

Student name: Hoang Long Tran

Student ID: s223128143

SIT225: Data Capture Technologies

Activity 2.1: Working with sensor - DHT22

DHT22 is a temperature and humidity sensor.

Hardware Required

Arduino Board

DHT22 sensor

USB cable

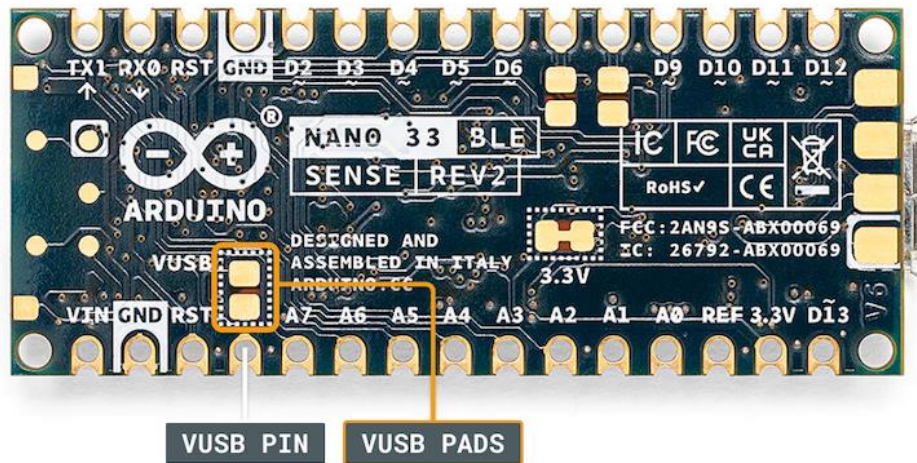
Software Required

Arduino programming environment

Known issue, action required

Arduino Nano 33 IoT board operates on 3.3 V, it needs to be arranged to make it 5 V. The Arduino board has a pin called VUSB or VBUS and there are 2 pads next to the pin. **These two pads must be shorted to enable the pin** (see detail here

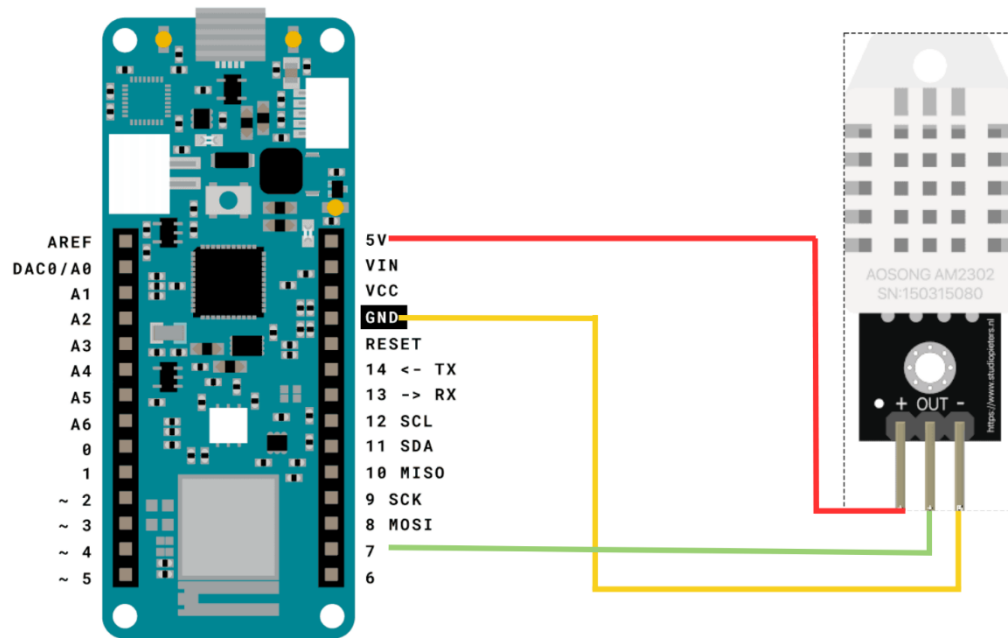
<https://support.arduino.cc/hc/en-us/articles/360014779679-Enable-5-V-power-on-the-VUSB-or-VBUS-pin-on-Nano-boards>).



To test, you can use a wire and connect these 2 pads manually by hand to see if data is coming through DHT22 sensor. For long data collection, you will need to solder the wire permanently to connect the pads. Seeking help from tutors there is an on-campus facility called Maker Space where you can do it.

