POLITEKNIK SULTAN SALAHUDDIN ABDUL AZIZ SHAH

SMART DOOR LOCK

NAME	REGISTRATION NO
HANIS NAZIRAH BINTI ROSDI	08DEU20F1005

JABATAN KEJURUTERAAN ELEKTRIK

1 2022/2023

POLITEKNIK SULTAN SALAHUDDIN ABDUL AZIZ SHAH

SMART DOOR LOCK

NAME	REGISTRATION NO
HANIS NAZIRAH BINTI ROSDI	08DEU20F1005

This report submitted to the Electrical Engineering Department in fulfillment of the requirement for a Diploma in Electrical Engineering

JABATAN KEJURUTERAAN ELEKTRIK

1 2022/2023

CONFIRMATION OF THE PROJECT

The project report titled " Smart Door Lock" " has been submitted, reviewed and verified as a fulfills the conditions and requirements of the Project Writing as stipulated

Checked by:

Supervisor's name : EMY SATIRA AZRIN BINTI MOHAMED HAKKE

Supervisor's signature:

Date :

Verified by:

Project Coordinator name :

Signature of Coordinator :

Date :

"I acknowledge this work is my own work except the excerpts I have already explained to our source"

1. Signature : 

Name : **HANIS NAZIRAH BINTI ROSDI**

Registration Number : **08DEU20F1005**

Date : **5 DECEMBER 2022**

DECLARATION OF ORIGINALITY AND OWNERSHIP

TITLE : SMART DOOR LOCK

SESSION: 1 2022/2023

1. I, **1. HANIS NAZIRAH BINTI ROSDI (08DEU20F1005)**

   is a final year student of **Diploma in Electrical Engineering, Department of Electrical, Politeknik Sultan Salahuddin Abdul Aziz Shah**, which is located at **Persiaran Usahawan, Seksyen U1, 40150 Shah Alam, Selangor**. (Hereinafter referred to as 'the Polytechnic').

2. I acknowledge that 'The Project above' and the intellectual property therein is the result of our original creation /creations without taking or impersonating any intellectual property from the other parties.

3. I agree to release the 'Project' intellectual property to 'The Polytechnics' to meet the requirements for awarding the **Diploma in Electrical Engineering** to me.

Made and in truth that is recognized by;

a) **HANIS NAZIRAH BINTI ROSDI**) .
(Identification card No: - 0203140081156)) .....
) **HANIS NAZIRAH BINTI ROSDI**

In front of me, **EMY SATIRA AZRIN BINTI MOHAMED HAKKE** (Click here to enter text.)) .....
As a project supervisor, on the date:) Click here to enter text.

ACKNOWLEDGEMENTS

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I am highly indebted to (PUAN EMY SATIRA AZRIN BINTI MOHAMED HAKKE) for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

ABSTRACT

In recent trends, smart buildings have become the base for the Internet of Things (IoT). The usage of the internet is increased by connecting the devices in the homes to make the places more comfortable, provident, delightful, and secure. The proposed approach addresses a security aspect in smart home technologies, namely the door lock system. The door lock system determines the security by allowing the owner to monitor the buildings with a Smartphone-controlled, Wi-Fi connected system using NodeMCU ESP32. A microcontroller is chosen for this project is built to experiment and develop the security breaches. Users can open or close the door lock by installing the developed android application in devices like tablets, smartphones and laptops. by providing the login credentials.

ABSTRAK

Dalam trend terkini, bangunan pintar telah menjadi asas untuk Internet of Things (IoT). Penggunaan internet dipertingkatkan dengan menyambungkan peranti di rumah untuk menjadikan tempat lebih selesa, menggembirakan dan selamat. Pendekatan yang dicadangkan menangani aspek keselamatan dalam teknologi rumah pintar, iaitu sistem kunci pintu. Sistem kunci pintu menentukan keselamatan dengan membenarkan pemilik memantau bangunan dengan sistem bersambung Wi-Fi yang dikawal telefon pintar menggunakan NodeMCU ESP32. Pengawal mikro dipilih untuk projek ini dibina untuk mencuba dan membangunkan pelanggaran keselamatan. Pengguna boleh membuka atau menutup kunci pintu dengan memasang aplikasi android yang dibangunkan dalam peranti seperti tablet, telefon pintar dan komputer riba. dengan menyediakan bukti kelayakan log masuk.

TABLE OF CONTENTS

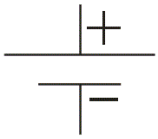
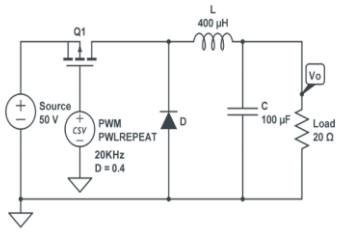
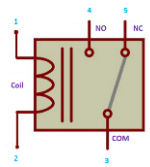
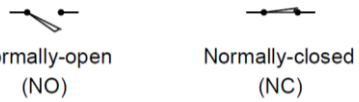
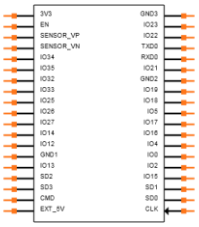
LIST OF TABLES

TABLE **TITLE** **PAGE**

LIST OF FIGURES

FIGURE **TITLE** **PAGE**

LIST OF SYMBOLS

Battery	
Step-down converter	
Relay module	
Limit switch	
Resistor	
NodeMCU ESP32	

CHAPTER 1

INTRODUCTION

1.1 Introduction

The project I want to implement is SMART Door Lock. The SMART Door Lock project combines software and hardware where it is controlled through electric current and electronically controlled using applications on a smartphone. The function is the same as the traditional lock mango but the SMART Door Lock does not require a regular lock. Locking or opening the door can be controlled on the app on a smartphone anywhere. Other additional features such as receiving notifications directly to the smartphone in the event of a hazard incident.

1.2 Background Research

With the rapid advancement of the IoT market, companies tend to focus on the time-to market and releasing product as fast as possible instead of developing a secure substantial product. This leaves many IoT product with inadequate protection against various forms of malicious attacks. IoT security is an ever growing problem and even if there is a significant amount of research on the topic there is not much substantial work about implementations or standardizations that could solve this problem. IoT security is of utmost importance as the aftermath of security breaches in IoT can be devastating. A breach in a smart car or smart door lock could lead to stolen products or even casualties in some extreme cases. Even if an undetected breach is not exploited but still existing it gives the product owner a false sense of security which is ethically unacceptable. Because of the inconsistency of IoT products, their architecture and the technology used it is impossible to develop consistent security measures that cover the entire spectrum of different devices. Therefore shall the IoT products be developed around safety standards instead of the other way around. For this research, I want to develop a secure smart door lock to access their home, office and shop. The SMART

Door Lock will be our use case in this research and will represent the typical IoT device in our society.

