1. **INTRODUCTION**

**1.1 Introduction**

The Basic requirement of security can be achieved by designing electrical or mechanical locks that are designed with one or a few keys, but for locking a big area many locks are required. [Electronic locking systems](http://www.elprocus.com/password-based-electronic-lock-system/) are preferable over mechanical locks, to resolve the security problems that are associated with the mechanical locks.

Nowadays every device’s operation is based on digital technology. For instance, token-based digital identity devices, Fort-token mobile and digital-based door lock systems for [auto door opening or closing](http://www.edgefxkits.com/atmega-based-garage-door-opening) are all based on digital technology. These locking systems are used to control the movement of the door and are functional without requiring a key to lock or unlock the door. These locking systems are controlled by a keypad and are installed at the side hedge of the door.

This Password based security system provides the same features and serves as a helping hand to people in their day-to-day lives by providing enhanced security. The password is set by the user and is known to only the authorized people and hence the system or the house or the premises are protected. If any unauthorized person tries to access the premises then an alert message will be sent to the registered number in the form of email as well as message notification in the App. In this way, if an intruder is trying to do any inappropriate activity the user will be notified about this and any wrong activities can be prevented from happening.

**1.2 Motivation**

The main motivation behind creating this project was to provide personalized security to people at a cheaper cost and easier way to operate. Security has always been a major concern and in order to maintain that different security measures are implemented.This project has considered that and created a secure access for a door which needs a password to open the door. Automated systems have less manual operations, so that the flexibility, reliabilities are high and accurate. Hence every field prefers automated control systems, especially in the field of electronics. But in our day to day lives, it becomes important to have our own personalized security system which can always help us to monitor any suspicious activities and always have a view on our house or systems.

The Aims and Objectives for Password Based Security System will be seen in the next section.

**1.3 Aim & Objectives**

**Aim:**

*The aim of this project is to provide enhanced security by using password for authentication of the user and thus provide access only to the persons who are authorized in that premises or house or system.The goal of the project is to develop a unique system through mobile technology which can control various units of the houses, industries, and also provides a security system.*

**Objectives:**

Following are the main objectives of Password Based Security System:

* Increase the *security level* to prevent an unauthorized unlocking of the door.
* Give the *flexibility* to the user to change or reset the password in case the user forgets that combination.
* Lock the door by using *password.*
* To give user more secure yet *cost-efficient* way of door locking system.
* Make the device as *affordable* as possible so that the technology reaches maximum households and saves maximum security threats.
* Employ the *power of IoT* to connect the device to the internet so that alerts can be conveyed to the user at any given location and at any given point in time.
* Make use of multiple IoT platforms (*Blynk*) to eliminate the possibility of failure due to platform dependency.
* Make the device *User friendly* and hassle *proof.*

The following chapter deals with the review of Literature.

1. **REVIEW OF LITERATURE**

**2.1 Arduino UNO**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of the Arduino Software. The Uno board is the first in a series of USB-based Arduino boards, and it and version 1.0 of the Arduino IDE were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes pre programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

**2.2 Arduino NodeMCU**

NodeMCU is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform. It includes [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from Espressif Systems, and hardware which is based on the ESP-12 module. The term *NodeMCU* by default refers to the firmware rather than the development kits. The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)) scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1).

*In our Password Based Security System we are using NodeMCU.*

**2.3 Raspberry Pi**

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) or cases. However, some accessories have been included in several official and unofficial bundles.

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

According to the Raspberry Pi Foundation, more than 5 million Raspberry Pis were sold by February 2015, making it the best-selling British computer. By November 2016 they had sold 11 million units, and 12.5m by March 2017, making it the third best-selling *general purpose computer*. In July 2017, sales reached nearly 15 million. In March 2018, sales reached 19 million.

**2.4 Intel Galileo**

*Intel Galileo* is the first in a line of [Arduino](https://en.wikipedia.org/wiki/Arduino)-certified development boards based on [Intel](https://en.wikipedia.org/wiki/Intel) x86 architecture and is designed for the maker and education communities. Intel released two versions of Galileo, referred to as Gen 1 and Gen 2. These development boards are sometimes called *Breakout boards*.

The board was discontinued on June 19, 2017.

Intel Galileo combines Intel technology with support for Arduino ready-made hardware expansion cards (called "shields") and the Arduino software development environment and libraries.[6] The development board runs an open source Linux operating system with the Arduino software libraries, enabling reuse of existing software, called "sketches". The sketch runs every time the board is powered. Intel Galileo can be programmed through OS X, Microsoft Windows and Linux host operating software. The board is also designed to be hardware and software compatible with the Arduino shield ecosystem.

**2.5 Real world Problems**

In today’s world, where security has become so important it is necessary to have a personalized security system for better overview of our own house or premises or system. There are many security systems available but in these we don’t get the privilege to set the protection as per our needs and convenience. Hence, a personalized security system is required where the user can control, access and monitor according to us.

Following are the different requirements for different real world problems:

* High Security Areas:

**Password based security system** can be modified to **Fingerprint based security system** at many places like Industries, Offices, and Colleges or even at our home. Using Fingerprint will be a fine combination of “Biometrics technology” and “Embedded system technology”. Fingerprint sensor is the main part of the system. It makes use of Biometric sensor to detect fingerprint. It is also called as Biometric sensor. Fingerprint sensor uses various types of techniques like ultrasonic method, optical method or thermal technique. The use of fingerprint scanner makes the security system more secure.The main feature or specialty of fingerprint is that it is unique. It gives this project the high level security than other security systems. Person recognition using the Fingerprint identification is used since a long time. Most common example is used in criminal cases.

How does the Fingerprint Scanner work?

Fingerprint is a pattern made up of *ridges and valleys* on our fingertip skin. While storing the entry in database, scanner takes an image of these patterns and stores in its own memory. Then while performing search operation, it again takes pattern of fingerprint of that user who needs to gain access. This pattern is compared with all patterns previously stored in memory. In short, it performs a task which is related to Digital image processing. It performs various iterations and executes matching algorithms and if it finds exact match then it gives out fingerprint ID number. Otherwise it gives out error signal.

* Medium Security Areas:

**Password based security system** can be modified to **Camera based security system.** A PIR sensor is used to detect the presence of any person and a Pi Camera is used to capture the images when the presence it detected.Whenever anyone or intruder comes in range of PIR sensor, PIR Sensor triggers the Pi Camera through Raspberry Pi.

How does the PIR Sensor work?

