

# **ADVANCED SAFE LOCKER CONTROLLING SYSTEM BY USING APPLICATION AND IOT**

A  
MINI PROJECT REPORT

Submitted in the partial Fulfillment of the requirements for the award of the  
Degree of

**BACHELOR OF TECHNOLOGY  
IN  
ELECTRONICS AND COMMUNICATION ENGINEERING**

**Submitted By**

- |   |                   |
|---|-------------------|
| <b>1. Jaddu Venkata Rama Subhash</b>    | <b>19R21A04E1</b> |
| <b>2. M Umesh Chandra</b>               | <b>19R21A04F1</b> |
| <b>3. Parisi Chinni Sri Venkata Sai</b> | <b>19R21A04G0</b> |
| <b>4. Potharla Sai Abhinav</b>          | <b>19R21A04G5</b> |

**UNDER THE GUIDANCE OF**

**Ms.Sailaja**  
**Associate professor**



**MLR** Institute of Technology

(Autonomous)

(Affiliated to JNTUH, Hyderabad)

Dundigal, Hyderabad-500043

**2019-2023**

(Autonomous)

(Affiliated to JNTUH, Hyderabad)

Dundigal, Hyderabad-500043



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

## ***CERTIFICATE***

This is to certify that the project *entitled “Advanced safe locker controlling system by using application and Iot”* is the bonafied work done by *Jaddu Venkata Rama Subhash (19R21A04E1), M Umesh Chandra(19R21A04F1), Parisi Chinni Sri Venkata Sai(19R21A04G0), Potharla Sai Abhinav (19R21A04G5)* in partial fulfillment of the requirement for the award of the degree of B.Tech in Electronics and Communication Engineering, during the academic year 2022-23.

**Internal Guide**

**Head of the Department**

**External Examiner**

# ACKNOWLEDGEMENT

We express our profound thanks to the management of **MLR Institute of Technology**, Dundigal, Hyderabad, for supporting us to complete this project.

We take immense pleasure in expressing our sincere thanks to **Dr.K.Srinivasa Rao**, Principal, MLR Institute of Technology, for his kind support and encouragement.

We are very much grateful to **Dr S.V.S Prasad**, Professor & Head of the Department, MLR Institute of Technology, for encouraging us with his valuable suggestions.

We are very much grateful to Ms. Sailaja, Associate professor for his unflinching cooperation throughout the project.

We would like to express our sincere thanks to the teaching and non teaching faculty members of ECE Dept., MLR Institute of Technology, who extended their help to us in making our project work successful.

## Project associates:

Jaddu Venkata Rama Subhash	(19R21A04E1)
M Umesh Chandra	(19R21A04F1)
Parisi Chinni Sri Venkata Sai	(19R21A04G0)
Potharla Sai Abhinav	(19R21A04G5)

## **ABSTRACT**

A safe home is of paramount importance in today's world. Even after using heavy metal lockers which are hard to open, there are many reasons for you to worry, such as lost or stolen keys. Everyone prefers to protect their home, office, facilities, etc. Today, many new technologies have been born to overcome the shortcomings of traditional iron cabinets. These alternatives not only help keep your home secure, but also allow remote access within the specified ranges to your locker.

The Internet of Things is one of the technologies that makes our daily lives easier by providing solutions to many of these problems. Our idea is concerned with the keyboard used in this project, it looks like a normal keyboard, but its intake numbers or characters are different from the regular one. We can change keypad functionality based on user requirements. Users can unlock lockers by installing Android apps developed for devices, such as tablets, smartphones by providing login security patch (username and password). Application authenticated against a database on the cloud. The keyboard functions are set to normal in the app initially, then the locker can be unlocked using general keypad functionality. To an authorized user has provided three chances to unlock the locker if the user enters wrong password. If user exceeded three chances system gets locked for some period of time.

## **TABLE OF CONTENTS**

<b>ABSTRACT</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>vii to ix</b>
<b>LIST OF TABLES</b>	<b>ix</b>

### **Page No.**

<b>CHAPTER 1: Introduction.....</b>	<b>1-6</b>
-------------------------------------	------------

<b>1.1 Summary</b>	
<b>1.2 Introduction to IOT</b>	
<b>1.2.1 Introduction</b>	
<b>1.2.2 Main components used in IOT</b>	
<b>1.2.3 IOT Enablers</b>	
<b>1.2.4 Working with IoT Devices</b>	
<b>1.2.5 Applications of IOT</b>	
<b>1.2.6 Advantages of IOT</b>	
<b>1.2.7 Disadvantages of IOT</b>	

<b>CHAPTER 2: LITERATURE SURVEY.....</b>	<b>7-8</b>
--	------------

<b>CHAPTER 3: METHODOLOGY.....</b>	<b>9-44</b>
------------------------------------	-------------

<b>3.1 Proposed System</b>	
<b>3.2 Hardware components</b>	
<b>3.2.1 NodeMCU Microcontroller (ESP8266)</b>	

3.2.2 Relay module	
3.2.3 Jumper wires	
3.2.4 Membrane keypad	
3.2.5 Solenoid Lock	
3.3 Software with Cloud Design	
3.3.1 Aurdino IDE	
3.3.2 Firebase	
3.3.3 MIT App Inventor	
3.3.4 MIT AI2 Companion	
3.4 Block Diagram	
3.5 Working	
3.5.1 Application on Software	
3.5.2 Hardware Working	
3.6 Flow Chart	
 CHAPTER 4: Performance Analysis.....	 45-56
4.1 Results	
4.2 Hardware coding	
 CHAPTER 5: Conclusion and Future Scope.....	 57-59
5.1 Advantages	
5.2 Disadvantages	
5.3 Application	
5.4 Future scope	
5.5 Conclusion	

## LIST OF FIGURES

<b>S.NO</b>	<b>FIG.NO:</b>	<b>DESCRIPTION</b>	<b>PAGE NO:</b>
1.	3.1	NodeMCU module	11
2.	3.2	NodeMCU pin description	11
3.	3.3	Relay module	13
4.	3.4	Parts of Relay Module	14
5.	3.5	parts of single channel relay module	15
6.	3.6	Relay Internal Working	16
7.	3.7	Different types of jumper wires	17
8.	3.8	Membrane Keypad rows and columns pin ports	18
9.	3.9	Membrane keypad Internal transistor setup	19
10	3.10	Internal layers used in membrane keypad construction	20
11	3.11	Rear Adhesive layer	21
12	3.12	Magnetic field created by the power supply	23
13	3.13	DC-C Frame Solenoid	24
14	3.14	DC-D Frame Solenoid	25
15	3.15	Linear Solenoid	25

16	3.16	Solenoid valve	26
17	3.17	Solenoidal lock	27
18	3.18	Aurdino ide application on pc	28
19	3.19	Google firebase logo	30
20	3.20	MIT Website welcome page	31
21	3.21	Android app logo	32
22	3.22	AI companion on webpage	33
23	3.23	Ways to connect with APP inventor	34
24	3.24	Representation	34
25	3.25	Login screen on	35
26	3.26	New user screen on application	36
27	3.27	Forgot password screen on Application	37
28	3.28	OTP screen on mobile application	41
29	3.29	keypad replica on application	38
30	3.30	keypad in remote access screen	39
31	3.31	menu bar on application	40
32	3.32	Control screen on Application	41
33	3.33	Reset password screen on application	42
34	3.34	Number swap screen on application	43
35	3.35	Relay screen on application	44
36	4.1	serial monitor output	45



37	4.2	cloud test value updated to “1”	45
38	4.3	serial monitor output	46
39	4.4	cloud test value updated to “0”	46
40	4.5	initial locker key on cloud	47
41	4.6	updated locker key on cloud	47
42	4.7	serial monitor output when safe opens with keypad	48
43	4.8	serial monitor message	48
44	4.9	when relay control was disabled	49
45	4.10	Remote access modification	49

## **LIST OF TABLES**

<b>S.NO</b>	<b>TABLE NO:</b>	<b>DESCRIPTION</b>	<b>PAGE NO:</b>
1	3.1	Node MCU pin description	12
2	3.2	Single-Channel Relay Module Pin Description	13
3	3.3	Keypad control based on row and column pins	20

among all the values

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1. Introduction:**

The purpose of this project is to design a system that provides high security to the lockers, homes, offices and even bank vaults. We all know that in this generation everything was connected to Internet and can be accessed by anywhere if we have an internet connectivity. Security is a primary concern for every individual where humans cannot find ways to provide security to their confidential belongings manually. Instead, humans find an alternative solution that provides reliable and automated security. In this system we are using the membrane keypad of the devices was communicate with the mobile application through the cloud networking. Keypad will change its functionality based on the layout arrangement on mobile application. User can also control the power supply to the relay gate from ESP3622. We are using solenoidal lock in this project that works in linear motion. If power supply is on for the lock it unlocks, if not it will not open.

If a person tries to open the safe a notification was sent to the user through the application. When a person enters the four-digit pin to unlock the system the safe will cross check the pin entered on firebase. Firebase will check the pin entered on user database buckets. If he enters the correct password, then the safe unlocks. If user exceeds three attempts device locks for some time for the safety purposes.

### **1.2. INTERNET OF THINGS(IOT):**

#### **1.2.1 Introduction:**

The Internet of Things (IoT) is the networking of physical objects with electronics built into their architecture to enable communication and the detection of interactions between them or with the surrounding environment. IoT-based technology will provide higher levels of services in the coming years, effectively altering how people go about their daily lives. Just a few categories where IoT is well established include improvements in medicine, power, gene therapies, agriculture, smart cities, and smart homes.

IoT is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.

### 1.2.2 Main components used in IoT:

**Low-power embedded systems:** Less battery consumption, high performance are the inverse factors that play a significant role during the design of electronic systems.

**Sensors:** Sensors are a major part of any IoT applications. It is a physical device that measures and detect certain physical quantity and convert it into signal which can be provide as an input to processing or control unit for analysis purpose.

Different types of Sensors:

- Temperature Sensors.
- Image Sensors.
- Gyro Sensors.
- Obstacle Sensors.
- RF Sensor.
- IR Sensor.
- MQ-02/05 Gas Sensor.
- LDR Sensor.
- Ultrasonic Distance Sensor.

### 1.2.3 IoT Enablers:

- RFIDs: uses radio waves in order to electronically track the tags attached to each physical object.
- Sensors: devices that can detect changes in an environment.

- Nanotechnology: as the name suggests, these are extremely small devices with dimensions usually less than a hundred nanometers.

#### **1.2.4 Working with IoT Devices:**

- **Collect\_and Transmit Data:** For this purpose, sensors are widely used they are used as per requirements in different application areas.
- **Actuate device based on triggers produced by sensors or processing devices:** If certain condition is satisfied or according to user's requirements if certain trigger is activated then which action to performed that is shown by Actuator devices.
- **Receive Information:** From network devices user or device can take certain information also for their analysis and processing purposes.
- **Communication Assistance:** Communication assistance is the phenomena of communication between 2 network or communication between 2 or more IoT devices of same or different Networks.
- This can be achieved by different communication protocols like: MQTT, Constrained Application Protocol, ZigBee, FTP, HTTP etc.

#### **1.2.5 Applications of IOT:**

##### **Smart Homes:**

Smart home applications with the use of smart sensors are becoming popular now. Any smart device can be configured and connected to the internet and control using simple mobile application.

##### **Self-driven Cars:**

We have seen a lot about self-driven cars. Google tried it out, Tesla tested it, and even Uber came up with a version of self-driven cars that it later shelved. Since it is human lives on the

roads that we're dealing with, we need to ensure the technology has all that it takes to ensure better safety for the passenger and those on the roads.

The cars use several sensors and embedded systems connected to the Cloud and the internet to keep generating data and sending them to the Cloud for informed decision-making through Machine Learning. Though it will take a few more years for the technology to evolve completely and for countries to amend laws and policies, what we are witnessing right now is one of the best applications of IoT.

### **IoT Retail Shops:**

If you haven't already seen the video of Amazon, Go – the concept store from the eCommerce giant, you should check it out right away. Perhaps this is the best use of the technology in bridging the gap between an online store and a retail store. The retail store allows you to go cashless by deducting money from your Amazon wallet. It also adds items to your cart in real-time when you pick products from the shelves.

If you change your mind and pick up another article, the previous one gets deleted and replaces in your cart with the new item. The best part of the concept store is that there is no cashier to bill your products. You don't have to stand in line but just step out after you pick up your products from shelves. If this technology is effective enough to fetch more patronage, this is sure to become a norm in the coming years.

### **Farming:**

Farming is one sector that will benefit the most from the Internet of Things. With so many developments happening on tools farmers can use for agriculture, the future is sure promising. Tools are being developed for Drip Irrigation, understanding crop patterns, Water Distribution, drones for Farm Surveillance, and more. These will allow farmers to come up with a more productive yield and take care of the concerns better.

**Wearables:**

Wearables remain a hot topic in the market, even today. These devices serve a wide range of purposes ranging from medical, and wellness to fitness. Of all the IoT startups, Jawbone, a wearables maker, is second to none in terms of funding.

**Smart Grids:**

One of the many useful IoT examples, a smart grid, is a holistic solution that applies an extensive range of Information Technology resources that enable existing and new gridlines to reduce electricity waste and cost. A future smart grid improves the efficiency, reliability, and economics of electricity.

**Industrial Internet:**

The Industrial Internet of Things consists of interconnected sensors, instruments, and other devices connected with computers' industrial applications like manufacturing, energy management, etc. While still being unpopular in comparison to IoT wearables and other uses, market researchers like Gartner, Cisco, etc., believe the industrial internet to have the highest overall potential.

**Telehealth:**

Telehealth, or Telemedicine, hasn't completely flourished yet. Nonetheless, it has great future potential. IoT Examples of Telemedicine include the digital communication of Medical Imaging, Remote Medical Diagnosis & Evaluations, Video Consultations with Specialists, etc.

**Smart Supply-chain Management:**

Supply chains have stuck around in the market for a while now. A common example can be Solutions for tracking goods while they are on the road. Backed with IoT technology, they are sure to stay in the market for the long run.

**Traffic Management:**

Car traffic management in large cities can be greatly improved with the help of the Internet of Things (IoT). The Internet of Things helps us stay informed and improves traffic monitoring by allowing us to use our mobile phones as sensors to collect and share data from our vehicles

through apps like Waze or Google Maps. This feeds and improves the data on the various routes to the same destination, distance, and estimated arrival time.

