SIT111: Task 3.4P - Arduino Control using Sensors

Learning Objective

To build and understand a basic Arduino circuit that uses a sensor to collect data readings.

Summary - TL; DR

1. Read through the materials on the unit site.
2. Build and test the Arduino circuit, run experiments.
3. Submit:
    - Summary and reflection
    - Outcome from activities:
        - Photos, codes, videos of the constructed circuit or experiments
        - Describe any additional insights or knowledge learned during the active learning activities

---

Your Task

You have the choice to create **one of two** systems. You can either build:

- A soil moisture sensing system
- A temperature and humidity sensing system

Tips:

Different sensors may have pins that are mapped differently. You may need to tweak the codes to get your desired output.

E.g. the DF Robot soil moisture sensor pins are (GND,VCC,SIG) while Sparkfun soil moisture sensor has (VCC,GND,SIG). You may need to do some online research for your own sensor model.

Option 1: Soil Moisture Sensing System

Program an Arduino to interface with the Soil Moisture Sensor, capturing and displaying soil moisture levels as a percentage on a computer's Serial Monitor. This exercise focuses on analog signal processing and data representation in a practical environmental monitoring context.

Materials Required

- Arduino Uno (or similar Arduino board)

- Soil Moisture Sensor
- Breadboard
- Jumper wires
- USB cable to connect the Arduino to a computer
- Arduino IDE installed on the computer

Circuit Assembly:

- Connect the Soil Moisture Sensor's VCC pin to the Arduino's 5V output.
- Connect the GND and OUTPUT pins.
- You may use the breadboard and jumper wires to facilitate these connections.

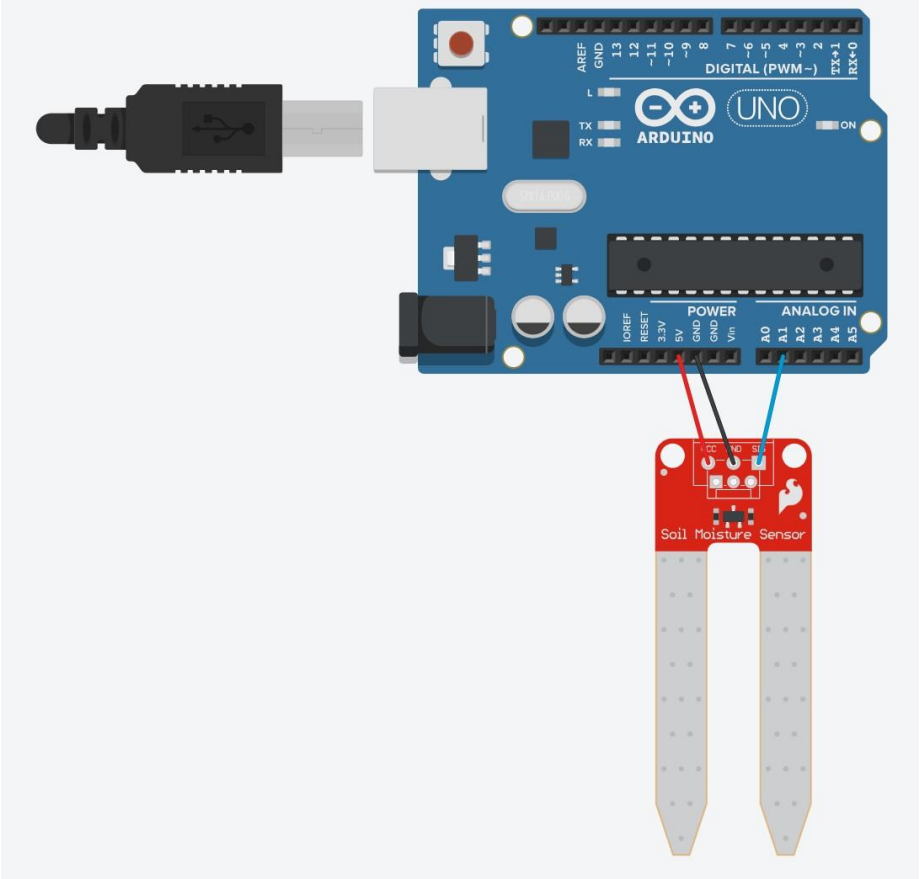

Figure 1: Sample Circuit - Soil Moisture

Programming:

- Write an Arduino sketch to read the analog value from the soil moisture sensor.
- Convert the analog reading to a more interpretable form, like a percentage.

- Output the moisture level to the Serial Monitor for real-time monitoring.

```
int sensorPin = A1; // Analog input pin connected to the soil moisture sensor

void setup() {
  Serial.begin(9600); // Initialize serial communication
}

void loop() {
  int sensorValue = analogRead(sensorPin); // Read the value from the sensor int moistureLevel =
  map(sensorValue, 0, 1023, 100, 0); // Map it to a 0-100% range

  Serial.print("Moisture Level: ");
  Serial.print(moistureLevel); Serial.println("%");

  delay(2000); // Delay between readings
}
```

Option 2: Temperature and Humidity Sensing system

Program an Arduino to interface with the DHT22 Temperature and Humidity Sensor that accurately measures and displays temperature and humidity readings. This exercise focuses on digital signal processing and environmental data monitoring.

