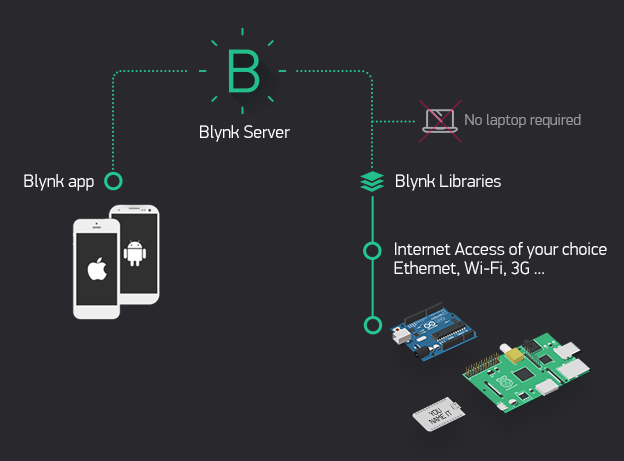
**Guide for Digital lock with NodeMCU and Blynk**

1. **Requirements:**

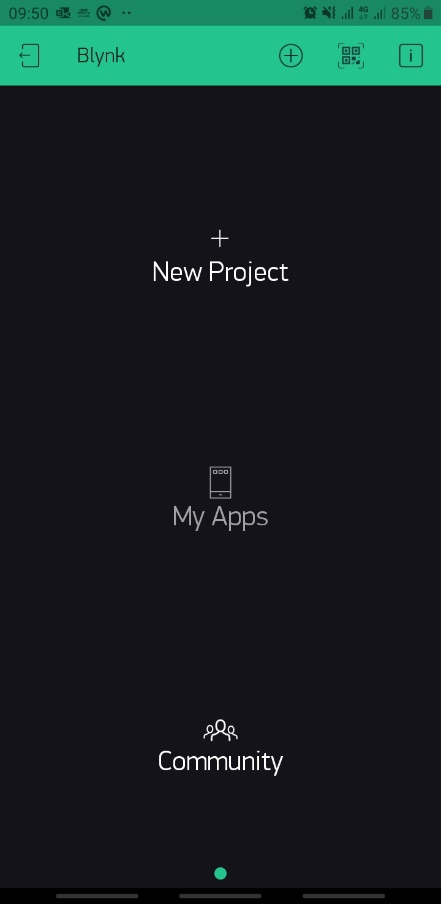
* Use NodeMCU or Wemos D1 with ESP8266 wifi
* Create an digital lock
* Send information about lock status to smart phone application (Blynk)
* Your app can open or close the lock

1. **Setup Blynk on smartphone**

How does our application work?



Go to Application store and Search for “Blynk”, install it! You got an app as below



Setting Blynk with new Project:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Notice:

After click Create project, the Authentication Key will send to your register email.

Add widgets: LCD and Lock/Unlock button

Click the “+” button on Blynk screen

|  |  |  |
| --- | --- | --- |
| Screen of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated | A picture containing electronics, player, holding, computer  Description automatically generated |

1. **Setup NodeMCU circuit**

Circuit:

**Note: you can do the same with Wemos D1 board and the 4x4 keypad.**

A close up of a device

Description automatically generated

Pin connection:

D0 – D6 connect to pin 7 to 1 of keypad (pin 7 on 1st of the left hand side, pin 1 on the right hand side )

D7 connect to LED

D8 connect to Servo

1. **Code (You can see the code file on CMS)**

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <Keypad.h>

#include <Servo.h>

#include <string.h>

// Create an instance lcd from WidgetLCD object

WidgetLCD lcd(V1);

// You should get Auth Token in the Blynk App.

char auth[] = "INPUT YOUR AUTHENTICATION KEY HERE";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "YOUR WIFI SSID HERE";

char pass[] = "YOUR WIFI PASSWORD";

// Define some constants and variables

const byte rows = 4;

const byte columns = 3;

int holdDelay = 700;

int n =3;

int state = 0;

char key = 0;

int pos = 0;

//Default password is 0000

String default\_passwd = "0000";

//variable to store the user input for password

String input\_passwd = "";

//Define keys for lock and unlock or change password function

char lock\_key = '\*';

char unlock\_key = '#';

char change\_pass\_key = '-'; //press \* key for more than 3 second

// Create an instance for servo motor

Servo servo\_d8;

//Define characters matrix

char keys[rows][columns] =

{

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

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

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

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

};

//Define pins for every row of keypad

byte rowPins[rows] = {D0, D1, D2, D3};

//Define pins for every column of keypad

byte columnPins[columns] = {D4, D5, D6};

// Create an instance for our keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, columnPins, rows, columns);

void setup()