Steps:

Step	Action
1	<p>Connect your DHT22 Temperature and Humidity Sensor to the Arduino board. Note that the pin layout in the image below may look different than the board you may have.</p>



- Pick a red male-female jumper wire and attach the female end to pin 1 (VCC pin) on the sensor. Plug the male end into the Arduino board's 5V power pin.
- Pick a blue male-female jumper wire and attach the female end to pin 2 (DATA pin) on the sensor. Plug the male end into the Arduino board's digital data pin 2.
- Pick a black male-female jumper wire and attach the female end to pin 4 (GND) on the sensor. Plug the male end into the Arduino board's GND pin.
- Sensor's pin 3 is not used.

2 Connect your Arduino board to your computer using the USB cable.

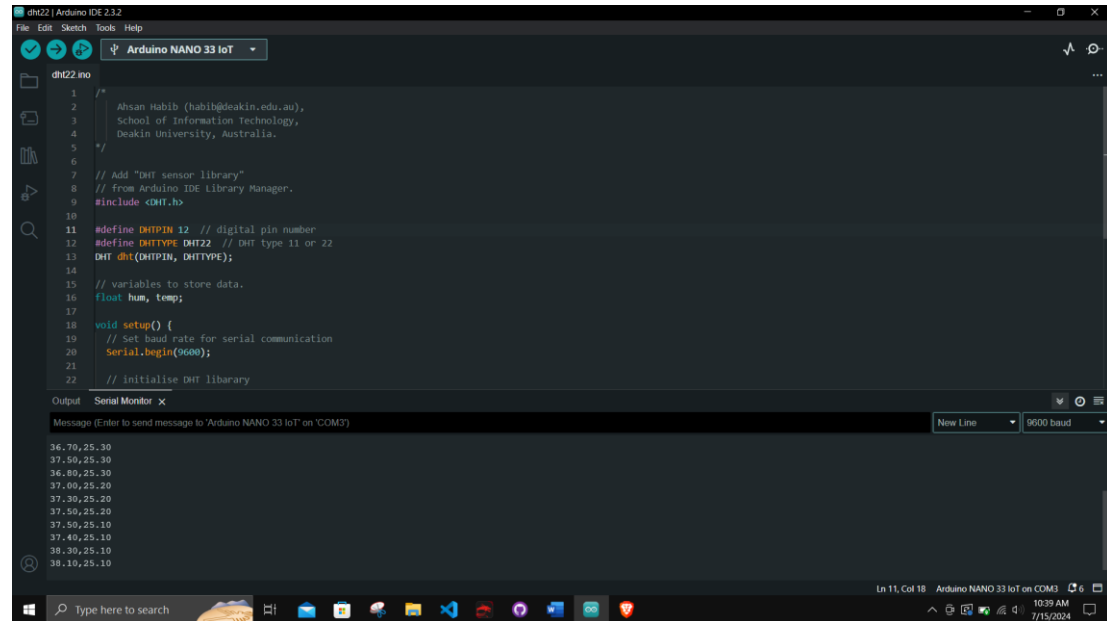
3 Write an Arduino sketch (or download it from https://github.com/deakin-deep-dreamer/sit225/blob/main/week_2/sketch_dht22.ino) which looks like below. Compile the code in Arduino IDE, deploy to the board and observe output in the Arduino IDE serial monitor.

	<pre> 6 7 // Add "DHT sensor library" 8 #include <DHT.h> 9 10 #define DHTPIN 2 // digital pin number 11 #define DHTTYPE DHT22 // DHT type 11 or 22 12 DHT dht(DHTPIN, DHTTYPE); 13 14 // variables to store data. 15 float hum, temp; 16 17 void setup() { 18 // Set baud rate for serial communication 19 Serial.begin(9600); 20 21 // initialise DHT library 22 dht.begin(); 23 } 24 25 void loop() { 26 // read data 27 hum = dht.readHumidity(); 28 temp = dht.readTemperature(); 29 30 // Print data to serial port - a long way 31 // 32 // Serial.print("Humid: "); 33 // Serial.print(hum); 34 // Serial.print(" %, Temp: "); 35 // Serial.print(temp); 36 // Serial.println(" Celsius"); 37 38 // Print data to serial port - a compact way 39 Serial.println(String(hum) + "," + String(temp)); 40 41 // wait a while 42 delay(15*1000); 43 } </pre>
4	<p>Question: A spec of the DHT22 sensor is given in the link below. It mentions that the sampling rate is 0.5 Hz.</p> <p>https://lastminuteengineers.com/dht11-dht22-arduino-tutorial</p> <ol style="list-style-type: none"> What does the sampling rate mean? Where is this used in the Arduino code? <p>Answer:</p> <ol style="list-style-type: none"> The sampling rate means the interval of time the sensor can take in data. The measurement is in Hz, but we can convert it to seconds by $1/0.5 = 2$ seconds. The `delay` in the Arduino code pause the program for enough time for the dht sensor to capture the next interval of data.

