","6","7","8","9"
};

const std::string charRandomCharacters[][1]{
	"Tick",
	"Cross",
	"Exclamation Mark",
	"Question Mark",
	"Happy",
	"Sad",
	"Down",
	"Up",
	"Left",
	"Right",
	"Square",
	"Heart",
	"Circle",
	"Star",
	"Pacman",
	"Ghost"
};

// Paste your upper-case letters (A-Z) here
std::uint64_t hexLettersUpper[][1]{
```

```
        0xc3c3ffffc3663c18,
        0x3f7f637f3f737f3f,
        0x3c7ee70303e77e3c,
        0x1f7f73e3e3737f1f,
        0xffff037f7f03ffff,
        0x0303033f3f03ffff,
        0x3c7ee7f303e77e3c,
        0xc3c3c3ffffc3c3c3,
        0xffff18181818ffff,
        0x1e3f33303030fefe,
        0xe3733b1f1f3b73e3,
        0xffff030303030303,
        0x6363636b7f7f7763,
        0xc3e3f3fbdfcfc7c3,
        0x3c7ee7c3c3e77e3c,
        0x03033f7fe3e37f3f,
        0xe03c7ed7c3e77e3c,
        0xe3733b1f7fe3ff7f,
        0x7effe07e3f03ff7e,
        0x181818181818ffff,
        0x7effc3c3c3c3c3c3,
        0x081c366363636363,
        0x63777f7f6b636363,
        0xc3e77e3c183ce7c3,
        0x181818183c7ee7c3,
        0xffff0e1c3860ffff
};

// Paste your lower-case letters (a-z) here
const uint64_t hexLettersLower[][1] = {
        0x7cfee7fee0fe7e00,
        0x3f7fe3e37f3f0303,
        0x3c7ee703e77e3c00,
        0xfcfec7c7fefcc0c0,
        0x7c7e077f677e3c00,
        0x0c0c0c3f3f8cfc78,
        0x7efec0f8ccccfcf8,
        0x6363637f3f030303,
        0x3e1c1c1c1c001c1c,
        0x3c7eeee0e000e0e0,
        0xe6763e0e3e760606,
        0x1c1c1c1c1c1c1c1c,
        0x636b6b6b7f370300,
        0x636363637f3f0300,
        0x3c7ee7c3e77e3c00,
        0x0303033f677f3f03,
        0x70f0303e333f3e30,
        0x030303637f3f0300,
        0x7effe07e07ff7e00,
        0x387858187e7e1818,
        0xbcfec6c6c6c6c600,
        0x107cfec6c6c6c600,
        0x3e7f6b6b6b636300,
        0xc3663c183c66c300,
        0x7cfec3c0fcc6c6c6,
        0xfefe0c3860fefe00
};

// Paste your digits (0-9) here
const uint64_t hexDigits[][1] = {
        0x3c66cfdbf3e37e3c,
  0x7e7e1818181e1c18,
```

```
  0xffff0c38f0e77e3c,
  0x7effe378e0e77e3c,
  0x30307f73363c3830,
  0x7effc3e07f03ffff,
  0x7efee77f07c7fe3c,
  0x1818183070e3fffe,
  0x7effe73cffe77e3c,
  0x3e7fe0fee7e77e7c
};

// Paste your random characters here (make sure to update charRandomCharacters to
match)
const uint64_t hexRandomCharacters[][1] = {
	0x0c0e1b3160c08000,
	0xc3663c183c66c300,
	0x1818001818181818,
	0x1818001878c2e63c,
	0x3c7ec38100666666,
	0x81c37e3c00666666,
	0x081c3e7f1c1c1c1c,
	0x1c1c1c1c7f3e1c08,
	0x00080cfefffe0c08,
	0x0010307fff7f3010,
	0xffffc3c3c3c3ffff,
	0x183c7efffffff6600,
	0x3c7ee7c3c3e77e3c,
	0xc3667e3c7eff1818,
	0x3cfe7f1f3fee7e3c,
	0x557f7f7f577f3e1c
};

const int LETTERS_LENG = sizeof(hexLettersUpper) / sizeof(hexLettersUpper[0]);
const int DIGITS_LENG = sizeof(hexDigits) / sizeof(hexDigits[0]);
const int RANDOM_CHARACTERS_LENG = sizeof(hexRandomCharacters) /
sizeof(hexRandomCharacters[0]);

std::string rearrangeForMatrix(std::string binaryStr) {
	std::string tempStr;
	for (int i = 0; i < 63; i++) {
		tempStr = binaryStr.substr(binaryStr.length() - 1, binaryStr.length());
		binaryStr.insert(0 + i * 1, tempStr);
		binaryStr.erase(binaryStr.length() - 1, binaryStr.length());
	}
	return binaryStr;
}

/* Print Arduino friendly letters array */
void printArduinoFriendlyCode(hexData data[], const uint64_t hexValues[], const
std::string characters[], const std::string name, const int LENGTH) {
	for (int i = 0; i < LENGTH; i++) {
		if (i == 0) {
			std::cout << "const char " << name << "[][65] PROGMEM = { //
Store in PROGMEM to save SRAM" << std::endl;
		}
		std::string binaryVal = std::bitset<64>(hexValues[i]).to_string();
		data[i] = hexData(characters[i], binaryVal);
		data[i].hexValue = rearrangeForMatrix(data[i].hexValue);
		data[i].hexValue.insert(0, "\"");
		data[i].hexValue.append("\"");
		data[i].hexValue.insert(0, "{ ");
		if (i != LENGTH - 1) {
			data[i].hexValue.append("},\n");
```

```
		}
		else {
			data[i].hexValue.append("}");
		}
		std::cout << "  // [" << i << "] " << data[i].character << ": \n  " <<
data[i].hexValue << std::endl;
	}
	std::cout << "};\n" << std::endl;
}

int main() {
	hexData data[LETTERS_LENG];
	hexData dataDigits[DIGITS_LENG];
	hexData dataRandomCharacters[RANDOM_CHARACTERS_LENG];
	for (int i = 0; i < 4; i++) {
		if (i == 0) {
			printArduinoFriendlyCode(data, *hexLettersUpper,
*charLettersUpper, stringVal[i], LETTERS_LENG);
		}
		else if (i == 1) {
			printArduinoFriendlyCode(data, *hexLettersLower,
*charLettersLower, stringVal[i], LETTERS_LENG);
		}
		else if (i == 2) {
			printArduinoFriendlyCode(dataDigits, *hexDigits, *charDigits,
stringVal[i], DIGITS_LENG);
		}
		else {
			printArduinoFriendlyCode(dataRandomCharacters,
*hexRandomCharacters, *charRandomCharacters, stringVal[i], RANDOM_CHARACTERS_LENG);
		}
	}
}
```

J.2. Permutation Generator Program & Distance Checker

```
/*
* Permutation generator + distance checker: a program that generates all possible
permutations for a set of given input values.
*                                                                              It then
calculates the distance associated with each number sequence
*
*  This program allows simple copy + paste data in either a code friendly array, or
for values
*  to paste directly into excel; just change the printCodeFriendlyArray boolean.
*
*      printCodeFriendlySequences: set true to print sequences in a code friendly
array.
*                                set false to print sequences in a Microsoft Excel
friendly array.
*/

#include <iostream>
#include <algorithm>    // Needed for uint_x variables
#include <stdlib.h>
#include <set>
#include <math.h>
#include <iomanip>
#include <vector>
#include <string>

#define MARKER_NUM 6
```

```
/* Calculate the factorial (!) using an iterative process, where each
digit up to the value digit is multiplied together
*/
uint32_t calcFactorial(uint8_t value)
{
	uint32_t result = 1;
	for (int i = 1; i <= value; i++)
		result *= i;
	return result;
}

int main()
{
	bool printCodeFriendlySequences = true;
	bool badInput;
	std::string input;
	size_t INPUT_LEN;
	char permutationVals[10] = { '0' };
	do {
		badInput = false;
		std::cout << "Enter all the possible marker values as digits (e.g.
123456) -> ";
		std::cin >> input;
		INPUT_LEN = input.length();
		for (size_t i = 0; i < INPUT_LEN; i++) {
			if (!isdigit(input[i])) {
				std::cout << "Value entered was not all digits. Try
again." << std::endl;
				badInput = true;
				break;
			}
		}
	} while (badInput);
	for (size_t i = 0; i < INPUT_LEN; i++) {
		permutationVals[i] = input[i];
	}
	int lengthOfSequence = strlen(permutationVals);
	uint32_t PERMUTATIONS_LEN = calcFactorial(lengthOfSequence) /
calcFactorial(lengthOfSequence - (lengthOfSequence));
	std::cout << "P(n,r) = " << PERMUTATIONS_LEN << "\r\n\r\n";
	std::sort(permutationVals, permutationVals + lengthOfSequence);
	std::vector<uint32_t> permutations(PERMUTATIONS_LEN);
	//uint32_t *permutations = new uint32_t[PERMUTATIONS_LEN];
	uint32_t count = 0;
	std::cout << "\r\n -------------------- \r\n      Permutations\r\n ------------
--------" << std::endl;
	do {

		std::cout << permutationVals << std::endl;
		permutations[count] = atoi(permutationVals);
		count++;

	} while (std::next_permutation(permutationVals, permutationVals +
lengthOfSequence));

	do {
		badInput = false;
		std::cout << "Enter the distance from the centre to any marker -> ";
		std::cin >> input;
		INPUT_LEN = input.length();
		for (size_t i = 0; i < INPUT_LEN; i++) {
```

```
                if (!isdigit(input[i])) {
                    std::cout << "Value entered was not a number. Try again."
<< std::endl;
                    badInput = true;
                    break;
                }
            }
        } while (badInput);

        std::cout << "------------------------" << std::endl;
        uint8_t edgeLength = stoi(input);
        std::set<double> uniqueDistances;
        //const uint32_t PERMUTATIONS_LEN = sizeof(permutations) /
sizeof(permutations[0]);
        const double SHORT_DIAGONAL = sqrt(3) * edgeLength;
        const int8_t LONG_DIAGONAL = 2 * edgeLength;

        printf("Edge length = %d\r\nShort Diagonal = %.2f\r\nLong Diagonal = %d\r\n----
--------------------\r\n", edgeLength, SHORT_DIAGONAL, LONG_DIAGONAL);

        /* ---------- Calculate distances ---------- */
        std::vector<double> distances(PERMUTATIONS_LEN);
        std::cout << "\r\n -------------------- \r\n      Distances\r\n ---------------
-----" << std::endl;

        for (uint32_t i = 0; i < PERMUTATIONS_LEN; i++) {
            double distance = edgeLength;
            uint32_t number = permutations[i];
            uint8_t one = (number / 100000U) % 10;
            uint8_t two = (number / 10000U) % 10;
            uint8_t three = (number / 1000U) % 10;
            uint8_t four = (number / 100U) % 10;
            uint8_t five = (number / 10U) % 10;
            uint8_t six = (number / 1U) % 10;
            //    printf("%d %d %d %d %d %d\r\n", one, two, three, four, five, six);

            // Check there are not two successive adjacent runs, if there are remove
this sequence permutation
            uint8_t adjacentRun = 0;
            int8_t currentRun;
            bool oneTwoWasAdjacent = false;
            bool twoThreeWasAdjacent = false;
            bool threeFourWasAdjacent = false;
            bool fourFiveWasAdjacent = false;
            bool removeSequence = false;
            for (int j = 0; j < 3; j++) {
                if (j == 0) {
                    currentRun = two - one;
                }
                else if (j == 1) {
                    currentRun = three - two;
                }
                else if (j == 2) {
                    currentRun = four - three;
                }
                else if (j == 3) {
                    currentRun = five - four;
                }
                if (currentRun == 1 || currentRun == -1 || currentRun == 5 ||
currentRun == -5) {
                    if (j == 0) {
                        oneTwoWasAdjacent = true;
```

```
				}
				else if (j == 1) {
					twoThreeWasAdjacent = true;
				}
				else if (j == 2) {
					threeFourWasAdjacent = true;
				}
				else if (j == 3) {
					fourFiveWasAdjacent = true;
				}
			}
		}
		if (oneTwoWasAdjacent && twoThreeWasAdjacent) {
			removeSequence = true;
		}
		else if (twoThreeWasAdjacent && threeFourWasAdjacent) {
			removeSequence = true;
		}
		else if (threeFourWasAdjacent && fourFiveWasAdjacent) {
			removeSequence = true;
		}
		if (!removeSequence) {
			char oneC[2];
			char twoC[2];
			char threeC[2];
			char fourC[2];
			char fiveC[2];
			char sixC[2];
			_itoa_s(one, oneC, 10);
			_itoa_s(two, twoC, 10);
			_itoa_s(three, threeC, 10);
			_itoa_s(four, fourC, 10);
			_itoa_s(five, fiveC, 10);
			_itoa_s(six, sixC, 10);

			for (int j = 0; j < 5; j++) {
				char tempDistanceC[3];
				if (j == 0) {
					strcpy_s(tempDistanceC, oneC);
					strcat_s(tempDistanceC, twoC);
				}
				else if (j == 1) {
					strcpy_s(tempDistanceC, twoC);
					strcat_s(tempDistanceC, threeC);
				}
				else if (j == 2) {
					strcpy_s(tempDistanceC, threeC);
					strcat_s(tempDistanceC, fourC);
				}
				else if (j == 3) {
					strcpy_s(tempDistanceC, fourC);
					strcat_s(tempDistanceC, fiveC);
				}
				else if (j == 4) {
					strcpy_s(tempDistanceC, fiveC);
					strcat_s(tempDistanceC, sixC);
				}
				uint8_t tempDistance = atoi(tempDistanceC);
				switch (tempDistance) {
				case (12): distance += edgeLength; break;
				case (13): distance += SHORT_DIAGONAL; break;
				case (14): distance += LONG_DIAGONAL; break;
```

```
                        case (15): distance += SHORT_DIAGONAL; break;
                        case (16): distance += edgeLength; break;

                        case (21): distance += edgeLength; break;
                        case (23): distance += edgeLength; break;
                        case (24): distance += SHORT_DIAGONAL; break;
                        case (25): distance += LONG_DIAGONAL; break;
                        case (26): distance += SHORT_DIAGONAL; break;

                        case (31): distance += SHORT_DIAGONAL; break;
                        case (32): distance += edgeLength; break;
                        case (34): distance += edgeLength; break;
                        case (35): distance += SHORT_DIAGONAL; break;
                        case (36): distance += LONG_DIAGONAL; break;

                        case (41): distance += LONG_DIAGONAL; break;
                        case (42): distance += SHORT_DIAGONAL; break;
                        case (43): distance += edgeLength; break;
                        case (45): distance += edgeLength; break;
                        case (46): distance += SHORT_DIAGONAL; break;

                        case (51): distance += SHORT_DIAGONAL; break;
                        case (52): distance += LONG_DIAGONAL; break;
                        case (53): distance += SHORT_DIAGONAL; break;
                        case (54): distance += edgeLength; break;
                        case (56): distance += edgeLength; break;

                        case (61): distance += edgeLength; break;
                        case (62): distance += SHORT_DIAGONAL; break;
                        case (63): distance += LONG_DIAGONAL; break;
                        case (64): distance += SHORT_DIAGONAL; break;
                        case (65): distance += edgeLength; break;
                        }
                }
                std::cout << distance << std::endl;
                double roundedDistance = round(distance * 10000) / 10000; //
Round value to 4 decimal points for unique adding
                distances[i] = roundedDistance;
                for (uint32_t j = 0; j < i; j++)
                        uniqueDistances.insert(roundedDistance);
        }
        else {
                distances.erase(distances.begin() + i);
                permutations.erase(permutations.begin() + i);
                PERMUTATIONS_LEN--;
        }
}

/* ---------- Print the counts for corresponding unique distances ---------- */
const size_t UNIQUE_VALS = uniqueDistances.size();
std::vector <uint32_t> counts(UNIQUE_VALS, 0); // Initialise a new counting
vector all with a value of zero
int index = 0;
std::cout << "\r\n -------------------- \r\n  Unique value counts\r\n ---------
-----------" << std::endl;
for (std::set<double>::iterator i = uniqueDistances.begin(); i !=
uniqueDistances.end(); i++) {
        for (uint32_t j = 0; j < PERMUTATIONS_LEN; j++) {
                if (distances[j] == *i) {
                        counts[index]++;
                }
        }
```

```
			std::cout << std::left << std::setw(3) << index + 1 << "| " <<
std::setw(9) << *i
				<< std::right << " = " << counts[index] << std::endl;
			index++;
		}
		/* ---------- Print all possible sequences for the user input ---------- */
		do {
			badInput = false;
			std::cout << "--------------------------------------------\r\n"
				<< "Enter the index value (1 - " << UNIQUE_VALS << ") to
determine all possible corresponding sequences -> ";
			std::cin >> input;
			INPUT_LEN = input.length();
			for (size_t i = 0; i < INPUT_LEN; i++) {
				if (!isdigit(input[i])) {
					std::cout << "Value entered was not a number. Try again."
<< std::endl;
					badInput = true;
					break;
				}
			}
			if (!badInput) {
				int64_t value = stoi(input);
				if (value < 1 || value > UNIQUE_VALS) {
					std::cout << "Value entered was not within the index range
specified. Try again." << std::endl;
					badInput = true;
				}
			}
		} while (badInput);

		std::cout << "--------------------------------------------\r\n" << std::endl;
		uint8_t val = stoi(input);
		double distanceVal = *std::next(uniqueDistances.begin(), val - 1); // Find the
corresponding unique distance for the index chosen

		std::cout << " ------------------------------------ \r\n  All possible
sequences for " << distanceVal
			<< "\r\n ------------------------------------" << std::endl;
		if (printCodeFriendlySequences)
			std::cout << "const uint32_t permutations[] PROGMEM {\r\n  ";
		uint8_t numPrinted = 0;
		for (uint32_t i = 0; i < PERMUTATIONS_LEN; i++) {
			if (distances[i] == distanceVal) {
				numPrinted++;
				if (printCodeFriendlySequences) {
					if (numPrinted != counts[val - 1]) {
						std::cout << permutations[i] << ", ";
					}
					else {
						std::cout << permutations[i] << std::endl;
					}
					if (numPrinted % 15 == 0) {		// Carriage Return &
Line Feed (CR+LF) every 15 sequences,
						std::cout << "\r\n  ";			// so it does not
trail along one line
					}
				}
				else {
					std::cout << permutations[i] << std::endl;
				}
			}
```

```
    }
    if (printCodeFriendlySequences)
        std::cout << "};\r\n";
    std::cout << "-----------------------------------" << std::endl;
}
```

J.3. Master Program

```
/*
   Master component of the Sports Agility Tester.

