Department of Electronic and Computer Engineering

Electronic Design Project 3B (EDPB301)

Assessment Number 5A

Final Report

Project Title: Radar Security System

By

Group: E Project Number: 49		
Surname	**Initials**	**Student Number**
Thakurdin	**T. Y**	**22139454**

Plagiarism Declaration

1. I know and understand that plagiarism is using another person's work and pretending it is one's own, which is wrong.

2. This report is my own work.

3. I have appropriately referenced the work of other people I have used.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his/her own work.

Thakurdin T Y	22139454	
Surname and Initials	Student Number	Signature

19 November 2023

Date

Abstract

The purpose of this project, the Radar Security System, is to improve security in either residential, facility or industrial environments. Using an Arduino microcontroller with a Python GUI for ease of operation, the system integrates the RCWL-0516 radar module as the primary sensor and a secondary PIR sensor for validation checks.

The primary focus of the project is to detect unauthorized movements, the system then alerts personnel to possible breaches by blinking LEDs, sounding an alarm, and displaying breached areas on an LCD. Furthermore, the doppler-radar operation and obstruction-penetration capabilities of the RCWL-0516 make it well-suited for outdoor security, while PIR sensors validate the readings, which lowers the number of false alarms.

The results obtained from the testing procedures proves that the project is functional and effective in detecting motion in the designated areas. Additionally, the project takes a multidisciplinary approach, as it incorporates knowledge from the BEngTech program at DUT from modules such as engineering physics, digital systems, and electronic circuit design.

This Radar Security System is a cost-effective option for a variety of industries, as it has a heavy focus on practicality and technical feasibility. The original target market for this project was low-income or startup businesses, as the project is purely motion detection based and has a low power consumption which underscores the affordability and dependability of the security system.

Table of Contents

List of Figures

List of Tables

Abbreviations

AC	Alternating Current
BEngTech	Bachelor of Engineering Technology
CSIR	Council for Scientific and Industrial Research
DC	Direct Current
DUT	Durban University of Technology
GUI	Graphic User Interface
I2C LCD	Inter-Integrated Liquid Crystal Display
I/O	Input/Output
LDR	Light Dependant Resistor
LED	Light Emitting Diode
P(IR)	Passive (Infrared)
RADAR	Radio Detection and Ranging
SMD	Surface Mount Device
USB	Universal Serial Bus

Equations

Chapter 1 - Introduction

1.1 Introduction and Project Overview

Security is a fundamental concern for both households and larger facilities, and technological advancements have paved the way for smarter and more effective security solutions. The aim of this project was to develop a Radar Security System that combines multisensory technologies together to function as inputs to an Arduino microcontroller which would activate a buzzer alarm system and notify the user of an intrusion. This security system is controlled via a Python GUI for intuitive arming and disarming. This system was designed to detect unauthorized movements, alert occupants to potential security breaches in specific areas, and draw attention to the breached areas.

This Radar Security System is a multi-disciplinary project that required elements of electronic engineering, computer programming, and engineering physics. The Radar Security System was designed to detect and respond to any unauthorized movement within its designated areas. The system's primary sensor is the RCWL-0516 radar module, which accurately senses motion using Doppler-Radar technology, with a secondary PIR sensor for validation checks to reduce false alarms. Once armed and upon detecting motion, the system triggers the various alarm components until the system is disarmed.

1.2 Background and Case Studies

1.2.1 InnoSenT's Radar Technology

The company InnoSenT states that RADAR technology uses electromagnetic waves to detect, locate, and track objects [1]. Radars transmit electromagnetic waves that bounce off objects, enabling analysis of the reflected signals to determine an object's presence, distance, speed, and direction [1].

In security systems, radar is used for surveillance and intrusion detection in large areas, such as borders or critical infrastructure, to identify unauthorized intrusions. Additionally, radar provides perimeter security with virtual radar fences, triggering alerts upon detecting movements within the radar's range [1]. Radar also enhances maritime security by monitoring maritime traffic, tracking ships, and identifying suspicious activities [1].

Modern security systems often combine radar with cameras, infrared sensors, and data analytics to improve situational awareness [1]. Despite some limitations, radar remains essential in security applications due to its versatility and reliability.

Radar sensors offer contactless and anonymous detection, providing useful data about objects, vehicles, animals, or persons in their vicinity [1]. These sensors do not require direct contact with the detected material and can reliably measure even at long distances. Furthermore, the electromagnetic waves of the radar sensors can penetrate nonconductive materials without severely hindering the sensor's functionality [1].

1.2.2 Dahua Technology's Radar Perimeter Protection

Dahua Technology outlines the main challenges faced by security systems that do not make use of radar technologies such as a high false rate encountered by infrared sensors and conventional visual detection systems [2]. These systems often encounter a high false alarm rate triggered by factors such as small animals, leaves, and even flags, resulting in a considerable number of missed alarms.

Secondly Dahua Technology's state that traditional perimeter protection and monitoring systems are susceptible to weather and environmental conditions, impacting their detection accuracy [2]. Furthermore, stating that adverse weather conditions, such as rain, snow, and fog, as well as extremely dark environments, can hinder the system's effectiveness [2].

Lastly Dahua technologies outlined another limitation of traditional perimeter protection and monitoring systems being their use of a single-point monitoring approach, which fails to provide comprehensive tracking of a target within the monitored area [2]. This lack of global information hinders the system's ability to offer a complete and holistic view of intrusion events.

1.2.3 Magos Systems' Perimeter Intrusion Detection System

Magos Systems discusses the aspects of their perimeter intrusion detection system which contribute to its versatility and reliability in various applications.

The system has wide area coverage, achieved through its 30-degree elevation angle which eliminates dead zones, ensuring extensive coverage extending up to the installation pole [3]. Moreover, Magos' solution requires minimal infrastructure due to its low power consumption, low bandwidth communication, and long-range coverage capabilities of up to 1 Km, leading to reduced maintenance requirements [3].

Lastly, through the use of microwave radar technology operating in a specific frequency band, the impact of weather conditions such as rain, snow, and fog are greatly minimized [3]. Furthermore, since microwaves can penetrate nonconductive materials, a sturdy housing can enclose the radar without greatly degrading its performance.

1.2.4 DroneBot Workshop's Experiments with the RCWL-0516 Doppler Radar Motion Sensor

The RCWL-0516 radar module is a cost-effective and versatile Doppler radar sensor which is widely used for motion detection and proximity sensing. Its operation is based on the principles of the Doppler effect, where emitted microwaves are reflected off objects which the module uses to detect frequency shifts caused by moving objects within its range [4].

The module has lower power consumption and can operate at voltages between 4V and 24V at an operating current of 2.8A [4]. Furthermore, the module can operate effectively without a microcontroller. However, the addition of the microcontroller allows for better integration with other components as well as to create a latching circuit for an alarm system such that when the radar module detects movement, the alarm sounds and remains active until deactivated by the user [4].

Lastly, the radar module cannot penetrate conductive materials like metals, thus conductive materials can be used to limit the range and direction of the module's sensing which can be used to mitigate interference from other nearby RCWL-0516 modules [4].

1.2.5 The Meerkat Wide Area Surveillance System

The Meerkat Wide Area Surveillance System was developed by the CSIR to combat rhino poaching. The system leveraged state-of-the-art technologies to operate in the challenging environment of the Kruger National Park and detect and classify individuals and wildlife across expansive terrains which significantly mitigated rhino poaching incidents [5].

The system integrated the Reutech Radar System's ground surveillance radar sensor, which was used for target detection and localization, along with a CSIR-engineered electro-optic day and night sensor, capable of accurate human-animal classification [5]. The acquired data was then displayed on a geo-referenced electronic map within the control room, which aided in swift and coordinated responses for effective counter-poaching operations [5]. Though the Meerkat system was successful at mitigating poaching incidents, it showed the importance of automating its functions to reduce operational costs and enhance the effectiveness of 24/7 wide area surveillance.

1.3 Objectives, Specifications and Limitations

1.3.1 Objectives

- Motion Detection: Implement a reliable motion detection mechanism using the RCWL-0516 radar module and PIR sensor to identify unauthorized movements within the designated range of the system.
- Alarm Activation: Develop a responsive alarm system that promptly triggers a buzzer alarm upon detecting any potential security breaches, alerting occupants, and nearby individuals.
- Breached Area Notification: Display the breached area on an LCD (for example, a message will be displayed stating: Breached area 1)
- Python GUI: Create a user-friendly graphical interface using Python to facilitate arming and disarming of the security system, providing users with intuitive control.
- Protective Enclosure: A nonconductive enclosure was built to protect the components from adverse environmental conditions without severely degrading the sensing functionality as well as mimic the appearance of a facility which the security system monitors.

1.3.2 Specifications

- RCWL-0516 Radar Module: The core sensor of the system, leveraging Doppler radar technology to detect motion accurately within its defined range.
- PIR Motion Sensor Module: The PIR motion sensors are secondary validation sensors which aids in mitigating false alarms that may occur from the RCWL-0516 Radar Module.
- Arduino Microcontroller: Serving as the control centre, the Arduino Mega 2560 processes the inputs from the Python GUI as well as the signals from the radar module to trigger the alarm system accordingly.

- Buzzer Alarm: A piezo buzzer alarm provides an audible alert in response to detected motion, ensuring immediate attention to potential security breaches.
- Python Software: By employing the tkinter library in Python, the system features a graphical user interface with soft keys for convenient control of arming and disarming functions.
- Housing/Enclosure: The housing for the project is made up of a model that mimics a facility that required monitoring, additionally it protects the aforementioned components.

1.3.3 Limitations

- Range Constraints: The effective range of the RCWL-0516 radar module is finite, and additional radar modules would be required for coverage of multiple areas.
- Purely Motion Detection: The system focuses solely on motion detection and does not differentiate between authorized and unauthorized movements, relying on users to interpret the context of detections. Thus, the system will be deployed solely for motion detection in restricted access areas. Furthermore, both the PIR and RCWL-0516 modules lack the ability to measure distance, shape, or speed thus the system is only capable of motion detection.
- Independent System: The current proposal does not integrate with external security networks or communication protocols as it functions as a standalone solution.
- Power Constraints: This security system was designed to have a low power consumption and thus the selected components reflect this decision.

1.4 Significance and Benefits

Security systems are highly significant as they offer protection, peace of mind, convenience, and quick responses to possible dangers. This increases the safety of people and property, and acts as a deterrent to potential intruders, lowering risks and elevating overall security.

Additionally, the combination of radar technology and PIR sensors is useful for large scale facilities being able to detect distant movements with radar technology and close-range confirmation of living beings with PIR sensors.

1.4.1 Theft Deterrence

Simply knowing that a facility is protected by a security system is usually enough to deter low level threat intrusions, combining this security system with a small group of security personnel can greatly increase the theft deterrence capabilities.

1.4.2 Property Protection

Similar to theft deterrence, the security system can aid in protecting the property since not every unauthorized entry is a thief, there are also vandals and arsonists who seek to damage or deface property rather than steal.

1.4.3 Enhancing Response Efficiency

An LCD display that shows the relevant security information provides security personnel real-time feedback. Based on the information displayed and the alarm sound and light, they can very easily and strategically determine their best course of actions. Furthermore, the displayed information also aids in informing authorized personnel on the status of the system.

1.4.4 Lower Insurance Costs

Businesses as well as homeowners, may experience lower insurance costs after installing a security system since security systems decrease the chance of loss, insurance companies frequently give policyholders who install them discounts [6]. This decrease in costs can be very beneficial to the target market for this security system, namely low income, or start-up businesses.

1.5 Safety Measures

In South Africa, the Occupational Health and Safety Act 85 of 1993 was created to protect the health and safety of both people at work and people in connection with the use of or activities related to plant and machinery [7].

The following safety measures were taken to follow the Occupational Health and Safety Act 85 of 1993.

- The project innately adds to the protection of the health and safety of personnel in the facility.
- The RCWL-0516 module was selected because they use non-ionizing radiation, which is safe for human exposure as it doesn't have enough energy to ionize atoms or molecules since it operates at 3.2GHz using microwave radar technology.
- Purely motion detection: This ensures compliance with privacy laws as the system can only determine whether a living being has entered a protected area or not.
- Small delay after arming system: Allows for fewer false alarms. This improves the reliability of the system thus providing greater protection to the health and safety personnel in the facility.
- Password protected: Password protection reduces the risk of harm to employees and the facility by protecting sensitive data and preventing unauthorized access.
- Emergency contact information: By providing emergency contact information, the system aids in improving the reaction times of emergency units.
- Quit function: Personnel can end the serial connection to the Arduino and close the application by clicking the "Quit" button. It's critical to properly close the connection to avoid any problems or data loss.
- Validation sensors: The PIR sensors aid in minimizing false alarms from the radar module making the system more reliable and thus aiding to the protection of personnel in the facility. Furthermore, PIR sensors pose no risk to the health and safety of people due to their low power consumption and passive nature since there is no radiation emission.

