Design and development of a low-cost high-performance vehicle mounted UHF RFID system for tracking goods and inventory

Gareth Hoskins

213523380

In fulfilment of the degree of Master of Science in Mechatronic Engineering at the School of Mechanical Engineering, University of KwaZulu-Natal.

October 2019

As the candidate's supervisors, we agree to the submission of this thesis.
Supervisors: Prof. Glen Bright; Dr. Jared Padayachee

DECLARATION 1 – PLAGIARISM

I, Gareth Oregan Hoskins declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.

2. This thesis has not been submitted for any degree or examination at any other university.

3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
    a. Their words have been re-written but the general information attributed to them has been referenced;
    b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.

5. This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed:

………………………………………………………………………………………

DECLARATION 2 - PUBLICATIONS

DETAILS OF CONTRIBUTION TO PUBLICATIONS that form part and/or include research presented in this thesis (include publications in preparation, submitted, *in press* and published and give details of the contributions of each author to the experimental work and writing of each publication)

Publication 1

> 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), Bloemfontein, South Africa, "Human-Robot Interaction: The Safety Challenge (An integrated frame work for human safety)", by G.O Hoskins, Prof G. Bright, Dr J. Padayachee

Publication 2

> The South African Journal of Industrial Engineering (SAJIE), "Robot safety: robotic serial arm detects human presence using proximity sensors and a Bluetooth mobile application", by G.O. Hoskins, Prof G. Bright, Dr J. Padayachee

Signed:

………………………………………………………………………………………

ACKNOWLEDGMENTS

"Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning." – Albert Einstein

"Speak reasonably, and reasonable men will listen." – Maria Thorpe, Assassins Creed "The Secret Crusade"

I wish to thank Prof. Glen Bright and Dr. Jared Padayachee for their support and supervision in the completion of this study.

I am indebted to my family, Mr. Oregan Percival Mark Hoskins, Dr. Geraldine Ann Hoskins, Mrs. Nicole Megan Barlow, and Mr. Nicholas Luke Barlow, for their continuous support and encouragement throughout.

I would also like to thank Miss Nirvana Naidu for providing me with continuous moral and emotional support.

I extend my thanks to Mr. Brandon Robert Lotter for the years of friendship that led to my advancement within the engineering field.

Lastly, I would like to thank Mr. Mark Brian Govender for sponsoring the experimental equipment required for this study, as well as the support.

Abstract

Automated manufacturing has become a crucial part of the industrial sector however, the inclusion of humans into a production line remains a necessity in certain scenarios. Human-robot interaction involves the implementation of both robots and humans into a single action. Unfortunately, this has lead to many industrial accidents taking place. Studies show that a large population of robot related injuries are inflicted upon the human operators. The aim of this project is to design and develop a flexible safety system for the operator which promotes human-robot collaboration, as well as a system that complies with robot safety standards.

Human-robot interaction is a vastly wide field of study, and only individual scenarios may be analysed. This has lead to a number of safety systems developed for this application. One of the most notable robot safety systems is the ABB SafeMove, which involves the use of a human proximity sensor that directly dictates the actions of the robot in operation. The proposed safety system in this project is mainly inspired by this design.

The proposed system involves the incorporation of safety layers placed around the robotic system. Each layer reacts specifically when a human enters the monitored zone. The layers then convey various signals to the robot in operation, where it will adjust itself accordingly. To test the proposed system, the construction of a small scale test rig was performed. The test rig allowed the experimentation of various electronic components to be easily interchanged to optimise the safety of the user. One of the layers included the design of a mobile application, which provides the user with a personal sense of identification towards the robot.

A list of experimental troubleshooting and testing was constructed and should be used for further experimentation when utilizing these components. Overall, the core idea of the study proved successful, and can be suggested that it is incorporated into a full-scale manufacturing scenario.

Acronyms

DOT	Degrees of Freedom
HRI	Human Robot Interaction
SIS	Safety Integrated System
MIT AI 2	Massachusetts Institute of Technology Application Inventor 2
ABB	ASEA Brown Boveri
PLC	Programmable Logic Controller
RFID	Radio Frequency Identification
VCC	Voltage Common Collector
GND	Ground
TXD	Transmitter
RXD	Receiver
IDE	Integrated Development Environment
PLA	Polylactic Acid Filament

Contents

Table of Figures

List of Tables

Chapter 1

Introduction

1.1 Background Information

Automated manufacturing has become a crucial part of the industrial sector. The use of robots is found in a wide array of industries, which can be as complex as automobile manufacturers, right down to the production of food products. Due to an automated system's ability to perform accurate, high-speed repeated actions, the efficiency of a manufacturing line is maximised. However, manufacturing robots lack the ability of immediate adaptation to a certain situation, without requiring specific programming. These robots also lack the specific creativity, which can be required during a manufacturing process. Unlike humans, robots are limited by their movement's degrees of freedom (DOF), the design of their end effectors as well as the programming of their controller and environment-sensory capabilities [1].

Fully automated lines, without a doubt, possess the highest production efficiency. The need for human involvement has become scarce over time, especially in developed countries. Although line efficiency is maximised, this may not be in the best interest from an economical point of view in developing countries. The inclusion of human workers in a production line is a necessity for improving a production line's flexibility. This has resulted in the requirement for 'Human-Robot Interaction (HRI)' to increase. The crux of this study revolves around the promotion of HRI.

Automation is directly proportional to the line production. Unfortunately, due to a robot's high operational speed, various hazards start to arise in terms of human health and safety. Most robots and robotic equipment are contained within a physical barrier, to completely separate humans and robots within a working space [2]. While this eliminates the health and safety risks of the human workers, it compromises the ability of robots and humans to work side by side. While these fences or cages prevent any harm to human workers, they also require a larger amount of floor space, financial cost and prevent most forms of HRI. However, a balance can exist by combining the high-speed repetition of a robot, with the creativity of a human worker to maximise production efficiency.

1.2 Topic Description

HRI can simply be described as the collaboration between robot and man, to achieve a common goal. It incorporates the major skill set of both parties, in an attempt to achieve maximum possible line efficiency. The main concern of this practice is the health and safety of the human workers involved. The conventional solution is to place the robotic system within a physical barrier. This, however, limits HRI. The aim of this study is investigate the possibilities of removing these physical barriers, and implementing a different form of a control/security system to increase the capabilities of HRI [3].

The primary concern is both the physical health and safety of the workers involved, while performing various duties. These duties can range from scheduled maintenance on the robotic system, to being involved in the physical manufacturing process. This study proposes a system that must be able to identify each individual worker in a typical industrial plant. Every worker must be given a specific set of privileges that will determine which actions he or she may perform on a robotic system. These privileges are determined by the worker's experience and their specific job role. While the system may allow certain workers to proceed into the robotic area, various fail-safes are to be implemented in the event of unauthorized entry. This system must allow HRI to take place in various forms, as well as minimise any hazards to physical and psychological health and safety.

The system includes the use a mobile application that serves as a form of personal identification for every human on the factory floor. The application can be customized for a specific manufacturing company. While controlled through a mainframe, each worker's activity can be monitored. Activities that occur within a robots workspace will be recorded and can therefore be monitored by those in charge.

1.3 Research Question

The topic discussed raises the following questions: **It is possible to employ a flexible integrated safety system (SIS), which removes the physical boundaries placed around a robotic system, to promote HRI, while maintaining a safe working environment for the human involved?** Similarly: **Can the robot system identify each individual human that approaches the work area, and react accordingly?**

Upon further investigation, the following sub-questions were raised in order to structure the study:

- Can a test rig be conducted to simulate a proximity security system, using low cost sensors and actuators?
- Can a 3rd party mobile application be developed to serve as a form of personal identification, to enable the robot to identify each human that approaches the assembly zone?
- Does the proposed system provide a satisfactory safety standard when compared to a full scale scenario?
- What are the limitations of the proposed system?

The significance of solving the above scenarios aids in covering part of the extremely broad field of HRI. The study will also provide an in depth tutorial on how to write a mobile application for robot controlling purposes. The simulation can serve as a new idea on how to produce an efficient robot security system. The mobile application can be customized for a specific manufacturing company, depending on the requirements. This would enable the company to keep track of every factory floor worker, and what jobs they are performing at certain times.

1.4 Aims & Objectives

The aim of this research was to investigate the possibility, and practicality, of implementing a sensory safety system, with the aid of a mobile application, around a piece of automated machinery in a general advanced manufacturing assembly line. The safety must also promote human-robot collaboration (HRI), as well as comply with regular robotic safety standards. To investigate the possibilities of the mentioned scenario, the following objectives are listed in order to structure the study performed:

- Research of the current safety protocols around robotics in industry
- Develop a test rig in order to perform experimental methods
- Research various electronic proximity security/monitoring systems available for the required application
- Develop a sensory network and coding for the computational system
- Research the capabilities of a mobile application communicating with a robot controller
- Develop a mobile application (Android based) for the sole purpose of communication with the robot controller
- Install the sensory system as well as the mobile application onto the test rig
- Test and analyse an experiment involving a combination of the mobile application and a proximity sensor
- Refine/Optimise the security system to avoid any access to loopholes
- Optimise the mobile application layout and user interface

Certain objectives may overlap one another, such as the installation and experimentation processes.

1.5 Project Limitations and Assumptions

For the purposes of this study, a safety-integrated-system (SIS) will be developed. The main controller used will be a standard Arduino. The SIS is comprised of various 3rd party sensors and actuators, and will be installed directly onto a robotic controller. The robotic system must also utilize an Arduino as the main controller, to avoid any electronic communicational complications. This study will purely focus on the human aspect of HRI, and will only yield basic robotic responses as a result. Advance kinematics and robotic path planning algorithms can be substituted for the basic robotic response, but will not be covered in this study.

The development of the SIS will cover 2 main aspects. These aspects will include the design of the physical controller unit, as well as the design and construction of the mobile application. The controller unit must be able to communicate with the mobile application. Various human-detection sensors will be directly wired to the controller. After which, the controller will be directly installed onto the robotic system, where it will issue standard operational instructions, as well as keep the system in a safe state.

The experimental procedure will mainly focus on returning a basic robotic reaction. The robotic system should immediately enter a safe state, should unauthorized personnel enter the monitored area. The mobile application should allow the authorized personnel to manipulate the robotic system accordingly. Potential hazards will be assessed, as well as the potential hazard reduction. Once the basic robotic responses prove to be successful, they can then be substituted with more complex path-planning algorithms.

1.6 Dissertation Chapter Overviews

The following sections briefly describe the layout of the dissertation, and provide a short overview of each chapter.

1.6.1 Literature Review

This chapter will focus on the research performed during the study. The main areas of focus will be on the areas of interest in the topic of HRI, to determine in which area this study will belong. Current industrial systems will be analyzed, as well as a methodology of hazard identification in a specific robotic system. Current safety systems will be assessed and compared. Various proximity sensors will then be investigated in order to identify the most appropriate ones to be utilized.

1.6.2 Proposed System Overview

The chapter will provide the basis for the entire study. The study involves a hypothetical scenario that can be applied to a full-scale manufacturing process. The scenario will then be modelled into a conceptual design for the safety system to operate within. A robotic arm test was required in order to install the safety system onto. It was then described how the robotic arm was designed and assembled using various 3D printed components. This robot provided the mechanical system in the mechatronics model.

1.6.3 Electronic Design

This chapter will cover the physical and software design of the SIS’s controller unit. An in depth description of how to wire the Bluetooth module onto the Arduino will be given. Special care must be given to the specific wiring of the units, as experimental methods have lead to damages of various components. The network of sensors and actuators provided the electronic system in the mechatronics model.

1.6.4 Arduino Controller Programming

This chapter will focus on the design of the computational system. Once the sensors have been connected to the controller, the controller is then paced onto the robotic system. A process flow diagram will explain the order of coding the controller is expected to follow. The specific coding used for experimental purposes can be found in the appendix. The coding for the Arduino provided the computer system in the mechatronics model.

1.6.5 Mobile Application Development

This chapter will cover the design of the mobile application. The basic application layout is shown. An explanation on how the application will communicate with the Arduino is given, through the form of serial communication. The coding of the application will be done in the form of a block diagram, used in MIT AI2. The application provided the control system for the mechatronics model.

1.6.6 Experimental Troubleshooting & Testing

This chapter discusses the various errors and optimisation methods involved during the physical construction of the system. Individual pieces of equipment were added to the system separately and tested. The components were then synchronised to run as one unit. Various pieces of electronic components were damaged during the testing phase. The errors in method were isolated and should be avoided in future experiments.

1.6.7 Discussion &Validations

This chapter will examine each individual research question raised in the aims and objectives. Each question will be discussed with respect to the overall study. Thereafter, the main problem statement will be examined and concluded.

1.6.8 Conclusion

This chapter will cover the individual contributions made by the study. Conclusions will respect to each research question will be examined. The different engineering field contributions will be discussed, as well as the overall conclusion of the study. Lastly, any future recommendations will be made should the study continue.

Chapter 2

Literature Review

2.1 Introduction

The following chapter served as the secondary research performed during the study. The research question requires an understanding of current industrial safety systems, in order to form a basis for the conceptual design. A core definition of HRI was investigated, in order to determine which area of HRI this study contributes to. According to the research question which is directed at "**maintaining a safe working environment for the human involved**", identifying potential hazards was investigated in order to be reduced. Current existing flexible systems were also investigated, in order to conceptualize a design.

2.2 Structure of Mechatronics

When investigating the probable technologies to be used in this study, it can be concluded that the study will revolve around mechatronics. Mechatronics may be described as a simple combination of mechanical and electrical engineering, where the electrical aspects include electronic, computer and control engineering. It is an extremely vast field that is described through various definitions. According to [4], mechatronics may be described as the "synergistic use of precision engineering, control theory, computer science and sensor and actuator technology, to design improved products and processes." Other definitions are on a similar path. Figure 2.1 describes the combination of the various fields that provide the make-up of mechatronics.

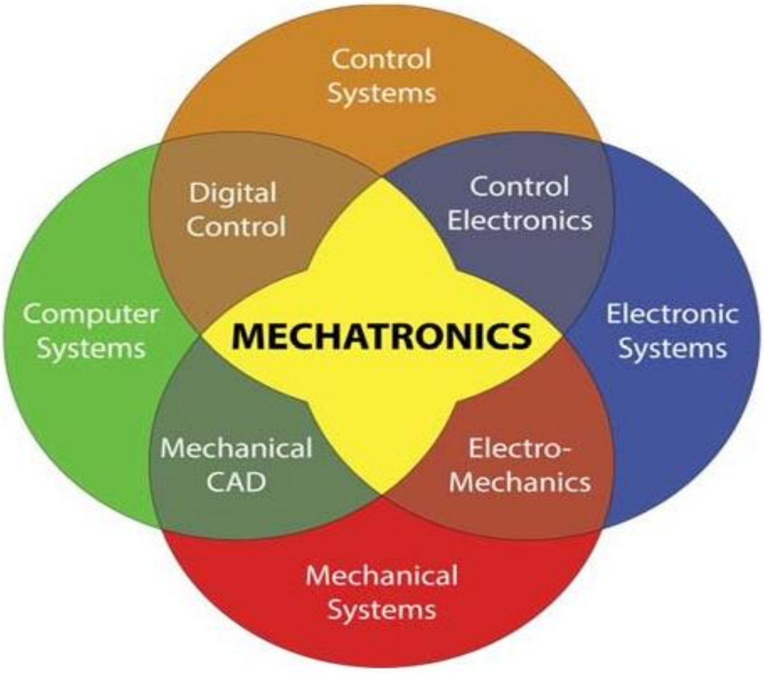

Figure 2.1: Mechatronics Structure [5]

The study will be based on figure 2.1. The 4 main aspects, namely, computer, electronic, mechanical and control systems will be analyzed and developed. Each system will be constructed separately, and then assembled to function as a single unit. Once the systems have successfully synergized, they will be tested and put through a refinement process.

The definition of mechatronics as well as the system model will be transitions into the form of the current study. The following listings describe the physical system relationships:

- Mechanical system – Robot arm design
- Electronic system – Physical connection of various sensors and motors to the controller
- Computer system – Design of the controller coding to compute the various incoming signals from the sensors
- Control system – Design of the mobile application to continuously provide feedback to the computer system

These systems will be designed separately in the following chapters to come.

2.3 Human – Robot Interaction (HRI)

As we enter into the 4th industrial revolution, "The Internet of Things", robots/robotic elements are progressively becoming part of our everyday lives. In the past and currently, robots are used to perform tasks that humans cannot. However, with a rapid increase in applications for robots in general, a human interacting closely with robots is slowly becoming a normality of everyday life. This has lead to the development of the field of study known as HRI.

HRI is the design and implementation of a robotic system that utilizes elements that are both human and automation. However, it is considered a broader field than just the engineering aspect. [6] describes HRI as a cross-disciplinary field that spans across robotics, sociology, psychology, biology and interaction design. The 'interaction design' area deals with the general safety of the human involved.

Before HRI can be implemented, the completed interaction between the 2 parties must be summarized. HRI can be categorized into 6 main themes [7]:

1. Detection and understanding of human activities
    - Allowing robots to try and understand human behaviours for path planning
2. Multimodal interaction
    - Verbal and non-verbal cues to formulate a natural environment during HRI
3. Social learning and skill acquisition
    - Enables new users to easily acquire knowledge on robot skills
4. Cooperation and collaboration
    - Robots must work alongside humans as peers and not tools
5. Long-term interaction
    - The study of keeping users engaged once the novelty has expired
6. Robot operations outside industry (social)
    - Assistive robots in therapy, supporting the elderly, rehabilitation (Contains various safety and ethical issues)

Unfortunately, studies and experiments involving HRI cannot be applied over a broad spectrum. HRI studies are widely innovative, and researchers cannot agree on one specific method of experimentation. Both qualitative and quantitative statistical methods are applied [8]. Data is extracted from these various experiments, and a 'common practice' regarding HRI is built upon and developed. This has lead to experimental procedures to be extremely situational dependant and heavily constrained.

HRI involves the installation of robots into a dense human environment. This brings about the main issue involving safety of the humans involved.

2.4 Hazard Identification

A crucial part of analyzing HRI is to identify the main sources of potential physical and psychological hazards. Unlike humans, most, if not all, robots are pre-programmed to follow a specific set of instructions, which can include a situational adaption to a certain degree. Robots do not possess personal creativity, and are limited to following instructions coded into their mainframe. This results in the limitations of safety features programmed into an industrial robot.

Robotic movement can be easily determined and predicted, whereas human movement cannot. Therefore, it is the responsibility of the human involved to ensure that a safety protocol is frequently used. According to [9], most human/robot accidents occur during non-standard operating conditions, such as during inspections, maintenance, testing and unauthorized use of a robot. It is for this reason that the main scope of this project analyzes the human aspect of HRI.

Risk evaluation is a crucial of the safety design process. If a risk/potential hazard is not accessed, then it cannot be controlled and accounted for. The risk assessment process is described in ISO 12100. Figure 2.2 explains the steps to be followed during the hazard identification process.

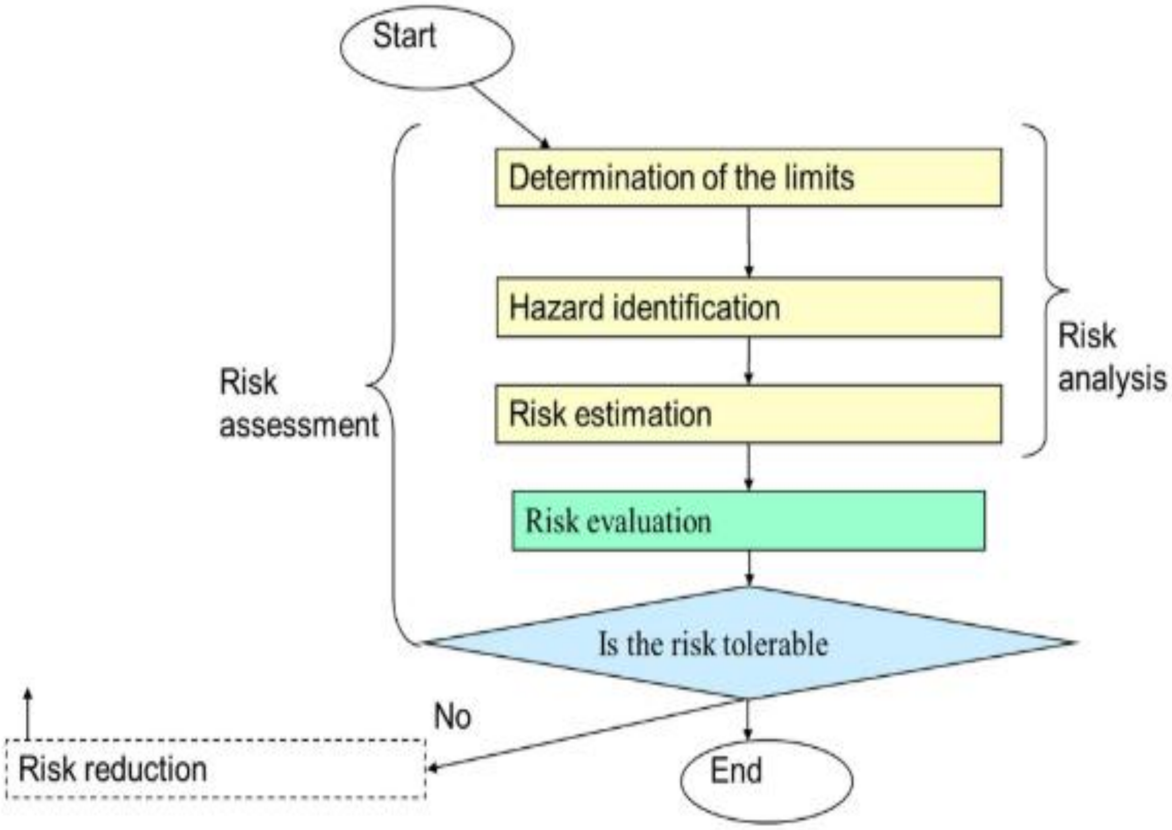

Figure 2.2: Risk Assessment Procedure [5]

As shown in Figure 2.1, the risk assessment is broken up into two main stages, namely the risk analysis and the risk evaluation. The risk estimation step involves taking into account the probability of a specific hazard, as well as assessing its effects on physical and mental health [10]. This will then determine the risk reduction methods to be prioritized and employed.