   Copyright (C)2018 Reuben Smith

*/

/*----------------------------------------------------*/
/*               #include Header Files                */
/*----------------------------------------------------*/
#include <Arduino.h>
#include <stdint.h>
/* --- Include .h file needed for simple timekeeping --- */
#include <StopWatch.h>
/* --- Include .h files needed for LiDAR --- */
#include <Wire.h>
#include <LIDARLite_v3HP.h>
/* --- Include .h file needed for Stepper & Driver --- */
#include "DRV8834.h"
/* --- Include .h files needed for NRF24L01 --- */
#include <SPI.h>
#include "RF24.h"
#include "nRF24L01.h"
#include "printf.h" // For debugging
/* --- Include .h files needed to generate true random values --- */
#include <Entropy.h>
#include <math.h>
#include <stdlib.h>

/*----------------------------------------------------*/
/*          Set-up Arduino Digital Pins               */
/*----------------------------------------------------*/
#define photoInterruptPin 2    // Photointerrupter
#define buttonPin         3    // Button module
#define blueLedPin        5
#define redLedPin         6    // RGB module. Note: Breakout has mixed up R
& G output (likely because it has GRB colour order)
#define greenLedPin       7
#define piezoPin          13   // Piezo buzzer
#define nrf24irqPin       19   // nRF24L01 interrupt pin
#define laserPin          24   // Low power (<1mW) laser diode module
#define cePIN             49   // Chip-Enable (nRF24L01)
#define csnPin            53   // Chip-Select-Not (nRF24L01)

/*----------------------------------------------------*/
/*          Set-up Stepper Motor & Driver             */
/*----------------------------------------------------*/
#define MOTOR_STEPS 200 // 200 steps / 360 = 1.8 degrees
#define RPM 120
#define CALIBRATION_RPM 60
#define MICROSTEPS 32 // 1.8 / 32 = 0.05625 degrees

// Note: DRV8834 has micrstepping modes: 1, 1/4, 1/8, 1/16, 1/32
```

```
#define DIR 8
#define STEP 9
#define M0 10
#define M1 11
#define ENABLE 12

volatile DRV8834 stepper(MOTOR_STEPS, DIR, STEP, ENABLE, M0, M1);  // All
variables handled in ISRs need to be defined as volatile

volatile double angle = 0;

/*-----------------------------------------------------*/
/*               Set-up LiDAR Lite v3HP                */
/*-----------------------------------------------------*/
LIDARLite_v3HP lidarLite;
#define FAST_I2C
const uint8_t RUNNING_AVG_NUM = 300;
uint16_t lidarDistances[RUNNING_AVG_NUM];
uint8_t runningAvgIndex = 0;
uint32_t runningAvgTotal = 0;
uint16_t runningAverage = 0;

/*---------------------------------------------------------------------------
*/
/*  Set-up nRF24L01 radio & pipe addresses for master and slave
devices   */
/*---------------------------------------------------------------------------
*/
RF24 radio(cePIN, csnPin);
const uint64_t pipes[7] = { 0x3e64b727ffLL, 0x714e2b65c1LL, 0x714e2b65c2LL,
0x714e2b65c3LL, 0x714e2b65c4LL, 0x714e2b65c5LL, 0x714e2b65c6LL };
const char leastSignificantByte[][3] = { "ff", "c1", "c2", "c3", "c4",
"c5", "c6" };

/*-----------------------------------------------------*/
/*          Set-up Bluetooth Low Energy HM-10          */
/*-----------------------------------------------------*/
HardwareSerial & bluetooth = Serial2; // Use pointer for code reading
simplicity
bool btCommandReceived = false;
char btCommand[20];
const uint8_t BT_MAX = sizeof(btCommand) / sizeof(btCommand[0]);
/*-----------------------------------------------------*/
/*          Declare Test Configuration Values          */
/*-----------------------------------------------------*/
#define MAX_TESTS 800
#define MARKER_NUMBER 2
//#define MARKER_DISTANCE 70 // Distance markers are from the centre (in
cm).
#define ACCURACY_TOLERANCE 20 // Tolerance of LiDAR measurement accuracy
(in cm).
#define POSITION_VALIDATION_TIME 5000
#define MAX_COUNTDOWN 10
#define DIFFICULTY_NUM 3
#define DIFFICULTY1 1
#define DIFFICULTY2 2
#define DIFFICULTY3 3
#define D1_DISTANCE 100 // Distance markers are from the centre (in cm).
#define D2_DISTANCE 200
#define D3_DISTANCE 1000
```

```
#define UPPER_CHARACTER_SET 0
#define LOWER_CHARACTER_SET 1
#define DIGIT_CHARACTER_SET 2
#define CHARACTER_SET_NUM 3

uint8_t testDifficulty = 1;
/*----------------------------------------------------*/
/*          Set-up Time Keeping Variables             */
/*----------------------------------------------------*/
StopWatch time(StopWatch::MILLIS);
volatile StopWatch buttonTime(StopWatch::MILLIS);
StopWatch nrf24Time(StopWatch::MICROS);
StopWatch testTime(StopWatch::MICROS);

/*----------------------------------------------------*/
/*            Declare Other Global Variables          */
/*----------------------------------------------------*/
volatile uint8_t calibrationCount = 0;
uint8_t currentLEDColour[] = {0, 0, 0};
uint8_t prevLEDColour[] = {0, 0, 0};

/*----------------------------------------------------*/
/*          Declare Button Interrupt Flags            */
/*----------------------------------------------------*/
#define LONG_PRESS 2000
#define DEBOUNCE_DELAY 50
volatile bool buttonISRHigh = false;
volatile bool buttonISRLow = false;

/*----------------------------------------------------*/
/*                Declare Other Flags                 */
/*----------------------------------------------------*/
bool reset = false;
volatile bool calibrate = true;
bool readyMarker1 = false;
bool readyMarker2 = false;
bool readyMarker3 = false;
bool readyMarker4 = false;
bool readyMarker5 = false;
bool readyMarker6 = false;

bool rxNextTestMsg = false;
bool runNextTest = false;

/*----------------------------------------------------*/
/*        Declare Possible Sequence Permutations      */
/*----------------------------------------------------*/
#define REPEAT_ACCEPTABLE 50 // Minimum number of test runs before a
repeated sequence can occur
const uint32_t permutations[] PROGMEM {
  124536, 125346, 126354, 132546, 132564, 134256, 134652, 136542, 142356,
143562, 145326, 146532, 152346, 154236, 154632,
  156324, 156342, 162534, 163542, 164352, 213645, 214653, 215463, 231465,
235641, 236451, 241653, 243615, 243651, 245163,
  245361, 251643, 253461, 254613, 256431, 261435, 261453, 263451, 265143,
265341, 312546, 312564, 314562, 316254, 316452,
  324156, 325164, 326514, 341562, 342516, 346152, 352164, 354126, 354162,
356214, 356412, 361542, 362154, 364512, 365124,
  412653, 413265, 415623, 416235, 421365, 421563, 423615, 423651, 425613,
431625, 435261, 436215, 451263, 452613, 453621,
```

```
  461325, 461523, 463215, 465213, 465231, 512436, 512634, 514326, 516324,
516342, 521346, 523164, 524316, 526134, 532416,
  532614, 534126, 534162, 536124, 541326, 542136, 546312, 562314, 563124,
564132, 613425, 614235, 615243, 621435, 621453,
  623145, 623541, 625431, 631245, 632451, 634215, 635421, 641235, 643125,
643521, 645213, 645231, 651423, 652431, 653241

  };

const uint8_t PERMUTATIONS_LEN = sizeof(permutations) /
sizeof(permutations[0]);

/*-----------------------------------------------------*/
/*    Store Test Sequences, Character Sets & Times    */
/*-----------------------------------------------------*/
#define MAX_SET_REPEATS 3 // Maximum number of test runs a repeated
chracter set can occur in a row
uint8_t pastPermutations[MAX_TESTS];
uint32_t pastTestTimes[MAX_TESTS];
uint8_t prevCharacterSet;
uint8_t repeatedCharacterSets = 0;
uint16_t testNum = 0;

/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/*                      Set-up                         */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
void setup() {
  /* ---------- Set-up PC COM Serial & Bluetooth ---------- */
  Serial.begin(115200); // For debugging
  bluetooth.begin(115200);
  printf_begin();
  Serial.print(F("/*-----------------------------------------------------
*/\r\n"
                 "     Master component of the Sports Agility
Tester.\r\n\r\n"
                 "     Copyright (C)2018 Reuben Smith\r\n/"
                 "*-----------------------------------------------------
*/\r\n\r\n"));
  // Blink the LED module green three times to indicate power ON
  for (int i = 0; i < 2; i++) {
    modifyLed(0, 100, 0);
    delay(200);
    modifyLed(0, 0, 0);
    delay(200);
  }

  /* ---------- Define digital output for pins of corresponding hardware --
-------- */
  pinMode(laserPin, OUTPUT);
  //pinMode(ssPin, OUTPUT); // Mega SS pin output so nRF24L01 is able to
send & receive (Required for Arduino Mega)
  pinMode(photoInterruptPin, OUTPUT);
  pinMode(buttonPin, OUTPUT);
  /* ---------- Enable pull-up resistor ready for interrupts ---------- */
  pinMode(photoInterruptPin, INPUT_PULLUP);
  pinMode(buttonPin, INPUT_PULLUP);
  //  pinMode(nrf24irqPin, INPUT_PULLUP);
  /* ---------- Attach interrupts for button, photointerrupter and nRF24L01
IRQ pin ---------- */
  attachInterrupt(digitalPinToInterrupt(photoInterruptPin),
photoInterruptISR, FALLING);
```

```
  attachInterrupt(digitalPinToInterrupt(buttonPin), buttonISR, CHANGE);
  //  attachInterrupt(digitalPinToInterrupt(nrf24irqPin), nrf24ISR,
FALLING);

  /* ---------- Initialize Arduino I2C (for communication to LidarLite) ---
------- */
  Wire.begin();
#ifdef FAST_I2C
  TWBR = ((F_CPU / 400000UL) - 16) / 2; // Set I2C frequency to 400kHz
#endif

  lidarLite.configure(3); // 0 is default, 3 is  max  range

  /* ---------- Set-up NRF242L01 radio ---------- */
  radio.begin();

  // Set-up auto Ack Payloads
  radio.setAutoAck(true);                    // Ensure autoACK is enabled
  radio.enableAckPayload();               // Allow optional ack payloads
  radio.setPayloadSize(7);                // Send 2-byte payloads
  radio.setDataRate(RF24_1MBPS);

  //  radio.setDataRate(RF24_250KBPS); // Faster with better range
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(108);
  radio.setRetries(0, 15);
  radio.openWritingPipe(pipes[0]);
  for (int i = 1; i < MARKER_NUMBER; i++) { // Open reading pipes with
slave devices
    radio.openReadingPipe(i, pipes[i + 1]);
  }

  /* ---------- Set motor RPM and microstep value ---------- */
  stepper.begin(CALIBRATION_RPM, MICROSTEPS);
  stepper.disable(); // Stop motor from moving

  /* ---------- Set-up random number generator ---------- */
  Entropy.initialize();

  /* ---------- Indicate set-up is complete ---------- */
  tone(piezoPin, 700, 150); // [Pin#, Frequency (in Hz), Duration (in ms)]
  delay(250);
  tone(piezoPin, 700, 150);
  Serial.println(F("--------------- nRF24L01 Details ---------------"));
  radio.printDetails();
  Serial.println(F("-------------------------------------------------"));
  // Check the nRF24L01 is connected correctly
  if (radio.isChipConnected()) {
    Serial.println(F("nRF24L01 is Connected!"));
  } else {
    Serial.println(F("nRF24L01 is not Connected!"));
  }
  printf_P(PSTR("-------------------------------------------------\r\n"));
  printf_P(PSTR("Set-up complete.\r\n---------------------------------------
----------\r\n"));
}

/* #################################################### */
/*                      Main loop                       */
/* #################################################### */
void loop() {
```

```
  /* -------------------------------------------------------- */
  /*                Initialisation & Calibration              */
  /* -------------------------------------------------------- */

  /* ---------- Initalise variables ---------- */
  uint16_t distance;
  uint8_t  newDistance = 0;
  uint16_t markerDistance = 0;
  int16_t MARKER_ANGLE = (360 / MARKER_NUMBER);
  bool markerConnected = false;

  /* ---------- Reset booleans if difficulty change (button press), reset
or BT command ---------- */
  if (reset) {
    reset = !reset;
    Serial.println(F("RESET OCCURRED"));
    for (int i = 0; i < 3; i++) {
      tone(piezoPin, 550, 500);
      delay(600);
    }
    /* ---------- Reset the button time in case a debounce occurred just
after a long press ---------- */
    buttonTime.stop();
    buttonTime.reset();
    delay(2000);
  }
  modifyLed(100, 40, 0);
  if (btCommandReceived) {
    btCommandReceived = !btCommandReceived;
  }

  printf_P(PSTR("Marker angle now = %d\r\n"), MARKER_ANGLE);

  /* ---------- Alter test distance based on the difficulty ---------- */
  switch (testDifficulty) {
    case 1:
      markerDistance = D1_DISTANCE;
      break;
    case 2:
      markerDistance = D2_DISTANCE;
      break;
    case 3:
      markerDistance = D3_DISTANCE;
      break;
  }
  /* ---------- Calculate acceptable position range based on distance +
tolerance ---------- */
  uint16_t  LOWER_LIMIT = (markerDistance - ACCURACY_TOLERANCE);
  uint16_t UPPER_LIMIT = (markerDistance + ACCURACY_TOLERANCE);

  printf_P(PSTR("Marker distance now = %d\r\n"), markerDistance);

  /* ---------- Calibrate the stepper angle ---------- */
  calibrateAngle();

  /* -------------------------------------------------------- */
  /*                Connect To & Set-up Markers               */
  /* -------------------------------------------------------- */
  for (int i = 0; i < MARKER_NUMBER; i++) {
    if (reset) {
      break;
```

```
    }
    time.stop();
    time.reset();
    time.start();
    printf_P(PSTR("Waiting to connect to marker %d\r\n"), i + 1);
    modifyLed(100, 60, 5);
    while (!markerConnected) {
      /* ---------- Check if button long press, reset command or BT command
---------- */
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      char dataReceived[3];
      /* ---------- Send the pipe address the Marker needs to use ---------
- */
      radio.stopListening();
      radio.openWritingPipe(pipes[0]);
      radio.write( &(leastSignificantByte[i + 1]),
sizeof(leastSignificantByte[i + 1]));
      //      printf_P(PSTR("Sent %s\r\n"), leastSignificantByte[i + 1]);
      //                radio.startListening();
      byte pipeNum;
      radio.openReadingPipe(0, pipes[1]); // Pipe 0 is shared with the
writing pipe so reopen reading pipe with Slave 1
      radio.startListening();
      while (radio.available(&pipeNum) ) { // If data was received
        time.stop();
        radio.read( &dataReceived, sizeof(dataReceived) );
        /* ---------- Check if the message matches the slave address LSB --
-------- */
        if (strstr(dataReceived, leastSignificantByte[i + 1]) != nullptr) {
          printf_P(PSTR("[RX] Marker %d connected with address LSB %s,
took: %lu milliseconds to connect\n\r"),
                   i + 1, dataReceived, time.value());
          markerConnected = true;
        }
      }
      delay(50); // Try and send the pipe address every 50 milliseconds
    }
    markerConnected = false;
    bool settingUpMarker = true;
    time.reset();
    /* ---------- Rotate the LiDAR to the next Marker position ----------
*/
    if (i != 0) {
      //Serial.print("Steps required for rotation: ");
      //Serial.println(calcStepsForRotation(MARKER_ANGLE));
      angle += MARKER_ANGLE;
      stepper.enable();
      stepper.move(calcStepsForRotation(MARKER_ANGLE));    // forward
revolution
      stepper.disable();
    }

    printf_P(PSTR("Waiting for user to position marker number %d\r\n"), i +
1);
    modifyLed(100, 0, 0);
    uint8_t distanceCount = 0;
```

```
    // The following booleans ensure only one message is sent for each
status change
    bool withinLimits = false;
    bool outsideLimits = false;
    digitalWrite (laserPin, HIGH); // Turn Laser Diode Module On
    /* ------------------------------------------------------- */
    /*          Assist Marker Positioning with LiDAR           */
    /* ------------------------------------------------------- */
    while (settingUpMarker) {
      // The following booeans confirm if an auto Ack(knowledge) packet has
been received or not
      bool receivedCorrectAckMsg = false;
      bool receivedIncorrectAckMsg = false;