Chapter 2 – Design

2.1 System Block Diagram:

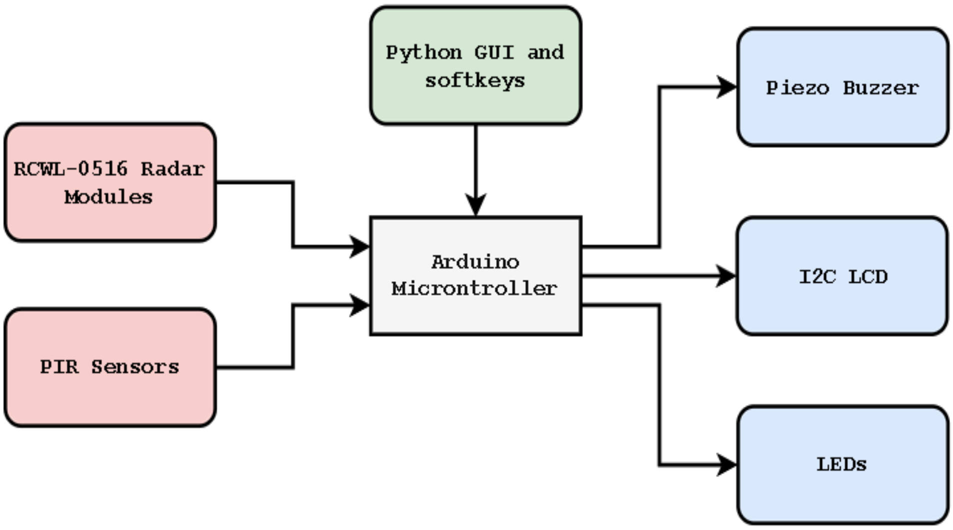


Figure 2. 1 System Block Diagram

The project is a single and independent system controlled by an Arduino Mega 2560 microcontroller, powered by USB. The microcontroller utilizes the RCWL-0516 radar modules as its primary input devices since this is a radar security system. Following this, the PIR motion sensor modules are used for validation to detect living organisms, this choice was made to mitigate false alarms and improve the overall functionality of the system.

If both the radar module and its corresponding PIR module return a high value, while the area is armed via the GUI, indicating a double positive detection, the alarm sounds and is generated by a piezo buzzer at a tone alternating between 0.8Khz and 1KHz which was selected through an empirical testing method. Furthermore, a red, 5mm LED lights up in the breached area to draw further attention and an I2C LCD displays a warning message stating which area or areas have been breached to alert occupants that may not be able to hear the alarm or see the LEDs.

Lastly, the arming and disarming process is controlled via a Python GUI with softkeys. Both arming and disarming require a four-digit passcode to prevent the risk of accidental or malicious use of the system. Lastly, the user interface has simple and intuitive design however to further assist personnel during a crisis, a help softkey was implemented that displayed relevant contact information.

2.2 Circuit Diagram, Schematic and Operation

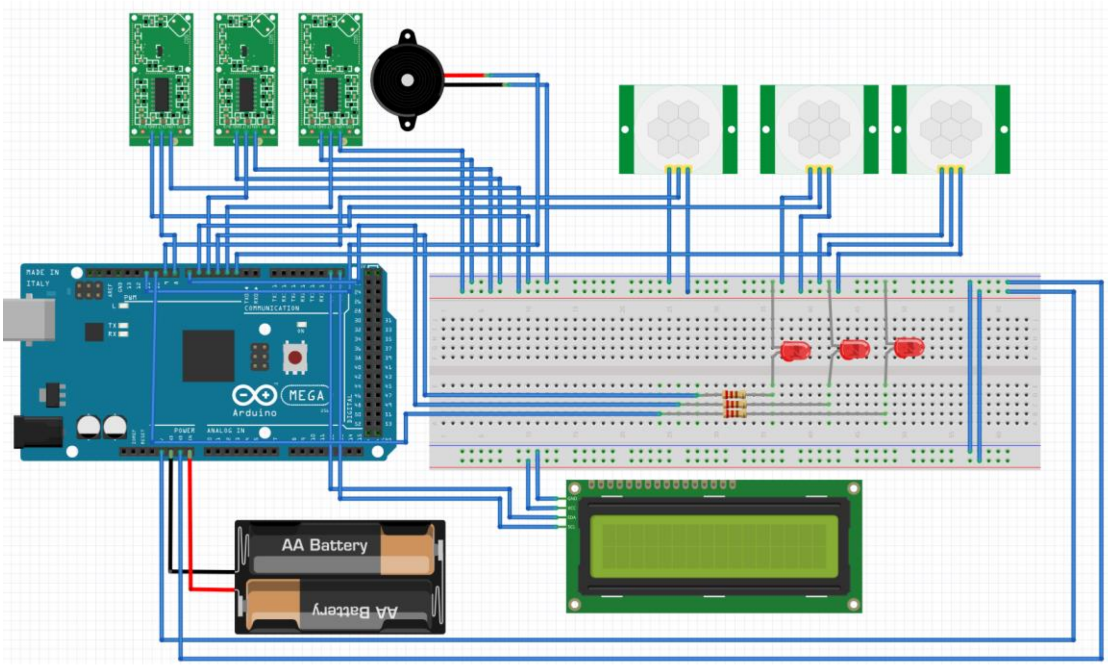


Figure 2. 2 System Circuit Diagram

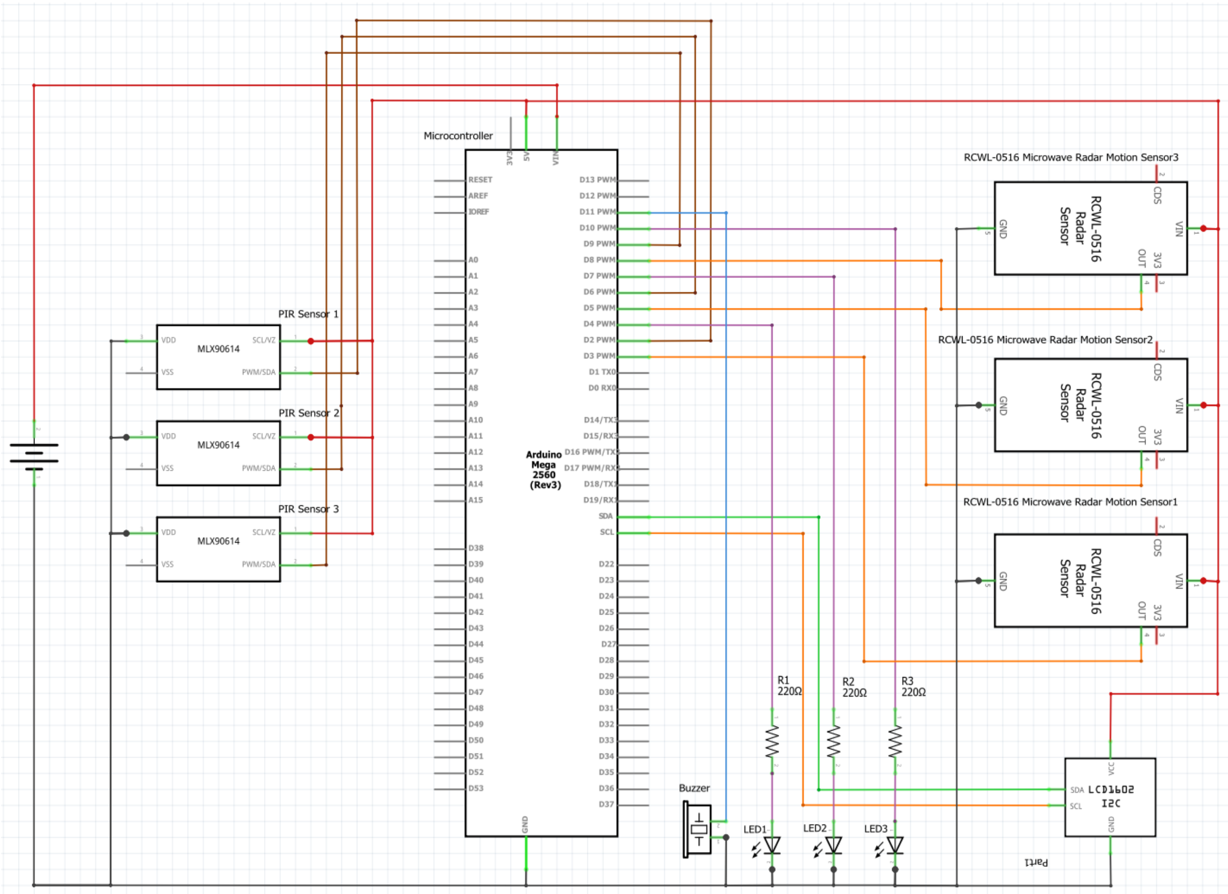


Figure 2. 3 System Circuit Schematic

The block diagram in **Figure 2.1** was used as the starting point to design and develop the circuit diagram in **Figure 2.2** using the Fritizing software. Additionally, **Figure 2.3**, the system schematic was also designed in Fritizing and referred back to as it provides a neater and simpler visual understanding of the connections within the project. Though **Figure 2.2** and **2.3** depict DC batteries, this was purely for simulation purposes as the Arduino is powered by a USB cable; reason being, the Arduino needs to communicate with GUI via serial communication across a USB, thus the system is more concise and compact without an external battery source. Furthermore, constructing a back-up power supply capable of powering the computer or laptop with the GUI is beyond the scope of this project.

2.2.1 RCWL-0516 Module:

The RCWL-0516 module, as shown in **Figure 2.4**, is a doppler-radar-based motion sensor that operates at microwave frequencies. It emits continuous microwave signals and measures the frequency shift caused by any moving object within its detection range.

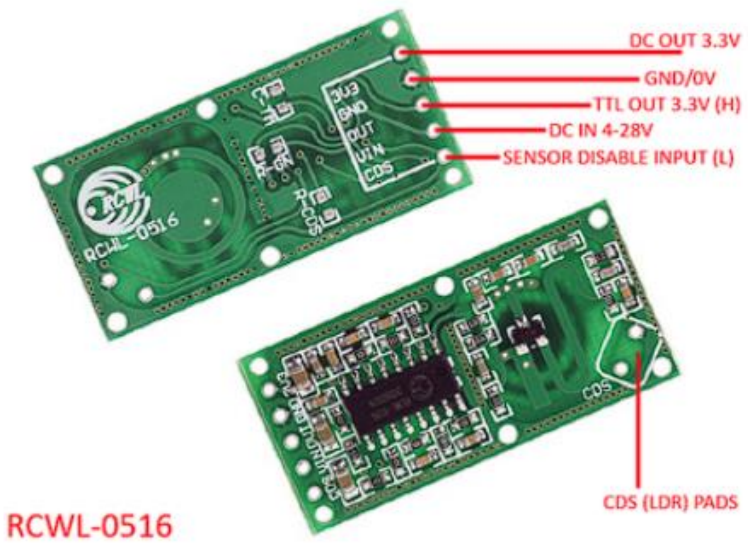


Figure 2. 4 RCWL-0516 radar module [8]

This module has a detection range, up to 7m [8], and adjustable time delay settings to customize the system behaviour based on the specific application requirements. Furthermore, an LDR can be soldered to the module to allow the system to only operate in low light conditions if necessary [8].

However, this module faces challenges with false alarms as it only detects movement, ergo they cannot distinguish between an intruder and a moving, inanimate object, thus PIR validation sensors were implemented.

Three RCWL-0516 modules are used in this project and their detection range is a 120° cone which reaches up to 7m [8]. Furthermore, these modules are low power devices making it ideal for use with an Arduino microcontroller. Additional specifications for this module are tabulated in **Table 2.1**.

Ultrasonic sensors were one of the alternatives taken into consideration for this part. However, since the main objective of the project was to develop a radar security system, a radar module was more appropriate to implement, however radar technology is extremely expensive thus the RCWL-0516 was chosen due to its doppler-radar motion detection as well as its cost-effective price.

Table 2. 1 RCWL-0516 pin specifications

Pin	Function
3V3	3.3V regulated output
GND	Ground
OUT	Trigger (high = 3.3V, normally low = 0V)
VIN	4 – 28V supply voltage
CDS	Light sensor relator

2.2.2 HC-SR501 PIR Motion Sensor Module:

To reduce false alarms and enhance the reliability of the system, PIR sensors are used for validation. PIR sensors detect changes in the infrared radiation emitted by living beings and other heat-emitting objects. The Arduino Uno uses the output signal from the PIR sensors to validate the motion detected by the RCWL-0516 modules. When both the RCWL-0516 and PIR sensors detect motion together, it increases the certainty that the detected target is an intruder and not a false alarm caused by environmental factors.