**PIR sensors are** sensitive to infrared radiation so they detect body heat and

notice abrupt changes in temperature. Active **motion sensors** send out a signal.

Raspberry pi sends commands to Pi camera to click the picture and save it. After it, Raspberry Pi creates a mail and sends it to the defined mail address with recently clicked images. The mail contains a message and picture of intruder as attachment.

**2.2 Existing Solutions**

There are many security systems are available in the market and different types of security systems can be used to maintain security.

Some of the existing solutions to security systems are:

* There is an access control system keypad lock available on Eassycart

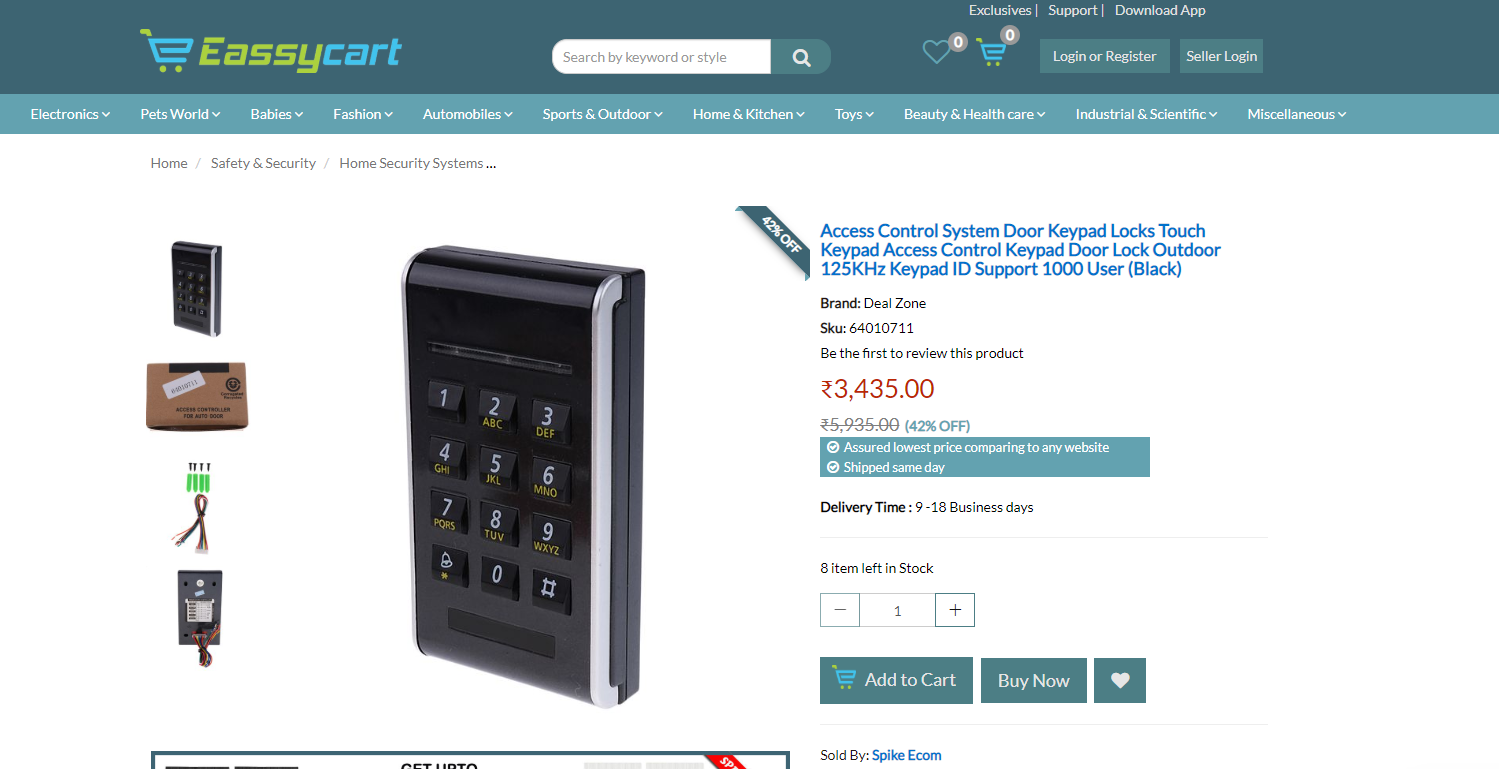


Fig. 2.1

However this is a simple password based system that unlocks the door without the status of the door being known. There is no functionality of a remote alert email or smartphone notification. Our device aims at including all this functionality at the cost offered for this product.