Analysis of traffic patterns over a long period is another **IoT application**. It provides an idea of what might happen during peak hours. Commuters will be better prepared to avoid traffic and delays by being made aware of possible alternatives.

### **Water and waste management:**

Many cities are adopting water recycling using water treatment units. Using an **IoT application**, you can see how much wastewater is being produced, how much is being consumed in a specific area, and how waste production is changing over time.

We can effectively deal with this problem using **Internet of Things applications** and smart sensor technology. With a smart waste management system, authorities will be able to predict how much waste will be generated in a specific location, how to properly process it, when to clear it, and how to analyze data for future planning, among other things.

#### **1.2.6 Advantages of IOT:**

- Improved productivity of staff and reduced human labor.
- Efficient operation management
- Better use of resources and assets
- Cost-effective operation
- Improved work safety
- Thorough marketing and business development
- Improved customer service and retention
- Better business opportunities
- More trustworthy image of the company

#### **1.2.7 Disadvantages of IOT:**

- Security flaws.
- Associated costs.
- Power supply dependence.
- Network dependence.



## **CHAPTER 2**

### **LITERATURE SURVEY**

**Journal Title:**

IoT Enhanced Smart Locking System Shanthini Ma, Vidya Gb, Arun Rc

Institute of Electrical and Electronics Engineers (IEEE)

ISSN: 978-1-7281-5821-1/20

The main objective of this paper is to enhance the security of the locking system.

The hardware requirements used in the journal are

1. Servo motor.
2. Piezo buzzer.
3. Arduino UNO.
4. Bluetooth module.

The mobile device will be sending a signal via Bluetooth to the Arduino circuit that acts as a connection between the smartphone and the servo motor after proper authentication is provided using the database. The use of Bluetooth on smartphones is to provide ease of access with better security than the conventional key.

**Journal Title:**

Digital Keypad Security System Based on Arduino with GSM Module, Alarm, and Temperature Sensor Jishant Singha, Rashmi Rogesb, Sandeep Sharmac, Anuradha Bhasind, Risheek Kumar\*e, Jatin Gaurf

Social Science Research Network (SSRN)

ISSN: 978-1-7281-5821-1/20

The Digital Keypad Security System on the Arduino chipset using two different systems; Keypad, and Global System for Mobile Communication.

These two arrangements function on a 4-figure passkey combination which is entered through the keypad. The Global System for Mobile Communication module is key in stabilizing a link between the mobile phone and the Arduino chipset. After this, the passkey is entered by the end user to access the system.

The Global System for Mobile Communication module is responsible for sending text messages in case of any wrong passkey attempts and uses an alarm as a warning of unauthorized intrusion.

### **Journal Title:**

Digital Smart Door Lock Security Using Arduino Uno Microcontroller ATIF AFROZ

Iconic research and engineering (IRE) Journals

ISSN:2456-8880

In this research paper, the design of a simple digital door lock is demonstrated considering components (Arduino uno, Buzzer, Servomotor, membrane keypad) and their different functions and importance. The whole system is operated by a pin. When a user enters a PIN into the security system installed at any entrance, the system captures the PIN and compares it with the stored PINs for a match. If the captured PIN matches with any of the stored PINs, access granted is displayed on the LCD and the door opens; otherwise, access denied is displayed on the LCD and the door remains closed. For change of PIN, the user presses # button then the system asks for current PIN and matches it with stored PINs for a match. If a match was found, the user will be asked to enter the new PIN twice, else the user is asked for the current PIN again.

### **EXISTING SYSTEM:**

As name indicates the safes are used to store the extremely sensitive information or to store the precious elements in it. In general cases, most people use locks to store their land documents, gold ornaments, etc. In market most of the safe locks are costs around 40,000.rs-80,000.rs per safe locker based on safety level. Some updated safes have app control but only through the Bluetooth or Wi-Fi connection. These are not the permanent solutions if the person was away from the safe. These safe locks contain fixed password to unlock, if someone has stolen the password, they can easily access the lock. Thereby safety is at risk.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 PROPOSED SYSTEM:**

General digital keyboard is used in this project, but its functions are controlled through the App. Users unlock lockers by installing Android app developed, on devices such as tablets, smartphones, and laptops by providing login information such as username and password authenticated against a database on the Internet. Initially the keypad functions are set to normal in the app. The user can swap the positions of the numbers and same reflects on the safe digital keypad. Authorized user is provided three chances to unlock the locker if the user enters wrong password. System got locked in some period of time when user exceeds the number of chances to unlock. Application also provides a notification when safe gets unlocked and stores it on database

#### **3.2. Hardware components**

1. NodeMCU Microcontroller (ESP8266)
2. Relay module
3. Membrane Keypad
4. Solenoidal lock
5. Jumper wires

##### **3.2.1 NodeMCU Microcontroller (ESP8266)**

Open-source prototype board designs are available for the NodeMCU open-source firmware. NodeMCU is a combination of "node" and "MCU" (micro-controller unit). The related development kits are not technically considered to be "Nodemcu" devices; rather, the moniker refers to the firmware.

Prototyping board designs and firmware are both open sources.

The Lua programming language is employed by the firmware. The firmware was created using the Espressif Non-OS SDK for ESP8266 and is based on the eLua project. It makes extensive use of open-source initiatives like SPIFFS and lua-cjson. Users must choose the

components necessary for their project and create a firmware specific to their requirements due to resource limitations. Additionally, support for the 32-bit ESP32 has been included.

A circuit board acting as a dual in-line package (DIP) that incorporates a USB controller with a smaller surface-mounted board holding the MCU and antenna is the prototyping hardware that is frequently utilized. The DIP format's selection makes breadboard prototyping simple. The ESP-12 module of the ESP8266, which is a Wi-Fi SoC combined with a Tensilica Xtensa LX106 core and is extensively utilized in IoT applications, served as the design's basic foundation.

There are two versions of NodeMCU that are currently available: ESP-12 in version 0.9 and ESP-12E in version 1.0, where E stands for "Enhanced."

The ESP8266 was released shortly after NodeMCU was developed. The ESP8266 started being produced by Espressif Systems on December 30th, 2013. On October 13, 2014, Hong uploaded the first file of the nodemcu-firmware to GitHub, launching NodeMCU. Two months later, when developer Huang R committed the gerber file for an ESP8266 board called devkit v0.9, the project was expanded to incorporate an open-hardware platform. Later that month, Tuan PM contributed to the NodeMCU project and converted the Contiki MQTT client library to the ESP8266 SoC platform.

As a result, NodeMCU was able to implement the MQTT IoT protocol by utilizing Lua to connect to the MQTT broker. Another significant development occurred on January 30, 2015, when Devsaurus transferred the u8glib to the NodeMCU project. This allowed NodeMCU to easily operate LCD, Screen, and other devices.

In order to make it relatively simple to change the Arduino IDE to support alternative toolchains so that Arduino C/C++ could be compiled for these new processors, Arduino.cc had to modify the Arduino IDE as they started creating new MCU boards based on non-AVR processors, such as the ARM/SAM MCU used in the Arduino Due.

They accomplished this by introducing the SAM Core and the Board Manager. The group of software parts needed by the Board Manager and the Arduino IDE to translate an Arduino C/C++ source file into the target MCU's machine language is known as a "core." The "ESP8266 Core for the Arduino IDE" was created by some ESP8266 enthusiasts as an Arduino core for the ESP8266 Wi-Fi SoC. This has emerged as a top platform for software development including NodeMCUs, boards.

Pins on GPIO The 17 GPIO pins on the NodeMCU/ESP8266 can be dynamically allocated to various I2C, I2S, UART, PWM, IR remote control, LED light, and button operations. Each GPIO with digital capability can be adjusted to high impedance, internal pull-up, or internal pull-down.

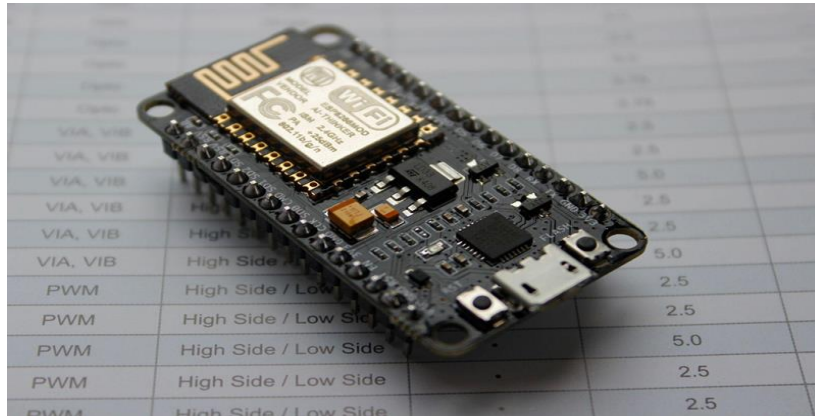


Fig 3.1: NODEMCU module

### NodeMCU Development Board Pinout Configuration

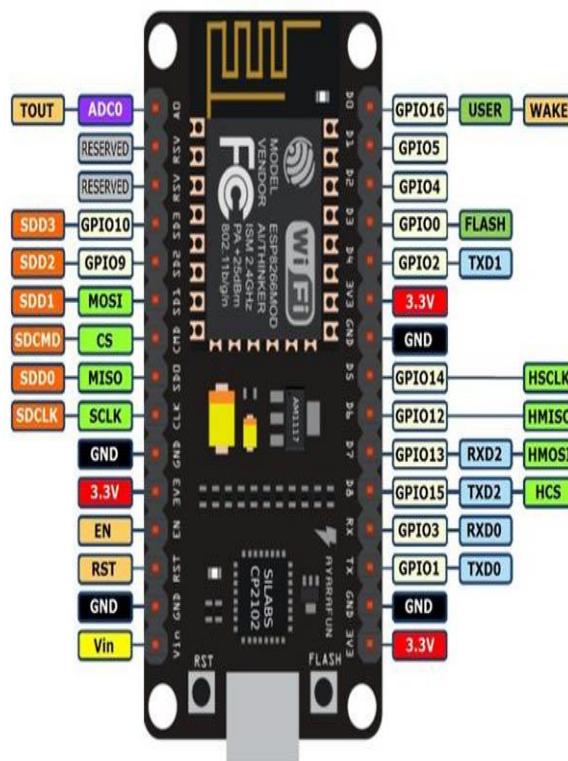


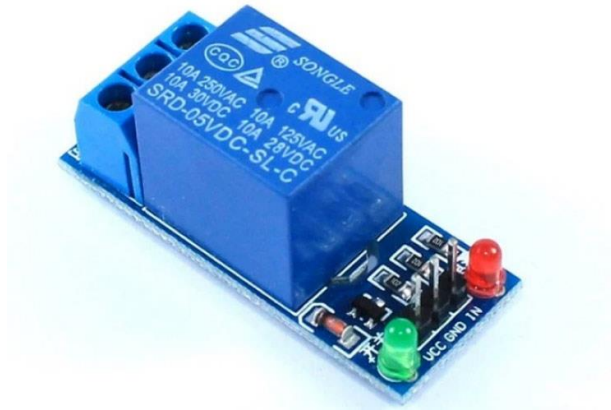
Fig 3.2: NODEMCU pin diagram.

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	<p><b>Micro-USB:</b> NodeMCU can be powered through the USB port</p> <p><b>3.3V:</b> Regulated 3.3V can be supplied to this pin to power the board</p> <p><b>GND:</b> Ground pins</p> <p><b>Vin:</b> External Power Supply</p>
Control Pins	<b>EN, RST</b>	The pin and the button reset the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

*Table:3.1 Pin description of node mcu*

### 3.2.2 Relay module

Relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises of components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not.

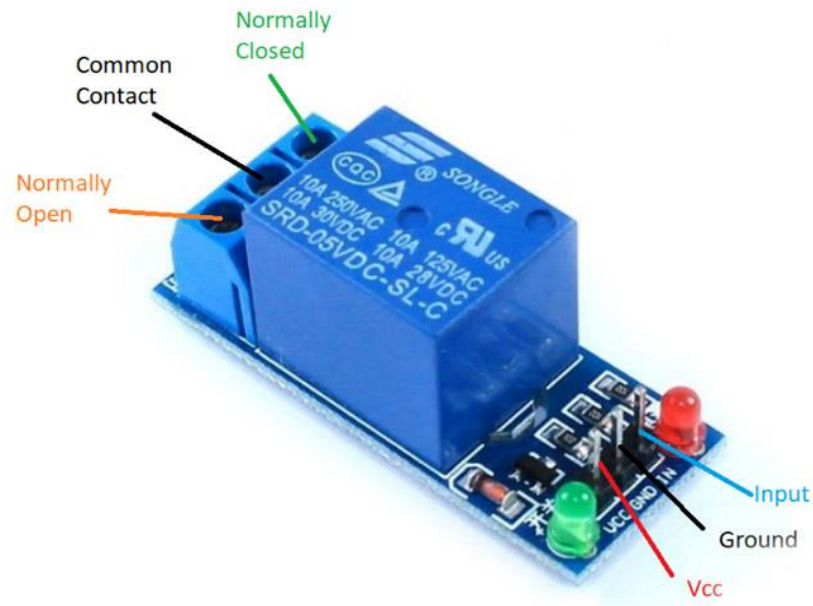


*Fig3.3: Relay module.*

#### Single-Channel Relay Module Pin Description

Pin Number	Pin Name	Description
1	Relay Trigger	Input to activate the relay
2	Ground	0V reference
3	VCC	Supply input for powering the relay coil
4	Normally Open	Normally open terminal of the relay
5	Common	Common terminal of the relay
6	Normally Closed	Normally closed contact of the relay

*Table 3.2 pin Description of Relay Module*



*Fig3.4: parts of relay module.*

### Single-Channel Relay Module Specifications

Supply voltage – 3.75V to 6V

Quiescent current: 2mA

Current when the relay is active: ~70mA

Relay maximum contact voltage – 250VAC or 30VDC

Relay maximum current – 10A

### Understanding 5V Single-Channel Relay Module

The single-channel relay module is much more than just a plain relay, it contains components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active.

