Internet of Things (IOT) Workshop

FSKM UiTM Shah Alam

Prepared by: Muhammad Azizi (mazizi@fskm.uitm.edu.my)

Pre-requisite:

1. Myduino IOT Training Kit
2. Laptop with Internet Connection
3. Smartphone with hotspot enabled
4. Knowledge about C Programming
5. Curiosity, Open Mind and Fun ☺

Activity 1: Installing Arduino IDE & Connecting to NodeMCU

Before we can begin using Arduino microcontroller, we need to install Arduino IDE software to write and upload programs to Arduino compatible boards. We also need to make sure the PC was installed with correct driver and Arduino IDE has been installed with NodeMCU hardware library. Lastly, the Arduino IDE must be connected with correct port which connected to NodeMCU.

1.1 Download and install the latest Arduino IDE

You can download from here: https://www.arduino.cc/download_handler.php?f=/arduino-1.8.10-windows.exe

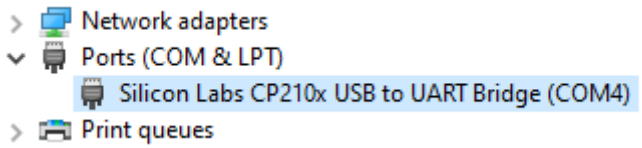
1.2 Plug in your NodeMCU board to your laptop's USB port

1.3 Installing CP2102 USB Driver

Open up your Microsoft Window Device Manager, Right click on CP210X USB and choose update Driver. Click on browse my computer for driver software. Locate CP2102 USB driver from this following path

64 bit Computer: C:\Program Files (x86)\Arduino\drivers\CP210x_6.7.4

32 bit Computer: C:\Program Files\Arduino\drivers\CP210x_6.7.4

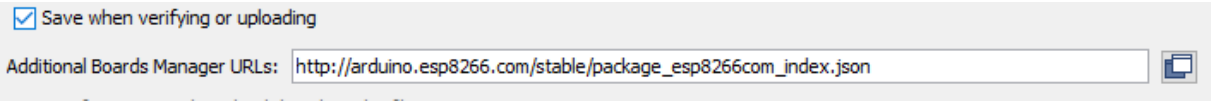


Click next button to proceed with installation, you can close the device manager once installation completed. Please also take note about NodeMCU port (above diagram shows NodeMCU is connected to COM4), we need the information to setup Arduino IDE later.

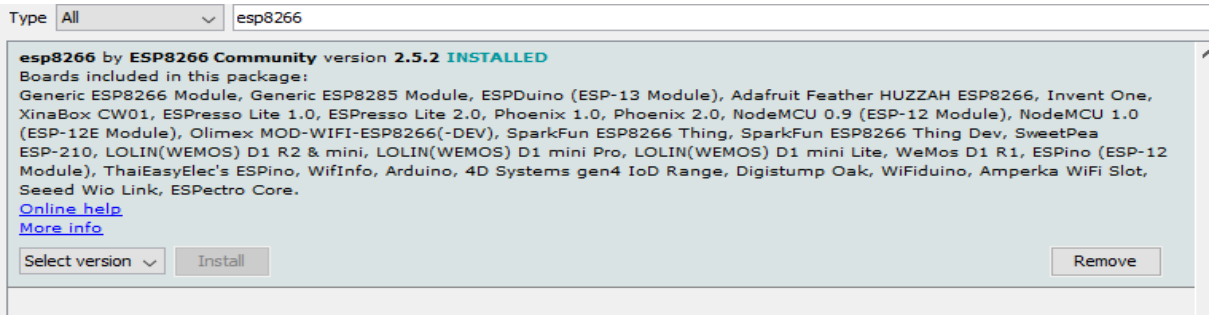
1.4 Installing NodeMCU Hardware Library in Arduino IDE

First you need to run Arduino IDE which you just installed. Then you need open preferences window, **File > Preferences**. In the preferences window, find the field for "Additional Boards Manager URLs. Paste the following URL and click OK.

Paste this – http://arduino.esp8266.com/stable/package_esp8266com_index.json



Then you need to open Boards Manager in Arduino IDE, **Tools > Board > Board Manager**. In the search bar type ESP8266 and select esp8266 by esp8266 community. Click Install to start the hardware library installation.



1.5 Select Board & Port

After you have finish with hardware library installation, we need to make sure the Arduino IDE is choosing the right board type and port. Open **Tools > Board**, make sure it chosen **"NodeMCU 1.0 (ESP-12E Module)**. Then open **Tools > Port**, make sure it chosen the COM port as per your device manager.