1.3 Problem Statement

Traditional house locks are getting out of date. SMART Door Lock are commonly used nowadays with most homes. SMART Door Lock look more sophisticated and make home security safer. With SMART Door Lock you don't have to physically open the door. SMART Door Locks are also not like other digital doors but are otherwise operated via Wi-Fi. Time is not wasted finding the door lock because the SMART Door Lock can remotely lock the door.

1.4 Research Objectives

The main objective of this project is to design and build a SMART Door Lock that allows users to unlock a door via Wi-Fi using application on a smartphone.

More specifically the principle objective of this research are:

1. To implement a direct smartphone notification to the user when the house is breached.
2. To develop a system that users can control door locks remotely and anywhere.

1.5 Scope of Research

1. This project is focusing at the community including everyone who owns the home, all shop owners and all office workers.
2. The emphasis is control open and close doors using the mobile app.
3. The main controller is using NodeMCU ESP32.

1.6 Project Significance

Traditional keys are easy to lose, can be copied easily, and are simply outdated when it comes to modern security locks. Keyless entry systems require every authorized user to have a unique credential that provides them access to the building. This reduces the chances of an intruder stealing or copying a traditional key and gaining access. SMART Door Lock it is easier to manage. With the traditional lock and lock method, every time the occupant leaves the house, the lock needs to be reassembled and a new key will have to be created for security purposes. This increases the risk of intruders. An electronic keyless system can be reconfigured without removing the entire system. With everything available in the online system, the owner can add a new and ineligible user without having to replace the key card. SMART Door Lock has remote access. Cloud-managed systems also allow remote access. This feature is especially useful. The owner can control by locking or opening the door remotely and anywhere.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Among the existing smart door lock systems, designed using different technologies, a few selected systems are discussed below, along with their features. As soon as a person is detected, the door would open, and they would be welcomed. Among the existing smart door lock systems, designed using different technologies, some selected systems are discussed below, along with their features. Once the 'DOOR LOCK' button is pressed, the door will open.

2.2 Motor Skill Challenges in Autistic Children (Literature Review Topic 1)

A door enables to enter a room without breaking through a wall. Also, a door enables for privacy, environmental or security reasons. The example that challenges which is the biometric system sometimes is sensitive and will not be able to sense the biological pattern of the employer's fingerprint due to sweat and other factors. Next, people tend to misplace their key or RFID card. Apart from that, people tend to forget their pin number for a door lock. The objective of this paper is to present a secret knock intensity for door lock security system using ESP32 and mobile. This project works by using a knock intensity and send the information to mobile application via wireless network to unlock or lock the door.

2.2.1 Previous Research (Subtopic Literature Review Topic 1)

This chapter extend the literature reviews that cater the information in accordance with the objectives of this project. The relevant information and other extra features were gathered as shown below.

(Door Sensor)

Arduino et al. [2] The door sensor is widely used in security area, it is used to detect or monitor entrances such as door. The door sensor is also known as “entry sensors” or “contact sensors”. Door sensor includes two components which is one reed switch with two pins and one magnet. Just like a normal switch or button, we do not need to distinguish the two pins of the reed switch. The magnet is attached to the door(moving part and the reed switch is attached to the door frame(fixed part). The two components are in contact when the door is closed. When the magnet is close to the reed switch, the reed switch circuit is closed. When the magnet is far from the reed switch, the reed switch circuit is open. If we connect a pin of the reed switch to GND, thee other pin of the reed switch to Arduino’s input pin with a pullup resitor(internal or external). When the magnet is closed to the reed switch, the value in Arduino’s input pin is LOW. When the magnet is far from the reed switch, the value in

Arduino’s input pin is HIGH. So, to check the state of the door, we just check the state o Arduino’s input pin where if the state is LOW, the door is closed. If the state is HIGH, the door is open. To detect the door-opening or door-closing events, we can detect the state change on the Arduino input pin. If the state changes from LOW to HIGH, the door-opening event is detected. If the state changes from HIGH to LOW, the door-closing event is detected. But for my SMART Door Lock, I use the NodeMCU ESP8266 microcontroller which is smaller in size and has Wi-Fi.

(Design and Development of Smart Lock)

N.Hashim et. Al [6] In this design, an Android smartphone with WiFi access is functioning as the transmitter. The receiver parts consist of a PIC, WiFi module, relay, and solenoid door lock. When the correct IP address and port number are received, the solenoid lock will be activated and it will unlock the door for 5 seconds. Besides, there is a reset button to unlock the door from inside the house. This is useful for emergency exit especially for home owner. The development of this project consists of software and hardware implementation. Two main soft wares used in this design are Eclipse for building the Android application and C language

for PIC to control the door locking. As for the hardware part, the design of the door lock was chosen carefully in order to increase the security of the door locking besides saving energy. In this project solenoid lock was chosen where the lock design is focused to an electrically operated door system that has high reliability. The locking mechanisms are holding a latch keeper in locking position to prevent opening of the door. This condition of system revealed that the solenoid is in unlocked position when it is energized and does not require electricity when solenoid is in locked position. Thus this situation leads to electric saving characteristic design. This project concentrates on Android application. The Android Development Kit (SDK) [5-7] is used where it provide libraries needed to interface with hardware. VOL. 11, NO. 5, MARCH 2016 ISSN 1819-6608 ARPN Journal of Engineering and Applied Sciences ©2006-2016 Asian Research Publishing Network (ARPN). All rights reserved. www.arpnjournals.com 3310 Android system architecture is the Android that created on top of the open-source Linux 2.6 Kernel. The Android team chooses to use this Kernel. It provides a proven core features to develop the Android operating system.

(Smartphone activated door lock using Arduino Uno)

Arun et al. [3] The aim is to control a door lock without keys, and using a smartphone to accomplish this. The communication medium will be the internet (WiFi module-ESP8266). The Arduino UNO R3 to control the servo and interface with the ESP8266 (GPIO pin-0). The concept is very simple we are going to control the ESP8266 (GPIO pin 0) from our smartphone with an app. The GPIO pin 0 state will be recognised by Arduino using the digitalRead command and does the required operations on the Servo. The ESP-8266 WiFi Modules will help our project to get connected to the cloud. This is very cheap and easy to use. In this project we will be using only the GPIO pin-0 of the ESP8266. The High Torque Servo to turn the lock lever when signal is sent from the Arduino. It is powered by the Arduino 5v pin itself and it works well with it. (Only when the Arduino is powered with USB cable.) The Sheet Metal is used to make the case for the servo motor which allows it to properly hold the motor on the door and the rotating part properly fixed to the lever of the lock. The case is up to your creativity you can use any other material/thing to make the case. TO HOLD THE MOTOR ON THE DOOR. This is used only in the

making of the project to program the ESP8266 module as it can't be programmed directly from Arduino.