* Another Security system available is of Godrej

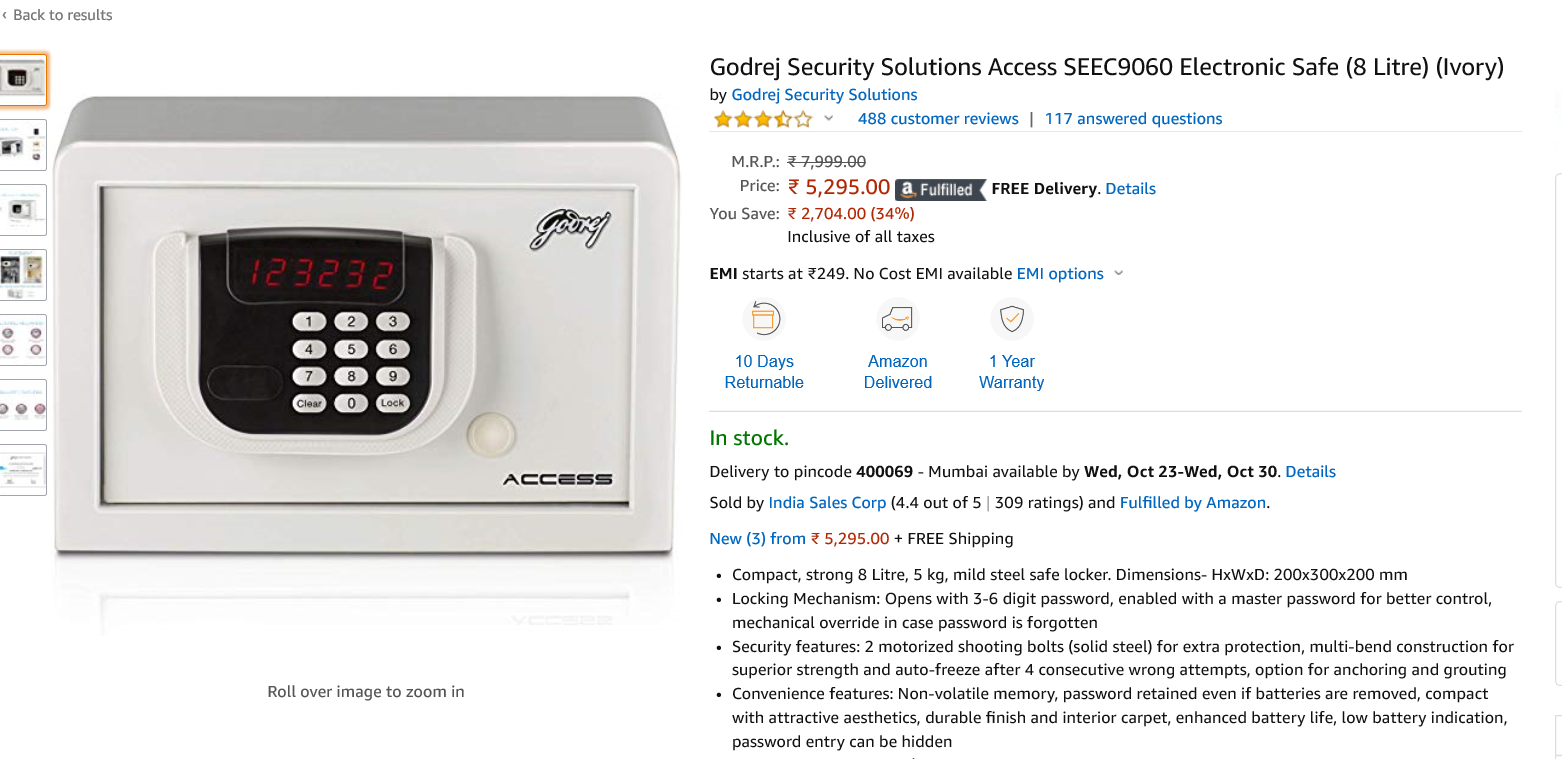


Fig. 2.2

This system is expensive as compared to ours. Our password based security system is a cheaper alternative. Some systems/products have keypad security but the do not have wrong password alert system and live tracking of the status of the system.

**2.3 Hardware & Software Requirements with Cost**

The hardware components required to build this project are as follows:

* Arduino NodeMCU
* Keypad 4x3
* Jumper Wires
* Breadboard
* LEDs

The software used for this project are:

* Arduino IDE
* Blynk app for alert messages and dynamic display of input

**Hardware:**

**NodeMCU**:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

**NodeMCU** is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform.It includes [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)) scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1). NodeMCU was created shortly after the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) came out.



Fig. 2.3

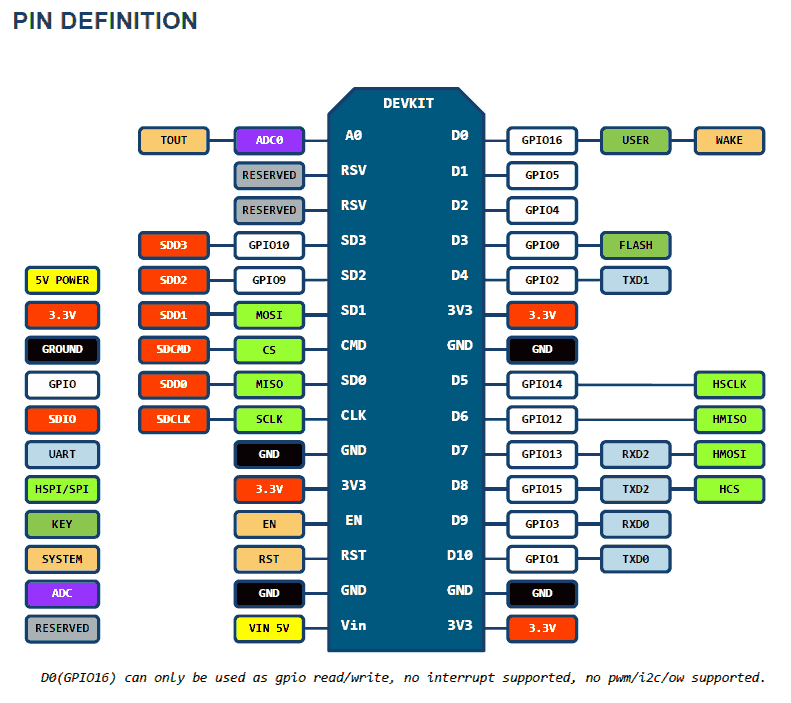
**Pin Diagram Of NodeMCU:** ****

Fig. 2.4

**Keypad**:

This **Matrix Keypad 4 X 3** has 12 buttons, arranged in a telephone-line 3x4 grid. The keys are connected into a matrix,hence only 7 microcontroller pins (3-columns and 4-rows) are required to scan the pad.

### Specifications of Matrix Keypad 4 X 3:

* Flexible
* Can be attached on any surface
* Only 7 microcontroller pins are needed to access 12 buttons

### Dimensions:

* Size: 70mm x 77mm x 1mm
* Weight: 7.5gm

Keypad is a library for using *matrix* style keypads with the Arduino. As of version 3.0 it now supports multiple keypresses.

It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user.Keypad library is part of the [Hardware Abstraction](https://playground.arduino.cc/Code/HardwareAbstraction/) libraries.Version 3.0 has just been posted (19 July 2012) and was rewritten to support multi-key presses by default. But for those who still need the original single-keypress functionality, the library is fully backwards compatible.You won't need external resistors or diodes because the library uses the internal pullup resistors and additionally ensures that all unused column pins are high-impedance.

Fig. 2.5

**Jumper Wires:**

****

Fig. 2.6

A jump wire (also known as jumper wire, or jumper) is an [electrical wire](https://en.wikipedia.org/wiki/Electrical_wire), or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a [breadboard](https://en.wikipedia.org/wiki/Breadboard) or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the [header connector](https://en.wikipedia.org/wiki/Pin_header#Header_connector) of a circuit board, or a piece of test equipment.