1.6 Test Your NodeMCU Board

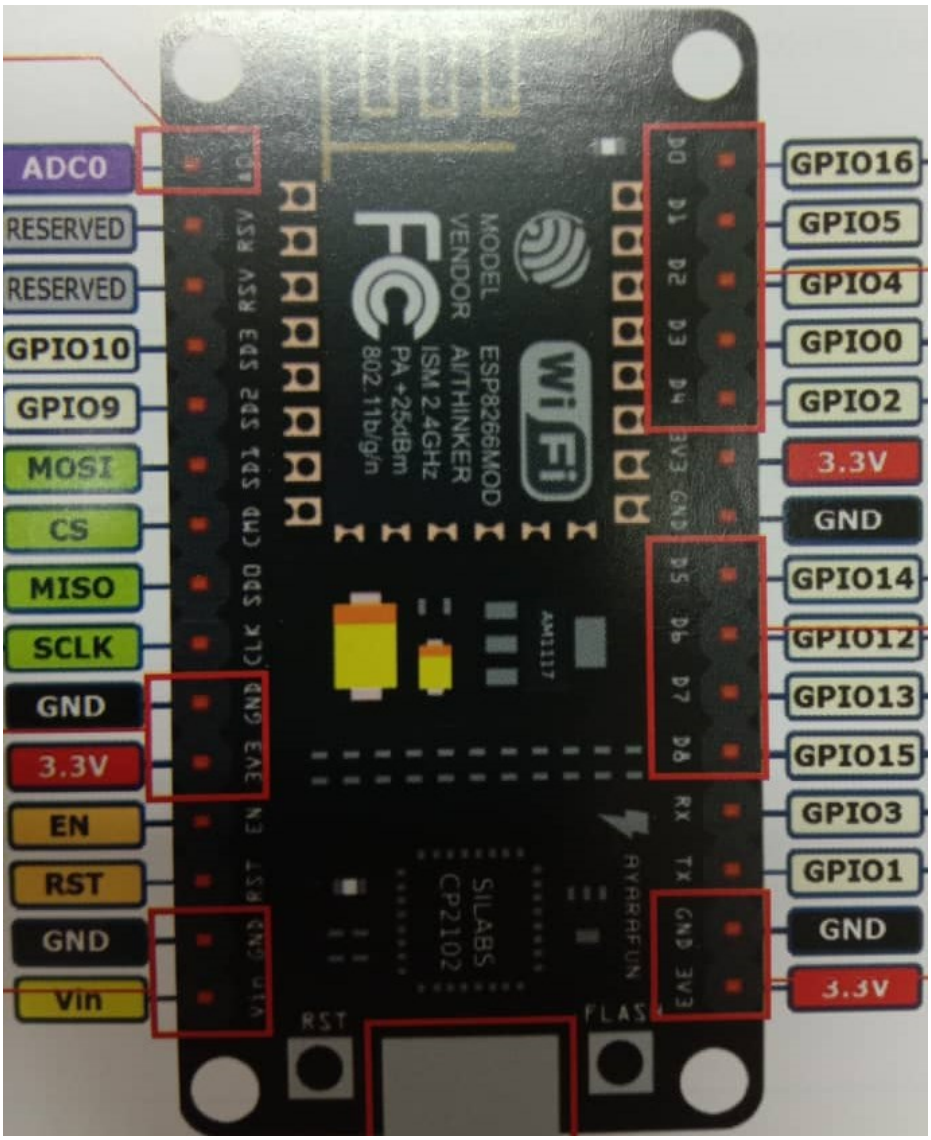
To test whether you have successfully connected your Arduino IDE to your NodeMCU board, you can write simplest code to blink the On-board LED light of NodeMCU. Type the following code into your Arduino IDE.

```
---------------------------------------------------------------------------------------------------------------------
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);  //This will initialized the on-board LED as an output
}
void loop() {
  digitalWrite(LED_BUILTIN, LOW);  //Turn on LED
  delay(1000);  // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);  // Wait for two seconds
}
---------------------------------------------------------------------------------------------------------------------
```

Click upload to upload the code to NodeMCU Board. If the blue LED blink, congratulations.

Activity 2: Digital Output – Running Lights

The NodeMCU board comes with Analogue Input, Digital Input/Output, USB Port and Power Rail. The Pin as follows:



Power Rail

- GND and 3.3V can provide ground and power to the external circuit. VIN and GND receive supply voltage to power it up.

Analogue Input

- 1 Analogue channel input is physically labelled as A0 on the board, while in coding labelled as ADC0.
- Has a voltage conversion of 0 to 1 Volts into digital values of 10-Bits (0 to 1023).

Digital Input / Output

- 9 Digital Pins are available which physically labelled as D0 to D8. In Coding logically it is labelled as GP1O 1,3,15,13,12,14,2,0,4,5,6.
- Each pins are bidirectional, it can be set to be an INPUT or an OUTPUT.
- All Digital pins provide 10-Bits PWM (0 to 1023).

