“The important thing is not to stop questioning. Curiosity has its own reason for existing.” – Albert Einstein

Internet of Things (IOT) Workshop for Trainers

FSKM UiTM Shah Alam

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

Pre-requisite:

1. Myduino IOT Training Kit and Raspberry Pi
2. Wireless Access Point
3. Laptop with Internet Connection
4. Smartphone
5. A bit of Knowledge Programming
6. Curiosity, Open Mind and Fun ☺

Activity 0: Installing Arduino IDE & Connecting to NodeMCU

Before we can begin using Node MCU 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.

0.1 Download and install the latest Arduino IDE

You can download from here: https://downloads.arduino.cc/arduino-1.8.13-windows.exe

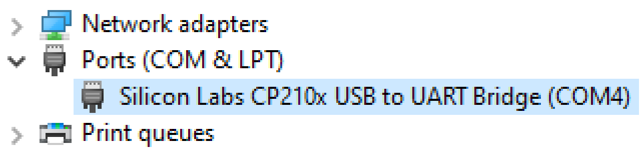
0.2 Plug in your NodeMCU board to your laptop’s USB port

0.3 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

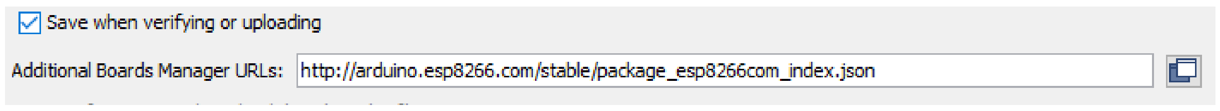


Click next button to proceed with installation, you can close the device manager once installation completed. Please also take note about NodeMCU port (above diagram shows NodeMCU is connected to COM4), we need the information to setup Arduino IDE later.

0.4 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.

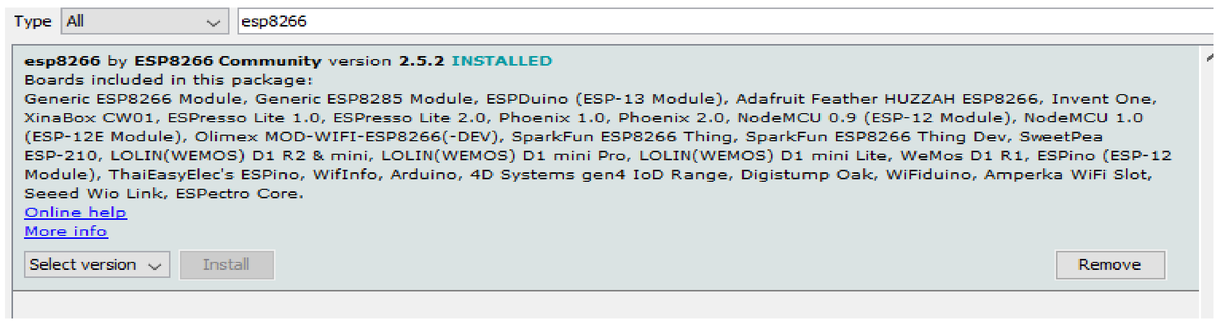
Paste this – http://arduino.esp8266.com/stable/package_esp8266com_index.json



Save when verifying or uploading

Additional Boards Manager URLs: 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.



Type All esp8266

esp8266 by **ESP8266 Community** version **2.5.2** INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

0.5 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.

