MINI PROJECT REPORT

On

IOT BASED SMART LOCK

Submitted by

Shivam Kumar
Roll No: 171500320
Shikha Bansal
Roll No: 171500314

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University
Mathura- 281406, INDIA

2019

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our mentor **Mr. Mandeep Singh** for guiding us. We deeply respect the mentor for his vast knowledge, numerous suggestions, and strong passion to complete this project. Valuable discussions with him not only made our work smooth but also encouraged us to think more professionally in the field of Internet of Things.

We also thank all our teaching and non-teaching staff for their support and well wishes.

Finally, We would like to express our deepest gratitude to our parents and friends for their encouragement and support.

- SHIVAM KUMAR AND SHIKHA BANSAL

ABSTRACT

This project presents an IoT based door locking System. Most of the major door lock security systems have several loopholes which could be broken down to gain access to desired places and it creates a concern for a secure lifestyle and proper working environment. Human identification field is very significant and has undergone rapid changes with time. An important and very reliable human identification method is fingerprint identification. Fingerprint of every person is unique. So this helps in identifying a person or in improving security of a system. Fingerprint of a person is read by a special type of sensor. Fingerprint sensors can be interfaced with a microcontroller. In this project we use a fingerprint sensor to read one's identity to automatically operate the door. For this ,we use a microcontroller to enable the door opening or closing if the matching between scanned data and the already existing data is correct. Comparison is done inside the fingerprint module itself and its output is given to the microcontroller. This is Node MCU based project.

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

DECLARATION

We hereby declare that the project work entitled "**IOT BASED SMART LOCK**" **s**ubmitted to the GLA University Mathura**,** is a record of an original work done by our team under the guidance of **Mr. Mandeep Singh, Technical Trainer of GLA University, Mathura.**

Name of Candidates: 1. Shivam Kumar(171500320)

2. Shikha Bansal(171500314)

Course: B.Tech (CSE)

Year: III

Semester: 6

Department of CEA,

GLAU Mathura

TABLE OF CONTENT

CHAPTER 1 INTRODUCTION

1.1 SIGNIFICANCE AND MOTIVATION

A **smart lock** is an electromechanical lock which is designed to perform locking and unlocking operations on a door when it receives such instructions from an authorized device using a wireless protocol and a cryptographic key to execute the authorization process. It also monitors access and sends alerts for the different events it monitors and some other critical events related to the status of the device. Smart locks can be considered part of a smart home.

Most smart locks are installed on mechanical locks (simple types of locks, including deadbolts) and they physically upgrade the ordinary lock. Recently, smart locking controllers have also appeared at the market.

Smart locks, like the traditional locks, need two main parts to work: the lock and the key. In the case of these electronic locks, the key is not a physical key but a smartphone or a special key fob configured explicitly for this purpose which wirelessly performs the authentication needed to automatically unlock the door.

Smart locks allow users to grant access to a third party by means of a virtual key. This key can be sent to the recipient smartphone over standard messaging protocols such as e-mail or SMS. Once this key is received the recipient will be able to unlock the smart lock during the time previously specified by the sender.

Smart locks are able to grant or deny access remotely via a mobile app. Certain smart locks include a built-in Wi-Fi connection that allows for monitoring features such as access notifications or cameras to show the person requesting access.[2] Some smart locks work with a smart doorbell to allow the user to see who and when someone is at a door.

Biometrics refers to the automatic identification of a living person based on physiologi-cal or behavioral characteristics for authentication purpose. Among the existing biomet-ric technologies are the face recognition, fingerprint recognition, finger-geometry, hand geometry, iris recognition, vein recognition, voice recognition and signature recognition, Biometric method requires the physical presence of the person to be identified. This emphasizes its preference over the traditional method of identifying what you have such as, the use of password, a smartcard etc. Also, it potentially prevents unauthorized admittance to access control systems or fraudulent use of ATMs, Time Attendance Systems, cellular phones, smart cards, desktop PCs, Workstations, vehicles and com-puter networks. Biometric recognition systems offer greater security and convenience than traditional methods of personal recognition. Fingerprint recognition represents the oldest method of biometric identification which is dated back to 2200 BC.

The use of fingerprints as a personal code has a long tradition and was already used.This system focuses on the use of fingerprints for door opening and closing. The fingerprint of valid users of the are to be enrolled in a database. Before any user can open the door, his/her fingerprint image is matched against the fingerprints in the database while users with no match in the database are prevented from opening the door. A microcontroller stores the data equivalent to the fingerprint of the master user. Comparison between this enrolled fingerprint and the fingerprint of the person who is about to open the door is done by the microcontroller. If both the fingerprints are identical, the control circuitry of the microcontroller sends appropriate signals to the motor relays operating the door. If the fingerprints are not identical, microcontrollers send signals to the alarm circuitry to warn about an unauthorised use.

1.2 AIM AND OBJECTIVE

A smart lock is an electronic and mechanical locking device that opens wirelessly with an authorized users' authentication.

In a smart home, smart locks allow a homeowner to enter their home or provide others access without requiring a traditional key. Instead, the user uses their smartphone or a key fob to wirelessly verify and mechanically unlock the door. Smart locks are an extension of home automation into home security. As a connected device, smart locks can be considered a part of the internet of things (IoT).

Many smart lock systems use mobile apps or websites to allow homeowners to grant access to third parties by sending a virtual key. Virtual keys can be sent by SMS text message or email, enabling access to guests or service personnel. Once received, these encrypted digital keys allow access to the smart lock for a preset period of time. On top of regulating access, many smart locks log access, providing the means to monitor use of a given secured door. Some smart locks feature a camera, which provides a picture of those accessing the door and makes for an easily referenced photo log.

Potential vulnerabilities in smart locks do create security concerns. Security advisors recommend that smart locks should not necessarily be thought of as more secure than a conventional lock and key and recommend that users think carefully about settings and options.

CHAPTER 2 WORKING PRINCIPLE (IOT)

Internet of Things (IoT) is a combination of interrelated devices which includes computing devices, microcontrollers, different types of sensors and machines which are interconnected with each other over the internet to perform a designated task with no human to computer interaction.

IoT is a technology that connects different devices to the internet and communicates with the user where the user can remotely control and monitor the device or just use the information fed to the system by the sensors and the use of IoT has exponentially increased in the recent years due to the availability of different technologies which will make it easy to connect to the internet.

IoT is integrated with four different characteristics which make it perform the task it is designated to.

Sensors and devices are devices which may have special properties or perform simple tasks to get information from the environment such as in this scenario to detect the presence of a vehicle with the help of sensors. The task of the devices mainly is to grab information from the environment.

Connectivity, this is the most important part of IoT which is to connect the sensors or the devices to a cloud, a platform that can be accessed through the internet. So, we should make sure that the internet connectivity is given. To achieve this, different methods are used such as using GSM (which is the most common form) or by Wi-Fi or connecting directly via Ethernet, which will route to the internet.

Data Processing, once the data is transmitted over the internet to the cloud/server then there have to be a data process which would be performed and displayed to the user.