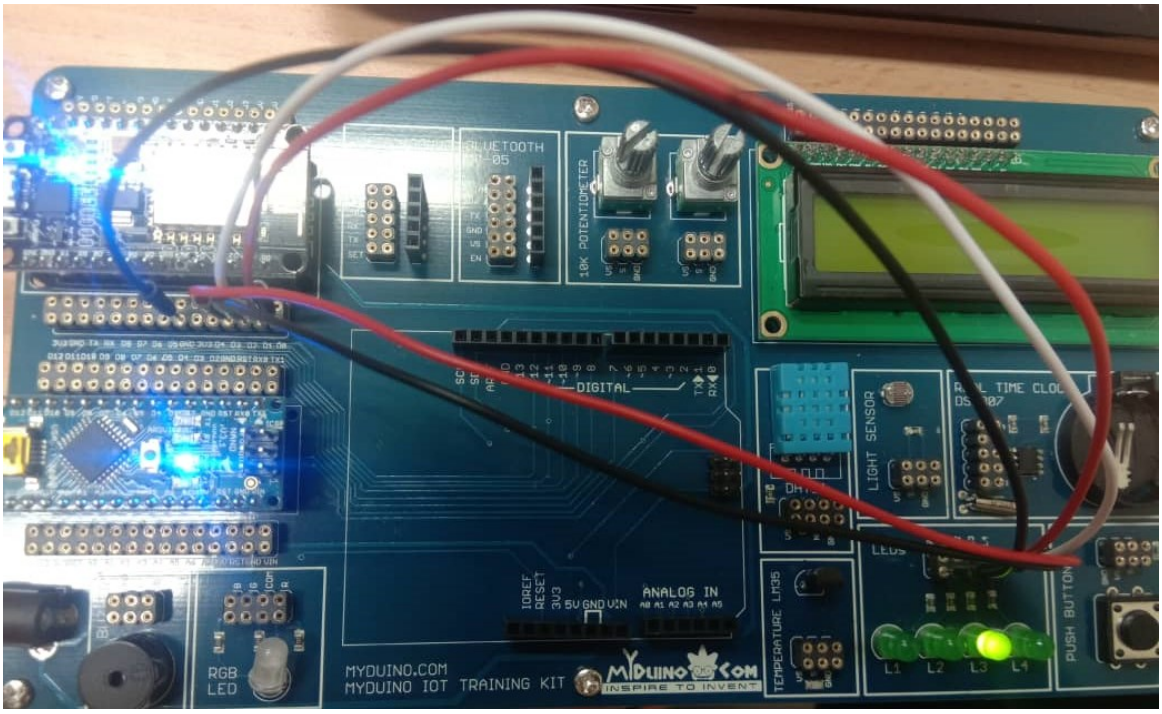
USB Port

- Used to power up the board and also uploading code to the board

An Arduino board has digital pin which outputs 5V at 40mA. We can use the digital pin output to turn on LED light. The LED only requires a voltage of 2V and a current of 20mA. Therefore, we need to put in a resistor that will reduce the 5V voltage. The LED in MyDuino IOT training kit has already been wired with a resistor. In Activity 2 we will try to create a running light by connecting 4 LED into NodeMCU digital Pins.

2.1. Connect LED to NodeMCU board

First we need to physically connect the 4 LED to NodeMCU, please connect LED 1 to D0, LED2 to D1, LED3 to D2, LED4 to D3 and LED ground to GND using jumper cable. Please refer picture below



2.2 Upload Running light code into NodeMCU

After you have finished with physical connection, type below code into your Arduino IDE and upload it into NodeMCU board. You should see the LED lights up one by one.

```
void setup()
{
  pinMode(16, OUTPUT); //Initialized D0,D1,D2,D3 as output
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(0, OUTPUT);
}
void loop()
{
  digitalWrite(16, HIGH); //Turn on D0
  delay(1000);
  digitalWrite(16, LOW); //Turn off D0
  delay(1000);

  digitalWrite(5, HIGH); //Turn on D1
  delay(1000);
  digitalWrite(5, LOW); //Turn off D1
  delay(1000);

  digitalWrite(4, HIGH); //Turn on D2
  delay(1000);
  digitalWrite(4, LOW); //Turn off D2
  delay(1000);

  digitalWrite(0, HIGH); //Turn on D3
  delay(1000);
  digitalWrite(0, LOW); //Turn off D3
  delay(1000);
}
```

2.3 Modify On Your Own

Now please modify the physical connecting and coding so that the 4 LED connect to pin D5, D6, D7 and D8. You also need to modify the LED sequence, please light up LED 1 and 2 at the same time and after that LED 3 and 4