0.6 Test Your NodeMCU Board

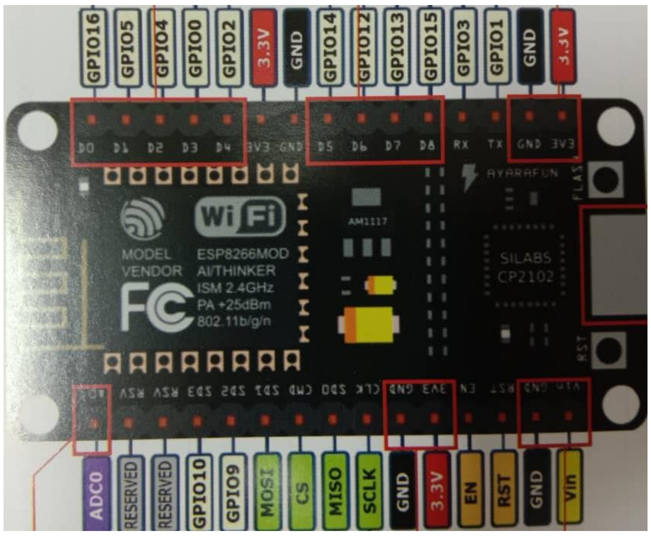
To test whether you have successfully connected your Arduino IDE to your NodeMCU board, you can write simplest code to blink the On-board LED light of NodeMCU. Type the following code into your Arduino IDE.

```
--------------------------------------------------------------------------------------------------------
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 1: Digital Input and Output GPIO

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 1 we will try to use the GPIO Digital output to turn on a LED and read digital input from a switch.

1.1. Physical Connection

First, we need to physically connect the 2 LED and push button to the Node MCU board. Kindly refer the following table

Node MCU Pins	Sensor or Actuators Pins
D0	Push Button S
3V3	Push Button VS
GND	Push Button GND
D1	LED L1
D2	LED L2
GND	LED GND

1.2 Upload code into NodeMCU

After you have finished with physical connection, type code from repository below into your Arduino IDE and upload it into Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_1

1.3 Observe Your Project

Please observed your IoT kit, is LED 1 Blinking?

____

Did LED 1 turn on when you push the button?

____

Why sometimes there is a delay to turn on LED 2 when you push the button?

____

Activity 2: GPIO Non-Blocking

Did you notice there is a delay turning on LED 2 when after you have pushed the button?

This is because to make LED 1 blink, the previous activity code use delay() function to make the LED turn on for 1 second and then turn off for 1 second.

But the downside of using delay() function is, everything must also stop and wait for the delay() function to finish. Thus when you push the button, if the delay() function is currently running, the LED 2 must wait for the delay to end. In microcontroller such as Node MCU there is no operating system to handle interrupt or multitasking. But luckily there is an alternative which is using millis() function.

In activity 2 we will use millis() function to make non-blocking function for blinking LED 1 and at the same time read digital input from push button to turn on LED 2.

2.1. Physical Connection

Use the same physical connection as activity 2

2.2 Upload code into NodeMCU

Use the code from the following code repository

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_2

2.3 Observed

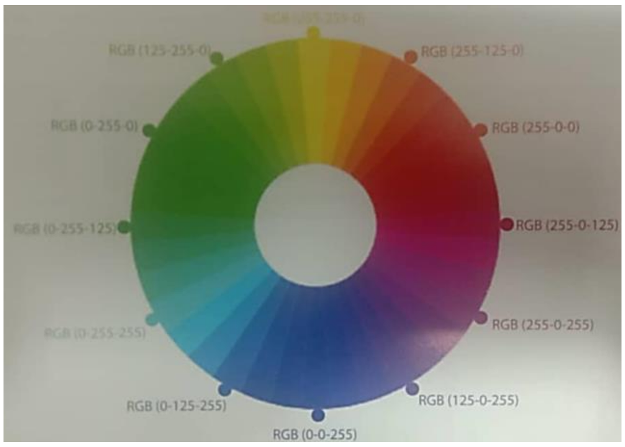
Now observed what happen when you push the button. Is there any delay to turn on LED 2 when you push the button?

____

Activity 3: Analog Output – RGB Light

An analog 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.

Node MCU 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 Node MCU 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 Physical Connection

First, we need to physically connect the RGB LED’s PIN to the Node MCU board. Kindly refer the following table

Node MCU Pins	RGB LED
D0	R
D1	G
D2	B
GND	COM

3.2 Upload RGB Light code into Node MCU

Use the code from the following code repository. You should see the RGB LED producing variety of colour due to different analogue signal.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_3

3.3 Play Around with Colour

Try to play around with color by modifying the value in this function

analogWrite(D0, random(0,255));

Or you can challenge yourself by writing Analog Output for buzzer. You can use **tone(D0, 1000);** function.

Activity 4: Analog Input (LM35 Sensor)

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.

4.1 Connect LM35 Temperature Sensor to NodeMCU Board

In this activity, you need to connect LM35 sensor pin to node MCU analog pin and LED. Kindly refer table below

Node MCU Pins	RGB LED
A0	LM35 S
3V3	LM35 VS
GND	LM35 GND
D0	LED L1
GND	LED GND

4.2 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 Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_4

4.3 Observe your project

After uploading the code, Node MCU will periodically print out the temperature reading to console. You should able to see current temperature reading from Arduino IDE console. To open console click **Tools > Serial Monitor**

Did the LED turn on if you touch the LM35 sensor? Why is that?

____

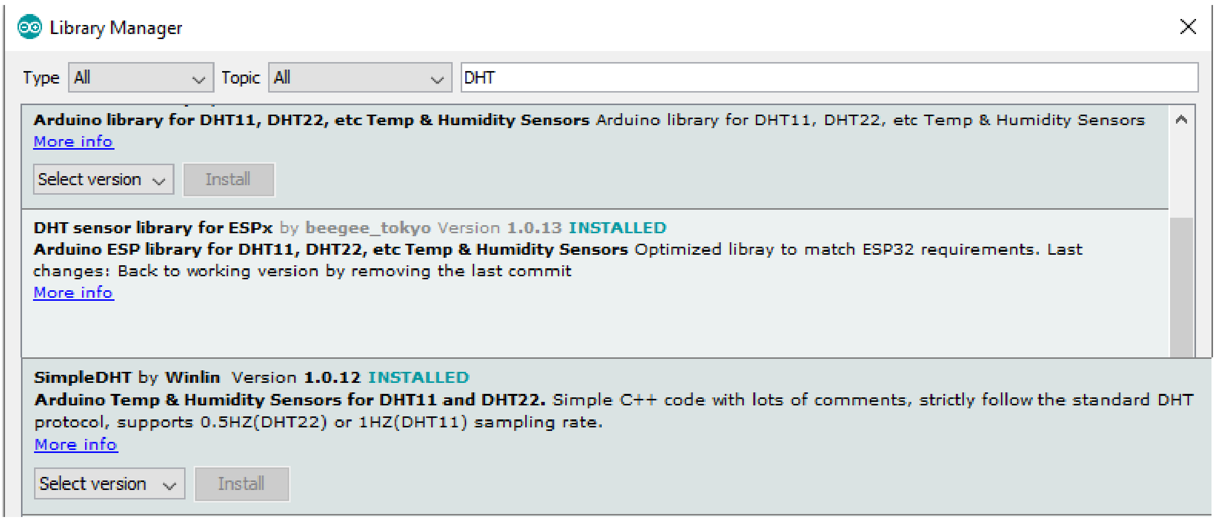
Activity 5: DH11 Sensor Module

In previous activity, the LM35 sensor produce a value in analog data, so we need to use analog input pin at Node MCU. There is also sensor in the market which convert the analog data to digital data before sending the reading to the microcontroller. Example of such sensor is DHT11, below is the technical specification of the sensor:

- Ultra-low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

5.1 Install DHT11 Sensor Library to Arduino Library

In this activity we will use DHT11 sensor to capture surrounding temperature and humidity, therefore we need to ensure our Arduino IDE has been installed with DHT11 libraries. To installed DHT11 library we need to open 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.



Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
More info
Select version Install

DHT sensor library for ESPx by beegee_tokyo Version 1.0.13 INSTALLED
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last changes: Back to working version by removing the last commit
More info

SimpleDHT by Winlin Version 1.0.12 INSTALLED
Arduino Temp & Humidity Sensors for DHT11 and DHT22. Simple C++ code with lots of comments, strictly follow the standard DHT protocol, supports 0.5HZ(DHT22) or 1HZ(DHT11) sampling rate.
More info
Select version Install

5.2 Physical Connection

In this activity, you need to connect DHT11 sensor module pin to node MCU. Kindly refer table below

Node MCU Pins	DH11
D4	D
3V3	VS
GND	GND

5.3 Upload the Code into Node MCU Board

After you have finished with physical connection, type below code into your Arduino IDE and upload it into Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_5

5.4 Observe the Sensor Value

You should able to see current temperature reading from DHT11 sensor on Arduino IDE console. To open console click Tools > Serial Monitor

What happen to the sensor reading when you touch the sensor?

____

Activity 6: Displaying Sensor Data via Build in Web Server

So far in our activity, we have read sensor data and print it out to the console. In this activity we will display the sensor data using Build-in web server inside Node MCU. The sensor data can be access using a web browser.

6.1 Setting up Internet Connection using Smartphone Hotspot or Wireless Access Point.

For this activity to work, the Node MCU must be connected to a Wifi Network. You can provide Wifi connectivity via your smartphone hotspot or using Wireless Access Point provided in the lab.

6.2 Physical Connection

In this activity, you need to connect DHT11 sensor module pin to node MCU just as previous activity. Kindly refer table below

Node MCU Pins	DH11
D4	D
3V3	VS
GND	GND

6.3 Upload the Code into Node MCU Board

After you have finished with physical connection, type below code into your Arduino IDE and upload it into Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_6

You need to modify the ssid and password variable to reflect your Wifi connection.
const char* ssid = "Your WIFI SSID";

const char* password = "Your WIFI Password";

6.4 Observe the Sensor Value from a Web Browser

You can observe the sensor value using your web browser, just type in the IP address assign to the Node MCU at the URL bar. You can find out the Node MCU IP address from the Node MCU console.

Activity 7: Sending Sensor Data via TCP Socket

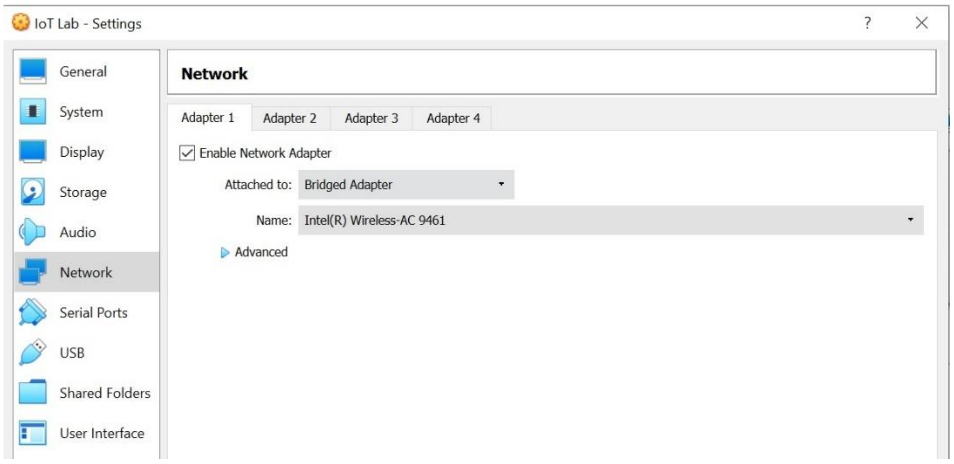
We also have the option to send sensor data to another host using TCP socket. Node MCU has built in library which we can use to ease the process of using TCP socket for communication. In this activity, we will use a python script running in a Linux machine to pull data sensor data from Node MCU using TCP Socket.

7.1 Setting up Virtual Machine

For this activity to work, you need to setup a virtual machine running with Linux. You need to install virtual box to your PC and import pre-made virtual machine (.ova) to your virtual box. The Virtual box installer and the virtual machine image (.ova) file can be obtained from the lab material folder as follows:

https://drive.google.com/drive/folders/1mRhOkEeIl8sB2lJpYAti_eZYo5Q0s26W?usp=sharing

Before you start your virtual machine, make sure you change the configuration of your virtual machine network adapter to bridged adapter. Kindly refer following figures.



Enable Network Adapter: ✓
Attached to: Bridged Adapter
Name: Intel(R) Wireless-AC 9461

After you have configured the network adapter, you can start your virtual machine. It takes several minutes for your virtual machine to boot up. **Make sure your PC and Node MCU is connected to the same Wi-Fi network**. After that you can use putty or any other SSH client to connect to your virtual machine. Please login to your virtual machine using this credential:

Username: user

Password: user123

After that, you need to create a Python file “client.py” in your home folder. The code can be obtained from the following code repository. You can create the file by using nano text editor (type nano at your linux terminal)

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_7

7.2 Physical Connection

After you have setup your virtual machine, you need to connect DHT11 sensor module pin to node MCU just as previous activity. Kindly refer table below

Node MCU Pins	DH11
D4	D
3V3	VS
GND	GND

7.3 Upload the Code into Node MCU Board

After you have finished with physical connection, type below code into your Arduino IDE and upload it into Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_7

You need to modify the ssid and password variable to reflect your Wifi connection.