(Smartphone activated door lock using NodeMCU ESP8266)

IoT et al. [4] In this project, we are going to make an IOT based Door Lock System using NodeMCU, Solenoid Lock, and Adafruit IO. Here NodeMCU will act as the main controller and connect the user to the door lock system using the Internet. This allows the user to locker unlock his home's door lock by using a smartphone from anywhere in the world. Connections for this IoT Smart Door Lock are very simple as we are only connecting a solenoid lock, relay module, and a buzzer with NodeMCU ESP8266. The input pin of the relay is connected to the D5 pin of NodeMCU while VCC and Ground pins are connected to Vin and GND pin of NodeMCU. The positive pin of the buzzer is connected to the D6 pin of NodeMCU, and the GND pin is connected to the GND of NodeMCU. [5] With built in ESP8266 Wi-Fi module, it can be connected to the Internet without using additional component. It act as the mediating way of communication between software and hardware devices. An open-source fimware and development kit that is perfect to build IoT prototype. It includes firmware which runs on ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

(An IoT Applcations over the internet using Adafruit IO)

Adafruit IO is an open data platform that allows to aggregate, visualize, and analyze live data on the cloud. Using Adafruit IO, we can upload, display, and monitor your data over the internet, and make your project IoT enabled. We can control motors, read sensor data, and make cool IoT applications over the internet using Adafruit IO. For test and try, with some limitation, Adafruit IO is free to use. We have also used Adafruit IO with Raspberry Pi,

Arduino and ESP32. Adafruit IO represents Adafruit's desire to continually develop solutions that support makers and hardware enthusiasts in general. It is essentially an integrated cloud service for IoT devices designed primarily, like most other cloud services, to retrieve and store data. However, to make things more suitable for use by

its target audience, Adafruit included features that facilitate real-time data visualization using graphs, gauges, etc., Email/webhook notifications, and Control of devices from the internet. It also features IFTTT and zapier integrations which give us the ability to interact with 100s of web services like RSS feeds and Twitter. It comes with a limited but wholly sufficient free plan as well as a paid plan which gives full unrestricted access for around 10$/month.

2.3 Control System (Literature Review Topic 2)

Nowadays the home security system is very poor. This project is consists of a SMART Door Lock system in this was developed on a great solution to improve the home safety system. Arduino IDE software in this I used Wi-Fi devices to operate the door lock in the home. The microcontroller ESP32 is connected to relay module that controlling the door lock. In this method, all data are controlling by the Blynk application in that app I created a button to control the home door lock which is open and close form the Blynk app we can control from anywhere is function is called as IoT.

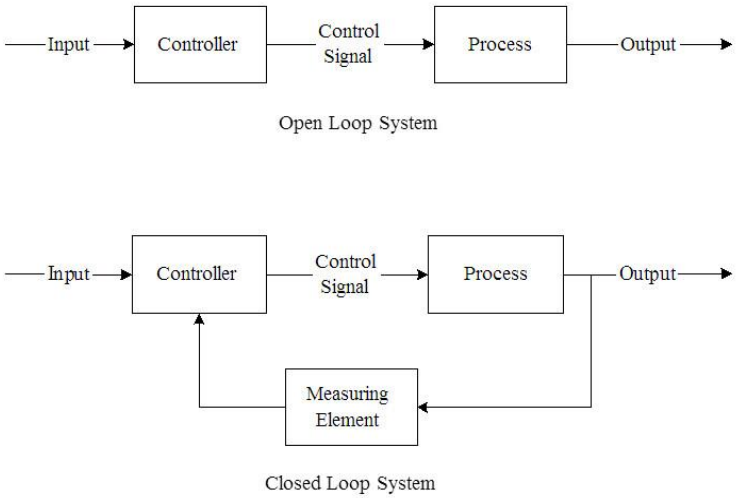


Figure 2. 1: Block diagram of open loop and closed loop system

2.3.1 Microcontroller

A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip. Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in vehicles, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances, among other devices. They are essentially simple miniature personal computers (PCs) designed to control small features of a larger component, without a complex front-end operating system (OS). A microcontroller is embedded inside of a system to control a singular function in a device. It does this by interpreting data it receives from its I/O peripherals using its central processor. The temporary information that the microcontroller receives is stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. It then uses its I/O peripherals to communicate and enact the appropriate action. Microcontrollers are used in a wide array of systems and devices. Devices often utilize multiple microcontrollers that work together within the device to handle their respective tasks.

2.3.2 Programmable Logic Control (PLC)

A programmable logic controller (PLC), or programmable controller is an industrial digital computer that has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, machines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnosis. PLCs communicate, monitor and control automated processes like assembly lines, machine functions, or robotic devices. A PLC's functions are divided into three main categories: inputs, outputs and the CPU. PLCs capture data from the plant floor by monitoring inputs that machines and devices are connected to.

2.3.3 NodeMCU ESP32

NodeMCU is an open source Lua based firmware for the ESP32 and ESP8266 WiFi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif ESP-IDF. With the NodeMCU-ESP32, comfortable prototyping is possible with simple programming via Luascript or the Arduino IDE and the breadboard-compatible design. This board has 2.4 GHz dual-mode Wifi and a BT wireless connection. In addition, a 512 KB SRAM and a 4MB flash memory are integrated into the microcontroller development board. The board has 21 pins for interface connection, including I2C, SPI, UART, DAC, and ADC.

CHAPTER 3

RESEARCH METHODOLOGY

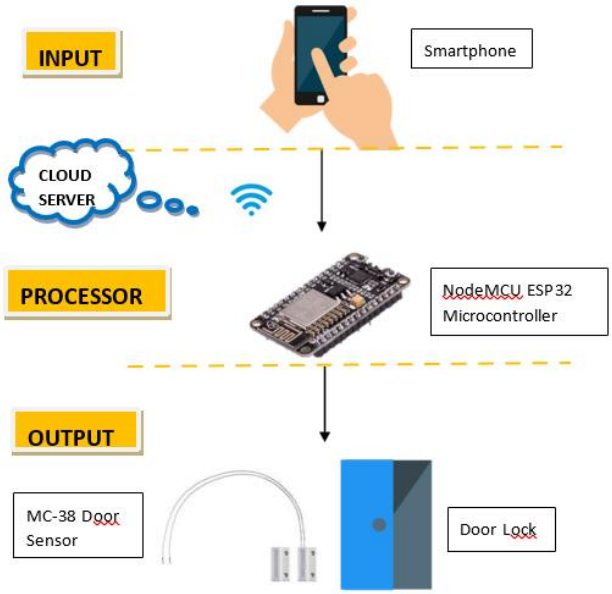
3.1 Introduction

The main goal of this project was to design and build a door lock system that allows users to unlock a door via application on a smartphone, through a door sensor implanted on the door. In this methods chapter, the process of implementing this mechanism will be discuss with detailed. The research was started by confirming the need of potential users for such a system to design and build a door lock system.