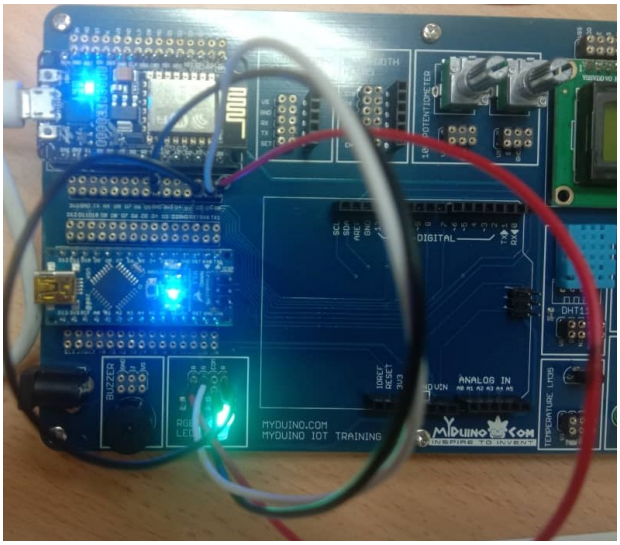
Activity 3: Analogue Output – RGB Light

An analogue signal can produce a range of values unlike a digital signal that has only two values of HIGH and LOW. Having an LED as a digital output, it can only be turn ON or OFF with no option to control its brightness. To do that, we need to have a range of voltage values in between 0V to 5V and this is what analogue is.

NodeMCU microcontroller is digital, thus a technique called Pulse Width Modulation (PWM) could simulate the functions of an analogue signal by having the capabilities of producing varying output. In NodeMCU board all digital pins can provide PWM function. In activity 3 we will send an analogue signal to the RGB led so it can produce variety of light colour.

3.1 Connect RGB LED to NodeMCU board

First we need to physically connect the RGB LED to NodeMCU, please connect LED R to D0, LED G to D1, LED B to D2, and LED ground (COM) to GND using jumper cable. Please refer picture below



3.2 Upload RGB Light code into NodeMCU

After you have finished with physical connection, type below code into your Arduino IDE and upload it into NodeMCU board. You should see the RGB LED producing variety of colour due to different analogue signal.

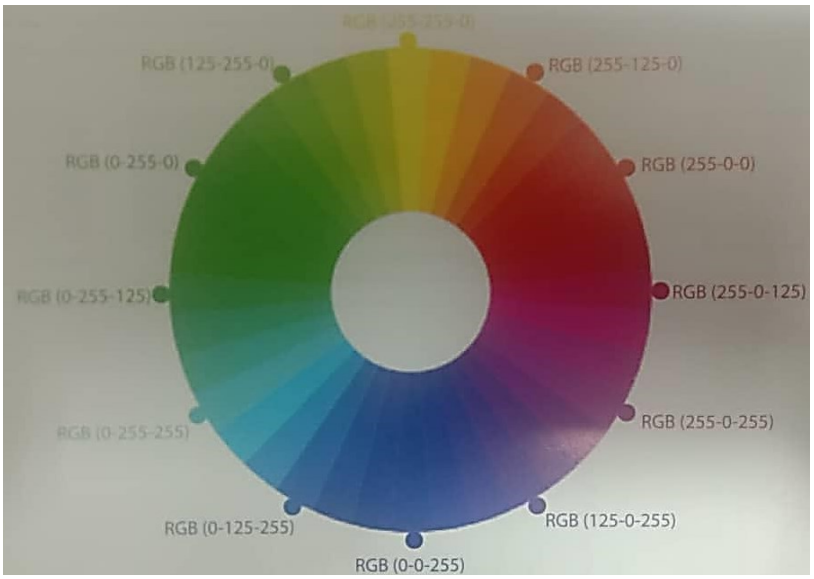
```
 void setup()
{
  pinMode(16, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
```

```
}
void loop()
{
 analogWrite(16, random(0,255));
 analogWrite(5, random(0,255));
 analogWrite(4, random(0,255));
 delay(500);

 analogWrite(16, random(0,255));
 analogWrite(5, random(0,255));
 analogWrite(4, random(0,255));
 delay(500);
 analogWrite(16, random(0,255));
 analogWrite(5, random(0,255));
 analogWrite(4, random(0,255));
 delay(500);
}
```

3.3 Modify On Your Own

Now I need you to try modifying the code so that you can mimic a traffic light using NodeMCU. You can refer diagram below for RGB colour scheme:

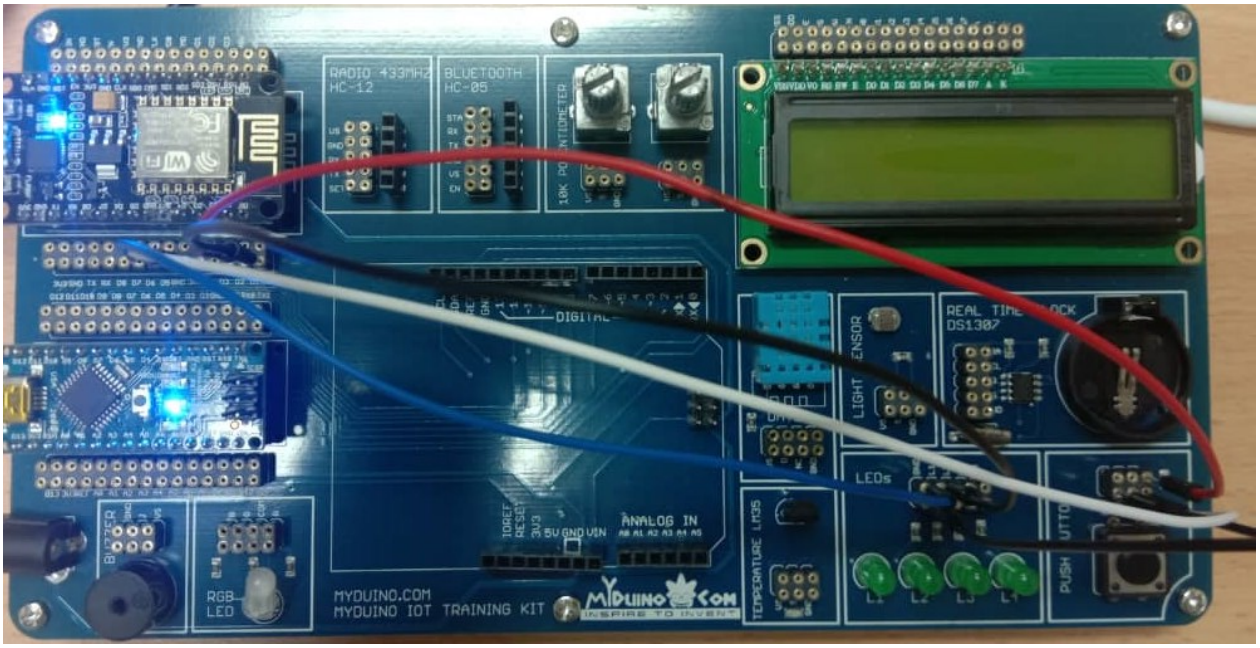


Activity 4: Digital Input – Push Button

NodeMCU digital pin can also be made to receive digital Input. In activity 4 we will use push button to send digital input to NodeMCU pin and light up an LED.

4.1 Connect Push Button and LED to NodeMCU board

In this activity, you need to connect Push Button S pin into NodeMCU D1 pin, then LED 1 pin into NodeMCU D0 pin. Then connect Push Button VS into NodeMCU 3V pin. Finally connect Push Button Ground into LED ground and LED ground into NodeMCU GND pin. Please refer following diagram



4.2 Upload Push Button code into NodeMCU

After you have finished with physical connection, type below code into your Arduino IDE and upload it into NodeMCU board. You should able to light up the LED when you push the button.

```
void setup()
{
  pinMode(16, OUTPUT);
  pinMode(5, INPUT);
}
void loop()
```

```
{
 if (digitalRead(5) == HIGH)
 {
  digitalWrite(16, HIGH);
 }
 else
 {
  digitalWrite(16, LOW);
 }
}
```

4.3 Modify On Your Own

Now I need you to modify this activity. I want you to add buzzer, the buzzer will produce a sound when you push the button. You can use Arduino tone function to use the buzzer

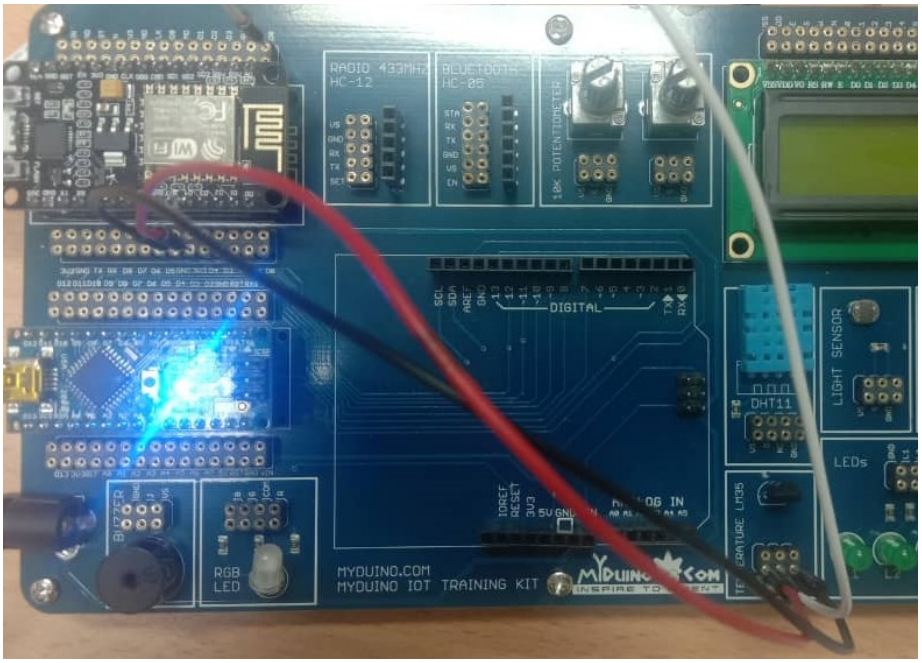
tone({pin no}, 247, 250);

Activity 5: Analogue Input – Digital Thermometer

Reading digital input (e.g. Push Button) only allows the detection of two states binary signal of HIGH or LOW. Real world works in analogue with varying values such as temperature, light, humidity and other physical force. In this activity, we will learn how to use analogue input pin to read a temperature using a sensor and display it to Arduino IDE console.

