Lab 1

Essentials: Grammar, Semantics, and Interaction

Applied Python Programming with AI and Raspberry Pi Interfaces

Instructor: Dr. Vikas Thammanna Gowda Semester: ABCD 20YX
Points: 100 Assign: *TBD* Due: *TBD*

Name: ____

Overview

In this lab, you will connect a DHT22 digital temperature sensor and a 20 × 4 LCD display to a Raspberry Pi to create a simple digital thermometer.

The lab is split into two parts:

- **Part 1:** You will be provided with the *complete code, circuit diagram, setup, and wiring instructions*, along with a *brief in-class demonstration.* In this part, you will use the DHT22 sensor to measure the ambient temperature in degrees Celsius, and humidity as a percentage of relative humidity and display the readings on the LCD.
- **Part 2:** A collaborative activity to extend the code to convert and display the temperature in Fahrenheit and Kelvin as well.

Grading Rubric:

Criteria	**Points**	**Grade**
Part 1 circuit set up	10	
Part 1 Output: correctness and documentation clarity	15	
Part 2 circtit set up	15	
Part 2 Code: use of descriptive identifiers	10	
Part 2 Code: meaningful comments	10	
Part 2 Output: correctness and documentation clarity	20	
Reflection and Analysis	20	
Total	**100**	

Comments:

General Instructions

- Turn off power before you touch any wires or parts. Always shut down the Raspberry Pi from the menu before unplugging it.
- The Pi's pins are 3.3 V only-never put 5 V on a GPIO pin. Use a resistor with every LED (about 220-330 Ω).
- Don't plug motors, relays, or other high-power parts straight into the Pi. Use a driver board or transistor, and give those parts their own power supply with a shared ground.
- Keep the area safe: no drinks near the electronics, handle boards by the edges, keep wires neat and short, and wear eye protection when cutting leads.
- Check before you power on: compare your circuit to the diagram, use a multimeter to spot mistakes, then do short test runs and watch for anything getting hot.
- If you smell burning, see smoke, or a part feels too hot, turn off power immediately and get the teacher. We only use low-voltage DC—never plug into wall outlets unless told and supervised.

Learning Objectives

- Understand how to read sensor data (temperature) from the DHT22 using Raspberry Pi.
- Use variables to store sensor values and apply arithmetic operators for unit conversion (Celsius to Fahrenheit and Kelvin).
- Distinguish and convert between data types (e.g., float to string).
- Format and display multi-line text on a 20 × 4 LCD.
- Gain experience integrating hardware (sensor + LCD) with Python code.

Required Components:

- Raspberry Pi (any model with 40 GPIO pins) with Raspbian/Raspberry Pi OS installed.
- Breadboard and jumper wires.
- DHT22 temperature/humidity sensor with a 10 kΩ resistor for the required pull-up between its data pin and 3.3V
- 20x4 Character LCD.

Part 1: Reading Celsius Temperature and Humidity

In the first part of the lab, we will run a Python program on the Raspberry Pi to read the temperature from the DHT22 and display it on the LCD in Celsius. This will introduce how the program uses variables and their data types to store the sensor data, and how it outputs it to the display.

Illustration

Follow these steps to assemble the LED circuit. **Make sure your Raspberry Pi is shut down or powered off while wiring the circuit to avoid any accidental short circuits or damage.**

DHT22 Sensor Wiring:

The DHT22 has three pins: VCC (power), Data out, and GND. We will use a 10 kΩ resistor as a pull-up on the data line (connecting data to VCC) to ensure reliable communication. Wire the DHT22 to the Raspberry Pi as follows:

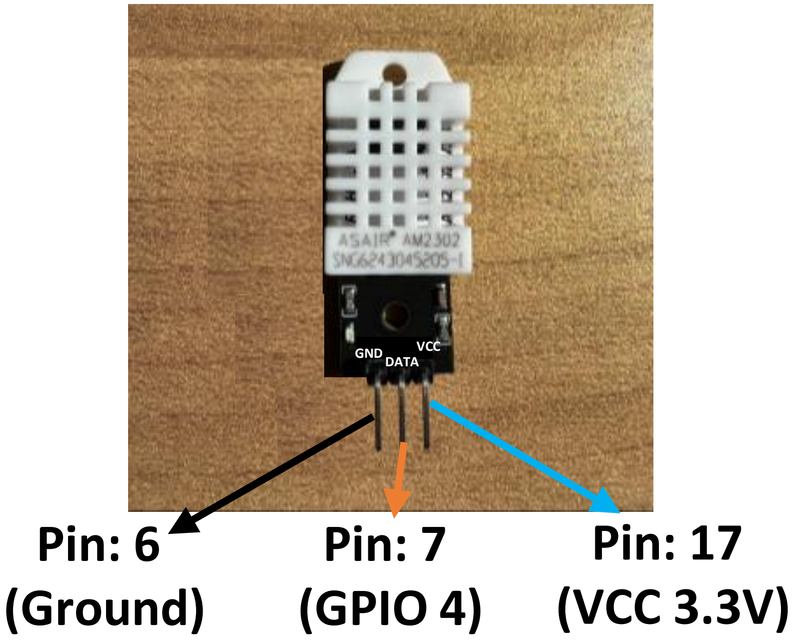

Figure 1: DHT22 Sensor connection to Raspberry Pi's GPIO pins

1. Place a 10 kΩ resistor between the DHT22's VCC pin (pin 1) and Data pin (pin 2) on the breadboard. This resistor acts as a pull-up, tying the data line to 3.3V when idle. (If you had a DHT22 module on a breakout board, this resistor would typically be built-in.)
2. Connect DHT22 pin 3 (VCC) to the Raspberry Pi's 3.3V power pin (e.g. physical pin 17 on the Pi)
3. Connect DHT22 pin 2 (Data) to a Raspberry Pi GPIO input pin. In our example, we'll use GPIO4 (physical pin 7) which is a common default for DHT sensors.
4. Connect DHT22 pin 1 (GND) to a ground pin on the Raspberry Pi (physical pin 6 or any other GND)

Note: The DHT22 should be powered at 3.3V (not 5V) when used with the Pi, to keep the data signal at Pi-safe voltage levels. Also, the sensor cannot be read more often than once every 2 seconds (approx.), as it needs time to take a measurement.

LCD Display Wiring:

For the 20×4 character LCD, we assume it has an I²C interface adapter attached. This adapter greatly simplifies the wiring by using only four connections.

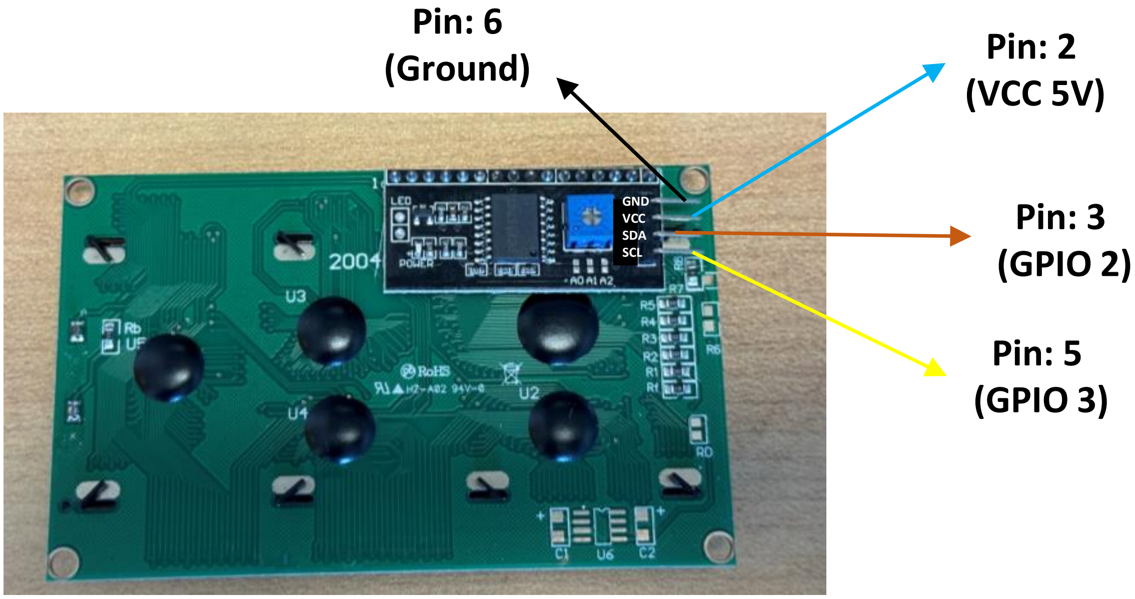

Figure 2: LCD connection to Raspberry Pi's GPIO pins.