IoT
SMART CITY
SMART VEHICLES
SMART HEALTH CARE
SMART GRIDS
SMART FARMING
SMART LOGISTICS
SMART INDUSTRIES
SMART HOMES

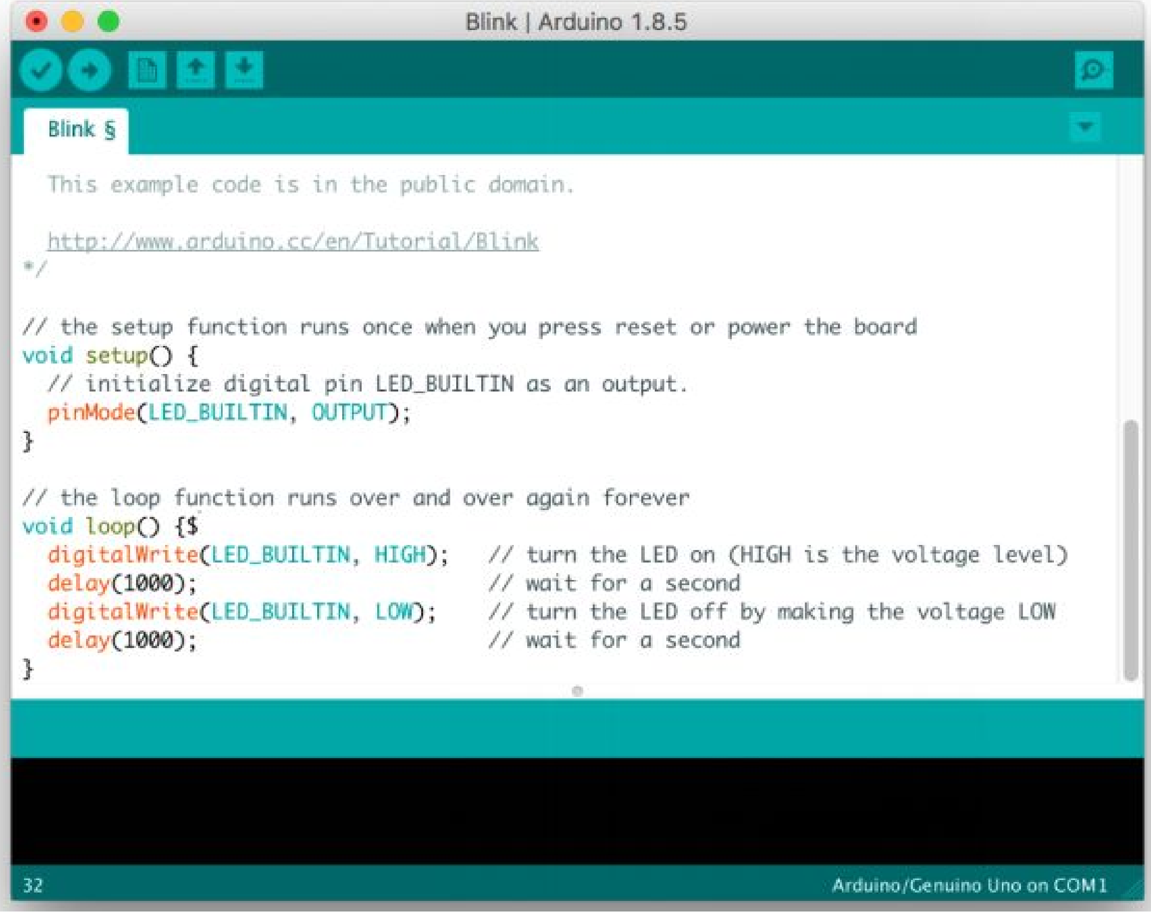
User Interface is the platform in which the data could be communicated with the user and also, the platform in which the user can remotely send instructions to devices to perform particular tasks such to activate or deactivate, on or off etc. The users can connect to the IoT application via a mobile or desktop application but the most commonly used platform as there is no need to design specific native apps and desktop apps for different vendors but just implement the user communication module in the web interface as it is compatible with all devices.

Our project will be following the general requirements of an IoT system working towards developing and deploying a stable and responsive system. The project will start out with the sensors found in parking areas and slots, which is connected to the NodeMCU ESP 32, which sends data collected from the different sensors to server and updating in real time for both the admin and the users that are interested in the system. The different components of the project are governed by the general rules of IoT and specific rules of the device in use.