5.1 Connect LM35 Temperature Sensor to NodeMCU Board

In this activity, you need to connect LM35 sensor S pin to NodeMCU A0 pin, then you need to connect LM35 VS pin to NodeMCU 3V pin and LM35 GND pin to NodeMCU GND pin. Please refer diagram below.



5.2 Upload Temperature Reading Code into NodeMCU Board

After you have finished with physical connection, type below code into your Arduino IDE and upload it into NodeMCU board. You should able to see current temperature reading from Arduino IDE console. To open up console click **Tools > Serial Monitor**

```
void setup()
{
 Serial.begin(9600);
}
void loop()
{
 int Dout = analogRead(A0);
 float Vin = (Dout*0.00488);
 float temp = Vin*100;
 Serial.println(temp);
 delay(500);
}
```

5.3 Modify On Your Own

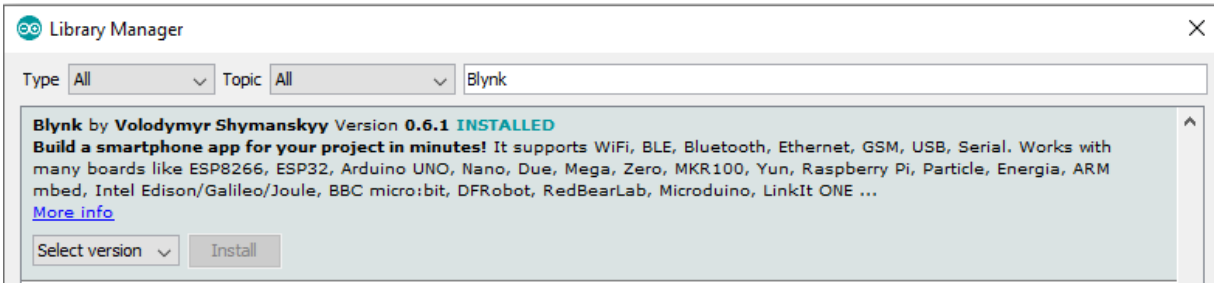
Now I need you to modify this activity. I want you to produce a sound if the temperature reaches certain threshold value (e.g. 42 Celsius).

Activity 6: IOT using Blynk

Internet of Things (IOT) is the inter-networking of physical devices with the Internet. With IOT we can monitor the status, obtain sensor reading and control our devices over the Internet. In this activity, we will use Blynk's as an IOT platform.

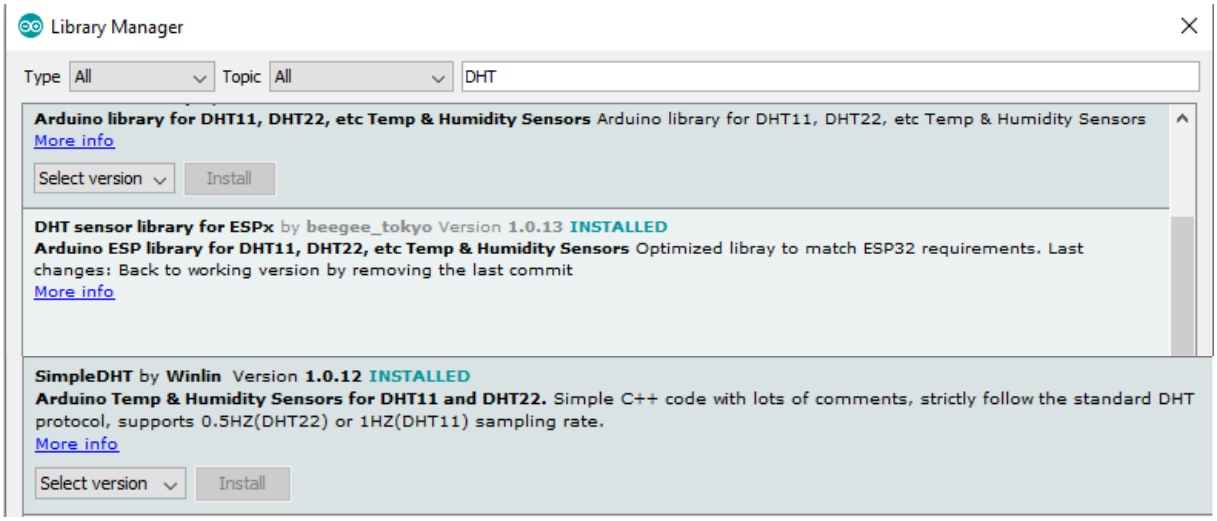
6.1 Install Blynk Library to Arduino IDE

Before we begin, we need to make sure our Arduino IDE has been installed with Blynk Platform library. To installed Blynk Library we need to open up Library Manager **Sketch > Include Library > Manage Libraries**. Then you need to type **Blynk** in the search bar and click install