Materials Needed:

- Arduino Uno (or similar Arduino board)
- DHT22 Temperature and Humidity Sensor
- Breadboard
- Jumper wires
- USB cable to connect the Arduino to a computer
- Arduino IDE installed on the computer
- DHT sensor library installed in the Arduino IDE

Circuit Assembly:

- Connect the DHT22 sensor's VCC pin to the Arduino's 5V output.
- Connect the GND pin to one of the Arduino's GND pins.
- Connect the DATA pin to a digital pin on the Arduino (e.g., pin 2).

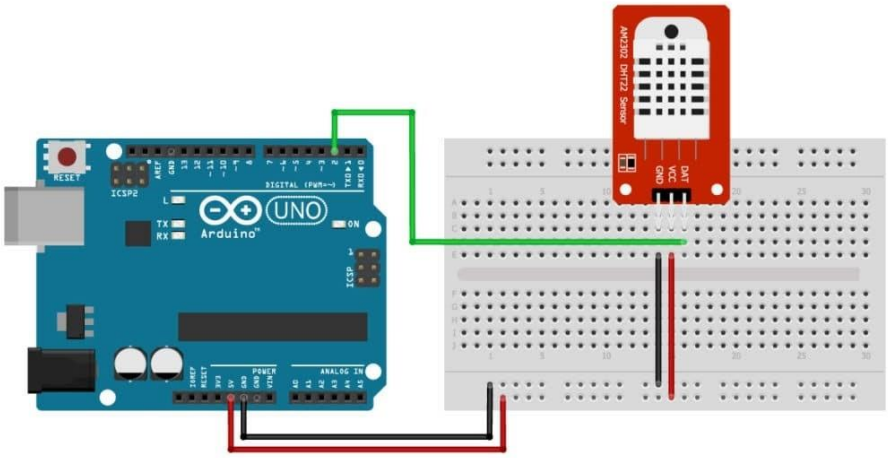

Figure 2: Sample Circuit - Temp and Humidity

Programming:

- Include the DHT sensor library in your Arduino sketch.
- Initialise the DHT22 sensor, specifying the digital pin connected to the DATA pin.
- In the loop(), retrieve the temperature and humidity from the DHT22 sensor.
- Output the temperature and humidity readings to the Serial Monitor.

```
#include "DHT.h"

#define DHTPIN 2          // Digital pin connected to the DHT sensor #define

DHTTYPE DHT22 // DHT 22 (AM2302) DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600); dht.begin();
}

void loop() {
  float humidity = dht.readHumidity(); float
  temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
```

```
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("%, Temp: ");
  Serial.print(temperature); Serial.println(" Celsius");

  delay(2000);
}
```

Uploading and Testing:

- Connect the Arduino to the computer using the USB cable.
- Use the Arduino IDE to upload your sketch to the board.
- Open the Serial Monitor to view the real-time temperature and humidity readings **OR**
- Test the sensor by placing it in different soil conditions (dry, moist, wet) and observe the readings on the Serial Monitor.

Prepare Your Submission

Once you feel confident that you have achieved the learning goals, you can prepare a submission to demonstrate this. This will contain three sections: summary of what you learnt, reflection on your learning, and evidence of study and practice.

Section 1: Summary

Summarise what you have done and what you have learnt from the experiment. This should be a personal summary, written so that it will be useful to you should you need to quickly revise these concepts and tools in the future. Capture the most important aspects from the materials in the unit site and anything else you find related to this topic.

Section 2: Reflection

Reflect on your learning by responding to the following prompts:

- How do you know you have achieved the learning goals?
- What is the most important thing you learned from this and why?
- How does the content or skills learned here relate to things you already know?
- Where or when do you think it will be useful?

Note:: The content for the first two sections should not exceed 500 words or 1 printed page.

Section 3: Evidence of study and practice

This section will contain evidence of your outputs from the learning activities for this task:

- Screenshot of the Arduino IDE successsfully uploading the code
- Your code
- Collected readings from the sensor(s)
- A short video of the working hardware (YouTube or Panopto)

Upload Your Submission

Once you have all the evidence in place, login to CICRA VLE and mark the task as **Ready for Feedback**. The submission process will ask you to upload evidence of completion of the task. For quizzes, please include a screenshot showing your quiz score. For Active Learning Session problems, you must submit evidence that you yourself had completed the activities. While working in groups/pairs is welcome, you must have evidence of your own contributions.

The system will also ask you to reflect on what unit learning outcomes have been achieved by this task.

Engage with Feedback

To get the task marked as **Complete**, you need to engage with the feedback you receive. Your tutor will review your submission and may ask you to clarify aspects of your learning, redo parts of the task, or include aspects you have missed. You may be asked to discuss the task in class or online. Use these discussions as an opportunity to help develop and validate your understanding.

If you are asked to resubmit, *make sure your subsequent submission includes a comment that describes how you have addressed the feedback you received*. This needs to demonstrate how you have addressed all the aspects indicated by your tutor in their feedback on your learning.