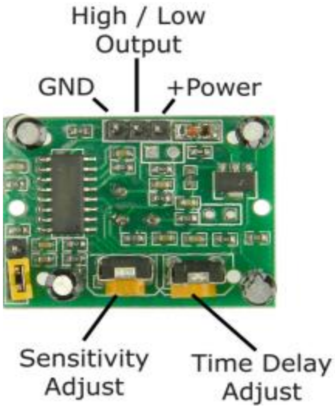


Figure 2. 5 PIR motion sensor module [9]

The PIR sensor modules used in this project and depicted in **Figure 2.5**, have a detection angle of 100° and are used for close range validation of a threat [9]. In a large-scale security system, the radar sensors could give early warning detection of movements at a long range while the PIR sensors could confirm said threat once it had moved closer to the facility, unfortunately for this small-scale project, it was not applicable to allow the radar and PIR modules to detect at their full range as it causes false alarms.

The IR sensor is housed in a metal casing that is hermetically sealed for better immunity against noise, temperature changes, and humidity, additionally, the IR sensing element is shielded by an IR-transmissive material for protection [9]. Lastly, a Fresnel lens is placed over the IR sensor to improve the ability of the sensor to identify changes in heat patterns by focusing and directing thermal radiation onto the sensor element [9]. Additional specifications for the module are tabulated in **Table 2.2**.

Alternatives considered for this part included video camera surveillance however, this would increase the price of this project substantially. Additionally, IR sensors were also considered which tend to be more accurate than PIR, however they draw more power and do not detect the presence of a living organism, rather they detect if the IR light they emitted has been reflected thus making it obsolete for the purpose of validation checks for the radar module.

Table 2. 2 HC-SR501 PIR module pin specifications

Pin	**Function**
VCC	4.5 – 12V supply voltage
GND	Ground
OUT	Trigger (High if motion detected)

2.2.3 Arduino Mega 2560 Control Centre:

The Arduino Mega 2560 serves as the control centre of the radar security system and is depicted in **Figure 2.6**. It receives signals from the RCWL-0516 modules , PIR sensors and GUI while also powering the entire system. The Arduino continuously monitors the input signals and processes the data using appropriate algorithms to determine if an intruder is present. The Arduino Mega was selected as opposed to the Arduino Uno due to its greater pin capacity. The Arduino Uno provided just enough pins for the proposed project but allowed no room for any further connections which would severely limit expansion which may be required at a later stage. The pin constraints of the Arduino Mega are far less hindering and allows for further development of the system. Lastly, the final version of the code uploaded to the Arduino is shown in **Annexure A.**

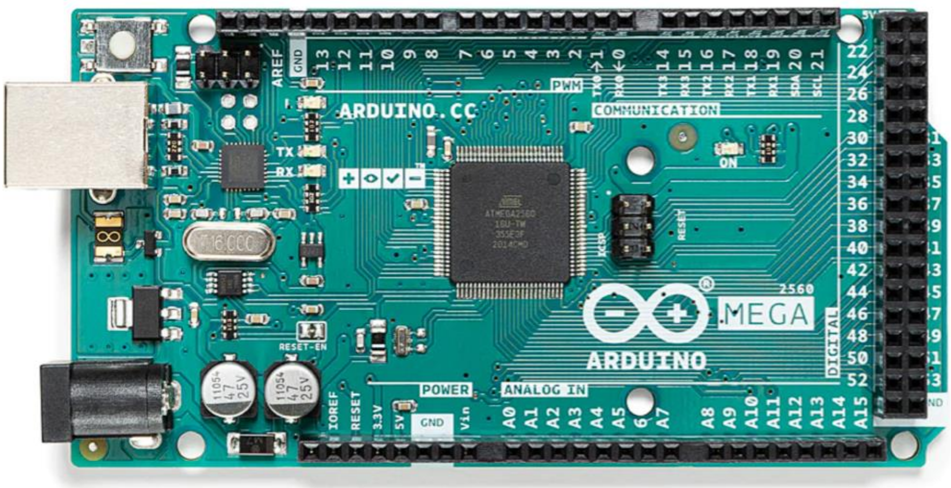


Figure 2. 6 Arduino Mega 2560 [10]

2.2.4 Buzzer alarm and LED Activation:

Upon detecting an intruder, the Arduino Uno triggers the piezo buzzer to sound an alarm, alerting nearby personnel of the security breach. Directly after this, the Arduino activates an LED located in the area where the intruder was detected. This LED serves as a visual indicator for security personnel to locate the intruder quickly and take appropriate action as well as for signalling unarmed personnel to avoid or evacuate this area.

A Piezo buzzer is an active electronic component that generates sound through the piezoelectric effect. When an electric signal is applied to the buzzer, it vibrates and produces sound waves, creating the alarm sound. The piezo buzzer was chosen as it is an active component and does not require additional components to adjust its tone.

The three red LEDs used in this project are passive electronic components that emit light when current flows through them in the forward direction. They require a current limiting resistor to protect them from an overload.

$$R = \frac{v}{I} = \frac{Supply\ Voltage - Voltage\ Drop}{Supply\ Current} = \frac{5V - 1.9V}{30mA} = 103.33\ \Omega \qquad [2\text{- }1]$$

Equation 2-1 was used to determine a suitable value for a current limiting resistor to protect the LEDs, the calculated value was then rounded up to a more common resistor value of 220**Ω**.

2.2.5 I2C LCD

Figure 2.7 shows the 16×2 LCD with a serial I2C module. After being soldered together the I2C LCD was implemented to inform occupants of the security breach. This feature may be helpful to occupants with a hearing disability. Since the warning light activates in the breached area, occupants on the other side of the facility may not be able to see it due to walls and other obstructions.

The I2C serial communications interface allows the LCD to interface with the Arduino with fewer wires which aids in cable management and construction efficiency [11].

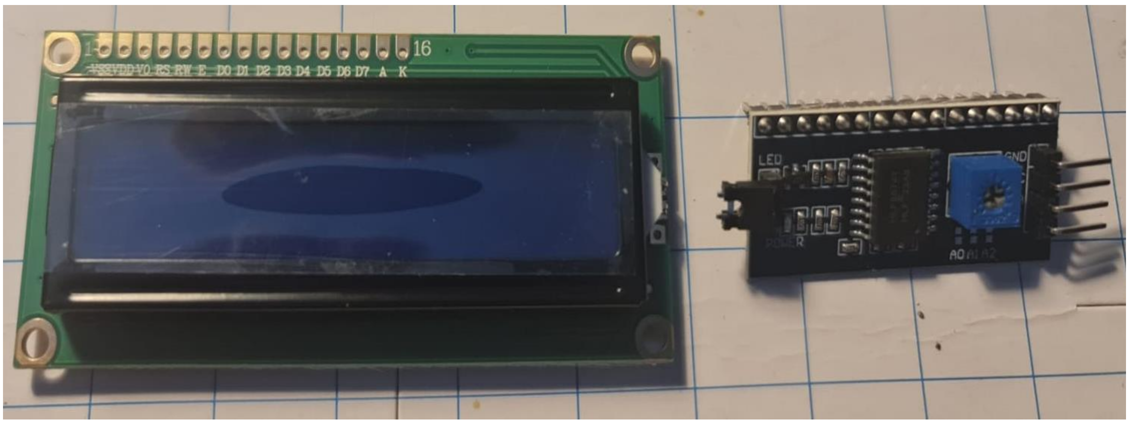


Figure 2. 7 I2C LCD

2.2.6 Python GUI and Soft Keys

The Python programming language was used to create a GUI for this project, depicted in **Figure 2.8**, to enable intuitive user control by authorized personnel. Python was chosen as it has a large database of libraries and an active and informative community of experienced programmers which aids in troubleshooting coding related errors. Two libraries played a crucial role in the development of this GUI, PySerial and TKinter. TKinter was used to develop the aesthetic and functions of the GUI while PySerial created a serial connection between the Python GUI and the Arduino microcontroller to allow bi-directional serial communication via a USB cable. Users interact with the GUI to send the arm and disarm instructions to the Arduino, which then uses process the data and controls the alarm system accordingly.

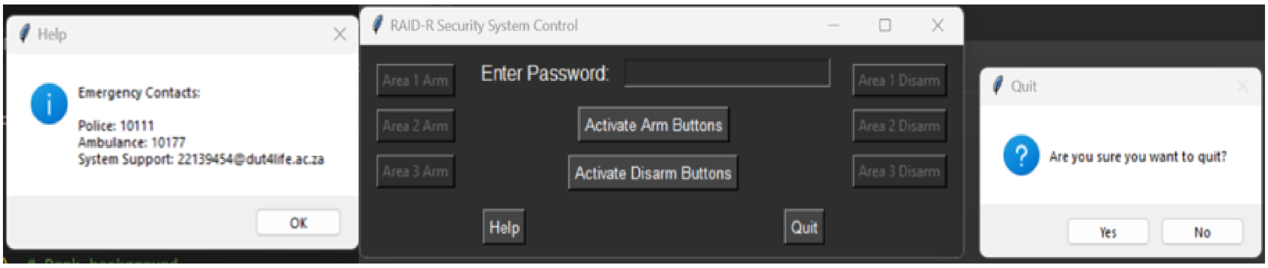


Figure 2. 8 System GUI

The user must enter the correct password (1234) to be able to arm the system and a separate password (5678) is required to disarm the system. Additionally, there a quit function was implemented which ends the serial communication between the GUI and Arduino thus leaving the Arduino in whatever state it was previously in prior to quitting. Additionally, a help function was implemented that displays relevant emergency contact information such as the South African Police Services (10111), National Ambulance Service emergency number (10177) and the system support email (22139454@dut4life.ac.za). 8Lastly the code to run the GUI is shown in **Annexure B**.

Chapter 3 – Construction & Testing

3.1 Model

The finalized design of the housing facility is displayed in **Figure 3.1**, the model depicts a scale of 1:100 with a perimeter of 33 cm by 45 cm. Lastly the LCD is located on the facility premises to emulate a notifications screen for the facility.

Figure 3. 1 Project Facility Model

$$A = L \times B = (0.33 \times 100) \times (0.45 \times 100) = \ 1485\, m^2 \qquad [3\text{-}1]$$

Equation [3-1] depicts the square meterage of the facility that this model was scaled to, which equates to 0.1485 m^2 for the model. Where A represents area in squared meters, L represents the length of the perimeter and B refers to the breadth of the perimeter.

3.2 Software

- Arduino IDE: Used to develop and upload Arduino code. The IDE simplifies and facilitates Arduino code development, testing, and uploading.
- Fritzing: Used to design circuit diagram and schematic. Fritzing has a large community which aids in acquiring and implementing nonstandard electronic components for simulation purposes.
- Microsoft Excel: Used to construct tables and was used due to its popularity and familiarity which made managing project-related data is significantly easier.
- GanttPRO: Used to construct the project Gantt chart. GanttPRO was selected for its ease of use and cost effectiveness.
- Draw.io: Used to create block diagram in **Figure 2.1** and flowchart in **Annexure C**; it was selected for its intuitive interface and diverse library of shapes and effects.
- Python: Used to develop the GUI with the aid of the tkinter and PySerial libraries. Python was selected due to its simplicity and versatility.

3.3 Tools and Materials

The construction and testing process of this project involved first inspecting and testing each component, such as the resistor and LED tests depicted in **Figure 3.2**. Following this, subsystems were built and tested separately on a breadboard and then interfaced to create the breadboard prototype of the final project. The radar modules and PIR sensors were held to face in the same direction with crocodile clips from the compact workstation tool and the microcontroller was connected to the motion detection sensors, alarm components and GUI. The breadboard prototype was then reconstructed on a veroboard and placed in the constructed housing.