**Breadboard:**

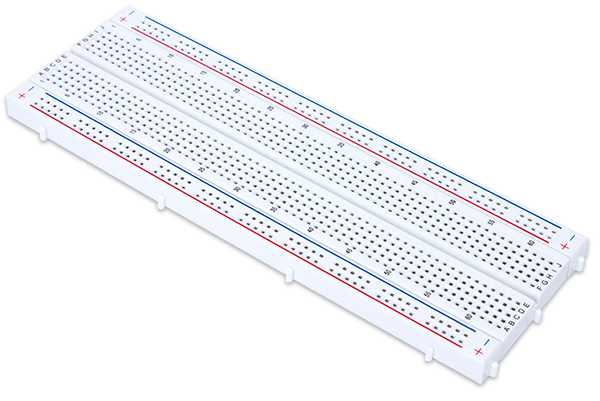
****

Fig. 2.7

A breadboard is a construction base for [prototyping](https://en.wikipedia.org/wiki/Prototype) of [electronics](https://en.wikipedia.org/wiki/Electronic_circuit). Originally the word referred to a literal bread board, a polished piece of wood used for slicing breadIn the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require [soldering](https://en.wikipedia.org/wiki/Soldering), it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design.

**SOFTWARE:**

**Arduino IDE :**

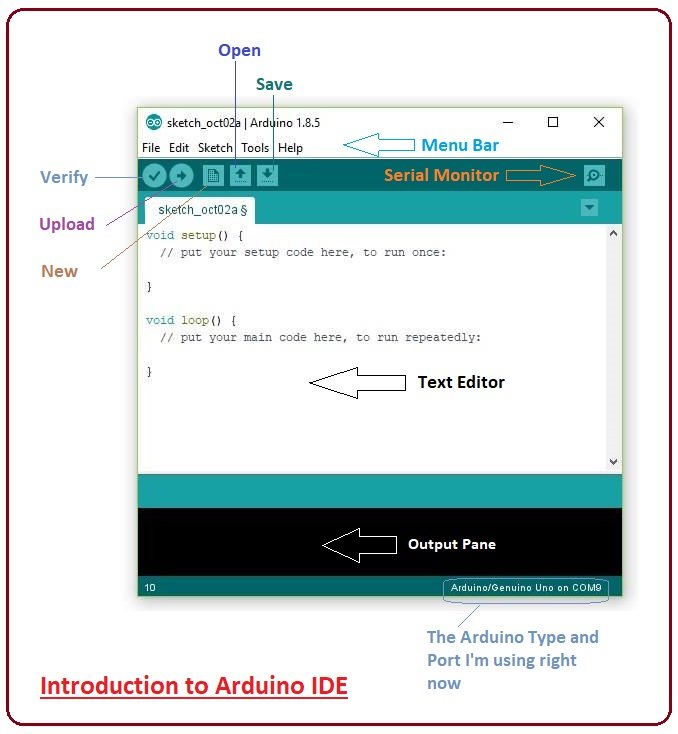
****

Fig. 2.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.This software can be used with any Arduino board.

The [Arduino](https://en.wikipedia.org/wiki/Arduino) integrated development environment ([IDE](https://en.wikipedia.org/wiki/Integrated_development_environment)) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). 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.

The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2. The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring.The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) 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](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution

**Blynk App:**

Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It’s a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

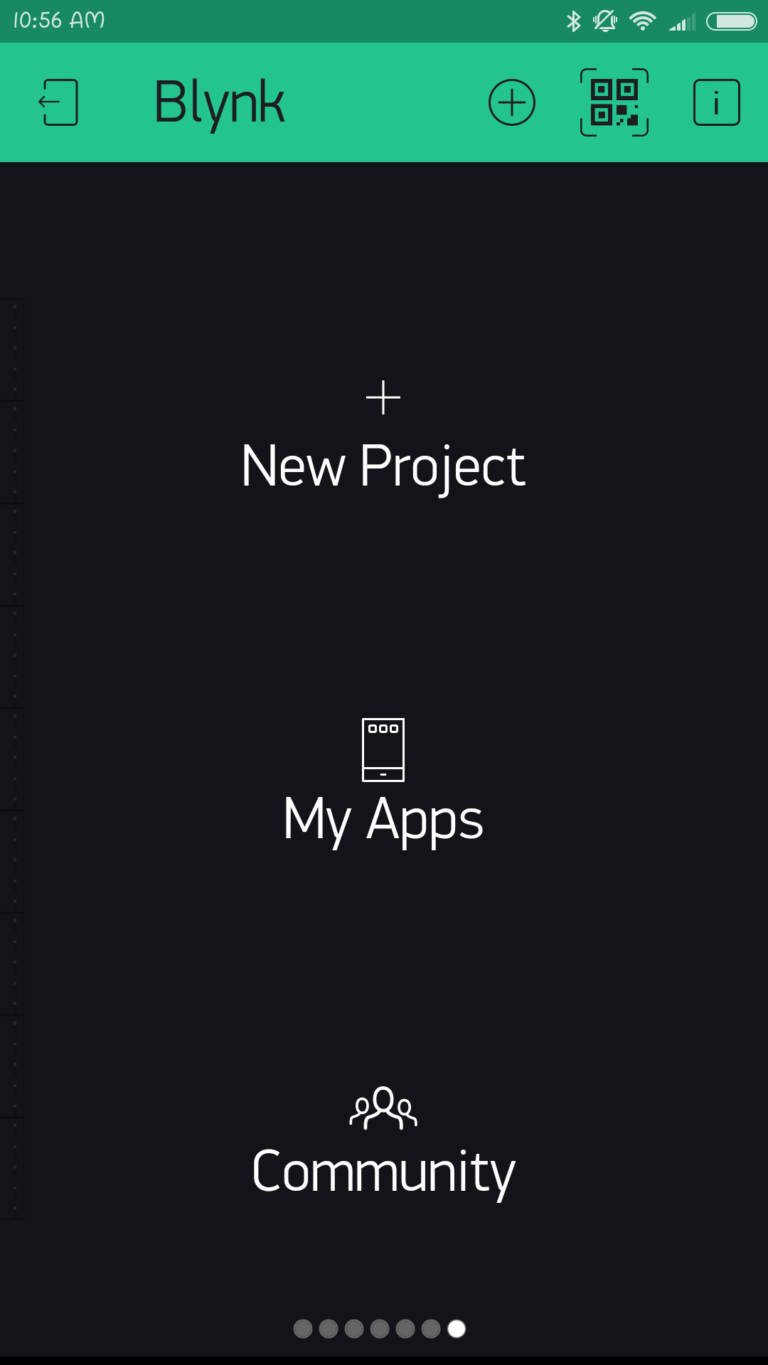


Fig. 2.9

Blynk application can be found from the following links –

[1. Android Blynk App](https://play.google.com/store/apps/details?id=cc.blynk)

[2. IOS Blynk App](https://itunes.apple.com/us/app/blynk-control-arduino-raspberry/id808760481?ls=1&mt=8)

You’ll also need to install the Blynk Arduino Library, which helps generate the firmware running on your ESP8266.