First is the screw terminal block. This is the part of the module that is in contact with mains so a reliable connection is needed. Adding screw terminals makes it easier to connect thick mains cables, which might be difficult to solder directly. The three connections on the terminal block are connected to the normally open, normally closed, and common terminals of the relay.



The second is the relay itself, which, in this case, is a blue plastic case. Lots of information can be gleaned from the markings on the relay itself. The part number of the relay on the bottom says “05VDC”, which means that the relay coil is activated at 5V minimum – any voltage lower than this will not be able to reliably close the contacts of the relay.

There are also voltage and current markings, which represent the maximum voltage and current, the relay can switch. For example, the top left marking says “10A 250VAC”, which means the relay can switch a maximum load of 10A when connected to a 250V mains circuit. The bottom left rating says “10A 30VDC”, meaning the relay can switch a maximum current of 10A DC before the contacts get damaged.

The 'relay status LED' turns on whenever the relay is active and provides an indication of current flowing through the relay coil.

The input jumper is used to supply power to the relay coil and LEDs. The jumper also has the input pin, which when pulled high activates the relay.

The switching transistor takes an input that cannot supply enough current to directly drive the relay coil and amplifies it using the supply voltage to drive the relay coil. This way, the input can be driven from a microcontroller or sensor output. The freewheeling diode prevents voltage spikes when the relay is switched off. The power LED is connected to VCC and turns on whenever the module is powered.

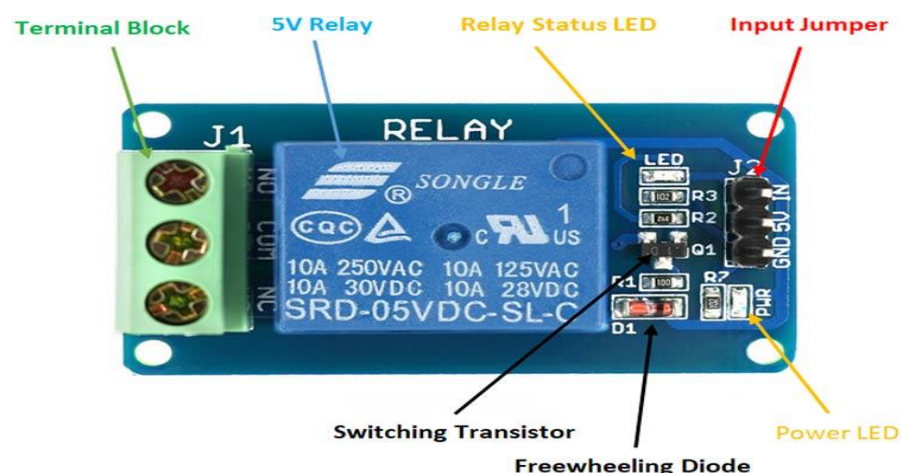
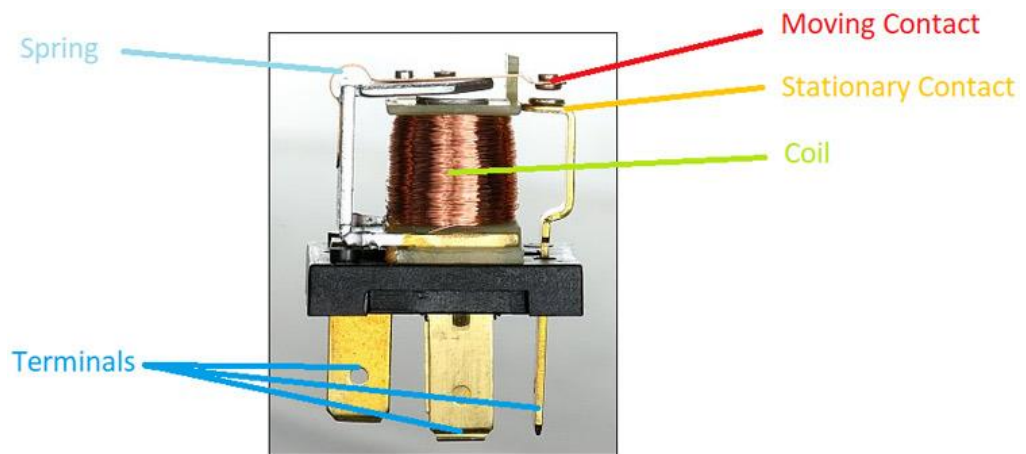


Fig3.5: *parts of single channel relay module.*

## Working of relay



*Fig3.6: Relay Internal Working.*

The relay uses an electric current to open or close the contacts of a switch. This is usually done using the help of a coil that attracts the contacts of a switch and pulls them together when activated, and a spring pushes them apart when the coil is not energized.

There are two advantages of this system – First, the current required to activate the relay is much smaller than the current that relay contacts are capable of switching, and second, the coil and the contacts are galvanically isolated, meaning there is no electrical connection between them. This means that the relay can be used to switch mains current through an isolated low voltage digital system like a microcontroller.

### 3.2.3 Jumper wires

An electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them - simply "tinned") is known as a jump wire (also known as a jumper, jumper wire, or DuPont wire), and it is typically used to connect the parts of a breadboard or other prototype or test circuit, internally or with other machinery or components, without soldering.

A breadboard, a circuit board's header connector, or a piece of test equipment can all be used to create slots for the "end connectors" that hold individual jump wires.

Jumper wires come in a variety of varieties. While some have separate connectors, others have the same kind of electrical connector on both ends. Typical connectors include:

To connect to/with a breadboard or female header connector, use solid tips. Increased mounting density of components and jump wires is possible on a breadboard without worrying about short circuits because of how the pieces are arranged and how simple they are to install. To distinguish between the various operating signals, the jump wires come in a variety of sizes and hues.

Crocodile clips are used for a variety of tasks, including temporarily connecting prototype sensors, buttons, and other parts to equipment or parts with arbitrary connectors, wires, screw terminals, etc.

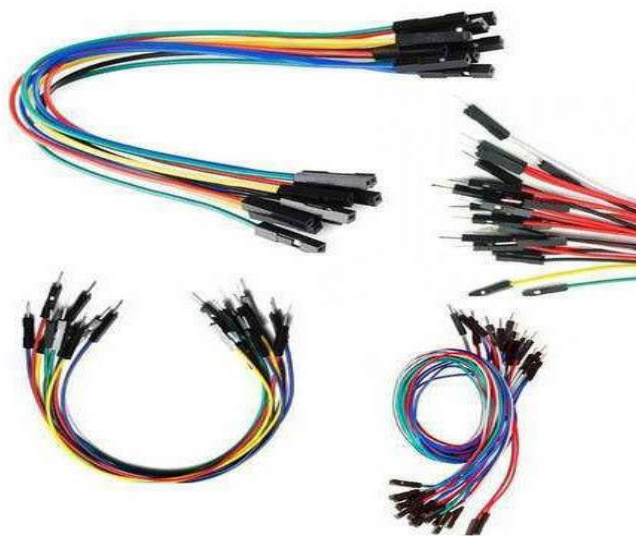
On test equipment, banana connectors are frequently utilized for DC and low-frequency AC signals.

In telephone (RJ11) and computer networking, registered jacks (RJnn) are frequently utilized (RJ45).

For music, low-resolution composite video transmissions, or other low-frequency applications requiring a shielded cable, RCA connectors are frequently employed.

Radio frequency signals are transported between circuits, test apparatus, and antennas via RF connectors.

RF jumper cables are corrugated wires that are smaller, more flexible, and used to link antennas and other components to network wiring. Jumpers are also utilized in base stations to link radio equipment and antennas. The diameter of 1/2" jumper cables are typically the most flexible.



*Fig3.7: Different types of jumper wires.*

### 3.2.4 Membrane keypad

Membrane keypads are a great place to start when adding key input to a project because they are affordable, durable, and water-resistant. Building a number of projects that require human input for menu selection, password entering, or robot action will benefit greatly from learning how to interface them with an Arduino.

The most popular membrane keypad sizes are the 43 keypad (12 keys) and the 44 keypads. Membrane keypads come in a range of sizes (16 keys). They are simple to use for everyone and have a layout resembling a typical telephone keypad.

The input method most frequently used is the keypad or keyboard. Usually applied to devices having microcontroller and processor architectures. This part will cover the "43" matrix keypad, whose fundamental comprehension is crucial. Basic here refers to the device's design and key detecting system's operation.

In this case, R1, R2, R3, and R4 correspond to rows 1, 2, 3, and 4, respectively, and C1, C2, and C3 refer to columns 1, 2, and 3, respectively.

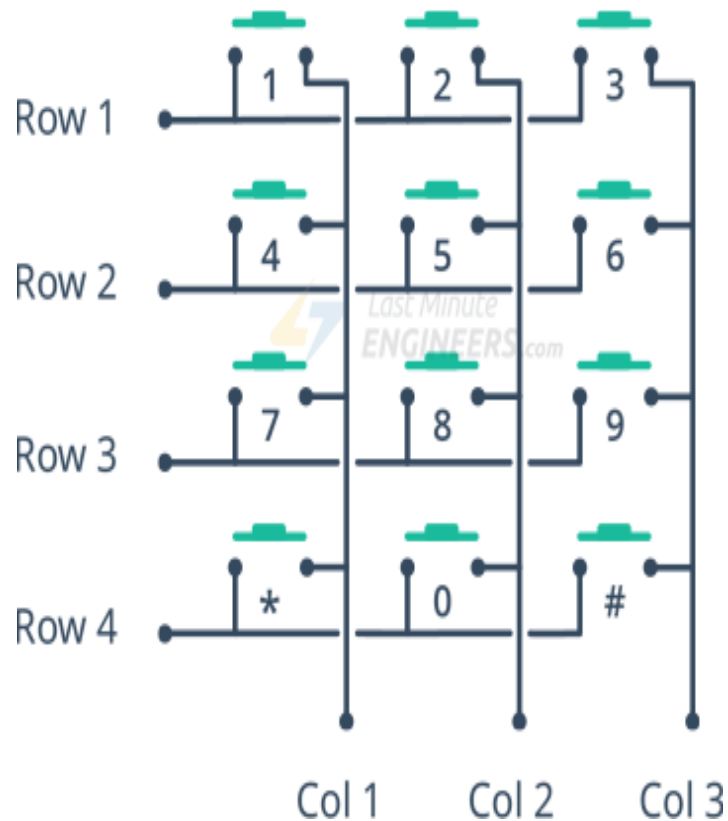


Fig3.8; Membrane Keypad rows and columns pin ports.

### Keypad Working:

The matrix keypad consists of pushbutton contacts that are connected to the row and column lines. There is one pin for each column and one pin for each row. So the  $4 \times 4$  keypad has  $4 + 4 = 8$  pins, while the  $4 \times 3$  keypad has  $4 + 3 = 7$  pins.

This illustration of a basic  $4 \times 3$  keypad arrangement demonstrates how the internal conductors connect the rows and columns.



*Fig3.9: Membrane keypad Internal transistor setup.*

When the button is pressed, one of the rows is connected to one of the columns, allowing current to flow between them. When the key '4' is pressed, for instance, column 1 and row 2 are connected.

By identifying which column and row are connected, we can determine which button has been pressed.

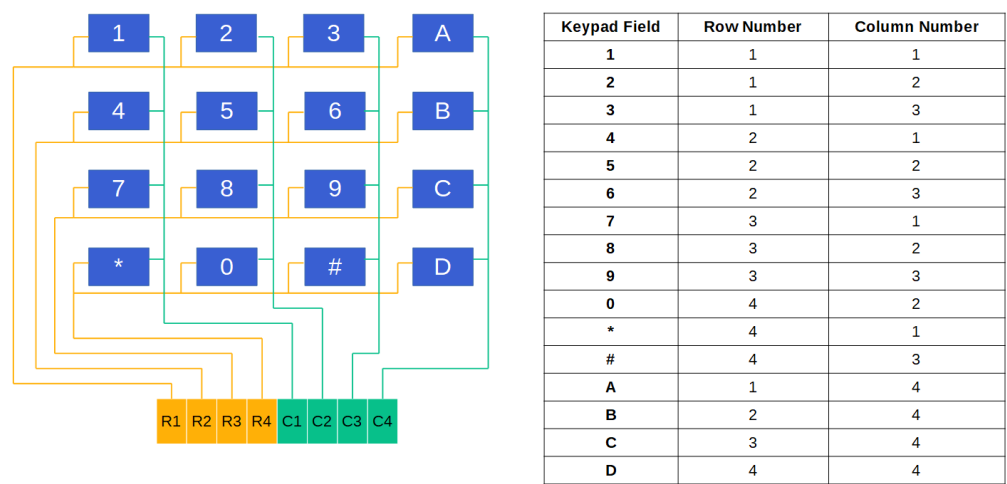


Table 3.3 Keypad control based on row and column pins among all the values.

Membrane Keypad Construction:

Membrane keypads are made of a thin, flexible membrane material and typically have six layers:

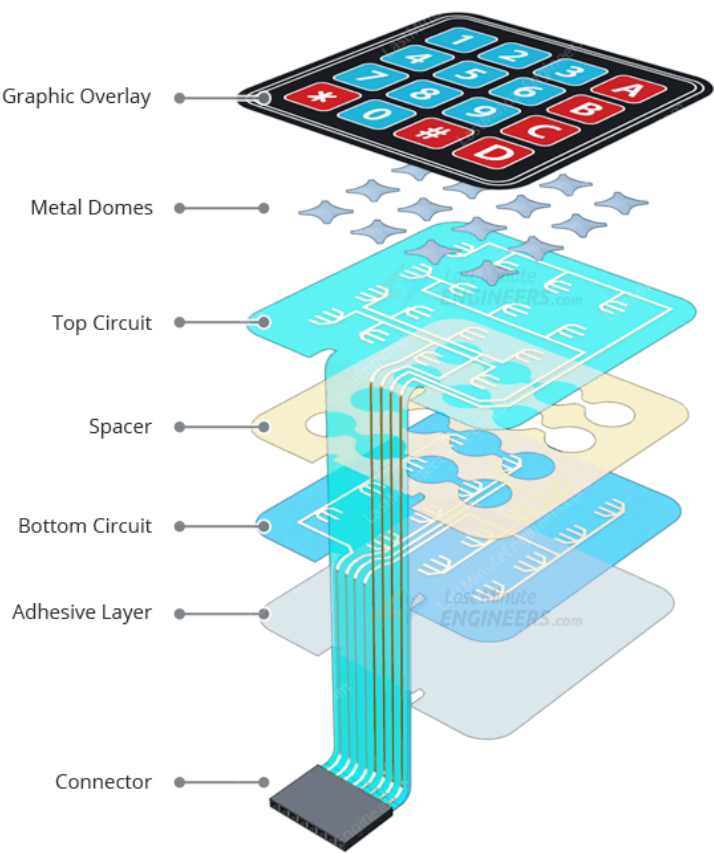


Fig3.10: Internal layers used in membrane keypad construction.