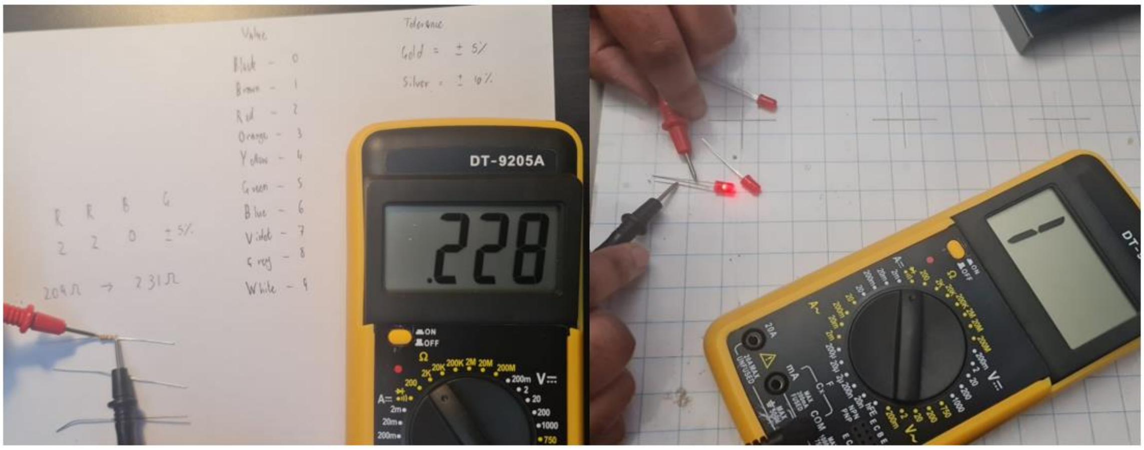


Figure 3. 2 LED and Resistor Testing

3.3.1 Materials and Components:

- Arduino Mega 2560
- RCWL-0516 Radar Module
- PIR Motion Sensors
- Jumper wires
- Breadboard
- LED lights
- Piezo Buzzer
- I2C serial interface module & LCD

3.3.2 Tools:

- Soldering iron and soler wire
- Screw drivers
- Wire cutters
- Needle nose plier
- Multimeter
- Compact Workstation
- Scissors
- Stanley Knife

3.4 Soldering

Soldering included attaching header pins to components as well as interfacing the LCD with the I2C module, shown in **Figures 3.3** and **3.4.** Additionally, the final prototype was soldered onto a veroboard based on the veroboard schematic depicted in **Annexure D.**

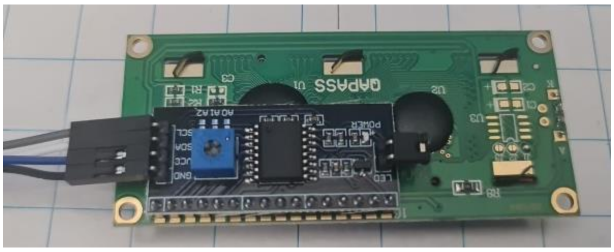


Figure 3. 3 Soldered I2C LCD

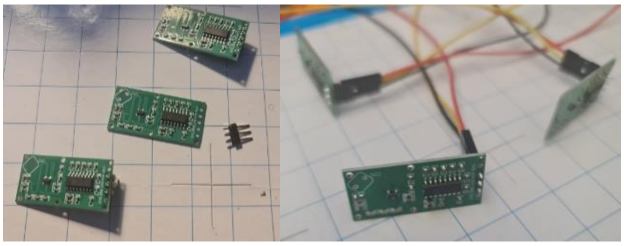


Figure 3. 4 Soldered RCWL-0516 Modules

3.5 Breadboard Prototype

Figure 3.5 shows the breadboard prototype of the 3-area-monitoring prototype which was constructed according to **Figures 2.2** and **2.3** and the wiring guide from **Table 3.1**. The final breadboard prototype was an expansion upon the 1-area-monitoring subsystem. The three areas operate separately but share control over the buzzer alarm.

Creating a breadboard prototype is essential for effective and adaptable circuit testing. It reaffirms component compatibility, helps identify and resolve problems, and allows for multiple project iterations without making a permanent commitment. Additionally, validating the project's viability and functionality at this stage was crucial to reducing errors before moving on to the final implementation.

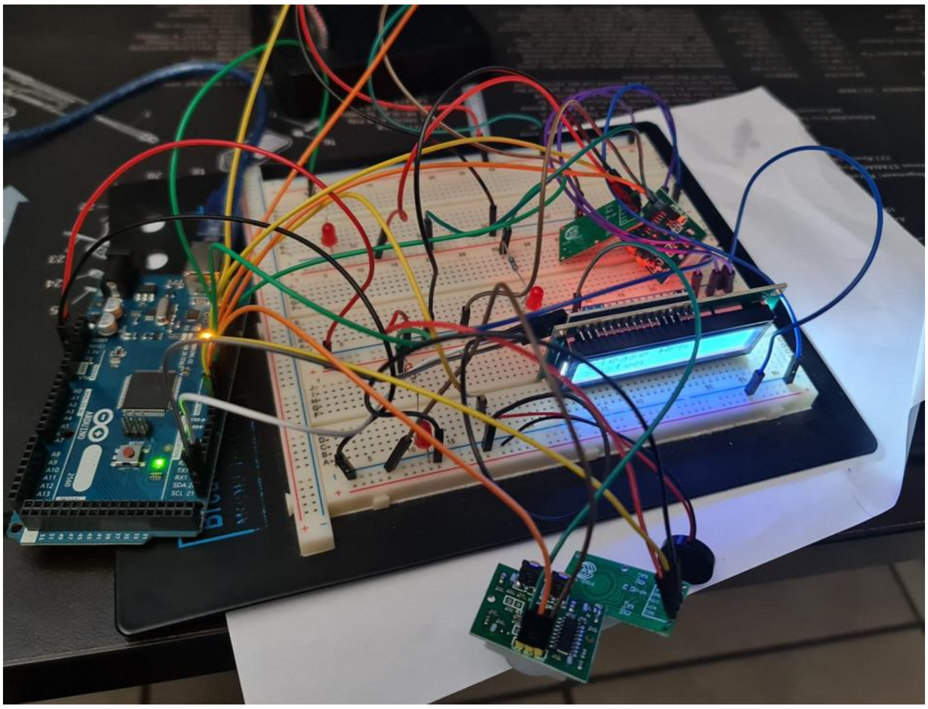


Figure 3. 5 Final Breadboard Prototype

Table 3. 1 System Wiring Guide

Component Pin	Arduino Pin
RCWL OUT (1)	2
RCWL OUT (2)	5
RCWL OUT (3)	8
PIR OUT (1)	3
PIR OUT (2)	6
PIR OUT (3)	9
LED ANODE (1) + R1	4
LED ANODE (2) + R2	7
LED ANODE (3) + R3	10
Buzzer Anode	11
LCD SDA	SDA
LCD SCL	SCL
RCWL Vin (1)	5V
RCWL Vin (2)	5V
RCWL Vin (3)	5V
PIR Vin (1)	5V
PIR Vin (2)	5V
PIR Vin (3)	5V
LCD Vcc	5V
RCWL GND (1)	GND
RCWL GND (2)	GND
RCWL GND (3)	GND
PIR GND (1)	GND
PIR GND (2)	GND
PIR GND (3)	GND
LED Cathode (1)	GND
LED Cathode (2)	GND
LED Cathode (3)	GND
Buzzer Cathode	GND
LCD GND	GND

3.6 Procedure

3.6.1 Component Integration

- Inspected and interfaced project components to the Arduino Mega 2560, by following the wiring guide in **Table 3.1** and using jumper wires, furthermore, inspected each connection to ensure that each component was properly powered and grounded.
- The RCWL-0516 radar module and PIR motions sensor were connected to the respective digital I/O pins.
- Interfaced the I2C LCD to the relevant pins on the Arduino.
- The 5mm red LED and a 220**Ω** current limiting resistor were connected across another digital I/O pin and ground.
- Lastly the piezo buzzer was connected across a digital I/O pin and ground.

3.6.2 Arduino Programming

- Arduino code was written to test each component individually and then in combination with one another.
- The code involved reading input data from the RCWL-0516 module and PIR motion detector to detect intrusions.
- The I2C LCD was programmed to display relevant system status messages.
- The LED and piezo buzzer would activate to indicate alarms once both motion detectors returned a high value.
- Implemented serial communication between GUI and Arduino.

3.6.3 Python GUI

- Developed a Python based GUI using Tkinter.
- The GUI takes a user input for a password to arm the system and a separate password to disarm the system as well as emergency contact information and a quit function which would end serial communication.
- Established serial communication between the GUI and the Arduino. This communication is essential for arming and disarming the system.

3.6.4 Testing Procedures

3.6.4.1 Hardware Testing

- Verified that each component was fully functional, the components were then properly connected and powered.
- Tested the RCWL-0516 module through a practical simulation by observing if it triggers when motion was detected.
- Similarly, tested the PIR motion detector by moving in front of it and verifying if it correctly detects motion.
- Repeated the sensor tests with the combination of the two.
- Checked if the LED and piezo buzzer respond to Arduino commands.

- Verified threat detection by displaying notifications on the I2C LCD for a 1-area-monitoring subsystem, code for the Arduino and GUI of this subsystem can be found in **Annexures E** and **F**.
- Subsequently, verified threat detection by displaying notifications on the I2C LCD for 3-area-monitoring breadboard prototype.
- After soldering the relevant components and connections onto the veroboard, a continuity test was performed, using the multimeter, on each rail to ensure no short circuits were present.

Figure 3.6 depicts the testing of the RCWL-0516, the primary sensor of the system. The red signal denotes the analogue signal while reading movement whereas the green signal depicts a movement large enough to force the RCWL-0516 output to go high which is the digital input for the Arduino. Similar results were observed from the PIR sensor.

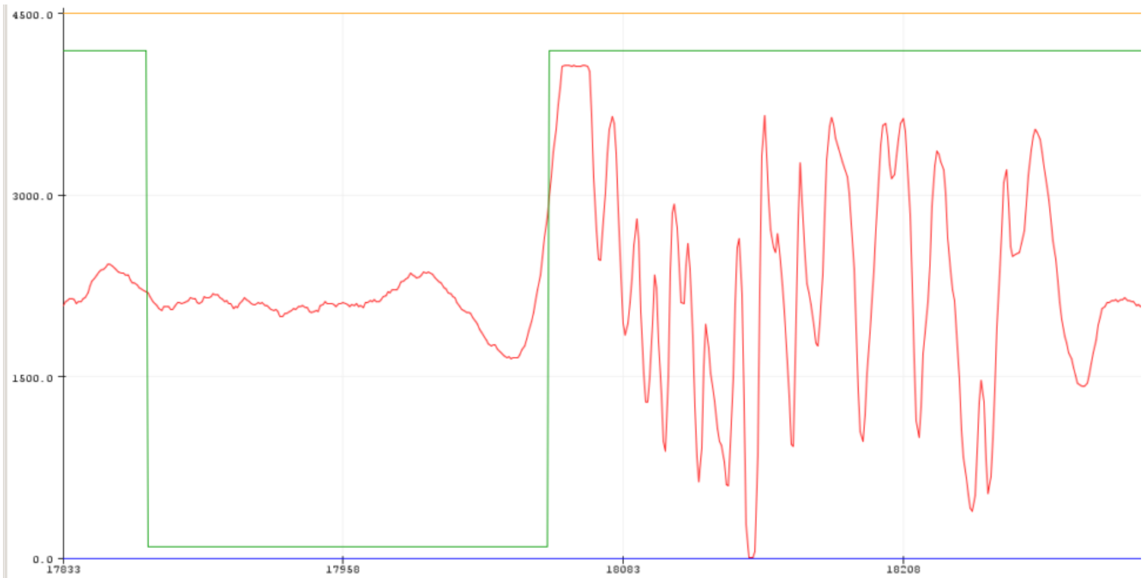


Figure 3. 6 RCWL-0516 Detection Analogue (R) & Digital (G) RCWL-0516 Signals

3.6.4.2 Functional Testing

- Tested the Python GUI to ensure that it could successfully send the appropriate arming and disarming commands to the Arduino to activate and deactivate the specific subsystems monitoring area 1, 2 and 3.
- Verified that the GUI password protection system works as intended and clears the password after each entry.

- Tested the functionality of the system by triggering both sensors in each area followed by removing one of the sensors output pins to ensure that the system was not relying on either sensor but rather the combination of both, as shown in **Annexure G**.
- Verified that the LED and piezo buzzer responded to the motion detection confirmation from the sensors as expected. However, unstoppable looping occurred due to being unable to disarm the system, however by resetting the microcontroller and through an iterative coding method, the code was adjusted to fix this issue, forcing the alarm to deactivate once disarmed.

3.6.4.3 Communication Testing

- Checked the serial communication between the Python GUI and Arduino. PySerial allows for bidirectional serial communication which aided in displaying system status messages on both the LCD and GUI.
- Implemented a try and except function to the Arduino during serial communication to notify the user when the Arduino is not connected.

3.6.4.4 Error Handling & Fool proofing

- Tested various error scenarios, such as entering incorrect passwords or pressing the same button twice in an attempt to change the arm and disarm state, to ensure that the system handles these cases appropriately.
- Since the system was intended for use by others, the GUI was designed to be ultra simple and intuitive making it easier to operate and maintain. Additionally, emergency contact details are available to call or email for assistance from the relevant authorities.

3.6.5 Documentation

- All the steps taken during the development and testing process were documented and will be used as a reference for future expansions or troubleshooting.

Chapter 4 - Results

4.1 Experimental Design and Setup

The following experiments were designed to evaluate the performance of the Radar Security System:

- **Sensitivity Analysis:** The sensitivity of the PIR sensors was systematically adjusted with the objective of verifying the upper and lower detection ranges to determine the limitations. This calibration process involved fine-tuning the sensor to detect changes in infrared radiation within its field of view. The sensor worked as expected at all ranges up to 7m and within its 110° cone of vision. Secondly, The RCWL-0516 sensitivity was not adjusted as this adjustment is meant to be permanent by soldering an SMD resistor across the relevant pads on the module as well as to emphasize that radar technology is generally used for long range detection.
- **Intrusion Detection Scenarios:** Intrusion detection scenarios encompassed slow, deliberate movements, rapid actions, and sequences of multiple consecutive motions. The aim was to evaluate the response of the system to various types of motion and assess its ability to cause the alarm to sound and ensure the monitoring, alarm and notification systems remained active until disarmed. Through this empirical method, the responses of the system were noted and used to develop a more dependable security system.
- **User Interaction Evaluation:** Usability tests were run on multiple people. They were allowed to interact with the Python GUI, alarm and disarm the system as well as the trigger the alarm. This was done to observe how users, regardless of their technical backgrounds, interacted with the Python GUI.