5

Question: Take a screenshot of your Serial Monitor displaying temperature & humidity sensor data logs. Add the image here.

Answer:



The screenshot shows the Arduino IDE 2.3.2 interface. The top pane displays the sketch file `dht22.ino` with the following code:

```
1  /*
2   * Ahsan Habib (habib@deakin.edu.au),
3   * School of Information Technology,
4   * Deakin University, Australia.
5   */
6
7  // Add "DHT sensor library"
8  // from Arduino IDE Library Manager.
9  #include <DHT.h>
10
11 #define DHTPIN 12 // digital pin number
12 #define DHTTYPE DHT22 // DHT type 11 or 22
13 DHT dht(DHTPIN, DHTTYPE);
14
15 // variables to store data.
16 float hum, temp;
17
18 void setup() {
19   // Set baud rate for serial communication
20   Serial.begin(9600);
21   // initialise DHT library
22 }
```

The bottom pane shows the Serial Monitor output, displaying a series of temperature and humidity readings separated by commas, such as `36.70,25.30`, `37.50,25.30`, `36.80,25.30`, `37.60,25.20`, `37.30,25.20`, `37.50,25.20`, `37.50,25.10`, `37.40,25.10`, `38.30,25.10`, and `38.10,25.10`. The Serial Monitor is set to 9600 baud.

Activity 2.2: Working with sensor - HC-SR04

HC-SR04 is an Ultrasonic sensor.

Hardware Required

Arduino Board

HC-SR04 Ultrasonic sensor

USB cable

Software Required

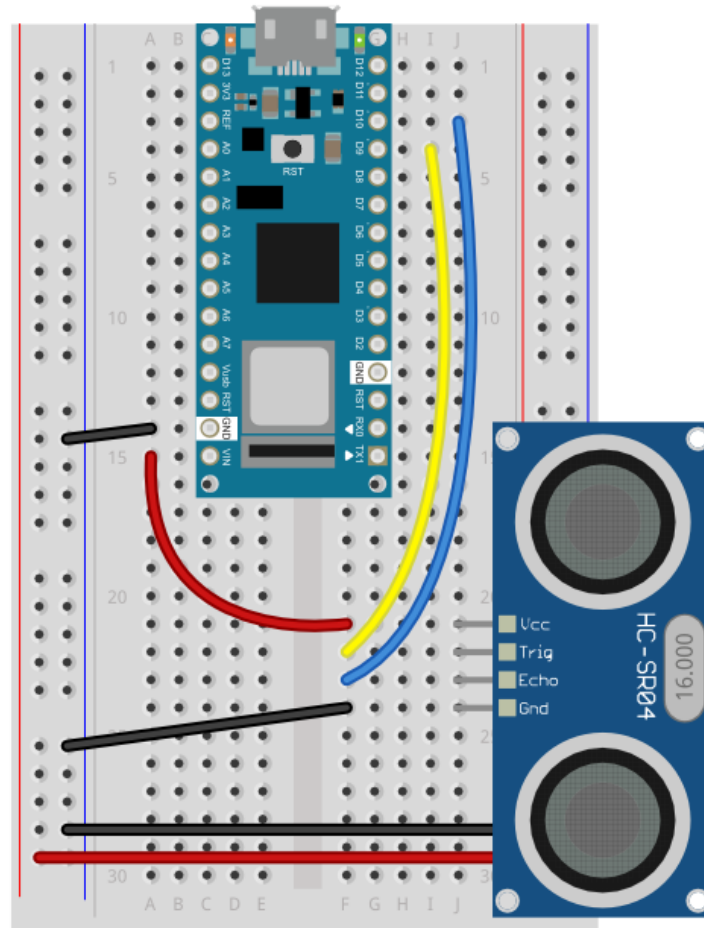
Arduino programming environment

Known issue, action required

The same known issue applies to SR04 which operates at 5 V source while the Arduino Nano 33 IoT supplies 3.3 V. It requires 2 VUSB (or VBUS) pads to be shorted to enable the pin.