**Graphic Overlay** – Graphic overlays are typically made of polyester because it has better flex life than polycarbonate.

**Metal Domes** – This layer houses metal domes or polydomes that provide tactile feedback.

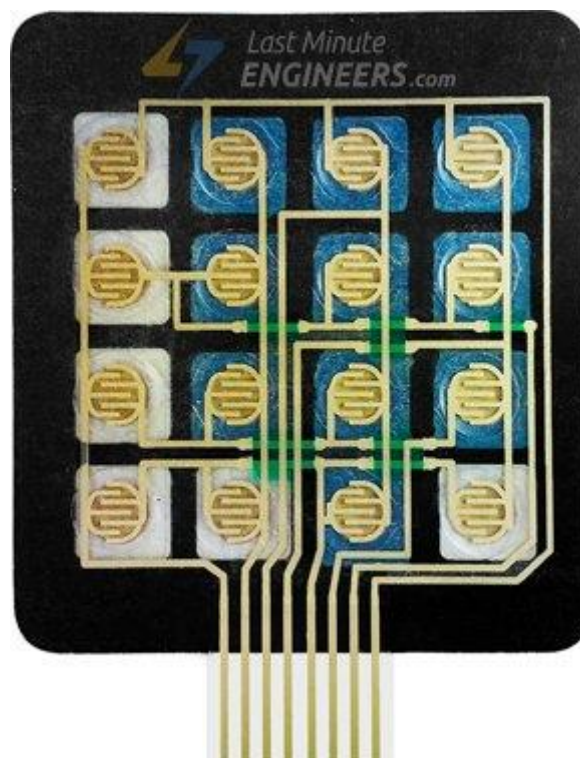
**Top Circuit Layer** – This is typically a polyester printed layer with silver-filled electrically conductive inks. This layer terminates as a flexible tail that connects to the outside world.

**Spacer** – This layer separates the top and bottom circuits, allowing the switch to remain normally open until the keypad is pressed.

**Bottom Circuit Layer** – This is typically a polyester printed layer with silver-filled electrically conductive inks. This layer also terminates as a flexible tail.

**Rear Adhesive Layer** – This layer sticks the keypad to almost anything, which is convenient.

If you peel the paper backing off the keypad, you can see how it is made.



*Fig3.11: Rear Adhesive layer.*



In order to reduce the number of I/O connections, as you can see, all rows and columns are wired together. If this were not the case, interfacing 16 individual pushbuttons, for example, would require 17 I/O pins, one for each pushbutton and one for a common ground. By connecting rows and columns, only 8 pins are required to control the entire 4×4 keypad. This technique of controlling a large number of inputs using fewer pins is known as Multiplexing.

### 3.2.5 Solenoid Lock

Magnetic locks specify latches for electrical locking and unlocking. Power-on unlock and power-on lock-and-hold are available, which can be used depending on the situation. The power-on unlock type can unlock only while the solenoid is on. This type of door is locked and will not open in the event of a power failure or cable disconnection. Provides excellent security. This type is mainly used where security is required. The power-on lock type can lock the door while the solenoid is ON. When the power is off, the door will be unlocked. This type unlocks the door when the wire is disconnected due to a fire or accident, and is used as an emergency exit during firefighting activities and evacuation rather than as a crime prevention measure. The holding type performs two operations, locking and unlocking, by applying a positive or negative pulse voltage to the solenoid, and holds the de-energized state at any position. It is characterized by energy savings, since the electromagnet does not have to be on all the time. Continuous power supply and intermittent power supply are classified into continuous power supply designed to supply rated voltage continuously for a long time without exceeding a certain temperature rise limit, and continuous power supply designed to supply constant voltage for a specified period of time. There is an intermittent power supply. without exceeding the specified temperature rise limit.

### Solenoid

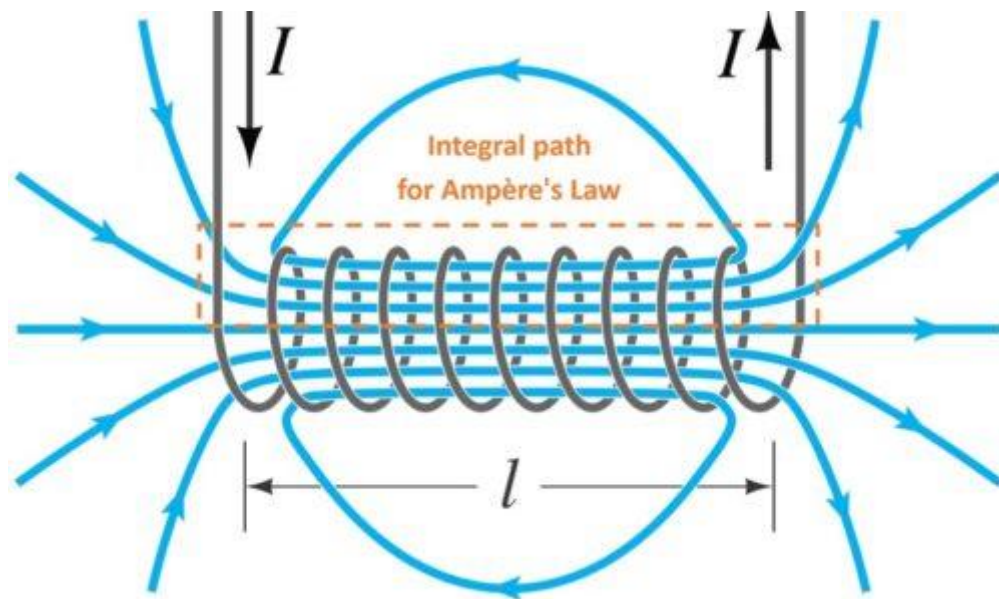
An electric current flowing through a coiled wire creates a magnetic field. When this wire is wound around a ferromagnetic or ferrimagnetic material, a magnet is created known as an electromagnet. Since a magnetic field is produced as long as the current exists in the wire, the subsequent electromagnet has a temporary magnetic effect. When the current decreases to zero, there will be no magnetic effect.

Derived from two Greek words: **Solen (pipe)** and **Eidos (coil)**, the solenoid is a type of an electromagnetic device that converts electrical energy into mechanical energy. It is generally



made by tightly wounding wires in a helix shape around a piece of metal. Whenever an electric current passes through it, a magnetic field is created.

As previously stated, the power of the magnetic field depends on the electric current. Therefore, by varying the current as per our need, we can easily magnetise and demagnetise the electromagnet, enabling us to control the magnetic fields for different requirements.



*Fig3.12: Magnetic field created by the power supply.*

### Working Principle

A solenoid works on electromagnetism and electromagnetic force. It consists of a round cylindrical coil that has several wire turns and a metal rod inside the coil that is free to move. When an electric current is provided to the coil, a magnetic field is generated due to which the metal core or rod inside the coil gets attracted towards the direction where the magnetic flux is high. This electromagnetic effect in a solenoid enables any connected plunger or armature to move as per our need.

In this way, we can control the magnetic field of a coil by controlling and in turn use it for controlling the mechanical movement of metalcore.

The formula for the magnetic field in a solenoid is:

$$B = \mu IN/L$$

Where,

$B$  = Magnetic field,  $\mu$  = Permeability,  $N$  = number of turns,

$I$  = current of coil,  $L$  = length of coil

Turns density,  $n = N/L$  (Number of turns per unit length)

So, from this formula, we can see that to increase the magnetic force produced in a solenoid coil, we will have to increase the number of turns,  $N$  and the current,  $I$ .

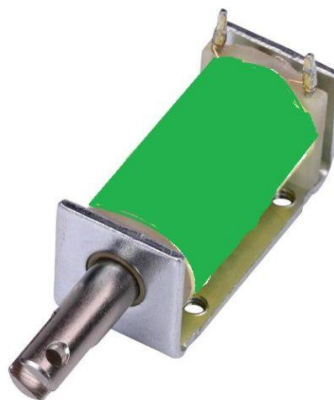
## Types Of Solenoids

### AC Laminated Solenoid

It has a very high initial attracting force and a very short closing time. It is made with laminated metal or insulated thin sheets that are individual and assembled.

### DC-C Frame Solenoid

As its name states, this solenoid is constructed in such a way that it has a letter 'C' like frame cover around the coil. This type is widely used in gaming machines.



*Fig3.13: DC-C Frame Solenoid*

### DC-D Frame Solenoid

As its name says, this solenoid has a coil that is covered by two 'D' frames on two sides. This type is generally used in AC power applications.



*Fig3.14: DC-D Frame Solenoid*

### Linear Solenoid

This type of solenoid has a freely movable steel or iron rod called a plunger inside a round cylindrical shaped coil. The iron rod is allowed to freely move in or out of the cylindrical coil depending on the current applied.



*Fig3.15: Linear Solenoid*

### Rotary Solenoid

It is a special type of solenoid where the magnetic force is converted into a rotational force or a rotary motion. It consists of an armature core mounted on a flat disk.

When a current is provided, the armature gets attracted towards the stator and the flat disk rotates.

## **Applications**

### **Solenoid Valve**

The solenoid valve is a simple device in which a solenoid is used for controlling and regulating the flow of fluid. It has a coil with a free movable plunger or an iron rod with a spring inside it. When we energize the coil, the plunger moves from its position due to magnetic attraction and when we cut the power to the coil, the plunger comes back to its original position with the help of a spring. As soon as the plunger comes in the path of the flowing fluid, its flow stops.



*Fig 3.16: Solenoid Valve.*

### **Solenoid Lock**

Here we use the movement of the solenoid plunger for the locking and unlocking mechanism. These solenoid locks are widely used in electronic and biometric password-based locks. It consists of a strong metal plunger that can move. When the coil gets magnetized due to an electric field, the plunger moves to perform the lock and unlock mechanism.



*Fig 3.17: Solenoidal Lock.*

### **3.3 Software with Cloud Design**

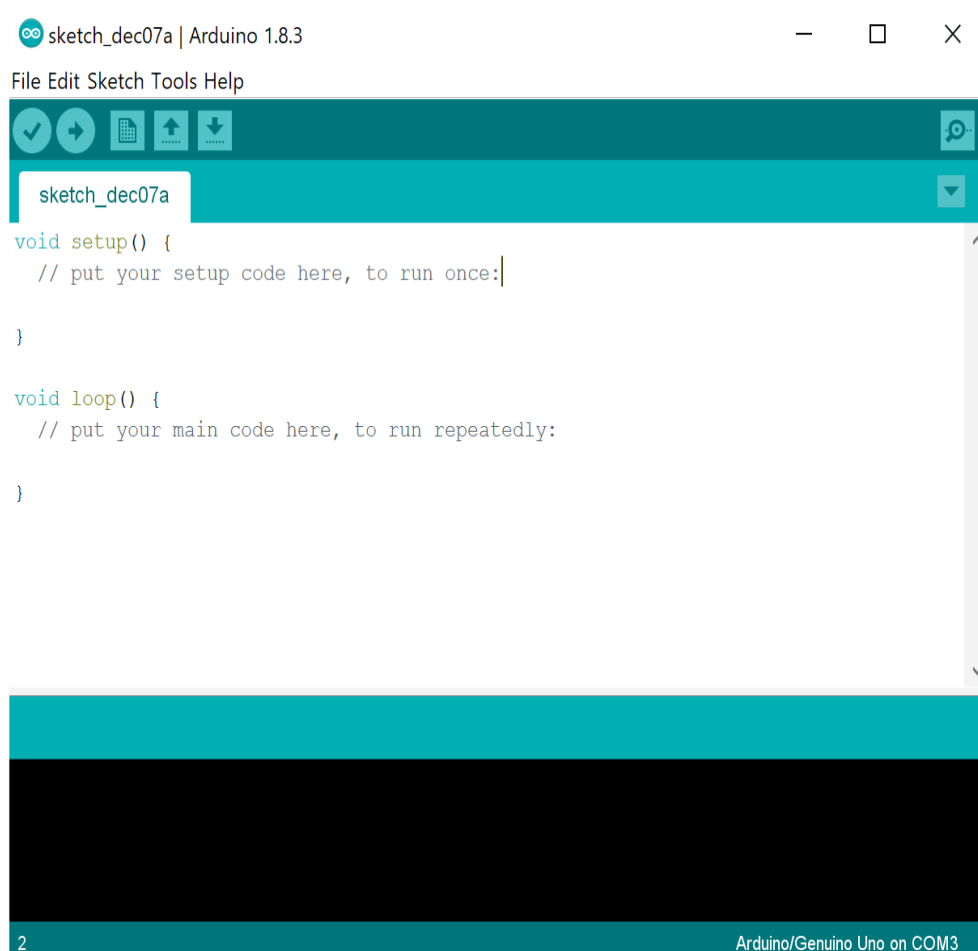
#### **3.3.1 Aurdino IDE**

Arduino IDE is an open-source software program that allows users to write and upload code within a real-time work environment. As this code will thereafter be stored within the cloud, it is often utilized by those who have been searching for an extra level of redundancy. The system is fully compatible with any Arduino software board.

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

## Main Functions and Uses

Arduino IDE can be implemented within Windows (11, 10, 8.1, 8, 7), Mac and Linux operating systems. The majority of its components are written in JavaScript for easy editing and compiling. While its primary intention is based around writing codes, there are several other features worth noting. It has been equipped with a means to easily share any details with other project stakeholders. Users can modify internal layouts and schematics when required. There are in-depth help guides which will prove useful during the initial installation process. Tutorials are likewise available for those who might not have a substantial amount of experience with the Arduino framework.



*Fig 3.18: Aurdino IDE application on PC.*

## PROS

1. This is open-source software and no subscription fees will be necessary.
2. Enhanced and intuitive tools provide users with access to advanced coding applications.