4.2 Data Collection and Analysis

- **Motion Detection Accuracy:** The PIR sensor demonstrated an accuracy rate of 80%, 8/10 times. The PIR sensor excelled at detecting rapid, close-range motion however it struggled slightly with distant and very sluggish movements.
- **Intrusion Detection and Alarm Activation:** The alarm sounds and the LCD displays in which area the intrusion has occurred while a threat was detected. Since the Arduino doe does not have multithreading capabilities, it cannot perform tasks simultaneously. Thus, latching the alarm would prevent the system from monitoring the separate areas. However, by implementing the millis() function, the system mimics simultaneous commands which allows the alarm to sound while

motion was continuously detected which allows for an accurate assessment of where the intrusion was taking place while still being able to monitor the other areas.

- **User Interaction and Control:** Usability tests confirmed the intuitive nature of the simplified Python GUI. Participants were pleased with the simple to grasp interface as well the updated dark theme aesthetic.

4.3 Results Conclusions

- The PIR sensor is best suited for close range motion detection, as it was intended to be used, and its high accuracy readings as a validation sensor increase the reliability of the system, making it a compelling choice for security solutions.
- The ability to detect various types of motion, including slow and rapid movements, positions the system as a promising solution for security applications. The system was reworked to activate the alarm immediately once both sensors in an area had detected motion and remain active while motion was detected which, in theory, would allow security personnel to better identify the breached area as well as possibly cause the intruders to panic.
- The user-centric design of the Python GUI was meant to keep it super simple in order to allow less technically skilled, but still authorized, individuals to interact with and control the security system. This aids in the adaptability and reliability of the security system as it requires neither extensive nor expensive training to use. Furthermore, by updating the aesthetic of the GUI to a dark theme, users felt it was more unique and conveyed a greater sense of confidentiality and security.

Thus, the performance of the security system reaffirms its capabilities and potential for deployment as a security solution for low-income and startup businesses. Future maintenance and expansions will likely involve further refinements and aesthetic changes to enhance its capabilities and visual appeal.

Figure 4. 1 Working Project

Chapter 5 – Project Management

5.1 Work Breakdown Structure

Table 4. 1 Work Breakdown Structure

	Task name	Start date	End date	Predecessor
		2023-07-19	**2023-11-21**	
1	**Prerequisutes and Proposals**	**2023-07-19**	**2023-08-21**	
1.1	Project Research	2023-07-19	Ongoing	
1.2	Circuit Design	2023-07-22	2023-07-26	1.1
1.3	Draft Proposal Writing	2023-07-26	2023-07-31	1.2
1.4	Further Research	2023-07-31	2023-08-11	1.3
1.5	Final Proposal Writing	2023-08-11	2023-08-21	1.4
2	Proposal Accepted	2023-08-25	2023-08-25	1
3	**Code Development**	**2023-08-26**	**2023-09-15**	**2**
3.1	Radar Module Coding	2023-08-26	2023-08-31	1.5
3.2	PIR Module Coding	2023-08-31	2023-09-05	3.1
3.3	Matlab Coding	2023-09-05	2023-09-12	3.2
3.4	Alarm Coding	2023-09-11	2023-09-15	3.3
4	**Progression**	**2023-09-16**	**2023-10-03**	**3**
4.1	Progress Presentation Preparation	2023-09-16	2023-10-02	3.4
4.2	Progress Presentation	2023-10-02	2023-10-03	4.1
5	Parts Received	2023-10-06	2023-10-06	4
6	**Prototype Phase**	**2023-10-07**	**2023-11-15**	**5**
6.1	Project Prototype	2023-10-07	2023-10-14	4.2
6.2	Testing and Results Analysis	2023-10-14	2023-10-17	6.1
6.3	Troubleshooting and Problem Solving	2023-10-17	2023-10-21	6.2
6.4	Working Project	2023-10-21	2023-10-21	6.3
6.5	Draft Report Writing	2023-10-21	2023-10-31	6.4
6.6	Final Report Writing	2023-10-31	2023-11-15	6.5
7	**Presentation and Demonstration**	**2023-11-16**	**2023-11-21**	**6**
7.1	Presenatation Preparation	2023-11-16	2023-11-20	
7.2	Project Presentation	2023-11-20	2023-11-21	7.1
7.3	Project Demonstration	2023-11-20	2023-11-21	7.1

Table 5. 2 Project Tasks and Duration

Task Name	Start (Date)	End (Date)	Duration (Days)	Duration (Hours)
Project research	2023-07-19	2023-07-22	3	7
Circuit Design	2023-07-22	2023-07-26	4	5
Report writing - draft proposal	2023-07-26	2023-07-31	5	12
Further research	2023-07-31	2023-08-11	11	7
Report writing - final proposal	2023-08-10	2023-08-20	10	9
Code development and error checking	2023-08-26	2023-09-15	20	20
Progress report preparation	2023-09-16	2023-10-02	16	10
Project progress report	2023-10-02	2023-10-03	1	2
Project prototype	2023-10-07	2023-10-15	8	8
Testing and results analysis	2023-10-14	2023-10-17	3	4
Troubleshooting and problem solving	2023-10-17	2023-10-21	4	5
Report writing - draft report	2023-10-21	2023-10-31	10	12
Report writing - final project report	2023-10-31	2023-11-19	19	9
Project presentation preparation	2023-11-19	2023-11-23	4	7
Project presentation	2023-11-23	2023-11-23	0	0.5
Project demonstration	2023-11-23	2023-11-23	0	0.25

5.2 Project Gantt Chart

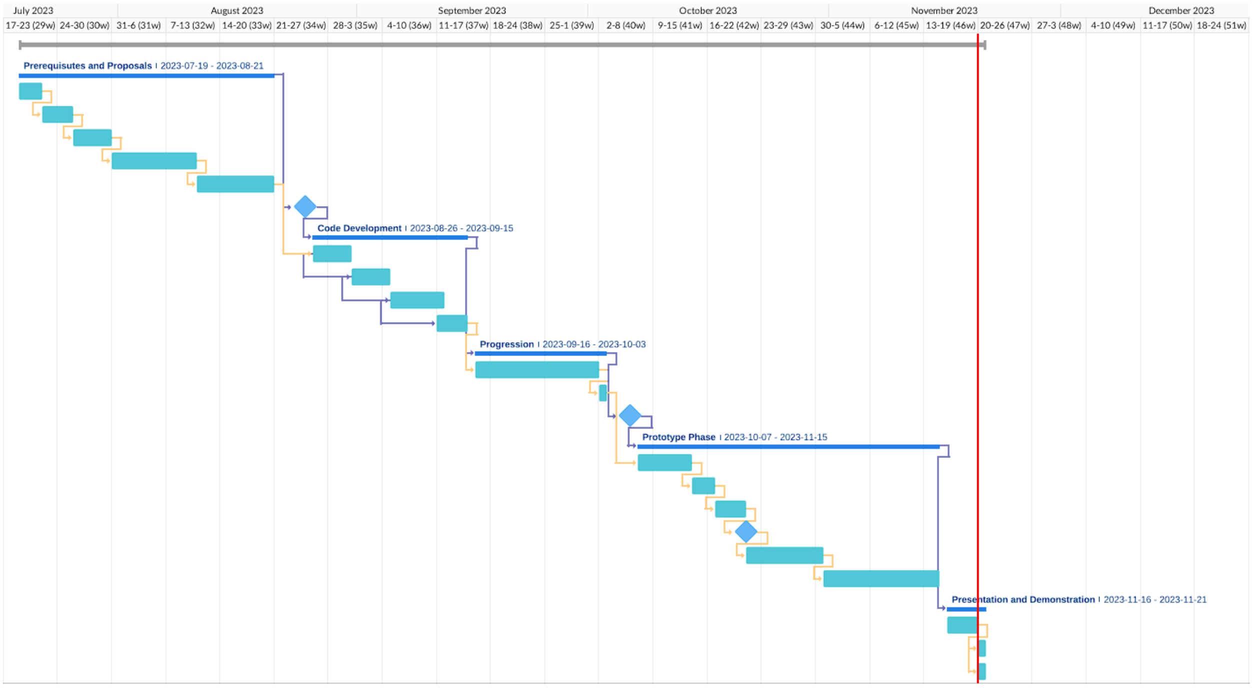


Figure 5. 1 Gantt Chart

5.3 Cost of Materials

Table 5. 3 Cost of Materials

Part	Cost Per Unit	Quantity	Total	Supplier
Arduino Mega 2560	R 670,77	1	R 670,77	Mantech
RCWL-0516 Radar Module	R 23,00	3	R 69,00	Mantech
HC-SR501 PIR Motion Sensor Module	R 92,95	3	R 278,85	DIY Electronics
I2C LCD	R 143,87	1	R 143,87	Mantech
PT1704 Piezo Buzzer	R 12,77	1	R 12,77	Mantech
Red LED	R 1,86	3	R 5,58	Mantech
Total Proposed Project Costs			R 1 180,84	

5.4 Project Management Conclusion

The project has followed the schedule quite closely, there was a delay in the construction and prototyping of the project due to a lack of components, namely the PIR sensors, however the final project was completed on time and results were documented into this report.

When comparing the proposed cost of materials depicted in **Annexure H**, with the current cost of materials, a minor price increase occurred due to ordering PIR sensors from a different supplier and shipping costs. Apart from this minor setback, the only significant changes to the project were positive progressions towards completing it.

Chapter 6 – BEngTech Level Knowledge

6.1 Technical Literacy

This module demonstrated the use of technology tools for accessing, managing, creating, and communicating technical information. Topics included experimental methods, technical report writing, referencing standards, spreadsheet usage, and adherence to relevant standards.

6.2 Engineering Physics 1A & 1B

This module provided knowledge on applying physics principles to engineering analysis. Such as electromagnetic waves and radiation as well as the doppler effect and electrical properties of components and circuits. The Radar and PIR modules used in this project make use of these physics principles to detect movement using the doppler effect and infrared radiation respectively.

6.3 Analogue Electronics 1A & 1B

These modules provided an understanding of the fundamentals of analogue electronics and basic circuit design. They explained the construction and analysis of electronic circuits and demonstrated how to use electronic circuit simulation software.

6.4 Digital Electronics 1A & 1B

These modules provided an understanding of the construction and operation of digital components and technologies such as displays and microprocessors as well as the construction and analysis of digital electronic circuits.

6.5 Computer Programming 2A

This module taught the fundamental concepts of programming using a high-level language. It included topics like top-down design methods, integrated development environments (IDEs), GUI design, control structures, program timers, and file access.

6.6 Electronic Circuit Design 2A & 2B

These modules built-on the previously mentioned fundamental electronics knowledge to study advanced circuit design, construction, and simulation using through hole technology and surface mount devices.

6.7 Fundamentals of Microcontrollers 2A and Embedded Systems 2B

These modules introduced working with microcontroller systems as well as how to design flowcharts and develop code to control microcontroller processes.

6.8 Electronic Design Project 3A

This module had an emphasis on project development, design analysis, effective communication, and time management skills. Furthermore, it supplied a project which required knowledge and skills acquired from multiple BEngTech modules.

6.9 BEngTech and this Project

This Radar Security System requires the culmination of the theories, principles and skills that were taught in the aforementioned modules. The knowledge and skills acquired from these modules aided in the conceptualization, design, and documentation of this project.

Furthermore, these acquired skills and knowledge were used during the construction phase and will be used during the demonstration and presentation phases to ensure the project meets the required levels of a professional engineering technologist.

Chapter 7 – Project Originality

7.1 Low Power Consumption

The low power consumption feature of the system was designed to add to its efficiency and sustainability. By optimizing energy usage through the implementation of low power modules like the RCWL-0516 radar module, the system can operate without imposing excessive strain on power resources. This feature allows for extended surveillance periods thus reducing maintenance requirements and costs in the long term. In South Africa specifically, this low power consumption feature would not put excessive strain on the power grid and potentially save small business owners money instead of purchasing and maintaining more power intensive security systems.

7.2 Specific Area Monitoring

The system's focus on specific area monitoring increases its effectiveness by directing attention to areas with restricted access. This capability minimizes false alarms caused by irrelevant movements thus reducing the strain on security personnel and system resources. By concentrating its detection on designated zones, the system optimizes response times, enhancing the overall security of the facility. This feature allows for greater resource management, lower false alarm rates and expansion possibilities.

7.3 Intuitive User Interface

The Python GUI of this project prioritizes an intuitive and aesthetic user interface which aimed to streamline its operation and management. This approach minimizes the learning curve for security personnel which leads to faster more efficient responses and overall greater security. This feature reduces the complexity when using the system without harming the performance or efficiency of the system. Furthermore, relevant emergency contact details are easily accessible allowing for faster responses to intrusions or system errors.