3.2 Project Design and Overview.

As mention in previous chapter, the design controller is using closed-loop system with ESP32 as the main controller. The design of the controller circuit using ESP32 is realize using Proteus Software and then convert to PCB circuit. It consists of the battery, solenoid lock, step-down converter, relay module, and Wi-Fi. When the switch button is ON, the Wi-Fi will start blinking and connect to the Blynk. The door will lock or unlock when controlled through a smartphone. Limit switch will be attached to the door for the security system. The ‘Home breached!’ notification will be received on a mobile application.

3.2.1 Block Diagram of the Project



3.2.2 Flowchart of the Project 2

Figure 3. 1 shows the circuit diagram of the whole system. It is shows the process of the smart door lock functioning. At first, people with a mobile app that is connected to a smart door lock have to open or close the door. If people choose to open the door, the solenoid is unlocked which is the door is open. Then, the door sensor detects the door has opened it will display the status unlock in a mobile app. If people choose to close the door, the solenoid is locked which is the door is closed. Then, the door sensor detects the door has closed it will display the status lock in a mobile app. In case their people want to open or unlock the door without a mobile application the people who own the mobile application will receive a ‘Home breached!’ notification.

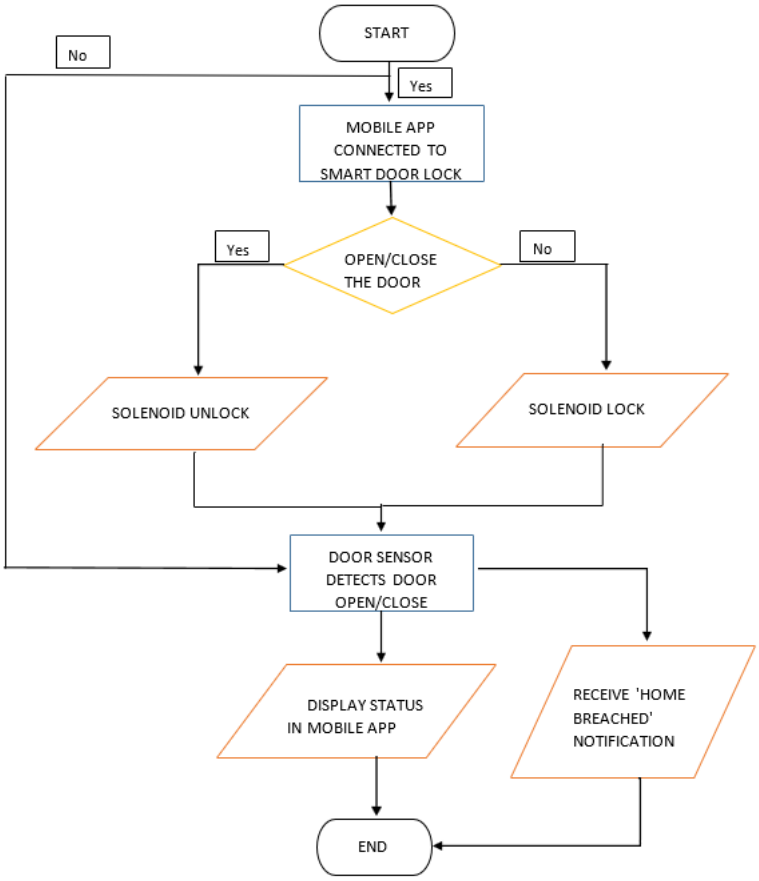


Figure 3. 1: Flow chart of operation of the system

*Images may be subject to copyright

3.2.3 Project Description

The purpose of this project is to develop a system that can lock or unlock the door using the application on a smartphone. This innovation uses a limit switch for security. The notification will be received on a smartphone when the door is breached. The mobile application will then display the status of the door whether it is locked or unlocked to prevent breaching. Solenoid lock that functions as a door lock and unlock. This function can be controlled in the application.

3.2.4 Schematic Circuit

Figure 3. 2 shows the overall circuit diagram of this project that each electrical component is connected with the ESP32.

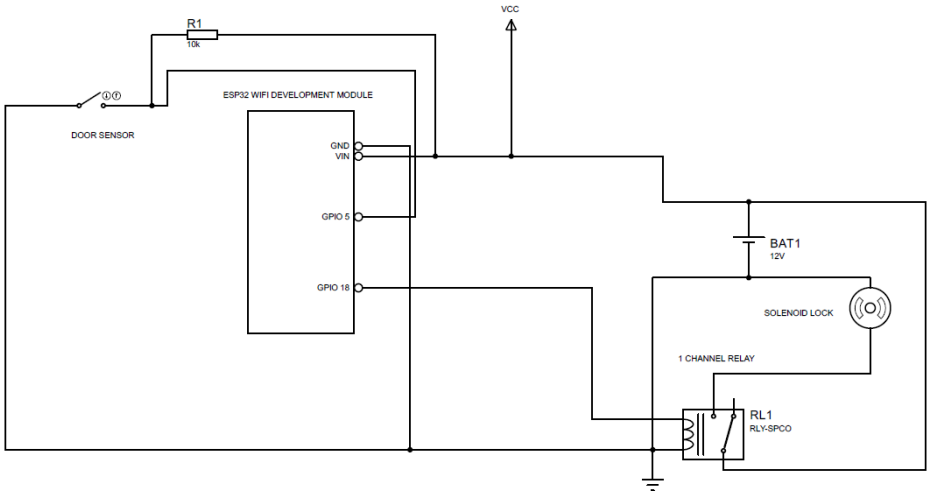


Figure 3. 2: Circuit Diagram

*Images may be subject to copyright

3.2.5 Description of Main Component

This project will undergo the process of design development. The process of designing is one of the crucial parts of developing foldable SMART Door Lock. The hardware that use in this project is divided into 5 types which are NodeMCU ESP32, limit switch, solenoid lock, step-down converter and relay module. A relay module is used to lock or unlock the door. This relay module will be connected to the solenoid lock. Next, ESP32. ESP32 allows to construct of programs in code segments to perform individual tasks. The code will be programmed in ESP32 to give instructions to other hardware to work. A step-down converter converts high voltage to low voltage, usually transforming AC current to DC current.Then, the next hardware is the door sensor. The limit switch will be used in this project. The limit switch is used for a foldable part and it is two magnetic sensors that are attached to the door. A Wi-Fi module is used to connect the device to the Blynk application. Four 3.7V batteries are also used as a power source for the relay module.