## CONS

1. Arduino IDE is meant for those with coding experience.
2. There is no telephone number dedicated to live customer support

### 3.3.2 Firebase:

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Realtime syncing makes it easy for your users to access their data from any device: web or mobile, and it helps your users collaborate. Realtime Database ships with mobile and web SDKs so you can build apps without the need for servers. You can also execute backend code that responds to events triggered by your database using Cloud Functions for Firebase.

When your users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. The Realtime Database integrates with Firebase Authentication to provide simple and intuitive authentication for developers. You can use our declarative security model to allow access based on user identity or with pattern matching on your data. Firebase helps you develop high-quality apps, grow your user base, and earn more money. Each feature works independently, and they work even better together.

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real-time events continue to fire, giving the end-user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and

when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great real-time experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then structure it accordingly.



*Fig 3.19: Google firebase logo.*

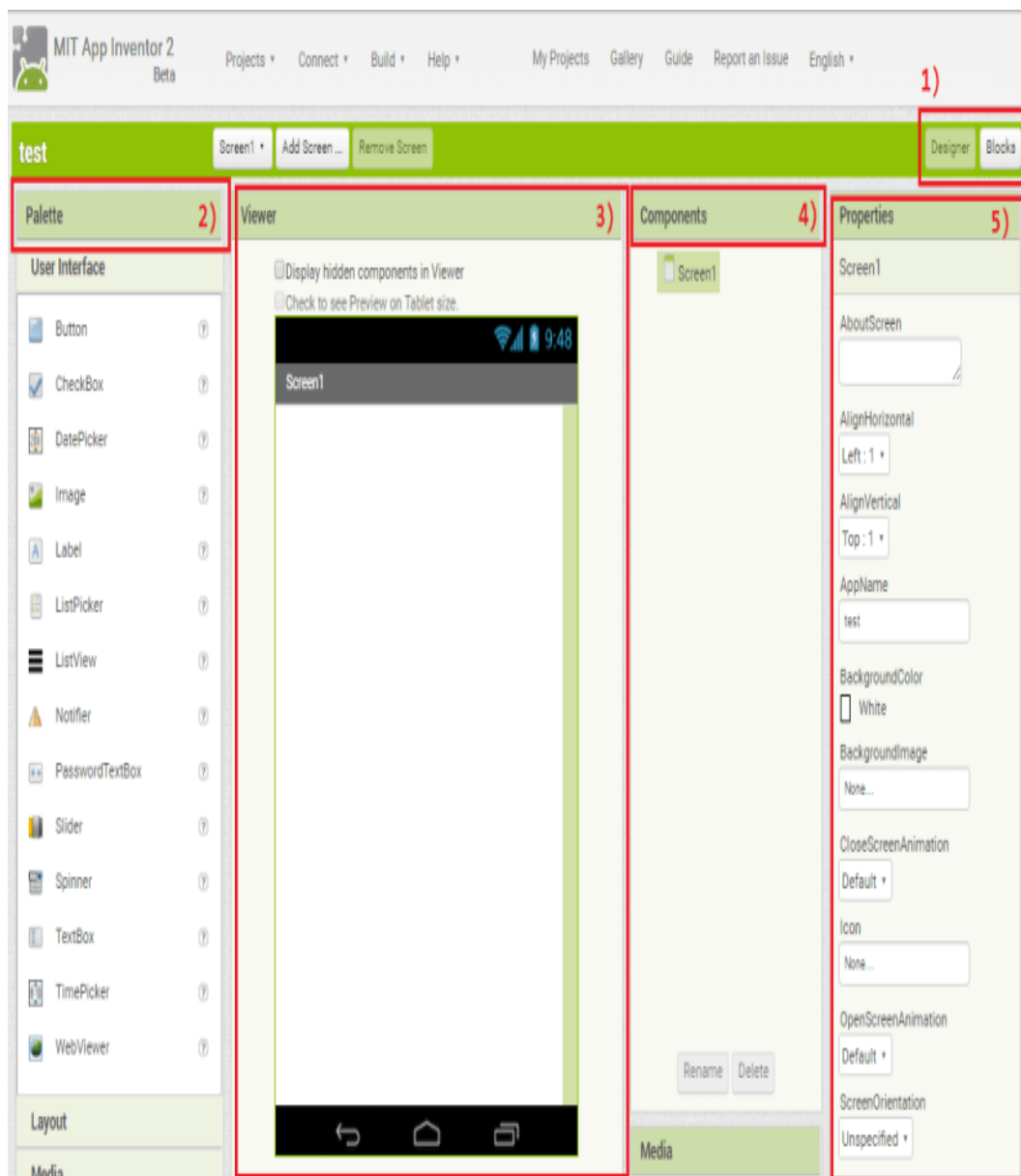
### **3.3.3 MIT App Inventor**

MIT App Inventor is a web application integrated development environment originally provided by Google and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software (apps) for two operating systems (OS): Android (operating system) Android, and iOS, which, as of 8 July 2019, is in final beta testing.

It is free and open-source software released under multi-licensing dual licensing: a Creative Commons license attribution Creative Commons Attribution Share-alike 3.0 unported license and an Apache License 2.0 for the source code. It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the Star logo,



which allows users to drag and drop visual objects to create an application that can run on mobile devices. In creating App Inventor, Google drew upon significant prior research in educational computing, and work done within Google on online development environments. App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasize that programming can be a vehicle for engaging powerful ideas through active learning.



*Fig 3.20: MIT Website welcome page.*

### 3.3.4 MIT AI2 Companion



*Fig 3.21: Android app logo.*

The MIT AI2 Companion app is used to download and run apps created with App Inventor, a web-based app building tool, on an Android phone or tablet. App Inventor has a useful tool for continuously seeing your app in real time on an Android device during each step of the development process. You can find the MIT AI2 Companion app in the Google Play Store by performing a search for “MIT AI2 Companion.”

**Step 1:** Download and install the MIT App Inventor Companion app on your Android or iOS device.

Open the Google Play store or Apple App store on your phone or tablet, or use the buttons below to open the corresponding page:

After downloading, step through the instructions to install the Companion app on your device. You need to install the MIT App Inventor Companion app only once, and then you can leave it on your phone or tablet for whenever you use App Inventor. Alternatively, you can scan the following **QR codes** to get either the iOS or Android app:

- **For iOS**, scan the code to go to the Companion app on the Apple App Store.

- **For Android**, scan the code to download the Android .APK file for the Companion app directly to your device. (Using an .APK file requires sideloading app on your device and updating the app manually in the future.)

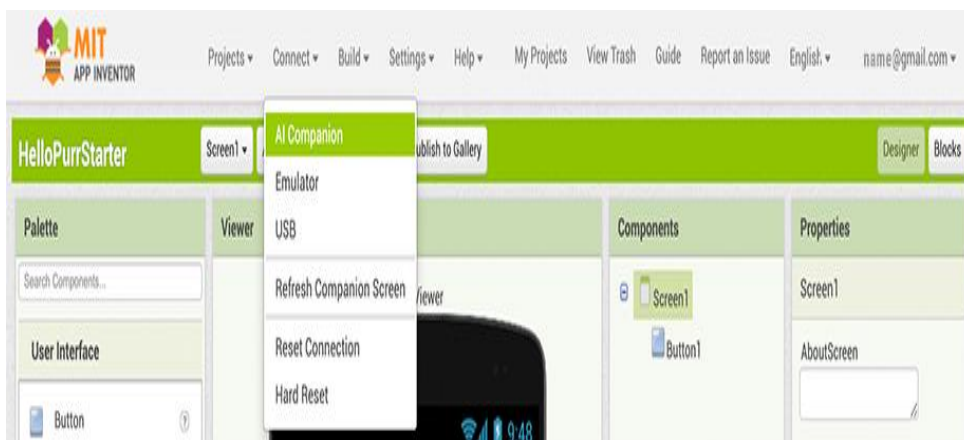
**Step 2:** Connect both your computer and your device to the **SAME** Wi-Fi network

App Inventor will automatically show you the app you are building, but only if your computer (running App Inventor) and your device (running the Companion) are connected to the same Wi-Fi network.

**Step 3:** Open an App Inventor project and connect it to your device.

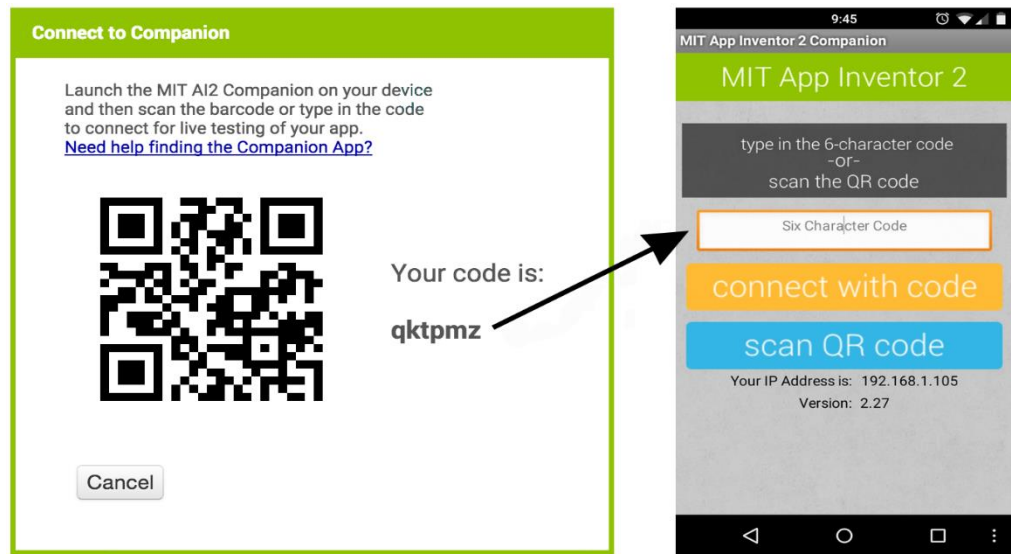
Go to App Inventor and open a project (or create a new one — use *Project > Start New Project* and give your project a name).

Then Choose “Connect” and “AI Companion” from the top menu in your browser:



*Fig 3.22: AI companion on webpage.*

A dialog with a QR code will appear on your PC screen. On your device, launch the MIT App Companion app just as you would do any app. Then click the “Scan QR code” button on the Companion, and scan the code in the App

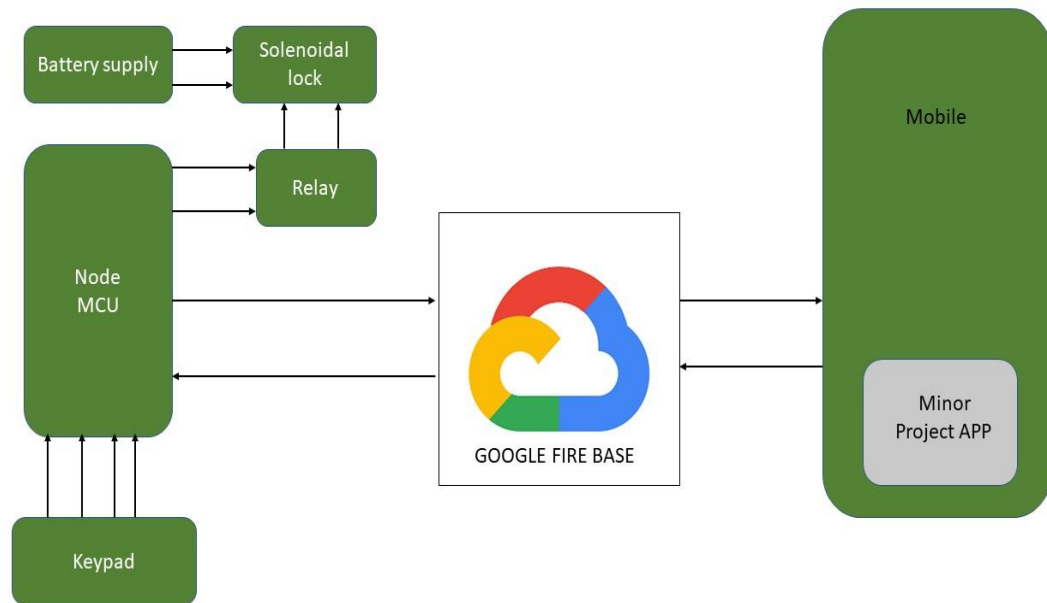
**Inventor window:***Fig3.23: Ways to connect with APP inventor.*

Within a few seconds, you should see the app you are building on your device. It will update as you make changes to your design and blocks, a feature called “live testing.”

If you have trouble scanning the QR code or your device does not have a scanner, type the code shown on the computer into the Companion’s text area on your device exactly as shown. The code is directly below where the screen on your PC shows “Your code is” and consists of six characters. Type the six characters and choose the orange “Connect with code.” Do not type an Enter or carriage return: type just the six characters followed by pressing the orange button.

*Fig.3.24 Representation*

### 3.4. Block Diagram:



### 3.5 Working:

#### 3.5.1 Application/ Software:

#### Login Screen:

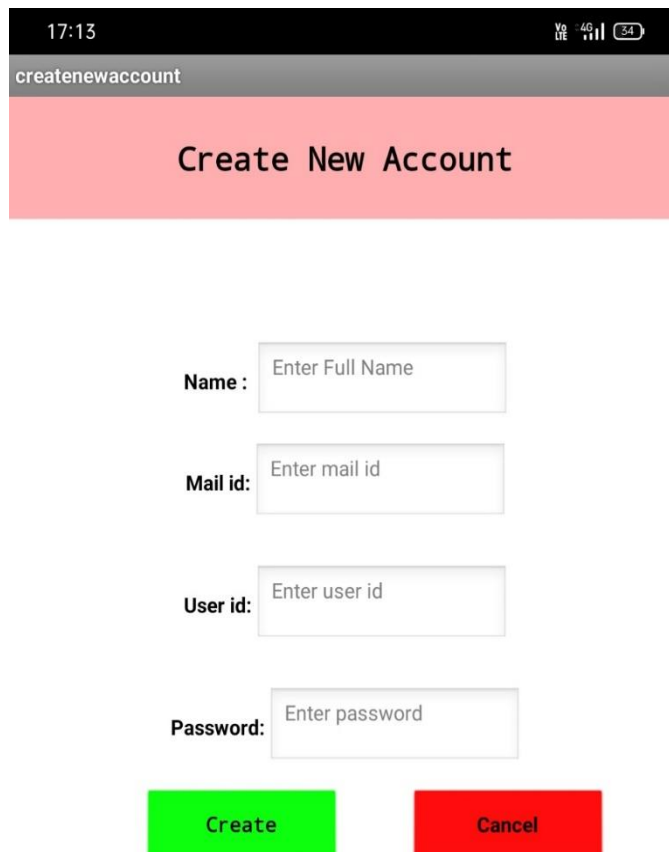


Fig 3.25: Login screen on

Login page contains three different redirecting pages:

- New user.
- Forgot password.
- Login button

If you are the new user you have to provide name of the candidate, email-id, new user Id, password. If you submit it once, a new data base for your device will create and in database it creates a new credentials block under your specified database, stores all the data of your specified account. Internal working of page redirecting was given as following.