CHAPTER 3 SOFTWARE REQUIREMENT

3.1. ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.



```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Fig 1. Arduino Uno IDE

The source code for the IDE is released under the GNU General Public License, version- 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging and other advanced features.

3.2. EMBEDDED C:

Embedded C Programming is the soul of the processor functioning inside each and every embedded system we come across in our daily life, such as mobile phone, washing machine, and digital camera. Each processor is associated with an embedded software. The first and foremost thing is the embedded software that decides functioning of the embedded system. Embedded C language is most frequently used to program the microcontroller.

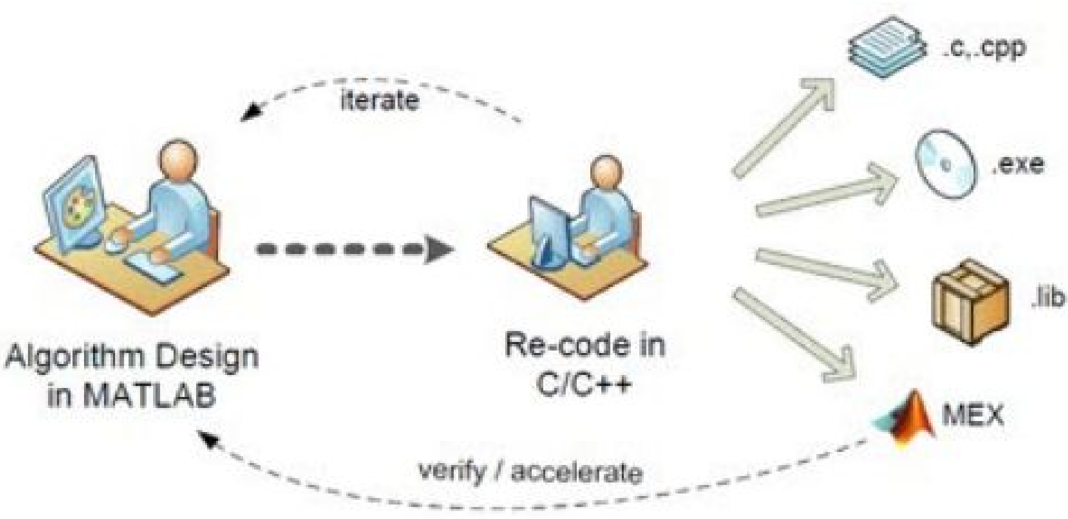


fig 2. Embedded C

Earlier, many embedded applications were developed using assembly level programming. However, they did not provide portability. This disadvantage was overcome by the advent of various high level languages like C, Pascal, and COBOL. However, it was the C language that got extensive acceptance for embedded systems, and it continues to do so. The C code written is more reliable, scalable, and portable. C language was developed by Dennis Ritchie in 1969. It is a collection of one or more functions, and every function is a collection of statements performing a specific task. C language is a middle-level language as it supports high-level applications and low-level applications. Before going into the details of embedded C programming, we should know about RAM memory organization.

Salient features of the language:

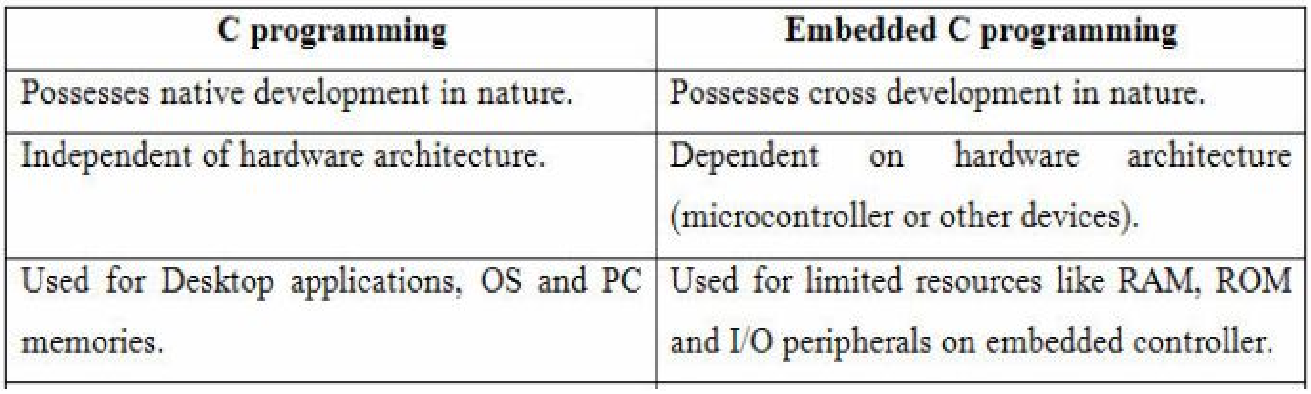
· C language is a software designed with different keywords, data types, variables, constants, etc.

· Embedded C is a generic term given to a programming language written in C, which is associated with a particular hardware architecture.

· Embedded C is an extension to the C language with some additional files.

The embedded system designers must know about the hardware architecture to write programs. These programs play a prominent role in monitoring and controlling external devices. They also directly operate and use the internal architecture of the microcontroller, such as interrupt handling, timers, serial communication and other available features.

Differences between C and Embedded C:



C programming	Embedded C programming
Possesses native development in nature.	Possesses cross development in nature.
Independent of hardware architecture.	Dependent on hardware architecture (microcontroller or other devices).
Used for Desktop applications, OS and PC memories.	Used for limited resources like RAM, ROM and I/O peripherals on embedded controller.

Table 1: Diff. b/w C and Embedded C

Advantages of embedded C program:

· It takes less time to develop an application program.

· It reduces complexity of the program.

· It is easy to verify and understand.

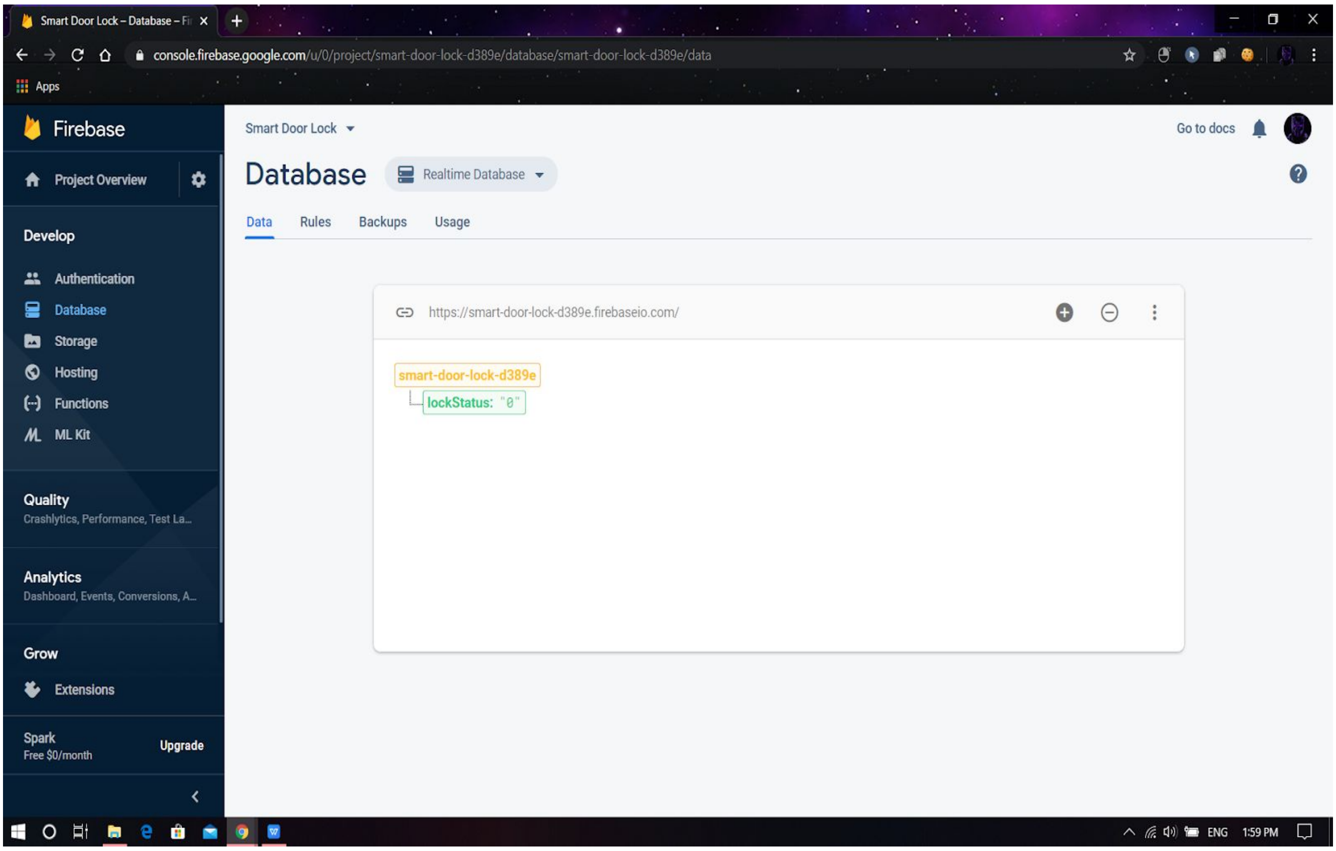
· It is portable in nature from one controller to another.

We hope that we have been successful in providing an easy and approachable way for the beginners of Embedded C programming. Better understanding of Embedded C programming is the most essential prerequisite for designing embedded based projects.

In addition to this, a better understanding and proper knowledge about embedded C programming help students immensely in the selection of a rewarding career.

3.3. FIREBASE:

Firebase is a Backend-as-a-Service (BaaS) that started as a YC11 start-up and grew up into a next-generation app-development platform on Google Cloud Platform.



https://smart-door-lock-d389e.firebaseio.com/