      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      /* ---------- Uncomment below to see continuous distance measurement
values ---------- */
      //            if (distanceCount <= RUNNING_AVG_NUM) {
      //              for (int i = 0; i < RUNNING_AVG_NUM; i++) {
      //                newDistance = distanceContinuous(&distance);
      //                runningAverage = calcRunningAverage(&newDistance,
&distance);
      //                if (newDistance) {
      //                  distanceCount++;
      //                }
      //              }
      //            }
      /* ---------- Calculate the running average ---------- */
      newDistance = distanceContinuous(&distance);
      runningAverage = calcRunningAverage(&newDistance, &distance);
      if (newDistance)
      {
        /* ---------- Uncomment below to print the running average distance
---------- */
//        printf_P(PSTR("Running average = %u\r\n"), runningAverage);
        //      printf_P(PSTR("time = %lu\r\n"), time.elapsed()); // Print
the time
        /* ---------- If the marker is positioned within the distance
limits ---------- */
        if (runningAverage >= LOWER_LIMIT && runningAverage <= UPPER_LIMIT)
{
          if (!withinLimits) {
            withinLimits = true;
            outsideLimits = false;
            if (time.state() == StopWatch::RESET || StopWatch::STOPPED) {
              time.start();
            }
            //            Serial.println(F("Marker in correct position...
wait 5 seconds to confirm..."));
            modifyLed(0, 100, 0);

            /* Send msg to Marker using nRF24L01 that it is in the correct
position */
            bool nothingReceived = true;
            if (!receivedCorrectAckMsg) {
              while (nothingReceived) {
                char correctPosition[3] = "CP"; // "CP" => Correct Position
```

```
                char dataReceived[3];
                nrf24Time.start();
                radio.stopListening();
                radio.openWritingPipe(pipes[0]);
                if (radio.write( &correctPosition, sizeof(correctPosition)
)) {
                  Serial.println(F("Sent CORRECT position!"));
                  //                  nrf24Time.start();
                  //                  radio.startListening();
                }
                radio.openReadingPipe(0, pipes[1]);
                radio.startListening();
                delay(25);
                while (radio.available() ) {
                  nrf24Time.stop();
                  radio.read( &dataReceived, sizeof(dataReceived) );
                  if (strstr(dataReceived, "CP") != nullptr) {
                    printf_P(PSTR("[RX] Got response %s, round-trip delay:
%lu microseconds\r\n"),
                             dataReceived, nrf24Time.value());
                    nothingReceived = false;
                    receivedCorrectAckMsg = true;
                    receivedIncorrectAckMsg = false;
                  }
                  break;
                }
                if (nrf24Time.elapsed() > 1000000 ) { // If waited longer
than 100ms after sending, stop trying to send and exit the loop
                  break;
                }
                //                  delay(30);
              }
            }
          }
          /* ---------- Marker positioned in correct position for longer
than validation time ---------- */
        } else { /* ---------- Marker not in the correct position ---------
- */
          if (!outsideLimits) {  // Send message only if a new status
            withinLimits = false;
            outsideLimits = true;
            //              Serial.println(F("Marker not in correct
position..."));
            modifyLed(100, 0, 0);

            /* Send msg to Marker using nRF24L01 that it is in not in the
correct position */
            bool nothingReceived = true;
            if (!receivedIncorrectAckMsg) { // Keep looping until the
uncorrect acknowledge packet received
              while (nothingReceived) {
                char incorrectPosition[3] = "IP"; // "IP" => Incorrect
Position
                char dataReceived[3];
                nrf24Time.start();
                radio.stopListening();
                radio.openWritingPipe(pipes[0]);
                if (radio.write( &incorrectPosition,
sizeof(incorrectPosition) )) {
                  Serial.println(F("Sent INCORRECT position!"));
                }
```

```
                radio.openReadingPipe(0, pipes[1]);
                radio.startListening();
                delay(25);
                while (radio.available() ) {
                  nrf24Time.stop();
                  radio.read( &dataReceived, sizeof(dataReceived) );
                  if (strstr(dataReceived, "IP") != nullptr) {
                    printf_P(PSTR("[RX] Got response %s, round-trip delay:
%lu microseconds\r\n"),
                             dataReceived, nrf24Time.value());
                    nothingReceived = false;
                    receivedCorrectAckMsg = false;
                    receivedIncorrectAckMsg = true;
                  }
                  break;
                }
                if (nrf24Time.elapsed() > 1000000 ) { // If waited longer
than 100ms, stop trying to send
                  break;
                }
                //                delay(30);
              }
            }
            nrf24Time.stop();
            nrf24Time.reset();
            time.stop();
            time.reset();
          }
        }
        if (time.state() == StopWatch::RUNNING && time.elapsed() >=
POSITION_VALIDATION_TIME) {
          // Marker in correct position for over 5 seconds
          //          Serial.println(F("Marker confirmed in correct
position! Moving to next marker position..."));
          modifyLed(30, 100, 40);

          /* Send msg to Marker using nRF24L01 that it is in the correct
position */
          bool nothingReceived = true;
          if (!receivedCorrectAckMsg) {
            while (nothingReceived) {
              char positionValidated[3] = "PV"; // "PV" => Position
Validated
              char dataReceived[3];
              nrf24Time.start();
              radio.stopListening();
              radio.openWritingPipe(pipes[0]);
              if (radio.write( &positionValidated,
sizeof(positionValidated) )) {
                Serial.println(F("Sent POSITION VALIDATED!"));
                //                nrf24Time.start();
                //                radio.startListening();
              }
              radio.openReadingPipe(0, pipes[1]);
              radio.startListening();
              delay(25);
              while (radio.available() ) {
                nrf24Time.stop();
                radio.read( &dataReceived, sizeof(dataReceived) );
                if (strstr(dataReceived, "PV") != nullptr) {
```

```
                  printf_P(PSTR("[RX] Got response %s, round-trip delay:
%lu microseconds\n\r"),
                           dataReceived, nrf24Time.value());
                  nothingReceived = false;
                  settingUpMarker = false;
                }
                break;
              }
              if (nrf24Time.elapsed() > 1000000 ) { // If waited longer
than 100ms after sending, stop trying to send and exit the loop
                break;
              }
              //              delay(30);
            }
          }
          // Send to marker to turn off LED display.

        }
      }
    } // while (settingUpMarker)
    digitalWrite (laserPin, LOW); // Turn Laser Off
    time.stop();
    time.reset();
    nrf24Time.stop();
    nrf24Time.reset();
    //settingUpMarker = true;
  }

  /* ******************** ALL MARKERS NOW CONNECTED ********************
*/

  /* ------------------------------------------------------- */
  /*                  Test Set-up & Run Loop                 */
  /* ------------------------------------------------------- */
  bool runTests = true;
  while (runTests) {
    checkButtonFlags();
    if (reset) {
      break;
    }
    bool noRepeat;
    printf("------------------ \r\nTEST NUMBER %d\r\n", testNum + 1);

    /* ------------------------------------------------------- */
    /*             Determine Test Marker Sequence              */
    /* ------------------------------------------------------- */
    do {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      noRepeat = true;
      pastPermutations[testNum] = Entropy.random(0, PERMUTATIONS_LEN);
      printf_P(PSTR("[Entropy] Generated test sequence index %u w/ sequence
%lu\r\n"),
               pastPermutations[testNum], pgm_read_dword_near(permutations
+ pastPermutations[testNum]));
      for (int i = (testNum - 1); i > (testNum - 1) - REPEAT_ACCEPTABLE; i-
-) {
```

```
        if (i < 0) { // Break if i is outside of the index range (when
testNum < REPEAT_ACCEPTABLE)
          break;
        }
        if (pastPermutations[testNum] == pastPermutations[i]) {
          printf_P(PSTR("[WARNING] Sequence matched one performed in test
%d (%lu). Generating a new sequence.\r\n"),
                   i, pgm_read_dword_near(permutations +
pastPermutations[testNum]));
          noRepeat = false;
          break;
        }
      }
    } while (!noRepeat);

    /* ------------------------------------------------------- */
    /*         Store Marker Sequences into char Array          */
    /* ------------------------------------------------------- */
    char markerSequence[7];
    uint32_t number = pgm_read_dword_near(permutations +
pastPermutations[testNum]); // Retrieve sequence from PROGMEM
    //    printf_P(PSTR("Sequence as a uint32_t NUMBER: %lu\r\n"), number);
    sprintf_P(markerSequence, PSTR("%lu"), number); // Convert number to
char array sequence
    //    printf_P(PSTR("Sequence as a char ARRAY: %s\r\n"),
markerSequence);

    /* ------------------------------------------------------- */
    /*  Break Down Marker Sequence Into Individual Components  */
    /* ------------------------------------------------------- */
    uint8_t one = (number / 100000U) % 10;
    uint8_t two = (number / 10000U) % 10;
    uint8_t three = (number / 1000U) % 10;
    uint8_t four = (number / 100U) % 10;
    uint8_t five = (number / 10U) % 10;
    uint8_t six = (number / 1U) % 10;

    //    printf_P(PSTR("Individual component check: %u %u %u %u %u
%u\r\n"),
    //                                     one, two, three, four, five,
six);

    /* ------------------------------------------------------- */
    /*              Send Markers the Test Sequence             */
    /* ------------------------------------------------------- */
    printf_P(PSTR("[TX] Sending [MARKER SEQ] %s to Markers\n\r"),
markerSequence);
    bool proceedToNextStep = false;
    while (!proceedToNextStep) {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      nrf24Time.start();
      radio.stopListening();
      radio.openWritingPipe(pipes[0]);
      radio.write(&markerSequence, sizeof(markerSequence)); // Send the
sequence to all the markers
      radio.openReadingPipe(0, pipes[1]);
      radio.startListening();
```

```
      delay(30); // Delay for time to receive the message back
      char dataReceived[7];
      uint8_t pipeNum;
      while (radio.available(&pipeNum)) {
        nrf24Time.stop();
        radio.read( &dataReceived, sizeof(dataReceived) );
        printf_P(PSTR("[RX] Got [MARKER SEQ] %s on PIPE %u, round-trip
delay: %lu microseconds\n\r"),
                 dataReceived, pipeNum, nrf24Time.value());
        if (strstr(dataReceived, markerSequence) != nullptr) {
        switch (pipeNum)  {
          case (0) : readyMarker1 = true; break;
          case (1) : readyMarker2 = true; break;
          case (2) : readyMarker3 = true; break;
          case (3) : readyMarker4 = true; break;
          case (4) : readyMarker5 = true; break;
          case (5) : readyMarker6 = true; break;
        }
        }
      }
      nrf24Time.reset();
      if (readyMarker1 && readyMarker2) { // FOR DEBUGGING
//      if (readyMarker1 && readyMarker2 && readyMarker3 && readyMarker4 &&
readyMarker5 && readyMarker6) {
        proceedToNextStep = true;
      }
    }
    resetMarkerReadyStatus();

    /* ------------------------------------------------------- */
    /*              Determine Test Character Set              */
    /* ------------------------------------------------------- */
    char characterSequence[7];
    uint8_t characterSequenceInts[6];
//    uint8_t characterSequenceInts[MARKER_NUMBER];
    uint8_t numberOfPossibilities;
    uint8_t lowerASCII, upperASCII;
    bool repeatMaxReached;
    uint8_t characterSet;
    do {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      repeatMaxReached = false;
      characterSet = Entropy.random(0, CHARACTER_SET_NUM); // Randomly
choose a character set (Upper, Lower or Digits)
      switch (characterSet) {
        case (UPPER_CHARACTER_SET) :
          printf_P(PSTR("[Entropy] Generated character set:
UPPERCASE\r\n"));
          // ASCII decimal values 65 - 90 represent char uppercase letters
'A' - 'Z'
          lowerASCII = 'A';
          upperASCII = 'Z';
          break;
        case (LOWER_CHARACTER_SET) :
          printf_P(PSTR("[Entropy] Generated character set:
LOWERCASE\r\n"));
```

```
          // ASCII decimal values 97 - 122 represent char lowercase letters
'a' - 'z'
          lowerASCII = 'a';
          upperASCII = 'z';
          break;
        case (DIGIT_CHARACTER_SET) :
          printf_P(PSTR("[Entropy] Generated character set: DIGITS\r\n"));
          // ASCII decimal values 48 - 57 represent char uppercase letters
'0' - '9'
          lowerASCII = '0';
          upperASCII = '9';
          break;
      }

      /* ---------- Check the same character set has not been conducted
MAX_SET_REPEATS in a row ---------- */
      if (testNum != 0 && characterSet == prevCharacterSet) {
        repeatedCharacterSets++;
        if (repeatedCharacterSets == MAX_SET_REPEATS) {
          printf_P(PSTR("[WARNING] Character set repeated too many times!
Generating a new set...\r\n"));
          repeatMaxReached = true; // Force a new character set generation
for this run
          repeatedCharacterSets--;
        }
      }
    } while (repeatMaxReached);
    if (characterSet != prevCharacterSet) {
      repeatedCharacterSets = 0;
    }
    prevCharacterSet = characterSet; // Store the current test character
set for next time

    /* -------------------------------------------------------- */
    /*          Determine Test Character Sequence               */
    /* -------------------------------------------------------- */
    for (int k = 0; k < 6; k++) { // FOR DEBUGGING
//    for (int k = 0; k < MARKER_NUMBER; k++) {
      characterSequenceInts[k] = Entropy.random(lowerASCII, upperASCII +
1); // Generate ASCII decimal value within limits
      for (int i = (k - 1); i > -1 ; i--) {
        if (i < 0) { // Break if i is outside of the index range
          break;
        }
        if (characterSequenceInts[k] == characterSequenceInts[i]) {
          k--;
          break;
        }
      }
    }

    printf_P(PSTR("[Entropy] Generated character set: "));
    for (int i = 0; i < MARKER_NUMBER; i++) {
      printf_P(PSTR("%u "), characterSequenceInts[i]);
    }

    /* ---------- Concatenate the corresponding chars of the ASCII decimal
values into a char array ---------- */
    sprintf(characterSequence, "%c%c%c%c%c%c",
(char)characterSequenceInts[0], (char)characterSequenceInts[1],
            (char)characterSequenceInts[2], (char)characterSequenceInts[3],
```

```
            (char)characterSequenceInts[4],
(char)characterSequenceInts[5]);

    /* ------------------------------------------------------- */
    /*             Print Marker & Character Sequence          */
    /* ------------------------------------------------------- */
    printf_P(PSTR("\r\n\r\n[MARKER SEQUENCE] Test sequence index %u
corresponds to sequence: %lu\r\n"),
             pastPermutations[testNum], pgm_read_dword_near(permutations +
pastPermutations[testNum]));
    printf_P(PSTR("[CHARACTER SEQUENCE] Character sequence: %s\r\n"),
characterSequence);

    /* --------------------------------------------------------- */
    /*  Inform Markers to be Ready to Receive Character Sequence */
    /* --------------------------------------------------------- */
    proceedToNextStep = false;
    while (!proceedToNextStep) {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      char nextStep[3] = "NS";
      nrf24Time.start();
      radio.stopListening();
      radio.openWritingPipe(pipes[0]);
      radio.write(&nextStep, sizeof(nextStep)); // Send the sequence to all
the markers
      radio.openReadingPipe(0, pipes[1]);
      radio.startListening();
      delay(30); // Delay for time to receive the message back
      char dataReceived[3];
      uint8_t pipeNum;
      while (radio.available(&pipeNum)) {
        nrf24Time.stop();
        radio.read( &dataReceived, sizeof(dataReceived) );
        printf_P(PSTR("[RX] Got %s on PIPE %u, round-trip delay: %lu
microseconds\n\r"),
                 dataReceived, pipeNum, nrf24Time.value());
        if (strstr(dataReceived, nextStep) != nullptr) {
        switch (pipeNum)  {
          case (0) : readyMarker1 = true; break;
          case (1) : readyMarker2 = true; break;
          case (2) : readyMarker3 = true; break;
          case (3) : readyMarker4 = true; break;
          case (4) : readyMarker5 = true; break;
          case (5) : readyMarker6 = true; break;
        }
        }
        break;
      }
      nrf24Time.reset();
      if (readyMarker1 && readyMarker2) { // FOR DEBUGGING
//      if (readyMarker1 && readyMarker2 && readyMarker3 && readyMarker4 &&
readyMarker5 && readyMarker6) {
        proceedToNextStep = true;
      }
    }
    resetMarkerReadyStatus();
```

```
    /* ------------------------------------------------------- */
    /*           Send Markers the Character Sequence          */
    /* ------------------------------------------------------- */
    printf_P(PSTR("[TX] Sending [CHARACTER SEQ] %s to Markers\n\r"),
characterSequence);
    proceedToNextStep = false;
    while (!proceedToNextStep) {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      nrf24Time.start();
      radio.stopListening();
      radio.openWritingPipe(pipes[0]);
      radio.write(&characterSequence, sizeof(characterSequence)); // Send
the character sequence to all the markers
      radio.openReadingPipe(0, pipes[1]);
      radio.startListening();
      delay(30); // Delay for time to receive the message back
      char dataReceived[7];
      uint8_t pipeNum;
      while (radio.available(&pipeNum)) {
        nrf24Time.stop();
        radio.read( &dataReceived, sizeof(dataReceived) );
        printf_P(PSTR("[RX] Got [CHARACTER SEQ] %s on PIPE %u, round-trip
delay: %lu microseconds\n\r"),
                 dataReceived, pipeNum, nrf24Time.value());
        switch (pipeNum)  {
          case (0) : readyMarker1 = true; break;
          case (1) : readyMarker2 = true; break;
          case (2) : readyMarker3 = true; break;
          case (3) : readyMarker4 = true; break;
          case (4) : readyMarker5 = true; break;
          case (5) : readyMarker6 = true; break;
        }
      }
      nrf24Time.reset();
      if (readyMarker1 && readyMarker2) { // FOR DEBUGGING
//      if (readyMarker1 && readyMarker2 && readyMarker3 && readyMarker4 &&
readyMarker5 && readyMarker6) {
        proceedToNextStep = true;
      }
    }
    resetMarkerReadyStatus();