```
const char* ssid = "Your WIFI SSID";

const char* password = "Your WIFI Password";
```

7.4 Observe the Sensor Value from a Linux Shell/Terminal

Once you have finish setting up your virtual machine and Node MCU, you need to modify some variable in the ‘’client.py” file in your linux machine. You need to change the IP address in the s.connect() function to reflect your Node MCU IP address. You can find out your Node MCU IP address from Arduino terminal monitor.

s.connect(('192.168.1.20', port))

After you have modify the IP address, you can begin to run the Python code in your Linux Machine. You can type this in your linux terminal:

```
python3 client.py
```

**Make Sure you are In your Linux home folder.

While the Python code is running, you should see the Python code polling the sensor data.

Activity 8: Pushing Sensor Data via HTTP Post

Other than using TCP socket to communicate the sensor data, we also have the option to push the sensor data from Node MCU to the Web services using HTTP Post.

In this activity, you need to use the Virtual machine which you have setup in the previous activity.

8.1 Modify the Web Code

Before we begin setting up the Node MCU, we need to find out the IP address of our VM. We can use the following command

```
$ifconfig
```

After you have determined the IP address of the VM, you need to modify the PHP code in your Linux machine. You can modify using this command in your terminal:

```
$sudo nano /var/www/html/esp-data.php
```

```
$sudo nano /var/www/html/post-esp-data.php
```

You need to modify the **$servername = "192.168.0.208"**; variable so that it reflects your VM IP.

8.2 Physical Connection

After you have setup your virtual machine, you need to connect DHT11 sensor module pin to node MCU just as previous activity. Kindly refer table below

Node MCU Pins	DH11
D4	D
3V3	VS
GND	GND

8.3 Upload the Code into Node MCU Board

After you have finished with physical connection, type below code into your Arduino IDE and upload it into Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_8

You need to modify the ssid and password variable to reflect your Wifi connection.