```
smart-door-lock-d389e
└ lockStatus: "0"
```

fig3. Firebase console (Database Section)

<u>So what is Firebase</u>?

· **Real-time Database** : Real-time Database is a cloud-hosted database with the data stored as JSON (JavaScript Object Notation). It provides the real-time synchronization between every connected client - any changes or update is automatically sent to all users. The offline mode does not cause data loss as the Database SDK stores new data to disk and when after the connection is re-established, the synchronization takes place. The application server is not required for the access to Database from mobile device or web browser.

· **File Storage :** Firebase Storage provides a simple way to save binary files most often images, but it could be anything to Google Cloud Storage directly from the client. Firebase Storage has its own system of security rules to protect your GCloud bucket from the masses, while granting detailed write privileges to your authenticated clients.

· **Authentication :** Authentication is a service for ensuring the protection of user's data using a password, phone number, or various social authentications. It includes very convenient SDKs (Software Development Kits), ready-to-use UI libraries, and backend services. There are two ways to sign in your users: FirebaseUI, a drop-in solution that can be easily customized to match your app design - or Firebase Authentication SDK, a solution for manual implementation of several methods of sign-in.

· **Hosting :** Hosting is a web content hosting for developers. It ensures fast and secure deployment of web applications to a global content-delivery network (CDN). Hosting provides release management and version review by using one-click rollbacks.

Firebase Advantages:

· Easy-to-use and well-documented.

· High integration capacities.

· High performance.

· Suitable for real-time applications.

· Wide functionality.

· Support for web, Android, and iOS.

· Simple control dashboard.

· Can replace any piece of backend software.

· High scalability.

Firebase Disadvantages:

· Limited free plan.

· Difficult execution of complex queries for data.

· Learning curve requires time.

· Different architectures and approaches to tools.

· Lack of optimized search and query functions.

3.4. ANDROID STUDIO:

Android is one of the most popular mobile device platforms. The Android platform allows developers to write managed code using Java (http://www.developer.com/java) to manage and control the Android device. Android Studio is a popular IDE developed by Google for developing applications that are targeted at the Android platform. Note that Android Studio has replaced Eclipse as the IDE of choice for developing Android applications.

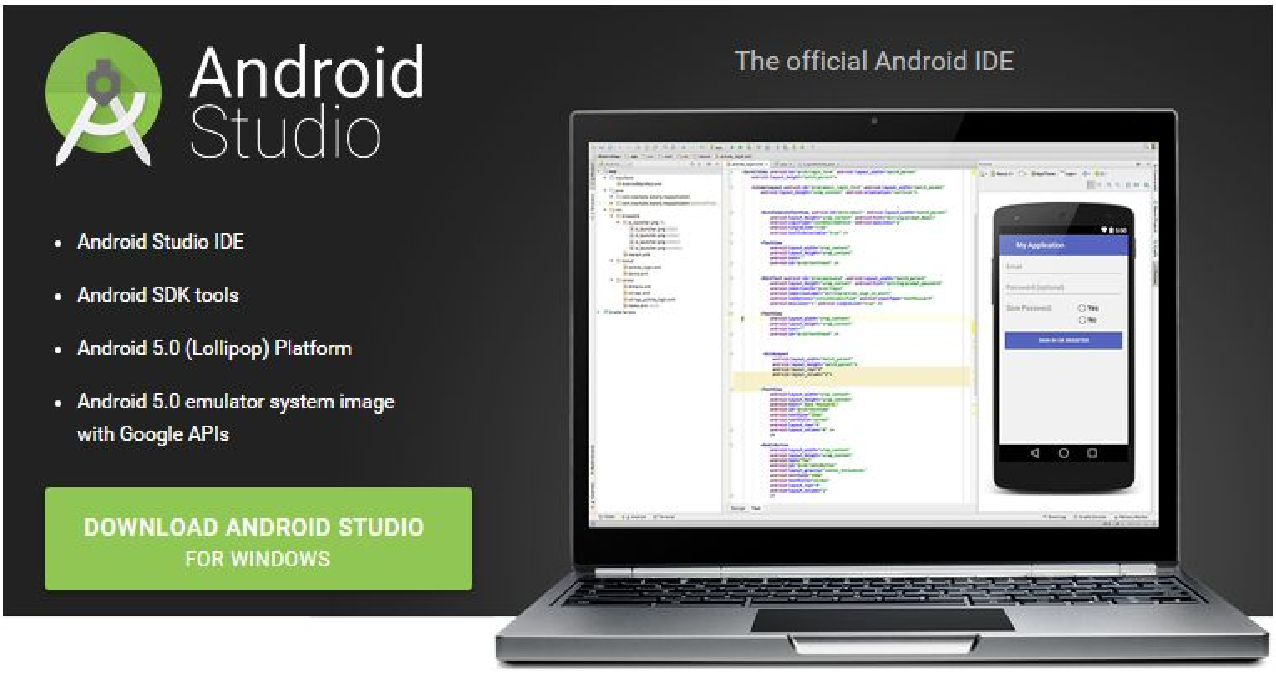

Android Studio
The official Android IDE
Android Studio IDE
Android SDK tools
Android 5.0 (Lollipop) Platform
Android 5.0 emulator system image with Google APIs
DOWNLOAD ANDROID STUDIO FOR WINDOWS

fig 4. Android Studio(Application Development)

<u>The following features are provided in the current stable version</u>:

1· Gradle-based build support

2· Android-specific refactoring and quick fixes

3· Lint tools to catch performance, usability, version compatibility and other problems

4· ProGuard integration and app-signing capabilities

5· Template-based wizards to create common Android designs and components

6· A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.

7· Support for building Android Wear apps.

8· Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

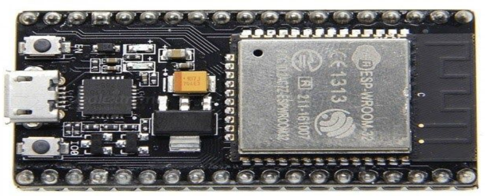
9· Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

CHAPTER 4 HARDWARE REQUIREMENT

4.1. NODE MCU ESP32:

The NodeMCU ESP-32S is one of the development boards created by NodeMCU to evaluate the ESP-WROOM-32 module. It is based on the ESP32 microcontroller that boasts Wifi, Bluetooth, Ethernet and Low Power support all in a single chip.