7.4 Multi-Sensor System

The combination of radar detection and PIR motion detection decreases the likelihood of false alarms. This synergy between the different sensor modules optimizes accuracy and minimizes the systems blind spots since radar can be used to monitor large areas to detect movements, while PIR modules excel at close-range heat detection. Thus, by combining their strengths, this system provides accurate intruder detection. However, on this small-scale project the long-range detection capabilities cannot be used to their full potential.

7.5 Overall Project Originality

The originality of the project was derived from the implementation of all the previously listed features and capabilities into a single security solution to create an easy to use, effective, efficient, and dependable radar security system that can operate with low power consumption and a low level of false alarms.

Chapter 8 – Problems & Solutions

8.1 General Problems and Solutions

Table 8. 1 General Problems and Solutions

Problems	**Solutions**
Arduino Uno had insufficient pin capacity.	Substitute Arduino Mega 2560 microcontroller instead to facilitate additional components.
Advanced radar technology can be expensive, and the project's budget is a constraint in acquiring the most sophisticated sensors with extended capabilities.	Downsize the project’s capabilities to the necessities and make use of sufficient alternatives. Such as using the RCWL -0516 Doppler radar module.
Limited detection range.	Implement designated area monitoring to prioritize motion detection in restricted areas.
False alarms.	Implement secondary PIR sensors to validate radar detection and mitigate false alarms.
User interface may be too confusing, especially to individuals under duress during a security breach.	Made the user interface intuitive and simple and included a help function which provides emergency contact information.
Loose wiring connections.	Soldered correctly and inspected each connection
Defective Components.	Inspected and assessed all components to ensure proper working before project construction.
Lack of necessary knowledge to complete project.	Performed continuous research on aspects related to the project to gain a greater understanding as well as asked for assistance from qualified individuals such as lecturers, tutors, and lab technicians.

8.2 Complex Problems

8.2.1 Interference

The radar modules were tested in close proximity to each other, 5cm apart, and worked independently with no interference between each other however, when an attempt was made to integrate wireless communications for notifications to project via connecting an ESP 8266 Wi-Fi module, the Wi-Fi transmitter was too close to the radar modules causing severe false positives during detection phase due to interference with the radar module. This issue was easily resolved by moving the transmitter 2m away from the sensor, which is a viable solution for a large-scale facility, however, for this small-scale project, 2m away at this scale was inefficient and unsuitable. Thus, the implementation of the wireless communications was cancelled to minimize the risk of false alarms, decrease the cost of production, and ensure that this project could safely and accurately notify the user of an intruder.

8.2.2 Lack of Components

PIR sensors were ordered in the part list however they were not supplied. Multiple attempts to purchase PIR sensors were made from companies such as MANTECH who were extremely low on stock, with only 1 PIR sensor on hand, A1 Radio who stated the PIR sensors on their website were discontinued and DIY electronics. An order for 3 more PIR sensors was placed with DIY electronics and arrived during the week of November the 7th.

Thus, to overcome this issue, the single PIR sensor ,purchased from MANTECH, was used to develop the subsystem of a single area which was then expanded upon by integrating the remaining PIR sensors from DIY Electronics to allow the project to monitor 3 separate areas as it was designed to.

8.2.3 Improper Project Operation

Initial versions of the project did not function as desired with issues such as unintentional looping of the code and incorrect responses to the sensor readings. However, through an iterative method, the code was refined to function as expected of a security system and has expected and repeatable outcomes.

Chapter 9 - Conclusion

The final prototype has been built and based on the results from the testing procedures, the project was fully functional and showed potential as a reliable and efficient motion detection security system. However, before it can be implemented into facilities, it must first be vetted and cleared by the South African Security Association as the system would be evaluated in relation to the Private Security Industry Regulatory Authority Act of 2001 and the Security Industry Regulatory Authority Regulations [12].

This radar security system aims to provide an accurate and practical solution for detecting intruders that have breached restricted areas and alerting all occupants of the facility to the intrusion. The RCWL-0516 module was selected as opposed to components such as ultrasonic sensors due to the project requiring the system to be radar based. Furthermore, the selection of the RCWL-0516 radar modules was based on its doppler-radar operation and obstruction-penetration capabilities, making it suitable for outdoor security applications. The cost-effective price-to-performance ratio and compact design of the radar modules enhances both its own and the overall viability of the system.

The PIR sensors were selected for this project as they are secondary detection devices used to validate the readings from the radar modules. Additionally, since the radar modules only detect movement, the PIR sensors heat detection capabilities aid in limiting the likelihood of false alarms. Thus, the combination of radar technology and PIR sensors ensures accurate and reliable intruder detection while the implementation of a Python GUI with softkeys and password protection, provides intuitive and safe ease of use.

Furthermore, this project combines various skills and knowledge, such as circuit design and analysis as well as python and Arduino programming, which were acquired while studying for the BEngTech degree in Electronic and Computer Engineering at DUT, to design an effective radar-based security system with potential security applications in various industries.

In conclusion, this radar security system has a strong emphasis on setting realistic goals and ensuring the technical feasibility of the integrated technologies, with the aim of creating a reliable and affordable security system, for low income or startup business, that would alert occupants of a facility to an intrusion to allow for fast responses from security personnel.

References

[1] "Basics of radar technology - what is radar and how does it work?," *InnoSenT* [Online]. Available: https://www.innosent.de/en/radar/ [Accessed 27 July 2023].

[2] "Radar Perimeter Protection," *Dahua Technology* [Online]. Available: https://www.dahuasecurity.com/sa/solutions/solutionsByApplication/797 [Accessed 27 July 2023].

[3] "Perimeter Security - Intrusion Detection System," *Magos Systems* [Online]. https://magossystems.com/ [Accessed 27 July 2023].

[4] "Experiments with the RCWL-0516," *DroneBot Workshop*, Jan. 12, 2019 [Online]. Available: https://dronebotworkshop.com/rcwl-0516-experiments/ [Accessed 27 July 2023].

[5] "Meerkat Wide-Area Surveillance Systems" CSIR [Online]. Available: https://www.csir.co.za/surveillance-systems [Accessed 27 July 2023].

[6] "Advantages of an Alarm System on a Business Insurance Policy," *Tivly* [Online]. Available: https://tivly.com/alarm-system-insuranceadvantages#:~:text=Most%20insurance%20companies%20will%20offer [Accessed 27 October 2023].

[7] "Occupational Health and Safety Act," *South African Government* [Online]. Available: https://www.gov.za/documents/occupational-health-and-safety-act [Accessed 17 November 2023].

[8] "Microwave Radar Sensor RCWL 0516," *Raspberry Valley* [Online]. Available: https://raspberry-valley.azurewebsites.net/RCWL-0516-Microwave-Radar-Sensor/ [Accessed 29 July 2023].

[9] "HC-SR501 PIR Motion Detector," *MPJA* [Online]. Available: https://www.mpja.com/download/31227sc.pdf [Accessed 27 October 2023].

[10] "Arduino Mega 2560 Rev3," *Arduino* [Online]. Available: https://store.arduino.cc/products/arduino-mega-2560-rev3 [Accessed 11 August 2023].

[11] "MD0069 - LCD DISPLAY MODULE 1602 WITH I2C INTERFACE," *Mantech* [Online]. Available: https://www.mantech.co.za/ProductInfo.aspx?Item=15M8246 [Accessed 29 July 2023].

[12] "Governing Legislation," *Security Association of South Africa* [Online]. Available: https://www.sasecurity.co.za/governing-legislation/ [Accessed 17 November 2023].

Annexure A: Final Project Arduino Code

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

// Define pins for the first security system
#define RCWLPin1 2
#define PIRPin1 3
#define LEDPin1 4

// Define pins for the second security system
#define RCWLPin2 5
#define PIRPin2 6
#define LEDPin2 7

// Define pins for the third security system
#define RCWLPin3 8
#define PIRPin3 9
#define LEDPin3 10

// Define the pin for the alarm buzzer
#define BuzzerPin 11

// Flags to track system and threat states
bool alarmActive = false;
bool systemArmed1 = false;
bool systemArmed2 = false;
bool systemArmed3 = false;
bool threatDetected1 = false;
bool threatDetected2 = false;
bool threatDetected3 = false;

// Record the start time of the alarm
unsigned long alarmStartTime = 0;

// Variable to store the current message displayed on the LCD
String currentDisplay = "";

void setup() {
  // Set input/output modes for pins of the first security system
  pinMode(RCWLPin1, INPUT);
  pinMode(PIRPin1, INPUT);
  pinMode(LEDPin1, OUTPUT);

  // Set input/output modes for pins of the second security system
  pinMode(RCWLPin2, INPUT);
  pinMode(PIRPin2, INPUT);
```

```
  pinMode(LEDPin2, OUTPUT);

  // Set input/output modes for pins of the third security system
  pinMode(RCWLPin3, INPUT);
  pinMode(PIRPin3, INPUT);
  pinMode(LEDPin3, OUTPUT);

  // Set pin mode for the alarm buzzer
  pinMode(BuzzerPin, OUTPUT);

  // Initialize and configure the LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Please Arm");
  lcd.setCursor(0, 1);
  lcd.print("System");

  // Start serial communication
  Serial.begin(9600);
}

// Function to produce an alternating sound for the alarm
void alarmSound() {
  static bool isHighTone = false;
  static unsigned long lastChangeMillis = 0;

  if (millis() - lastChangeMillis >= 500) {
    int frequency = 1000;  //frequencies for a lower pitch

    if (isHighTone) {
      frequency = 800;
      delay(50);  // delay after high tone
    }

    tone(BuzzerPin, frequency, 500);  // alarm duration

    isHighTone = !isHighTone;
    lastChangeMillis = millis();
  }
}

// Function to stop the alarm sound
void stopAlarmSound() {
  noTone(BuzzerPin);
}
```

```
void loop() {
  // Check for incoming commands through serial communication
  if (Serial.available() > 0) {
    char command = Serial.read();

    // Process the received command and update system states accordingly
    if (command == '1' && !systemArmed1) {
      systemArmed1 = true;
      threatDetected1 = false;  // Clear threat detection when arming
      updateDisplay("System 1 Armed");
    } else if (command == '2' && !systemArmed2) {
      systemArmed2 = true;
      threatDetected2 = false;  // Clear threat detection when arming
      updateDisplay("System 2 Armed");
    } else if (command == '3' && !systemArmed3) {
      systemArmed3 = true;
      threatDetected3 = false;  // Clear threat detection when arming
      updateDisplay("System 3 Armed");
    } else if (command == '4' && systemArmed1) {
      systemArmed1 = false;
      threatDetected1 = false;  // Clear threat detection when disarming
      digitalWrite(LEDPin1, LOW);
      lcd.clear();
      lcd.print("System 1 Unarmed");
      stopAlarmSound();
    } else if (command == '5' && systemArmed2) {
      systemArmed2 = false;
      threatDetected2 = false;  // Clear threat detection when disarming
      digitalWrite(LEDPin2, LOW);
      lcd.clear();
      lcd.print("System 2 Unarmed");
      stopAlarmSound();
    } else if (command == '6' && systemArmed3) {
      systemArmed3 = false;
      threatDetected3 = false;  // Clear threat detection when disarming
      digitalWrite(LEDPin3, LOW);
      lcd.clear();
      lcd.print("System 3 Unarmed");
      stopAlarmSound();
    }
  }

  // Monitor each security system's area for potential threats
  monitorArea(RCWLPin1, PIRPin1, LEDPin1, systemArmed1, threatDetected1);
  monitorArea(RCWLPin2, PIRPin2, LEDPin2, systemArmed2, threatDetected2);
  monitorArea(RCWLPin3, PIRPin3, LEDPin3, systemArmed3, threatDetected3);
```

```
  // short delay to balance system speed and control
  delay(10);
}

// Function to update the LCD display with a new message
void updateDisplay(String message) {
  // Only update the display if the message has changed
  if (currentDisplay != message) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(message);
    currentDisplay = message;
  }
}

// Function to monitor the area for potential threats
void monitorArea(int rcwlPin, int pirPin, int ledPin, bool systemArmed, bool
&threatDetected) {
  if (systemArmed) {
    // Read the status of the RCWL and PIR sensors
    bool RCWL_detection = digitalRead(rcwlPin);
    bool PIR_detection = digitalRead(pirPin);

    // Trigger the alarm when both RCWL and PIR detect motion
    if (RCWL_detection == HIGH && PIR_detection == HIGH) {
      threatDetected = true;

      // Update the display with information about breached areas
      updateDisplay("Breached Area(s):");

      lcd.setCursor(0, 1);

      // Indicate which security systems detected a threat
      if (threatDetected1) {
        lcd.print(" 1 ");
      }

      if (threatDetected2) {
        lcd.print(" 2 ");
      }

      if (threatDetected3) {
        lcd.print(" 3 ");
      }

      // Activate the LED and sound the alarm
      digitalWrite(ledPin, HIGH);
      alarmSound();
```