Most industrial robotic accidents involve the operator. According to [11], studies showed that 72% of the reported injuries were on the robot operators. Maintenance workers were among the others. Unfortunately, it is extremely difficult to isolate a specific hazard, as hazards can vary depending on a specific robot model, and working environment. ISO categorizes these hazards into the following categories between humans and robots [12]:

- Mechanical hazards – Unintended physical contact
- Electrical hazards – Contact with live wiring or exposure to arc flash
- Thermal hazards – Contact with hot surfaces
- Noise hazards – Inability to communicate to other workers

The hazards mentioned in the ISO standards focus mainly on the robotic element. To take into account every potential hazard during HRI, the risk from the human element must be examined. Figure 2.3 describes the various potential hazards that can be prevented from the human elements point of view.

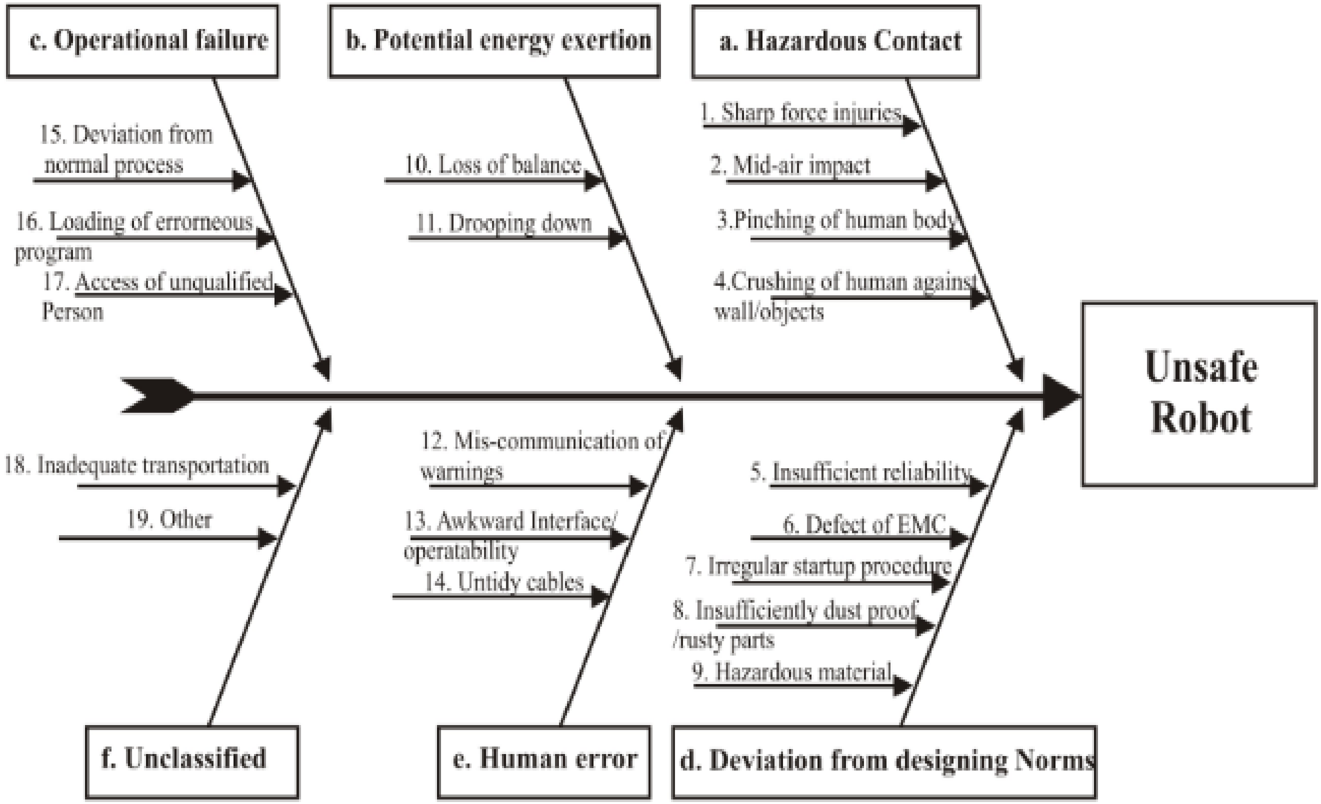

Figure 1.3: Non-Programmable Potential Hazards [4]

Most safety features prevent these hazards from occurring, by placing physical barriers and various proximity sensors around the robotic system. Every robotic system is different and requires a specific safety system, depending on the potential risks involved [13]. Figure 2.3 describes the most basic solution to a specific risk.

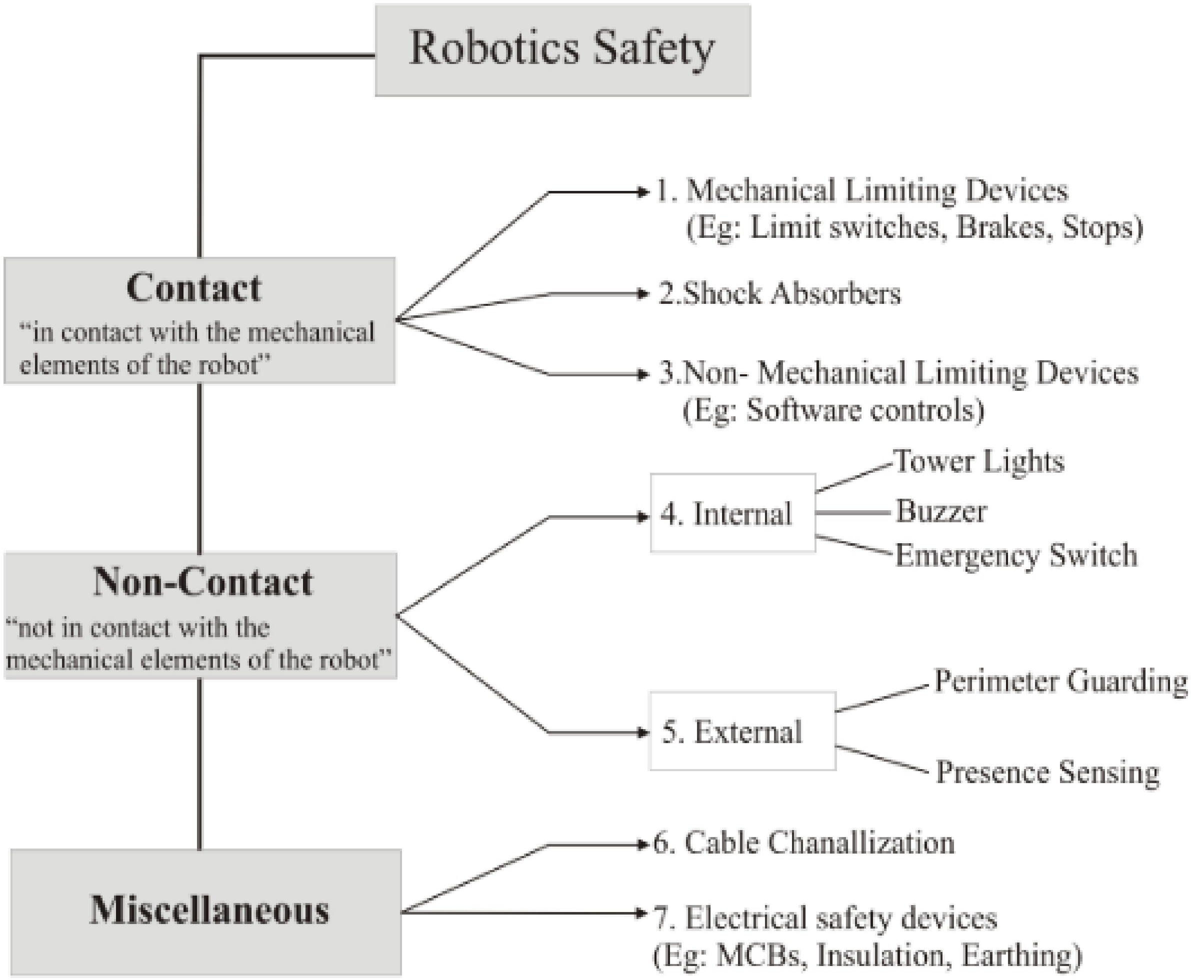

Figure 2.4: Safety Devices for Various Potential Hazards [4]

These safety designs do prevent these hazards from occurring, but compromise the potential of HRI. The proposed safety system will include various elements as shown in Figure 2.3, but must contain a flexible authorization, depending on the individual involved and the specific operation. The crucial standards for industrial robot safety are found in ANSI/RIA R15.06-1999. This stipulates what safety devices should be set in place for a specific installed end effecter [14].

2.5 ABB SafeMove

ASEA Brown Boveri (ABB) is a Swedish-Swiss corporation that specializes in robotics and automation technology. ABB are responsible for the development of the ABB SafeMove and ABB SafeMove2 safety systems. These systems are utilized in various automated industries that require HRI during the operations. SafeMove completely replaces expensive and bulky mechanical barriers around a robotic system, with dedicated electronic and systematic software, to improve HRI in a safe environment [15]. If a hazardous situation arises, one that the SafeMove system does not predict, a cut-off switch will be activated, disabling the robotic system.

According to [16], SafeMove highlights four aspects of HRI. These aspects include removing the physical barriers between robotic and human workers, saving floor space in the design of the manufacturing system, less expenses from an economical point of view and finally, further promoting HRI. Unlike the security systems used in conventional robotics, SafeMove is a flexible control system, and can be considered a safety 'add-on', in the form of a controller for an industrial robot [17]. The controller places restrictions on the robot's movements and speeds, depending on the immediate situation, making for a safer environment for human workers.

2.5.1 ABB SafeMove Technicalities

Various iterations of SafeMove have been developed, depending on the physical applications. All versions are comprised of a variety of sensors that are used to detect the immediate position of a human. A safety computer constantly monitors the robot's movements and path planning. It should be noted that the robot controller and the ABB safety work independently from each other, and must be checked for synchronisation during start-up [15]. A variety of sensory signals are sent through to the controller unit, where they are processed. The PLC then issues an instruction to the robotic system to react accordingly.

The SafeMove system constructs a virtual monitored zone around a robotic cell. This zone can be broken down into various sectors, depending on the application. Figure 2.4 illustrates an example of a virtual zone.

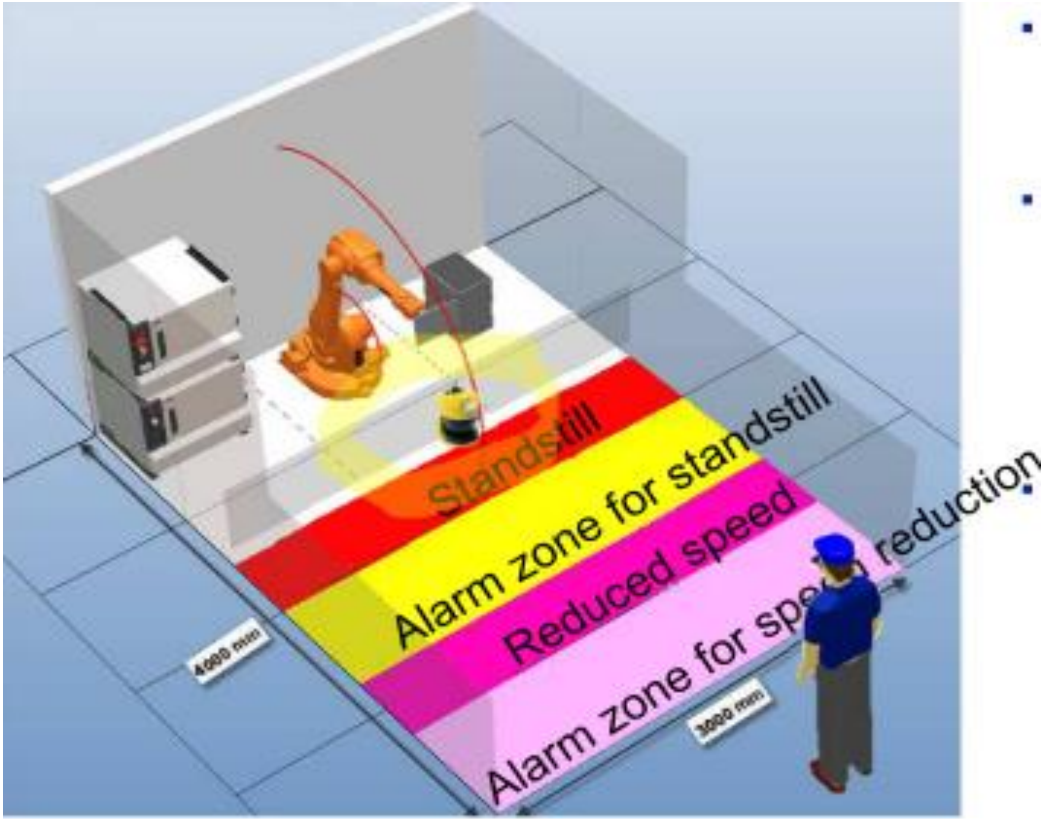

Figure 2.5: SafeMove Virtual Zones [12]

As shown in Figure 2.4, three monitoring zones are used. As a human enters a specific zone, the robot will adjust its operational speed accordingly. The example above makes use of a laser sensor, connected directly to the controller, to determine the proximity of a human. It should be noted that other versions contain light curtains, camera systems and pressure mats. A similar approach will be used in this dissertation, where various levels/zones will be contrasted.

SafeMove2 is a refined version of its predecessor. Unlike the original SafeMove, SafeMove2 no longer requires complex cabling, as the safety module contains built-in safety fieldbuses, removing the explicit need for safety PLC's [18]. Figure 2.5 describes the basic layout of SafeMove2, whilst connected to a serial robot's controller.

Figure 2.6: SafeMove Systematic Layout [13]

As shown in Figure 2.6, any discrete I/O sensor could be connected straight to the robotic control system. Flexible software allows for easy programming of the connected sensors. The programming tool, Robot Studio, is used, to program the various safety parameters required. By using a built-in teach pendant, the user may manually map the layout of a 3-dimensional safety zone, and simulate a model by incorporating the various sensors used [19]. Figures 2.7 and 2.8 shows an example of a teach pendant used, as well as a simulation setup.

Figure 2.7: SafeMove2 Teach Pendant [14]

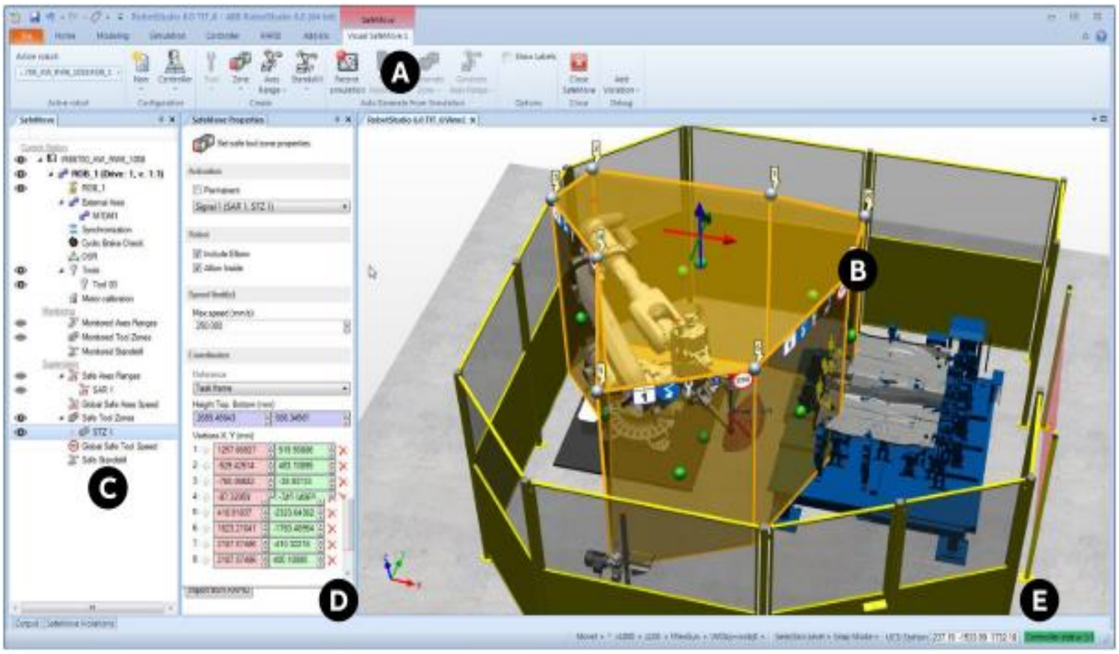

Figure 2.8: 3D Model Simulation using SafeMove2 [14]

The user can model a flexible virtual zone around the robotic system, and combine the I/O sensors at various locations. By programming the PLC, the system can be configured to behave in the desired manner. Overall, the ABB SafeMove system provides a robust method of mapping out a monitored zone around a robot, whilst incorporating discrete sensors to monitor human behaviour. The system will then act accordingly, such as reducing operational speed, or activating a kill switch. This concept will be used in the design phase of this dissertation.

2.6 Proximity Sensors

The following sections will investigate the various proximity sensors that can be installed onto the SIS, in order to detect a human presence. Research will also be conducted on how these sensors are currently used in the safety of industrial robots.

2.6.1 Radio Frequency Identification (RFID) Sensors

The applications of RFID technology have increased dramatically in recent years, ranging from access control to buildings, airport luggage control as well as supply chain management [20]. The overall system utilizes an electro-magnetic field, spanning over a certain region, to specifically identify multiple, or individual electrical devices, without the use of human assistance. The proximity systems primarily consist of a reader (active tag), transponder (passive tag) and a main computer for the storage of the data [21]. Various types of RFID devices exist, but they can be categorised into two main groups: passive and active. These devices usually take the form of a card or tag (it should be noted that active tags are assumed to send and receive signals) [22]. Figure 2.8 describes a simplified layout or a proximity detection system.

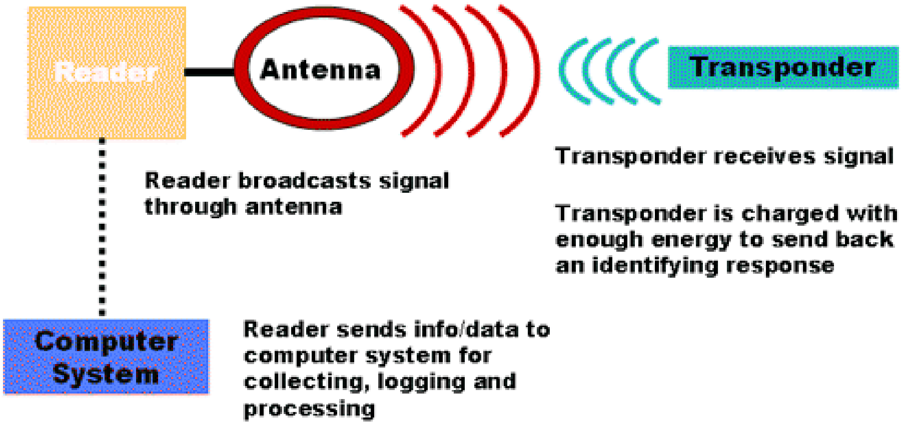

Figure 2.9: Proximity Detection System [18]

Active tags require a battery source, and are responsible for creating the magnetic field. Due to the fact that these tags require an external power source, the performance life span is limited. For simple applications, a system can be comprised of a single active tag, representing the core of the workspace. Various passive tags can be introduced into this magnetic field, and will react accordingly. A further explanation of each category will follow.

2.6.1.1 Active RFID Tags

As stated before, the reader will generate an electro-magnetic field comprised of a specific radio frequency, usually ranging between 120kHz to 130MHz [21]. It should be noted that the selected frequency has a direct impact on the range of the generated field. The range mentioned should create a field radius of roughly 1.5m in length. However, various 3rd party electronic devices within the generated field, possess risk of receiving disturbances from the radio waves projected. This should be taken into account when selecting an operating bandwidth.

A proximity based system does not have to be comprised specifically of the 3 devices mentioned above, but can rather be comprised of a unique combination of the devices, depending on the application and scenario. As displayed in an experiment by [23], active tags were placed on each individual member. Each member then transmitted its own bandwidth, essentially creating its own ID, to be recognised by the system. The data can be processed computationally, to determine the exact location of the tag (or wearer). This proved an efficient method for specific proximity detection of multiple bodies. However, the main point of concern is the life span of these tags. As stated before, active tags always require a power source and must be constantly maintained.

The readers are not limited to only generating one frequency of radio waves. The design of a remote programmer was explained by [21], where a single active tag was used to control various passive tags, which in turn, conveyed instructions to a robot. The following image shows the active device alongside the respective PLC.