1. Connect the LCD module's VCC pin to the Pi's 5V power (physical pin 2 or 4), and the LCD GND to a Pi GND pin 6.
2. Connect the LCD's SDA pin to the Pi's SDA line. On Raspberry Pi, SDA corresponds to GPIO 2 (physical pin 3)
3. Connect the LCD's SCL pin to the Pi's SCL line, which is GPIO 3 (physical pin 5).

Set up

The wiring setup should look similar to the following diagram. Ensure all connections are secure, double-check the pin numbers, and tripple-check the ground connection. Check-in with your instructor before powering on the Raspberry Pi.

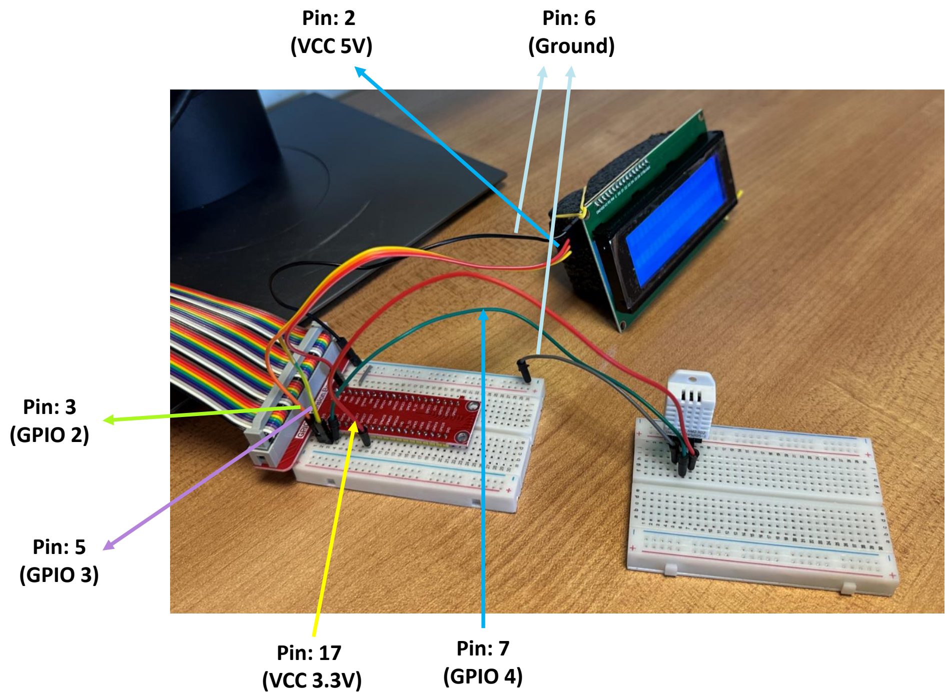

Figure 3: Wiring set-up.

Open terminal on your Raspberry Pi and type the following command:

- Load virtual environment:

```
source vir-env-name/bin/activate
```

- Open idle:

```
python -m ildelib
```

Run and Observe

It's time to test the circuit and code. Make sure your Run the privided Python script.

Record your observations:

No.	**Temperature**	**Humidity**
1		
2		
3		
4		
5		
6		
7		

Table 1: Temperature in °C and Humidity Reading

Part 2: Converting Temperature to Fahrenheit and Kelvin

After getting the Celsius display working, the next step (and main goal of this lab exercise) is to modify the code to also display the temperature in Fahrenheit and Kelvin. This will involve using arithmetic operators in our code to convert the values, and handling additional variables for the new values. This is where we practice using operators to compute new results from existing data.

Formulas for Conversion

The formulas to convert temperature scales are well known:

- **Celsius to Fahrenheit:**

$$F = \left(\frac{9}{5} \times C\right) + 32$$

- **Celsius to Kelvin:**

$$K = C + 273.15$$

So, for example, if the sensor reads 25.0°C, then:

$$^{\circ}\mathrm{F} = 25.0 \times \frac{9}{5} + 32 = 77.0^{\circ}\mathrm{F}$$

$$K = 25.0 + 273.15 = 298.15\ \mathrm{K}$$

Record your observations:

No.	**°C**	**°F**	**°K**
1	25.0	77.0	298.15

Table 2: Temperature reading

Reflection and Analysis

1. How did using variables help in organizing your code and making it easier to modify?
2. Was it clear when and why you needed to convert between data types (e.g., float to string for display)?
3. Did your displayed values match your expectations (e.g., were the conversions accurate)? How did you verify this?
4. How confident do you feel now about applying basic arithmetic operations in code?
5. What was the most challenging part of combining hardware readings with software display?
6. **Further extension:** Discuss how his project can be extended.