AGTE 411: IoT in Agriculture Lab

- Paul Nugent (Precision Agriculture)

License: This assignment is licensed under CC BY-NC 4.0

Purpose:

This lab introduces students to basic circuit measurements, sensor calibration, and control logic implementation using Arduino. The goal is to develop skills in connecting and programming environmental sensors to collect calibrated data and apply control logic for hardware interactions.

Students will gain practical experience in integrating sensors and actuators, writing Arduino code, and building a simple micrcontroller system for temperature and humidity control.

Duration: 2 Hours

Learning Objectives

By the end of this assignment, students will be able to:

1. Demonstrate the operation and calibration of electronic sensors for temperature and humidity.
2. Apply their knowledge of Arduino programming to produce and display calibrated sensor outputs.
3. Implement logic to control hardware components (LED and fan) based on sensor input.

Materials Needed

- Rigol DC Power Supply
- HMP45 Temperature and Humidity Sensor
- Arduino MKR WiFi 1010
- Groov LED and relay attachments
- 12V computer case fan
- Assorted wires and connectors

Assignment Structure

Assignment Title: Lab 4: Calibrated Temperature and Humidity Sensors

This assignment is part of a course on the Internet of Things (IoT) in agriculture. It serves to reinform knowledge on electronic sensors and the Arduino microcomputer and introduce relays and sensor calibration. The lab is broken into two parts.

In the first part, students are given an HMP45 temperature and humidity sensor and an Arduino. They must use the Arduino to collect the output voltages and convert them to calibrated temperature and humidity measurements.

In the second half of the lab, the students need to implement logic in their board to turn on and off an LED and a 12V fan connected to the Arduino via a relay.

Student 1 _____

Student 2 _____ Date _____

Lab 4: Calibrated Temperature and Humidity Sensors

HMP45 Temperature and Humidity Measurement

The HMP45 is a scientific-grade temperature and humidity sensor often used in climate weather stations. Although these sensors have been replaced in the field and may not provide fully accurate measurements, they remain reliable for laboratory exercises. In this lab, you will connect the HMP45 sensor to an Arduino, calibrate its output, display the temperature and humidity readings, and implement control logic to activate a fan using a relay.

Wiring Tables

In this lab we will be using wiring tables rather than drawn wiring diagrams to describe the connections of your boards. Wiring tables document and organize the connections between components in an electronic circuit. They provide a structured reference of how each pin on a sensor, microcontroller, or power supply is connected. By first outlining connections in a table format, we can ensure accuracy during setup, identify potential wiring problems, and easily modify the configuration before the circuit is built. An example wiring table is below.

Wiring Table Example:

HMP45 Pins	Connection
Temperature	Analog Pin A0 (Arduino)
Humidity	Analog Pin A1 (Arduino)
12V Power	12V Power Supply

Part 1 Wiring the HMP45 to the Arduino

Refer to the diagrams in Figure 1 which show the MKR WiFi 1010 and carrier board pinouts and Figure 2 which shows the HMP45 sensor wiring. Use these diagrams, along with your knowledge of power supply connections and grounding, to connect the power supply, Arduino (in the carrier board), and the HMP45 sensor. Document your connections in the table on page 3, noting that some pins may require multiple connections. Ensure all connections are accurate and secure before powering the circuit.