3.2.5.1 NodeMCU ESP32

NodeMCU is an open source Lua based firmware for the ESP32 and ESP8266 WiFi SOC from Espressif and uses an on-module flash-based SPIFFS file system. NodeMCU is implemented in C and is layered on the Espressif ESP-IDF. With the NodeMCU-ESP32, comfortable prototyping is possible with simple programming via Luascript or the Arduino IDE and the breadboard-compatible design. This board has 2.4 GHz dual-mode Wifi and a BT wireless connection. In addition, a 512 KB SRAM and a 4MB flash memory are integrated into the microcontroller development board. The board has 21 pins for interface connection, including I2C, SPI, UART, DAC, and ADC.

3.2.5.2 Solenoid Lock

In conventional door lock, there is key to pull or push the latch, and we have to operate it manually, but in solenoid lock, the latch can be operated automatically by applying a voltage. Solenoid lock has a low-voltage solenoid that pulls the latch back into the door when an interrupt (Pushbutton, Relay) is activated. The latch will retain its position until the interrupt is enabled. The operating voltage for the solenoid lock is 12V. That can also use 9V, but it results in slower operation. Solenoid door locks are mainly used in remote areas to automate operations without involving any human effort.

3.2.5.3 Limit Switch

They are used for controlling machinery as part of a control system, as a safety interlocks, or to count objects passing a point. A limit switch is an electromechanical device that consists of an actuator mechanically linked to a set of contacts. When an object comes into contact with the actuator, the device operates the contacts to make or break an electrical connection. Limit switches are used in a variety of applications and environments because of their ruggedness, ease of installation, and reliability of operation. They can determine the presence or absence, passing, positioning, and end of travel of an object. They were first used to define the limit of travel of an object; hence the name “Limit Switch”.

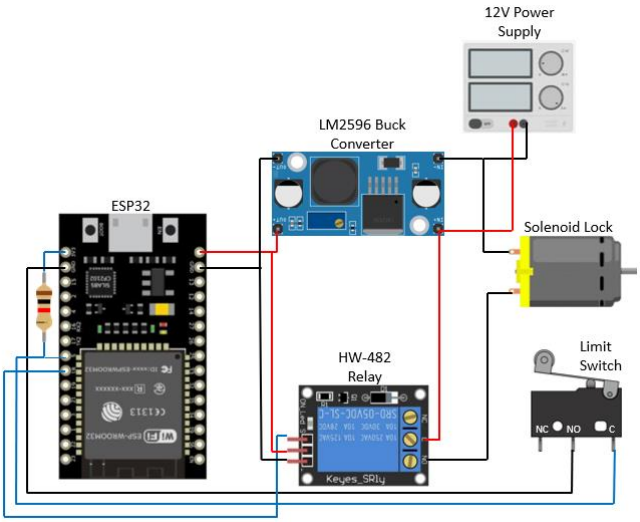
3.2.5.4 Step-down Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter). Its name derives from the inductor that “bucks” or opposes the supply voltage.

3.2.5.5 Relay Module

Relay Module is an electrically operated switch. It is responsible for completing the connection of solenoid lock and the power supply.

3.2.6 Circuit Operation



CHAPTER 4

RESULTS AND DISCUSSION

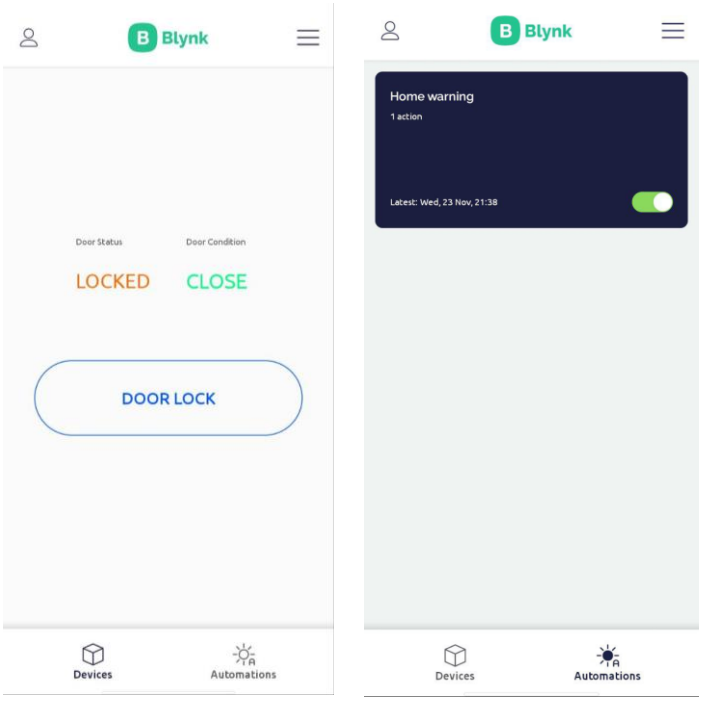
4.1 Introduction

I successfully implemented a SMART Door Lock system with seamless integration of cloud backends, on-board logic units such as ESP32 and user-friendly mobile applications. All the features suggested in the beginning, I implemented and tested, including some additional features. I ensure that requests and user information are secure. I was able to build a mockup as a door system, using several materials.

4.2 Results and Analysis

DATA	RESULT
Times period to lock or unlock the door using an application on a smartphone	LOCK = 6 seconds UNLOCK = 2 seconds
Times period to receive a notification when the door is breached	Less than 5 seconds
Range of Wi-Fi connection between application and ESP32	5 meters
Control the door lock using the Blynk application	All devices (android and ios)

When the device is connected to an application on a smartphone, the door can be controlled using the application. The range of Wi-Fi connection between application and ESP32 is 5 meters. The ‘DOOR LOCK’ button is pressed to open the door. The solenoid lock that functions as a door lock and unlock will unlock in less than 2 seconds. Within 6 seconds if the door is not opened, the solenoid lock will re-lock which is the door will be locked again. For security when you are not at home, 'Home warning' is activated to receive notifications through the application when there is a home invasion. Notification received in less than 5 seconds. All devices including android and ios can use this SMART Door Lock.



4.3 Discussion

The person who has been authenticated will be able to use the application and lock or unlock the door with a single click. A test is performed to test the ESP32 output in giving output on the solenoid lock and relay module for opening and closing the systems to check systems automatically lock or unlock the door. The results show that when the solenoid is in the lock position, the door status will display ‘LOCKED’ and the door condition will display ‘CLOSE’, while when it is in the unlock position, the door status will display ‘UNLOCKED’ and the door condition will display ‘OPEN’. When the lock is in the rest state, it remains locked.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter concludes the work done in this report.