    /* ------------------------------------------------------- */
    /*            Wait for All Markers to be Ready            */
    /* ------------------------------------------------------- */
    bool allMarkersReady = false;
    while (!allMarkersReady) {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      char dataReceived[3];
      uint8_t pipeNum;
      while (radio.available(&pipeNum) ) {
        nrf24Time.stop();
        radio.read( &dataReceived, sizeof(dataReceived) );
```

```
        radio.writeAckPayload(pipeNum, &dataReceived, sizeof(dataReceived)
);
        printf_P(PSTR("[RX] Got msg %s on PIPE %u.\n\r"), dataReceived,
pipeNum);
        if (strstr(dataReceived, "MR") != nullptr) {
          switch (pipeNum)  {
            case (0) : readyMarker1 = true; break;
            case (1) : readyMarker2 = true; break;
            case (2) : readyMarker3 = true; break;
            case (3) : readyMarker4 = true; break;
            case (4) : readyMarker5 = true; break;
            case (5) : readyMarker6 = true; break;
          }
        }
      }
      nrf24Time.reset();
      if (readyMarker1 && readyMarker2 && readyMarker3 && readyMarker4 &&
readyMarker5 && readyMarker6) {
        allMarkersReady = true;
      }
    }
    resetMarkerReadyStatus();

    /* -------------------------------------------------------- */
    /*              Generate a Random Countdown                 */
    /* -------------------------------------------------------- */
    uint8_t countDown = Entropy.random(0, MAX_COUNTDOWN + 1); // Use
Entropy to pick a random countdown anywhere from 1-10 seconds
    printf_P(PSTR("[COUNTDOWN] Count down time: %u\r\n"), countDown);
    bool countingDown = true;
    time.start();
    while (countingDown) {
      if (time.elapsed() >= countDown * 1000) {
        countingDown = false;
      }
    }
    time.stop();
    time.reset();
    /* ******************** TEST RUNNING ********************* */

    /* -------------------------------------------------------- */
    /*        Indicate First Marker to Present Stimulus         */
    /* -------------------------------------------------------- */
    bool nothingReceived = true;
    while (nothingReceived) {
      char dataToSend[3] = "GO";
      char dataReceived[3];
      uint8_t pipeNum;
      nrf24Time.start();
      radio.stopListening();
      radio.openWritingPipe(pipes[0]);
      radio.write(&dataToSend, sizeof(dataToSend));
      radio.openReadingPipe(0, pipes[1]);
      radio.startListening();
      testTime.start();
      while (radio.available(&pipeNum)) {
        if (pipeNum == one - 1) {
          nrf24Time.stop();
          radio.read( &dataReceived, sizeof(dataReceived) );
          printf_P(PSTR("[RX] Got msg %s on PIPE %u. Round trip delay %lu.
\n\r"), dataReceived, pipeNum,  nrf24Time.value());
```

```
          nothingReceived = false;
        }
      }
    }

    /* ------------------------------------------------------- */
    /*       Listen for Recorded Test Times from Markers       */
    /* ------------------------------------------------------- */
    uint32_t totalTestTimes = 0;
    for (int i = 0; i < 6; i++) {
      uint32_t receivedTime;
      nothingReceived = true;
      uint8_t listenPipe;
      switch (i) {
        case (0): listenPipe = one - 1; break;
        case (1): listenPipe = two - 1; break;
        case (2): listenPipe = three - 1; break;
        case (3): listenPipe = four - 1; break;
        case (4): listenPipe = five - 1; break;
        case (5): listenPipe = six - 1; break;
      }
      while (nothingReceived) {
        uint8_t pipeNum;
        while (radio.available(&pipeNum)) {
          if (pipeNum == listenPipe) {
            nrf24Time.stop();
            radio.read( &receivedTime, sizeof(receivedTime) );
            printf_P(PSTR("[RX] Got time %lu on PIPE %u. Round trip delay
%lu. \n\r"), receivedTime, pipeNum,  nrf24Time.value());
            radio.stopListening();
            radio.openWritingPipe(pipes[0]);
            radio.write(&receivedTime, sizeof(receivedTime));
            radio.openReadingPipe(0, pipes[1]);
            radio.startListening();
            nothingReceived = false;
          }
        }
      }
      totalTestTimes += receivedTime;
    }
    testTime.stop();
    /* ******************** TEST COMPLETE ******************** */

    pastTestTimes[testNum] = testTime.value();
    testTime.reset();

    /* ------------------------------------------------------- */
    /*             Send Results to Phone/Computer              */
    /* ------------------------------------------------------- */
    bluetooth.print(F("========= RESULTS =========\r\nTEST "));
    bluetooth.print(testNum + 1);
    bluetooth.print(F("=========\r\nTime = "));
    bluetooth.print(pastTestTimes[testNum]);
    bluetooth.print(F("\r\nMarker Sequence = "));
    bluetooth.print(pastPermutations[testNum]);
    bluetooth.print(F("\r\n"));

    printf_P(PSTR("========= RESULTS =========\r\nTEST %u=========\r\nTime
= %lu\r\nMarker Sequence = %u\r\n"), testNum + 1, pastTestTimes[testNum],
pastPermutations[testNum]);
```

```
    printf_P(PSTR("Total from Marker test times = %lu\r\n"),
totalTestTimes);
    /* ------------------------------------------------------- */
    /*      Wait for NEXT Command Before Running Next Test     */
    /* ------------------------------------------------------- */
    rxNextTestMsg = true; //
    while (!runNextTest) {
      checkForBluetoothCommand();
      checkButtonFlags();
      if (reset) {
        break;
      }
      delay(25); // Block until NEXT command is received from user.
    }
    rxNextTestMsg = false;
    runNextTest = false;
    testNum++;
  } // while (runTests)
} //void loop()

/* #################################################### */
/*                      Functions                       */
/* #################################################### */

/* --------------------------------------------------------------------
   Set the RGB Led Module colour output.

   0 = off
   255 = maximum brightness
   ------------------------------------------------------------------- */
void modifyLed(uint8_t *red, uint8_t *green, uint8_t *blue) {
  analogWrite(redLedPin, red);
  analogWrite(greenLedPin, green);
  analogWrite(blueLedPin, blue);
  storeLEDColour(red, green, blue);
}

void storeLEDColour(uint8_t *red, uint8_t *green, uint8_t *blue) {
  currentLEDColour[0] = red;
  currentLEDColour[1] = green;
  currentLEDColour[2] = blue;
}

/* --------------------------------------------------------------------
   Calibrate angle function.

   This function is used in conjunction with the photointerrupter to
   calibrate the angle of the LiDAR to an origin value (facing ON/OFF
button).
   ------------------------------------------------------------------- */
void calibrateAngle() {
  Serial.println(F("Calibrating LiDAR angle..."));
  if (calibrate == false) {
    calibrate = true;
  }
  while (calibrate) {
    // Check if long press of button has occured (if so, reset system)
    checkForBluetoothCommand();
    checkButtonFlags();
    if (reset) {
      stepper.stop();
```

```
      stepper.disable();
      reset = true;
      calibrate = false;
      break;
    }
    // motor control loop - send pulse and return how long to wait until
next pulse
    unsigned waitTime = stepper.nextAction();

    // 0 wait time indicates the motor has stopped
    if (waitTime <= 0) {
      stepper.enable();
      stepper.startRotate(360);
    }
  }
  delay(500);
}

/* ---------------------------------------------------------------------
   Photointerrupter interrupt function used to calibrate the stepper motor.

   This function is called every time the photointerrupt pin reads LOW
(object blocking IR).
   Thus, the photointerrupter is used as an encoder for the stepper motor
to have an origin.
   -------------------------------------------------------------------- */
void photoInterruptISR() {
  if (calibrate) {
    stepper.stop();
    stepper.disable();
    calibrationCount++;
    if (calibrationCount == 3) {
      calibrate = false;
      calibrationCount = 0;
    }
  }
  angle = 0;
}

/* ---------------------------------------------------------------------
   Calculates the required steps to rotate a specified angle

   The stepper motor can therefore be rotated the specified number of steps
   to achieve the angle rotation desired.
   -------------------------------------------------------------------- */
int16_t calcStepsForRotation(int16_t angle) {
  return round(angle * MOTOR_STEPS * (float)MICROSTEPS / 360);
  //return round(degree * MOTOR_STEPS * (uint16_t)MICROSTEPS / 360);
}

/* ---------------------------------------------------------------------
   Button module interrupt function.

   This function is called every time the button module state CHANGES (is
pressed or released).
   If a CHANGE occurs, then an interrupt flag is set for the main code to
process further.
         - Using flags minimises the time within the ISR.

   The function checks the time since last press to ensure no extra
triggers occur due to debounce.
```

```
   -------------------------------------------------------------------- */
void buttonISR() {
  bool state = digitalRead(buttonPin);
  if (state == LOW) { // If the button is pressed
    buttonISRHigh = true;
    printf_P(PSTR("WE HERE\r\n"));
    if (buttonTime.state() == StopWatch::RESET || buttonTime.state() ==
StopWatch::STOPPED)  {
      buttonTime.reset();
      buttonTime.start();
    }
  }
  else {                // If the button is released
    buttonTime.stop();
    uint32_t t = buttonTime.value();
    if (t > DEBOUNCE_DELAY && t < LONG_PRESS) { //Legitimate button press
(not a debounce)
      buttonISRLow = true;
      printf_P(PSTR("Time since button press: %lu\r\n"), t);
    } else {
      buttonISRHigh = false;
      //Serial.println(F("Button debounce occured!"));
    }
  }
}

/* --------------------------------------------------------------------
    Button module interrupt flag checking function

   This function checks the status of the button ISR HIGH & LOW booleans to
see
   if the button was pressed or released. If it was, then process the
necessary code here.
   -------------------------------------------------------------------- */
void checkButtonFlags() {
  if (buttonTime.elapsed() >= LONG_PRESS) {
    modifyLed(0, 65, 100);
    reset = true;
  }
  if (buttonISRHigh) {
    for (int i = 0; i < 3; i++) {
      printf_P(PSTR("currentLEDColour[%d] = %u\r\n"), i,
currentLEDColour[i]);
      prevLEDColour[i] = currentLEDColour[i];
    }
    modifyLed(100, 0, 65);
    tone(piezoPin, 700, 150);
    buttonISRHigh = false;
  } else if (buttonISRLow) {
    printf_P(PSTR("Time since button press: %lu\r\n"), buttonTime.value());
    modifyLed(prevLEDColour[0], prevLEDColour[1], prevLEDColour[2]);
    testDifficulty++;
    if (testDifficulty > DIFFICULTY_NUM) {
      testDifficulty = 1;
    }
    printf_P(PSTR("DIFFICULTY CHANGED TO: %u\r\n"), testDifficulty);
        reset = true;
    stepper.stop();
    stepper.disable();
    calibrate = false;
    buttonISRLow = false;
```

```
  }
}

/* -----------------------------------------------------------------
   Checks if a BlueTooth Message has been received on the HM-10.

   Checks if the msg matches any of the following commands:
       NEXT : Once test is completed, send this when ready to run the next
test.
       MODE : Change the test difficulty (change the distance between
Markers
       RESET: Reset the program.
       RESULTS: Print the results (Time & Marker Sequence) of the last
test.
       TNUM: Prints the current test number.
       TESTX: Prints the results of Test X. // TO DO
   ------------------------------------------------------------------ */
void checkForBluetoothCommand() {
  uint8_t count = 0;
  while (bluetooth.available()) {
    char c = bluetooth.read();
    //  Debugging code ---- {
    if (count == 0) {
      Serial.print(F("\r\n-------   HM-10 >>"));
    }
    Serial.print(c);
    // ---- } end debugging code
    if (c == '\n') {
      btCommand[ count ] = '\0'; // NULL-terminate the character array
      // Use C string functions to parse the response
      if (strcmp_P( btCommand, PSTR("NEXT") ) == 0) {
        bluetooth.print(F("Command: NEXT\r\n"));
        if (rxNextTestMsg) { // Do nothing if the user sends this command
before the system is ready
          runNextTest = true;
        }
      } else if (strcmp_P( btCommand, PSTR("MODE") ) == 0) {
        testDifficulty++;
        if (testDifficulty > DIFFICULTY_NUM) {
          testDifficulty = 1;
        }
        bluetooth.print(F("Command: MODE. Test difficulty now = "));
        bluetooth.print(testDifficulty);
        bluetooth.print(F("\r\n"));
                reset = true;
      } else if (strcmp_P( btCommand, PSTR("RESET") ) == 0) {
        bluetooth.print(F("Command: RESET. Resetting...\r\n"));
        reset = true;
      } else if (strcmp_P( btCommand, PSTR("RESULTS") ) == 0) {
        bluetooth.print(F("Command: RESULTS.\r\nTEST "));
        bluetooth.print(testNum + 1);
        bluetooth.print(F("=========\r\nTime = "));
        bluetooth.print(pastTestTimes[testNum]);
        bluetooth.print(F("\r\nMarker Sequence = "));
        bluetooth.print(pastPermutations[testNum]);
        bluetooth.print(F("\r\n==============\r\n"));
      } else if (strcmp_P( btCommand, PSTR("TNUM") ) == 0) {
        bluetooth.print(F("Command: TNUM. Test no. = "));
        bluetooth.print(testNum + 1);
        bluetooth.print(F("\r\n"));
      } else if (strstr_P( btCommand, PSTR("TEST") ) != nullptr) {
```

```
        bluetooth.print(F("Command: TESTX. // TO DO"));
      }

      // Reset the counter to receive another response
      count = 0;
    }
    else if (c >= ' ') {
      if (count < BT_MAX - 1) {
        btCommand[ count++ ] = c;
      }
    }
    btCommandReceived = true;
    delay(2); // To ensure the whole message is received
  }
}
/* -------------------------------------------------------------------
   Read Continuous Distance Measurements from the LiDAR

   The most recent distance measurement can always be read from
   device registers. Polling for the BUSY flag in the STATUS
   register can alert the user that the distance measurement is new
   and that the next measurement can be initiated. If the device is
   BUSY this function does nothing and returns 0. If the device is
   NOT BUSY this function triggers the next measurement, reads the
   distance data from the previous measurement, and returns 1.

 * *** NOTE: function copied from Example code produced by manufacturer
Garmin.
             Available at:
https://github.com/garmin/LIDARLite_Arduino_Library
   ------------------------------------------------------------------ */
uint8_t distanceContinuous(uint16_t * distance)
{
  uint8_t newDistance = 0;

  // Check on busyFlag to indicate if device is idle
  // (meaning = it finished the previously triggered measurement)
  if (lidarLite.getBusyFlag() == 0)
  {
    // Trigger the next range measurement
    lidarLite.takeRange();

    // Read new distance data from device registers
    *distance = lidarLite.readDistance();

    // Report to calling function that we have new data
    newDistance = 1;
  }

  return newDistance;
}

/* -------------------------------------------------------------------
   Smooths the distance measurements from the LiDAR.

   The LiDAR will generally give a readings that can be quite noisy;
  // so calculating a running average means that the average is now
  // smoothed and produces less noisy data.
   ------------------------------------------------------------------ */
uint16_t calcRunningAverage(uint8_t *newDistance, uint16_t *distance) {
  if (newDistance) {
```

```
    runningAvgTotal -= lidarDistances[runningAvgIndex];
    lidarDistances[runningAvgIndex] = *distance;
    runningAvgTotal += lidarDistances[runningAvgIndex];
    runningAvgIndex++;
    if (runningAvgIndex >= RUNNING_AVG_NUM) {
      runningAvgIndex = 0; // Wrap around to initial circular linked list
value
    }
  }
  return (runningAvgTotal / RUNNING_AVG_NUM);
}

/* -----------------------------------------------------------------------
   Function that resets all Marker Ready Status flags.

   These flags are used to help the Master determine which Markers are
   ready so that it can move onto the next step of the test set-up.
   -------------------------------------------------------------------- */
void resetMarkerReadyStatus() {
  readyMarker1 = false;
  readyMarker2 = false;
  readyMarker3 = false;
  readyMarker4 = false;
  readyMarker5 = false;
  readyMarker6 = false;
}
```

J.4. Slave Program

```
/*
   Slave component of the Sports Agility Tester.