**Cost Analysis:**

**Hardware:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr No** | **Components** | **Rate(Rs/piece)** | **Quantity** | **Cost(Rs)** |
| 1 | Arduino NodeMCU | 350 | 1 | 350 |
| 2 | Keypad | 85 | 1 | 85 |
| 3 | Jumper wires | 70 | 1 | 70 |
| 4 | LEDs | 2 | 2 | 4 |
| 5 | Breadboard | 65 | 1 | 65 |

**Total : Rs. 574 /-**

**Software:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr No** | **Components** | **Rate(Rs/piece)** | **Quantity** | **Cost(Rs)** |
| 1 | Arduino IDE | FREE | 1 | FREE |
| 2 | Blynk App | FREE | 1 | FREE |

**Total : FREE**

The next chapter deals with the design and implementation details of Password Based Security System.

1. **DESIGN AND IMPLEMENTATION**

**3.1 Design Consideration**

Connecting products to the Internet of Things (IoT) is essential to manufacturers looking to stay competitive within their industry. Adding IoT capabilities gives consumers more features. When designing your first IoT device, there are 10 things to keep in mind:

1. Cost

“Smart” or IoT products help consumers and manufacturers alike, but they cost more. Both Ethernet and wireless technologies have come down below $10, so consider networking in your next product.

2. Network

The network technology you chose for your IoT product has distance and gateway/router issues. If you need to get to the Internet then you need Ethernet/Wi-Fi; if you are self-contained in a room or building then ZigBee, Z-Wave, and Bluetooth are available. Remember all wireless technologies need FCC certification.

3. Features

With an IoT connected product, companies can now add features to their products that were not possible or imagined. These features can get you direct access to the customer for updates, maintenance, and new revenue opportunities.

4. User interface

How the user interfaces with a product is important. Are you going to use buttons, LEDs, or a display on the product? Also what web and app interfaces are you going to provide?

5. Power

One of the first decisions should be the power source. If the device will be powered by batteries then all design decisions must consider how to preserve power. Frequency of communication does have an influence on power selection, too.

6. Size Size matters. Consider how the network will impact the size of the device. Connectors and antennas required by some networks will add to the size.

7. Antenna

All wireless networks use an antenna, internal or external to the product. The trend is to move the antenna inside the enclosure if it is plastic. All metal enclosures would require external antennas.

8. Cloud

Cloud applications provide products a user interface to the product and the data. There are private and public clouds. Most clouds have a standard API for developing your application.

9. Interoperability

Does your product needs to communicate with other vendors’ products? If so, then you need to adopt a standard set of protocols, such as Apple’s HomeKit, to communicate with other products.

10. Security

Security is becoming a major issue, so you need to design in as many layers of security as feasible. SSL and password are the minimum.

**3.2 Design Details**

In this article, we are going to implement a 3x4 keypad with the nodemcu to enter the password of the given security system. The keypad is a very easy to use device and gives input to the nodemcu to perform the further operations.

Following is the circuit diagram of the working model:

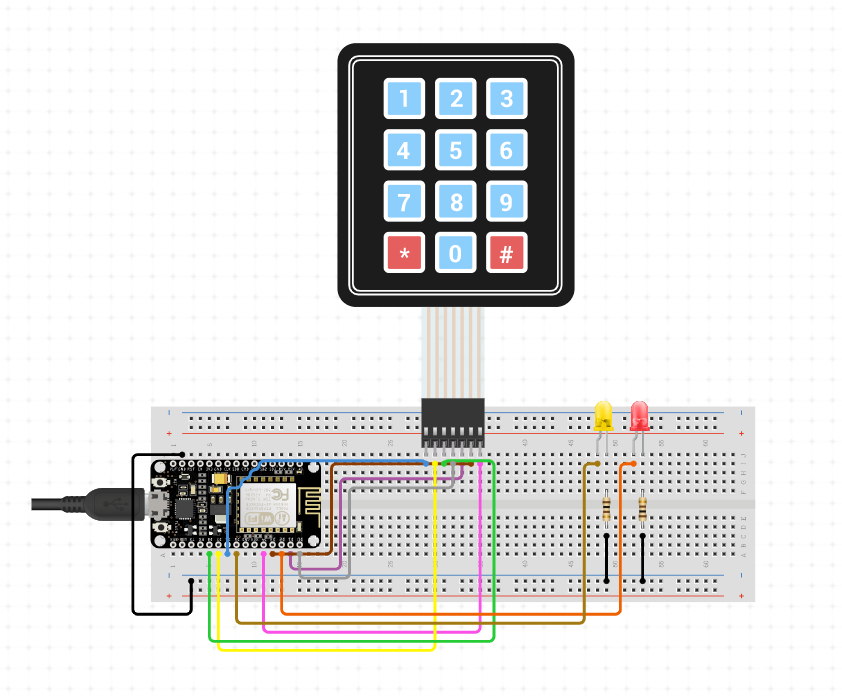


Fig. 3.1

As shown in the diagram, the keypad is used with the nodemcu along with the two LEDs, red and yellow, which are used to denote the status of the system.

The keypad has 7 pins which are connected to the digital pins of the nodemcu in the order

D3, D2, D1, D0, D7, D6, D5 from left-to-right in the keypad.

Two LEDs (red and yellow) are connected to the nodemcu along with suitable resistors connected in series. These LEDs are also connected to the digital pins of the nodemcu.

**Working of Keypad :**

If any of the keys in row1 of the matrix keypad is pressed, the corresponding column line will give low and similarly if the second key is pressed in row1, then the column line 2 will give low. This process is repeated for all the rows.