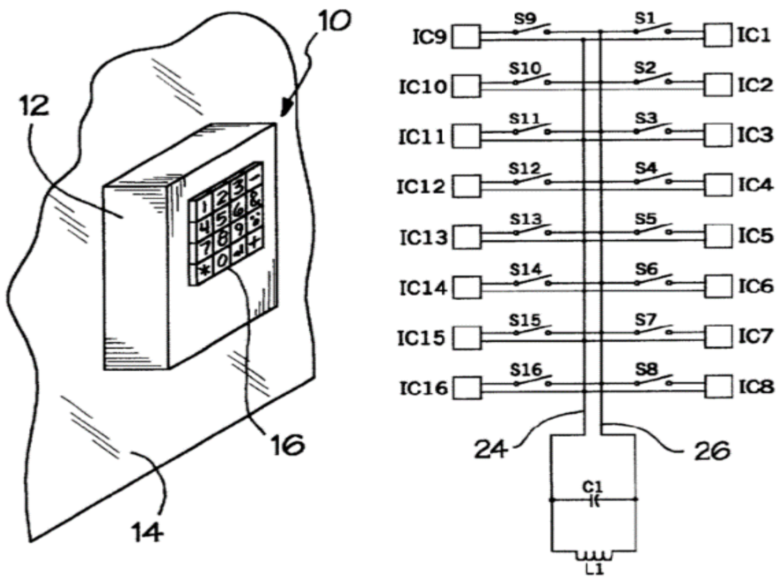

Figure 2.10: Programmable Reader for the Control of a PLC [16]

As shown in Figure 2.10, a user may punch in a number directly on the keyboard, which will then broadcast a specific frequency, and activate the respective passive tag. The invention proposed solution for complete wireless control over a robot's PLC, with an active tag as the controller.

2.6.1.2 Passive RFID Tags

Passive tags do not require a personal power source, resulting in their cost being minimal. Unlike the active tags, passive tags possess a larger flexibility in their design, and could be implemented into the form an ID card, surfaces stickers or implanted in biomedical devices [20]. The circuit design of these tags is very simple, as they are only comprised of a computational chip connected to a series of antenna wires. Figure 2.10 describes the basic layout of a conventional passive tag.

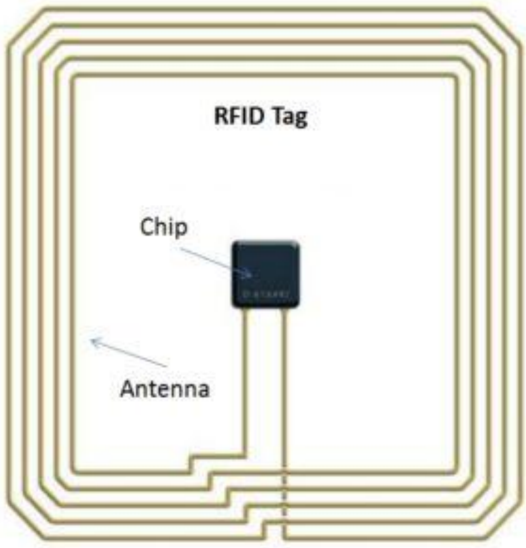

Figure 2.11: Passive RFID Tag Integrated Circuit [20]

As shown in Figure 2.10, the computational chip is powered through the connection of the antenna wires. The antenna receives a broadcasted signal, through the form of a magnetic wave, which then gets converted into a voltage to power the chip [24]. As stated in the previous section, an active tag can be designed to broadcast various signals, but passive tags may only respond to a certain bandwidth. Once the chip is activated, it will then send a return signal to the reader, completing the communication between the two devices.

It should be noted that the signal strength (distance) is directly proportional to the frequency of the projected bandwidth. Table 2.1 describes the three main bandwidths, according to [25], and their respective distances.

Table 2.1: Passive RFID Tag Operating Frequencies

Category	**Operating Frequency**	**Distance**
Low Frequency (LF)	125 – 134 KHz	1cm – 10cm
High Frequency (HF) & Near-Field Communication (NFC)	13.56 MHz (and above)	1cm – 1m
Ultra High Frequency (UHF)	865 – 960 MHz	5m – 6m

Frequency bandwidth is not the sole decider of the reading distance. Environmental conditions such as weather, temperature and the presence of foreign bandwidths can affect the operational efficiency of the RFID system. UHF passive tags can respond from distances up to 30m, given the correct environmental conditions [25].

In some cases, the design of the tag is not always dictated by the physical distances, but by the voltage requirement of the chip. A bandwidth frequency, as well as other tag design parameters, can be determined with respect to the required voltage of the chip. The following equation describes the relationship mentioned [26]:

$$V_{Tag} = 2\pi f NQB(Scos\alpha)$$

where:

V_{Tag} = Resultant voltage across the antenna

f = Frequency of the carrier signal

N = Number of coil turns in the antenna

Q = Quality factor of the integrated circuit

B = Strength of the magnetic field at the tag

S = Surface area of the coil

α = Angle of the field normal to the tag area

The above formula can be used to calculate any of the listed parameters required during the design process.

Overall, an RFID system is a cheap and efficient solution to a proximity detection system. It also possesses a high form of flexibility, as a various number of passive and active tag combinations could be used. However, the system's operating frequencies must be carefully taken into account, as foreign disturbances can interfere with the system's performance.

2.6.2 Bluetooth

There are 2 main options for using Bluetooth as a sensor. The first would be to use Bluetooth Low Energy (BLE) beacons. These beacons are utilized in a variety of applications, such a commercial stores and electronic toys. BLE beacons perform similarly to RFID tags. In an experiment, according to [27], these beacons can be installed onto robots for prototyping HRI. The system involves using a central Bluetooth device, that recognises the surrounding BLE beacons. Each beacon serves as an individual form in identification, as well as only requiring the power supplied by a coin battery. However, the issue remains that the user can only communicate 1 piece of information to the robot.

The second method involves using an individual's personal mobile phone through a mobile application. In a study by [28], a mobile application was used to directly control an Arduino driven car. The controller used an HC-06 Bluetooth module, to communicate with the mobile device. This allowed the user to relay a multitude of instructions to the robot system. Bluetooth module models work on a host/slave scenario. Depending on the scenario, the host device may connect with up to 7 other devices at a single point in time [29]. Some Bluetooth modules may operate in a dual mode, being able to act as both the host and slave.

The second option is much more viable for enabling a form of personal identification; where as the use of the BLE's is more for the design of a proximity sensor.

2.7 Conclusion

This section has investigated the in-depth definition of HRI. It was found that HRI covers a wide field of study. The hazard identification process identifies what specific aspects of the SIS must be analyzed closely. Existing SIS examples were studied, namely the ABB SafeMove system. Various proximities sensors, such as RFID's and Bluetooth beacons were investigated, in order to select the optimum choice for the required application.

Chapter 3

Proposed System Overview

3.1 Introduction

This chapter covers the initial idea with respect to the simulated system. Modern day manufacturing systems was investigated and a simplified scenario was generated. A scaled down version was designed. According to the research question, which is directed at "**to employ a flexible safety integrated system (SIS), which removes the physical boundaries placed around a robotic system**", a test rig was then constructed for the next phase of the study, which will serve as the robotic system described. The test rig must allow for flexible experimentation, where the connection of various sensors can be easily interchanged, and the controller programming can be altered. This allowed for optimum refinement of the system.

3.2 Overview

The proposed safety system is a "third-party" system that will be installed directly onto a robot. The focus is to create a flexible virtual zone around any piece of machinery that poses a potential hazard. For the purposes of this study, an automated robotic environment is simulated, where HRI will be required during a specific process step. The specific example is described as follows.

The simulation will consist of a robot performing an assembly. The robot assembles a mechanical part that requires a variation. Instead of designing a new robotic system to install the alteration, a human will be introduced into the system. A physical example is a robot assembling a brake pad. As the supplier, the same brake pads are manufactured for 2 different companies. A name brand will need to be installed onto the separate pads for each company. A human is introduced into assembly process to quickly install a name onto the pad. Figure 3.1 illustrates the example described.

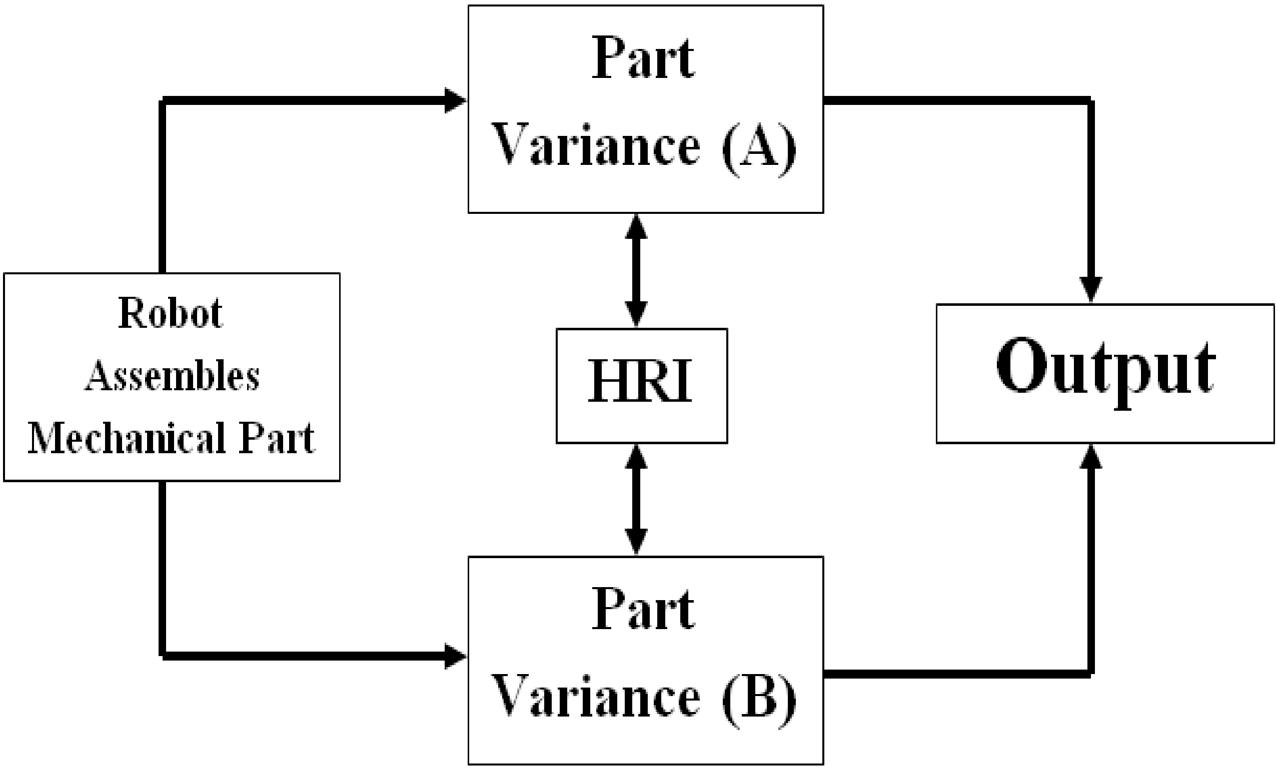

Figure 3.1: Human-Robot Interaction Simulation

As the human is incorporated into the assembly system, the issue of safety is now relevant. The assembly robot should be able to recognise that a human has entered the vicinity. It is common for industrial robots to operate at extremely high, repetitive speeds. These operating conditions pose a great threat to general safety of the humans in proximity. The robot needs to recognise that a human has entered the physical workspace. Standard industrial protocol will require the assembly line to be disabled temporarily. To avoid this, the robot should instead alter its path plan and operating speeds, to ensure safe HRI.

With dozens of workers on a factory floor, human error must be provided for. Not every worker will be permitted to perform an assembly concurrently with the robot. The robot must be able to identify each individual person, and decided on whether they are allowed to enter the workspace or not. A method for personal identification will be used. If the test for entering the workspace is failed, the system should temporarily shut down, and an alarm or buzzer should activate.

The last form of control is the immediate proximity of the robot. If a human is permitted to enter the workspace, there should a minimum distance that the human will be allowed to approach the robot system. This prevents human from coming into immediate contact with the robot. This distance will be measured from the maximum reach that the robot will cover during its specific path plan. A proximity sensor will be placed on the robot, to monitor the human's distance from the robot at all times.

3.3 Conceptual Design

The core of the simulation revolved around a robot performing an assembly. The safety system will be installed directly onto the robot's control system. Signals from the various sensors will ultimately decide what path plan and operating speed the robot will follow.

Three "layers" are incorporated into the design of the system, namely, L1, L2 and L3. This "layered" approach was inspired by [30]. Every layer is connected directly to the controller that processes the scenario data, and will then issue an instruction accordingly. Figure 3.2 describes the basic layout of the proposed system.

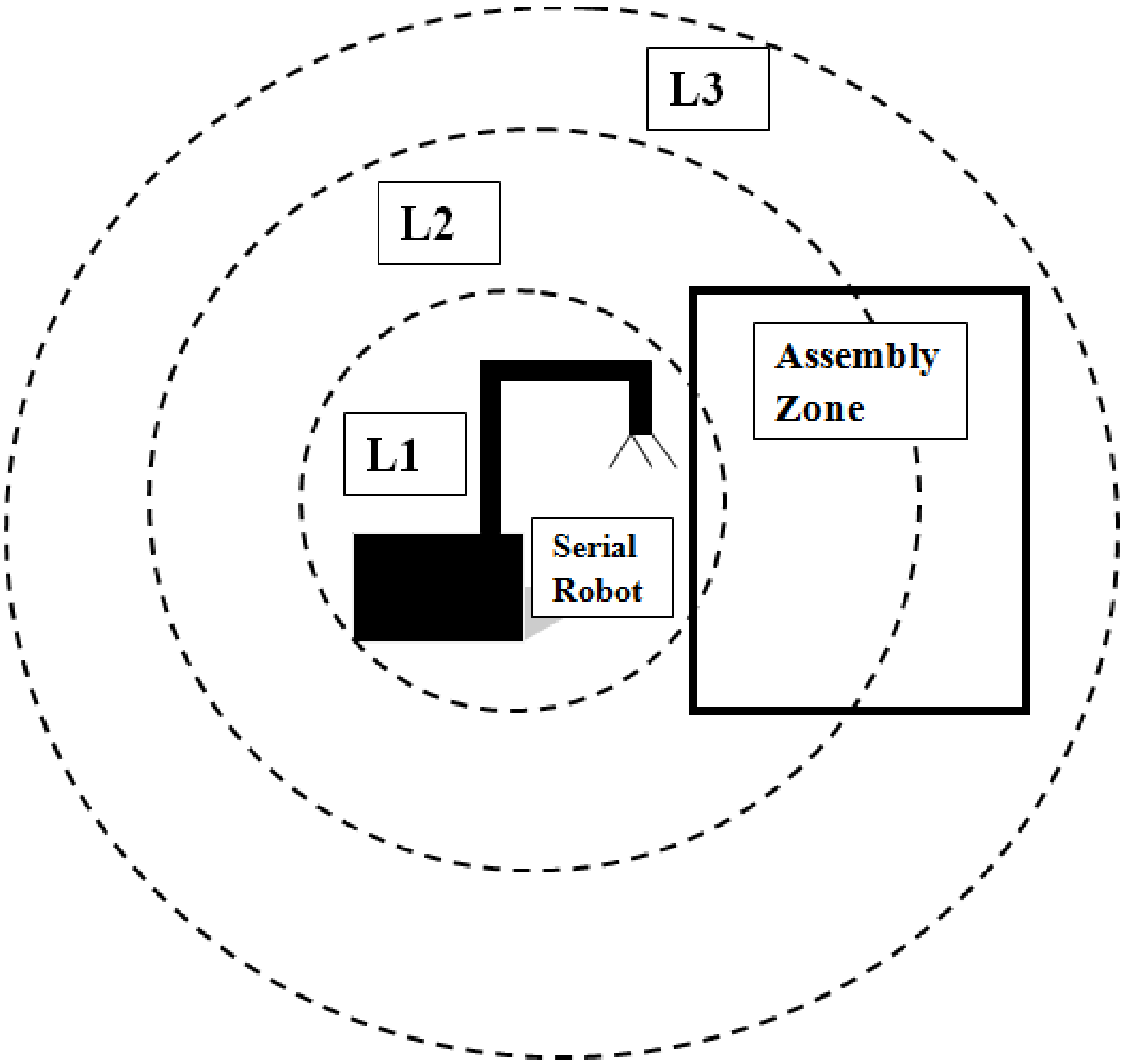

Figure 3.2: Proposed Safety System Layers

In Figure 3.2, a basic industrial scenario of a serial robot performing an assembly is shown. L3 encloses the entire system and sets the furthest boundary of security. This layer monitors a human presence entering and existing in the working area. The human has not entered a danger zone as of yet, however the robot is aware of the human presence. This can be designed using a variety of proximity sensors, such as light curtains, laser sensors and proximity mats. Camera systems are also a viable option, L3 can also detect the number of humans present within the monitored zone, and can place a limitation on how many humans are allowed inside.

L1 represents the inner most area, where no human should enter if the robot is in operation. The physical size of this area may vary depending on the machine and application, as well as the specified path plan. Should a human enter this area, a kill switch should be activated to stop all automated movement. Similarly, to L3, various sensors may be coupled onto the system to improve the safety rating.

The main focus of this study is the design of L2. Once a human enters L3, L2 will then grant permission to the human entering the area. Various permissions will be granted, depending on a number of factors. L2 involves the communication between a mobile phone and a controller unit, and monitors the activities occurring whilst the human is within the vicinity of the robot. The method of communication between the mobile device and controller will be Bluetooth. Through the use of a mobile phone, this serves as a form of personal identification. The mobile application will be loaded onto an individual user’s phone, creating a virtual identity for each user. This allows the robot to observe who has entered the workspace, and will then grant permission accordingly.

3.4 Serial Arm Model

The simulation described above will require the use of a robotic system to serve as a test rig. The design of the safety system is installed directly onto the robot’s controller, to manipulate it directly. For the purposes of this study, a small-scale serial arm is used. For simplicity, the drawing files for the serial arm was purchased and downloaded, as the link joints were previously designed for a specific motor. The serial arm model was based on the Little Arm 2c. Figures 3.3 and 3.4 describe the drawing files purchased.

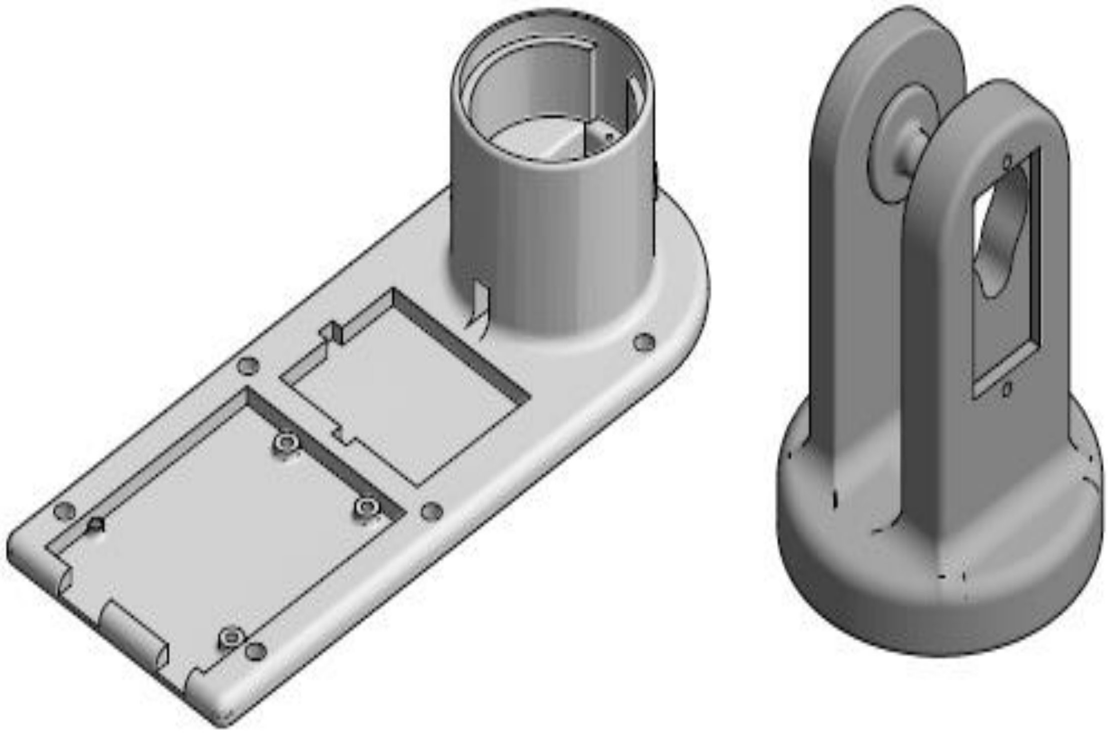

Figure 3.3: Little Arm 2c Base & Shoulder

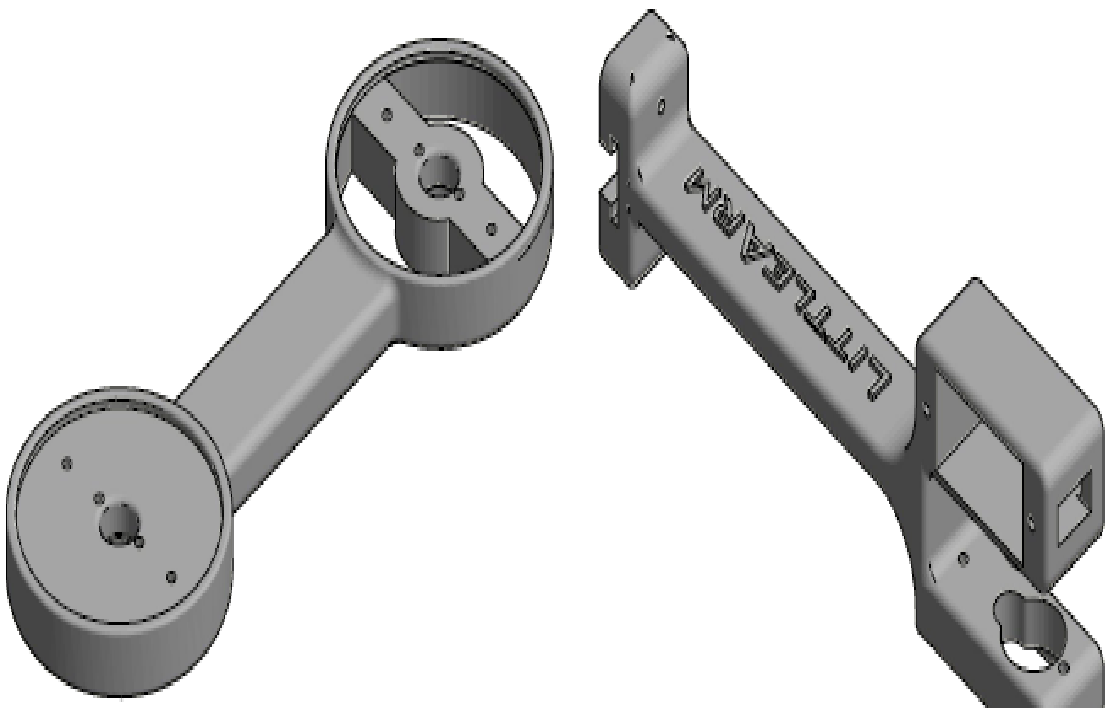

Figure 3.4: Little Arm 2c Upper Arm & Forearm

The design’s joints require the use of 3 MG90s Tower Pro servo motors. As describes in the Figures above, there are 4 main components. The base piece will also hold the controller unit. The parts were then 3D printed and assembled. A PLA filament was used over an ABS plastic, as the robot model will not be required to lift any heavy loads. Figure 3.5 shows the final assembly of the arm.