   Copyright (C)2018 Reuben Smith

*/

/*-----------------------------------------------------*/
/*               #include Header Files                 */
/*-----------------------------------------------------*/
#include <Arduino.h>
#include <stdint.h>
/* --- Include .h file needed for simple timekeeping --- */
#include <StopWatch.h>
/* --- Include .h files needed for NRF24L01 --- */
#include <SPI.h>
#include "RF24.h"
#include "nRF24L01.h"
#include "printf.h" // For debugging
/* --- Include .h files needed for IR Prox/Ambient Light/UV Sensor --- */
#include <Wire.h>
#include "Adafruit_SI1145.h"
/* --- Include .h files needed to generate true random values --- */
#include <Entropy.h>
#include <math.h>
#include <stdlib.h>
/* --- Include .h files needed for RGB LED Matrix --- */
#define FASTLED_ALLOW_INTERRUPTS 0
#define FASTLED_INTERRUPT_RETRY_COUNT 1
#include <FastLED.h>
/* --- Include .h files to access PROGMEM data (used to save space on
Arduino SRAM) --- */
#include <avr/pgmspace.h>
/* --- Include .h files required to put the Arduino to sleep --- */
#include <avr/sleep.h>
#include <avr/power.h>
#include <avr/wdt.h>

/*-----------------------------------------------------*/
/*         Set-up Arduino Digital Pins                 */
/*-----------------------------------------------------*/
#define buttonPin     3   // Button module
#define ledMatrixPin  6   // RGB LED Matrix
#define speakerPin    12  // Speaker
#define piezoPin      13  // Piezo buzzer
#define nrf24irqPin   19  // nRF24L01 interrupt pin
#define cePIN         49  // Chip-Enable (nRF24L01)
#define csnPin        53  // Chip-Select-Not (nRF24L01)

/*-----------------------------------------------------*/
/*         Set-up RGB Neopixel LED Matrix              */
/*-----------------------------------------------------*/
#define COLOR_ORDER GRB
#define CHIPSET     WS2812
#define WIDTH 16
#define HEIGHT 8
#define NUM_LEDS (WIDTH * HEIGHT)
#define BRIGHTNESS_INDOORS percentToPWM(50)
#define BRIGHTNESS_OUTDOORS percentToPWM(100)
```

```
bool useIndoorBrightness = true;

CRGB leds[NUM_LEDS];

/* --- Store LED matrix characters displayed --- */
char currentCharacter[65];
char prevCharacter[65];
uint8_t currentHue = 0;
uint8_t prevHue = 0;
bool progmem = true;
/*----------------------------------------------------*/
/*  Set-up SEN36002 IR Prox/Ambient Light/UV Sensor   */
/*----------------------------------------------------*/
Adafruit_SI1145 sensor = Adafruit_SI1145();
#define LIMIT_MULTIPLIER_BALL 0.02 // ( * 100 = %)
#define LIMIT_MULTIPLIER_HAND 0.005
//#define LIMIT_MULTIPLIER_BALL 0.02
#define RUNNING_AVG_NUM 25  // Trade-off => Noise vs. Responsive Data
Sensing

/* --- Upper and Lower Limit Multiplier (add/remove XX% of the baseline
value) --- */
float limitMultiplier = LIMIT_MULTIPLIER_HAND; // Toggle this value to
detect removal of ball or swiping of a hand

/* --- Toggle Si1145 sensor to retrieve '0' data values --- */
bool useProx0 = true;
bool useIR0 = false;
bool useALS0 = false;

/* --- Toggle Si1145 sensor to retrieve '1' data values --- */
bool useProx1 = false;
bool useIR1 = false;
bool useALS1 = false;

/* --- Sensor values stored in arrays to calc running avg (smooth the data)
--- */
uint8_t runningAvgIndex = 0;
uint16_t proxValues[RUNNING_AVG_NUM];
uint32_t proxRunningAvgTotal = 0;
uint16_t proxRunningAverage = 0;
uint16_t irValues[RUNNING_AVG_NUM];
uint32_t irRunningAvgTotal = 0;
uint16_t irRunningAverage = 0;
uint16_t alsValues[RUNNING_AVG_NUM];
uint32_t alsRunningAvgTotal = 0;
uint16_t alsRunningAverage = 0;

/* --- The Si1145 records two different sensor data values for each prox, -
-- */
/* --- ir and als, so store these values too                            -
-- */
uint16_t _proxValues[RUNNING_AVG_NUM];
uint32_t _proxRunningAvgTotal = 0;
uint16_t _proxRunningAverage = 0;
uint16_t _irValues[RUNNING_AVG_NUM];
uint32_t _irRunningAvgTotal = 0;
uint16_t _irRunningAverage = 0;
uint16_t _alsValues[RUNNING_AVG_NUM];
uint32_t _alsRunningAvgTotal = 0;
uint16_t _alsRunningAverage = 0;
```

```
/*-----------------------------------------------------------------------
*/
/*  Set-up nRF24L01 radio & pipe addresses for master and slave
devices   */
/*-----------------------------------------------------------------------
*/
#define MASTER_PIPE 1
#define PREV_MARKER_PIPE 2
#define NEXT_MARKER_PIPE 3
RF24 radio(cePIN, csnPin);
const uint64_t pipes[7] = { 0x3e64b727ffLL, 0x714e2b65c1LL, 0x714e2b65c2LL,
0x714e2b65c3LL,
                            0x714e2b65c4LL, 0x714e2b65c5LL, 0x714e2b65c6LL
                          };
const char leastSignificantByte[][3] = { "ff", "c1", "c2", "c3", "c4",
"c5", "c6" };

/*----------------------------------------------------*/
/*         Set-up Time Keeping Variables              */
/*----------------------------------------------------*/
StopWatch time(StopWatch::MICROS);
StopWatch testTime(StopWatch::MICROS);
volatile StopWatch buttonTime(StopWatch::MILLIS); // All variables handled
in ISRs need to be defined as volatile
StopWatch nrf24Time(StopWatch::MICROS);

/*----------------------------------------------------*/
/*            Declare Global Variables                */
/*----------------------------------------------------*/
#define MARKER_NUMBER 6
#define UPPER_CHARACTER_SET 0
#define LOWER_CHARACTER_SET 1
#define DIGIT_CHARACTER_SET 2
int8_t slaveNumber;
/*----------------------------------------------------*/
/*              Button Interrupt Flags                */
/*----------------------------------------------------*/
#define LONG_PRESS 2000
#define DEBOUNCE_DELAY 50
bool ballOnMarker = false;
volatile bool buttonISRHigh = false;
volatile bool buttonISRLow = false;

/*----------------------------------------------------*/
/*               Declare Other Flags                  */
/*----------------------------------------------------*/
bool calibrate = true;
bool reset = false;
/*-----------------------------------------------------------------------
*/
/*                   RGB LED Matrix
Characters                            */
/*-----------------------------------------------------------------------
*/
const char LETTERS_UPPER[][65] PROGMEM = { // Store in PROGMEM to save SRAM
  // [0] A:
  { "0001100000111100011001101100001111111111111111111100001111000011"},

  // [1] B:
  { "1111110011111110110011101111110011111110110001101111111011111100"},
```

```
// [2] C:
{ "0011110001111110111001111100000011000000111001110111111000111100"},

// [3] D:
{ "1111100011111110110011101100011111000111110011101111111011111000"},

// [4] E:
{ "1111111111111111110000001111111011111110110000001111111111111111"},

// [5] F:
{ "1111111111111111110000001111110011111100110000001100000011000000"},

// [6] G:
{ "0011110001111110111001111100000011001111111001110111111000111100"},

// [7] H:
{ "1100001111000011110000111111111111111111110000111100001111000011"},

// [8] I:
{ "1111111111111111000110000001100000011000000110001111111111111111"},

// [9] J:
{ "0111111101111111000011000000110000001100110011001111110001111000"},

// [10] K:
{ "1100011111001110110111001111100011111000110111001100111011000111"},

// [11] L:
{ "1100000011000000110000001100000011000000110000001111111111111111"},

// [12] M:
{ "1100011011101110111111101111111011010110110001101100011011000110"},

// [13] N:
{ "1100001111100011111100111111101111011111110011111100011111000011"},

// [14] O:
{ "0011110001111110111001111100001111000011111001110111111000111100"},

// [15] P:
{ "1111110011111110110001111100011111111110111111001100000011000000"},

// [16] Q:
{ "0011110001111110111001111100001111010110111111000111100000000111"},

// [17] R:
{ "1111111011111111110001111111111011111100110111001100111011000111"},

// [18] S:
{ "0111111011111111110000000111111000111111000000011111111101111110"},

// [19] T:
{ "1111111111111111000110000001100000011000000110000001100000011000"},

// [20] U:
{ "1100001111000011110000111100001111000011110000111111111101111110"},

// [21] V:
{ "1100011011000110110001101100011011000110011011000011100000010000"},

// [22] W:
```

```
  { "1100011011000110110001101101011011111110111111101110111011000110"},

  // [23] X:
  { "1100001111100111001111100000011000000111100011111110111001111000011"},

  // [24] Y:
  { "1100001111100111011111100011110000001100000011000000110000001100 0"},

  // [25] Z:
  { "1111111111111111000000110000111000011100001110000111111111111111"}
};

const char LETTERS_LOWER[][65] PROGMEM = { // Store in PROGMEM to save SRAM
  // [0] a:
  { "0000000001111110011111110000011101111111111001110111111100111110"},

  // [1] b:
  { "1100000011000000111111001111111011000111110001111111111011111100"},

  // [2] c:
  { "0000000000111100011111101110011111000000111001110111111000111100"},

  // [3] d:
  { "0000001100000011001111110111111111100011111000110111111100111111"},

  // [4] e:
  { "0000000000111100011111101110011011111110111000000111111000111110"},

  // [5] f:
  { "0001111000111111001100011111110011111100001100000011000000110000"},

  // [6] g:
  { "0001111100111111001100110011000111110000001101111111011111110"},

  // [7] h:
  { "1100000011000000110000001111110011111110110001101100011011000110"},

  // [8] i:
  { "0011100000111000000000000011100000111000001110000011100001111100"},

  // [9] j:
  { "0000011100000111000000000000011100000111011101110111111000111100"},

  // [10] k:
  { "0110000001100000011011100111110001110000011111000110111001100111"},

  // [11] l:
  { "0011100000111000001110000011100000111000001110000011100000111000"},

  // [12] m:
  { "0000000011000000111011001111111011010110110101101101011011000110"},

  // [13] n:
  { "0000000011000000111111001111111011000110110001101100011011000110"},

  // [14] o:
  { "0000000000111100011111101110011111000011111001110111111000111100"},

  // [15] p:
  { "1100000011111100111111101110011011111110011000000110000001100000 0"},
```

```
  // [16] q:
  { "0000110001111100111111001100110001111100000011000000111100001110"},

  // [17] r:
  { "0000000011000000111111001111110110001101100000011000000110000000"},

  // [18] s:
  { "0000000001111110111111111111000000111111000000111111111101111110"},

  // [19] t:
  { "0001100000011000011111100111111000011000000110100001111000011100"},

  // [20] u:
  { "0000000001100011011000110110001101100011011000110111111100111101"},

  // [21] v:
  { "0000000001100011011000110110001101100011011111110011111000001000"},

  // [22] w:
  { "0000000011000110110001101101011011010110110101101111111001111100"},

  // [23] x:
  { "0000000011000011011001100011110000001100000011110001100110110000011"},

  // [24] y:
  { "0110001101100011011000110011111100000011110000110111111100111110"},

  // [25] z:
  { "0000000001111110111111000001100001110000110000011111110111111 1"}
};

const char DIGITS[][65] PROGMEM = { // Store in PROGMEM to save SRAM
  // [0] 0:
  { "0011110001111110110001111100111111011011111100110110011000111100"},

  // [1] 1:
  { "0001100000111000011110000001100000011000000110000111111001111110"},

  // [2] 2:
  { "0011110001111110111001110000111100011100001100001111111111111111"},

  // [3] 3:
  { "0011110001111110111001110000011100011110110001111111111101111110"},

  // [4] 4:
  { "0000110000011100001110001101100110011101111111100000110000001100"},

  // [5] 5:
  { "1111111111111111111100000011111110000001111000011111111101111110"},

  // [6] 6:
  { "0011110001111111111100011111000001111111011100111011111101111110"},

  // [7] 7:
  { "0111111111111111111000111000011100000110000001100000011000000011000"},

  // [8] 8:
  { "0011110001111110111001111111111100111100111001111111111101111110"},

  // [9] 9:
  { "0011111001111110111001111110011101111111100000111111111001111100"}
```

```
};

const char RANDOM_CHARACTERS[][65] PROGMEM = { // Store in PROGMEM to save
SRAM
  // [0] Tick:
  { "0000000000000000100000011000001101000110011011000011100000001100000"},

  // [1] Cross:
  { "0000000011000011011001100011110000001100000111100011001101100011"},

  // [2] Exclamation Mark:
  { "0001100000011000000110000001100000011000000000000001100000011000"},

  // [3] Question Mark:
  { "0011110001100111010000110001111000001100000000000000110000001100"},

  // [4] Happy:
  { "0110011001100110011001100000000001000000111000011011111100011100"},

  // [5] Sad:
  { "0110011001100110011001100000000000011110001111110110000111000000"},

  // [6] Down:
  { "0011100000111000001110000011100011111110011111000011100000010000"},

  // [7] Up:
  { "0001000000111000011111001111111000111000001110000011100000111000"},

  // [8] Left:
  { "0001000000110000011111111111111101111111001100000001000000000000"},

  // [9] Right:
  { "0000100000001100111111101111111111111111000001100000010000000000"},

  // [10] Square:
  { "1111111111111111110000111100001111000011110000111111111111111111"},

  // [11] Heart:
  { "0000000001100110111111111111111111111111011111100011110000011000"},

  // [12] Circle:
  { "0011110001111110111001111100001111000011111001110111111000111100"},

  // [13] Star:
  { "0001100000011000111111110111111000111100011111100110011011000011"},

  // [14] Pacman:
  { "0011110001111110011101111111110011111000111111100111111100111100"},

  // [15] Ghost:
  { "0011100001111100111111101110101011111110111111101111111010101010"}
};


/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
/*                     Set-up                         */
/*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/

void setup() {
  /* ---------- Set-up PC COM Serial ---------- */
  Serial.begin(115200);
```

```
  printf_begin();

  Serial.print(F("/*------------------------------------------------------
*/\r\n"
                 "     Slave component of the Sports Agility
Tester.\r\n\r\n"
                 "     Copyright (C)2018 Reuben Smith\r\n/"
                 "*-------------------------------------------------------
*/\r\n\r\n"));
  /* ---------- Set-up RGB LED 8x8 Matrix ---------- */
  FastLED.addLeds<CHIPSET, ledMatrixPin, COLOR_ORDER>(leds,
NUM_LEDS).setCorrection(TypicalSMD5050);
  if (useIndoorBrightness) {
    FastLED.setBrightness(BRIGHTNESS_INDOORS);
    printf_P(PSTR("[BRIGHTNESS] Indoor Brightness set to: %u\r\n\r\n"),
BRIGHTNESS_INDOORS);
  } else {
    FastLED.setBrightness(BRIGHTNESS_OUTDOORS);
    printf_P(PSTR("[BRIGHTNESS] Outdoor Brightness set to: %u\r\n\r\n"),
BRIGHTNESS_OUTDOORS);
  }
  // Blink a green square on the LED matrix to indicate power ON
  for (int i = 0; i < 3; i++) {
    displayCharacter(RANDOM_CHARACTERS[12], 96, progmem);
    delay(400);
    displayCharacter(RANDOM_CHARACTERS[13], 96, progmem);
    delay(400);
  }
  /* ---------- Uncomment code to test brightness ---------- */
  //  uint8_t pwm = 100;
  //  for (int i = 0; i < 20; i++) {
  //    FastLED.setBrightness(percentToPWM(pwm));
  //    displayCharacter(RANDOM_CHARACTERS[12], 96, progmem);
  //    delay(1000);
  //    pwm -= 5;
  //    if (pwm < 0) {
  //      pwm = 0;
  //    }
  //  }

  /* ---------- Define digital output for pins of corresponding hardware --
-------- */
  pinMode(buttonPin, INPUT_PULLUP);
  //  pinMode(nrf24irqPin, INPUT_PULLUP);

  /* ---------- Attach interrupts for button and nRF24L01 IRQ pin ---------
- */
  attachInterrupt(digitalPinToInterrupt(buttonPin), buttonISR, CHANGE);
  //  attachInterrupt(digitalPinToInterrupt(nrf24irqPin), nrf24ISR,
FALLING);

  /* ---------- Set-up NRF242L01 radio ---------- */
  radio.begin();

  // Set-up auto Ack Payloads
  radio.setAutoAck(true);                    // Ensure autoACK is enabled
  radio.enableAckPayload();               // Allow optional ack payloads
  radio.setPayloadSize(7);                // Send 2-byte payloads
  radio.setDataRate(RF24_1MBPS);

  //  radio.setDataRate(RF24_250KBPS); // Faster with better range
```

```
  radio.setPALevel(RF24_PA_MAX);
  radio.setChannel(108);
  radio.setRetries(0, 15);
  radio.openReadingPipe(1, pipes[0]); // Open a reading pipe with the
Master device
  //  radio.openReadingPipe(0, pipes[0]); // was this

//  radio.maskIRQ(1, 1, 0); // Only interrupt if nRF24L01 received a
message (mask fail & TX)