```
const char* ssid = "Your WIFI SSID";
```

```
const char* password = "Your WIFI Password";
```

You also need to modify the ServerName to reflect your VM IP

const char* serverName = "http://192.168.0.208/post-esp-data.php";

8.4 Observe the Sensor Value from Your Web Browser

Once you have finish setting up your virtual machine and Node MCU, you can observe the value been send to your VM by putting your VM IP to your web browser URL

http://192.168.0.208/esp-data.php

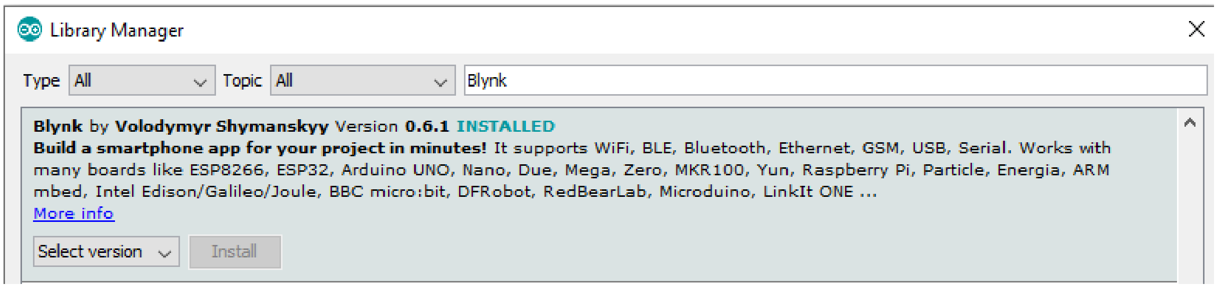
http://192.168.0.208/phpmyadmin

Activity 9: Using Node MCU with Blynk Platform

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.

9.1 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


Library Manager

Type All Topic All Blynk

Blynk by **Volodymyr Shymanskyy** Version **0.6.1** **INSTALLED**
Build a smartphone app for your project in minutes! It supports WiFi, BLE, Bluetooth, Ethernet, GSM, USB, Serial. Works with many boards like ESP8266, ESP32, Arduino UNO, Nano, Due, Mega, Zero, MKR100, Yun, Raspberry Pi, Particle, Energia, ARM mbed, Intel Edison/Galileo/Joule, BBC micro:bit, DFRobot, RedBearLab, Microduino, LinkIt ONE ...
More info

9.2 Physical Connection

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 table below

Node MCU Pins	Sensor/LED
D5	DHT11 D
3V3	DHT11 VS
GND	DHT11 GND
D0	LED L1
D1	LED L2
D2	LED L3
D3	LED L4
GND	LED GND

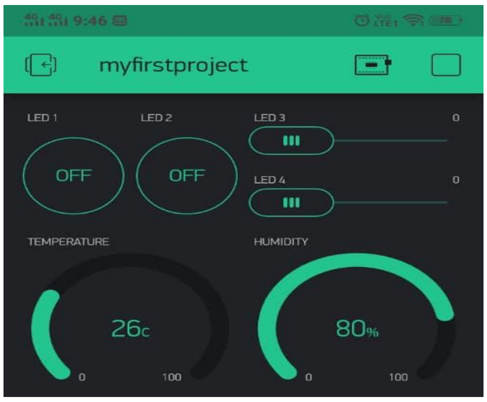
9.3 Install Blynk App to your smartphone and create a new project

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.

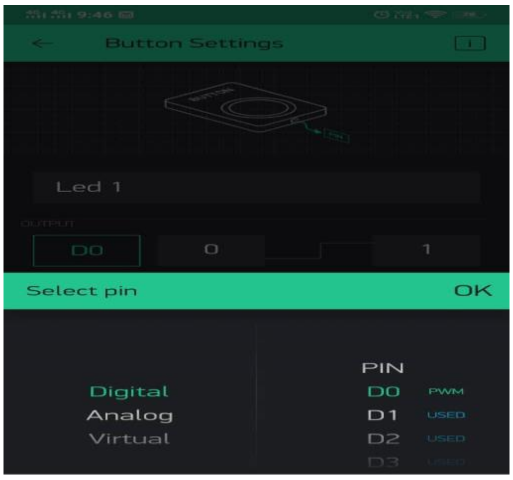
After you have completed the registration, you need to create your FirstProject. Blynk will email to you the Authentication key. You need to take note on the key as you need to put it in your Arduino code.

9.4 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:

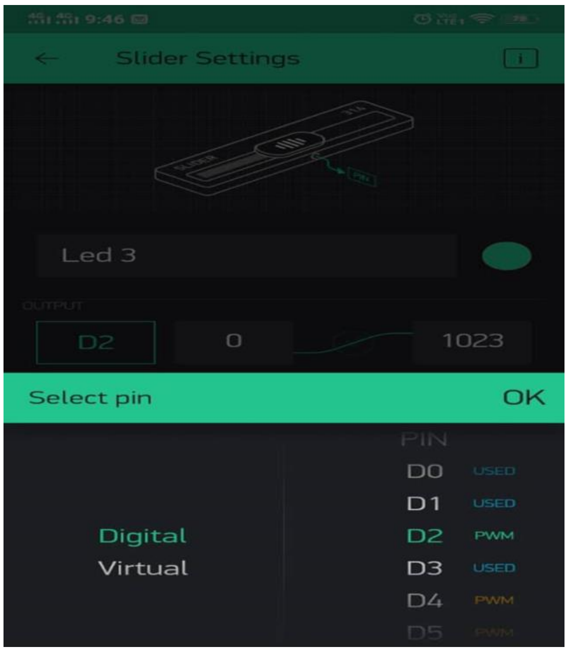


After that, you need to configure your button. Click on your button and name it LED 1 and then click on the Output and select the Pin as Digital and D0, please refer diagram below.



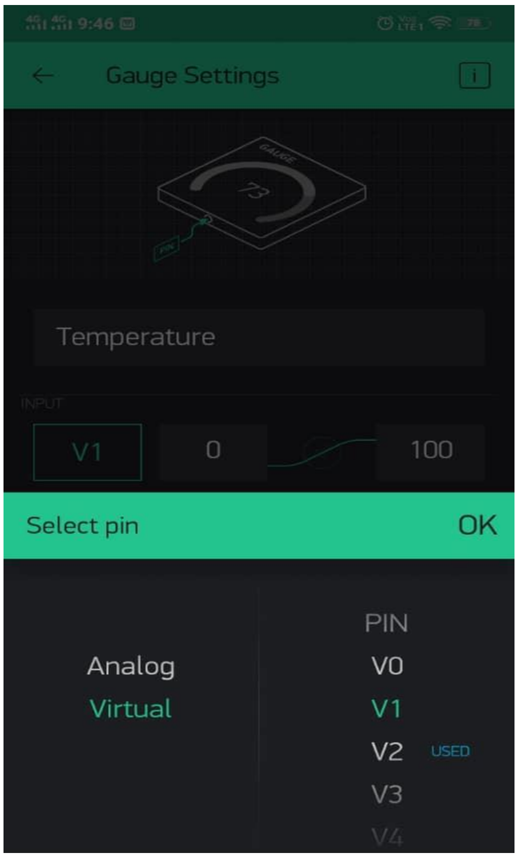
Now repeat the step for the second button, name it LED 2 and select the Pin as Digital D1.

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.

9.5 Setup Smartphone Hotspot or Access Point

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.

9.6 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.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_9

Activity 10: Pushing Sensor Data to Google Firebase

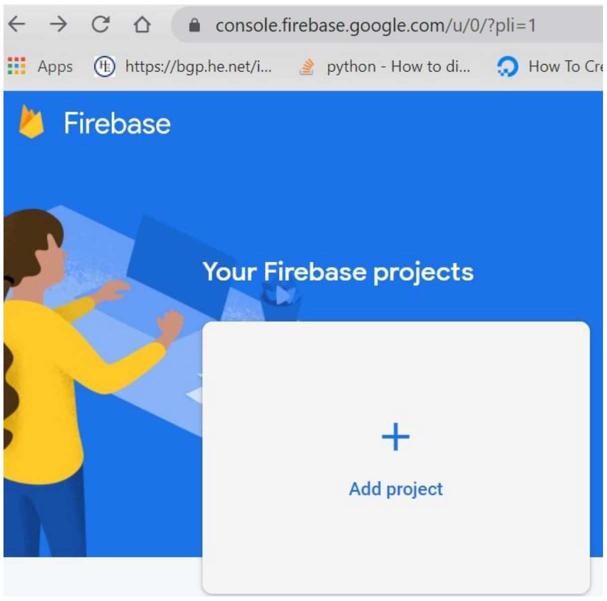
As an alternative to Blynk platform, we can also use Google Serverless platform which is Firebase to store sensor data from Node MCU. Firebase is a platform developed by Google for creating mobile and web applications. In this activity, we will push sensor data from Node MCU and store it in Firebase Realtime Database.

10.1 Register with Google Firebase

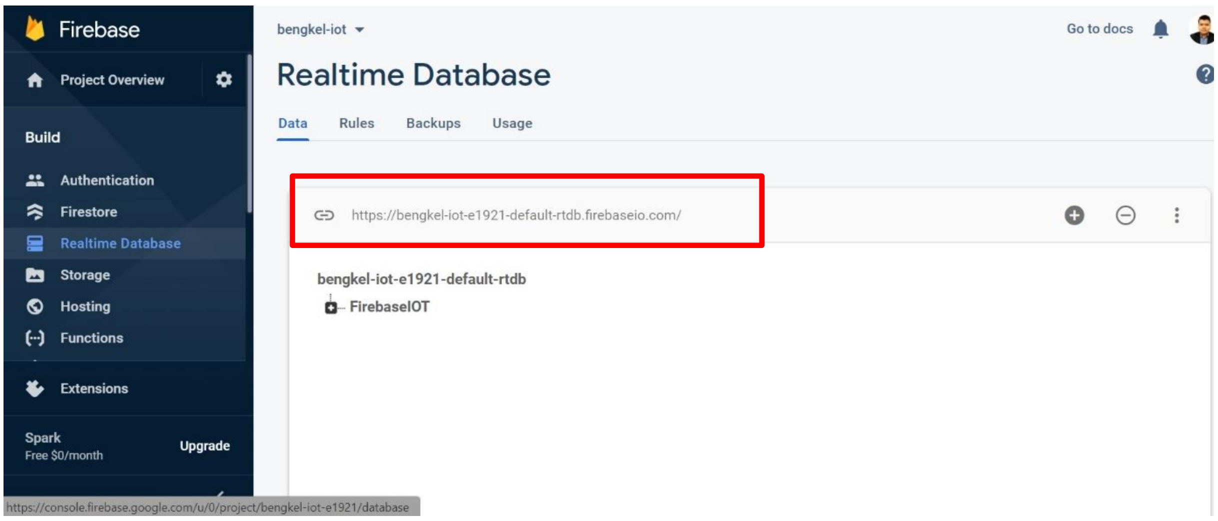
In this activity you need to be registered with Google Firebase to use their Realtime Database service. You can register and visit your firebase developer console here:

https://console.firebase.google.com/u/0/

After you have registered your firebase account, you need to create your first Firebase Projects. You can create your project by clicking "Add Project" button at your firebase console.



After that, you need to create your Realtime Database service. You can create new Realtime Database by clicking the Realtime Database option on the left panel. After you have successfully created your Realtime Database, please take note on your database destination URL. You need to use the URL to modify the Arduino code. Please refer the highlighted field in figure below.

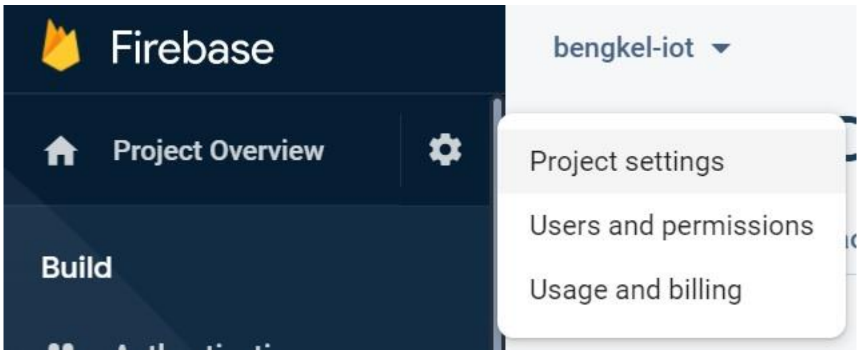


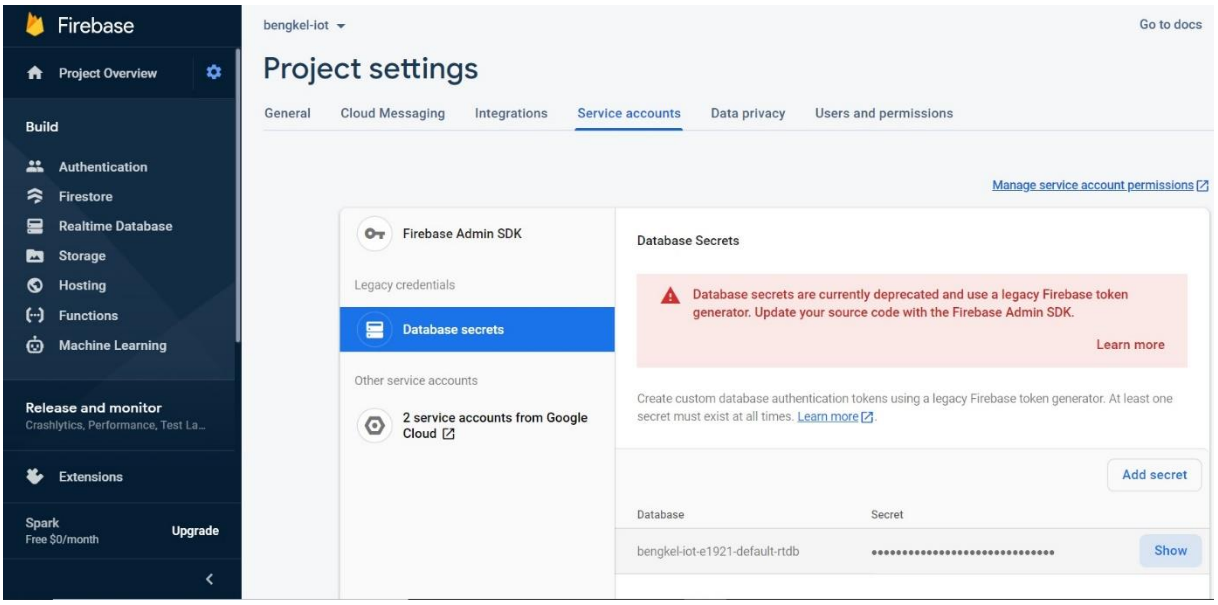
https://bengkel-iot-e1921-default-rtdb.firebaseio.com/

bengkel-iot-e1921-default-rtdb
FirebaseIOT

https://console.firebase.google.com/u/0/project/bengkel-iot-e1921/database

The next step will be to find out your Realtime Database secret. The secret is also needed when you are modifying the Arduino code later. To view your database secret, you need to click Project Overview > Project Settings > Service Accounts > Database Secret. Kindly refer figure below.





Database Secrets

Database secrets are currently deprecated and use a legacy Firebase token generator. Update your source code with the Firebase Admin SDK.

Learn more

Create custom database authentication tokens using a legacy Firebase token generator. At least one secret must exist at all times. Learn more

Database	Secret
bengkel-iot-e1921-default-rtdb	••••••••••••••••••••••••••••••••

10.2 Physical Connection

After you have setup your Realtime database in your Firebase Console, you need to connect DHT11 sensor module pin to node MCU. Kindly refer table below

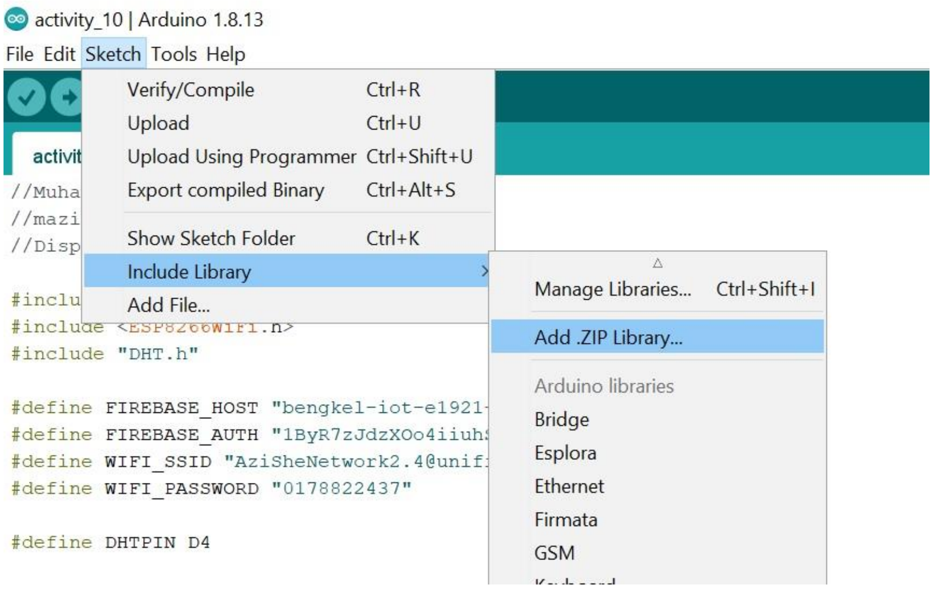
Node MCU Pins	DH11
D4	D
3V3	VS
GND	GND

10.3 Install Firebase Library to Arduino IDE

After you have setup your Realtime database in your Firebase Console, you need to install Firebase library to Arduino. First you must download the firebase library from URL below. You can also obtain the library code from our lab shared folder.

https://github.com/mobizt/Firebase-ESP8266/archive/refs/heads/master.zip

After you have downloaded the firebase library code, you need to add it to the Arduino IDE. To add the library, you need to click Sketch > Include Library > Add .Zip Library.


activity_10 | Arduino 1.8.13
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Bridge
Esplora
Ethernet
Firmata
GSM
activit