The screenshot shows a mobile application interface for creating a new account. The top status bar displays the time 17:13, 4G signal, and 34% battery. The app title bar reads 'createnewaccount'. The main content area has a pink background with the text 'Create New Account'. Below this, there are four labeled input fields: 'Name :', 'Mail id:', 'User id:', and 'Password:'. Each field contains a placeholder text: 'Enter Full Name', 'Enter mail id', 'Enter user id', and 'Enter password'. At the bottom, there are two buttons: a green 'Create' button and a red 'Cancel' button.

*Fig 3.26: New user screen on application*

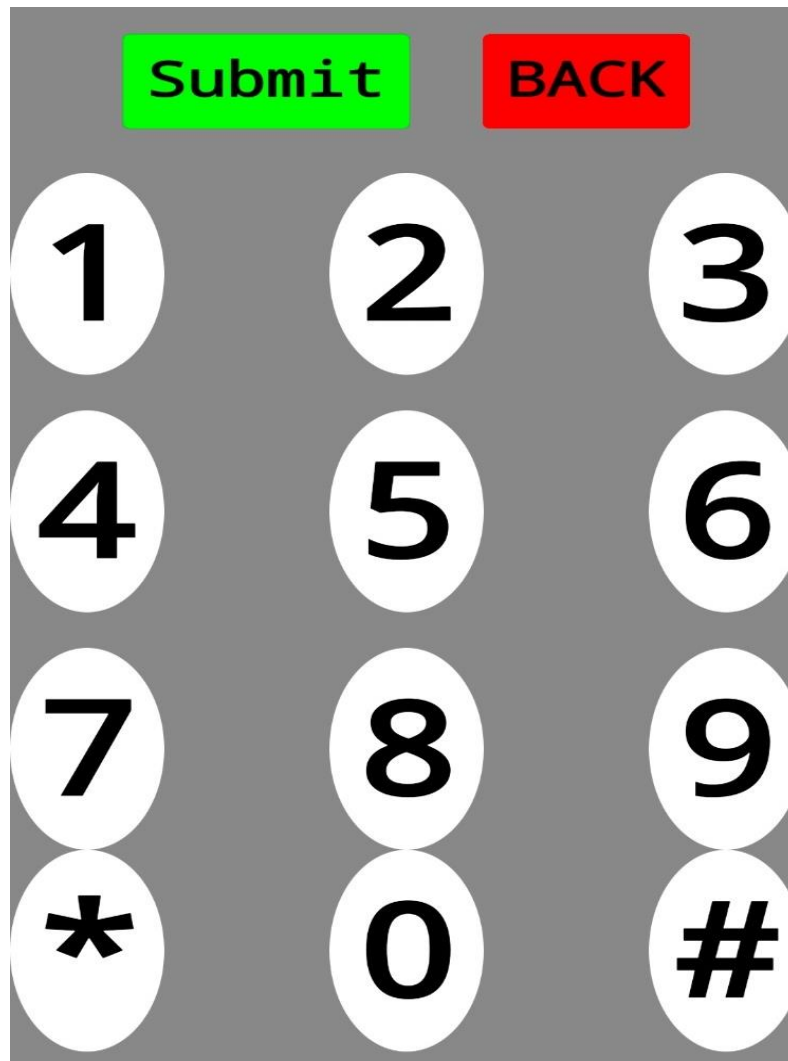
If login credentials are verified from database are correct then the screen redirect to remote access screen (or) main screen. Login has text box to take inputs. For password we are using password textbox. So, we can hide the password that enters from others.

If you forgot the password of your existing account, you can recover it by tap on forgot password button. If you once tap the button below display was shown to user on application.

*Fig 3.27: Forgot password screen on Application*

When we enter the user id and hits on submit button. The application checks on database for the email id which was provided by user in the registration process. After submitting the page redirects to the OTP screen. Here you have to enter the OTP which was sent to that user's mail Id. If you enter the correct OTP with in the time period the new password set page will open or else the user has the option as resend mail. If once you set the new password database was updated. So, user can login with new password to the application. The OTP page looks like the below image.

*Fig 3.28: OTP Screen on mobile application.*

**Main Screen:**

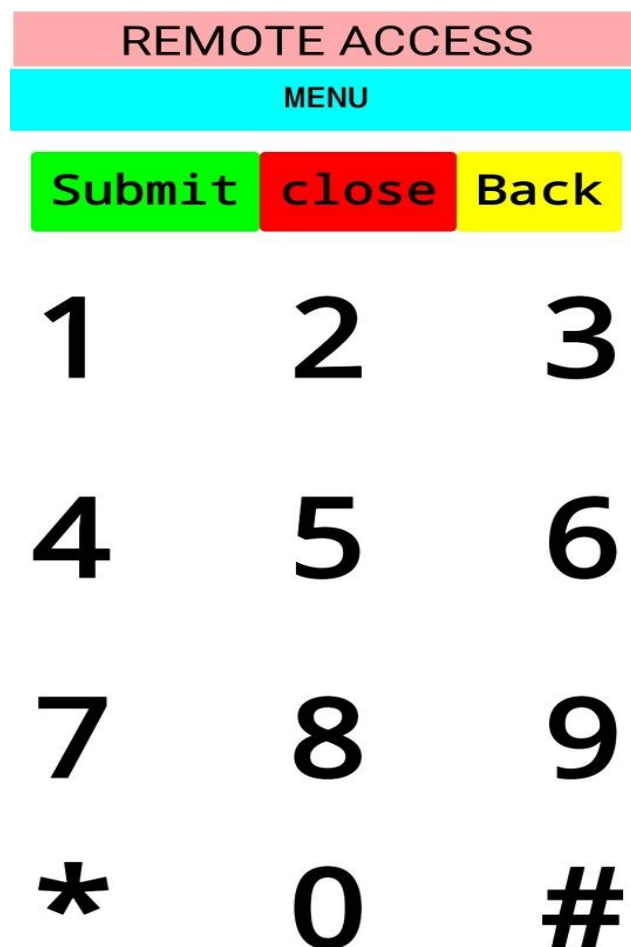
*Fig 3.29: keypad replica on application*

We used a keypad layout similar to the one used in the hardware components. This main page is used to change the safe locker's unlock pin. This page was directly connected to Google Fire Base using our account's access key, and through the database link updated the new password every time we updated the application. Initially, all of the buttons in this section are set to the default numbers. After changing the safe lock pin, press the submit button. Then only the new password will be updated on Google Fire Base at that point.



## REMOTE ACCESS:

A user can access the safe remotely. Consider a real time application that user has to open the safe for others but user won't trust that person. So in this kind of scenarios, you can unlock the safe through the application by entering the key on it. If you submit it the cloud will automatically send a signal to nodemcu that we have to unlock the locker and it will unlock physically.



*Fig 3.30: keypad in remote access screen*

In the same screen we have the redirecting options to several screens through menu tabular column button which was available in the top of the screen. By clicking on name of the screen you need to move it will redirect to that page. If you drop down the menu bar it will show like as in picture below.

## OVERALL CONTROL

---

RESET PASSWORD

---

NUMBER SWAP

---

CHANGE PIN

---

RELAY CONTROL

---

HELP

---

SIGN OUT

*Fig 3.31: menu bar on application*

### **Overall Control Screen:**

We can unlock and locks the safe without entering any pin if you have login with you. This is only for emergency cases. If you need you can also add an authentication page before entering this page to increase the security to the page access.

If we tap on open button the safe lock was open automatically, but it won't lock again after some time we have to close from application again. If it is open state no actions were taken.

If we tap on close button the safe gets locked again. If it is close state no actions were taken.

To get this option you have to follow the path mentioned below.

LOG-IN > MENU > Overall Control.



*Fig 3.32: Control screen on Application*

### **RESET PASSWORD:**

If the user wants to change the password the application login. User can achieve by using this screen. But user has to provide the old password in-order to change the password also he/she has to confirm the password by re-entering it.

If user gave the wrong previous password or two new passwords mismatched then the updating trail was failed.

To access this page the path as shown below:

LOGIN> MENU> RESET PASSWORD.

*Fig 3.33: Reset password screen on application*

### **NUMBER SWAP SCREEN:**

We can modify the values on the screen two keypad's backend by utilizing this page. Each number on the main screen is internally labelled in this case. Every number on this keypad has a label assigned to it. These labels can be changed internally through the number swap screen, and any information provided in the text boxes next to the numbers will have an impact on how the keypad function on the main screen.

For example, the number 1 has changed to 0. Then when you tap on 1 button on main screen it internally takes as 0. Because in the number swap screen it is fixed as 0.

After swapping the numbers positions that you want to change in keypad you have to submit it to replicate these operations to on-screen keypad. If you are failed to submit them no changes takes place internally.

If you want to reset the positions of each number you have to tap on reset button so all the text boxes will set to default numbers automatically.

To reach this screen you have to follow the below path.

LOGIN> MENU> NUMBER SWAP> Change the values> Submit.

The screen layout was given on below picture.

*Fig 3.34: number swap screen on application*

## RELAY CONTROL:

In order to control the opening and closing actions of solenoidal valve we are using this relay control screen. Through this screen we are setting password into special characters so now one can access the solenoidal lock because these characters not available in keypad. Even though they enter the correct password the locker won't open until and unless the enable the relay gate from APP.

The path to this screen was as given below.

LOGIN>MENU>RELAY CONTROL.

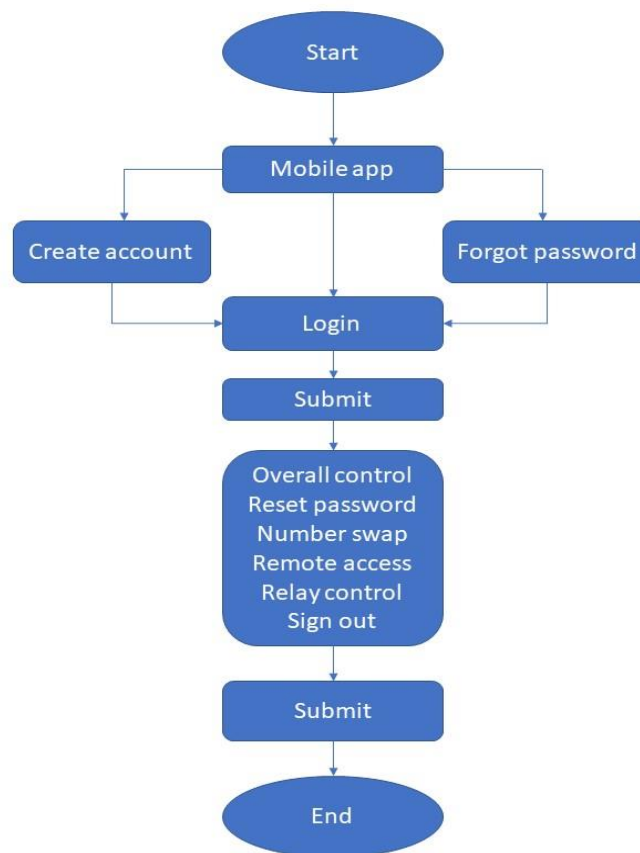
The screen of this relay control was shown on below figure, it just contains two buttons when you tap on enable button the supply to the relay was stops and also physical unlocking action

was blocked. Until you disable your access to relay switch. Until you tap on the disable button it won't set to normal state.



Fig 3.35: Relay screen on application

### 3.6 Flow Chart



## Chapter 4

### Performance Analysis

#### 4.1 Results:

- When the safe is open through application. The following message was displayed on screen of serial monitor. The cloud test value also updated as “1”.

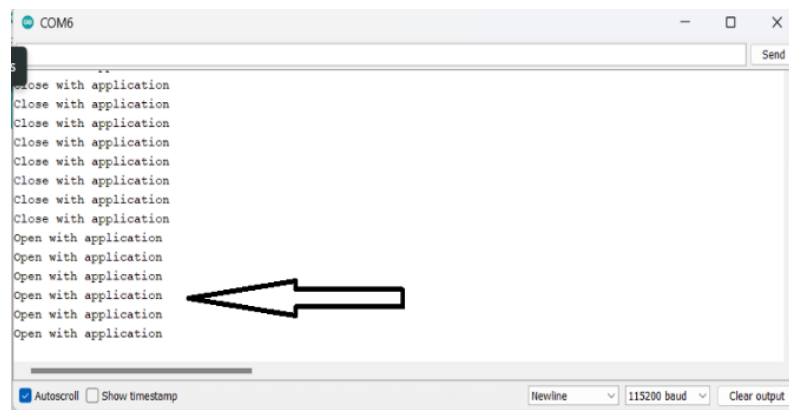


Fig:4.1 serial monitor output.