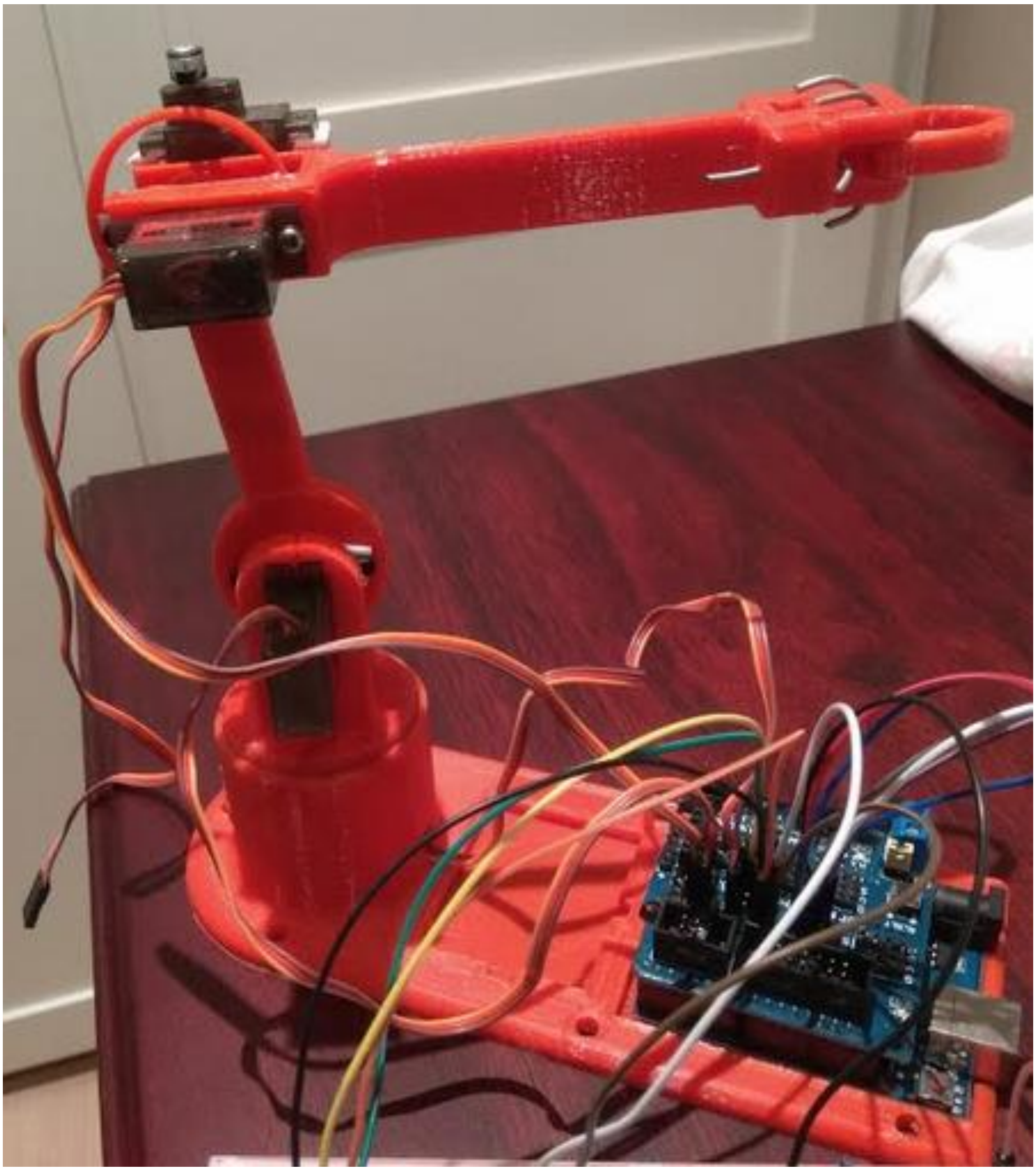

Figure 3.5: 3D Printed Arm

In the robot's shown neutral position, it stands roughly 15cm high. A controller unit was mounted on the base and will directly control the 3 motors to plot a specific path movement of the robot. The safety system is then wired directly to the controller, and the robot's reactions will be monitored.

3.5 Conclusion

The proposed system simulates a physical scenario of a robotic arm involved in an assembly process. The security system is coupled directly onto the robot’s controller. Three layers make up the security system, with the second layer being the main aspect of the study. The outer and inner layers are comprised of proximity sensors, to detect a human’s physical presence. The second layer is a mobile application that grants a human the permission to approach the robot’s assembly zone. This serves as a form of personal identification. The application may also allow the human to instruct the robot accordingly, depending on the permissions granted. The 3-D printed model developed will serve as the mechanical system in the mechatronics overview.

Chapter 4

Electronic Design

4.1 Introduction

The following chapter discusses the technicalities of the electronics involved. The requirements were listed in chapter 3. According to the research question, "**Can the robot system react accordingly**", the sensors and electronics installed into the system are directly responsible for the robot's reaction to an approaching human. Once the electronics were selected and purchased, they are then configured and wired into the correct scenario. Once they were connected, the controller was programmed for the intended functionality to take place. Thereafter, a final physical design was implemented.

4.2 Controller Selection

An Arduino controller was used as the central processing unit for this system. The main reason behind the selection of this controller is to easily couple the security system onto a robotic system. This will simplify the experimentation process as the programming will be manipulated easily. Figure 4.1 describes the basic controller function and setup.

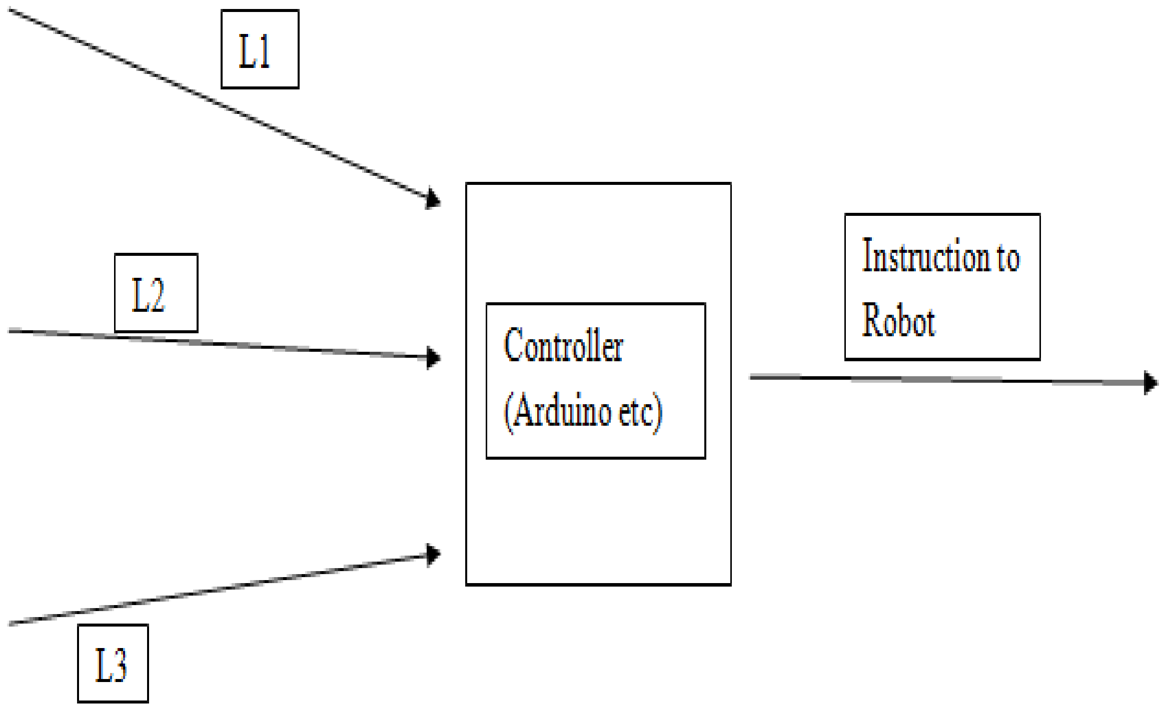

Figure 4.1: Controller Signal Process

As shown in Figure 4.1 the controller receives a constant flow of signals from the sensors making up the various layers. As mentioned before, L1 and L3 are each be comprised of a single sensory unit that will be wired and programmed directly onto the Arduino. These sensors will return a LOW/HIGH value that was utilized in the Arduino's coding.

The main points of interest with respect to the design of the controller are the wiring of the various sensors to the Arduino, as well as the logical programming of the controller. The following sections describe the various components and their functions.

4.3 Electronic Components

The following components were used in the physical design of the security system. The proximity sensors used for L1 and L3 could be interchanged with other, more accurate sensors. The following list describes the components used.

- Arduino Uno R3 – The Arduino will serve as the main controller for the entire system. The controller uses an ATmega16U2 microcontroller. The controller is directly responsible for the robot's movements, as well as processing the information from the various sensors.

- Arduino Sensor Shield v5.0 – The controller shield is placed directly on top of the Arduino. This simplifies the physical connection process for the electronic layout. This also ensures that each component receives the same voltage to avoid voltage splits among the components.
- MG90s Servo Motor – Each motor is connects each link of the robot's joints. These servo motors have a rotation range of 180 degrees. The motor has 3 connections to the Arduino, namely GRND, PWR and SIG. The PWR can be directly connected to a 5V power pin on the Arduino board. The signal pin is used to rotate the motor to a specified angle. The programming of the Arduino can be manipulated to vary the speed and direction of the motor.
- HC-06 Bluetooth Module – The Bluetooth module allows an external device to directly send instructions to the Arduino through Bluetooth. The mobile application is designed to send signals through this medium, as well as forming the makeup of the L2 layer.
- KY-012 Active Buzzer Module – The buzzer is used to create an alarm of the security system find a breach. When the module is activated, the entire robotic system comes to a standstill.
- Ultrasonic Distance Sensor HC-SR04 – This sensor monitors the L1 layer. This prevents a human from coming into immediate contact with the robot. The sensor is mounted directly onto the robot, and monitors the distance of objects directly in front. It can be placed into a rotating stand to increase the area monitored around the machine. The sensor works off basic acoustics. An ultrasonic sound is projected to bounce off the object in front of the sensor. The module then receives the reflected sound, and measures the time delay between signals in order to determine the physical distance of the object. The module can be programmed to project a sound over a specific interval.
- Electronic Mat – An electronic mat was constructed for this study. This makes up the L3 layer of the system. Two pieces of wood were covered with aluminium foil on one face. A piece of foam was stuck between the faces. When a human steps onto the surface, the foil surfaces touch. When wired to the controller, a signal will be recognised as someone has entered into the robotic area.
- Jumper Cables – Standard Arduino male and female jumper cables are used to connect the various components to the controller.

Figures 4.2, 4.3 and 4.4 describe the various components listed above.

Figure 4.2: Proximity Sensor Mat

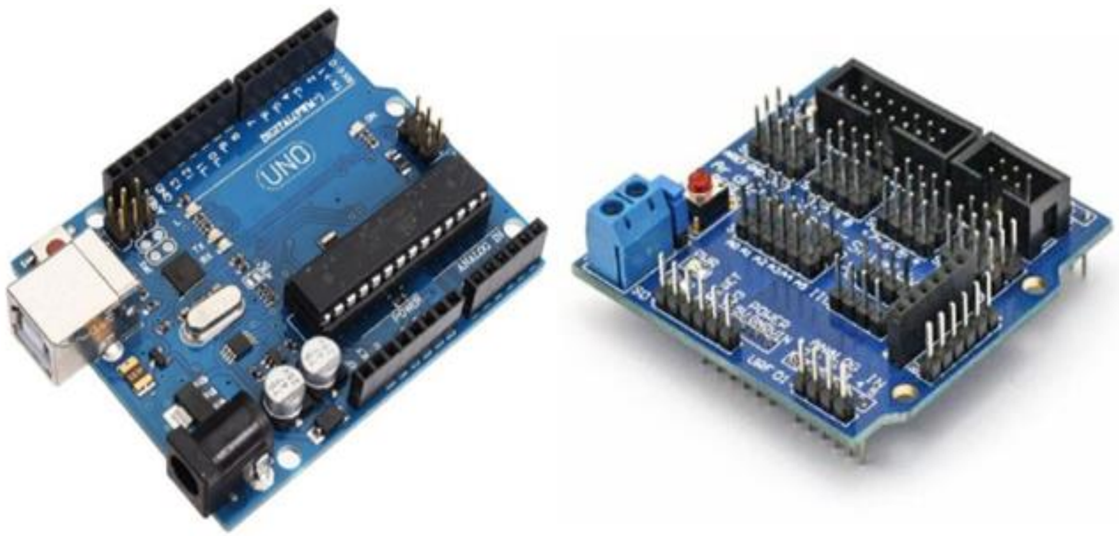

Figure 4.3: Arduino Uno R3 & Arduino Sensor Shield v5.0

Figure 4.4: MG90s Servo Motor

4.4 Wiring HC-06 Bluetooth Module

L2 will collect sent data, from the mobile device, and instruct the Arduino accordingly. The method of communication chosen for this application is Bluetooth. A HC-06 Bluetooth Module will be used. This specific module will only operate as a ‘slave’, which is required for this application as only one direction of communication is required. It should be noted that a HC-05 Bluetooth Module can also be used and must wired identically. Figure 4.5 shows the physical module as well as the pins to be connected.

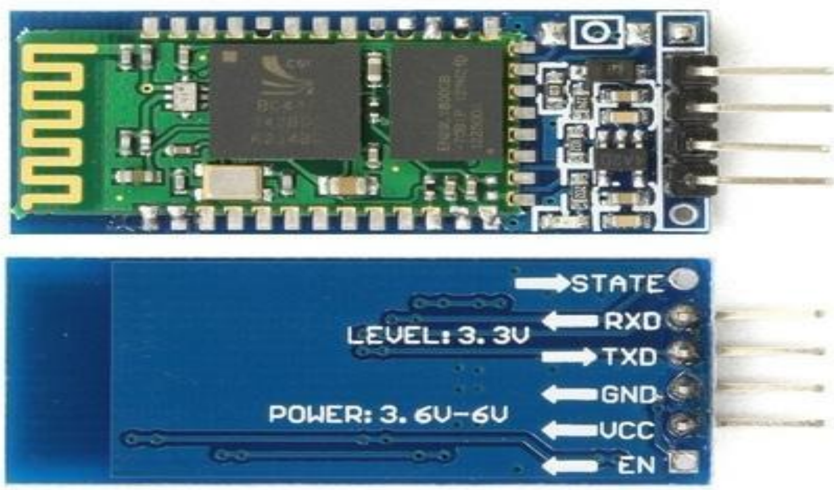

Figure 4.5: HC-06 Bluetooth Module

As shown in Figure 4.5, 6 connections are possible, however, only 4 will be utilized in this application (excluding STATE and EN). The VCC pin powers up the module, requiring a minimum of 3.6V, up to 6V. Standard Arduino pins release a 5V signal; therefore the VCC pin can be connected directly to the controller. The GND must be connected to a grounding pin.

The TXD and RXD pins represent the transmitter and receiver pin respectively. These pins control all the communication activity. It must be noted that the module transmitter pin must be paired with the Arduino receiver pin, and vice versa for the module receiver pin. If the wiring is reversed no serial communication will occur. The module transmitter pin can be directly wired to the Arduino.

As shown in Figure 4.5 the module RXD must not receive a signal greater than 3.3V. This poses a problem, as a standard Arduino will transmit a signal of 5V. A voltage bridge was placed on the module RXD pin in order to prevent damages to any module components. It should be noted that during the experimental phase of design, the module would not operate if it was directly connected to the Arduino, as component damages most likely occurred. Figure 4.6 describes the wiring of the module to the Arduino, using a standard electronic breadboard.

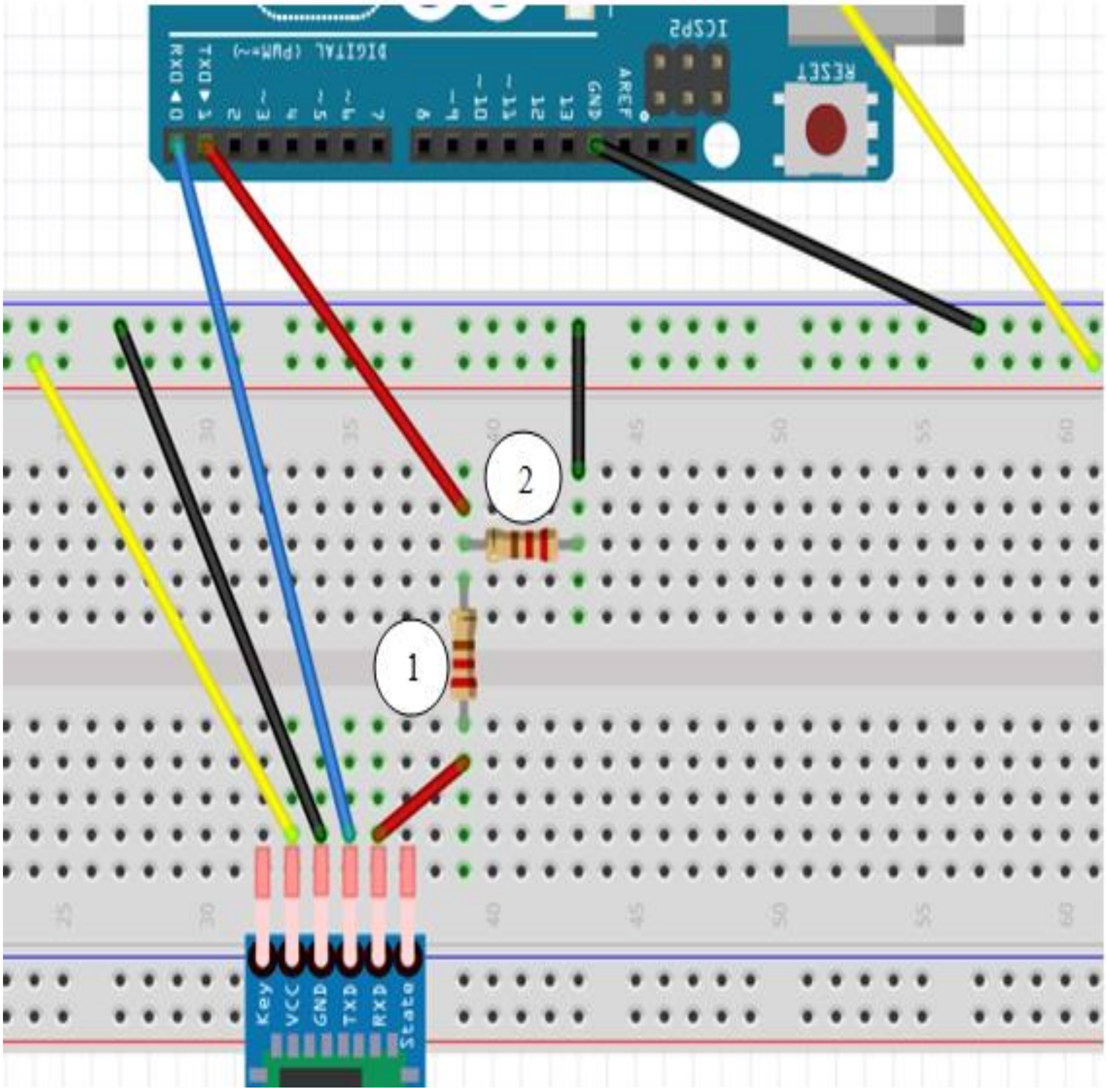

Figure 4.6: Wiring of Bluetooth Module to Arduino

Table 4.1: Arduino/Bluetooth Module Wiring Reference

Electronic Component	**Function**
Yellow wire	Voltage supply (5V)
Black wire	Ground connection
Red wire	Bluetooth receiver connection
Blue wire	Bluetooth transmitter connection
Resistor 1	1K Ohm resistor
Resistor 2	2K Ohm resistor

Figure 4.6 shows how a voltage bridge should be constructed in order to protect the receiving pin. The selected resistors split the voltage such that 66% of the voltage (3.33V) is sent to the module safely. Table 4.1 shows the colour-coding used for the wires in Figure 4.6. The Arduino supplies a constant 5V to the breadboard, and preferably must be powered using a 9V battery for experimental purposes. The L1 and L3 sensors were connected directly to the Arduino pins, where the received sensor signals will be programmed accordingly.