**Block Diagram:**

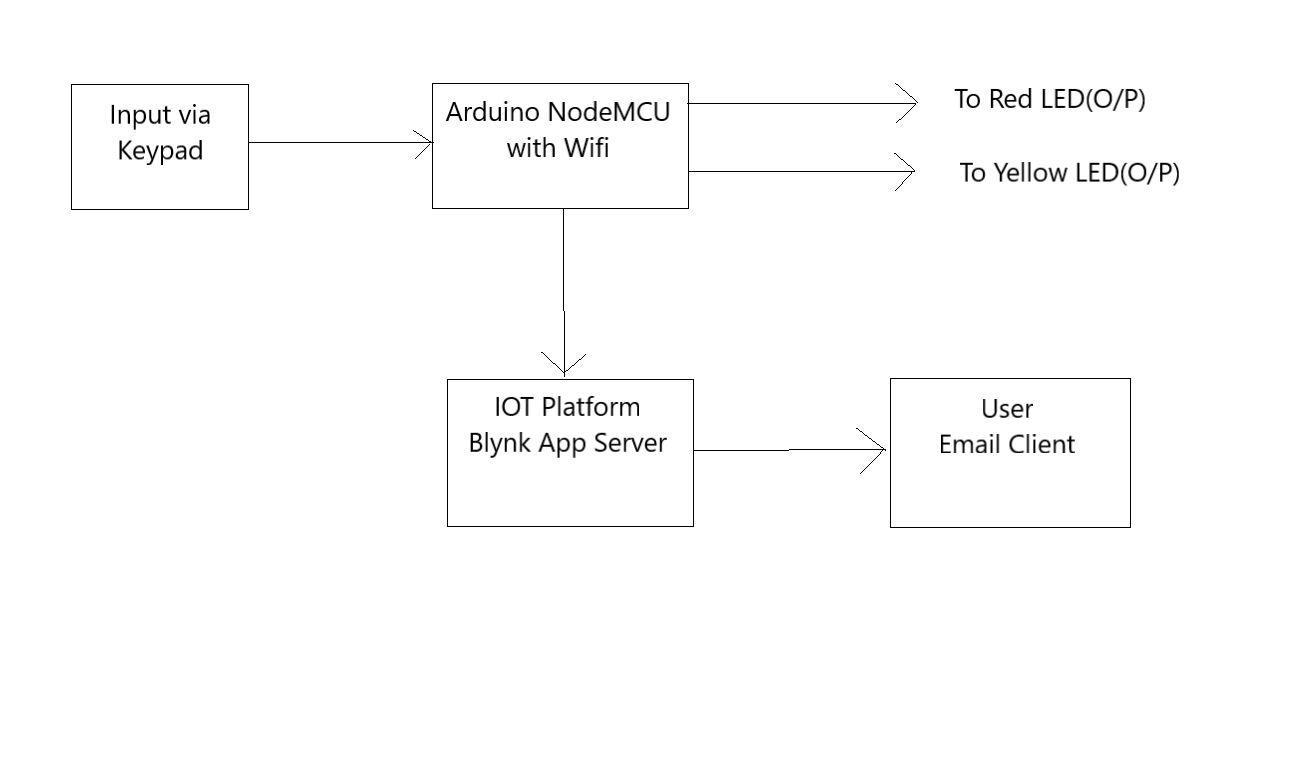
****

Fig. 3.2

**Program (Code)** :

#include<Blynk.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <Keypad.h>

#define Password\_Lenght 5 // Give enough room for 4 chars + NULL char

char Data[Password\_Lenght]; // 4 is the number of chars it can hold + the null char = 5

char Master[Password\_Lenght]= "4466";

byte data\_count = 0, master\_count = 0;

bool Pass\_is\_good;

char customKey;

int correct= 15;

int wrong= 2;

int wrong\_email = 0;

const byte ROWS = 4;

const byte COLS = 3;

char keys[ROWS][COLS] = {

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

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

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

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

};

bool door = true;

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

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

Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS); //initialize an instance of class NewKeypad

char auth[] = "dHON-oA9BzDHSItcujInBT9PFQrrg4tu"; //Put your token here

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "J"; //put wifi name here

char pass[] = "janhavi11"; //wifi password

BlynkTimer timer;

void setup()

{

// Debug console

Serial.begin(9600);

Blynk.begin(auth, ssid, pass);

// You can also specify server:

//Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);

//Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

pinMode(correct, OUTPUT);

pinMode(wrong, OUTPUT);

// Setup a function to be called every second

timer.setInterval(1000L, myTimerEvent);

delay(3000);

}

//0 stands for door open

//1 stands for door closed

void loop()

{

Blynk.run();

timer.run();

if (door == 0)

{

customKey = customKeypad.getKey();

if (customKey == '#')

{

door = 1;

Blynk.virtualWrite(V4, "System is locked.");

digitalWrite(correct, LOW); // turn the LED on

digitalWrite(wrong, HIGH); // turn the LED on

}

}

else Open();

}

void clearData()

{

while (data\_count != 0)

{ // This can be used for any array size,

Data[data\_count--] = 0; //clear array for new data

}

return;

}

void Open()

{

customKey = customKeypad.getKey();

if (customKey) // makes sure a key is actually pressed, equal to (customKey != NO\_KEY)

{

Data[data\_count] = customKey;

Blynk.virtualWrite(V5, Data);// store char into data array

data\_count++; // increment data array by 1 to store new char, also keep track of the number of chars entered

}

if (data\_count == Password\_Lenght - 1) // if the array index is equal to the number of expected chars, compare data to master

{

if (!strcmp(Data, Master)) // equal to (strcmp(Data, Master) == 0)

{

door = 0;

wrong\_email = 0;

digitalWrite(correct, HIGH); // turn the LED on

digitalWrite(wrong, LOW); // turn the LED on

Blynk.virtualWrite(V4, "Access Granted");

}

else

{

delay(1000);

door = 1;

wrong\_email = wrong\_email + 1;

Blynk.virtualWrite(V4, "Access Denied.");

Blynk.notify("Alert: Wrong Password.");

digitalWrite(wrong, HIGH); // turn the LED on

digitalWrite(correct, LOW); // turn the LED on

}

clearData();

}

if (wrong\_email == 3)

{

Blynk.email("janhavi.zarapkar@gmail.com","Access Denied","Alert!, Someone tried to access your system with invaid password more than 2 consecutive time.");

wrong\_email = 0;

}

}

void myTimerEvent()

{

Blynk.virtualWrite(V5, Data);