```
//Muha
//mazi
//Disp

#inclu
#include <ESP8266WiFi.h>
#include "DHT.h"

#define FIREBASE_HOST "bengkel-iot-e1921
#define FIREBASE_AUTH "1ByR7zJdzXOo4iiuh
#define WIFI_SSID "AziSheNetwork2.4@unif
#define WIFI_PASSWORD "0178822437"

#define DHTPIN D4
```

10.4 Upload the Code into Node MCU Board

After you have finished with physical connection and adding Firebase library, type below code into your Arduino IDE and upload it into Node MCU board.

Code Repository: https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_10

You need to modify the ssid and password variable to reflect your Wifi connection.

const char* ssid = "Your WIFI SSID";

const char* password = "Your WIFI Password";

You also need to modify the FIREBASE_HOST and FIREBASE_AUTH to reflect your Firebase’s Realtime Database. FIREBASE_HOST refers to your Realtime Database URL and FIREBASE_AUTH refers to your database secret.

#define FIREBASE_HOST "Your Firebase URL"

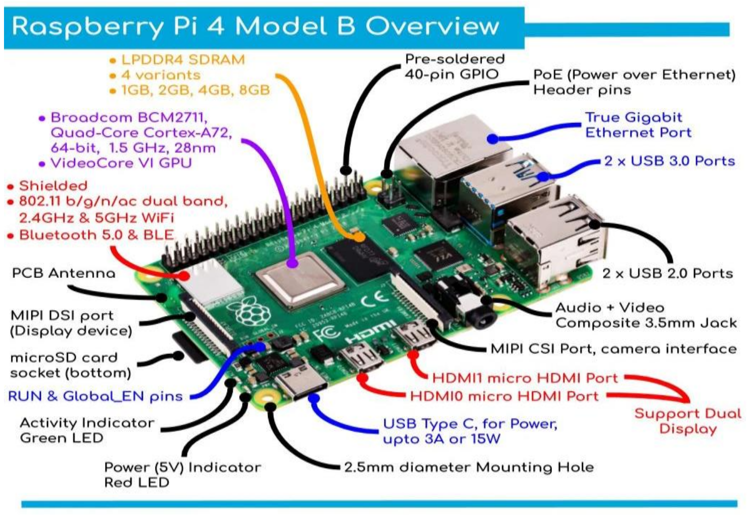
#define FIREBASE_AUTH "Your Firebase Secret"

10.5 Observe the Sensor Value from Your Firebase Console

After you have finish configuring your Firebase and your Node MCU, you can observe the sensor value being push to your Firebase database from your firebase console.

Activity 11: Introduction to Raspberry PI

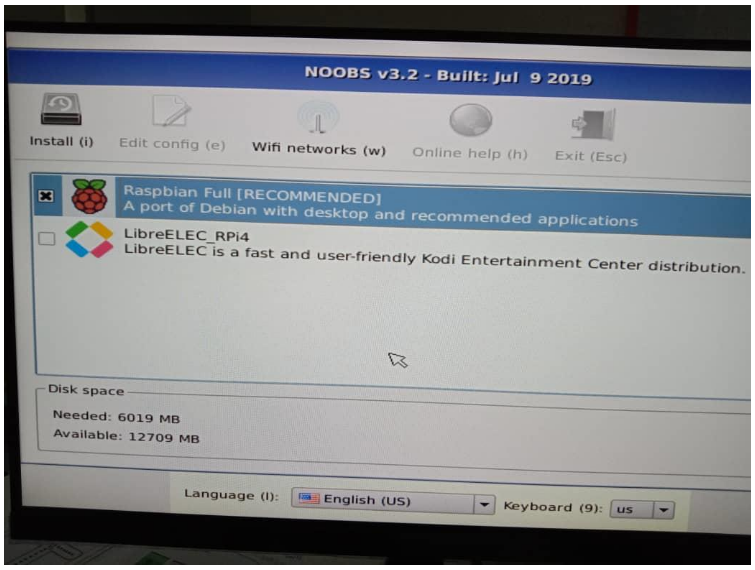
Besides Node MCU, FSKM Lab is also equipped with Raspberry PI 4 for IoT lab activity. Raspberry Pi 4 is a single-board PC equipped with Wi-Fi interface, GPIO pin and various other features. It is very suitable to be used in IoT project. It is widely used in many areas, such as for weather monitoring,[20] because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices. This is the specifications of Raspberry Pi 4 available in FSKM Lab:


Raspberry Pi 4 Model B Overview
• LPDDR4 SDRAM
• 4 variants
• 1GB, 2GB, 4GB, 8GB
Pre-soldered 40-pin GPIO
PoE (Power over Ethernet) Header pins
• Broadcom BCM2711, Quad-Core Cortex-A72, 64-bit, 1.5 GHz, 28nm
• VideoCore VI GPU
True Gigabit Ethernet Port
2 x USB 3.0 Ports
• Shielded
• 802.11 b/g/n/ac dual band, 2.4GHz & 5GHz WiFi
• Bluetooth 5.0 & BLE
PCB Antenna
2 x USB 2.0 Ports
MIPI DSI port (Display device)
Audio + Video Composite 3.5mm Jack
microSD card socket (bottom)
MIPI CSI Port, camera interface
RUN & Global_EN pins
HDMI1 micro HDMI Port
HDMI0 micro HDMI Port
Support Dual Display
Activity Indicator Green LED
USB Type C, for Power, upto 3A or 15W
Power (5V) Indicator Red LED
2.5mm diameter Mounting Hole

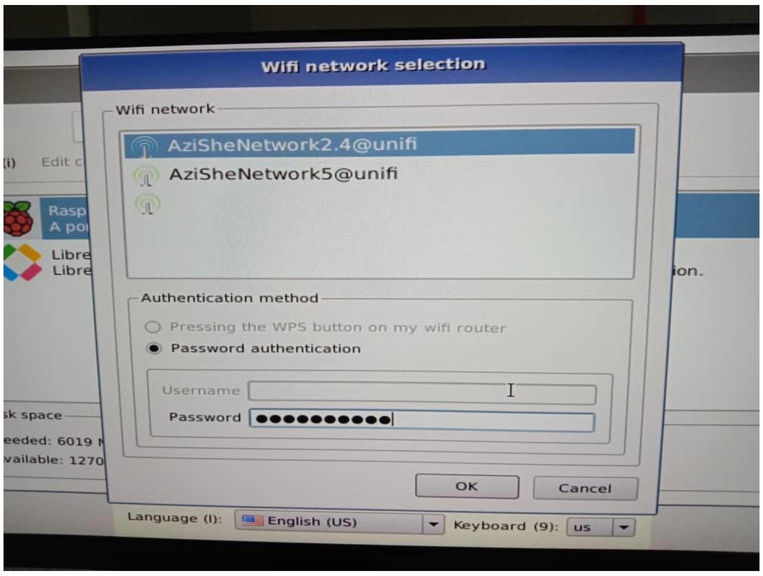
11.1 Connecting your Raspberry PI and Initial Start-up

To start using the raspberry Pi 4, you need to power up the raspberry Pi 4 using Type-C USB adapter. You need also need to connect the raspberry pi to monitor in the lab using Mini-HDMI to VGA adapter. The monitor will display the GUI interface which are needed for initial start-up setup. Please refer figure below for peripherals connection.