It is important to note that whilst uploading any code to the Arduino via USB, the RXD and TXD pins of the Arduino must not be connected to the Bluetooth module, as these pins are required for communication with the USB. Once the code is uploaded to the Arduino, these pins can be immediately reconnected to the Bluetooth module, which will begin serial communication. It is possible to reprogram any 2 pins on the Arduino to simulate the RXD and TXD pins accordingly, eliminating the need to remove the connection during uploading.

4.5 Conclusion

This chapter covered the various electronic components purchased and implemented into the security system. Specific models of the components were listed, and described the methodology of their implementation. Special notice must be made when performing the physical connection of the components. The sensory network provided the electronic system in the mechatronics model.

Chapter 5

Arduino Controller Programming

5.1 Introduction

The purpose of this chapter is to develop a design and structure for the Arduino's code. It is a direct addition to solving the same question as the chapter 4. Through the incorporation of the sensors and actuators, the computational system is now designed for a programmed response. The code must be tasked with recognizing each individual signal received by the various sensors attached. The controller then processes the signals received, and relays a single specific instruction for the robot to follow. The system must also recognise the safety limitations of various sensors, and should be able to employ a kill switch to ensure the safety of the human involved.

5.2 Signal Processing

The Arduino must be programmed to acknowledge every signal received. The scenario begins with the assumption that the robotic system is performing its standard operational duties. L3 is the first sensor that will detect a human presence. An internal check must be performed in order to assess whether or not the human is authorized to enter. This is done by referring to the serial communication activity i.e. the user must be logged into the mobile application. If no serial communication occurs, the kill switch system will be activated. Should L1 return a high signal at any time, the kill switch system will be activated.

If authorization is granted, the robotic system will assume standard HRI operations. Any specific instructions relayed from the mobile application will result in a prompt response from the robotic system. Figure 5.1 describes the flow chart for the overall process.

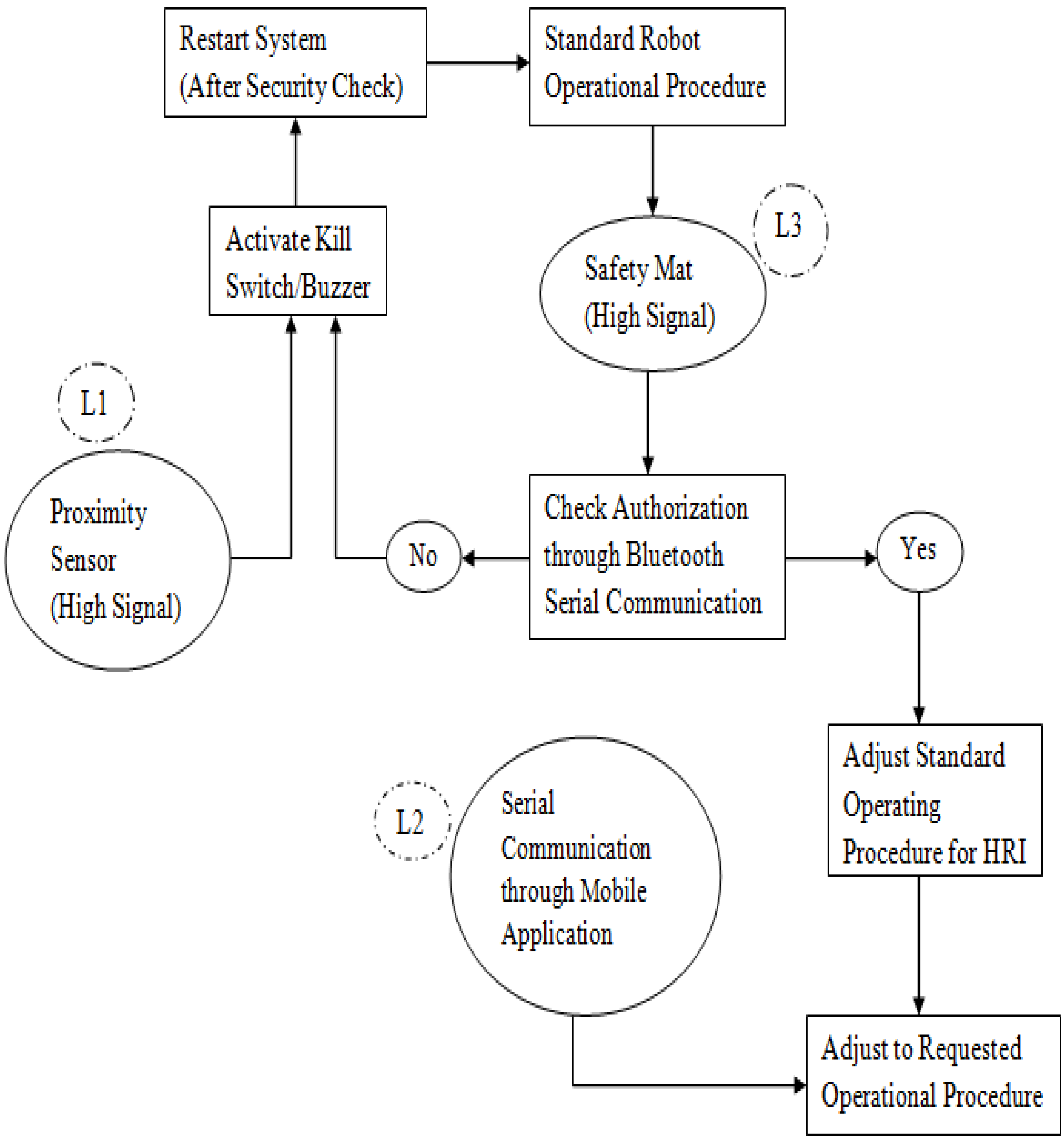

Figure 5.1: Arduino Program Flow Chart

Figure 5.1 describes the order of coding that the Arduino will follow. A series of “WHILE” statements are used to access the different scenarios when processing the various signals received from the sensors involved. For the purpose of this study, 2 robot path planning algorithms will be used. The first was used to represent the standard operation, the second will be used as an automatic adjustment as a human enters the monitored zone.. These algorithms will be coded directly into the scenario loops for the Arduino code.

5.3 Servo Motor Code

The servos rotate to a maximum range of 90 degrees in either direction. Unfortunately, the Arduino can only specify a single instruction to the servo at a time. That instruction moves the servo shaft to the specified angle. There is also no direct method to control the speed of the servo. The use of a “FOR” loop is required to control the rotational speed, which provides smoother control over the robot. The delay between each loop (measured in microseconds) determines the rotational speed, by taking the delay time and multiplying it by the number of increments. An example of the described loop is shown in Figure 5.2.

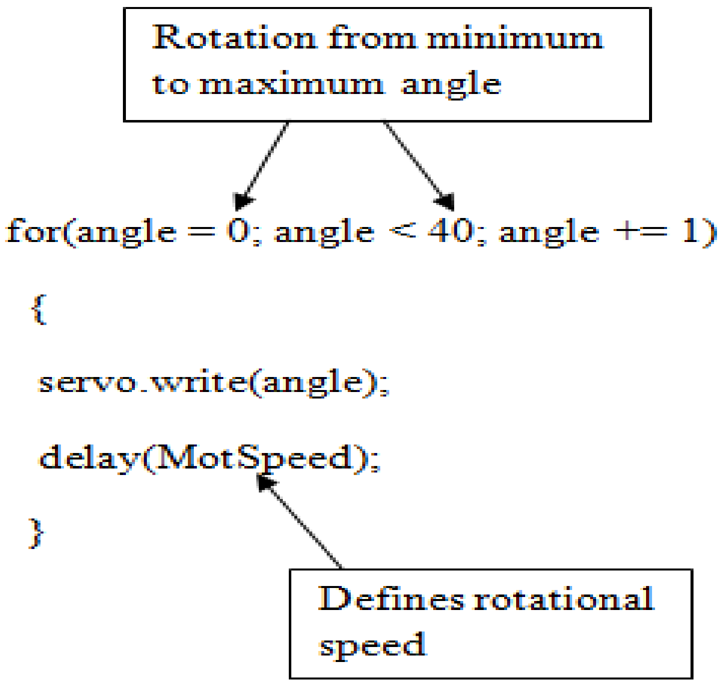

Figure 5.2: Arduino Servo Loop Code

Unfortunately, for this application, each of the 3 motors requires their own specific loop for each movement.

5.4 Ultrasonic Distance Sensor HC-SR04 Code

The sensor's governing pins are known as the trigger and echo in. The trigger pin is responsible for releasing a sound wave that is projected in the direction the sensor is facing. The sound waves bounce off an object found in front of the sensor. These wave return to the sensor where the echo pin acknowledges the signal. The Arduino then processes the time lapse. By using the speed of sound and halving the recorded time, the distance between the sensor and object can be recorded. Figure 5.3 shows an example of the code used.

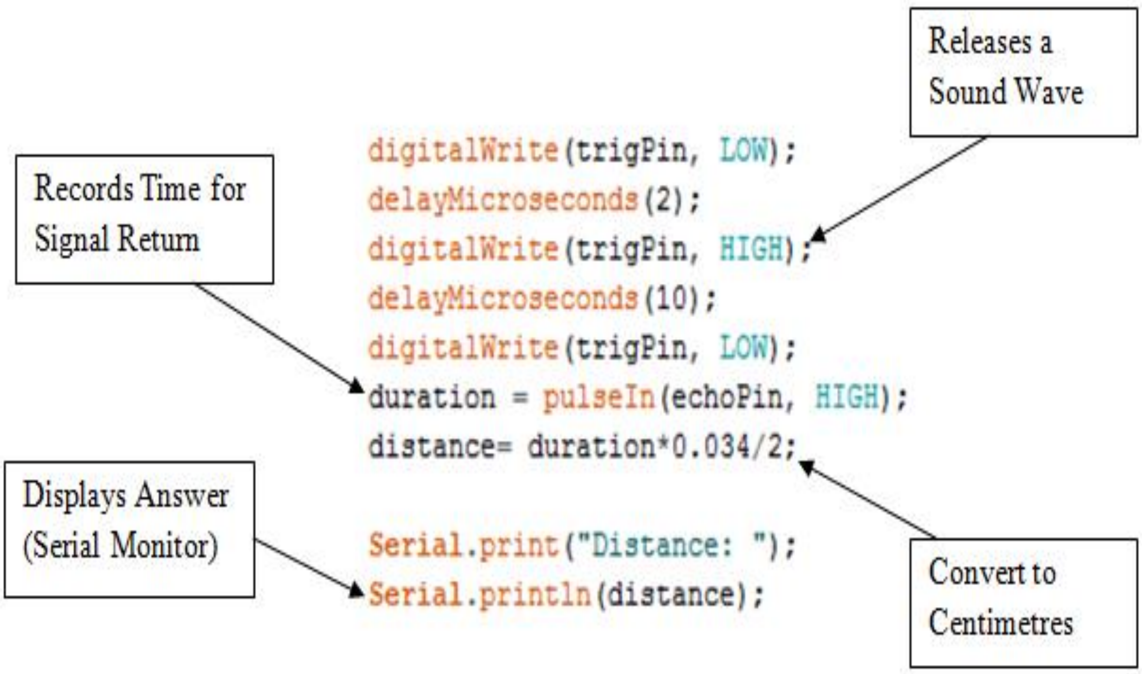

Figure 5.3: Ultrasonic Sensor Arduino Code

The "distance" variable is of most interest to the application. It is calculated using the duration determined by using the echo pin's signal. This parameter is halved to take into account the sound wave travelling both forward and back to the sensor. In Figure... the distance result is given in centimetres.

5.5 HC-06 Bluetooth Module Code

The Arduino needs to recognise when the mobile device connects with the Bluetooth module. A “While” loop needs to open once the connection is made, to then run the required code. When the connection is made, serial communication is commences. Figure 5.4 describes an example of the code.

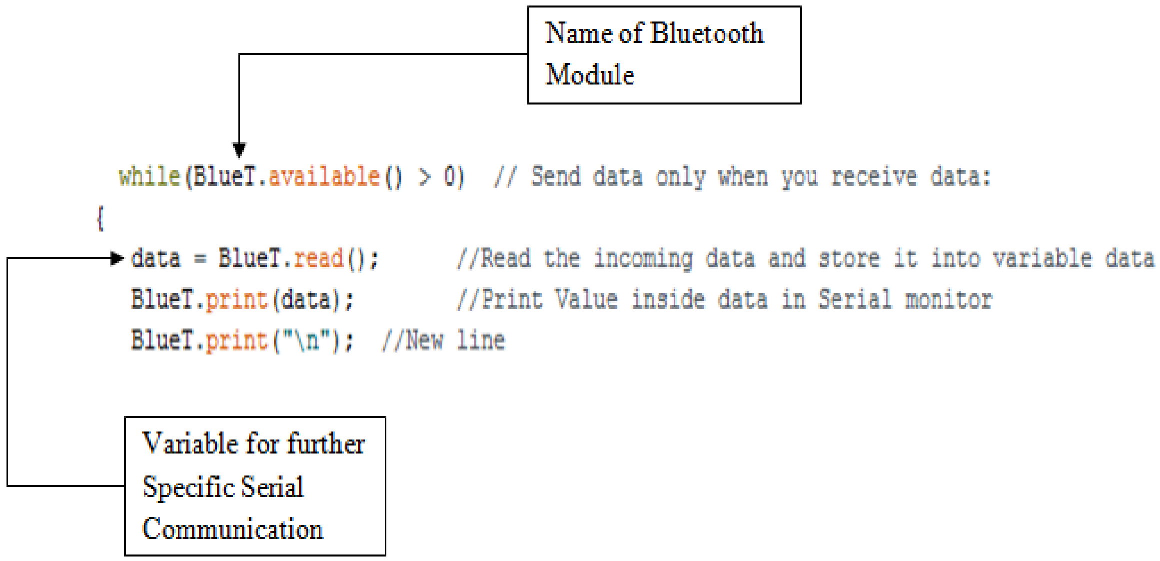

Figure 5.4: Serial Communication Code

The user may then relay a specific command to the robot’s controller. Once a button is clicked on the application, the robot controller will receive a specific piece of text, and react accordingly. Another “While” loop is used for this process. If another piece of text is sent, the “While” loop will break. Figure 5.5 describes the code used.

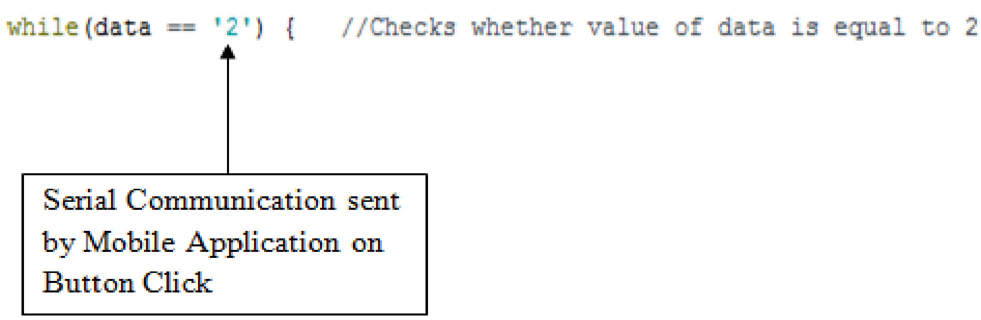

Figure 5.5: Specific Serial Communication Code

The specific text sent is not important, however, both the controller and mobile application must work with the same text for the same instruction.

5.6 Conclusion

This chapter provided the design of the code used for the Arduino. Once the components were physical arranged and connected to the controller, they then were required to be programmed. An algorithm was designed for the controller to relay specific instructions to the robot, depending on the signal received by the sensors. The controller’s code provided the computer system for the mechatronics model.

Chapter 6

Mobile Application Development

6.1 Introduction

The following chapter covers an in depth description of the design and development of the mobile application. The second part if the research question, "**Can the robot system identify each individual human that approaches the work area**", was covered in this chapter. The mobile application serves as personal identification function in the safety system. First, the functionality of the application was studied. An IDE was then selected for the development platform. Each aspect of the application was developed and tested individually, before all components were tied together. The application was then refined overall, for easier use.

6.2 Application Functional Design

The mobile application was designed specifically for Android based devices for simplicity. MIT App Inventor 2 (MIT AI2) was used, as it does not require an extensive knowledge of Java coding, but is still able to perform the required functions. MIT AI2 works on the basis of combining pre-set functions for the program to follow. The application makes up the L2 layer of the security system. It constantly monitors the activity that takes place within the robot workspace.

Should a human worker be required to enter the robot area, they must switch on the Bluetooth function on their personal mobile device, and open the application. The application is only given to official employees. A login screen will then appear, where the user is required to input their personal details. Once the login is accepted, depending on the account, a screen with various functions will appear. The user can then connect to the robot's controller through Bluetooth. Once the Bluetooth connection is authorized, the robot will allow the human to cross L3 without disruption. The robot will then assume a different path plan, as long as the human remains within the workspace.

The human worker can then relay a specific instruction to the robot, via the mobile application. These instructions will need to be pre-set into both the mobile application and the robot controller. Figure 6.1 describes the mentioned scenario.

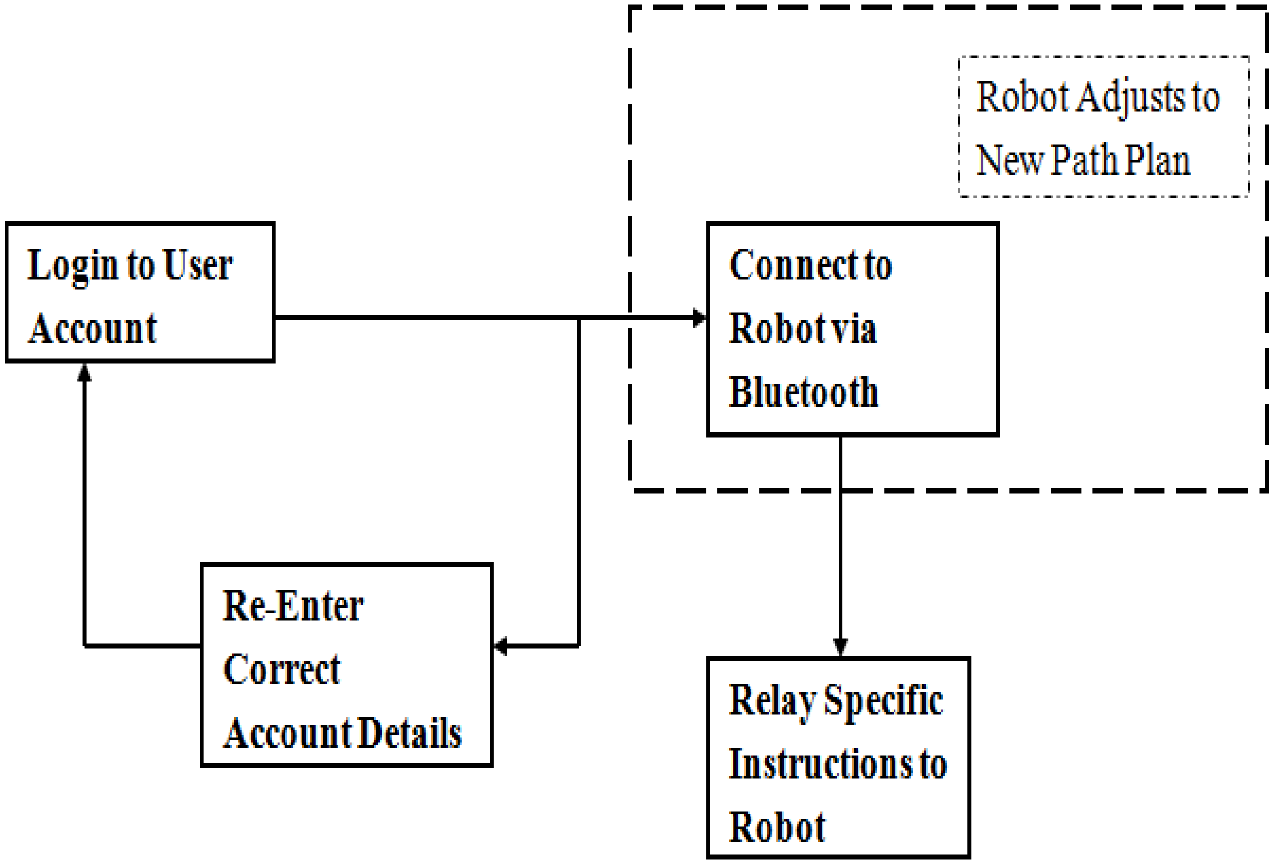

Figure 6.1: L2 Procedure

For the purposes of this study, basic path plans and instructions were coded into the system that was described in the following sections.

6.3 Account Login Design

As mentioned before, the mobile application controls L2 in the security system. The user is expected to open the application on their mobile device, connect to the controller via Bluetooth and receive authorization before entering L3. Receiving authorization is dependent on a number of factors. The main factor is the type of account that the user has been granted. A library was created and stored on a mainframe to manage every worker's account information and partition them into account types. Each account type possesses a level of authorization. Figure 6.2 describes the hierarchy of account types.