}

**3.3 GUI Design**

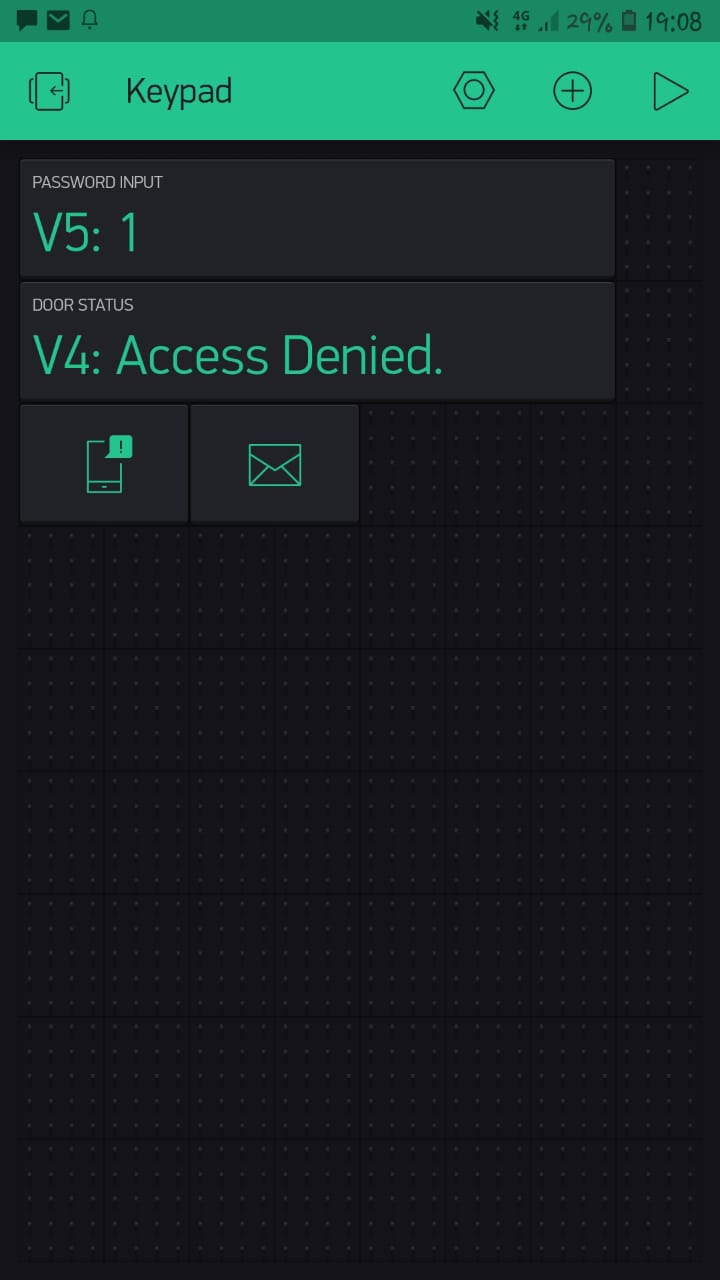
****

Fig. 3.3

**Password Input Field:**

In this field the password is seen which is entered by the user through the keypad.

**Door Status Field:**

In this field the status of the system will be seen.

If the user presses the right password then “Access Granted” will appear in this field otherwise “Access Denied” will be seen.

**Mobile Notification:**

If access is granted then a mobile notification appears on the blynk app. If access is denied then also mobile notification appears on the blynk app.

**Email Notification:**

If the user attempts wrong password more than 2 times then email notification comes on the verified user id.

The next chapter deals with the results and analysis of our System.

1. **RESULTS AND ANALYSIS**

**4.1 Implementation Details**

The connections are made as shown in the circuit diagram given in the previous section.

Following steps are followed to implement the model:

Step 1: Password is entered with the help of Keypad. The initial state of the LED will be the system is locked (shown by red LED)

Step 2: Depending on the password entered, there are two possibilities:

1. If the password is correct, access to the system is granted and the yellow LED glows signifying the same.
2. If the password is incorrect, the person gets two more chances and if he/she fails to enter the correct password in those two attempts also then an email will be sent to the registered email address of the authorized person saying that an intruder is trying to access the system. The red LED will also glow denoting the same.

**4.2 Results and Evaluation**

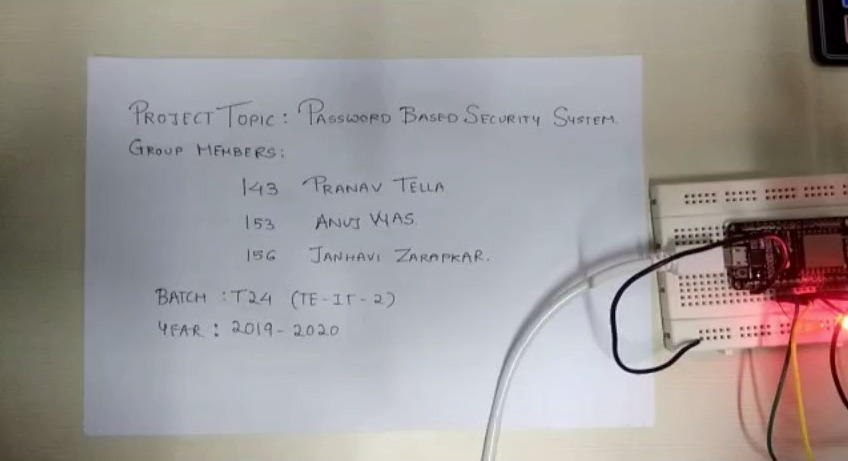
****

Fig. 4.1

These are this IoT project details.

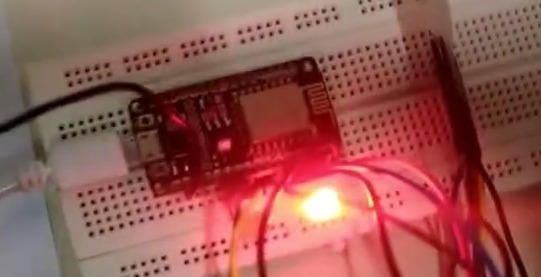
****

Fig. 4.2

This is the initial state of the Arduino NodeMCU along with the Red led glowing.

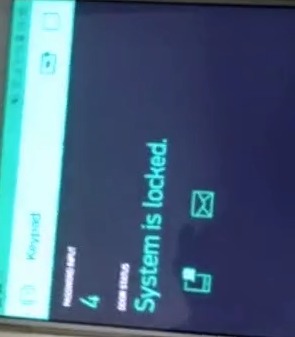
****

Fig. 4.3

This is the initial state of the of blynk app where it displays the status of the lock.

****

Fig. 4.4

Upon entering the correct password, the yellow led glows and the red led is switched off.

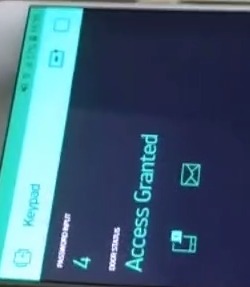
****

Fig. 4.5

The status of the lock on the blynk app immediately changes to Access Granted.