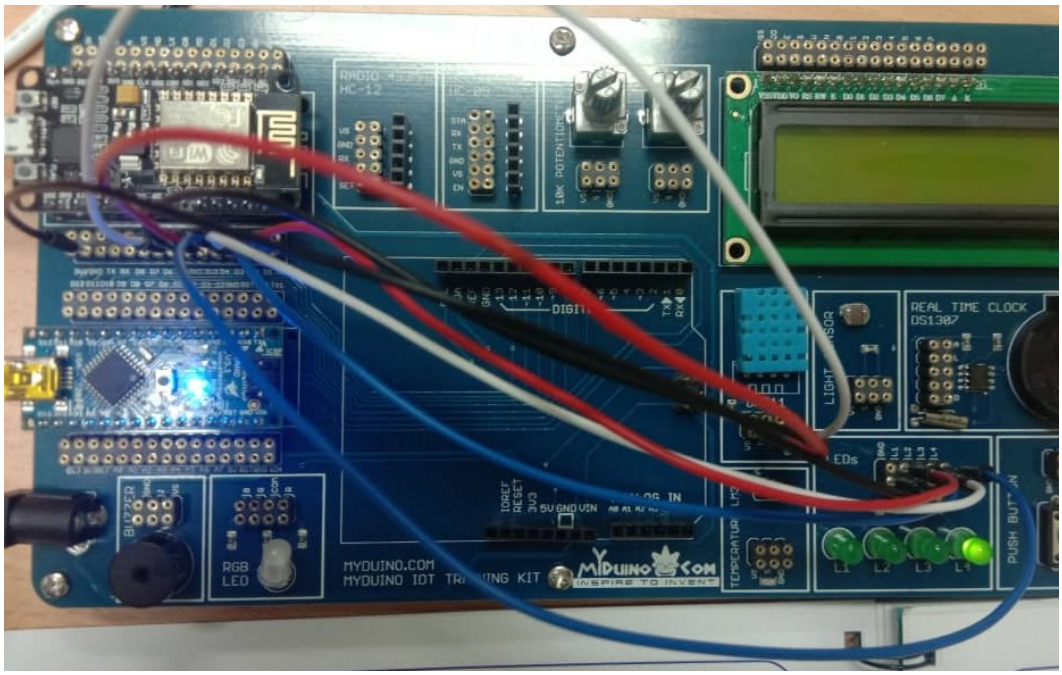
6.2 Install DHT11 Sensor Library to Arduino Library

In this activity we will use DHT11 sensor to capture surrounding temperature and also humidity, therefore we need to ensure our Arduino IDE has been installed with DHT11 libraries. To installed DHT11 library we need to open up Library Manager **Sketch > Include Library > Manage Libraries**. Then you need to type DHT in the search bar and ensure all DHT related library was installed.



6.3 Connect four LEDs and DH11 Sensor to NodeMCU Board

After that, we need to make sure the physical connection between our NodeMCU board, LED and DHT11 is properly connected using jumper wire. Please connect LED 1, 2, 3, 4 pins to NodeMCU D0, D1, D2 and D3 pins. Then please connect LED GND to NodeMCU GND Pin. After that, connect DHT11 sensor D pin to NodeMCU D5 pin and DHT11 VS pin to NodeMCU 3V pin and lastly connect DHT11 sensor GND pin to NodeMCU GND pin. Please refer diagram below.



6.4 Install Blynk App to your smartphone and create a new project

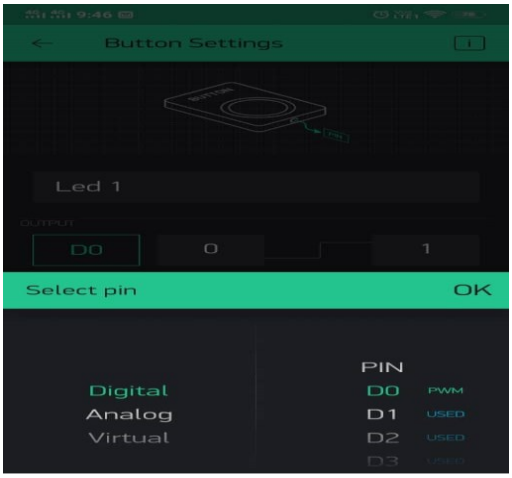
Now it times to install the Blynk App into your smartphone, you can download it from Apple Appstore or Google PlayStore. After you have installed, please run Blynk App from your smartphone and complete the registration process.

After you have completed the registration, you need to create your FirstProject. Blynk will email to you the Authentication key. You need to take note on the key as you need to put it in your Arduino code.