```
    } else if (threatDetected) {
      // Reset the alarm state if no motion detected after an intruder was
detected
      threatDetected = false;
      digitalWrite(ledPin, LOW);
      stopAlarmSound();

      // Clear the display if no threat detected
      updateDisplay("");
    }
  }
}
```

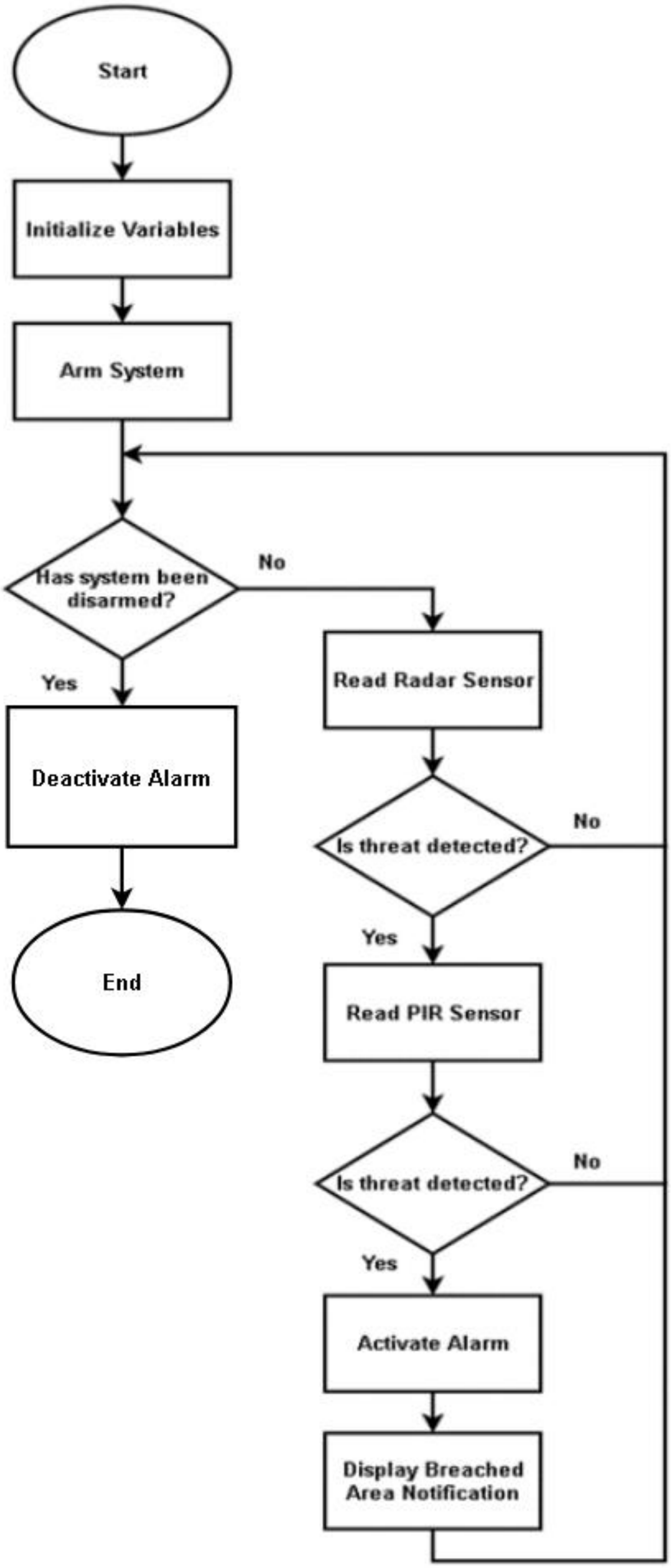
Annexure B: Final Project Python GUI code

```
import tkinter as tk
from tkinter import messagebox
import serial

# Establish a serial connection with an Arduino on 'COM3' port
ser = serial.Serial('COM3', 9600, timeout=1)

# Constants for arming and disarming passwords
PASSWORD_ARM = "1234"
PASSWORD_DISARM = "5678"

# Function to send a command to the Arduino via serial communication
def send_command(command):
    try:
        ser.write(command.encode())
        response = ser.readline().decode().strip()
        return response
    except serial.SerialException:
        return "Error: Arduino not connected."

# Function to create arm buttons dynamically
def create_arm_button(master, area_num):
    arm_command = str(area_num)
    btn = tk.Button(master, text=f"Area {area_num} Arm", command=lambda
a=arm_command: arm_disarm_area(a), state=tk.DISABLED, bg='#2E2E2E',
fg='#FFFFFF')  # Dark background, white text
    btn.pack(side=tk.TOP, fill=tk.X, padx=5, pady=5)
    return btn

# Function to create disarm buttons dynamically
def create_disarm_button(master, area_num):
    disarm_command = str(area_num + 3)
    btn = tk.Button(master, text=f"Area {area_num} Disarm", command=lambda
d=disarm_command: arm_disarm_area(d), state=tk.DISABLED, bg='#2E2E2E',
fg='#FFFFFF')
    btn.pack(side=tk.TOP, fill=tk.X, padx=5, pady=5)
    return btn

# Function to check entered password and enable corresponding buttons
def check_password(entry, password, buttons):
    entered_password = entry.get()
    entry.delete(0, tk.END)  # Clear the entry after checking

    if entered_password == password:
        for button in buttons:
            button['state'] = tk.NORMAL
```

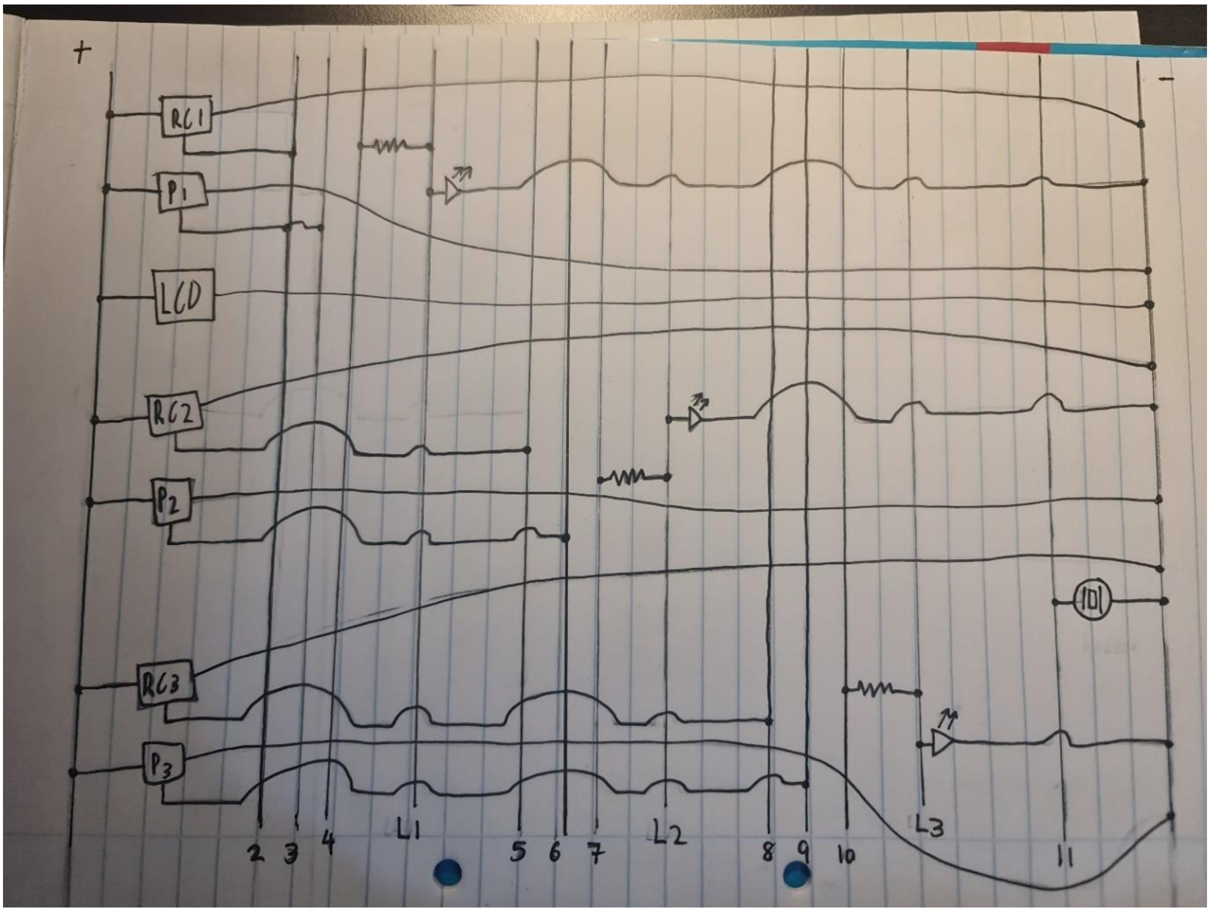
```
        notification_label.config(text="Armed buttons activated." if password
== PASSWORD_ARM else "Disarm buttons activated.")
    else:
        notification_label.config(text="Incorrect password. Please try
again.")

# Function to handle arming or disarming an area based on user input
def arm_disarm_area(area_command):
    try:
        response = send_command(area_command)
        area_num = int(area_command[-1])
        notification_label.config(text=f"Area {area_num} {'Armed.' if 'Armed'
in response else 'Disarmed.'}")
    except ValueError:
        notification_label.config(text="Error: Invalid area command.")

# Function to quit the application with a confirmation dialog
def quit_application():
    confirm_quit = messagebox.askyesno("Quit", "Are you sure you want to
quit?")
    if confirm_quit:
        root.destroy()

# Function to display emergency contacts and system support information
def show_help():
    help_message = "Emergency Contacts:\n\nPolice: 10111\nAmbulance:
10177\nSystem Support: 22139454@dut4life.ac.za"
    messagebox.showinfo("Help", help_message)

# Main application logic
if __name__ == "__main__":
    root = tk.Tk()
    root.title("Security System Control")
    root.configure(bg='#2E2E2E')  # Dark background

    # Frame for arm buttons
    arm_frame = tk.Frame(root, bg='#2E2E2E')
    arm_frame.pack(side=tk.LEFT, padx=10, pady=10, fill=tk.Y)

    # Frame for disarm buttons
    disarm_frame = tk.Frame(root, bg='#2E2E2E')
    disarm_frame.pack(side=tk.RIGHT, padx=10, pady=10, fill=tk.Y)

    arm_buttons = []
    disarm_buttons = []

    # Create arm and disarm buttons for three security areas
    for area_num in range(1, 4):
```

```
      arm_button = create_arm_button(arm_frame, area_num)
        disarm_button = create_disarm_button(disarm_frame, area_num)
        arm_buttons.append(arm_button)
        disarm_buttons.append(disarm_button)

    # Frame for password entry
    password_frame = tk.Frame(root, bg='#2E2E2E')
    password_frame.pack(pady=10)

    # Label for password entry
    password_label = tk.Label(password_frame, text="Enter Password:",
font=('Helvetica', 12), bg='#2E2E2E', fg='#FFFFFF')
    password_label.pack(side=tk.LEFT, padx=5)

    # Entry field for password input
      password_entry = tk.Entry(password_frame, show="*",
      font=('Helvetica',12), bg='#2E2E2E', fg='#FFFFFF')
    password_entry.pack(side=tk.LEFT, padx=5)

    # Button to activate arm buttons
    arm_password_button = tk.Button(root, text="Activate Arm Buttons",
command=lambda: check_password(password_entry, PASSWORD_ARM, arm_buttons),
font=('Helvetica', 10), bg='#444444', fg='#FFFFFF')
    arm_password_button.pack(pady=5)

    # Button to activate disarm buttons
    disarm_password_button = tk.Button(root, text="Activate Disarm Buttons",
command=lambda: check_password(password_entry, PASSWORD_DISARM,
disarm_buttons), font=('Helvetica', 10), bg='#444444', fg='#FFFFFF')
    disarm_password_button.pack(pady=5)

    # Button for help
    help_button = tk.Button(root, text="Help", command=show_help,
font=('Helvetica', 10), bg='#444444', fg='#FFFFFF')
    help_button.pack(side=tk.LEFT, anchor=tk.SW, padx=10, pady=10)

    # Button to quit the application
    quit_button = tk.Button(root, text="Quit", command=quit_application,
font=('Helvetica', 10), bg='#444444', fg='#FFFFFF')
    quit_button.pack(side=tk.RIGHT, anchor=tk.SE, padx=10, pady=10)

    # Label for notifications
    notification_label = tk.Label(root, text="", font=('Helvetica', 10),
bg='#2E2E2E', fg='#FFFFFF')
    notification_label.pack(side=tk.BOTTOM, anchor=tk.S, pady=10)