Account Library

Type of Account	Privileges
Inexperienced Worker	Only access monitored area upon special request
Experienced Worker	Allowed to enter and control robotic system at will
Plant Manager/Boss	Possesses access to every function
Maintenance Worker	Must follow a specific set of specialised instructions

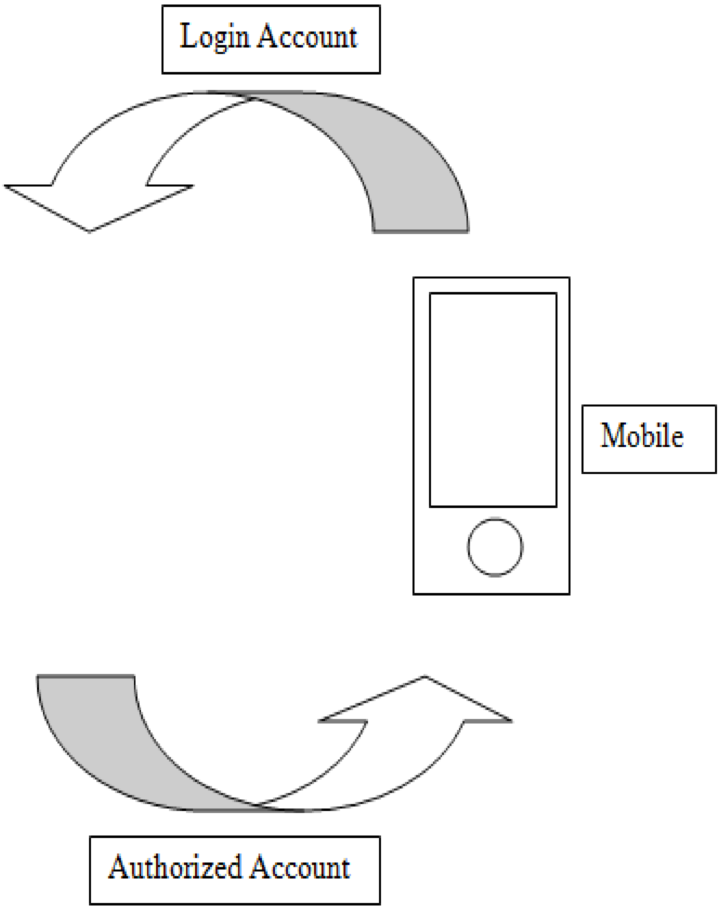

Figure 6.2: Hierarchy System of Accounts

Figure 6.2 shows how each human involved can be recognized individually by the robotic system, depending on their account type. This allows each human worker to possess their own form of personal identification. The partitioning of the various accounts is completely dependent on the specifications on request. Certain privileges are also manipulated per account accordingly, to compliment a specific scenario. Information feedback can also explain which specific individual has been authorized to work in the monitored zone, as well as the type of work they are performing during HRI.

The following describes an example of how the login system was designed for the mobile application.

The user-interface layout follows a basic design. It is comprised of a “Username” and “Password” box for login details. A “Login” button is found directly below, and has to be pushed once the details have been entered. Figure 6.3 describes the layout.

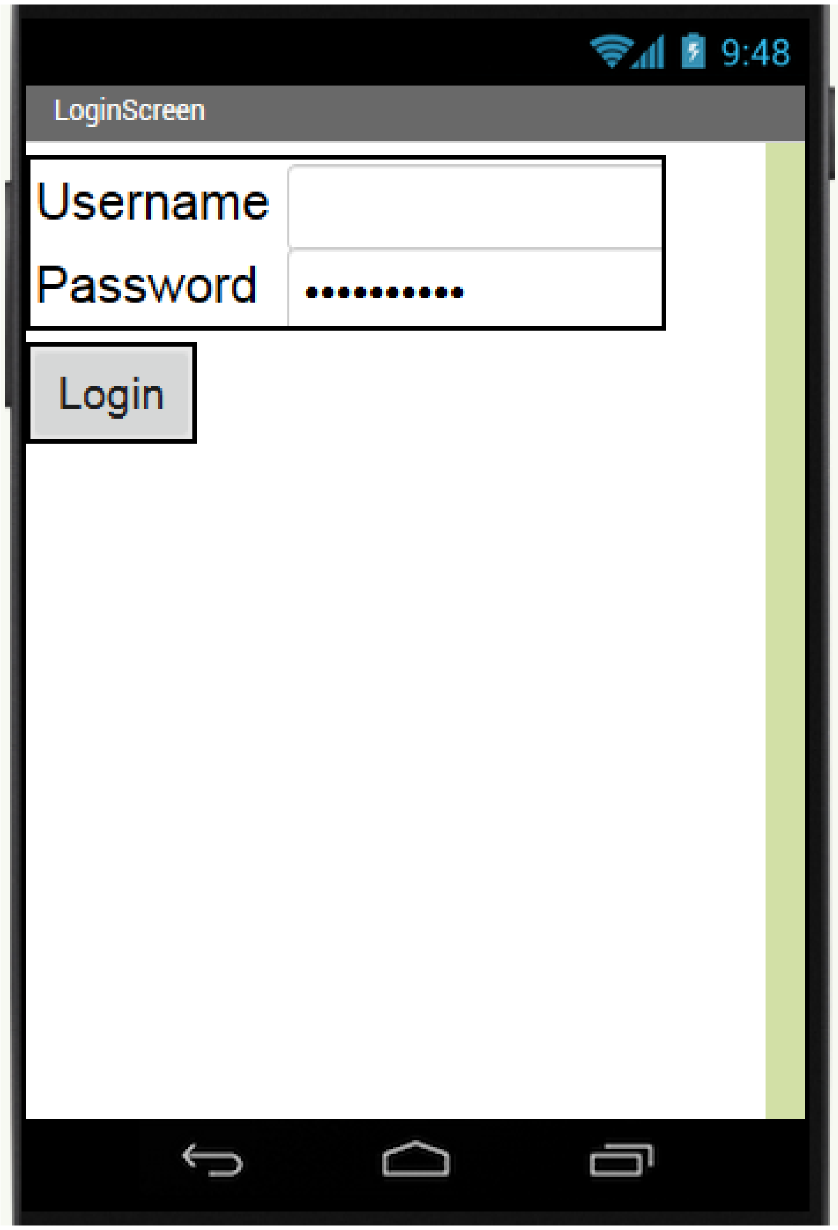

Figure 6.3: User Login Screen

A number of functions on MIT AI2 were used to construct the layout. Figure 6.4 describes the built in components used to construct the login screen layout.

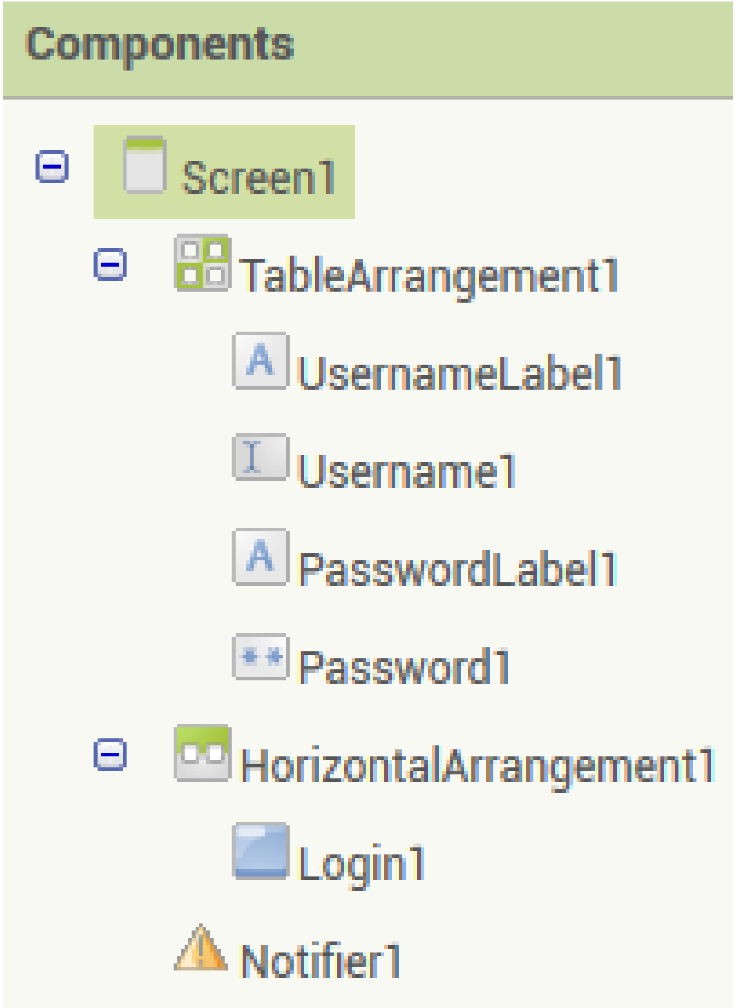

Figure 6.4: MIT AI2 Components

The various components in Figure 6.4 are explained as follows:

- Screen1 – This component refers to the current screen on display, as shown in Figure 6.4.
- Table/HorizontalArrangement1 – These components determine the physical layout of the screen's functions. Various subcomponents, such as button, labels and text blocks can be easily added and move into the correct place. This allows for a neater layout.
- Username/PasswordLabel1 - These components are simply descriptive blocks for the user to read. In this example, they are used to label the "Username" and "Password" blocks.

- Username1 – This component represents a text block, that the user may insert a piece of text, in this example, their username.
- Password1 – This component allows hidden text to be entered, in this example, the user's password.
- Login1 – This component places a button under the password box. The user pushes this button once they have entered in their personal details. This name is labelled as "Login" for visual purposes.
- Notifier1 – This component will trigger should the user input the incorrect details. A small message will appear at the bottom of the screen, notifying the user to re-enter their correct details.

Once the component layout is completed, they must be programmed accordingly. MIT AI2 uses a block diagram system. Pre-set coding functions are combined for the specified output. In the given example, every possible event revolves around the activation of the "Login" button. One of two scenarios may occur once the button is pressed. The user will remain on the login screen and be required to re-enter their information, or they will be granted access to the next screen. The next screen will consist of specific instructions that the user may relay to the robot controller. The user-type will determine what instructions the user will be allowed to convey.

As shown in Figure 5.2 various accounts were be created, depending on the working status of the employee. Accounts can be created, by registering each employee involved. Every employee is partitioned into an account type. These accounts are stored onto a system mainframe. The mainframe partitions the accounts by certain privileges granted. This give each employee a form of personal identification by the system.

In this study, 3 accounts were created, namely the "newguy", "experiencedguy" and "bossguy". These accounts were coded directly into the mobile application, without the use of an external storage system, for simplicity. Upon login, each account is granted a different array of privileges. Once the user has entered their personal details on the login screen, the login button is pressed. The coding for the login button appears in Figure 6.5, as follows.

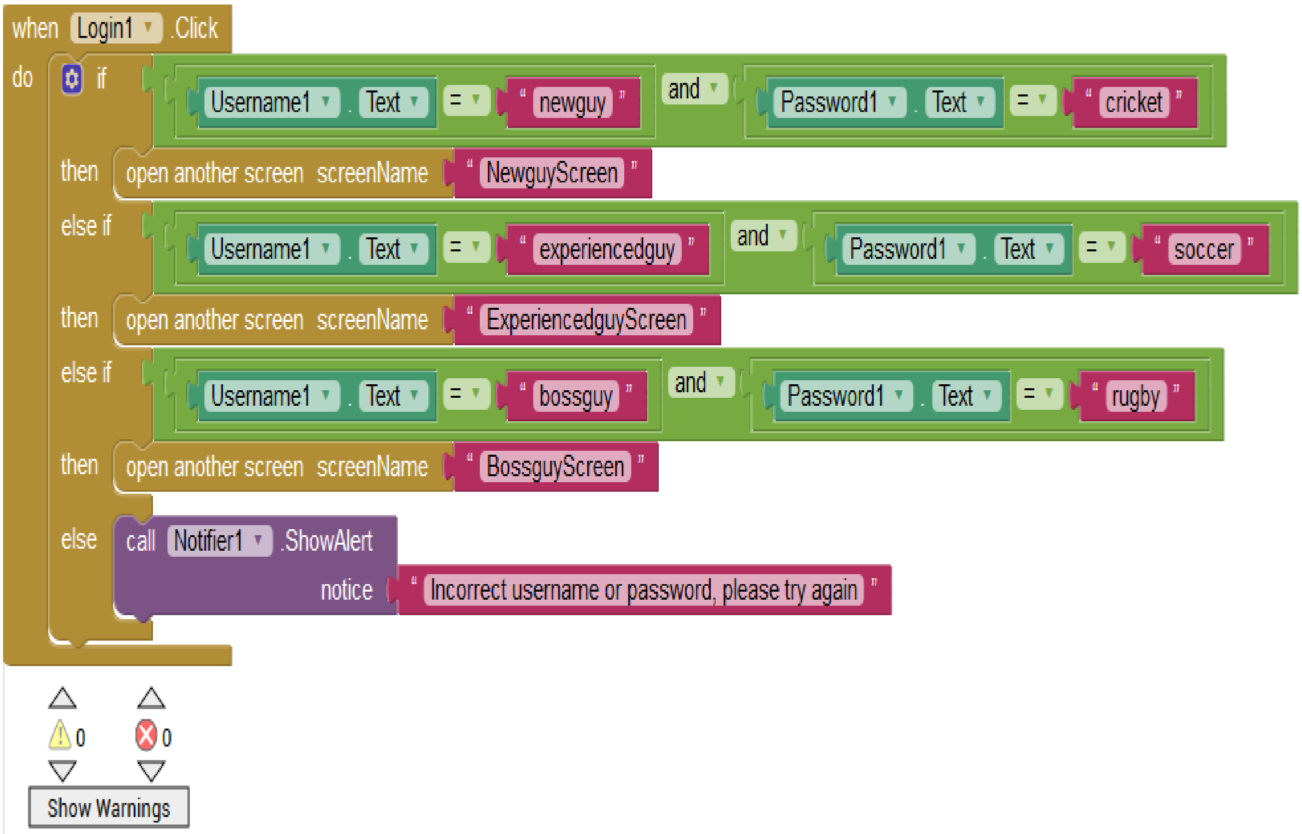

Figure 6.5: Block Code for Login Button

The code activates once the button is pressed. The program must first be notified to activated once the button is pressed or "clicked". Thereafter, a series of "If-Else" statements is processed, to predict a specific scenario. The first 3 "If-Else" statements refer to the 3 possible account selections. Each account possesses its own unique username and password. These components must be joined with an "and" command, in order to avoid combining another username with another password. If the account details match up to a specific scenario, the user will be granted access to one of the 3 screen selections listed in the block code. If the system cannot recognise the correct details, the "Else" statement points to a notification, that will inform the user to enter the correct details. The code proved successful.

6.4 Bluetooth Connection

When the authentication has been granted, the user will be granted access to a new screen. Every user-type screen will have the same initial function, and that is to connect to the robot system using Bluetooth. The user must make sure that the mobile devices Bluetooth feature is activated, before opening the mobile application. The first action the user must do is connecting to the robot's controller. The connection design proved successful.

The following example uses the "NewguyScreen" stated in the previous section. As stated before, only the specific user may have access to this screen. Figures 6.6 and 6.7 describe the layout design of the screen.

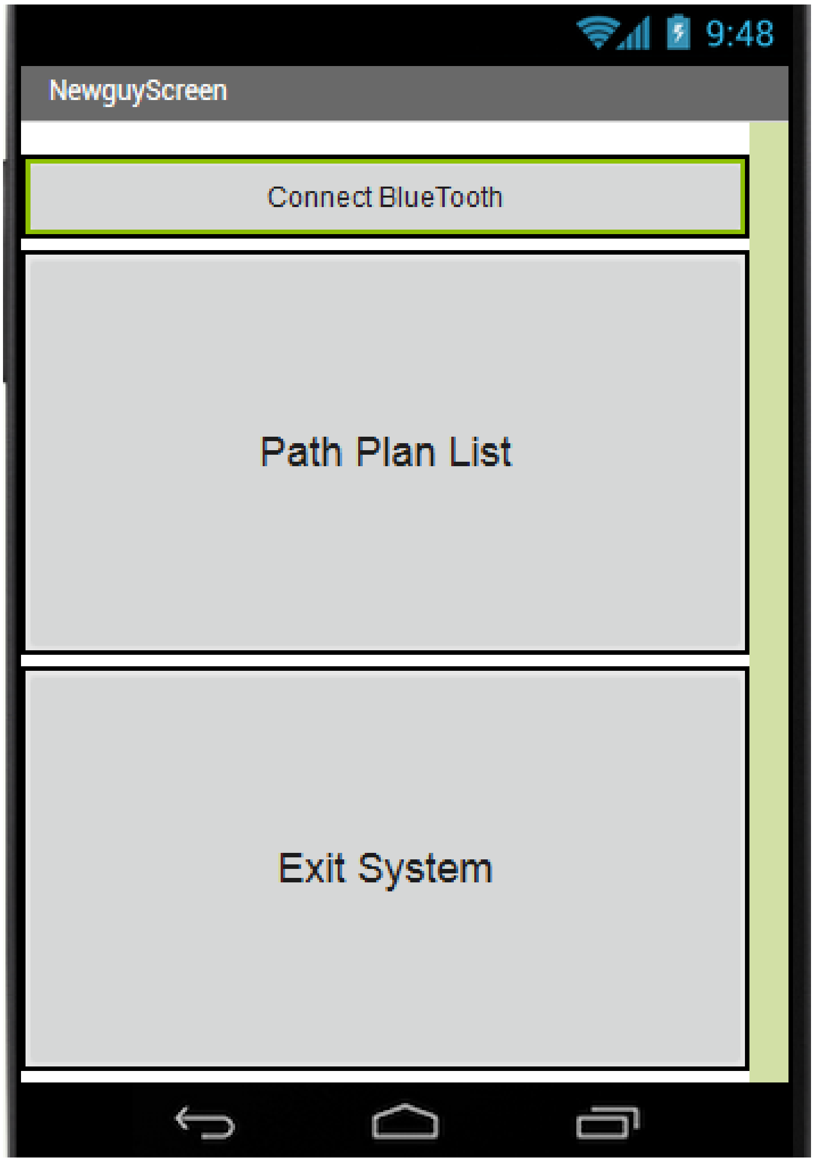

Figure 6.6: NewguyScreen Layout

The “Connect BlueTooth” button is the focus of this section. This button is not the same one used in the login screen. The icon is a pointer to a list.

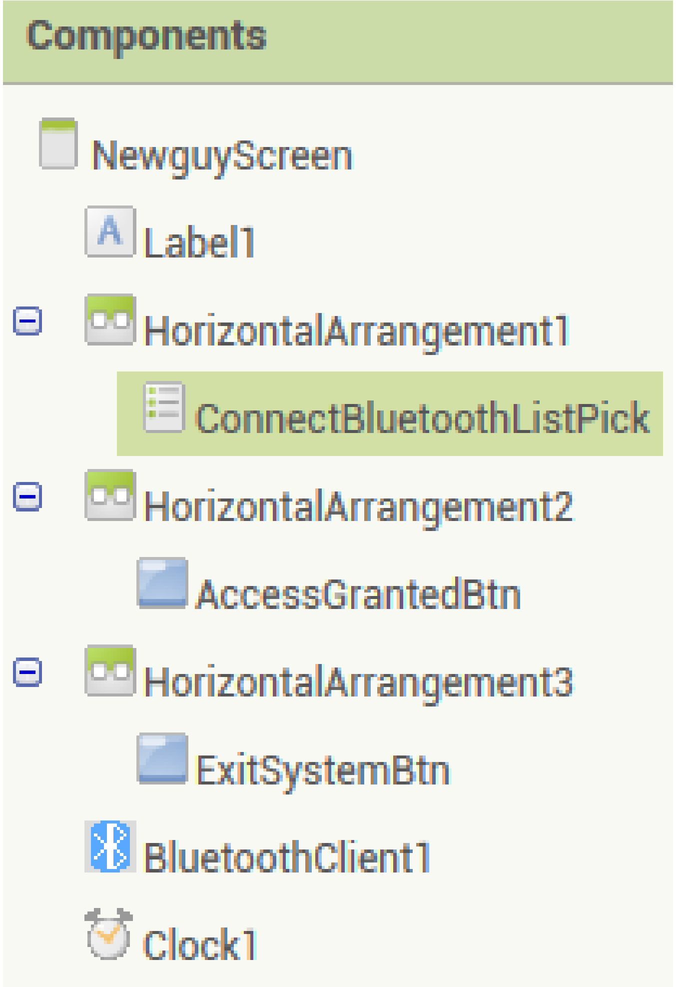

Figure 6.7: Bluetooth List Component

For the purposes of the Bluetooth connection, the application must tap into the mobile device’s personal Bluetooth feature. The Bluetooth list component must point to the mobile device’s Bluetooth pairing list, to select an array of possible devices to connect. Once the list is available, the user must select the robot controller. In this study, the HC-06 module will appear. The Bluetooth client component is then called to connect to the module. A clock component is used in the block coding, to change the display of the Bluetooth icon, to represent the connectivity state. Green will represent a connect, and red represents a disconnection. Figure 6.8 describes the connection method. These functions were implemented into the system.

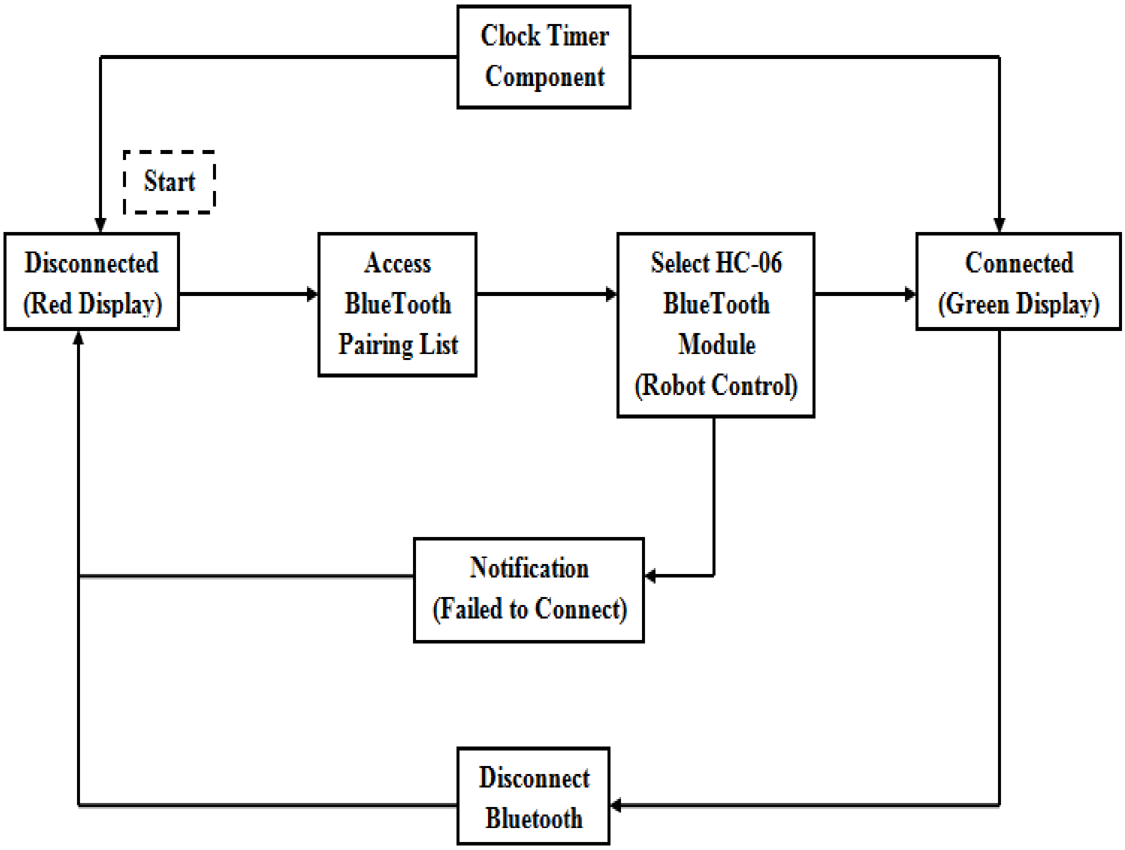

Figure 6.8: Bluetooth Connection Method

In terms of the MIT AI2 block coding, the Bluetooth list and clock components must be programmed. Figure 6.9 describes the basic layout of the programme.