Steps:

Step	Action
1	Connect your HC-SR04 sensor to the Arduino board. Note that the pin layout in the image below may look different than the board you may have.



2	Connect your Arduino board to your computer using the USB cable.
3	Write an Arduino sketch (or download it from https://github.com/deakin-deep-dreamer/sit225/blob/main/week_2/sketch_hcsr04_distance.ino) which looks like below. Compile the code in Arduino IDE, deploy to the board and observe output in the Arduino IDE serial monitor.

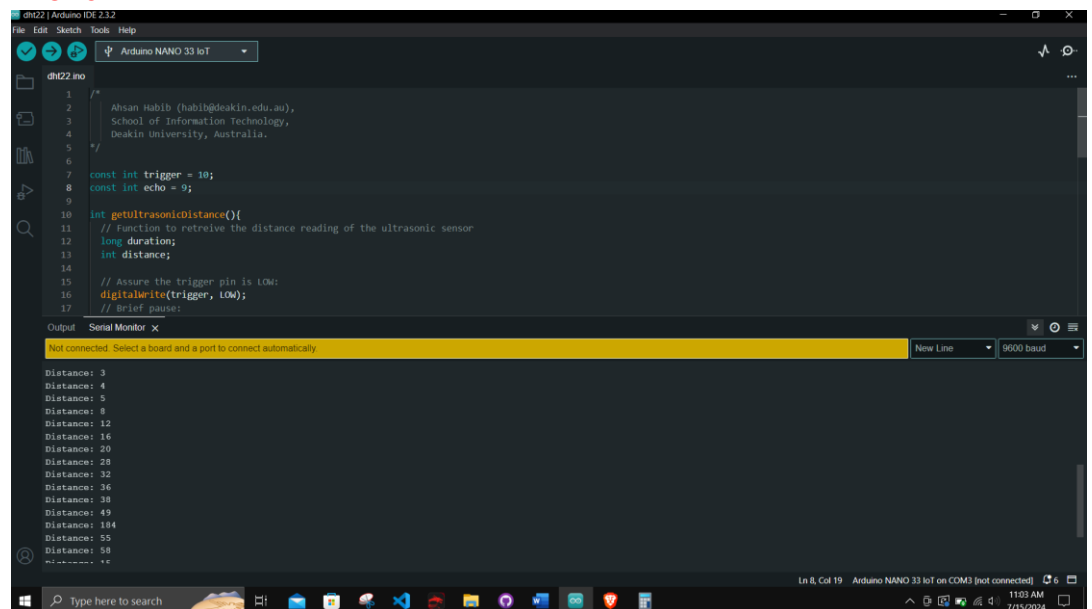
	<pre> 7 const int trigger = 2; 8 const int echo = 3; 9 10 int getUltrasonicDistance(){ 11 // Function to retrieve the distance reading of the ultrasonic 12 long duration; 13 int distance; 14 15 // Assure the trigger pin is LOW: 16 digitalWrite(trigger, LOW); 17 // Brief pause: 18 delayMicroseconds(5); 19 20 // Trigger the sensor by setting the trigger to HIGH: 21 digitalWrite(trigger, HIGH); 22 // Wait a moment before turning off the trigger: 23 delayMicroseconds(10); 24 // Turn off the trigger: 25 digitalWrite(trigger, LOW); 26 27 // Read the echo pin: 28 duration = pulseIn(echo, HIGH); 29 // Calculate the distance in centimeter (CM): 30 distance = duration * 0.034 / 2; 31 32 // Uncomment this line to return value in IN instead of CM: 33 //distance = distance * 0.3937008 34 35 // Return the distance read from the sensor: 36 return distance; 37 } 38 39 void setup() { 40 // Define inputs and outputs: 41 pinMode(trigger, OUTPUT); 42 pinMode(echo, INPUT); 43 44 // Start the serial monitor: 45 Serial.begin(9600); 46 } 47 48 void loop() { 49 // Print the distance to the serial monitor: 50 Serial.print("Distance: "); 51 Serial.println(getUltrasonicDistance()); 52 53 // Wait one second before continuing: 54 delay(1000); 55 } </pre>
4	<p>Question: Spec of SR04 is available here (https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf). Identify 2 critical aspects you should be careful about this sensor operation.</p> <p>Answer: The first aspect is to avoid connecting directly to a power supply and if the connection was made, the GND (ground) should be connected first, otherwise the module will malfunction. But since we are using an Arduino, not a raw power source like battery, then we don't need to worry about that. Also, we are establishing the ground and vcc connection before we plug to cable, whatever</p>

wires were connected first does not really mean anything, just need to keep in mind that the wires are properly connected to its corresponding pins.