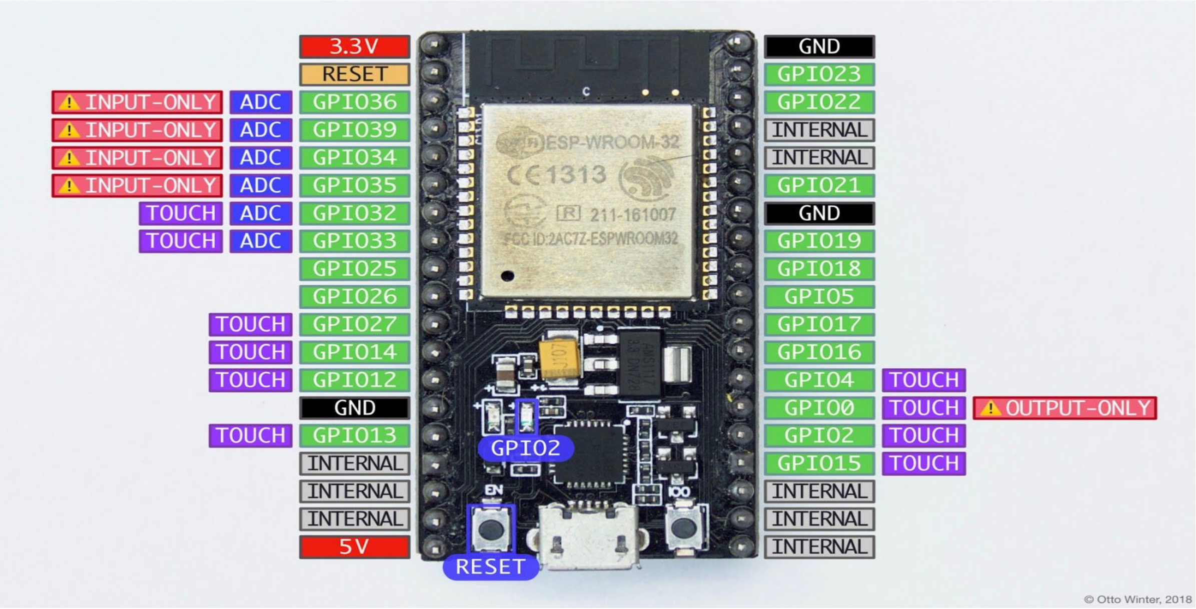


fig 5.1-Node Mcu ESP32 , fig 5.2 - Block Diagram

Device Summary :

- Microcontroller: Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 28
- Analog Input Pins (ADC): 8
- Analog Outputs Pins (DAC): 2
- UARTs: 3
- SPIs: 2
- I2Cs: 3
- Flash Memory: 4 MB
- SRAM: 520 KB
- Clock Speed: 240 Mhz
- Wi-Fi: IEEE 802.11 b/g/n/e/i:

○ Integrated TR switch, balun, LNA, power amplifier and matching network

○ WEP or WPA/WPA2 authentication, or open networks

POWER :

Power to the NodeMCU ESP-32S is supplied via the on-board USB Micro B connector or directly via the "VIN" pin. The power source is selected automatically.

The device can operate on an external supply of 6 to 20 volts. If using more than 12V, the voltage regulator may overheat and damage the device. The recommended range is 7 to 12 volts.

4.2. RELAY 1 CHANNEL:

A relay is an electrically operated device. It has a control system and (also called input circuit or input contactor) and controlled system (also called output circuit or output contactor). It is frequently used in automatic control circuits. To put it simply, it is an automatic switch to controlling a high-current circuit with a low-current signal.

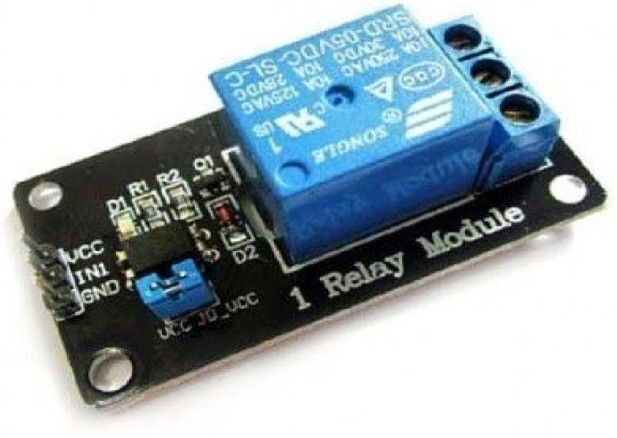


fig 6. 1-Channel Relay Module

ADVANTAGES :

The advantages of a relay lie in its lower inertia of the moving, stability, long-term reliability and small volume. It is widely adopted in devices of power protection, automation technology, sport, remote control, reconnaissance and communication, as well as in devices of electromechanics and power electronics. Generally speaking, a relay contains an induction part which can reflect input variables like current, voltage, power, resistance, frequency, temperature, pressure, speed and light etc. It also contains an actuator module (output) which can energize or de-energize the connection of controlled circuit. There is an intermediary part between input part and output part that is used to coupling and isolate input current, as well as actuate the output. When the rated value of input (voltage, current and temperature etc.) is above the critical value, the controlled output circuit of the relay will be energized or de-energized.

NB: input into a relay can be divided into two categories: electrical quantities (including current, voltage, frequency, power etc.) and non- electrical quantities(including temperature, pressure, speed, etc.)

FEATURES :

1. Good in safety. In power systems and high voltage systems, the lower current can control the higher one.
2. 1-channel high voltage system output, meeting the needs of single channel control
3. Wide range of controllable voltage.
4. Being able to control high load current, which can reach 240V, 10A.
5. With a normally-open (NO) contact and a normally-closed (NC) contact.

PINS OUT :

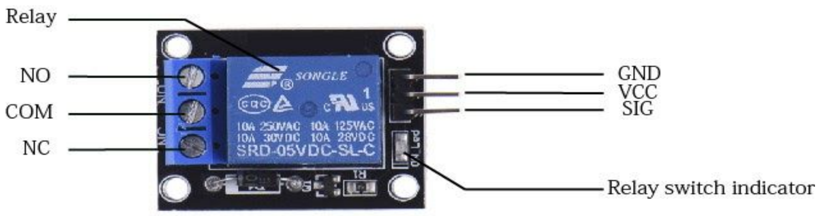


fig 7. Pin Diagram of Relay Module

INPUT :

It has a 1×3 (2.54mm pitch) pin header for connecting power (5V and 0V), and for controlling the relay. The pins are marked on the PCB:

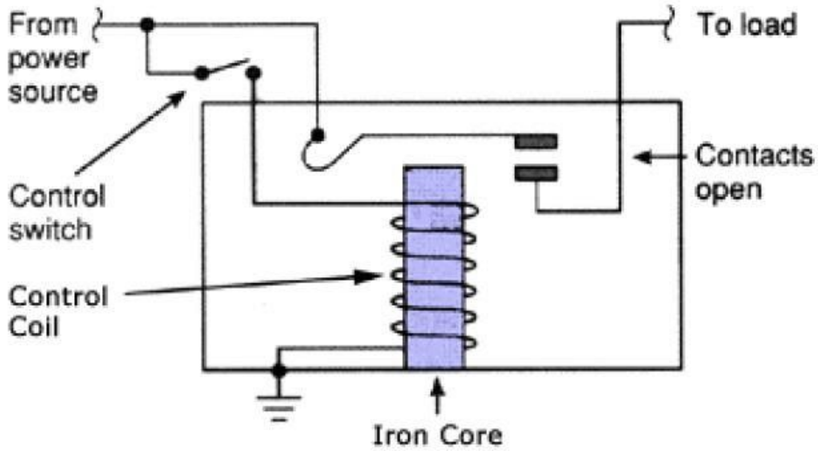
- GND – Connect 0V to this pin.
- SIG – Controls this relay, active Low! Relay will turn on when this input goes below about 2.0V
- VCC – Connect 5V to this pin. Is used to power the optocouplers.

OUTPUT :

The 1 channel relay module could be considered like a series switches: 1 normally Open (NO), 1 normally closed (NC) and 1 common Pins (COM).

- COM- Common pin.
- NC- Normally Closed, in which case NC is connected with COM when INT1 is set low and disconnected when INT1 is high.
- NO- Normally Open, in which case NO is disconnected with COM1 when INT1 is set low and connected when INT1 is high.

HOW RELAY WORKS ? The working of a relay can be better understood by explaining the following diagram given below.



There are 5 parts in every relay:

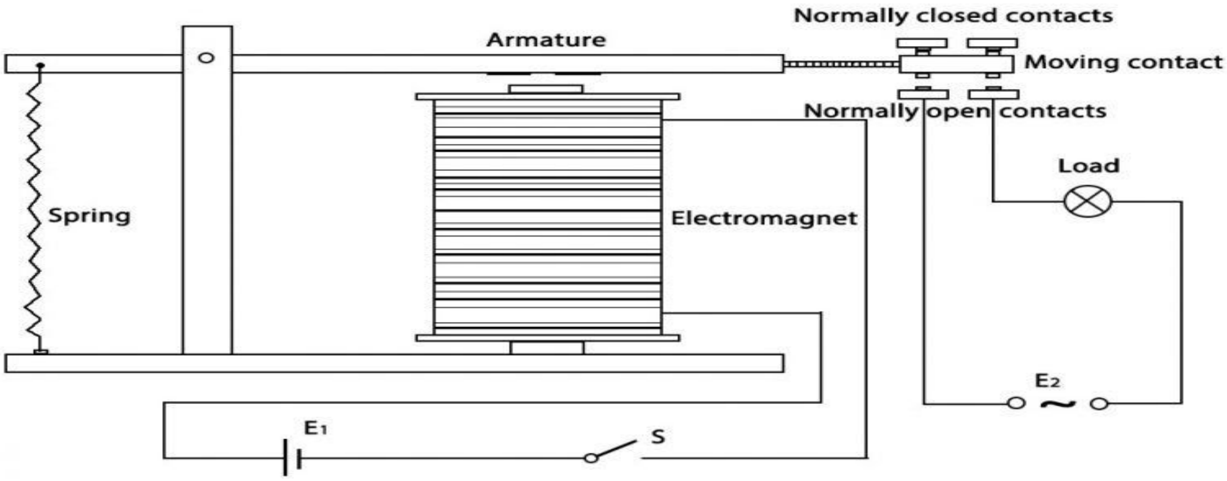


fig 8. Architecture of Relay Module

1. **Electromagnet** – It consists of an iron core wounded by coil of wires. When electricity is passed through, it becomes magnetic. Therefore, it is called electromagnet.

2. **Armature** – The movable magnetic strip is known as armature. When current flows through them, the coil is energized thus producing a magnetic field which is used to make or break the normally open (N/O) or normally close (N/C) points. And the armature can be moved with direct current (DC) as well as alternating current (AC).

3. **Spring** – When no currents flow through the coil on the electromagnet, the spring pulls the armature away so the circuit cannot be completed.

4. **Set of electrical contacts** – There are two contact points:

. *Normally open* – connected when the relay is activated, and disconnected when it is inactive.

. *Normally close* – not connected when the relay is activated, and connected when it is inactive.

5. *Molded frame* – Relays are covered with plastic for protection.

Principle :

The diagram shows an inner section diagram of a relay. An iron core is surrounded by a control coil. As shown, the power source is given to the electromagnet through a control switch and through contacts to the load. When current starts flowing through the control coil, the electromagnet starts energizing and thus intensifies the magnetic field. Thus the upper contact arm starts to be attracted to the lower fixed arm and thus closes the contacts causing a short circuit for the power to the load. On the other hand, if the relay was already de-energized when the contacts were closed, then the contact move oppositely and makes an open circuit.

As soon as the coil current is off, the movable armature will be returned by a force back to its initial position. This force will be almost equal to half the strength of the magnetic force. This force is mainly provided by two factors. They are the spring and also gravity.

Relays are mainly made for two basic operations. One is low voltage application and the other is high voltage. For low voltage applications, more preference will be given to reduce the noise of the whole circuit. For high voltage applications, they are mainly designed to reduce a phenomenon called arcing.

RELAY APPLICATIONS :

Relays are used to protect the electrical system and to minimize the damage to the equipment connected in the system due to over currents/voltages. The relay is used for the purpose of protection of the equipment connected with it.

These are used to control the high voltage circuit with low voltage signal in applications audio amplifiers and some types of modems.

These are used to control a high current circuit by a low current signal in the applications like starter solenoid in automobiles. These can detect and isolate the faults that occurred in power transmission and distribution systems. Typical application areas of the relays include :

- Lighting control systems
- Telecommunication
- Industrial process controllers
- Traffic control
- Motor drives control
- Protection systems of electrical power system
- Computer interfaces
- Automotive
- Home appliances

4.3. MOBILE FINGERPRINT SCANNER:

Fingerprint scanners are security systems of biometrics. They are used to unlock doors and in other security applications. During the 2010s fingerprint scanners became commonplace on mobile phones.

People have a pattern of ridges on their fingers. This fingerprint cannot be removed or changed. Every fingerprint is different from any other in the world. Because there are countless combinations, fingerprints are much used for identification.

fig 9. Mobile Fingerprint Scanner

Various Types

1. Optical reflexive

They are based on the oldest technique, that consists of placing your finger on a glass surface or a prism that is illuminated by a led diode. When the ridges of fingerprints touch the surface, the light is absorbed, while between these crests a total reflection occurs. The resulting light and dark areas are recorded in an image sensor.