{

// Debug console

Serial.begin(9600);

//Inititate Blynk

Blynk.begin(auth, ssid, pass);

//Setup LED

pinMode(D7, OUTPUT);

// connect signal pin of servo to pin number D7 on NodeMCU

servo\_d8.attach(D8);

servo\_d8.write(pos);

//Clear the LCD Widget and send status WELCOME!

lcd.clear();

lcd.print(0, 0, "WELCOME!");

delay(1000);

}

//Setup for Blynk to write data from App to Hardware

BLYNK\_WRITE(V0)

{

int pinValue = param.asInt();

if(pinValue == 1)

{

digitalWrite(D7, HIGH);

servo\_d8.write(90);

lcd.clear();

lcd.print(0, 0, "WARNING!");

lcd.print(0, 1, "UNLOCKED!");

}

else{

digitalWrite(D7, LOW);

servo\_d8.write(0);

lcd.clear();

lcd.print(0, 0, "WELCOME!");

lcd.print(0, 1, "LOCKED!");

}

}

void loop()

{

Blynk.run();

Key\_pressed();

}

// Define Unlock()

void Unlock()

{

digitalWrite(D7, HIGH);

lcd.clear();

lcd.print(0, 0, "WARNING!");

lcd.print(0, 1, "UNLOCKED!");

for (pos = 0; pos <= 180; pos += 1)

{

// tell servo to go to position in variable 'pos'

servo\_d8.write(pos);

// wait 15 ms for servo to reach the position

delay(15); // Wait for 15 millisecond(s)

}

delay(3000); //open door 3s then close

//lock by servo\_d8

for (pos = 180; pos >= 0; pos -= 1)

{

// tell servo to go to position in variable 'pos'

servo\_d8.write(pos);

// wait 15 ms for servo to reach the position

delay(15); // Wait for 15 millisecond(s)

}

lcd.clear();

lcd.print(0, 0, "WELCOME!");

lcd.print(0, 1, "LOCKED!");

digitalWrite(D7, LOW);

}

// Keypad\_press

void Key\_pressed()

{

key = function\_key(n);

if (key == unlock\_key)

{

Serial.print("Input password: ");

input\_passwd = input\_password(4);

if (input\_passwd == default\_passwd)

{

//Unlock by servo\_d8 and send alert to LCD

Unlock();

}

else

{

Serial.println("Wrong password!");

digitalWrite(D7, HIGH);

delay(500);

digitalWrite(D7, LOW);

delay(500);

digitalWrite(D7, HIGH);

delay(500);

digitalWrite(D7, LOW);

lcd.clear();

lcd.print(0, 0, "WARNING!");

lcd.print(0, 1, "WRONG PASSWORD!");

}

//Reset input\_passwd

input\_passwd = "";

key = 0;

}

if (key == change\_pass\_key)

{

default\_passwd = change\_password(4, default\_passwd);

delay(2000);

key =0;

}

}

// Define function for key

char function\_key(int n)

{

char temp = keypad.getKey();

if ((int)keypad.getState() == PRESSED)

{

if (temp != 0) {key = temp;}

}

if ((int)keypad.getState() == HOLD)

{

state++;

state = constrain(state, 1, n);

delay(holdDelay);

}

if ((int)keypad.getState() == RELEASED)

{

key += state;

state = 0;

}

delay(100);

Serial.println(key);

return key;

}

// Define function input\_password

String input\_password(int num\_char)

{

String passwd = "";

//Serial.print("Input password: ");

do

{

char temp = keypad.getKey();

if (temp != 0) {Serial.print(temp); passwd += temp;}

delay(100);

}

while (passwd.length() < num\_char);

Serial.println();

return passwd;

}

// Define function change\_password

String change\_password(int num\_char, String current\_passwd)

{

//Authenticate the old password:

Serial.print("Input old password: ");

String old\_passwd = input\_password(num\_char);

if (old\_passwd != current\_passwd)

{

Serial.println("Password does not match! Nothing changes");

return current\_passwd;

}

//New password

Serial.print("Input new password: ");

String new\_passwd = input\_password(num\_char);

//Confirm passwd

Serial.print("Input new password again: ");

String confirm\_passwd = input\_password(num\_char);

if (confirm\_passwd == new\_passwd)

{

Serial.println("Password has changed!!!");

return confirm\_passwd;

}

else

{

Serial.println("Password does not match! Nothing changes");

return current\_passwd;

}

}

1. **Test (See demo on class)**

Click on Play button on Blynk to start our apps

|  |  |  |
| --- | --- | --- |
| A picture containing clock, meter  Description automatically generated | A picture containing clock  Description automatically generated | A picture containing clock, meter  Description automatically generated |