    # Start the Tkinter event loop
    root.mainloop()
```

Annexure C: Flowchart



Annexure D: Veroboard Schematic



Annexure E: Arduino Code 1 Area

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

#define RCWLPin 2
#define PIRPin 3
#define LEDPin 4
#define BuzzerPin 5

bool alarmActive = false;
bool systemArmed = false;

void setup()
{
  pinMode(RCWLPin, INPUT);
  pinMode(PIRPin, INPUT);
  pinMode(LEDPin, OUTPUT);
  pinMode(BuzzerPin, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Please Arm");
  lcd.setCursor(0, 1);
  lcd.print("System");

  Serial.begin(9600);
}

void alarmSound()
{
  tone(BuzzerPin, 2000);
  delay(500);
  noTone(BuzzerPin);
}

void loop()
{
  if (Serial.available() > 0)
  {
    char command = Serial.read();
    if (command == 'A')
    {
      systemArmed = true;
      delay(1000); // Small delay after arming
```

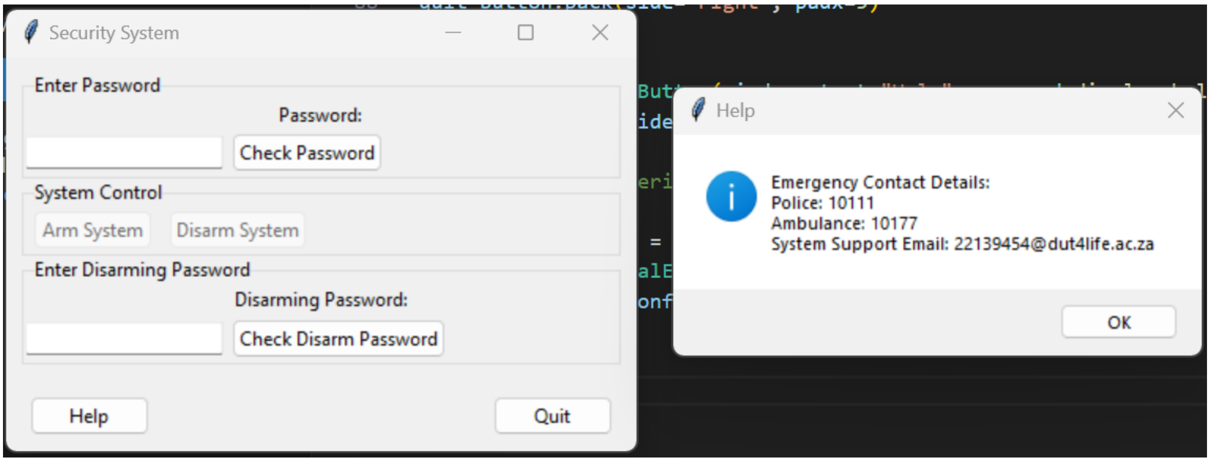
```
      lcd.clear();
      lcd.print("System Armed");
    }
    else if (command == 'D')
    {
      systemArmed = false;
      digitalWrite(LEDPin, LOW);
      noTone(BuzzerPin);
      // Clear or set the PIR and RCWL values to LOW
      digitalWrite(PIRPin, LOW);
      digitalWrite(RCWLPin, LOW);
      lcd.clear();
      lcd.print("System Unarmed");
    }
  }

  if (systemArmed)
  {
    bool RCWL_detection = digitalRead(RCWLPin);
    bool PIR_detection = digitalRead(PIRPin);
    if (RCWL_detection == HIGH && PIR_detection == HIGH)
    {
      alarmActive = true;
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Intruder In:");
      lcd.setCursor(0, 1);
      lcd.print("Area A");
      alarmSound();
      digitalWrite(LEDPin, HIGH);
    }
    else if (alarmActive)
    {
      alarmSound();
    }
  }
  else
  {
    digitalWrite(LEDPin, LOW);
    noTone(BuzzerPin);
    alarmActive = false;
  }
}
```

Annexure F: Initial 1 Area Python GUI

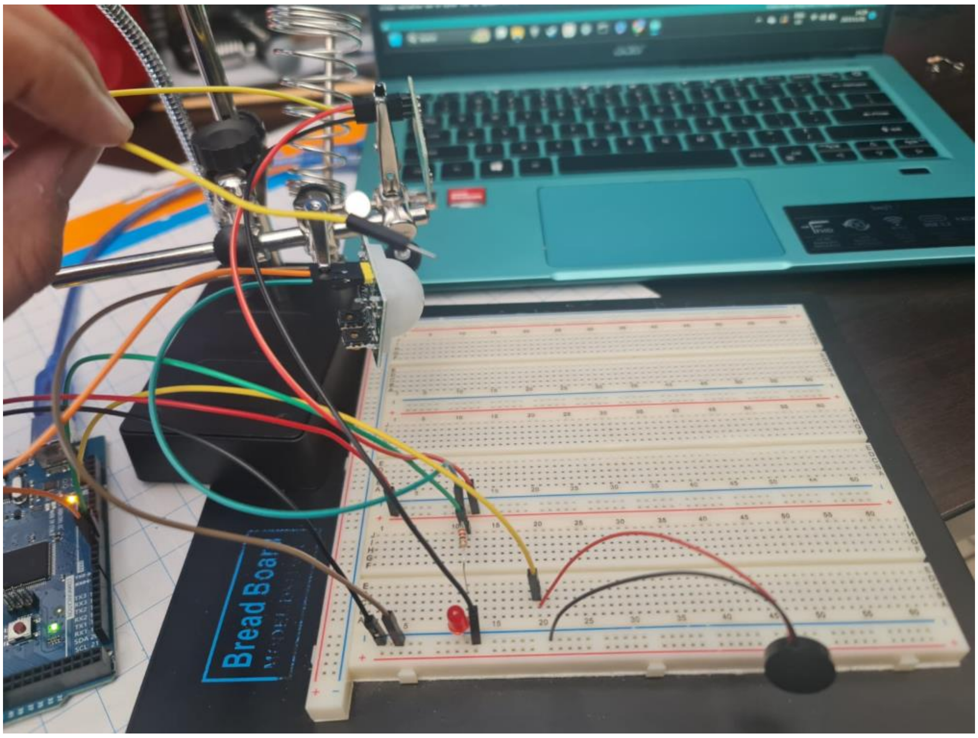
```
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
import serial

# Function to check the password
def check_password():
    entered_password = password_entry.get()
    if entered_password == "1234":
        status_label.config(text="Password is correct.")
        arm_button["state"] = "active"
        disarm_button["state"] = "disabled"
        disarm_password_entry.config(state="disabled")
        disarm_check_button["state"] = "disabled"
    else:
        status_label.config(text="Incorrect password!")

# Function to check the disarm password
def check_disarm_password():
    entered_password = disarm_password_entry.get()
    if entered_password == "4321":

        status_label.config(text="Disarm Password is correct.")
        disarm_button["state"] = "active"
    else:
        status_label.config(text="Incorrect disarm password!")

# Function to arm the security system
def arm_security_system():
    status_label.config(text="Security system armed.")
    arm_button["state"] = "disabled"
    disarm_button["state"] = "active"
    disarm_password_entry.config(state="normal")
    disarm_check_button["state"] = "active"
    arduino_serial.write(b'A')  # Send 'A' to arm the system

# Function to disarm the security system
def disarm_security_system():
    status_label.config(text="Security system disarmed.")
    arm_button["state"] = "active"
    disarm_button["state"] = "disabled"
    disarm_password_entry.config(state="normal")
    disarm_check_button["state"] = "active"
    arduino_serial.write(b'D')  # Send 'D' to disarm the system
# Function to quit the application
```

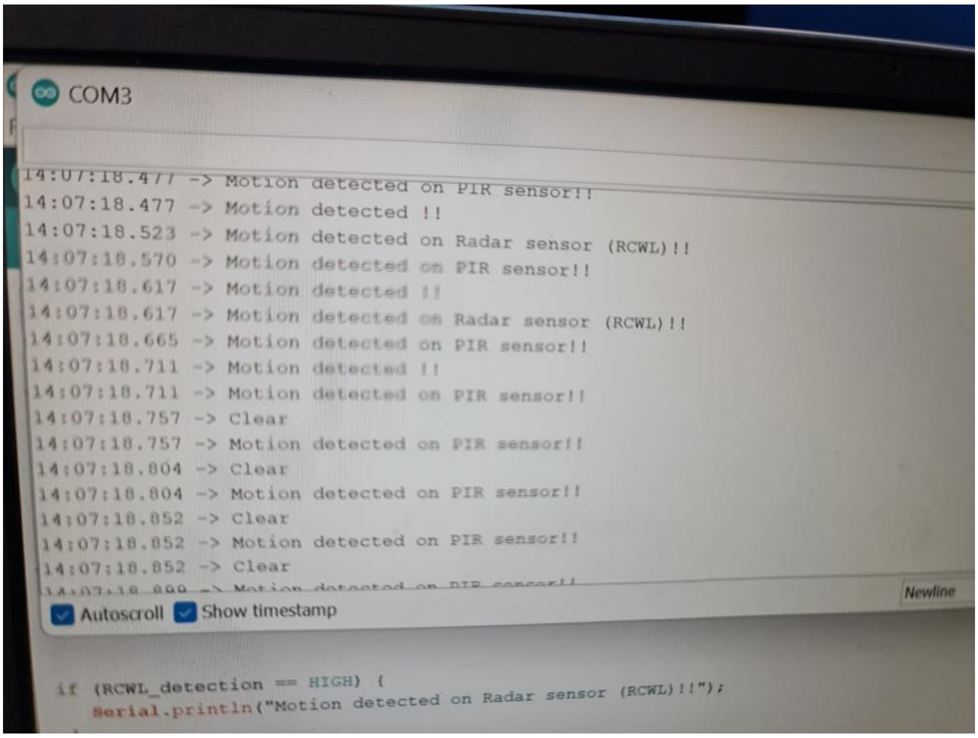
```
def quit_application():
    arduino_serial.close()
    window.destroy()

# Function to display emergency contact information
def display_help():
    help_text = "Emergency Contact Details:\nPolice: 10111\nAmbulance:
10177\nSystem Support Email: 22139454@dut4life.ac.za"
    messagebox.showinfo("Help", help_text)

# Main window
window = tk.Tk()
window.title("Security System")

# Window size and padding
window.geometry("400x250")

window.configure(padx=10, pady=10)

# Frame for password entry
password_frame = ttk.LabelFrame(window, text="Enter Password")
password_frame.pack(fill="both", expand="true")

password_label = ttk.Label(password_frame, text="Arm Password:")
password_label.pack()
password_entry = ttk.Entry(password_frame, show="*")  # Masking the password
with '*'
password_entry.pack()

# Frame for disarm password entry
disarm_password_frame = ttk.LabelFrame(window, text="Enter Disarm Password")
disarm_password_frame.pack(fill="both", expand="true")

disarm_password_label = ttk.Label(disarm_password_frame, text="Disarm
Password:")
disarm_password_label.pack()
disarm_password_entry = ttk.Entry(disarm_password_frame, show="*")  # Masking
the password with '*'
disarm_password_entry.pack()

# Frame for buttons
button_frame = ttk.LabelFrame(window, text="System Control")
button_frame.pack(fill="both", expand="true")

check_button = ttk.Button(button_frame, text="Check Arm Password",
command=check_password)
check_button.pack()
```

```python
arm_button = ttk.Button(button_frame, text="Arm System",
command=arm_security_system, state="disabled")
disarm_button = ttk.Button(button_frame, text="Disarm System",
command=disarm_security_system, state="disabled")
arm_button.pack(side="left", padx=5)
disarm_button.pack(side="right", padx=5)

disarm_check_button = ttk.Button(button_frame, text="Check Disarm Password",
command=check_disarm_password, state="disabled")
disarm_check_button.pack()

# Status Label
status_label = ttk.Label(window, text="", padding=(10, 0))
status_label.pack()

# Quit Button
quit_button = ttk.Button(window, text="Quit", command=quit_application)
quit_button.pack(side="right", padx=5)

# Help Button
help_button = ttk.Button(window, text="Help", command=display_help)
help_button.pack(side="left", padx=5)

# Initialize the serial connection to Arduino
arduino_serial = serial.Serial('COM3', 9600)

# Start the main event loop
window.mainloop()
```



Annexure G: Results of removing Radar module signal pin mid-operation





Annexure H: Proposed Cost of Materials

Part	Cost Per Unit	Quantity	Total	Supplier
Arduino Mega 2560	R 670,77	1	R 670,77	Mantech
RCWL-0516 Radar Module	R 23,00	3	R 69,00	Mantech
KE0054 PIR Motion Sensor Module	R 85,81	3	R 257,43	Mantech
I2C LCD	R 143,87	1	R 143,87	Mantech
PT1704 Piezo Buzzer	R 12,77	1	R 12,77	Mantech
Red LED	R 1,86	3	R 5,58	Mantech
Total Proposed Project Costs			R 1 159,42	