In practice there are some difficulties with this technique: wet and dry fingers make very different images, and the system is sensitive to dust and dirt on the surface. The unit is large and expensive. This system is also easy to cheat and if the skin is damaged, the fingerprint is not recognized correctly. The recognition of the fingerprint of older people is also difficult to do because their skin may not be elastic enough. In some circumstances this may result in false recognition. If the stored fingerprint was taken with less pressure, false acceptances may occur.

2. Capacitive

The sensor is a silicon integrated circuit with a surface that is covered by a large number of transducer elements (or pixels), with a typical resolution of 500 dpi. Each element contains two adjacent metal electrodes. The capacitance between the electrodes, which forms a feedback path for an inverting amplifier, is reduced when the finger is applied to said surface: it is further reduced when it detects ridges and less reduced when it detects the space between them.

The sensor is susceptible to electrostatic discharge. These sensors only work with normal healthy skins, they are not operative being used on skins with hard areas, calluses or scars. Moisture, grease or dust can also affect its operation.

3. Mechanical

It consists of tens of thousands of tiny pressure transducers that are mounted on the surface of the sensor. An alternative design uses switches that are closed when pressed by a ridge, but that stay open when they are under a valley. This only provides one bit of information per pixel, instead of working with a gray scale.

USE

The most important use area is in access control for computers. This is especially important for laptops and PDAs. Thanks to the price falling, more and more devices are equipped with sensors. Other devices with built-in fingerprint sensors include USB hard drives, USB memory modules and card readers. They are also available in mice and keyboards.

Sensors are increasingly used to secure financial transactions and change machines for "online" banking. In the future, the owner's fingerprint will be stored securely on identity cards and credit cards and may also be used for authentication of emails using digital signatures.

Direct physical access to rooms and devices can also be ensured by coupling fingerprint sensors with door opening systems.

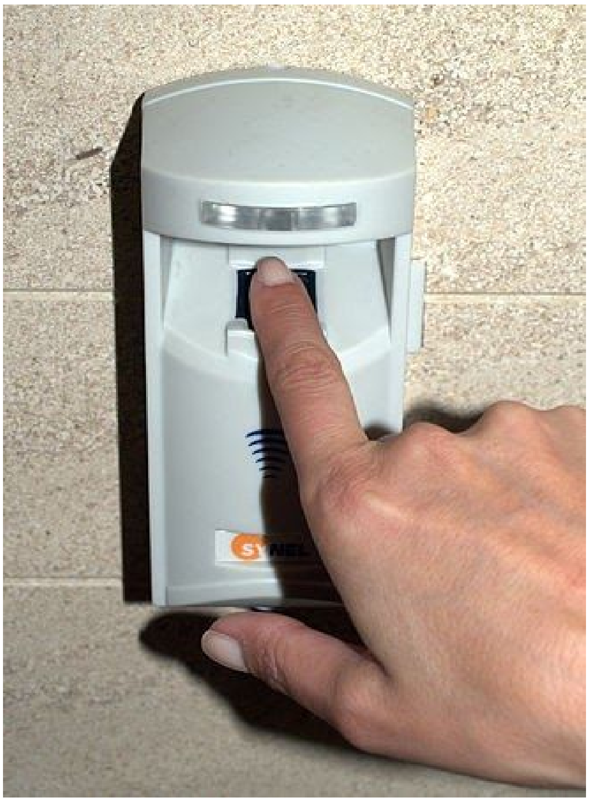


fig 10. Fingerprint Scanner

4.4. SOLENOID LOCK:

The **solenoid lock** denotes a latch for electrical locking and unlocking. It is available in unlocking in the power-on mode type, and locking and keeping in the power-on mode type, which can be used selectively for situations. The power-on unlocking type enables unlocking only while the solenoid is powered on.

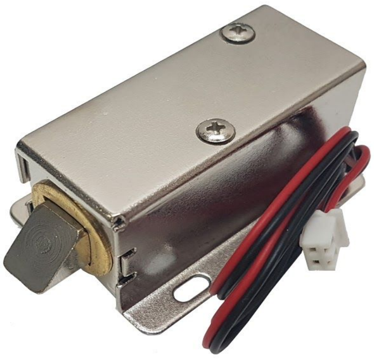


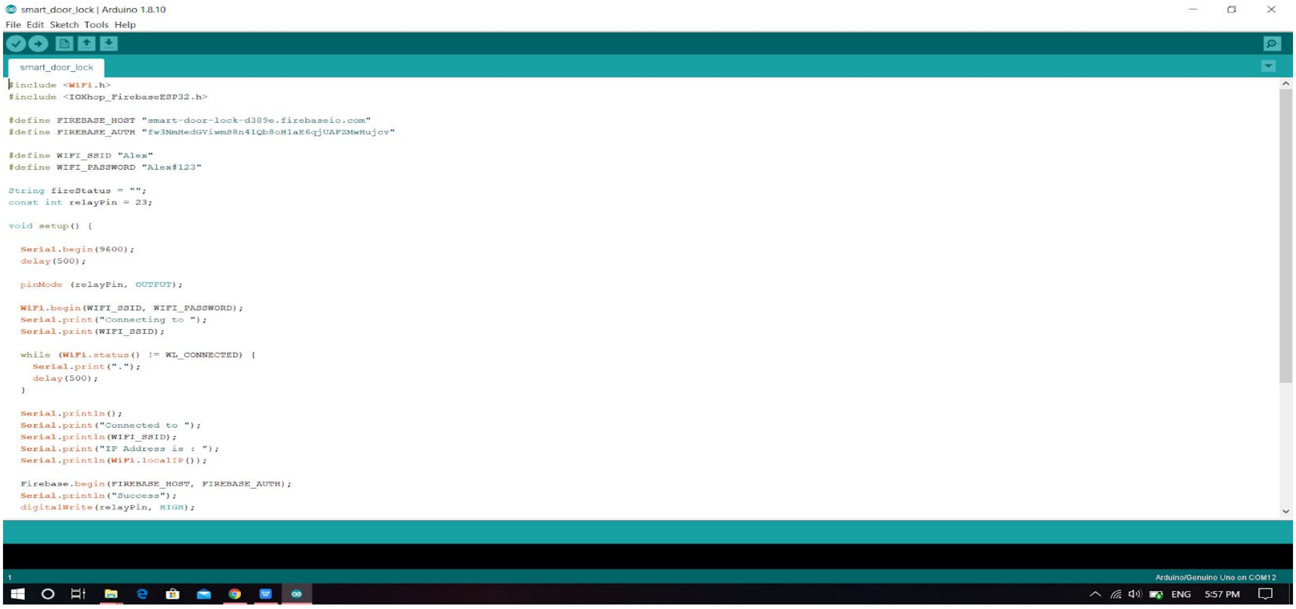
fig 11. Solenoid Lock

12V Solenoid lock has a slug with a slanted cut and a good mounting bracket. It's basically an electronic lock, designed for a basic cabinet, safe or door. When 9-12VDC is applied, the slug pulls in so it doesn't stick out and the door can be opened. It does not use any power in this state. It is very easy to install automatic door lock systems like electric door locks with the mounting board. This solenoid in particular is nice and strong.

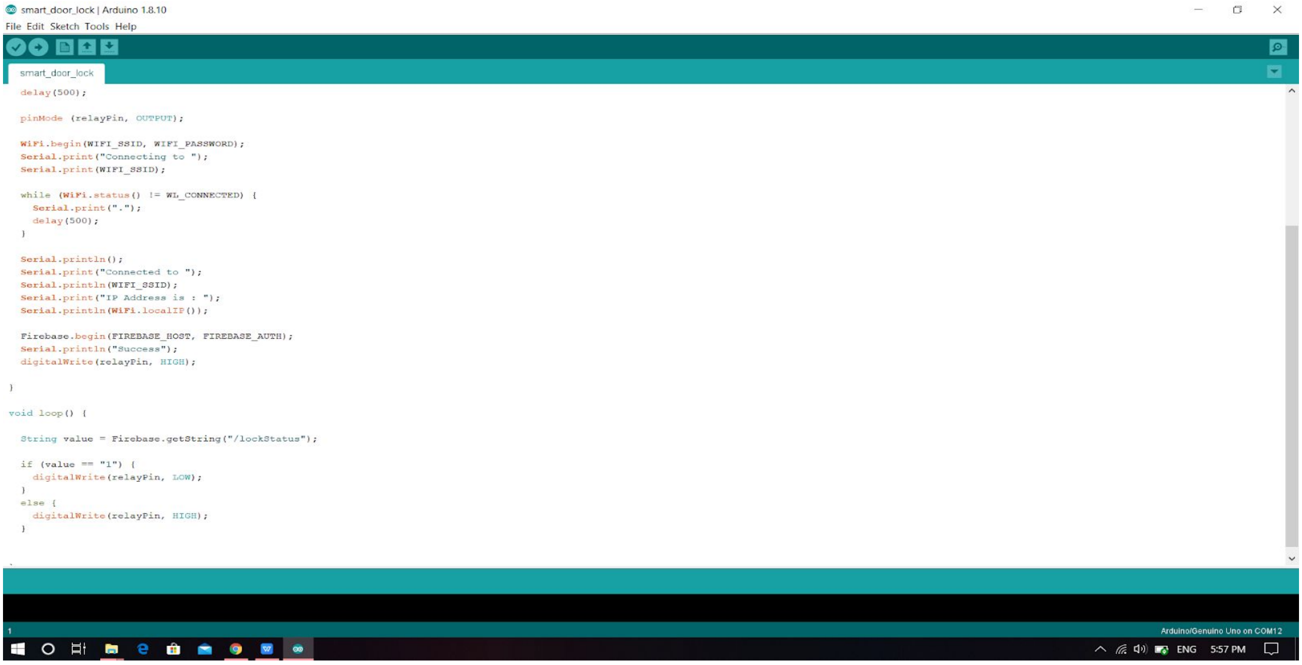
Specifications of 12V solenoid lock:

- Operating voltage : 12VDC
- Draws 650mA at 12V, 500 mA at 9V when activated
- Designed for 1-10 seconds long activation time
- Wire length: 222.25mm

CHAPTER 5 APPENDICES

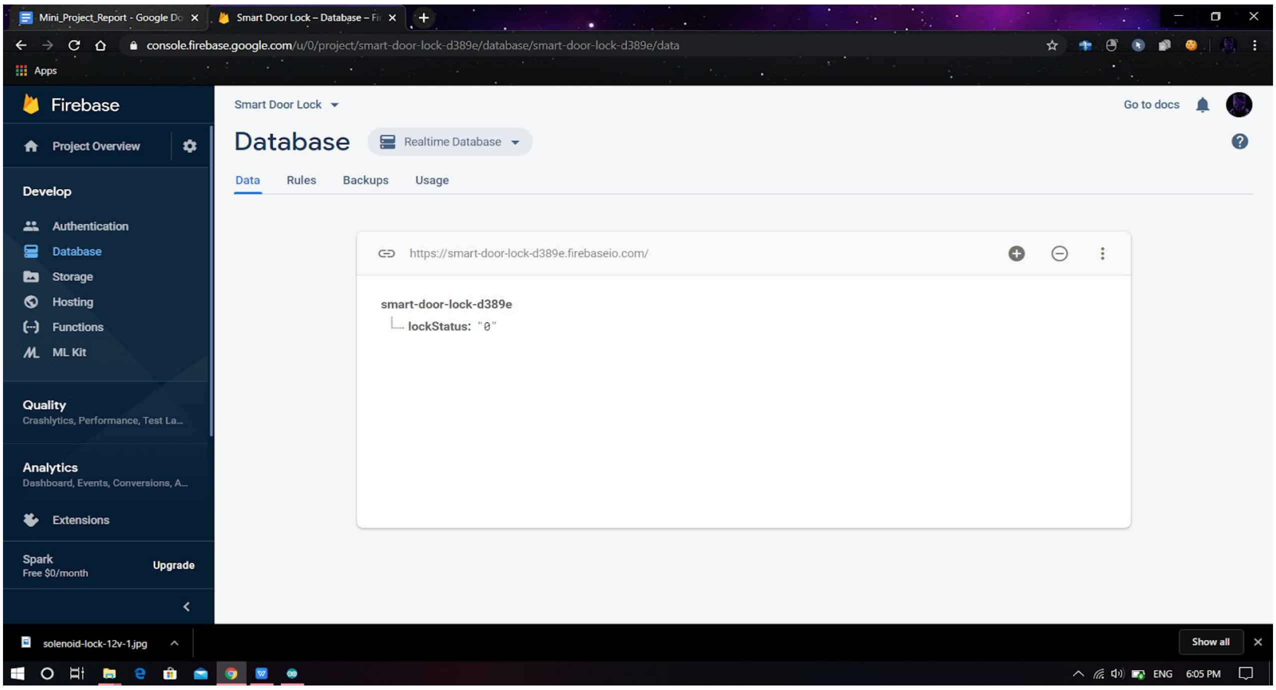

smart_door_lock | Arduino 1.8.10
File Edit Sketch Tools Help
smart_door_lock
1
Arduino/Genuino Uno on COM12
ENG
5:57 PM