5.2 Conclusion

In conclusion, this project was accepted by the public. Despite the fact that at the beginning of the project there are some problems such as difficult use and lack of value, the project can eventually be fixed and generally accepted. Such a project will take a long time to meet the established criteria. With the cooperation and guidance provided by the project supervisor, the project can be completed successfully. After various studies and experiments that have been done on this project, I was able to prove that the use of SMART Door Lock has successfully helped all home owners, all shop owners and all office workers as well as positively impacted the public as a whole, our project has met the criteria or objectives of the project as it can facilitate and can help people live a safe life. The system used is well received because it is easy to operate. Comparison with other methods further strengthens the usefulness of this project.

5.3 Suggestion for Future Work

With this project on the market, I believe that the demand for better security in the market can be met. Therefore, I believe and hope that this project will be further expanded. In this regard, I hope that with the creation of this innovation can attract more interest and anyone who wants to create or improve tools to help anyone. This innovation can not only meet our needs but also ease the burden. With this, it will not only help society but also encourage young people to continue to think creatively. Perhaps new ideas can give these innovations greater and can be widely applied not only to the door but also to other things.

5.4 Chapter Summary

The goal of this project was to provide an easy and convenient method for unlocking a front door by removing the need for the old-fashioned key. I start by evaluating the need for such a system by analyzing the results. The project has three main components: a ESP32 and a mobile application. The limit switch is attached to the door and is responsible for monitoring the door status and for security. Users can open the door by either tapping a button on the mobile app. The solenoid lock, unlock the door and the door will be open. The server is a Blynk application. It seamlessly runs door control and provides early warning for breached homes. If the 'Home warning' notification is activated and someone opens the door the warning message automatically sends. Blynk App makes it easy to quickly deploy a cloud application. To make sure that you don’t have unexpected guests, you can pick the days and times that your friends can get in. Furthermore, users can keep track of status the door home using the application that display the door condition either open or close. This project integrates a board computer like the ESP32 and a mobile application built in Blynk. These three components come together to create a seamless and convenient way for you and your family to control the doors of your home. Now you don’t have to worry about losing your key, getting locked out, or having your hands full with groceries, because the SMART Door Lock system has you covered.

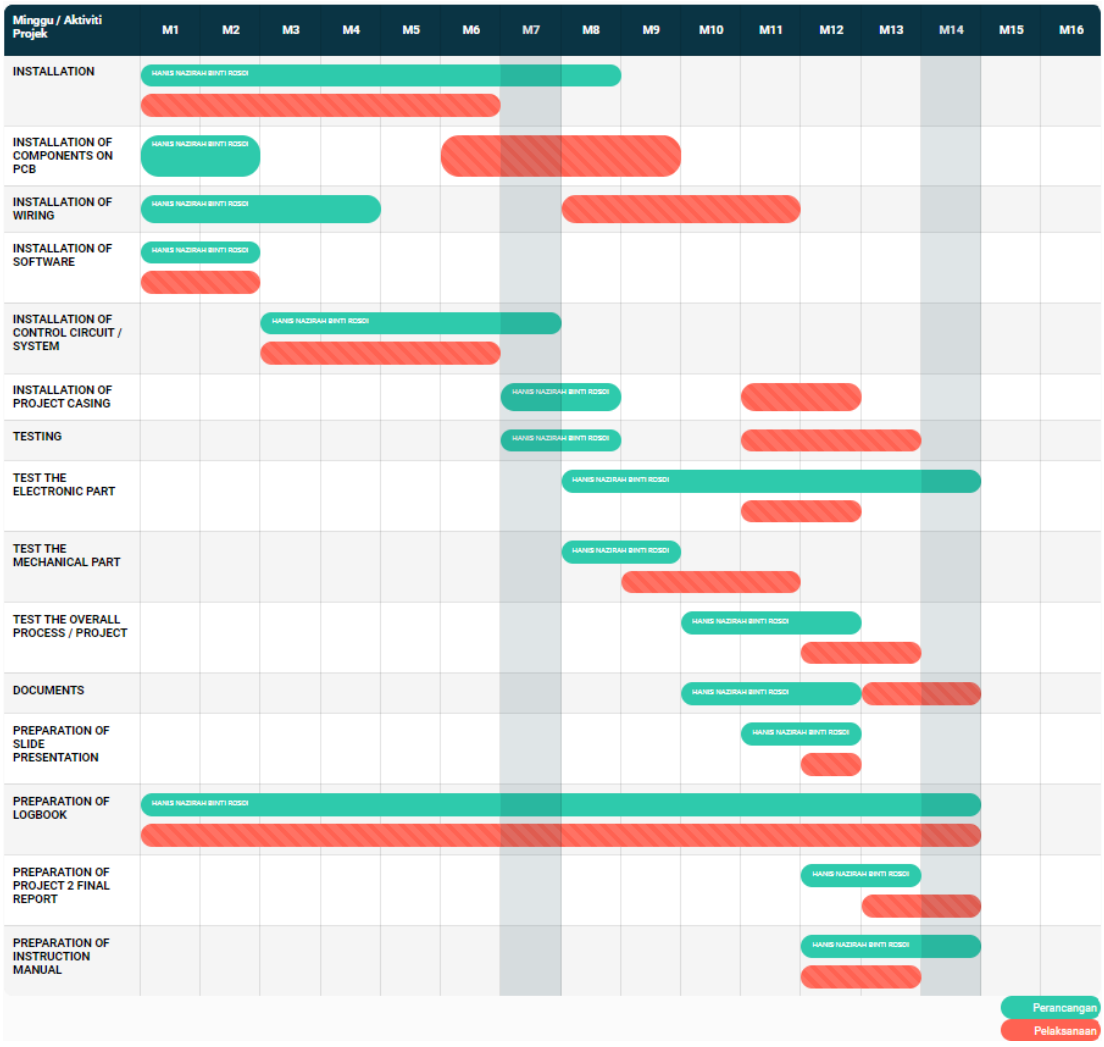
CHAPTER 6

PROJECT MANAGEMENT AND COSTING

6.1 Introduction

This project involves the cost of purchasing components and materials throughout its implementation. Components involving cost are hardware Solenoid Lock, NodeMCU ESP32, Limit Switch, Relay Module, Step-down Converter and 3.7V Battery. These components are purchased through online purchase methods to make it easier as well as save on costs. The overall gross budget estimate in the implementation of this project is RM138.08 and other expenses is postage at RM34.80 as shown in 6.3 according to this budget cost, this project is can be considered as a less costly project compared to other projects that can cost over a thousand ringgit. The cost of the project is also in line with one of the key features of a good project developer that is low cost but have a high quality project. Project activities are shown in the form of a Gantt Chart in 6.2 from week one to week 14.