6.5 Create Blynk Interface in the Apps

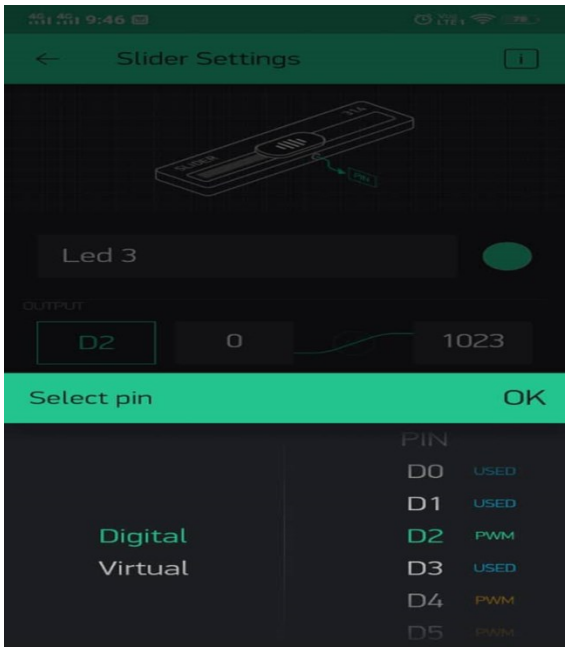
After you have created the first project in your Blynk App, you need to create the interface for you project. You need to add 2 Button, 2 Slider and 2 Gauge. Please refer diagram below:

After that, you need to configure your button. Click on your button and name it LED 1 and then click on the Output and select the Pin as Digital and D0, please refer diagram below.



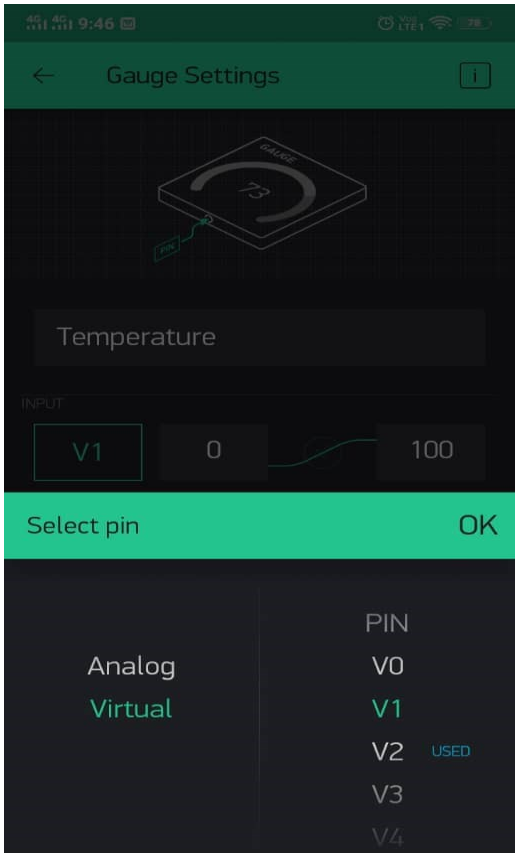
Now repeat the step for the second button, name it LED 2 and select the Pin as Digital D1.

After that, we need to configure the slider. Click on the slider and name it LED 3, select the output and select pin Digital D2. Please refer diagram below



Now repeat the step for the second slider, name it LED 4 and select the Pin as Digital D3.

Now you need to configure the gauge. Click the Gauge and named it Temperature. Click the input and select the pin as Virtual and select V1. Please refer diagram below.



Repeat the step for the second gauge, name it Humidity and select the pin as Virtual and select V2.

6.6 Setup Smartphone Hotspot

After that, you need to create a wifi hotspot using your smartphone so that your NodeMCU board can connect to your Wi-Fi and access the Internet. Please take note your Wifi SSID name and also it password, you need to put it in your arduino code.

6.7 Upload IOT Code into NodeMCU Board

After you have finished with physical connection and setup the interface in Blynk App, type below code into your Arduino IDE and upload it into NodeMCU board. After that open your firstProject in your Blynk app and click Play/Run Button, you could now turn on/off the LED and also see real-time reading of temperature and humidity from DH11 sensor.

```
#define BLYNK_PRINT Serial
```

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SimpleDHT.h>
char auth[] = "Put you Blynk Authentication Code Here";
char ssid[] = "Your Wifi Name";
char pass[] = "Your Wifi Password";

int pinDHT11 = D5;
SimpleDHT11 dht11(pinDHT11);
unsigned long previousMillis = 0;

void setup()
{
 Serial.begin(9600);
 Blynk.begin(auth, ssid, pass);
}
void loop()
{
  Blynk.run();
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis > 1500)
  {
    previousMillis = millis();
    byte temperature = 0;
    byte humidity = 0;
    dht11.read(&temperature, &humidity, NULL);
    Blynk.virtualWrite(V1, (int)temperature);
    Blynk.virtualWrite(V2, (int)humidity);
  }
}
```

References

1. Arduino Official Website, https://arduino.cc
2. Learning Arduino from Zero to Hero, 2nd Edition.
3. ESP8266 Community Forum, https://www.esp8266.com/index.php