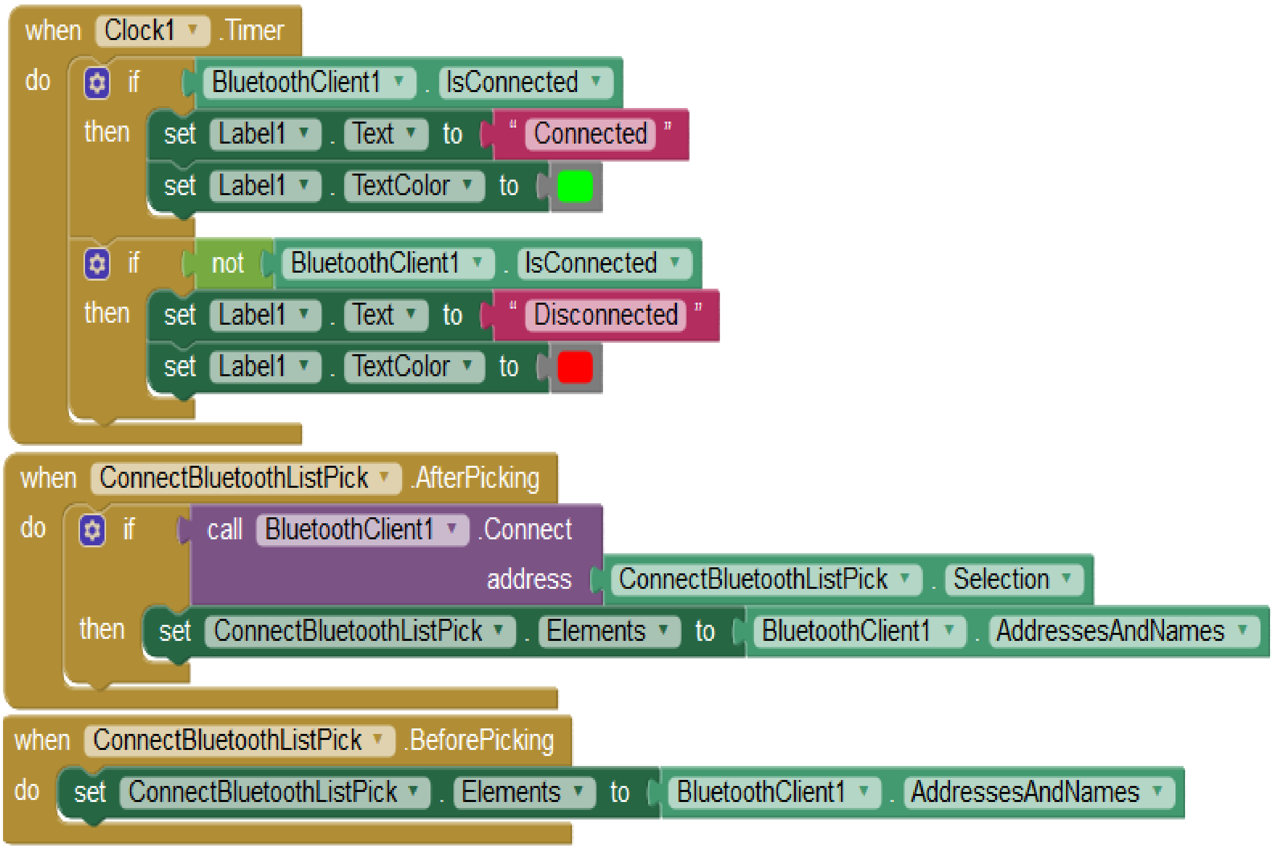

Figure 6.9: Bluetooth Connection Block Code

The clock component is directly connected to the mobile device's internal clock. This is used to activate at regular timed intervals. A check is performed each time to determine the connection status of the Bluetooth. A series of "If-Then" functions are used to set the colour of the Bluetooth label to represent the connection status. The list function points to the phone's Bluetooth pair list. Upon the initial click, the list is displayed. When the selected device is chosen, the application's Bluetooth client is activated and the connection is made. A separate button is made to disconnect from the controller unit.

6.5 Path-Plan Selection

After the Bluetooth connection has been maintained, the path-plan screen may be accessed. This part of the application is used to directly control the robot for a specific scenario. Depending on the scenario, specific movements need to be programmed to the mobile application as well as the robot controller. In this study, a control was setup to manipulate each motor. Figure 6.10 shows this example.

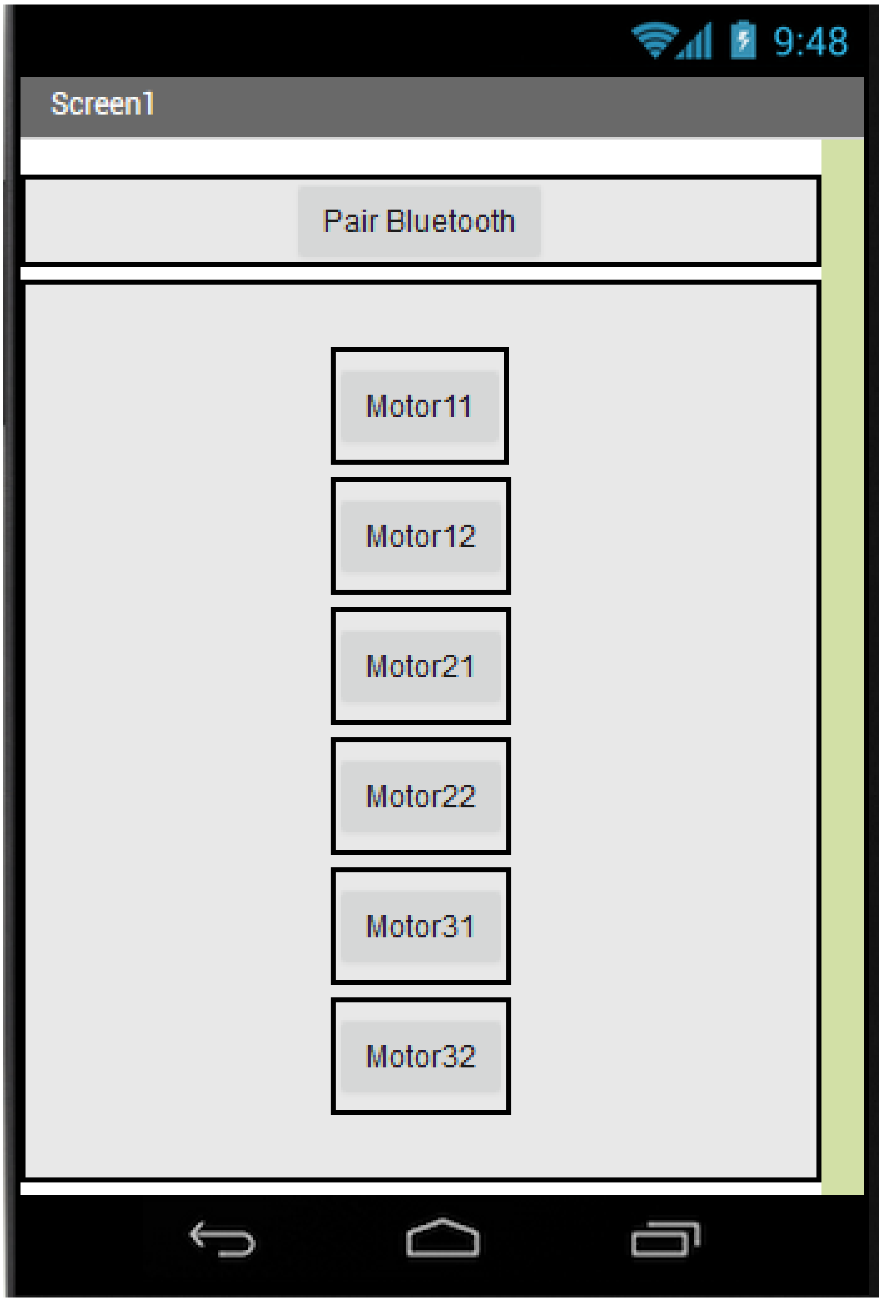

Figure 6.10: Path Plan User-Interface

The buttons on display directly control 1 of the 3 motors on the robot arm. The first number refers to the specific motor, from the "shoulder" to "elbow" to "wrist" respectively. The second number refers to the movement directly, by a certain degree range. The number 2 will return the motor to its original position. The motors can be instructed to move in any specified order.

Figure 6.11 shows the block programming for each individual button.

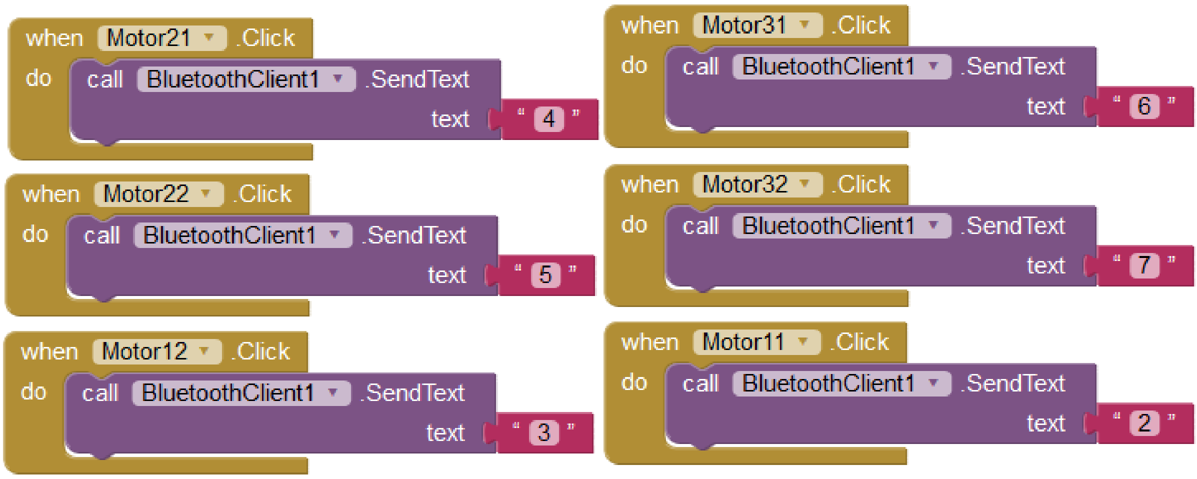

Figure 6.11: Block Programming for Robot Motors

Each motor button sends an individual piece of serial text to the robot's controller, in this case, the Arduino Uno. Every button must send a different text, in order to avoid an overlap of instruction. The specific text piece is irrelevant; however, the controller must identify that specific piece of text to the required instruction.

6.6 Conclusion

The development of the mobile application was broken into 3 sections, namely the account login, design of the Bluetooth connection, and relaying commands to the robot. The account login introduces a hierarchy system, where different accounts are given different privileges. This creates a form of personal identification, for the robot to identify a specific person. The user must then connect to the controller, using the Bluetooth function, where permission will be granted for the user to enter the workspace. Lastly, the user may relay any specific commands to the robot, depending on the desire situation. The mobile application provided the control system for the mechatronics model.

Chapter 7

Experimental Troubleshooting & Testing

7.1 Introduction

During the construction of the physical system, the various sensors and motors required a specific connection and programming, in order to function as a single unit. Each component was added individually and tested for the correct functionality. Once the components were all connected, further testing was performed in order to find the optimum method for the system's performance. Various errors were made that either caused harm to the physical components, or produced no output action, and should be avoided.

The following sections describe the main errors found during the testing performed. A description of the system as well as a solution is provided.

7.2 MG90s Servo Motor Connection

Three servo motors were required to construct the robot arm. The servo motors used required 3 connections each. A VCC and GND pin, as well as a signal pin for instruction relay. Standard Arduino VCC pins have an output of 5V. This is enough to provide power for 1 servo. Most online tutorials display diagram that show multiple motors connected to a single Arduino controller. This must be avoided.

During the experimental process, an attempt was made to try and run all 3 motors off the same Arduino VCC pin, in a parallel setup to receive the same voltage. This must be avoided for the following reasons:

- The Arduino draws too much current from its personal power source
- The motors do not operate as intended from lack of power
- Damage to the motors occurs from too much current

All scenarios were experienced. The Arduino board was damaged and required a replacement. A single motor was damaged and did not operate at all under future tests. The motors did not output the specified torque for the joint rotation.

The solution was to install a servo shield. The shield provides personal connections for each motor. Up to 8 motors can be connected and safely run off the same power source initially used. The Arduino was powered using a USB cable connected to a laptop computer. The servo shield also provides pin connections for the Bluetooth module, but was not used. The other sensors were also connected to the controller using the servo shield.

7.3 Proximity Sensor Coding Check

The proximity sensor, placed directly onto the robot, prevents the user from coming into direct contact with the robot. The sensor constantly monitors the human's physical distance from the robot. The Arduino is responsible for both the actions of the robot arm, as well as controlling the proximity sensor. Unfortunately, the controller can only process one signal at a time, as it follows an instruction per line of code.

The servos are controlled using the "For" loop. To simulate a simple pick and place action, 6 "For" loops are required in the controller code. Two loops are required for each motor to instruct an initial and return movement. The specifications such as the motor speed and maximum rotational angle can be altered through the loop. The loop moves the motor 1 degree per the specified interval delay.

An experiment was performed in order to incorporate the proximity sensor's code into the controller. On the first attempt, the proximity sensor's code was put after the 6 motor movements were made. This created window of down-time for the sensor, while the robot performed its movements, raising a question of safety. The time taken for one robot movement cycle was required. The following parameters were used before the calculations were performed:

- The delay in the motor loops was set to 40ms (acceptable robot movement speed)
- The rotation of each motor is set from 0 to 40 degrees

The time taken for the motors complete rotational speed can be calculated using the following formula:

$$T_M = \frac{T_D \; x \; (F_{\propto} - I_{\propto})}{\propto_i}$$

Where

T_M = Total time taken for the motor to complete its movement

T_D = Time set for the loops delay

$F_{\propto}$ = Final angles position

$I_{\propto}$ = Initial angles position

$\propto_i$ = Increment of the “For” loop

The following parameters can be found using the values from Figure 5.2. The time for a single motors movement is determined as follows:

$$T_M = \frac{40 \; x \; (40 - 0)}{1}$$

$$T_M = 1600 \; milliseconds = 1.6 \; seconds$$

If the proximity sensor’s code is placed at the end of the 6 loops, in the described scenario, the down-time would be roughly 10 seconds. This is far too big a window, as this allows time for hazards to occur before the system can recognise a potential threat.

The code was then placed inside each ‘For” loop, and was designed to perform a check after every iteration of a motor angle. This proved unsuccessful for unknown reasons. The motors nor the proximity sensor would function as intended. Speculations would suggest that the computational power of the Arduino could not handle the 2400 proximity checks in 10 seconds, while controlling the motors at the same time.

The solution was to place the code for the proximity sensor after each complete “For” loop. Reducing the number of checks performed from 2400 to 6 per complete cycle. The total down time between each check is also reduced to 1.6 seconds, as previously calculated. This appears to be the most acceptable method as the system will perform a security check after each complete joint movement.

7.4 Bluetooth Serial Communication Code

Once the Bluetooth module and the mobile device has established a connection through the application, the controller must recognise the communication. Various loops were tested to see which provided the easiest control. The “While” loop proved to be the most successful. A layered design of “While” loops were used. The outer most loop activates once the user logs into the application, and requests access permission to the controller. This initiates serial communication. Once the controller recognises the user, further instructions can be issued to the robot upon request, as shown in the above sections. When a new command is issued, the inner loop while breaks and activates a new one. Figure 7.1 describes the code used as described.

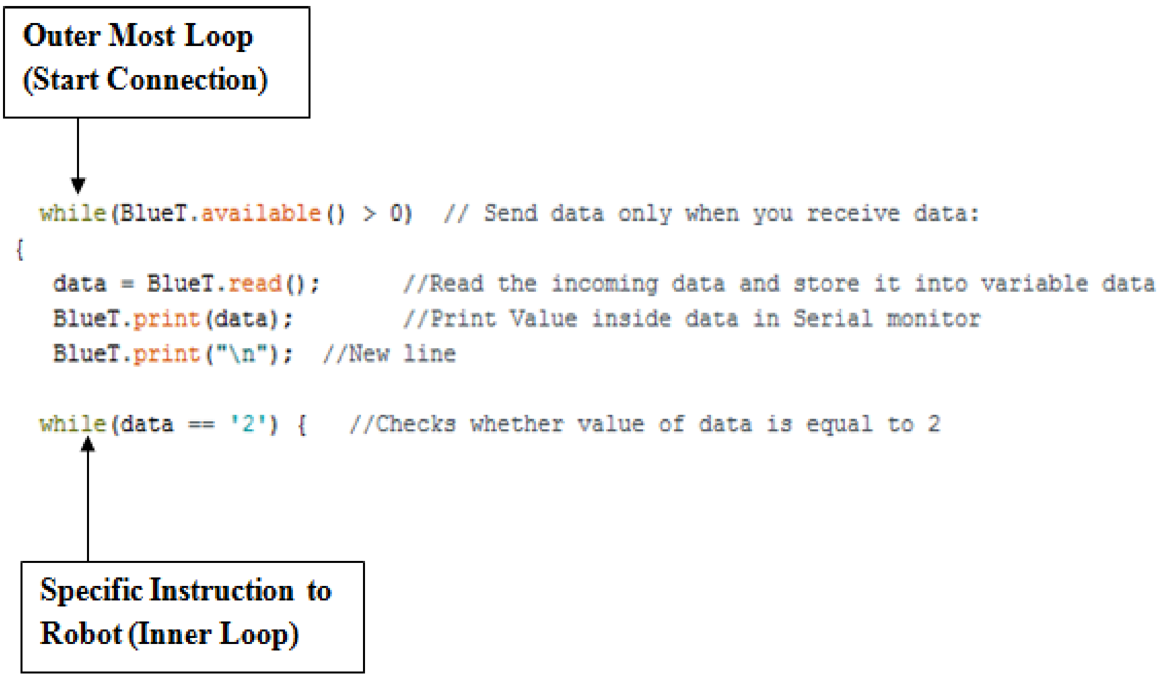

Figure 7.1: Serial Communication Code

When programming the mobile application, the user must be weary of ASCII conversions between the Arduino controller and interface used to programme the mobile application. MIT AI2 uses the exact text entered into the block programming. Upon using other IDE's, an ASCII number conversion was required.

7.5 HC-06 Bluetooth Module Connection

The Bluetooth module has 4 pins that require a connection. The standard instructions state to connect the VCC and GND pins to their respective slots on the Arduino. The module requires a 5V input, which is the standard output of the Arduino VCC pin. The TXD and RXD pins on the module must be connected to the inverse pins on the Arduino. Ie. This RXD pin on the module must be connected to the TXD pin on the Arduino. Switching these pins will result in a transmission signal not received by the next device.

It is not advised, however, to connect these pins to the Arduino's receiver and transmitter slots. Throughout the experimental process, the Arduino's code is constantly changed and must be re-uploaded. If the TXD and RXD pins are occupied on the controller, the code will not upload. The user will have to constantly remove and insert the Bluetooth module's communications pins between uploading the code. To avoid this, separate pins are re-assigned to simulate new communication pins for the Bluetooth module. Figure 7.2 shows an example of the code used.

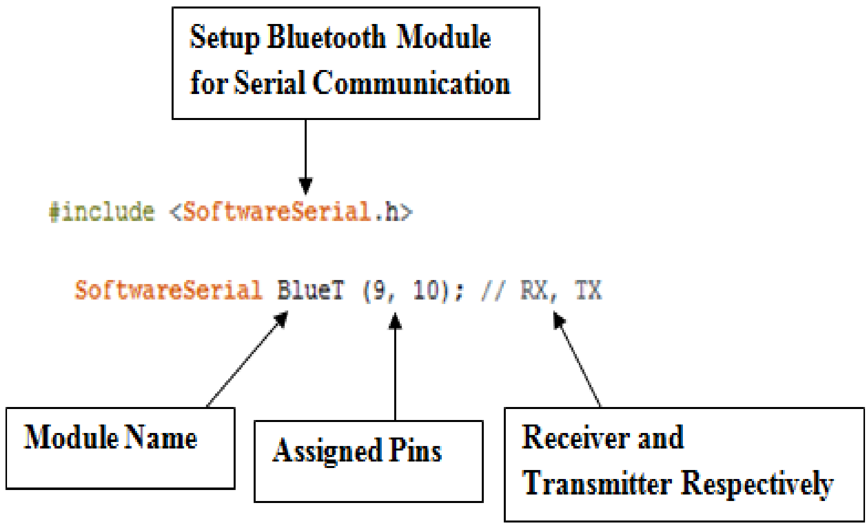

Figure 2: Set New Arduino Communication Pins

The code must be applied before the declaring of any variables before the “void setup()” loop begins.