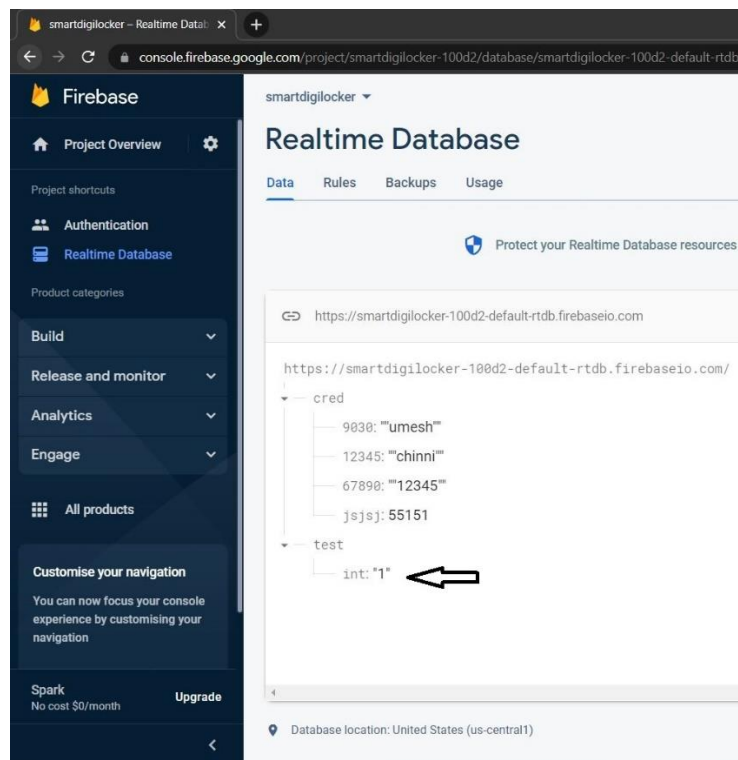


Fig:4.2: cloud test value updated to “1”.

- When safe locked from application. The following message was displayed on screen of serial monitor. The cloud test value also updated as “0”.

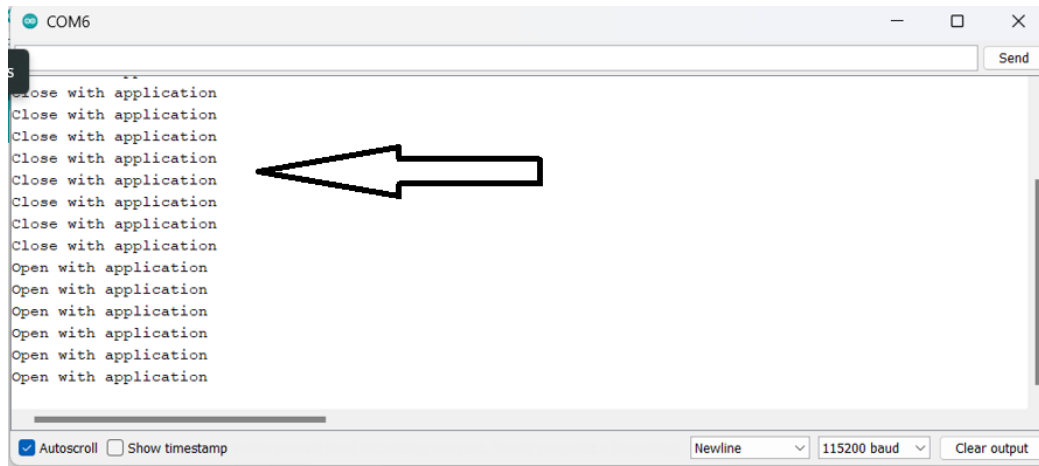


Fig:4.3: Serial monitor output.

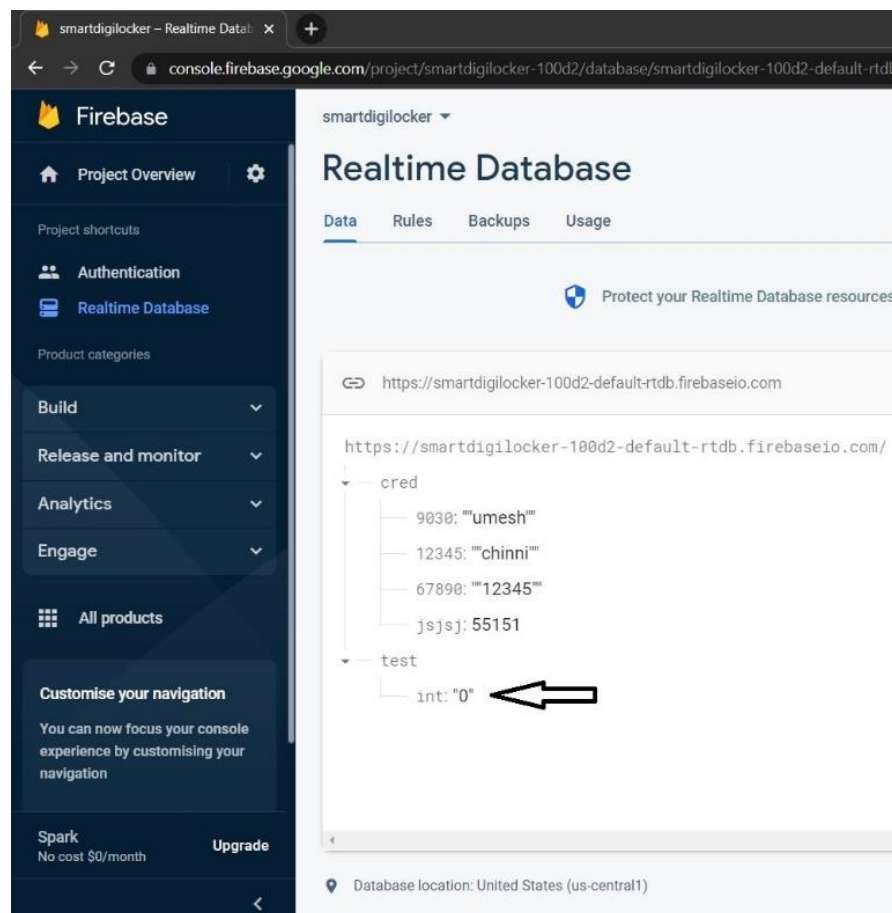




Fig:4.4: cloud test value updated to “0”.

- When updated password from mobile the database updated as follow.

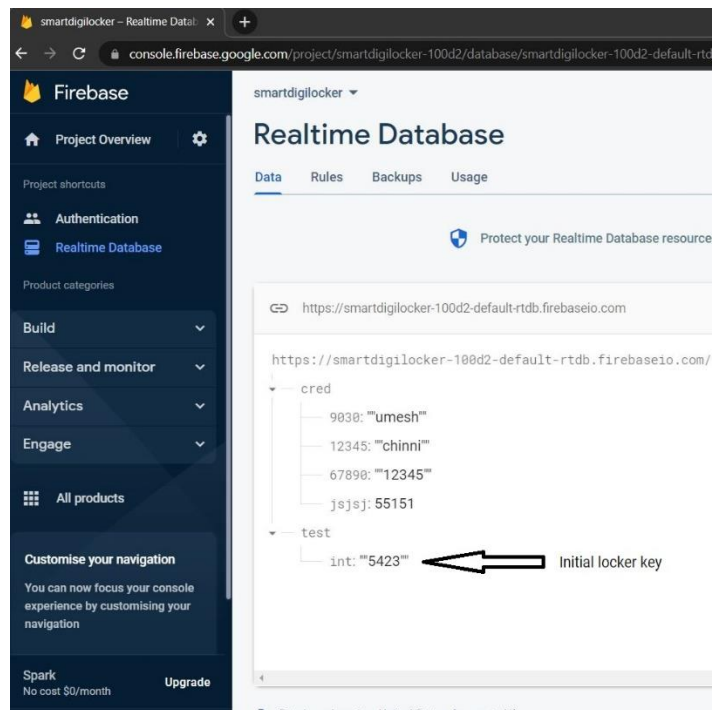


Fig:4.5 Initial locker key on cloud.

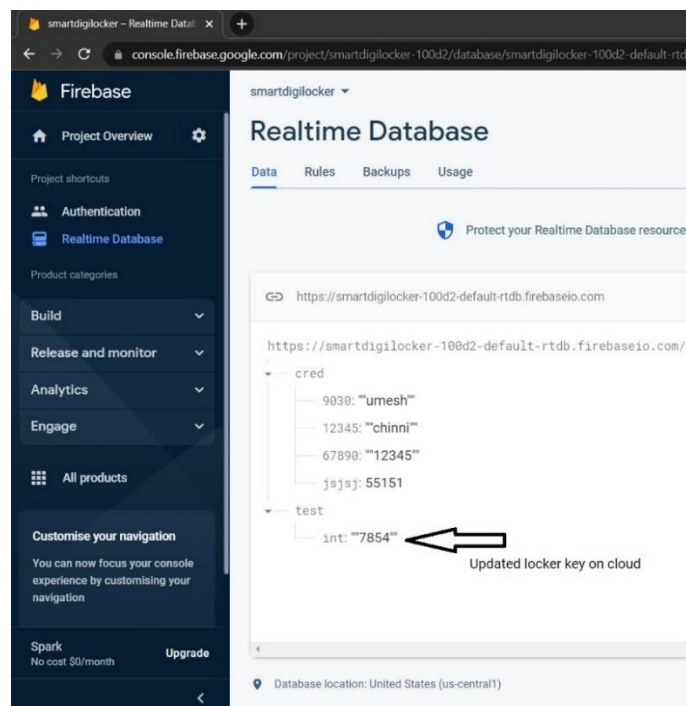


Fig:4.6 updated locker key on cloud.

- When safe locker opens through keypad installed on safe itself.

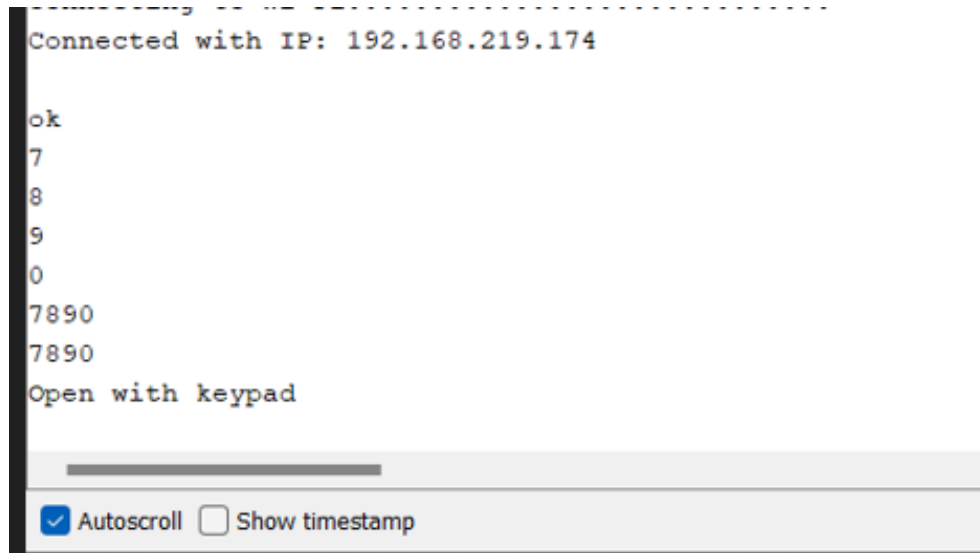


Fig:4.7: Serial monitor output when safe opens with keypad.

- When a person enters wrong password on keypad.

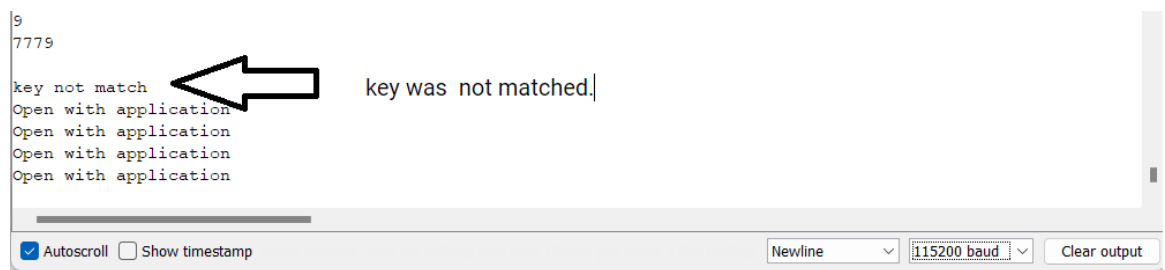


Fig:4.8: serial monitor message.

- When we disable the relay control the locker key was set to the “null” as we are using a number keypad a user was unable to unlock the safe until unless the user enables the relay function by tapping on enable button.
- When user taps on enable button, the value gets back from backup bucket used in google firebase.

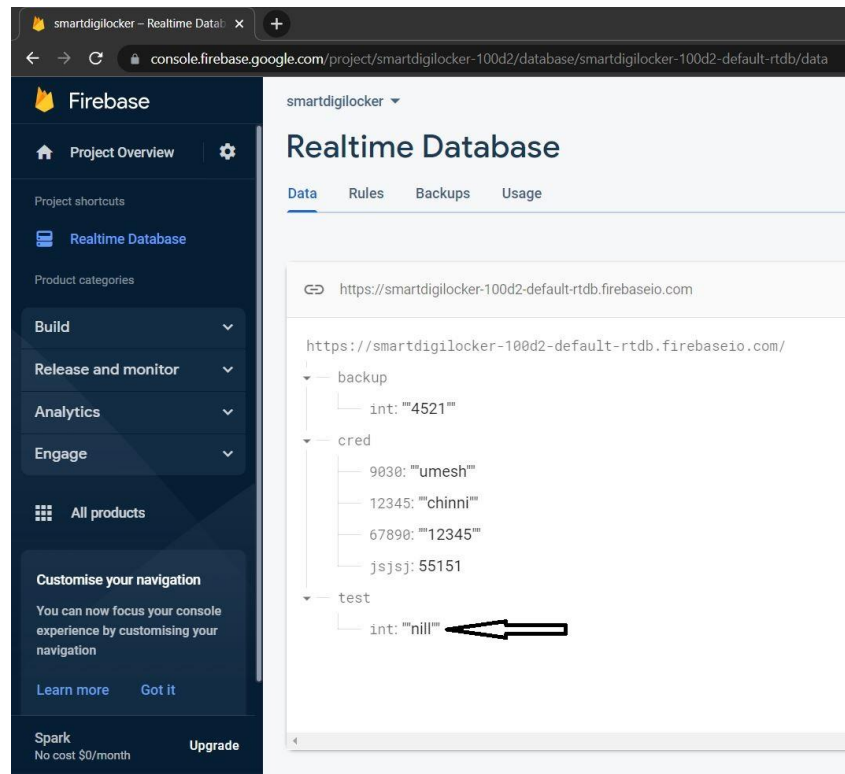


Fig:4.9: when relay control was disabled.