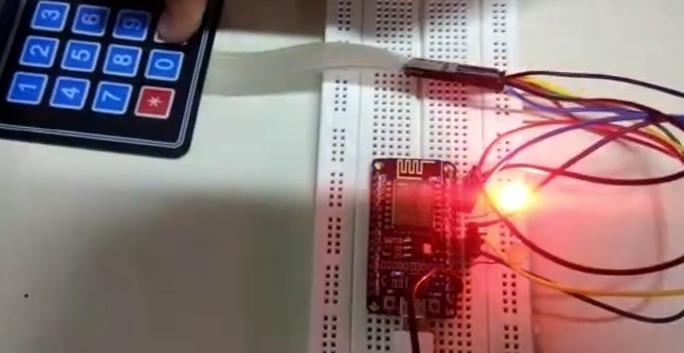
****

Fig. 4.6

On pressing the # key of the keypad, the status of the lock changes to System is locked and the red led starts glowing.

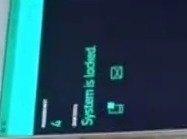
****

Fig. 4.7

This change is immediately reflected on the blynk app.

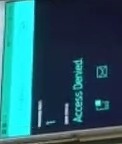
****

Fig. 4.8

Upon entering the wrong password, the status changes to Access Denied on the blynk app.

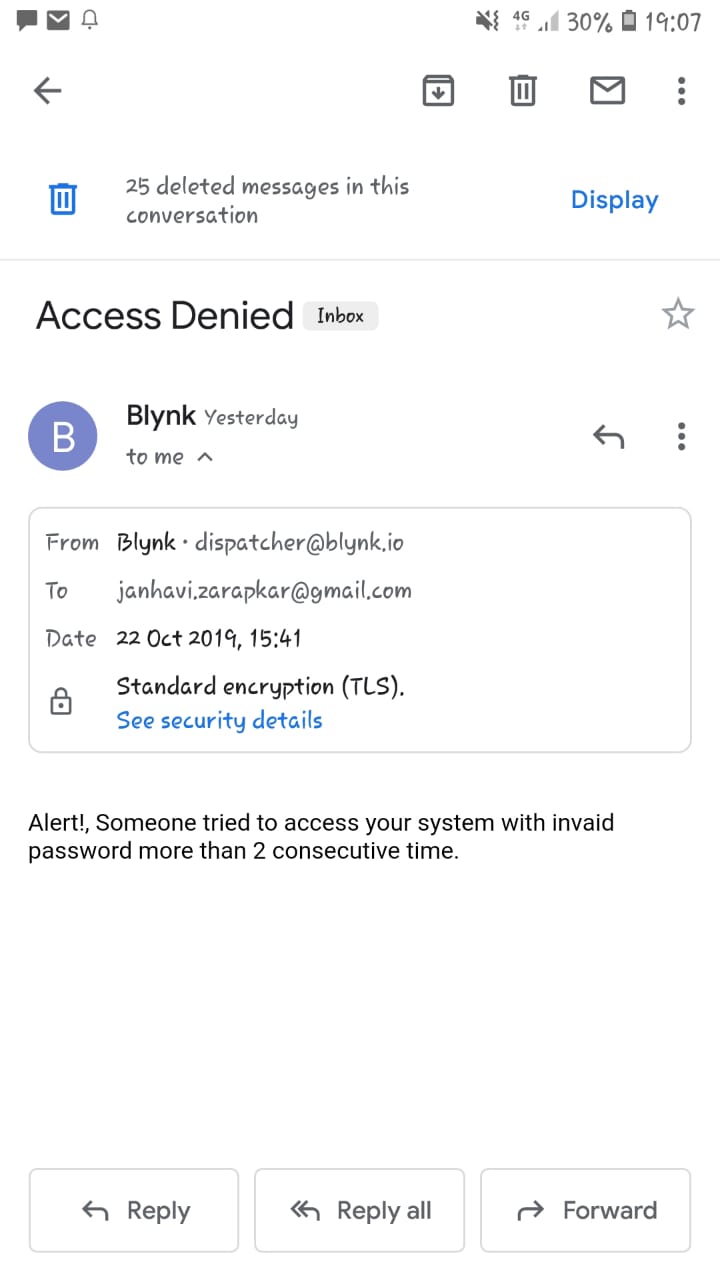
****

Fig. 4.9

Hence, if the user enters the correct password then he/she will be granted access to the system but if he/she fails to enter the valid password in 3 attempts then an email will be sent

**5. CONCLUSION AND FUTURE SCOPE**

Since, security has become a major concern for everyone it is important to have a security system that helps us have a look at any time if anyone is trying to enter the premises or access the system by using unethical means.

Using this device, personalized security can be provided for the people in their day-to-day lives and they themselves can monitor the systems as per their needs. As the user will get notified every time an intruder tries to access the system, this inappropriate intrusion can be prevented and measures can be taken to take the required action on this activity.

Hence, this device will provide better security in terms of managing and having an overview of our own system if anyone tries to access it in a wrong way.

The following project can be taken to new heights and can be used in many different ways to provide enhanced and more personalized security:

* The password that is kept can be changed everyday using a function that generates random numbers everyday and this same number will be sent to the authenticated users as soon as it is generated .

In this way, it will become very difficult for the intruder to access the system as the password will change every now and then and the attacker won’t be able to predict the correct password as there are thousands and millions of combinations possible.

* Sensors which allow people only on the basis of their physical attributes such as:

1. Fingerprint Sensors:

The biometrics of the only authorized people can be stored and only when these people access the system, the access will be granted.

1. Retina Scanners;

This will work similar to the fingerprint sensor but here the retina of the users will be scanned and stored.

**List of References**

<https://www.youtube.com/watch?v=GteMrHri6r8>

[https://www.arduino.cc](https://www.arduino.cc/)

<https://www.circuito.io/app?components=8653,9442,360216>

<https://techtutorialsx.com/2017/03/18/esp8266-interfacing-with-a-4x4-matrix-keypad/>

<https://circuitdigest.com/microcontroller-projects/keypad-interfacing-with-arduino-uno>

<https://en.wikipedia.org/wiki/Arduino>

<https://en.wikipedia.org/wiki/NodeMCU>

<https://en.wikipedia.org/wiki/Jump_wire>

<https://en.wikipedia.org/wiki/Breadboard>