6.2 Gant Chart and Activities of the Project



6.3 Cost and Budgeting

No.	Component and materials	The unit price	Quantity	Total
1	Solenoid lock	RM18.50	1	RM18.50
2	NodeMCU ESP32	RM35.00	1	RM35.00
3	Limit Switch	RM1.40	1	RM1.40
4	Relay Module	RM3.80	1	RM3.80
5	Step Down Converter	RM3.40	1	RM3.40
6	3.7V Battery	RM2.30	4	RM9.20
7	Other materials		7	RM23.98
	Total :			**RM95.28**
	List of other costing			
1	Transportation			RM8
2	Postage			RM34.80
3	Craft Work			
4	Internet			
5	Application			
	Total :			**RM42.80**
			Overall total	**RM138.08**

REFERENCES

[1] Kiranmai Chigurupati. (2021). Smart Door Lock System – Prototype. (1st ed.) [Online]. Available: https://www.learnelectronicsindia.com/post/smart-door-lock-system-prototype

[2] ArduinoGetStarted. (2021). Arduino-Door Sensor. (1st ed.) [Online]. Available: https://arduinogetstarted.com/tutorials/arduino-door-sensor

[3] Arun. (2018). WiFi Smart Door (Simplest Method). (1st ed.) [Online]. Available: IOT Smart Door (Simplest Method) - Arduino Project Hub

[4] IoT Design. (2020). IoT based Smart Door Lock System using NodeMCU (1st ed.) [Online]. Available: https://iotdesignpro.com/projects/iot-based-smart-door-lock-systemusing-nodemcu

[5] Ummi Annisa Binti Norarzemi, Mohd Shamian Zainal, Omar Abu Hassan, Mohd Hakimi Zohari, Danial MdNor, Shipun Anuar Hamzah, Masnani Mohamed, "Development of Prototype Smart Door System With IoT Application," Universiti Tun Hussein Onn Malaysia, Johor, Vol. 1No.1(2020) 245–256, 2020

[6] N. Hashim, N. F. A. M. Azmi, F. Idris, N. Rahim, "Smartphone Activated Door Lock using WiFi," Universiti Teknikal Malaysia, Melaka, OL. 11, No. 5, 2016

[7] G. Sowmya, G. Divya Jyothi, N Shirisha, K Navya, B Padmaja, "Iot Based Smart Door Lock System," Institute Of Aeronautical Engineering, Hyderabad, Telangana, 7 (3.6) 223225, 2018

[8] Aleksander IBRO, Augusto WONG, Mario ZYLA, "Smart Door Lock," Worcester Polytechnic Institute, MQP MXC – 0580, 2019

[9] Shashidhar Rudregowda, " Smart Door Lock System," International Journal for Modern Trends in Science and Technology, Karnataka, India, ISSN: 2455-3778, 2019

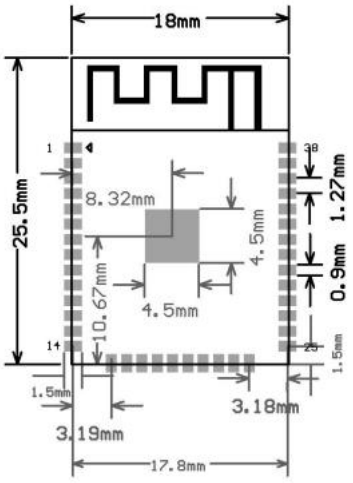
APPENDICES

APPENDIX A- DATA SHEET

2. Pin Definitions

2.1 Pin Layout

Figure 1: Top and Side View of ESP-32S



2.2 Pin Description

ESP-32S has 38 pins. See pin definitions in Table 3.

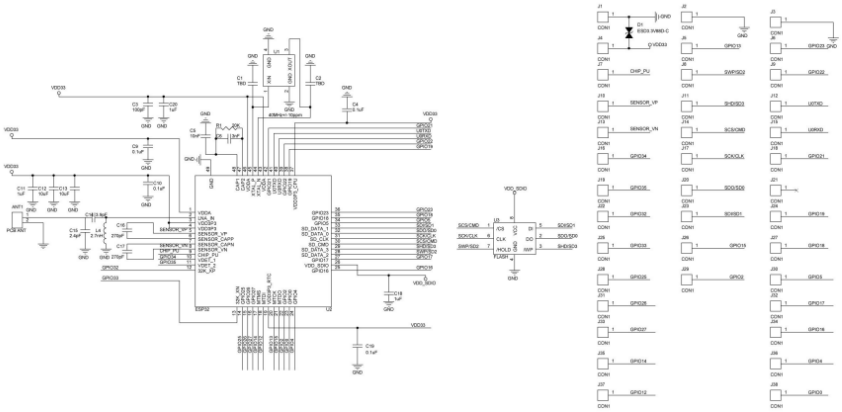
Table 3 Pin Descriptions

NO	Pin Name	Function
1	GND	Ground
2	3V3	Power supply
3	EN	Chip-enable signal. Active high
4	SENSOR_VP	GPI36, SENSOR_VP, ADC_H, ADC1_CH0, RTC_GPIO0
5	SENSOR_VN	GPI39, SENSOR_VN, ADC1_CH3, ADC_H, RTC_GPIO3
6	IO34	GPI34, ADC1_CH6, RTC_GPIO4
7	IO35	GPI35, ADC1_CH7, RTC_GPIO5
8	IO32	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
9	IO33	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
10	IO25	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
11	IO26	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1

12	IO27	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
13	IO14	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
14	IO12	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
15	GND	Ground
16	IO13	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
17	SHD/SD2	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
18	SHD/SD3	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
19	SCS/CMD	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
20	SCK/CLK	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
21	SDO/SD0	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
22	SDI/SD1	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
23	IO15	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
24	IO2	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
25	IO0	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
26	IO4	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1,

		SD_DATA1, EMAC_TX_ER
27	IO16	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
28	IO17	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
29	IO5	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
30	IO18	GPIO18, VSPICLK, HS1_DATA7
31	IO19	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
32	NC	-
33	IO21	GPIO21, VSPIHD, EMAC_TX_EN
34	RXD0	GPIO3, U0RXD, CLK_OUT2
35	TXDO	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
36	IO22	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
37	IO23	GPIO23, VSPID, HS1_STROBE
38	GND	Ground

5. Schematics



APPENDIX B- PROGRAMMING

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID           "TMPL_zeFZyIN"
#define BLYNK_DEVICE_NAME           "Quickstart Device"
#define                                                    BLYNK_AUTH_TOKEN
"O_ykmN9hkzxQrqmhTN4ZI9apTBniFc0n"


// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial


#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define SOLENOID 18
#define BZ 2
#define DOOR 5
#define IR2 17
```

```
#define Fan2 19
#define Light1 16
#define Light2 5

#define vCalibration 83.3
#define currCalibration 0.50


int DOORSENSOR=0;
int TDLY=0;
int F1Stat=0;
int F2Stat=0;
int L1Stat=0;
int L2Stat=0;
String DOORSTATUS="LOCKED";
int mode=0;
int SECURITY=0;
int Counter=0;
int DOORStat;
int IR2Stat;
int Check=0;
int TIMERx=0;

// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;
const int potPin2 = 35;
const int potPin3 = 32;
const int potPin4 = 33;
const int potPin5 = 25;




float ADC1,ADC2,ADC3,ADC4;

// variable for storing the potentiometer value
int potValue = 0;
float h=0,t=0;
float hx=0,tx=0;
int PIRSTAT=0;
int BIT=0;
int ALM1=0,ALM2=0,ALM3=0,ALM4=0;
int Ready=0;
int Ml=0;
String DOORCond="CLOSE";
String MinS="00";
String HourS="00";
String SecS="00";
int DataIn=0;
String DATA="";
```

```
String Temp1x="";
String PHx="";
String Temp2x="";
String Temp1y="";
String PHy="";
String Temp2y="";
String Temp3y="";
String Temp3x="";
String Temp4y="";
String Temp4x="";
String currentTime;
String currentDate;
String TimerGet="00:00:00";
int MODE=0;
int Hour=0;
int Min=0;
String MOTION="NONE";
String ReedDoor="NORMAL";
String ReedWindow="NORMAL";
String GASStat="NORMAL";
int Sec=0;
float Smoke,Mot,ReedW,ReedD;
float LEVEL=0;
int ALM=0;
int Val=100;
int Index=0;
float CV=0;
int CKN=0;
//---------------------------------------------------------------
int TDIS=0;
int Rly1=0;
int wait=0;
int Rly2=0;
int Rly3=0;
int Rly4=0;
int Rly5=0;
int pos1=0;
int pos=0;
int CK=0;
int STAT=0;
int COUNTER=0;
float AMP=0;
float SPEED=100;
float val=0;

//-------------------------------------------------------

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
```

```
// Set password to "" for open networks.
char ssid[] = "HOME";
char pass[] = "12345678";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes

BLYNK_WRITE(V10)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable
  Rly1=pinValue;
   if (pinValue==1){

    digitalWrite(SOLENOID,HIGH);
    F1Stat=1;
    DOORSTATUS="UNLOCKED";
    Blynk.virtualWrite(V0,DOORSTATUS);
  }

  }

BLYNK_WRITE(V11)
{
  int pin2Value = param.asInt(); // assigning incoming value from pin V1 to a variable
  Rly2=pin2Value;
  if (pin2Value==1){
  //  Serial.println("Fan 2 = ON");
    digitalWrite(Fan2,LOW);
    F2Stat=1;
  }
   if (pin2Value==0){
  //   Serial.println("Fan 2 = OFF");
     digitalWrite(Fan2,HIGH);
     F2Stat=0;
  }
  // process received value
}
BLYNK_WRITE(V12)
{
  int pin3Value = param.asInt(); // assigning incoming value from pin V1 to a variable
  Rly3=pin3Value;
  if (pin3Value==1){
  //  Serial.println("Light 1 = ON");
    digitalWrite(Light1,LOW);
    L1Stat=1;
  }
  if (pin3Value==0){
```

```
  //  Serial.println("Light 1 = OFF");
    digitalWrite(Light1,HIGH);
    L1Stat=0;
  }
  // process received value
}

BLYNK_WRITE(V13)
{
  int pin4Value = param.asInt(); // assigning incoming value from pin V1 to a variable
  Rly4=pin4Value;
  if (pin4Value==1){
  //  Serial.println("Light 2 = ON");
    digitalWrite(Light2,LOW);
    L2Stat=1;
  }
  if (pin4Value==0){
  //  Serial.println("Light 2 = OFF");
    digitalWrite(Light2,HIGH);
    L2Stat=0;
  }
  // process received value
}

BLYNK_WRITE(V14)
{
  int pin5Value = param.asInt(); // assigning incoming value from pin V1 to a variable
  Rly5=pin5Value;
  if (pin5Value==1){
    MODE=1;
  //  Serial.println("SECURITY ACTIVATED..");
  // Blynk.logEvent("secon", String("SECURITY MODE ACTIVATED"));
    Blynk.logEvent("secon");
    SECURITY=1;
  }
  if (pin5Value==0){
    MODE=0;
    //Blynk.logEvent("secoff", String("SECURITY MODE = OFF"));
    Blynk.logEvent("secoff");
  //  Serial.println("SECURITY OFF..");
    SECURITY=0;
    ALM1=0;
    ALM2=0;
    ALM3=0;

  }
  // process received value
}
```

```
//--------------------------------------------------------------------------

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
{
  // Change Web Link Button message to "Congratulations!"
 //              Blynk.setProperty(V3,              "offImageUrl",              "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
 //            Blynk.setProperty(V3,            "onImageUrl",              "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
 // Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-
need-to-blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{
   //---------------------------------------------------------------
static unsigned long timepoint = millis();


  if (millis() - timepoint > 1000U) //time interval: 1s
  {


if (F1Stat==1){
  TDLY++;
  Serial.println(TDLY);
  if (TDLY>4){
    digitalWrite(SOLENOID,LOW);
    DOORSTATUS="LOCKED";
    Blynk.virtualWrite(V0,DOORSTATUS);
    F1Stat=0;
    TDLY=0;
  }
}




  //-------------------------------------------------------------

      Serial.println(COUNTER);

delay(100);


Blynk.virtualWrite(V1,DOORSENSOR);
Blynk.virtualWrite(V2,DOORCond);
```

```
  }


  //-------------------------------------------------------------
}

void setup()
{

  int i,k;
  pinMode(BZ,OUTPUT);
  pinMode(DOOR,INPUT);
  pinMode(IR2,INPUT);
  pinMode(SOLENOID, OUTPUT);
  pinMode(Fan2, OUTPUT);
  pinMode(Light1, OUTPUT);
  pinMode(Light2, OUTPUT);

  digitalWrite(SOLENOID,LOW);
  digitalWrite(Light1,HIGH);
  digitalWrite(Light2,HIGH);
  digitalWrite(Fan2,HIGH);







  delay(3000);

  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  // Setup a function to be called every second
  timer.setInterval(1000L, myTimerEvent);
}

void loop()
{



//----------------------------CHECK  SENSOR-----------------------------------
if (digitalRead(DOOR)==0){
```

```
  DOORSENSOR=0;
 DOORCond="CLOSE";
 Serial.println("CLOSE");

}
if (digitalRead(DOOR)==1){
 DOORSENSOR=1;
  DOORCond="OPEN";
  Serial.println("OPEN");


}




//---------------------------------------------------------------------

  Blynk.run();
  timer.run();

}
```

APPENDIX C- PROJECT MANUAL/PRODUCT CATALOGUE