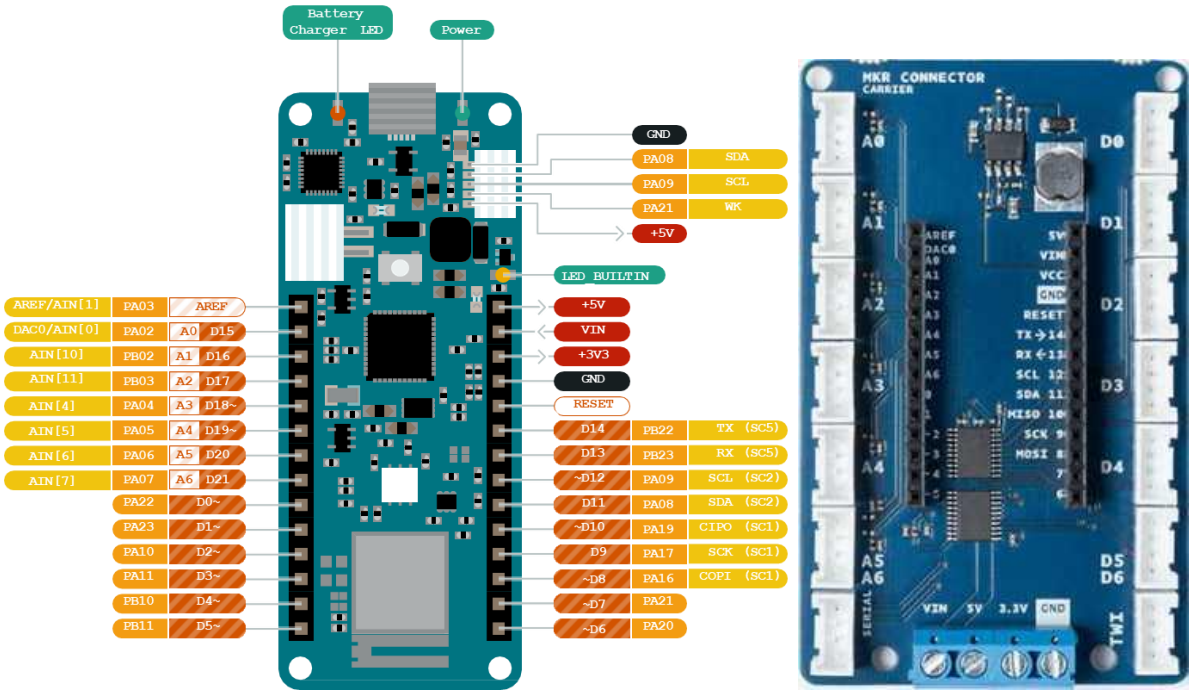

Figure 1. Arduino MKR WiFi 1010 Pinout (left), MKR Carrier board (right). Figure adapted from Arduino Nano 33 IoT Full Pinout by Arduino, n.d. (ABX00023-full-pinout.pdf), and (MKRConnectorCarrier.jpg) licensed under CC BY-SA 4.0. (https://creativecommons.org/licenses/by-sa/4.0/)

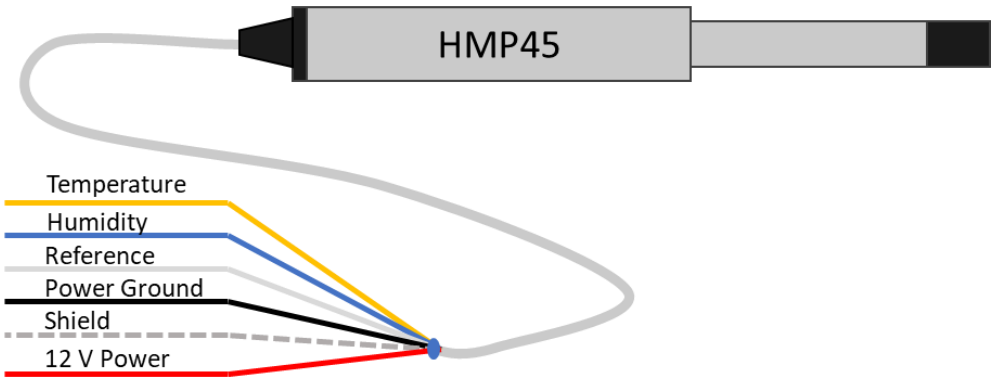

Figure 2. Campbell Scientific HMP45, showing the sensor and cable wires.

HMP45 Pins	Connection
Temperature	
Humidity	
Reference	
Shield	
12V Power	
Power Ground	

Arduino Pin	Connection
Carrier Vin (12V)	
Carrier GND	
A0	
A1	

Power Supply Pin	Connection
12V+	
Negative (12V)	
Earth GND	

Instructor Checkpoint: Have your instructor review your wiring tables and sign off before proceeding _____.

After approved by your instructor, build the circuit described in the wiring tables.

Instructor Checkpoint: Have your instructor review your circuit and sign off before proceeding _____.

Once all connections are approved, carefully power your circuit using the DC power supply.

Implement and upload Arduino code to read temperature and humidity data from the HMP45 sensor. Display the temperature (in Celsius) and humidity (%) on the serial monitor.

Instructor Checkpoint: Accurate Temperature and humidity display _____.

Part 2: Implementing Control Logic with LED and Relay-Controlled Fan

Step 1: Install and demonstrate the LED

1. **Power Down Your Board**
2. Install the LED into digital port 0 (D0) on the carrier board.
3. Power on your board
4. Show your instructor that the LED is functioning according to your specified logic.
    a. The LED should turn on and off as expected with no flicker when transitioning states.

Instructor Checkpoint: LED Operation _____.

Step 2: Connect the Relay and Fan

1. **Power Down Your Board**
2. Relay Connections:
    a. Connect the relay control input to digital port D1 on the carrier board.
    b. Connect the relay's 12V input to the power supply.
    c. Connect the relay's output terminals to the 12V fan, ensuring the polarity matches the fan's specifications.
3. Power on your board
4. Show your instructor that the fan is turning on according to your specified logic.
    a. The fan and LED should turn on and off as expected with no flicker.

Instructor Checkpoint: Fan Operation _____.

Analysis

To be completed digitally and submitted on D2L.

1. How did this lab improve your understanding of integrating hardware and software in an IoT system? Provide specific examples from the lab.

2. What was the most challenging part of this lab, and how did you overcome it?

3. Imagine you want to expand this system to measure additional environmental parameters, such as soil moisture or light intensity. What additional components would you need, and how would you modify your code to accommodate them?

4. How would you implement error handling in your Arduino code to manage potential sensor failure or incorrect readings?

Why is error handling critical in real-world IoT systems?