  Serial.println(F("--------------- nRF24L01 Details ---------------"));
  radio.printDetails();
  Serial.println(F("\r\n--------------- Hardware Connections --------------
-"));
  /* ---------- Check the nRF24L01 is connected correctly ---------- */
  if (radio.isChipConnected()) {
    Serial.println(F("nRF24L01 is Connected!\r\n"));
  } else {
    Serial.println(F("nRF24L01 is NOT Connected!\r\n"));
    Serial.println(F("   - Displaying nRF24L01 not connected error.\r\n"));
    for (int i = 0; i < 2; i++) {
      // Display nRF24L01 error
      displayCharacter(RANDOM_CHARACTERS[2], 0, progmem); // Display
exclamation mark to indicate error
      delay(2000);
      displayCharacter(LETTERS_LOWER[13], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[17], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[5], 0, progmem);
      delay(500);
      displayCharacter(DIGITS[2], 0, progmem);
      delay(500);
      displayCharacter(DIGITS[4], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[11], 0, progmem);
      delay(500);
      displayCharacter(DIGITS[0], 0, progmem);
      delay(500);
      displayCharacter(DIGITS[1], 0, progmem);
      delay(1000);
    }
    Serial.println(F("   - Finished displaying nRF24L01 not connected
error.\r\n"));
  }
  /* ---------- Check the SEN36002/Si1145 sensor is connected correctly ---
------- */
  if (sensor.begin()) {
    Serial.println(F("Si1145 is Connected!\r\n"));
  } else {
    Serial.println(F("Si1145 is NOT Connected!\r\n"));
    Serial.println(F("   - Displaying sensor not connected error.\r\n"));
    for (int i = 0; i < 2; i++) {
      // Display SENSOR error
      displayCharacter(RANDOM_CHARACTERS[2], 0, progmem); // Display
exclamation mark to indicate error
      delay(2000);
      displayCharacter(LETTERS_UPPER[18], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[4], 0, progmem);
      delay(500);
```

```
      displayCharacter(LETTERS_UPPER[13], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[18], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[14], 0, progmem);
      delay(500);
      displayCharacter(LETTERS_UPPER[17], 0, progmem);
      delay(500);
      displayCharacter(RANDOM_CHARACTERS[2], 0, progmem); // Display
exclamation mark to indicate error
    }
    Serial.println(F("   - Finished displaying sensor not connected
error.\r\n"));
  }

  /* ---------- Set-up random number generator ---------- */
  Entropy.initialize();

  /* ---------- Indicate set-up is complete ---------- */
  displayCharacter(RANDOM_CHARACTERS[0], 96, progmem); // Display green
tick
  tone(piezoPin, 800, 150); // [Pin#, Frequency (in Hz), Duration (in ms)]
  delay(250);
  tone(piezoPin, 800, 150);
  Serial.println(F("------------------------------------------------"));
  Serial.println(F("Set-up complete."));
  Serial.println(F("------------------------------------------------
\r\n"));
  delay(500);
  turnLEDsOff();
  //Serial.print(F("Free RAM: ")); Serial.println(checkFreeRAM());
}

/* ##################################################### */
/*                       Main loop                       */
/* ##################################################### */
void loop()
{
  if (reset) {
    reset = !reset;
    Serial.println(F("RESET OCCURRED"));
    for (int i = 0; i < 3; i++) {
      tone(piezoPin, 550, 500);
      delay(600);
    }
    turnLEDsOff();
    /* ---------- Reset the button time in case a debounce occurred just
after a long press ---------- */
    buttonTime.stop();
    buttonTime.reset();
    delay(2000);
  }
  /* ------------------------------------------------------- */
  /*                  Connect to the Master                  */
  /* ------------------------------------------------------- */
  bool connected = false;
  nrf24Time.start();
  Serial.println("Waiting for signal from Master to connect.");
    while (!connected) {
      radio.startListening();
```

```
      checkButtonFlags();
      if (reset) {
        break;
      }
      char dataReceived[3];
      byte pipeNum;
      while (radio.available(&pipeNum) ) {
        nrf24Time.stop();
        radio.read( &dataReceived, sizeof(dataReceived) );
        printf_P(PSTR("[RX] Data received: %s\r\n"), dataReceived);
        for (int i = 0; i < MARKER_NUMBER; i++) {
          /* ---------- Check if the message matches any slave addresses --
-------- */
          if (strstr(dataReceived, leastSignificantByte[i + 1]) != nullptr)
{ // Ignore all data after in case rubbish
            //      if (strcmp(dataReceived, leastSignificantByte[i + 1])
== 0) {
            printf_P(PSTR("Matched with unique LSB: %s\r\n"),
leastSignificantByte[i + 1]);
            slaveNumber = i + 1;
            printf_P(PSTR("[MARKER NUM] This Marker is number %d\r\n"),
slaveNumber);
            radio.stopListening();
            radio.openWritingPipe(pipes[slaveNumber]);
//            radio.write(&dataReceived, sizeof(dataReceived));
            radio.write(&leastSignificantByte[i + 1],
sizeof(leastSignificantByte[i + 1])); // Send the Master which Slave
connected
            //          radio.openReadingPipe(0, pipes[0]); // Reopen
reading pipe as pipe 0 is used by the writing pipe
            radio.startListening();
            printf_P(PSTR("Waited %lu ms to connect. Marker connected to
Master using address %s\r\n"), nrf24Time.value(), leastSignificantByte[i +
1]);
            connected = true;
          }
        }
      }
      delay(30);
    }
//  slaveNumber = 1; // For Debugging w/out nRF24L01
  /* ------------------------------------------------------- */
  /*             Display Marker Position Status              */
  /* ------------------------------------------------------- */
  bool settingUpMarker = true;
  char dataReceived[3];
  byte pipeNum;
    while (settingUpMarker) {
      checkButtonFlags();
      if (reset) {
        break;
      }
      while ( radio.available(&pipeNum)) {
        radio.read( &dataReceived, sizeof(dataReceived) );
//        radio.writeAckPayload(pipeNum, &dataReceived,
sizeof(dataReceived) ); // Send auto acknowledgement payload
        printf_P(PSTR("[RX] Data received: %s\r\n"), dataReceived);

        /* ---------- Correct Position Msg ---------- */
        if (strstr(dataReceived, "CP") != nullptr) {
          radio.stopListening();
```

```
          radio.write(&dataReceived, sizeof(dataReceived));
          radio.startListening();
          printf_P(PSTR("Marker in CORRECT position!\r\n"));
          displayCharacter(RANDOM_CHARACTERS[0], 96, progmem);
        }
        /* ---------- Incorrect Position Msg ---------- */
        else if (strstr(dataReceived, "IP") != nullptr) {
          radio.stopListening();
          radio.write(&dataReceived, sizeof(dataReceived));
          radio.startListening();
          printf_P(PSTR("Marker in INCORRECT position!\r\n"));
          displayCharacter(RANDOM_CHARACTERS[1], 0, progmem);
        }
        /* ---------- Position Validated in Correct Position Msg ----------
*/
        else if (strstr(dataReceived, "PV") != nullptr) {
          radio.stopListening();
          radio.write(&dataReceived, sizeof(dataReceived));
          radio.startListening();
          printf_P(PSTR("Marker position VALIDATED!\r\n"));
          displayCharacter(RANDOM_CHARACTERS[4], 96, progmem);
          tone(piezoPin, 800, 500); // [Pin#, Frequency (in Hz), Duration
(in ms)]
          settingUpMarker = false;
          delay(1000); // Delay so user knows the marker has had its
position validated
          break;
        }
      }
    }
  turnLEDsOff();
  /* ------------------------------------------------------- */
  /*                  Test Set-up & Run Loop                  */
  /* ------------------------------------------------------- */
  uint16_t prox, ir, als, _prox, _ir, _als, proxNoBall, irNoBall,
alsNoBall, _proxNoBall, _irNoBall, _alsNoBall;
  bool runningTests = true;
  while (runningTests) {
    checkButtonFlags();
    if (reset) {
      break;
    }

    /* ------------------------------------------------------- */
    /*            Calibrate Sensor Values with No Ball          */
    /* ------------------------------------------------------- */
    bool printBelow = true;
    if (useProx0 && useProx1 || useIR0 && useIR1 || useALS0 && useALS1 ||
printBelow) {
      printf_P(PSTR("------- Si1145 Sensor Values [NO BALL] -------
\r\n"));
//      time.start();
      for (int i = 0; i < (RUNNING_AVG_NUM * 5); i++) { // Preload the
running average arrays (five times to ensure valid values)
        /* ---------- Read sensor data from first source value (this data
is a lot more stable) ---------- */
        prox = sensor.readProx();    // PS1_DATA0
        ir = sensor.readIR();        // ALS_IR_DATA0
        als = sensor.readVisible();  // ALS_VIS_DATA0
        /* ---------- Read sensor data from second source value (less
stable data but more sensitive) ---------- */
```

```
        if (useProx1) _prox = sensor.readProxOther();      // PS1_DATA1
        if (useIR1) _ir = sensor.readIROther();         // ALS_IR_DATA1
        if (useALS1) _als = sensor.readVisibleOther();  // ALS_VIS_DATA1

        proxRunningAverage = calcRunningAverage(&prox, proxRunningAvgTotal,
proxValues);
        irRunningAverage = calcRunningAverage(&ir, irRunningAvgTotal,
irValues);
        alsRunningAverage = calcRunningAverage(&als, alsRunningAvgTotal,
alsValues);

        if (useProx1) _proxRunningAverage = calcRunningAverage(&_prox,
_proxRunningAvgTotal, _proxValues);
        if (useIR1) _irRunningAverage = calcRunningAverage(&_ir,
_irRunningAvgTotal, _irValues);
        if (useALS1) _alsRunningAverage = calcRunningAverage(&_als,
_alsRunningAvgTotal, _alsValues);

        incrementRunningAvgIndex();
      } // 5 times RUNNING_AVG_NUM (25) takes ~0.2sec to process
//      time.stop();
//      Serial.println(time.value());
      proxNoBall = proxRunningAverage;
      irNoBall = irRunningAverage;
      alsNoBall = alsRunningAverage;
      if (useProx1) _proxNoBall = _proxRunningAverage;
      if (useIR1) _irNoBall = _irRunningAverage;
      if (useALS1) _alsNoBall = _alsRunningAverage;

      printf_P(PSTR("[AVG] Prox: %u\r\n"), proxRunningAverage);
      printf_P(PSTR("[AVG] IR: %u\r\n"), irRunningAverage);
      printf_P(PSTR("[AVG] ALS: %u\r\n\r\n"), alsRunningAverage);
      if (useProx1) printf_P(PSTR("[AVG] _Prox: %u\r\n"),
_proxRunningAverage);
      if (useIR1) printf_P(PSTR("[AVG] _IR: %u\r\n"), _irRunningAverage);
      if (useALS1) printf_P(PSTR("[AVG] _ALS: %u\r\n\r\n"),
_alsRunningAverage);
    }
    /* ------------------------------------------------------- */
    /*              User Positions Ball on Marker              */
    /* ------------------------------------------------------- */
    putArduinoToSleep(); // Put the Arduino in sleep mode to save power,
pressing the button will trigger an INTERRUPT and will wake it
    /*********** Place Ball on Marker then Press Button to Continue
************/
    radio.flush_rx();
    delay(3000); // Allow time for person to move away from Marker
    /* ------------------------------------------------------- */
    /*                 Calibrate Sensor Values                 */
    /* ------------------------------------------------------- */
    printf_P(PSTR("\r\n------- Si1145 Sensor Values [WITH BALL] -------
\r\n"));
    for (int i = 0; i < (RUNNING_AVG_NUM * 5); i++) { // Preload the
running average arrays (five times to ensure valid values)
      /* ---------- Read sensor data from first source value ---------- */
      prox = sensor.readProx();     // PS1_DATA0
      ir = sensor.readIR();         // ALS_IR_DATA0
      als = sensor.readVisible();   // ALS_VIS_DATA0
      /* ---------- Read sensor data from second source value ---------- */
      if (useProx1) _prox = sensor.readProxOther();      // PS1_DATA1
      if (useIR1) _ir = sensor.readIROther();         // ALS_IR_DATA1
```

```
      if (useALS1) _als = sensor.readVisibleOther();  // ALS_VIS_DATA1
      //      printf_P(PSTR("Prox: %u\r\n"), prox);
      //      printf_P(PSTR("IR: %u\r\n"), ir);
      //      printf_P(PSTR("ALS: %u\r\n===================\r\n"), als);
      //      calcRunningAverage(&prox, &ir, &als);

      proxRunningAverage = calcRunningAverage(&prox, proxRunningAvgTotal,
proxValues);
      irRunningAverage = calcRunningAverage(&ir, irRunningAvgTotal,
irValues);
      alsRunningAverage = calcRunningAverage(&als, alsRunningAvgTotal,
alsValues);

      if (useProx1) _proxRunningAverage = calcRunningAverage(&_prox,
_proxRunningAvgTotal, _proxValues);
      if (useIR1) _irRunningAverage = calcRunningAverage(&_ir,
_irRunningAvgTotal, _irValues);
      if (useALS1) _alsRunningAverage = calcRunningAverage(&_als,
_alsRunningAvgTotal, _alsValues);

      incrementRunningAvgIndex();
      /* ---------- Use below code for  debugging sensor data ---------- */
      //            i--;
      //            delay(100);
      //            printf_P(PSTR("AVG Prox: %u\r\n"), proxRunningAverage);
      //            printf_P(PSTR("AVG IR: %u\r\n"), irRunningAverage);
      //            printf_P(PSTR("AVG ALS:
%u\r\n===================\r\n"), alsRunningAverage);
      //            if (useProx1) printf_P(PSTR("AVG _Prox: %u\r\n"),
_proxRunningAverage);
      //            if (useIR1) printf_P(PSTR("AVG _IR: %u\r\n"),
_irRunningAverage);
      //            if (useALS1) printf_P(PSTR("AVG _ALS:
%u\r\n===================\r\n"), _alsRunningAverage);
    } // 5 times RUNNING_AVG_NUM (25) takes ~0.2sec to process

    printf_P(PSTR("[AVG] Prox: %u\r\n"), proxRunningAverage);
    printf_P(PSTR("[AVG] IR: %u\r\n"), irRunningAverage);
    printf_P(PSTR("[AVG] ALS: %u\r\n\r\n"), alsRunningAverage);
    if (useProx1) printf_P(PSTR("[AVG] _Prox: %u\r\n"),
_proxRunningAverage);
    if (useIR1) printf_P(PSTR("[AVG] _IR: %u\r\n"), _irRunningAverage);
    if (useALS1)printf_P(PSTR("[AVG] _ALS: %u\r\n\r\n"),
_alsRunningAverage);

    /* ------------------------------------------------------- */
    /*           Determine which Sensor Values to Use          */
    /* ------------------------------------------------------- */
    //    bool useProx0, useIR0, useALS0;
    //    float calc, _calc;

    /* ---------- Choose the data that has the largest ratio between ball
on and off ---------- */
    if (useProx0 && useProx1) {
      float calc, _calc;
      calc = (proxRunningAverage < proxNoBall) ? (proxRunningAverage /
proxNoBall) : (proxNoBall / proxRunningAverage);
      _calc = (_proxRunningAverage < _proxNoBall) ? (_proxRunningAverage /
_proxNoBall) : (_proxNoBall / _proxRunningAverage);
      if (calc < _calc) {
        useProx0 = true;
```

```
        useProx1 = false;
      } else {
        useProx0 = false;
        useProx1 = true;
      }
    }

    if (useIR0 && useIR1) {
      float calc, _calc;
      calc = (irRunningAverage < irNoBall) ? (irRunningAverage / irNoBall)
: (irNoBall / irRunningAverage);
      _calc = (_irRunningAverage < _irNoBall) ? (_irRunningAverage /
_irNoBall) : (_irNoBall / _irRunningAverage);
      if (calc < _calc) {
        useIR0 = true;
        useIR1 = false;
      } else {
        useIR0 = false;
        useIR1 = true;
      }
    }

    if (useALS0 && useALS1) {
      float calc, _calc;
      calc = (alsRunningAverage < alsNoBall) ? (alsRunningAverage /
alsNoBall) : (alsNoBall / alsRunningAverage);
      _calc = (_alsRunningAverage < _alsNoBall) ? (_alsRunningAverage /
_alsNoBall) : (_alsNoBall / _alsRunningAverage);
      if (calc < _calc) {
        useALS0 = true;
        useALS1 = false;
      } else {
        useALS0 = false;
        useALS1 = true;
      }
    }
    /* -------------------------------------------------------- */
    /*                  Calculate Sensor Limits                 */
    /* -------------------------------------------------------- */
    uint16_t proxLowerLim, proxUpperLim, irLowerLim, irUpperLim,
alsLowerLim, alsUpperLim;
    //    uint32_t tempCalc; // Use 32-bit int in case the value exceeds
uint16_t maximum value

    /* ---------- Proximity values ---------- */
    if (useProx0) {
      printf_P(PSTR("[PS1_DATA0] Using Prox0 Sensor Data.\r\n"));
      calculateSensorLimits(proxRunningAverage, proxUpperLim,
proxLowerLim);
    } else if (useProx1) {
      printf_P(PSTR("[PS1_DATA1] Using Prox1 Sensor Data.\r\n"));
      calculateSensorLimits(_proxRunningAverage, proxUpperLim,
proxLowerLim);
    }

    /* ---------- IR values ---------- */
    if (useIR0) {
      printf_P(PSTR("[ALS_IR_DATA0] Using IR0 Sensor Data.\r\n"));
      calculateSensorLimits(irRunningAverage, irUpperLim, irLowerLim);
    } else if (useIR1) {
      printf_P(PSTR("[ALS_IR_DATA1] Using IR1 Sensor Data.\r\n"));
```

```
      calculateSensorLimits(_irRunningAverage, irUpperLim, irLowerLim);
    }