```
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>

#define FIREBASE_HOST "smart-door-lock-d389e.firebaseio.com"
#define FIREBASE_AUTH "fw3NmHedGViwmS8n41Qb8oH1aE6qjUAFZMwHujcv"

#define WIFI_SSID "Alex"
#define WIFI_PASSWORD "Alex#123"

String fireStatus = "";
const int relayPin = 23;

void setup() {

  Serial.begin(9600);
  delay(500);

  pinMode (relayPin, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial.println("Success");
  digitalWrite(relayPin, HIGH);
```


smart_door_lock | Arduino 1.8.10
File Edit Sketch Tools Help
smart_door_lock
1
Arduino/Genuino Uno on COM12
ENG
5:57 PM

```
  delay(500);

  pinMode (relayPin, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial.println("Success");
  digitalWrite(relayPin, HIGH);

}

void loop() {

  String value = Firebase.getString("/lockStatus");

  if (value == "1") {
    digitalWrite(relayPin, LOW);
  }
  else {
    digitalWrite(relayPin, HIGH);
  }

}
```

fig 12. Node MCU ESP32 Code



console.firebase.google.com/u/0/project/smart-door-lock-d389e/database/smart-door-lock-d389e/data

https://smart-door-lock-d389e.firebaseio.com/

```
smart-door-lock-d389e
└ lockStatus: "0"
```

fig 13. Firebase Console Realtime Database

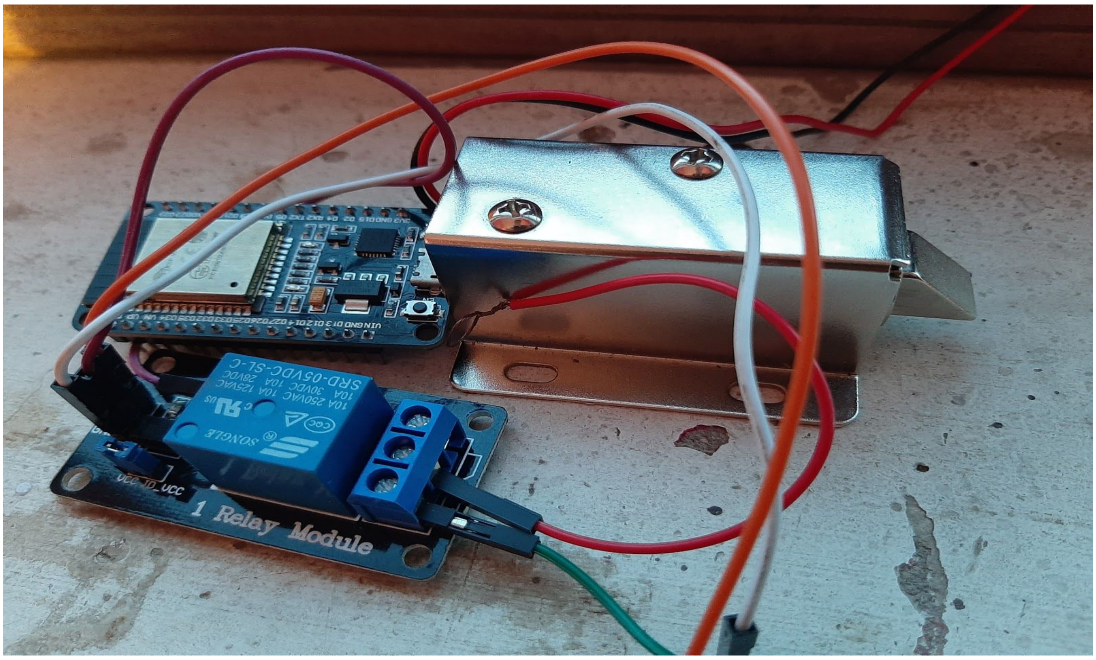


fig 14. Lock Status (When value = “0” then closed, When value = “1” then open)

CHAPTER 6 REFERENCES

- https://www.nodemcu.com/index_en.html
- https://www.arduino.cc/en/Main/Software
- https://docs.zerynth.com/latest/official/board.zerynth.nodemcu_esp32/docs/index.html
- https://en.wikipedia.org/wiki/Fingerprint_scanner
- https://en.wikipedia.org/wiki/Electromagnetic_lock
- https://randomnerdtutorials.com/guide-for-relay-module
- https://www.amazon.in/
- https://www.flipkart.com/
- https://www.google.com/