After the system has been booted up, you will be greeted by NOOBS program to perform the initial setup for the raspberry Pi 4. The first window will ask you to choose Operating System distribution to be installed, please choose Raspbian Full.


NOOBS v3.2 - Built: Jul 9 2019
Install (i)
Edit config (e)
Wifi networks (w)
Online help (h)
Exit (Esc)
Raspbian Full [RECOMMENDED]
A port of Debian with desktop and recommended applications
LibreELEC_RPi4
LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution.
Disk space
Needed: 6019 MB
Available: 12709 MB
Language (l): English (US)
Keyboard (9): us

After that you need to connect the raspberry Pi 4 to the Wi-Fi connection, please provide the password to connect to the Wi-Fi Network. After that, please click the Install icon to continue with the operating system installation process. After that you will be shown an installation window, the installation may take a while to complete. Please refer figure below


Wifi network selection

Wifi network

AziSheNetwork2.4@unifi

AziSheNetwork5@unifi

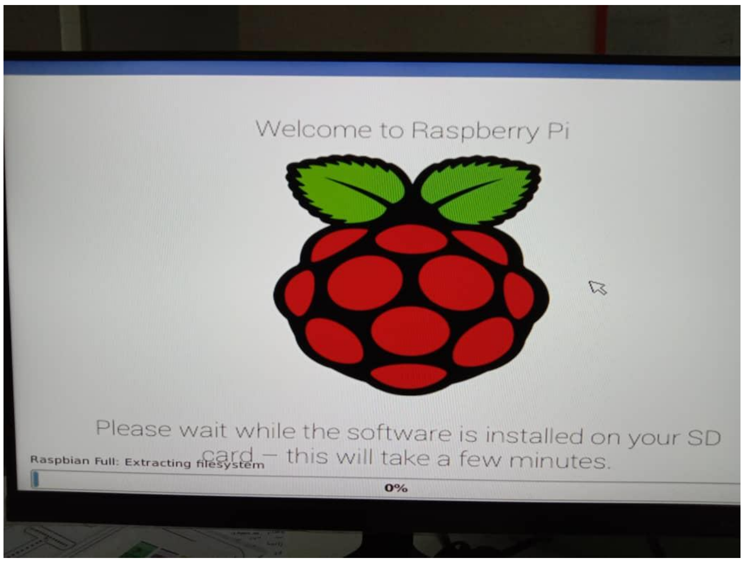
Authentication method

○ Pressing the WPS button on my wifi router

● Password authentication

Username:

Password: ●●●●●●●●●●●

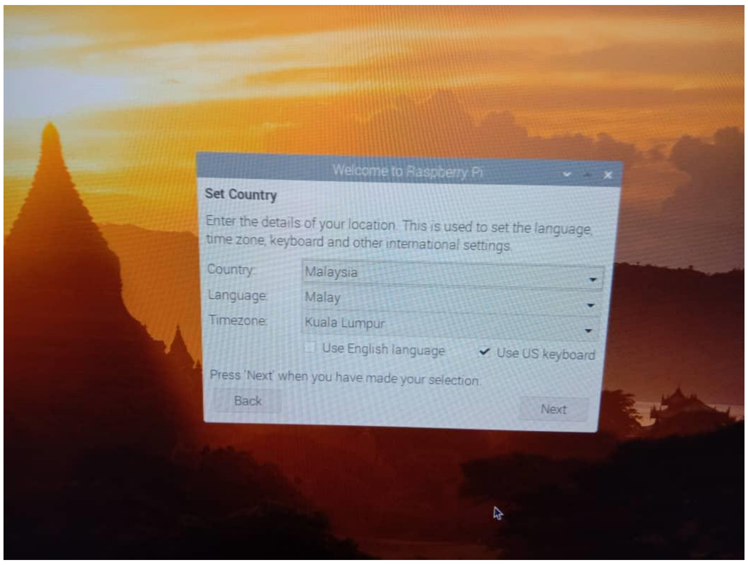

Welcome to Raspberry Pi

Please wait while the software is installed on your SD card – this will take a few minutes.

Raspbian Full: Extracting filesystem

0%

After the operating system installation process has completed, you will be prompted with a window to choose country of Installation. Please choose Malaysia as a country and use US keyboard. After that you will be prompted to change system password, just ignore the prompt and click next to finish the setup process. Please refer figure below for references.



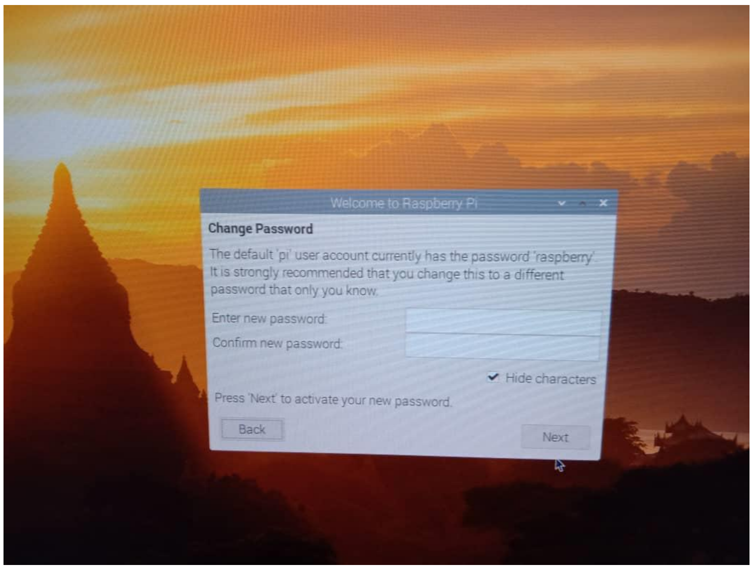
Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

Country: Malaysia
Language: Malay
Timezone: Kuala Lumpur

Use English language ✓ Use US keyboard

Press 'Next' when you have made your selection.



Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know.

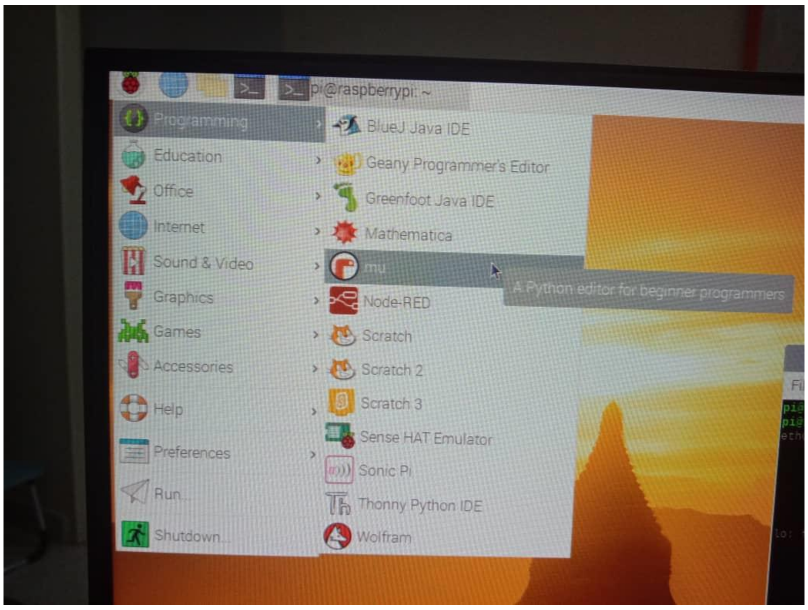
Enter new password:
Confirm new password:

✓ Hide characters

Press 'Next' to activate your new password.

11.2 Exploring Raspbian OS

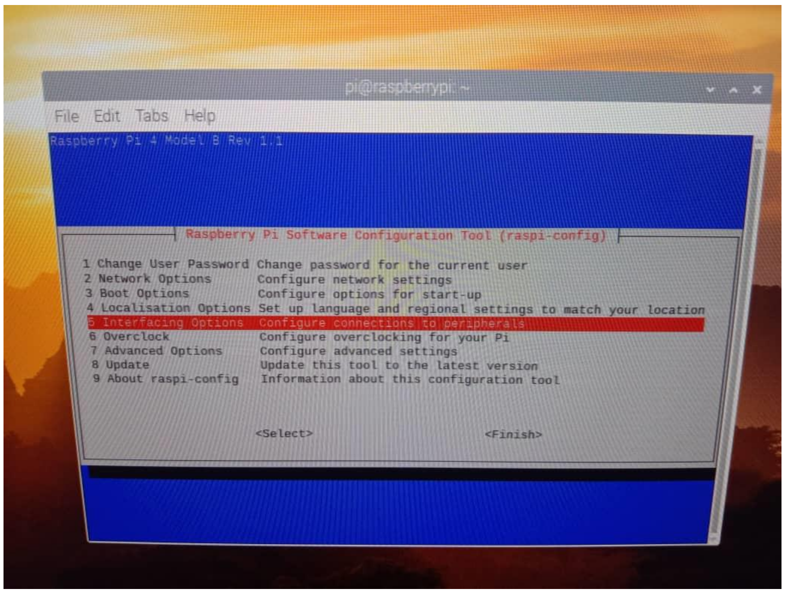
After the initial setup has finished, you will be brought to Raspbian desktop interface. Please explore the Raspbian interface and find out the software which already pre-installed with Raspbian OS.



11.3 Accessing your Raspberry Pi using SSH Client

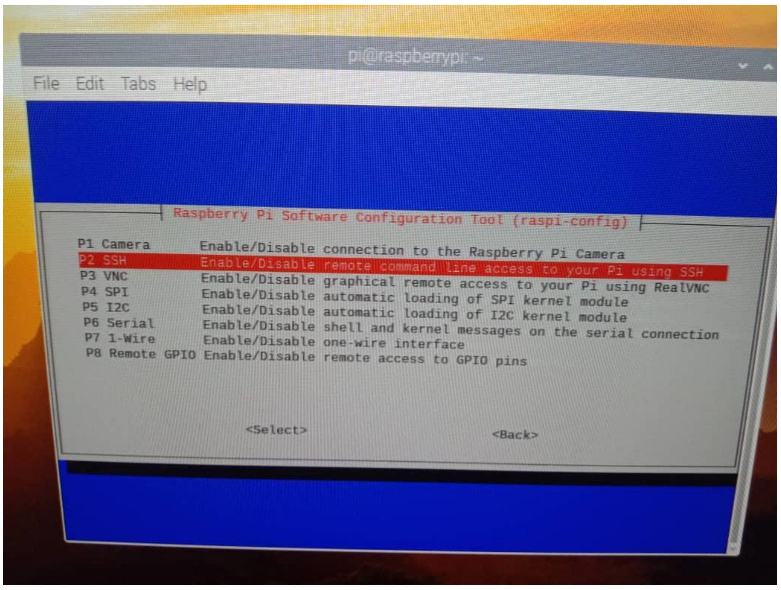
After you have finished exploring Raspbian OS, we need to setup SSH access for your Raspberry Pi 4. This is essential for the next activity. The first step is to enable SSH server on your Raspberry Pi. To enable SSH, you need to open up Raspbian Terminal. After that you need to run this command:

```
$ sudo raspi-config
```



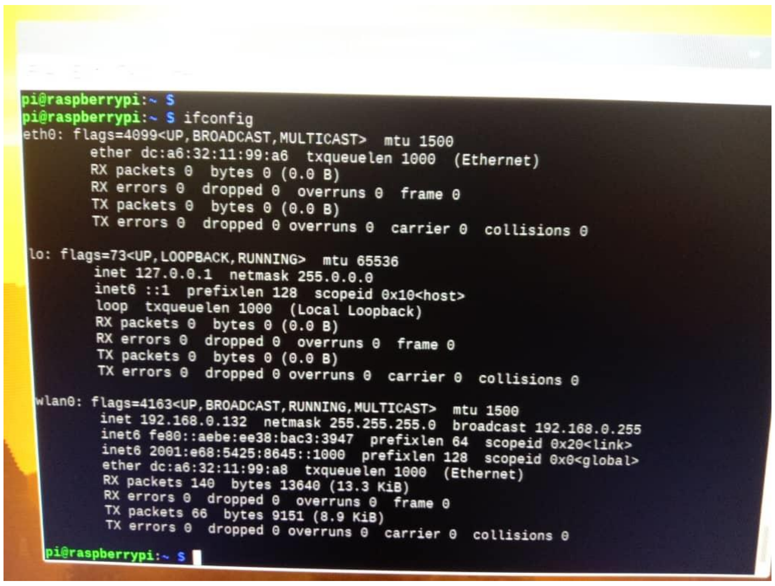
```
Raspberry Pi 4 Model B Rev 1.1

        Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your location
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

                  <Select>                  <Finish>
```

When you run the command on the terminal, you will be prompted with a menu. Please choose option 5: Interfacing Options.



```
pi@raspberrypi: ~
File  Edit  Tabs  Help

┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

P1 Camera       Enable/Disable connection to the Raspberry Pi Camera
P2 SSH          Enable/Disable remote command line access to your Pi using SSH
P3 VNC          Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI          Enable/Disable automatic loading of SPI kernel module
P5 I2C          Enable/Disable automatic loading of I2C kernel module
P6 Serial       Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire       Enable/Disable one-wire interface
P8 Remote GPIO  Enable/Disable remote access to GPIO pins

              <Select>                     <Back>
```

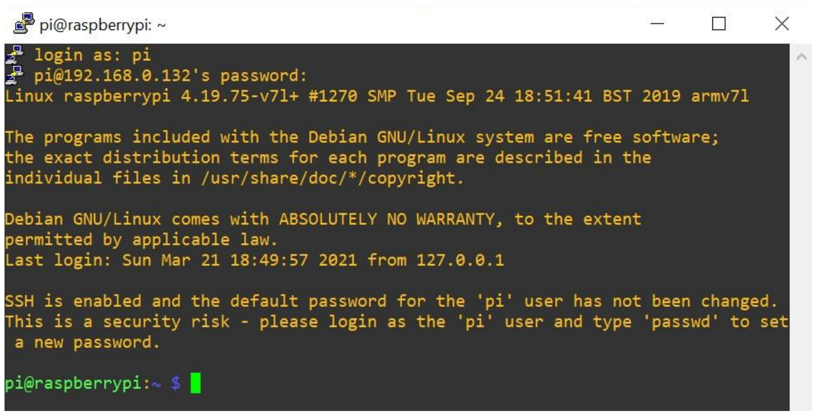
After that choose option P2: SSH to enable SSH on your raspberry Pi 4.