    /* ---------- ALS values ---------- */
    if (useALS0) {
      printf_P(PSTR("[ALS_VIS_DATA0] Using ALS0 Sensor Data.\r\n"));
      calculateSensorLimits(alsRunningAverage, alsUpperLim, alsLowerLim);
    } else if (useALS1) {
      printf_P(PSTR("[ALS_VIS_DATA1] Using ALS1 Sensor Data.\r\n\r\n"));
      calculateSensorLimits(_alsRunningAverage, alsUpperLim, alsLowerLim);
    }

    printf_P(PSTR("[LOWER] Prox: %u\r\n[UPPER] Prox: %u\r\n"),
proxLowerLim, proxUpperLim);
    printf_P(PSTR("[LOWER] IR: %u\r\n[UPPER] IR: %u\r\n"), irLowerLim,
irUpperLim);
    printf_P(PSTR("[LOWER] ALS: %u\r\n[UPPER] ALS: %u\r\n"), alsLowerLim,
alsUpperLim);
    /* -------------------------------------------------------- */
    /*         Listen for Sequence Message From Master          */
    /* -------------------------------------------------------- */
    bool nothingReceived = true;
//    char markerSequence[7] = "634215";  // FOR DEBUGGING W/ OUT NRF24L01
        char markerSequence[7];
        while (nothingReceived) {
          checkButtonFlags();
          if (reset) {
            break;
          }
          while (radio.available() ) {
            radio.read(&markerSequence, sizeof(markerSequence) );
            radio.stopListening();
            radio.write(&markerSequence, sizeof(markerSequence));
            radio.startListening();
            nothingReceived = false;
          }
        }
    printf_P(PSTR("\r\n------- Test Marker & Character Sequence -------
\r\n"));
    printf_P(PSTR("[RX] Received MARKER sequence: %s\r\n"),
markerSequence);
    /* -------------------------------------------------------- */
    /*  Break Down Marker Sequence Into Individual Components   */
    /* -------------------------------------------------------- */
    uint32_t number = strtoul(markerSequence, strlen(markerSequence), 10);
// Convert char array to an uint32_t
    printf_P(PSTR("Converted the char sequence to a number: %lu\r\n"),
number);

    uint8_t one = (number / 100000U) % 10;
    uint8_t two = (number / 10000U) % 10;
    uint8_t three = (number / 1000U) % 10;
    uint8_t four = (number / 100U) % 10;
    uint8_t five = (number / 10U) % 10;
    uint8_t six = (number / 1U) % 10;

    uint8_t thisMarker; // This Marker's position in the sequence (1 - 6)
    uint8_t prevMarker;
    uint8_t nextMarker;
```

```
    /* ---------- Determine this markers position & the previous and next
markers in the sequence ---------- */
    if (slaveNumber == one) {
      thisMarker = 1;
      prevMarker = 0; // Master
      nextMarker = two;
    } else if (slaveNumber == two) {
      thisMarker = 2;
      prevMarker = one;
      nextMarker = three;
    } else if (slaveNumber == three) {
      thisMarker = 3;
      prevMarker = two;
      nextMarker = four;
    } else if (slaveNumber == four) {
      thisMarker = 4;
      prevMarker = three;
      nextMarker = five;
    } else if (slaveNumber == five) {
      thisMarker = 5;
      prevMarker = four;
      nextMarker = six;
    } else if (slaveNumber == six) {
      thisMarker = 6;
      prevMarker = five;
      nextMarker = 0;
    }

    /* ---------------------------------------------------------- */
    /*  Listen for Master to Confirm Sending Character Sequence   */
    /* ---------------------------------------------------------- */
    bool waitForNextStep = true;
        while (waitForNextStep) {
          checkButtonFlags();
          if (reset) {
            break;
          }
          char nextStep[3] = "NS";
          char dataReceived[3];
          while (radio.available(&pipeNum)) {
            radio.read(&dataReceived, sizeof(dataReceived));
            Serial.println(dataReceived);
            if (strstr(dataReceived, nextStep) != nullptr) {
              printf_P(PSTR("[RX] Received: %s\r\n"), dataReceived);
              radio.stopListening();
              for (int i = 0; i < 5; i++) {
              radio.write(&nextStep, sizeof(nextStep));
              delay(slaveNumber * 5);
              }
              radio.startListening();
              waitForNextStep = false;
            }
          }
        }
        delay(500);

    /* -------------------------------------------------------- */
    /*    Listen for Character Sequence Message From Master     */
    /* -------------------------------------------------------- */
    nothingReceived = true;
```

```
//    char characterSequence[7] = "kuahsf"; // FOR DEBUGGING W/ OUT
NRF24L01
        char characterSequence[7];
        while (nothingReceived) {
          checkButtonFlags();
          if (reset) {
            break;
          }
          while (radio.available() ) {
            radio.read(&characterSequence, sizeof(characterSequence) );
            if (strlen(characterSequence) == 6) {
            printf_P(PSTR("[RX] Received CHARACTER sequence: %s\r\n"),
characterSequence);
            radio.stopListening();
            radio.write(&characterSequence, sizeof(characterSequence));
            radio.startListening();
            nothingReceived = false;
            }
          }
        }
    printf_P(PSTR("[RX] Received CHARACTER sequence: %s\r\n"),
characterSequence);
    /* ---------------------------------------------------------- */
    /*     Extract the Character this Marker Needs to Display     */
    /* ---------------------------------------------------------- */
    char characterToDisplay;
    uint8_t characterToDisplayInt;
    uint8_t characterSet;

    characterToDisplay = characterSequence[thisMarker - 1]; // Negate 1 as
array is 0 - 5
    printf_P(PSTR("[CHARACTER] Character to display on LED Matrix:
%c\r\n"), characterToDisplay);
    characterToDisplayInt = characterToDisplay;

    if (characterToDisplayInt >= 'A' && characterToDisplayInt <= 'Z') {
      characterToDisplayInt = characterToDisplay - 'A';
      characterSet = UPPER_CHARACTER_SET; // So the correct character array
is accessed
      printf_P(PSTR("[CHARACTER] UPPER character set to display character
index: %u\r\n"), characterToDisplayInt);
    } else if (characterToDisplayInt >= 'a' && characterToDisplayInt <=
'z') {
      characterToDisplayInt = characterToDisplay - 'a';
      characterSet = LOWER_CHARACTER_SET;
      printf_P(PSTR("[CHARACTER] LOWER character set to display character
index: %u\r\n"), characterToDisplayInt);
    } else if (characterToDisplayInt >= '0' && characterToDisplayInt <=
'9') {
      characterToDisplayInt = characterToDisplay - '0';
      characterSet = DIGIT_CHARACTER_SET;
      printf_P(PSTR("[CHARACTER] DIGITS character set to display character
index: %u\r\n"), characterToDisplayInt);
    }

    /* ------------------------------------------------------- */
    /*       Generate a Colour to Display for Stimulus         */
    /* ------------------------------------------------------- */
    uint8_t colourToDisplay = Entropy.random(0, 256);
    printf_P(PSTR("[COLOUR] Hue to display: %u\r\n"), colourToDisplay);
    /* ------------------------------------------------------- */
```

```
    /*            Inform Master this Marker is Ready            */
    /* -------------------------------------------------------- */
        bool readyForTest = false;
        while (!readyForTest) {
          checkButtonFlags();
          if (reset) {
            break;
          }
          char markerReady[3] = "MR";
          char dataReceived[3];
          radio.stopListening();
          //        radio.openWritingPipe(pipes[slaveNumber]);
          radio.write(&markerReady, sizeof(markerReady));
          //        radio.openReadingPipe(0, pipes[0]);
          radio.startListening();
          while (radio.available(&pipeNum)) {
            radio.read(&dataReceived, sizeof(dataReceived));
            if (strstr(dataReceived, markerReady) != nullptr) {
              displayCharacter(RANDOM_CHARACTERS[0], 96, progmem);
              readyForTest = true;
            }
          }
        }
    tone(piezoPin, 800, 150); // [Pin#, Frequency (in Hz), Duration (in
ms)]
    //    tone(speakerPin, 800, 150); // [Pin#, Frequency (in Hz), Duration
(in ms)]
    delay(250);
    tone(piezoPin, 800, 150);
    //    tone(speakerPin, 800, 150); // [Pin#, Frequency (in Hz), Duration
(in ms)]
    delay(2000); // Wait before opening reading pipes with next and
previous markers
    turnLEDsOff();

    /* -------------------------------------------------------- */
    /*        Listen on Previous and Next Marker Pipes          */
    /* -------------------------------------------------------- */
    if (prevMarker > 0) {
      radio.openReadingPipe(PREV_MARKER_PIPE, pipes[prevMarker]); // Open a
reading pipe with the previous Marker
    }
    if (nextMarker > 0) {
      radio.openReadingPipe(NEXT_MARKER_PIPE, pipes[nextMarker]); // Open a
reading pipe with the next Marker to receive Ack payload
    }

    /* ******************** TEST RUNNING ********************* */

    /* -------------------------------------------------------- */
    /*             Wait for Msg from Previous Marker            */
    /* -------------------------------------------------------- */
        nothingReceived = true;
        while (nothingReceived) {
          char dataReceived[3];
          byte pipeNum;
          while (radio.available(&pipeNum)) {
            // Ensure the msg is from the previous Marker (Master if first
in sequence)
            if (prevMarker == 0 && pipeNum == MASTER_PIPE || pipeNum ==
PREV_MARKER_PIPE) {
```

```
              radio.read( &dataReceived, sizeof(dataReceived));
              radio.writeAckPayload(pipeNum, &dataReceived,
sizeof(dataReceived)); // Send Ack payload
              nothingReceived = false;
            }
          }
        }

    /* ------------------------------------------------------- */
    /*      Start Timer + Present Audio & Visual Stimulus       */
    /* ------------------------------------------------------- */
    if (characterSet == UPPER_CHARACTER_SET) {
      displayCharacter(LETTERS_UPPER[characterToDisplayInt],
colourToDisplay, progmem);
    } else if (characterSet == LOWER_CHARACTER_SET) {
      displayCharacter(LETTERS_LOWER[characterToDisplayInt],
colourToDisplay, progmem);
    } else if (characterSet == DIGIT_CHARACTER_SET) {
      displayCharacter(DIGITS[characterToDisplayInt], colourToDisplay,
progmem);
    }
    testTime.start();
        tone(piezoPin, 700, 600);
    //turntone(speakerPin, 700, 1000);
    /* ------------------------------------------------------- */
    /*   Read Sensor Data to Determine When Ball is Removed     */
    /* ------------------------------------------------------- */
    bool showTime = true;
    bool ballRemoved = false;
    bool proxOutsideLim, irOutsideLim, alsOutsideLim;
    printf_P(PSTR("\r\nWaiting for ball to be removed...\r\n"));
    //    time.start();
    while (!ballRemoved) {
      proxOutsideLim = false;
      irOutsideLim = false;
      alsOutsideLim = false;

      if (useProx0) {
        prox = sensor.readProx();
        proxRunningAverage = calcRunningAverage(&prox, proxRunningAvgTotal,
proxValues);
        if (proxRunningAverage > proxUpperLim || proxRunningAverage <
proxLowerLim) {
          proxOutsideLim = true;
          //          printf_P(PSTR("ProxOutsideLim\r\n"));
        }
      } else if (useProx1) {
        _prox = sensor.readProxOther();
        _proxRunningAverage = calcRunningAverage(&_prox,
_proxRunningAvgTotal, _proxValues);
        if (_proxRunningAverage > proxUpperLim || _proxRunningAverage <
proxLowerLim) {
          proxOutsideLim = true;
          //          printf_P(PSTR("ProxOutsideLim\r\n"));
        }
      }
      if (useIR0) {
        ir = sensor.readIR();
        irRunningAverage = calcRunningAverage(&ir, irRunningAvgTotal,
irValues);
```

```
        if (irRunningAverage > irUpperLim || irRunningAverage < irLowerLim)
{
          irOutsideLim = true;
          //          printf_P(PSTR("IROutsideLim\r\n"));
        }
      } else if (useIR1) {
        _ir = sensor.readIROther();
        _irRunningAverage = calcRunningAverage(&_ir, _irRunningAvgTotal,
_irValues);
        if (_irRunningAverage > irUpperLim || _irRunningAverage <
irLowerLim) {
          irOutsideLim = true;
          //          printf_P(PSTR("IROutsideLim\r\n"));
        }
      }
      if (useALS0) {
        als = sensor.readVisible();
        alsRunningAverage = calcRunningAverage(&als, alsRunningAvgTotal,
alsValues);
        if (alsRunningAverage > alsUpperLim || alsRunningAverage <
alsLowerLim) {
          alsOutsideLim = true;
          //          printf_P(PSTR("alsOutsideLim\r\n"));
        }
      } else if (useALS1) {
        _als = sensor.readVisibleOther();
        _alsRunningAverage = calcRunningAverage(&_als, _alsRunningAvgTotal,
_alsValues);
        if (_alsRunningAverage > alsUpperLim || _alsRunningAverage <
alsLowerLim) {
          alsOutsideLim = true;
          //          printf_P(PSTR("alsOutsideLim\r\n"));
        }
      }

      incrementRunningAvgIndex();
      /* ---------- Check if ball has been removed ---------- */
      if (useProx0 && useIR0 && useALS0) {
        if (proxOutsideLim && irOutsideLim || proxOutsideLim &&
alsOutsideLim) {
          ballRemoved = true;
          printf_P(PSTR("BALL REMOVED!\r\n"));
          //        time.stop();
          //        if (showTime) {
          //          printf_P(PSTR("Time taken to detect: %lu
microseconds\r\n"), time.value());
          //          showTime = false;
          //        }
        }
      } else if (useProx0 && useIR0) {
        if (proxOutsideLim && irOutsideLim) {
          ballRemoved = true;
          printf_P(PSTR("BALL REMOVED!\r\n"));
        }
      } else if (useProx0) {
        if (proxOutsideLim) {
          ballRemoved = true;
          printf_P(PSTR("BALL REMOVED!\r\n"));
        }
      }
    }
```

```
    testTime.stop();
    tone(piezoPin, 700, 1000); // Debug feedback
    turnLEDsOff();
    printf_P(PSTR("Test time: %lu microseconds\r\n"), testTime.value());

    /* -------------------------------------------------------- */
    /*          Send Msg For Next Marker to Activate            */
    /* -------------------------------------------------------- */
    nothingReceived = false;
    while (!nothingReceived) {
      char dataToSend[3] = "GO";
      char dataReceived[3];
      radio.stopListening();
      radio.write(&dataToSend, sizeof(dataToSend));
      radio.startListening();
      while (radio.available(&pipeNum)) {
        if (pipeNum == NEXT_MARKER_PIPE) {
          radio.read(&dataReceived, sizeof(dataReceived));
          if (strstr(dataReceived, dataToSend) != nullptr) {
            nothingReceived = true;
          }
        }
      }
    }
    /* ---------- Next Marker has acknowledged deactivation ---------- */

    /* -------------------------------------------------------- */
    /*                   Send Time To Master                    */
    /* -------------------------------------------------------- */
    uint32_t t = testTime.value();
    nothingReceived = true;
    while (nothingReceived) {
      uint32_t dataReceived;
      radio.stopListening();
      radio.write(&t, sizeof(t));
      radio.startListening();
      while (radio.available(&pipeNum)) {
        if (pipeNum == MASTER_PIPE) {
          nothingReceived = false;
        }
      }
      delay(25);
    }

    /* ******************** This Marker's Role in the Test is COMPLETE
********************* */
    testTime.reset();
    ballOnMarker = false;
    delay(3000);
  } // while (runningTests)
} // void loop()

/* ##################################################### */
/*                       Functions                       */
/* ##################################################### */

/* ---------------------------------------------------------------------
   Function that assists in displaying a linear-like brightness adjustment
of LEDs.
```

```
   Due to the non-linearity of the eye's response to light, adjusting the
PWM
   duty cycle of the LEDs does not adjust the perceived brightness
linearly.

   Using an Excel generated trendline of a standard full-stop f-number
scale, the percentage
   value can be calculated into the corresponding PWM duty cycle value.
   -------------------------------------------------------------------- */
uint8_t percentToPWM(float percentage) {
  return round((exp(0.055451774445  * percentage)) - 1);
}

/* --------------------------------------------------------------------
   Function used to determine an LED pixel index value.

   When supplied with a set of coordinate values, this function returns the
   index for the corresponding pixel of a 2D LED matrix.

   XY(x,y) takes x (width) & y (height) coordinates:
        e.g.  leds[ XY(x,y) ] = CRGB(red, green, blue) OR CRGB::Red;
          OR  leds[ XY(x,y) ] = CHSV(hue, saturation, brightness value);
   -------------------------------------------------------------------- */
uint16_t XY( uint8_t x, uint8_t y)
{
  uint16_t i;
  if (x < 8) { // First matrix
    i = (y * WIDTH) + x;
  } else  { // Second matrix
    i = (y * WIDTH) + x + 56;
  }
  if (y > 0) {
    i -= 8 * y; // To maintain LED order, negate 8 * y from the index so it
does not "skip" rows
  }
  i = 127 - i; // Inverse the LED index since the "first" pixel is at the
bottom-right of the matrix
  return i;
}

/* --------------------------------------------------------------------
   Display a specificed character onto the LED matrix displays.

   This function reads a const char[64] array, and if the char is a '1',
the LED will be turned ON;
   if '0', the LED is turned OFF. (LED pixels range from 0 - 127)

   The array can be generated using the ConvertHexToMatrixCharacter.cpp
file (by Reuben Smith).

   By Using CHSV, various colours can be generated with the same
brightness, simply
   by changing the hue variable, which MUST be between 0 and 255 (bounds
checking is completed).