- When we open the safe with remote access. The test value becomes.

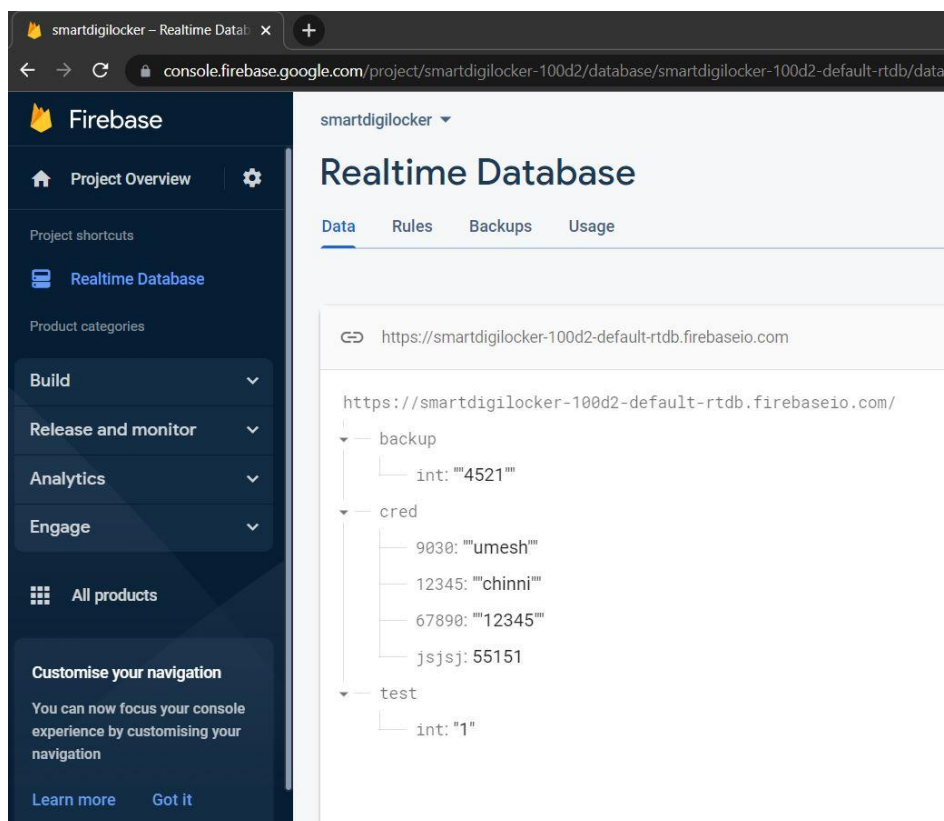


Fig:4.10: Remote access modification

## 4.2 Hardware Code:

```
#include <ESP8266WiFi.h>

#include <Firebase_ESP_Client.h>

#include <Keypad.h>

#include "addons/TokenHelper.h"

#include "addons/RTDBHelper.h"

#define WIFI_SSID "123456789"

#define WIFI_PASSWORD "123456789"

#define API_KEY "AIzaSyBiqFXyapja_jKNTFgvNkuYO90MJalInIQ"

#define DATABASE_URL "https://smartdigilocker-100d2-default-rtdb.firebaseio.com/"

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;

bool signupOK = false;

String intValue;

int lpin = 2;

const byte ROWS = 4;

const byte COLS = 3;

String code = "";

int len;

char hexaKeys[ROWS][COLS] = {

    {'1','2','3'},

    {'4','5','6'},
```

```

    {'7','8','9'},

    {'*','0','#'}

};

byte rowPins[ROWS] = { 16, 5, 4, 0}; //connect to the row pinouts of the keypad

byte colPins[COLS] = { 14, 12, 13}; //connect to the column pinouts of the keypad

Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS);

void setup(){

    pinMode(lpin, OUTPUT);

    Serial.begin(115200);

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    Serial.print("Connecting to Wi-Fi");

    while (WiFi.status() != WL_CONNECTED){

        Serial.print(".");

        delay(300);

    }

    Serial.println();

    Serial.print("Connected with IP: ");

    Serial.println(WiFi.localIP());

    Serial.println();

    config.api_key = API_KEY;

    config.database_url = DATABASE_URL;

```

```

if (Firebase.signUp(&config, &auth, "", "")){

    Serial.println("ok");

    signupOK = true;

}

else{

    Serial.printf("%s\n", config.signer.signupError.message.c_str());

}

config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);

}

void loop(){

    char customKey = customKeypad.getKey();

    if (customKey)

    {

        if (customKeypad.isPressed('1'))

        {

            code = code+customKey;

            Serial.println("1");

        }

        else if (customKeypad.isPressed('2'))

        {

```

```
code = code+customKey;

Serial.println("2");

}

else if (customKeypad.isPressed('3'))

{

code = code+customKey;

Serial.println("3");

}

else if (customKeypad.isPressed('4'))

{

code = code+customKey;

Serial.println("4");

}

else if (customKeypad.isPressed('5'))

{

code = code+customKey;

Serial.println("5");

}

else if (customKeypad.isPressed('6'))

{

code = code+customKey;

Serial.println("6");

}
```

```
else if (customKeypad.isPressed('7'))  
  
{  
  
    code = code+customKey;  
  
    Serial.println("7");  
  
}  
  
else if (customKeypad.isPressed('8'))  
  
{  
  
    code = code+customKey;  
  
    Serial.println("8");  
  
}  
  
else if (customKeypad.isPressed('9'))  
  
{  
  
    code = code+customKey;  
  
    Serial.println("9");  
  
}  
  
else if (customKeypad.isPressed('0'))  
  
{  
  
    code = code+customKey;  
  
    Serial.println("0");  
  
}  
  
len = code.length();  
  
if (len == 4)  
  
{
```



```

Serial.println(code);

intValue.remove(0,2);

intValue.remove(4,2);

Serial.println(intValue);

if (intValue == code)
{
    digitalWrite(lpin, LOW); // turn the LED on (HIGH is the voltage level)

    Serial.println("Open with keypad");

    delay(3000);

    digitalWrite(lpin, HIGH);
}

else

{
    Serial.println("key not match");
}

code = "";
}

}

if (Firebase.ready() && signupOK)
{
    if (Firebase.RTDB.getString(&fbdo, "/test/int"))
    {

```

```
intValue = fbdo.stringData();

//Serial.println(intValue);

if (intValue == "1")

{

    digitalWrite(lpin, LOW); // turn the LED on (HIGH is the voltage level)

    Serial.println("Open with application");

    delay(100);

}

else if (intValue == "0")

{

    digitalWrite(lpin, HIGH); // turn the LED on (HIGH is the voltage level)

    Serial.println("Close with application");

    delay(100);

}

delay(100);

}

else {

    Serial.println(fbdo.errorReason());

}

delay(100);

}

delay(100);

}
```

## **Chapter 5**

### **Conclusion and Future Scope**

#### **5.1. Advantages:**

##### **Provides High Security:**

- This project has Multiple Options to secure your data and your valuables.
- Every User is given a unique User id and password.
- Every User gets a Unique & Customized Device Setup for them.
- Their Application Login data is Stored on Google Firebase which is Difficult to Hack.

##### **Swapping Of Internal Arrangement of system.**

- The Generated 4-Digit PIN has Multiple number of Combinations.
- As user can Swap your Internal arrangement of the keypad, that in turn Creates more difficulty for Hackers, because Except the User nobody knows the Right Arrangement of the Keypad System.

##### **Provides Remote access to the Locker**

- As Long as there's WIFI or Internet Connected to the Locking System and to the Device in which you are using the Application. You can Have and Use all Controls of the Locking System by Logging In to your Application with Provided User Id and Password.

##### **A new Generation Technological System**

- We all have seen Key-Lockers, Key-Safe, PIN-Password, Fingerprint, Face Unlock etc.,
- But This Project Uses IOT System to not only provide PIN-Passcode system but this System provides Swapping of Internal organization of the Keypad with Remote Accessibility to you Locker from anywhere in the world with just "ONE CLICK" through your application.
- You can change your PIN of you Locker, Multiple times from wherever you are.
- 

##### **Cost Effective**

- This Locking system is very Cost efficient because the Components needed for the device are easily available and are not much complex in handling.

### **It is a Flexible system & Easy to Install**

- This Locking System is Easy to Setup and the Application is easy to Install into your Devices and provides good Lifespan.

### **5.2. Disadvantages:**

#### **WIFI Connection**

- This Locking Device & Mobile Application requires WIFI or Internet Connection as It is an IOT based system and as we are providing Remote Access. We need Internet to make it work.

#### **Requires Power Supply**

- This Locking Device Requires Energy to work, so there should be a Power Supply Given to the Device.

#### **Batteries Check**

- To solve the problem of power supply We can use Batteries, but they need to be checked occasionally.

### **5.3. Applications:**

- It can for Safe Locks at Home.
- It Can be Used for Bank Vaults.
- It can be used for personal desks.
- It can also be used for door locking systems.
- It can be used for personal Vaults.

### **5.4. Future scope:**

- The Present existing Safe Lock models can be Updated to The Proposed project for much Higher security.
- This project's number swapping technique can be applied to the UPI PIN keypads to protect it from the UPI Frauds.
- This project can be applied on your mobile or laptop passcodes.

- This project can be applied to your home door locking systems to provide access to your friends and family from where ever you are with just one click from your mobile application.
- This project can be applied to all the Bank Vaults.
- This Project's number swapping system can be applied to any of your Mobile applications & Gallery to Secure them.
- This project can create Customized Safe locks according to the User's Requirement.
- Modernization of IOT based Safety Locks is Possible

### **5.5. Conclusion:**

We can Conclude that this Proposed System provides High Security Smart Automated Locking Systems to your Homes, Safe, Bank Vaults etc. It can be Implemented at Low Cost with No Overhead like Drafting and Construction works. It is Difficult to be Manipulated or Hacked as it has multiple Number of combinations, Remote Accessibility i.e. it can be accessible for the User from anywhere and Number Swapping technology through which we can change the internal PIN number functionality of the external Keypad and the keypad in the Application. Your User ID's, Password, PIN code are stored in the Google Firebase Cloud Storage System, which is Highly Secured by the Google itself. Power Supply to the Device is Necessary However it can also work on Batteries so no power loss issues. This System is very flexible and easy to install as it has easily available and a smaller number of components. Every User gets a personally Customized Device Set-Up for them, and they can change their PIN-Passcode whenever and from wherever they are, provided there is Internet Connection & System Application available in their devices.

## REFERENCES

- [1].Shanthini M, vidya G, Arun G, “IoT Enhanced Smart Door Locking System,” in IEEE, Doi: October 24,2020 from IEEE Xplore.
- [2].Jishant Singh, Rashmi Roges, Sandeep Sharma, Anuradha Bhasin, Rasheed Kumar, Jatin Gaur, “Digital Keypad Security System Based on Arduino with GSM Module, Alarm, and Temperature Sensor” in IEEE general
- [3].Ayush Sudhakar Kohade, Neeraja Ravindra Khire, Khushi Ajay Bhartiya, Akshay Khubchandani, Pavankumar Bharat Khot, “Simulation of Keypad-Based Door unlocking system using tinker cad” in IEEE general
- [4].Atif Afroz,” Digital Smart Door Lock Security System Using Arduino Uno Microcontroller”, in IRE Journals, Doi: JUL 2022.
- [5].Ketan Rathod, Prof. Rambabu Vatti, Mandar Nandre, and Sanket Yenare (2017), “Smart door security using Arduino and Bluetooth application”, International Journal of Current Engineering and Scientific Research, 4(11), 73-77.
- [6].Lubhansh Kumar Bhute, Gagandeep Singh, Avinash Singh, Vikram Kansary, Preetam Rao Kale, and Shailendra Singh (2017), “Automatic Door Locking System Using Bluetooth Module”, International Journal for Research in Applied Science & Engineering Technology, 5(05), 1128-1131.
- [7].Rahul Satoskar and Akarsh Mishrac (2018), “Smart Door Lock and Lighting System using Internet of Things”, International Journal of Computer Science and Information Technologies, 9(05), 132-135
- [8].Nishad N. Gupte and Mihir R. Shelar (2013), “Smart Door Locking System”, International Journal of Engineering Research & Technology, 2(11), 2214-2217
- [9].Hashem Alnabhi, Yahya Al-naamani, Mohammed Al-madhehagi, and Mohammed Alhamzi (2020), “Enhanced Security Methods of Door Locking Based Fingerprint”, International Journal of Innovative Technology and Exploring Engineering, 9(03), 1173-1178
- [10]. Agbo David, O., Madukwe Chinaza, and Odinya Jotham, O. (2017), “Design And Implementation Of A Door Locking System Using Android App”, International Journal Of Scientific & Technology Research, 6(08), 198-203.

- [11]. Hashem Alnabhi, Yahya Al-naamani, Mohammed Al-madhehagi, and Mohammed Alhamzi (2020), “Enhanced Security Methods of Door Locking Based Fingerprint”, International Journal of Innovative Technology and Exploring Engineering, 9(03), 1173-1178.
- [12]. Shahid Sohail, Sajid Prawez, and Raina, C.K. (2018), “A Digital Door Lock System For The Internet Of Things With Improved Security And Usability”, International Journal of Advance Research, Ideas and Innovations in Technology, 4(03), 878-880.
- [13]. R. Das, and G. Tuna (2017), “Packet tracing and analysis of network cameras with Wireshark”, 5th International Symposium on Digital Forensic and Security(ISDFS).
- [14]. Haroon Iqbal, and Sameena Naaz (2019), “Wireshark as a Tool for Detection of Various LAN Attacks”, International Journal of Computer Science and Engineering, 7(05), 833-837.
- [15]. Chunnu Khawas, and Pritam Shah (2018), “Application of firebase in Android App Development – A Study”, International Journal of Computer Applications, 179(46), 49-53.
- [16]. Sivaganesan, D. (2019),” Design And Development Ai-Enabled Edge Computing For Intelligent-Iot Application”, Journal of trends in Computer Science and Smart technology (TCSST), 1(02), 84-94.
- [17]. Raj, J. S., and Ananthi, J. V. (2019), “Automation using IoT in greenhouse environment”, Journal of Information Technology, 1(01), 38-47.