```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether dc:a6:32:11:99:a6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.132  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::aebe:ee38:bac3:3947  prefixlen 64  scopeid 0x20<link>
        inet6 2001:e68:5425:8645::1000  prefixlen 128  scopeid 0x0<global>
        ether dc:a6:32:11:99:a8  txqueuelen 1000  (Ethernet)
        RX packets 140  bytes 13640 (13.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 66  bytes 9151 (8.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $
```

After you have successfully enabled OpenSSH server on your raspberry Pi 4. You need to know the IP address of your raspberry Pi. To check the Ip address you can use the following command on terminal

```
$ sudo ifconfig
```



```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.132's password:
Linux raspberrypi 4.19.75-v7l+ #1270 SMP Tue Sep 24 18:51:41 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar 21 18:49:57 2021 from 127.0.0.1

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $
```

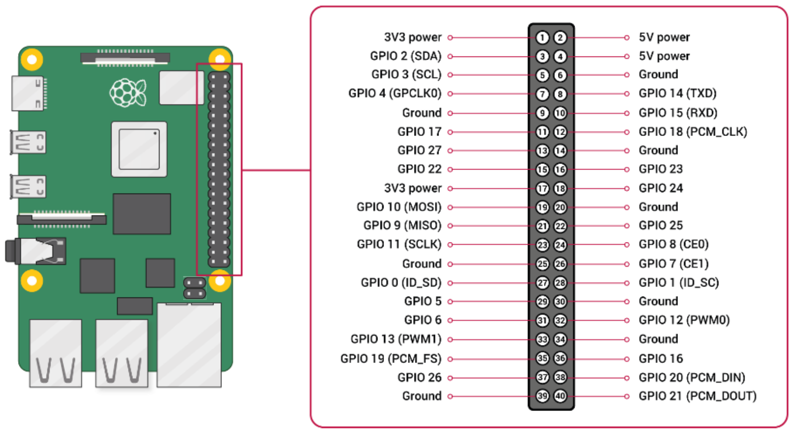
To access your raspberry PI using SSH, you can use SSH client such as Putty on your PC. You can connect using the IP address which you have check earlier and use the following credentials **Username:** pi

Password: raspberry

If you can access your raspberry Pi 4 devices from your PC using SSH client, you can proceed to next activity.

Activity 12: Using Raspberry PI GPIO

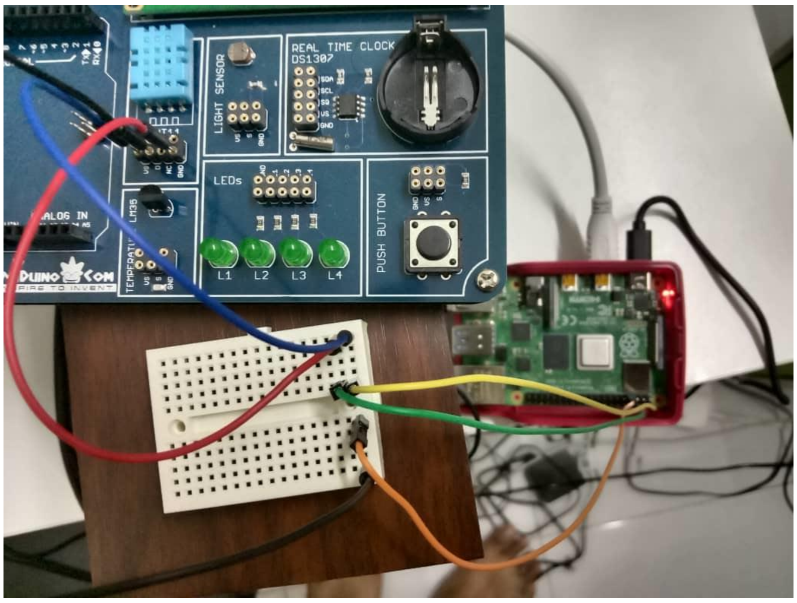
Raspberry Pi 4 is equipped with pre-soldered 40-pin GPIO. The GPIO pin can be used to connect various sensors and actuators to your raspberry Pi 4. The following figure shows the GPIO pin configuration for Raspberry PI. In this activity we will connect DHT11 sensor to raspberry pi GPIO pin and print out the sensor reading to the terminal.


3V3 power 1 2 5V power
GPIO 2 (SDA) 3 4 5V power
GPIO 3 (SCL) 5 6 Ground
GPIO 4 (GPCLK0) 7 8 GPIO 14 (TXD)
Ground 9 10 GPIO 15 (RXD)
GPIO 17 11 12 GPIO 18 (PCM_CLK)
GPIO 27 13 14 Ground
GPIO 22 15 16 GPIO 23
3V3 power 17 18 GPIO 24
GPIO 10 (MOSI) 19 20 Ground
GPIO 9 (MISO) 21 22 GPIO 25
GPIO 11 (SCLK) 23 24 GPIO 8 (CE0)
Ground 25 26 GPIO 7 (CE1)
GPIO 0 (ID_SD) 27 28 GPIO 1 (ID_SC)
GPIO 5 29 30 Ground
GPIO 6 31 32 GPIO 12 (PWM0)
GPIO 13 (PWM1) 33 34 Ground
GPIO 19 (PCM_FS) 35 36 GPIO 16
GPIO 26 37 38 GPIO 20 (PCM_DIN)
Ground 39 40 GPIO 21 (PCM_DOUT)

12.1 Physical Connection

Before we can begin, we need to physically connect the DHT11 sensor from the MyDuino IoT kit to the raspberry Pi. Kindly refer table and Figure below

Raspberry Pi GPIO Pins	DHT11
GPIO 4	D
5V Power	VS
GND	GND



12.2 Create Python Code

After you have finished with physical connection. You need to access your Raspberry Pi 4 using SSH as in activity 11. After you has successfully accessed the raspberry pi terminal you can proceed with the following steps:

We’ll be using the Adafruit DHT11 Python library. You can download the library using Git, so if you don’t have Git installed on your Pi already, enter this at the command prompt:

sudo apt-get install git-core

Note: If you get an error installing Git, run sudo apt-get update and try it again.

To install the Adafruit DHT11 library:

1. Enter this at the command prompt to download the library:

git clone https://github.com/adafruit/Adafruit_Python_DHT.git

2. Change directories with:

cd Adafruit_Python_DHT

3. Now enter this:

sudo apt-get install build-essential python-dev

4. Then install the library with:

sudo python setup.py install

5. Create "read.py" file in your home directory

nano read.py

you can get the code from this repository:
https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_12

12.3 Observe the Sensor Value from Your Raspberry Pi 4 Terminal

After you have finish with the python code, you can run it by using this command
cd ~
python read.py

If you everything went smoothly you should able to see the sensor value is printed out on the terminal.

Activity 13: Displaying Sensor via Flask using Raspberry PI

To make things more interesting, we can display the sensor value using a simple web server build using Flask library in Python. Flask is a micro web framework written in Python. It is classified as a microframework because it does not require tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Thus, it is suitable for lightweight IoT project.

13.1 Physical Connection

Before we can begin, we need to physically connect the DHT11 sensor from the MyDuino IoT kit to the raspberry Pi. The pin arrangement as previous activity, kindly refer table below

Raspberry Pi GPIO Pins	DHT11
GPIO 4	D
5V Power	VS
GND	GND

13.2 Create Python Code

After the physical has been sorted, we can begin to create the python code. Please access your raspberry pi terminal via SSH. You need to create "web.py" code in your home directory, use the following command:

```
nano /home/pi/web.py
```

Paste or type in the code in the following repository:
https://github.com/mebikarbonat/iot_workshop_trainers/tree/main/activity_13

After that we need to run the python code using the following command:

```
cd ~
python read.py
```

13.3 Observe the Sensor Value from Your Raspberry Pi 4 Terminal

To view the sensor data, you need to open web browser in your PC and type in your raspberry pi IP on the URL. If you everything went smoothly you should able to see the sensor value printed in JSON format.

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