   If char arrays x[] is stored in PROGMEM (isProgmem = true), access each
char using pgm_read_byte_near(x + i).
   -------------------------------------------------------------------- */
void displayCharacter(const char x[], uint8_t hue, bool isProgmem)
{
  FastLED.clear();
```

```
  if (hue < 0) {
    hue = 0;
  } else if (hue > 255) {
    hue = 255;
  }
  for (uint8_t i = 0; i < 64; i++) {
    char value;
    if (isProgmem) {
      //Serial.print((char)pgm_read_byte_near(x + i));
      value = pgm_read_byte_near(x + i); // Read the data from PROGMEM.
    } else {
      value = x[i];
    }
    if (value == '1') {
      leds[127 - i] = CHSV( hue, 255, 255);
      // Display character on chained array
      leds[127 - i - 64] = CHSV( hue, 255, 255);
    }
  }
  FastLED.show();
  if (isProgmem) {
    strcpy_P(currentCharacter, x);
    //  printf_P(PSTR("%s\r\n"), currentCharacter);
  }
  else {
    strcpy(currentCharacter, x);
    //    printf_P(PSTR("%s\r\n"), currentCharacter);
  }
  currentHue = hue;
}

/* ---------------------------------------------------------------------
   Clear the LED Matrix so nothing is showing and update CurrentCharacter.
    --------------------------------------------------------------------- */
void turnLEDsOff() {
  memset(&currentCharacter[0], '0', sizeof(currentCharacter) - 1);
  FastLED.clear();
  FastLED.show();
}
/* ---------------------------------------------------------------------
   Button module interrupt function.

   This function is called every time the button module state CHANGES (is
pressed or released).
   The function checks the time since last press to ensure no extra
triggers occur due to debounce.

 *  *** NOTE: because a pull-up resistor has been placed between VCC and
the button signal, the
              button is normally HIGH and becomes LOW when pressed.
   -------------------------------------------------------------------- */
void buttonISR() {
  sleep_disable(); //Disable sleep mode
  bool state = digitalRead(buttonPin);
  if (state == LOW) { // If the button is pressed
    buttonISRHigh = true;
    if (buttonTime.state() == StopWatch::RESET || buttonTime.state() ==
StopWatch::STOPPED)  {
      buttonTime.reset();
      buttonTime.start();
    }
```

```
  }
  else {                  // If the button is released
    buttonTime.stop();
    uint32_t t = buttonTime.value();
    if (t > DEBOUNCE_DELAY && t < LONG_PRESS) { //Legitimate button press
(not a debounce)
      buttonISRLow = true;
      //printf_P(PSTR("Time since button press: %lu\r\n"), t);
    } else {
      buttonISRHigh = false;
      //      Serial.println(F("Button debounce occurred!")); // Do nothing
    }
  }
}

/* -------------------------------------------------------------------
   Button module interrupt flag checking function

   This function checks the status of the button ISR HIGH & LOW booleans to
see
   if the button was pressed or released. If it was, then process the
necessary code here.
   ------------------------------------------------------------------- */
void checkButtonFlags() {
  if (buttonTime.elapsed() >= LONG_PRESS) {
    displayCharacter(RANDOM_CHARACTERS[2], 0, progmem);
    reset = true;
  }
  if (buttonISRHigh) {
    //    printf_P(PSTR("%s\r\n"), currentCharacter);
    strcpy(prevCharacter, currentCharacter);
    prevHue = currentHue;
    displayCharacter(RANDOM_CHARACTERS[2], 168, progmem);
    tone(piezoPin, 800, 150);
    buttonISRHigh = false;
  } else if (buttonISRLow) {
    //printf_P(PSTR("%s\r\n"), prevCharacter);
    memset(&prevCharacter[0], '0', sizeof(prevCharacter) - 1);
    displayCharacter(prevCharacter, prevHue, !progmem);
    printf_P(PSTR("Time since button press: %lu ms\r\n"),
buttonTime.value());
    //    ballOnMarker = true;
    buttonISRLow = false;
  }
}

/* -------------------------------------------------------------------
   Function that calculates and returns the corresponding running average,
   but also alters the total variable and stores to the values array.

   The Si1145 sensor will generally give a readings that can be quite
noisy;
   so calculating a running average means that the average of the reading
   measurements is now smoothed and produces less noisy data.

   Choosing the right RUNNING_AVG_NUM (array size) is important, as there
   is a trade off between noisy data and responsiveness of the sensor.
   ------------------------------------------------------------------- */
uint16_t calcRunningAverage(uint16_t *value, uint32_t &total, uint16_t
values[]) {
  total -= values[runningAvgIndex];
```

```
  values[runningAvgIndex] = *value;
  total += values[runningAvgIndex];
  return (total / RUNNING_AVG_NUM);
}

/* ----------------------------------------------------------------------
   Function that increments the running average index.

   This is called because multiple running average calculations need to be
conducted
   on the prox, ir and als values, so to maintain the same index position
for each
   array to avoid confusion.
   ---------------------------------------------------------------------- */
void incrementRunningAvgIndex() {
  runningAvgIndex++;
  if (runningAvgIndex >= RUNNING_AVG_NUM) {
    runningAvgIndex = 0; // Wrap around to initial circular linked list
value
  }
}

/* ----------------------------------------------------------------------
   Function that calculates the sensor limits based on the average value
obtained.

   The upper and lower limits are calculated based on the limitMultiplier,
which will add
   or subtract a percentage of the average value.
   ---------------------------------------------------------------------- */
void calculateSensorLimits(uint16_t &average, uint16_t &upper, uint16_t
&lower) {
  uint32_t tempCalc; // Use 32-bit int in case the value exceeds uint16_t
maximum value
  tempCalc = average + (average * limitMultiplier);
  if (tempCalc > UINT16_MAX) {
    upper = UINT16_MAX;
  } else {
    upper = tempCalc;
  }
  tempCalc = average - (average * limitMultiplier);
  if (tempCalc < 0) {
    lower = 0;
  } else {
    lower = tempCalc;
  }
}

/* ----------------------------------------------------------------------
   Put Arduino to sleep function.

   This function is called when putting the Arduino into a power saving
sleep mode
   (SLEEP_MODE_PWR_DOWN -> most power saving), which should use
approximately 10mA or even less.

   Since Entropy uses Watch Dog Timers (WDTs) in order to generate random
values, these interrupts
   cause the Arduino to wake, therefore, it is important to disable the
WDTs before sleeping.
```

```
   To wake the Arduino, the button needs to be LOW (normally HIGH as a
pull-up
   resistor has been used between VCC and button), which will trigger the
interrupt wakeArduino ISR.
   -------------------------------------------------------------------- */
void putArduinoToSleep() {
  printf_P(PSTR("Putting Arduino to sleep...\r\n"));
  Serial.flush();
  set_sleep_mode(SLEEP_MODE_PWR_DOWN); //Set sleep mode to full sleep
  sleep_enable();
  detachInterrupt(digitalPinToInterrupt(buttonPin)); // Detach buttonISR
from button press
  attachInterrupt(digitalPinToInterrupt(buttonPin), wakeArduinoISR, LOW);
// Attach wakeArduinoISR to button press
  power_all_disable(); // Disable all modules
  wdt_disable(); // Disable the Watch Dog Timers ran by Entropy
  sleep_cpu();// Activate sleep mode
  /* ---------- Arduino is now asleep! Press button to wake it up ---------
- */
  printf_P(PSTR("Arduino woke up!\r\n"));
  tone(piezoPin, 800, 150);
}

/* --------------------------------------------------------------------
   Wake Arduino interrupt function.t

   This ISR is called when the button is pressed during sleep mode.
   Once called, sleep mode is immediately disabled and this ISR is detached
so that it is not
   continuously called while the button is pressed.

   The Watch Dog Timer is reset so that the WDTs ran by Entropy can
continue to generate
   random values.

   In order to maintain functionality of the button during normal use, the
normal button ISR is
   attached only when the Arduino is not sleep.
   -------------------------------------------------------------------- */
void wakeArduinoISR() {
  sleep_disable();
  power_all_enable(); // Re-enable all modules
  wdt_reset(); // Reset the Watch Dog Timers ran by Entropy to continue
generating random values
  detachInterrupt(digitalPinToInterrupt(buttonPin)); // Detach this ISR
from the button press
  attachInterrupt(digitalPinToInterrupt(buttonPin), buttonISR, CHANGE); //
Attach buttonISR to button press
}

/* --------------------------------------------------------------------
   Check the available SRAM on the Arduino.

   This function returns the approximately available SRAM left on the
device.
   **Function obtained from
https://playground.arduino.cc/Code/AvailableMemory**
   -------------------------------------------------------------------- */
int checkFreeRAM() {
  extern int __heap_start, *__brkval;
  int v;
```

```
  return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}
```

Appendix K

K.1. Bill of Materials

Table K- 1: Final Bill of Materials for the final prototype assemblies

Part/Item	Supplier	CAT. #	Link	#	Unit Price	Shipping	Subtotal
Prototype Shield Mega for Arduino Mega	RobotDyn	6970622930488	https://robotdyn.com/prototype-shield-mega-for-arduino-mega.html	7	$4.68	$55.72	$88.48
Arduino Mega 2560 CH340G/ATmega2560-16AU	RobotDyn	MEGA-CH340G/ATmega2560	https://robotdyn.com/mega-2560-ch340g-atmega2560-16au.html	7	$11.06	As above	$77.42
LM2596 DC-DC Step-down Adjustable Power Supply Module	RobotDyn	6970622932031	https://robotdyn.com/lm2596-dc-dc-step-down-adjustable-power-supply-module-in-3-36-out-1-5-34v-3a.html	6	$2.14	As above	$12.84
Button Switch Module-Black	RobotDyn	6970622931393	https://robotdyn.com/button-switch-module-black.html	7	$1.00	As above	$7.00
Socket adapter for NRF24L01, with regulator 3.3V	RobotDyn	6970622932055	https://robotdyn.com/catalog/power/socket-adapter-for-nrf24l01-with-regulator-3-3v.html	7	$0.88	As above	$6.16
Bluetooth 4.0 HM-10 Master Slave Module	Little Bird Electronics	SU-HM-10	https://www.littlebirdelectronics.com.au/bluetooth-4.0-hm-10-master-slave-module	1	$25.65	$7.20	$32.85
Slip Ring with Flange - 22m diameter, 6 wires, max 240V @ 2A	Little Bird Electronics	AF-736	https://www.littlebirdelectronics.com.au/slip-ring-with-flange-22mm-diameter-6-wires-max-24	1	$21.87	As above	$21.87
Stepper Motor: Bipolar, 200 Steps/Rev, 42x38mm, 2.8V, 1.7 A/Phase	Little Bird Electronics	PL-2267	https://www.littlebirdelectronics.com.au/stepper-motor-bipolar-200-steps-rev-42x38mm-2.8v-1	1	$27.40	As above	$27.40
Speaker - 3 Diameter - 4 Ohm 3 Watt	Core Electronics	ADA1314	https://core-electronics.com.au/speaker-3-diameter-4-ohm-3-watt.html	6	$3.50	$0.00	$21.00
Adafruit Mono 2.5W Class D Audio Amplifier - PAM8302	Core Electronics	ADA2130	https://core-electronics.com.au/mono-2-5w-class-d-audio-amplifier-pam8302.html	6	$5.77	$0.00	$34.62

Piezo Speaker - PC Mount 12mm 2.048kHz	Core Electronics	COM-07950	https://core-electronics.com.au/piezo-speaker-pc-mount-12mm-2-048khz.html	7	$2.02	$0.00	$14.14
DRV8834 Low-Voltage Stepper Motor Driver Carrier	Core Electronics	POLOLU-2134	https://core-electronics.com.au/drv8834-low-voltage-stepper-motor-driver-carrier.html	1	$7.99	$0.00	$7.99
Garmin LIDAR-Lite v3HP	Johnny Appleseed	010-01722-10	https://www.ja-gps.com.au/Garmin/lidar-lite-v3hp/?utm_source=myshopping&utm_medium=cpc&utm_campaign=GPS+Accessories&utm_term=Garmin+LIDAR+Lite+v3HP	1	$229.00	$0.00	$229.00
WS2812 LED 5050 RGB 8x8 64 LED Matrix [65mmx65mm]	XM (ShenZhen) Electronic Trade Co.,Ltd (AliExpress)	N/A	https://www.aliexpress.com/store/product/WS2812-LED-5050-RGB-8x8-64-LED-Matrix-for-Arduino/1095279_32600043941.html?spm=2114.12010615.8148356.1.68384538vMFV2U	12	$6.65	$19.81	$99.61
SEN-36002: Silicon Labs Si1145 UV / Ambient Light / Proximity sensor, 3.0V-5V	Playing with Fusion (on eBay)	SEN-36002	https://www.ebay.com.au/itm/Silicon-Labs-Si1145-UV-Amibent-Light-Proximity-sensor-3-0V-5V-/232166755863#shpCntId	6	$16.13	$63.93	$160.71
TCST2103 - Transmissive Photo Interrupter, Phototransistor, Through Hole, 3.1 mm, 1 mm, 60 mA, 6 V	Element14	1470060	http://au.element14.com/vishay/tcst2103/sensor-optical-phototransistor/dp/1470060?ost=TCST2103&ICID=redirect-Y&CMP=os-geobanner-google	1	$2.84	$0.00	$2.84
COLLAR, STEEL, 1PC, 8MM	Element14	3471615	http://au.element14.com/huco/046101008far/collar-steel-1pc-8mm/dp/3471615?st=collar	1	$2.94	$0.00	$2.94
Transparent 4 Inch CCTV Replacement Acrylic Clear Camera Dome Protector Housing	Be in Control Store (AliExpress)	N/A	https://www.aliexpress.com/item/Transparent-4-Inch-Indoor-Outdoor-CCTV-Replacement-Acrylic-Clear-Camera-Dome-Protector-Housing-Surveillance-Cameras-Accessories/32508089534.html?spm=a2g0s.9042311.0.0.17e44c4dZHYiYG	1	$9.66	$24.23	$37.28
Arduino Compatible RGB LED Module	Jaycar	XC4428	https://www.jaycar.com.au/arduino-compatible-rgb-led-module/p/XC4428	1	$4.95	$0.00	$4.95

Arduino Compatible Red Laser Diode Module	**Jaycar**	**XC4490**	https://www.jaycar.com.au/arduino-compatible-red-laser-diode-module/p/XC4490	1	$4.95	$0.00	$4.95
100 PCS 10*3mm Self Adhesive Black Anti Slip Silicone Bumper Pads	**Shida (AliExpress)**	**N/A**	https://www.aliexpress.com/item/100-PCS-10-3mm-Self-Adhesive-Black-Anti-Slip-Silicone-Furniture-Bumper-Pads-Flat-Rubber-Feet/32611172066.html?spm=a2g0s.9042311.0.0.4eb34c4dUwxpKZ	1	$8.70	$26.74	$38.98
12V Metal Latching Black 16mm Waterproof Power Push Button Switch – w/ LED Light	**EARU AUTO (AliExpress)**	**N/A**	https://www.aliexpress.com/item/5V-12V-24V-110V-220V-LED-Locking-Latching-16mm-Waterproof-Car-Atuo-Power-Dash-Metal-Push/32813110023.html?spm=a2g0s.9042311.0.0.97dc4c4di0cdl4	7	$3.19	$32.38	$60.18
2x-NRF24L01-PA-LNA-Wireless-Transceiver-Communication-Module-2-4G-Antenna-TE862	**eshop1may: eBay**	**N/A**	https://www.ebay.com.au/itm/2x-NRF24L01-PA-LNA-Wireless-Transceiver-Communication-Module-2-4G-Antenna-TE862/263778367671?ssPageName=STRK%3AMEBIDX%3AIT&_trksid=p2057872.m2749.l2649	4	$10.78	$0.00	$47.43
Rust-Oleum 340g Ultra Cover 2X Satin Spray Paint - Canyon Black	**Bunnings**	**1580677**	https://www.bunnings.com.au/rust-oleum-340g-ultra-cover-2x-satin-spray-paint-canyon-black_p1580677	4	$11.89	$0.00	$47.56
Rust-Oleum 298g Satin Clear 2X Ultra Cover Spray Paint	**Bunnings**	**1580817**	https://www.bunnings.com.au/rust-oleum-298g-satin-clear-2x-ultra-cover-spray-paint_p1580817	1	$11.89	$0.00	$11.89
KAN Battery 5000mah 7.2v Stick Pack Tamiya Plug	**Hobby Habit**	**KAN-5000-3**	https://www.hobbyhabit.com.au/KAN-5000-3.html	7	$49.95	$0.00	$349.65
Football	**Kmart**	**N/A**	**N/A**	3	$19.00	$0.00	$57.00
Netball	**Kmart**	**N/A**	**N/A**	1	$5.00	$0.00	$5.00
Basketball	**Kmart**	**N/A**	**N/A**	1	$5.00	$0.00	$5.00
Rugby	**Kmart**	**N/A**	**N/A**	1	$5.00	$0.00	$5.00
						Total:	**$1,521.75**

9. Erratum Appendix

The author would like to take this opportunity to acknowledge that after submission of this thesis, further development was completed which solved the issue regarding communication between the radio frequency hardware (nRF24L01+ modules). This meant that intra-system communication was able to be successfully implemented, allowing both the master and slave devices to send and receive data with one another during the test. Thus, the test was able to be conducted by the system, activating each marker and presenting the stimuili which allowed an individual to deactivate a marker, thereby reactivating the next marker in the sequence. The marker and character sequence were confirmed to be random for each test performed. The time taken to remove the sport specific equipment from each individual marker was recorded and sent to the connected mobile phone or laptop, as well as a total time. Further development and refinements of the system is now being undertaken, as mentioned in Section 5: Future Work.