Student name: Yasmin Pokia

Student ID: s222245206

SIT225: Data Capture Technologies

# Activity 2.1: Working with sensor - DHT22

DHT22 is a temperature and humidity sensor.

## Hardware Required

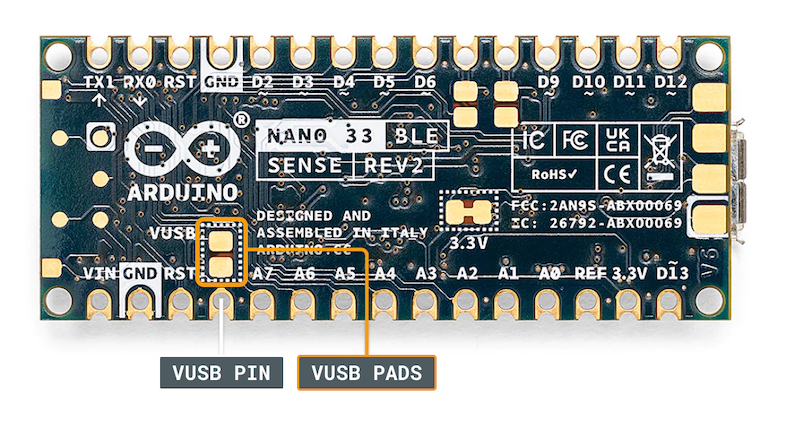
* Arduino Board
* DHT22 sensor
* USB cable

## 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 | 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.       1. 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. 2. 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. 3. 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. 4. 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. |
| 4 | Question: A spec of the DHT22 sensor is given in the link below. It mentions that the sampling rate is 0.5 Hz.  <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial>  i) What does the sampling rate mean?  ii) Where is this used in the Arduino code?  Answer: i) The frequency at which a sensor may acquire a new measurement is known as its sampling rate. Given that 0.5 Hz is half of 1 Hz, which is one reading per second, the DHT22 sensor may obtain a new measurement every two seconds at a sampling rate of 0.5 Hz. The DHT22 sensor has a 2-second time frame to update its humidity and temperature measurements. This determines the highest frequency at which you can get fresh data from the sensor.  ii) By making sure that the code doesn't try to read new data from the sensor more frequently than the sensor's designated sampling rate, the Arduino code follows the sampling rate. Usually, the loop() function is used to do this by adding a delay between each subsequent reading. |
| 5 | Question: Take a screenshot of your Serial Monitor displaying temperature & humidity sensor data logs. Add the image here.  Answer: |

# 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. |
| 4 | 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.  Answer: One critical aspect you should be careful about this sensor operation is the power supply voltage. The HC-SR04 operates on a power supply of 5V. Lower than 5V voltage applied to the sensor may cause it to malfunction or give false signals. For the sensor to work correctly, a steady 5V power source is needed. Make that the HC-SR04's VCC pin is connected to either the Arduino's 5V pin or the proper power supply.  Another critical aspect you should be careful about this sensor operation is minimum and maximum distance. The HC-SR04 sensor has a minimum measurable distance of approximately 2cm and a maximum measurable distance of up to 400cm. Accurate measurement of distances longer than 400 cm or less than 2 cm is not possible with this sensor. Make sure the object you are aiming for is inside this range. When an object is too close (less than 2 cm) or too far away (more than 400 cm), the sensor may give false or zero distance readings. Additionally, keep the sensor away from areas with obstacles in its path or where it could pick up false echoes. |
| 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: |

# 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. |
| 2 | 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.  Answer: One attribute of this sensor that I think is important to work with is the **accelerometer range**. The LSM6DS3 accelerometer can be configured with ranges of +/- 2g, +/- 4g, +/- 8g, and +/ 16g. The maximum acceleration that the accelerometer is capable of measuring is determined by its range. To accurately capture the intended level of acceleration in your application, you must choose a suitable range. For small motions, for instance, a range of ±2 g is appropriate, but for high-impact applications, ±16 g is required.  Another attribute is **sampling rate**, where the LSM6DS3 supports various sampling rates for its accelerometer, with a maximum rate of 6.66 kHz. The frequency at which the sensor changes its readings is dependent on the sampling rate. For applications needing high precision or fast reaction times, a higher sample rate makes updates possible more quickly and in-depth. To accurately catch changes in quick movements or vibrations, for example, a greater sample rate will be required.  The last attribute is **power consumption**, where the LSM6DS3 features low power consumption with different operating modes, including a low-power mode. In battery-operated or energy-sensitive applications, power consumption is crucial. Switching to low-power modes is beneficial for portable devices or systems where power efficiency is critical since it helps prolong battery life. Effective power mode management can have a big impact on the device's lifespan and overall performance. |
| 3 | Question: Take a screenshot of your Serial Monitor displaying sensor readings. Add the image here.  Answer: |
| 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: Summary of Findings  The LSM6DS3 sensor supports a maximum sampling rate of 6.66 kHz or 6660 Hz for the accelerometer. The current delay in the sketch is set to 1000 ms which is 1 second, which limits the rate at which new samples are taken and printed to the serial monitor to 1 sample per second. Given the maximum sampling rate of 6.66 kHz, the sensor can provide up to 6660 samples per second. By reducing the delay, the number of samples increase through changing the delay to 100 ms or 0.1 second, would output approximately 10 samples per second, and then further reducing it to 10 ms, this would give around 100 samples per second. You can obtain more data points per second and make greater use of the maximum sampling rate of the sensor by adjusting these settings. |

# 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. |