The second aspect is the surface area and the smoothness of the object we are detecting. The area needs to be no less than 0.5 square meters, and the object should be as smooth as possible, or it will effect to results of measuring.

- 5 **Question:** Take a screenshot of your Serial Monitor displaying distance values while you try to generate a periodic motion by moving your hand gradually back and forth towards the sensor. Add the image here.

Answer:



The screenshot shows the Arduino IDE interface. The code in the editor is as follows:

```
1 /*
2  * Author: Ahsan Habib (ahabib@deakin.edu.au),
3  * School of Information Technology,
4  * Deakin University, Australia.
5  */
6
7 const int trigger = 10;
8 const int echo = 9;
9
10 int getUltrasonicDistance(){
11   // Function to retrieve the distance reading of the ultrasonic sensor
12   long duration;
13   int distance;
14
15   // Assume the trigger pin is LOW:
16   digitalWrite(trigger, LOW);
17   // Brief pause;
```

The Serial Monitor window shows the following output:

```
Distance: 3
Distance: 4
Distance: 5
Distance: 8
Distance: 12
Distance: 16
Distance: 20
Distance: 28
Distance: 32
Distance: 36
Distance: 38
Distance: 49
Distance: 184
Distance: 55
Distance: 58
Distance: 58
```

The status bar at the bottom indicates "Ln 8, Col 19 Arduino NANO 33 IoT on COM3 [not connected]" and the date "7/15/2024".

Activity 2.3: Working with sensor - Accelerometer

LSM6DS3 module on the Arduino Nano 33 IoT is an accelerometer and gyroscope sensor.

Hardware Required

Arduino Nano 33 IoT Board (has inbuilt LSM6DS3 module)

USB cable

Software Required

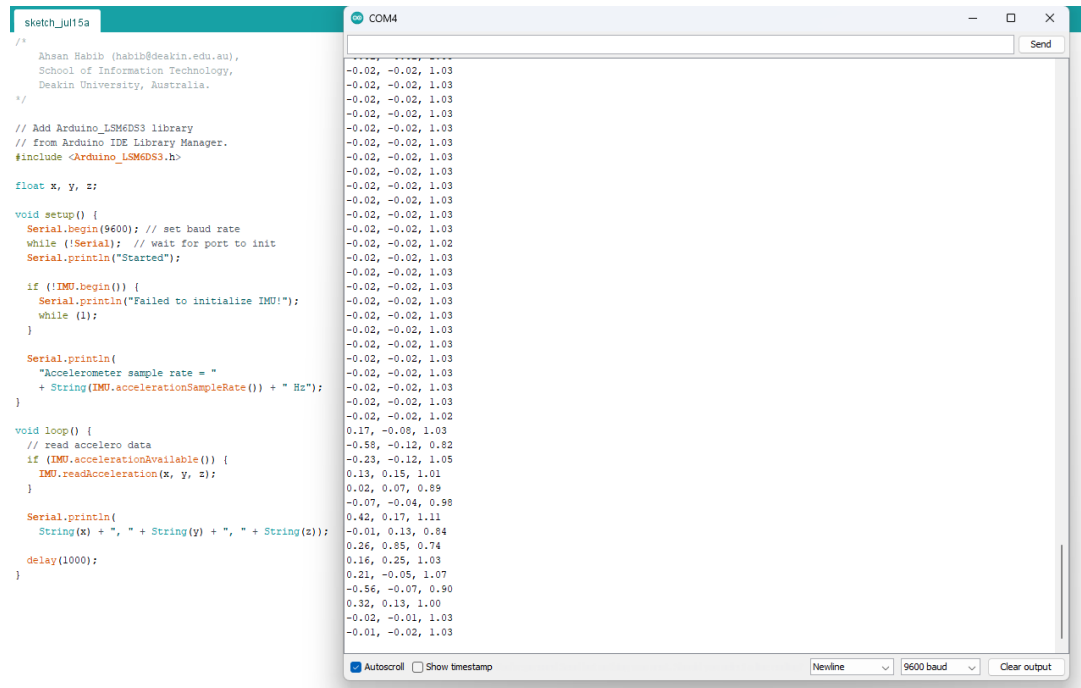
Arduino programming environment

Steps:

Step	Action
1	Write Arduino sketch (or download code from https://github.com/deakin-deep-dreamer/sit225/blob/main/week_2/sketch_accelero.ino) which looks like below. Compile the code in Arduino IDE, deploy to the board and observe output in the Arduino IDE serial monitor.

	<pre> 6 7 // Add Arduino_LSM6DS3 library 8 // from Arduino IDE Library Manager. 9 #include <Arduino_LSM6DS3.h> 10 11 float x, y, z; 12 13 void setup() { 14 Serial.begin(9600); // set baud rate 15 while (!Serial); // wait for port to init 16 Serial.println("Started"); 17 18 if (!IMU.begin()) { 19 Serial.println("Failed to initialize IMU!"); 20 while (1); 21 } 22 23 Serial.println(24 "Accelerometer sample rate = " 25 + String(IMU.accelerationSampleRate()) + " Hz"); 26 } 27 28 void loop() { 29 // read accelero data 30 if (IMU.accelerationAvailable()) { 31 IMU.readAcceleration(x, y, z); 32 } 33 34 Serial.println(35 String(x) + ", " + String(y) + ", " + String(z)); 36 37 delay(1000); 38 } </pre>
2	<p>Question: Spec of LSM6DS3 is available here (https://content.arduino.cc/assets/st_imu_lsm6ds3_datasheet.pdf). Identify at least 3 attributes of this sensor you think important to work with.</p> <p>Answer: The first attribute I found important is the make up of LSM3DS3 where it combines both accelerometer (linear motion) and gyroscope (angular motion, the turning velocity) to detection motion. The second attribute is flexible power consumption. It can perform in low or high-power consumption depending on the requirements of the task. The last attribute is the built-in embedded features. Things like event detection (free-fall, wakeup, tap), pedometer functions (step detector and counter), tilt detection, motion, sensors are all integrated directly into the hardware.</p>
3	<p>Question: Take a screenshot of your Serial Monitor displaying sensor readings. Add the image here.</p>

Answer:



```
/*
  Ahsan Habib (habib@deakin.edu.au),
  School of Information Technology,
  Deakin University, Australia.
*/

// Add Arduino_LSM6DS3 library
// from Arduino IDE Library Manager.
#include <Arduino_LSM6DS3.h>

float x, y, z;

void setup() {
  Serial.begin(9600); // set baud rate
  while (!Serial); // wait for port to init
  Serial.println("Started");

  if (!IMU.begin()) {
    Serial.println("Failed to initialize IMU!");
    while (1);
  }

  Serial.println(
    "Accelerometer sample rate = "
    + String(IMU.accelerationSampleRate()) + " Hz");
}

void loop() {
  // read accelero data
  if (IMU.accelerationAvailable()) {
    IMU.readAcceleration(x, y, z);
  }

  Serial.println(
    String(x) + ", " + String(y) + ", " + String(z));

  delay(1000);
}
```

COM4

Send

Autoscroll Show timestamp

Newline 9600 baud Clear output

4

Question: Identify the max sampling rate and consider reducing the delay (line 37 in the sketch) to increase the number of samples. Summarise your findings here.

Answer: The accelerometer data can be sampled at around 6.7 kilosamples per second. In the code, we can change the delay to 1 millisecond to increase the number of samples. The data are updated very fast.

<https://learn.sparkfun.com/tutorials/lsm6ds3-breakout-hookup-guide/all>

Activity 2.4: Plot data using Python Notebook

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. You can find detail in official website (<https://matplotlib.org>).

Hardware Required

No hardware required

Software Required

Python Jupyter notebook

Steps:

Step	Action
1	Download the Jupyter notebook week2_notebook.ipynb from here (https://github.com/deakin-deep-dreamer/sit225/blob/main/week_2/week2_notebook.ipynb). Follow the instructions in the notebook to carry out instructions and finally convert the notebook to PDF so you can combine it with this activity sheet PDF.