While connecting the communication pins to their news assigned slots, the RXD pin on the Bluetooth module must be connected with caution. The receiver pin on the module cannot receive a signal greater than 3.3V. Standard Arduino pins output a 5V signal. A voltage bridge as described in Figure 4.6 and Table 4.1 must be used before using the module. During experimental testing, the module was connected directly to the controller, and displayed various errors whilst trying to establish a connection with the mobile device. It is safe to assume the module was damaged. A new module was used using the connection mentioned above, and functioned as intended.

7.6 Conclusion

During the construction of the security system, refinement and optimisation of the system was constantly required. Each component was added to the system and tested for the required operation. The corrections and refinements came from an experimental procedure, and were not present in the components basic instructions or tutorials. The initial results of the first experiments resulted in damages to electronic components, as well as the system not operating as intended. The methods that resulted in damages to components should be avoided at all times.

Chapter 8

Discussion & Validations

8.1 Discussion

The purpose of the study was to investigate the possibilities of creating a flexible safety system to promote HRI in a robotic manipulator environment. The study involved the construction of a test rig, in order to perform various experimental methods, to test to the validity of the core idea. In the topic overview, the study was suggested to revolve around the model of mechatronics, which combines various fields of engineering. In the literature review, the field of mechatronics was investigated, and figure 2.1 describes the overall model. A number of aims and objectives were then listed in order to guide the study.

The following sections will investigate how these aims and objectives were met in order.

- Current safety standards were investigated, as well as standard ISO safety regulations around industrial robots. This study provided an in depth insight into how various potential risks should be identified, in order to design the SIS for specific hazard reduction.

- A robotic arm was 3-D printed, based on the design of the Little Arm 2C. This would serve as the test rig for future experimentation.
- Other robot safety systems were investigated, the most noticeable one being the ABB Safemove. The system utilizes proximity cameras, which determine a human's physical presence from the robot. The robot will then adjust its operating speed and position accordingly. The proposed system was inspired by this design. To reduce the extreme cost for such a design, ultrasonic sensors and electronic proximity mats, as well as a Bluetooth module sensor were used for the study. These sensors provided sufficient accuracy for the study.
- The development of the mobile application involved the investigation of which IDE to use. IDE's such as BlueJ and Java were attempted to develop the application, but proved too great a learning curve to use. MIT AI2 was found to be the most convenient, as in-depth knowledge of Java Script syntax was not required. The IDE made use of a block diagram coding system. The mobile application was developed to communicate directly with the robotic system, as well as provide the user with a personal identification when entering the robotic system workspace.
- The sensory systems were then synergized into a single unit, and installed onto the test rig. Experimentation was then performed, to refine the system. Various errors were made during the physical wiring of various components, such as the Bluetooth module and servo motors. These errors were rectified and the system was optimized. Thereafter, the user interface for the mobile application was improved to provide greater functionality. The system is still subject to specific loopholes, which require further experimentation to close. Overall, the core idea of the system proved successful.

Upon combining the 4 systems developed, into a single unit, the mechatronics model was completed for the study. Figure 8.1, below, describes the model with respect to this specific study. The model closely represents that in figure 2.1.

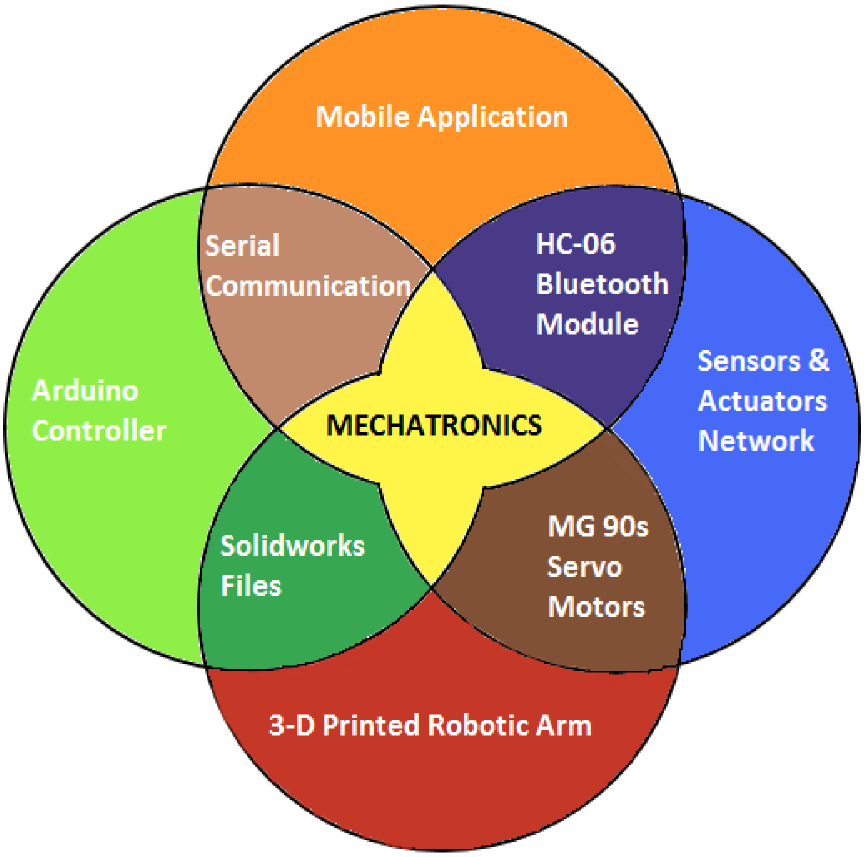

Figure 8.1: Design's Model of Mechatronics

Through the development of the synergized system, the research question, provided in chapter 1, was investigated. The possibility of creating a flexible safety environment does exist. The study shows that the core idea of installing virtual boundaries around a robotic system may provide the required safety according to ISO standards. This enables a higher level of HRI. The second part of the problem statement questions the robot's reaction to a specific individual. The system was able to identify an individual based on their personal mobile device. The robot's safety checks were reduced to occur every 1.6 seconds, during the experimental phase. This provides an acceptable window of time for the robot to identify the human in proximity. The study proved successful according to the problem statement; however, there are still refinements and optimizations that are required in order to test the core idea on a full scale model.

8.2 Validation

The following section lists and discusses the initial research questions. An attempt will then be made to answer these questions, through the substantiation of the study. This will also include the results achieved in various sections.

The first question investigates the financial aspect of developing a test rig for the study. Is it possible to construct a low-cost rig from easily accessible components? The cost of the entire test rig amounted to less than R1500. This cost includes the purchasing of the SolidWorks files for the 3-D printing of the small-scale robot arm. The components listed and required in the design chapter can be purchased from any DIY electronic store. The stated cost does exclude the components that were damaged and replaced during the experimentation phase. Other pieces of equipment, such as the computer used for programming the controller, as well as the mobile device used, were also excluded from the cost. If the robot was originally designed and not purchased, as well as the cost of the PLA excluded, the overall cost of the rig can be reduced to less than R1000. This is found to be reasonably cheap for an automated robot arm, that can be manipulated for experimental purposes.

The second question investigates whether a mobile device can serve as a form of personal identification, similar to an ID card or tag. It also asks whether the robot can identify each person through this form of identification. In the mobile development chapter, a "login" system was created. This login system holds every user's personal identification. After the user logs into their account, they can request permission to enter the robot's area. Those that do not have access to the area will not have the option available upon login. This is controlled through a main server that runs the hierarchy system, described in Figure... The system does prove successful in this regard, however, there does exist a number of loop-holes. The user's involved may share account login details, or trade mobile devices in order to access certain privileges. While these possibilities remain, it is safe to assume that people are less likely to share personal mobile devices, as opposed to work key-cards. As for the account details being shared, this still remains to be a problem in a multitude of fields. While this system displays these loop-holes, it is safe to say that it is just as effective as most security systems in place. The loop-holes may be closed by installing more personal features, such as finger-print readers and retinal scanners. The core idea of the system will still remain.

The third question takes into account the overall safety of the proposed system. The sensors used appeared to operate as intended, and all signals received by the controller were without delay. Once a safety condition was breached, the system came to an immediate hat, ensuring the safety of those involved. A major concern was with respect to the immediate proximity sensor placed onto the robot. After experimenting with the controller's code for both the sensor and the robot motors, it was noticed that a duration was present were a safety infringement could happen without being noticed by the system. After a refinement process, the duration of the possible infringement occurring, without the system recording it, was reduced to 1.6 seconds. This appeared to be reasonable, after taking into account the operating speed of the robot. It was also found that as the operating speed increased, this duration would decrease, allowing for a safer system.

The final question investigates the limitations of the proposed system. A number of limitations were found through both simplification and the design of the system. The system could not be installed onto a full scale industrial robot, as the robot came with its own personal software, and was not open to be tampered with, according to the manufacturer. The system required an open source controller to test the core idea. The system also assumes that the user's mobile device's battery is always charged, as the system is totally dependent on the application. The final assumption is that one user is allowed to access the workspace at a single point in time. The system was not able to identify more than one user, and will have to be controller through another party.

Chapter 9

Conclusion

9.1 Conclusion

The research study has provided an alternative approach to HRI. By removing the physical barriers found in most industrial workspaces, the user may collaborate with a robotic system during an assembly of a product. The design of the proposed system was developed, and was tested through the construction of a small-scale simulation. Once the electronic components were purchased, a robotic arm was 3-D printed. The final system was then assembled and ready for experimentation. An Arduino controller was used to automate the robot arm, as well as process all the signals sent by the various sensors. The application was made to collaborate with the constructed system. Through the connection of various electronic components, some were damaged in the experimentation process. These methods were listed and must be avoided in future work. Thereafter, a mobile application was developed using MIT AI2. The application used the mobile devices Bluetooth module, to send instructions directly to the robot controller. The application's functionality ranged from allowing the robot to identifying the user, as well as relaying specific instructions to the robot as required. The overall system proved to be low-cost, and provided an excellent platform for both experimentation and demonstration of the security system. The system constructed did contain limitations; however, these limitations can be easily removed once the proposed idea is translated to a full-scale design. The core idea of the system was achieved, through the development of the mobile application, as well as the construction of the small-scale model.

9.2 Summary of Contributions

This research study has provided contributions to both electronic and software engineering, as well as the vast field of HRI. The contributions made to electronic engineering from the study were the design and implementation of the electronic components for this specific functionality. The specific component models were listed, as well as the controller coding for these specific sensors. The biggest contribution was with respect to the physical wiring of certain components. Certain instructions by the manufacturers were not clear, and lead to damages in the electronics. A descriptive scenario was explained and should be followed when using these components for other applications. An original Arduino code was also developed and can be applied to many of the individual sensors.

The contributions made to software engineering were through the development of the mobile application. Various IDE's were investigated, but most required an intense knowledge of Java coding. It was found that MIT AI2 allows the development of mobile applications, without in-depth experience. A tutorial was provided on how the mobile application was developed, as well as connecting the device to the robot controller through Bluetooth. This Bluetooth connection is not only limited to robotics, and can be used in an array of various projects. This flexible, Bluetooth connection, method can be used on any modern day Smartphone, with an Android operating system. A method of relaying specific instructions to another Bluetooth device was also covered, as can be used to other projects as well.

The HRI contribution provided a unique method to employ safe interactions between a human and a robotic system. Modern day HRI does include the use of light curtains and cameras, where as this solution uses a mobile phone to serve as the safety control. By investigating this method, it has opened a field of study that investigates whether using a mobile phone is a viable method of identification for workers in an industrial setting. The study provides an example of employing a hierarchy system, where the activity of every worker can be constantly monitored, allowing for a new form of safety control for the factory floor.

9.3 Suggestions for Further Research

HRI is an extremely wide-spread field, where only specific scenarios can be analyzed at a single point in time. Upon further refinement of the system, more opportunities and ideas will arise. Each area discussed in this study can be improved upon, to achieve results that are more efficient.

Added layers may be incorporated into the proposed system, in order to improve the overall safety. The proximity mat used can be replaced with a larger mat to cover a bigger area, as well as one that can identify when more than one person has stepped into the area. The electronics used were cost efficient, however larger and more accurate sensors can be interchanged.

The core idea can then be looked to be installed onto a full-scale model. This will require a different strategy to programming the controller, as well as the connection of a mobile device. The mobile application can also be improved upon by linking it to an external server, in order to efficiently partition the different users into the suggested hierarchy system.

Lastly, combining the system with accurate path-planning kinematics will produce results that are more accurate, in comparison to the random movements used in this study.

Bibliography

[1] K. Dautenhahn, “The Encyclopedia of Human-Computer Interaction, 2nd Ed,” 2007. [Online]. Available: https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-robot-interaction.

[2] C. Clark, “Keep The Robot In The Cage—How Effective (And Safe) Are Co-Bots?,” 2019.

[3] T. Anandan, “Human-Robot Interaction: A Team Sport,” 2019.

[4] S. Ashley, “Getting a hold on mechatronics,” *Int. J. Eng. Educ.*, p. 119, 1997.

[5] J. Mynderse, A. Gerhart, L. Liu, and S. Arslan, “Multi-Course Problem-based Learning Module spanning across the Junior and Senior Mechanical Engineering Curriculum: Mechatronics, Fluid Mechanic, and Heat Transfer,” 2015.

[6] J. Vallverdu, *Handbook of Research on Synthesizing Human Emotion in Intelligent Systems and Robotics*. 2014.

[7] G. Piumatti, M. L. Lupetti, and F. Lmberti, “Human – Robot Interaction,” *Encycl. Comput. Sci. Technol. Second Ed.*, vol. 34, no. 1, pp. 487–496, 2017, doi: 10.1561/1100000005.

[8] K. Dautenhahn, “Methodology & themes of human-robot interaction: A growing research field,” *Int. J. Adv. Robot. Syst.*, vol. 4, no. 1 SPEC. ISS., pp. 103–108, 2007, doi: 10.5772/5702.

[9] K. Jani, “Report on standard safety practices for Industrial Robots,” 2016. doi: 10.13140/RG.2.2.20769.61285.

[10] T. Malm, “Guidelines to make safe industrial robot systems,” 2017. [Online]. Available: http://www.vtt.fi/inf/julkaisut/muut/2017/VTT-R-01109-17.pdf.

[11] C. Y, “Robot Safety: Overview of Risk Assessment and Reduction,” *Adv. Robot. Autom.*, vol. 05, no. 01, pp. 1–5, 2016, doi: 10.4172/2168-9695.1000139.

[12] M. Vasic and A. Billard, “Safety Issues in Human-Robot Interactions.”

[13] S. Honig and T. Oron-Gilad, “Understanding and resolving failures in human-robot interaction: Literature review and model development,” *Front. Psychol.*, vol. 9, no. JUN, p. 2, 2018, doi: 10.3389/fpsyg.2018.00861.

[14] American National Standard, “Ansi/ria r15.06 - an introduction to Robot and Robot System Safety,” 2015.

[15] Instrumentation Control, “Taming the robot: breaking down the barriers between man and machine,” 2007.

[16] ABB Robotics, “SafeMove Robot Safety Option SafeMove – new generation robot safety SafeMove.”

[17] B. Löfvendahl, “Augmented Reality Applications for Industrial Robots,” 2014.

[18] ABB Robotics, “SafeMove2: Product Overview,” 2017.

[19] A. Robotics, “SafeMove2: Protecting operators & enhancing robot safety.”

[20] D. Liu, R. Wang, K. Yao, X. Zou, and L. Guo, “Design and implementation of a RF powering circuit for RFID tags or other batteryless embedded devices,” *Sensors (Switzerland)*, vol. 14, no. 8, 2014, doi: 10.3390/s140814839.

[21] M. Casden, “Passive Remote Programmer For Induction Type RFID Readers,” 2001.

[22] R. Want, “An Introduction to RFID Technology,” *IEEE Pervasive Comput.*, pp. 25–33, 2006, doi: 10.1017/CBO9780511623806.

[23] S. Willis and S. Helal, “A passive RFID information grid for location and proximity sensing for the blind user,” *Univ. Florida Tech. Rep.*, no. March, 2004, [Online]. Available: https://www.cise.ufl.edu/tr/DOC/REP-2004-355.pdf.

[24] A. Kalnoskas and A. Pandey, “How do RFID tags and reader antennas work?,” 2017. [Online]. Available: https://endtimestruth.com/mark-of-the-beast/rfid/.

[25] S. Smiley, “Active RFID vs. Passive RFID: What’s the Difference?,” 2016.

[26] H. Lehpamer, *RFID Design Principles*. 2007.

[27] M. M. Scheunemann, K. Dautenhahn, M. Salem, and B. Robins, “Utilizing Bluetooth Low Energy for human-robot interaction,” 2016.

[28] R. Pahuja and N. Kumar, “Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller,” vol. 2, no. 7, pp. 14–17, 2014.

[29] M. Yan and H. Shi, “SMART LIVING USING BLUETOOTH- BASED ANDROID SMARTPHONE,” vol. 5, no. 1, pp. 65–72, 2013.

[30] O. Ogorodnikova, “Human Robot Interaction : The Safety Challenge (An integrated frame work for human safety),” 2010.

Appendix A

Arduino Code

This appendix is used to show the original Arduino code developed to process all the signals from the various sensors, and then for the Arduino to relay an instruction to the robot, based on these signals.

```
#include <Servo.h>
#include <SoftwareSerial.h>

  SoftwareSerial BlueT (9, 10); // RX, TX

  Servo servo;

  int buttonState = 0;
  int angle = 0;
  int MotSpeed1 = 10;
  int MotSpeed2 = 80;
  int MotSpeed3 = 40;
  int buzzer = 7;
  int trigPin = 6;
  int echoPin = 5;
  int ProxMat = 8;

  float duration;//Ultrasonic sensor reading
  float distance;//Ultrasonic sensor reading

  int MinDistance = 15;//Proximity Sensor Minimum Distance in cm
```

```
  char data = 0;//Bluetooth data

void setup() {

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(ProxMat, INPUT);

  servo.attach(2);
  servo.write(angle);

  Serial.begin(9600);
  BlueT.begin(9600);
}

void loop() {

  //L3 check start

   buttonState = digitalRead(ProxMat);

  while(buttonState == HIGH) {

    //Bluetooth check start

  while(BlueT.available() > 0)  // Send data only when you receive data:
{
```

```
 data = BlueT.read();     //Read the incoming data and store it into variable data
 BlueT.print(data);       //Print Value inside data in Serial monitor
 BlueT.print("\n");  //New line

while(data == '2') {   //Checks whether value of data is equal to 2

  //Ultrasonic sensor check start

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance= duration*0.034/2;

Serial.print("Distance: ");
Serial.println(distance);


while (distance < MinDistance) {

digitalWrite(buzzer,HIGH);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
```

```
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);

}

//Ultrasonic sensor check end

  digitalWrite(buzzer,LOW);

    for(angle = 0; angle < 40; angle += 1)
  {
   servo.write(angle);
   delay(MotSpeed3);
  }

   //Ultrasonic sensor check start

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);
```

```
while (distance < MinDistance) {

  digitalWrite(buzzer,HIGH);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);

}

//Ultrasonic sensor check end

  digitalWrite(buzzer,LOW);

  for(angle = 40; angle >= 0; angle -= 1)
  {
    servo.write(angle);
    delay(MotSpeed3);
  }

  delay(500);
```

```
  data = data;

  }

}

    //Bluetoohh check end

    //Ultrasonic sensor check start

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);


while (distance < MinDistance) {

  digitalWrite(buzzer,HIGH);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);

}

//Ultrasonic sensor check end

  digitalWrite(buzzer,LOW);

    for(angle = 0; angle < 40; angle += 1)
  {
   servo.write(angle);
   delay(MotSpeed2);
  }

  delay(500);

  //Ultrasonic sensor check start

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);


while (distance < MinDistance) {

  digitalWrite(buzzer,HIGH);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);

}

//Ultrasonic sensor check end

  digitalWrite(buzzer,LOW);

  for(angle = 40; angle >= 0; angle -= 1)
  {
```

```
  servo.write(angle);
  delay(MotSpeed2);
}

delay(500);

 buttonState = digitalRead(ProxMat);

}

for(angle = 0; angle < 40; angle += 1)
{
 servo.write(angle);
 delay(MotSpeed1);
}

delay(500);

//L3 check end

//L3 check start

buttonState = digitalRead(ProxMat);

while(buttonState == HIGH) {

  //Ultrasonic sensor check start

digitalWrite(trigPin, LOW);
```

```
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);


while (distance < MinDistance) {

  digitalWrite(buzzer,HIGH);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);

}

//Ultrasonic sensor check end
```

```
  digitalWrite(buzzer,LOW);

   for(angle = 0; angle < 40; angle += 1)
  {
   servo.write(angle);
   delay(MotSpeed2);
  }

  delay(500);

  //Ultrasonic sensor check start

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);


while (distance < MinDistance) {

  digitalWrite(buzzer,HIGH);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;

  Serial.print("Distance: ");
  Serial.println(distance);

}

//Ultrasonic sensor check end

  digitalWrite(buzzer,LOW);

  for(angle = 40; angle >= 0; angle -= 1)
  {
    servo.write(angle);
    delay(MotSpeed2);
  }

  delay(500);



   buttonState = digitalRead(ProxMat);



  }
```

```
  //L3 check end



  for(angle = 40; angle >= 0; angle -= 1)
  {
    servo.write(angle);
    delay(MotSpeed1);
  }

  delay(500);



}
```