CE406: Internet of Things
Lab 01: Introduction to Arduino and Sensor Interfacing

Instructor

September 29, 2025

Objective

This lab is designed to familiarize you with the Arduino development environment and basic sensor integration. By the end of this session, you will be able to:

1. Install and configure the Arduino IDE.
2. Write, compile, and upload a basic program (sketch) to an Arduino board.
3. Interface a DHT11 temperature and humidity sensor with the Arduino.
4. Read and display sensor data on the Serial Monitor.
5. Implement data processing, conditional logic, and user interaction based on sensor readings.

Hardware & Software Requirements

- An Arduino Uno board (or a compatible clone)
- A DHT11 Temperature and Humidity Sensor (3-pin module is preferred)
- A Breadboard
- Jumper Wires (Male-to-Female)
- A computer with a USB port
- Arduino IDE software

---

1 Installing the Arduino IDE

The Arduino Integrated Development Environment (IDE) is the primary software you will use to write code, compile it, and upload it to your Arduino board.

1. **Download:** Open your web browser and navigate to the official Arduino software page: `https://www.arduino.cc/en/software`.

2. **Select the Correct Version:** Download the appropriate version for your operating system (Windows, macOS, or Linux).

3. **Install:** Follow the on-screen instructions to install the software. When prompted on Windows, ensure you allow the installation of any required drivers.

4. **Connect Your Arduino:** Connect the Arduino Uno board to your computer using the USB cable. The green power LED on the board should light up.

5. **Configure the IDE:**

   - Launch the Arduino IDE.
   - Go to `Tools > Board > Arduino AVR Boards` and select **"Arduino Uno"**.
   - Go to `Tools > Port` and select the port your Arduino is connected to (e.g., `COMx` on Windows or `/dev/tty.usbmodemXXXX` on macOS/Linux).

2 Interfacing with the DHT11 Sensor

2.1 Installing the DHT11 Sensor Library

1. In the Arduino IDE, navigate to `Sketch > Include Library > Manage Libraries...`.
2. The Library Manager window will open. In the search bar, type `DHT sensor library`.
3. Find the library by **Adafruit** and click the "Install" button.
4. The IDE may ask you to install dependencies, such as the **Adafruit Unified Sensor** library. Click "Install All".
5. Once installation is complete, close the Library Manager.

2.2 Wiring the DHT11 Sensor

Connect the DHT11 sensor to your Arduino Uno using jumper wires as shown in the diagram and table below.

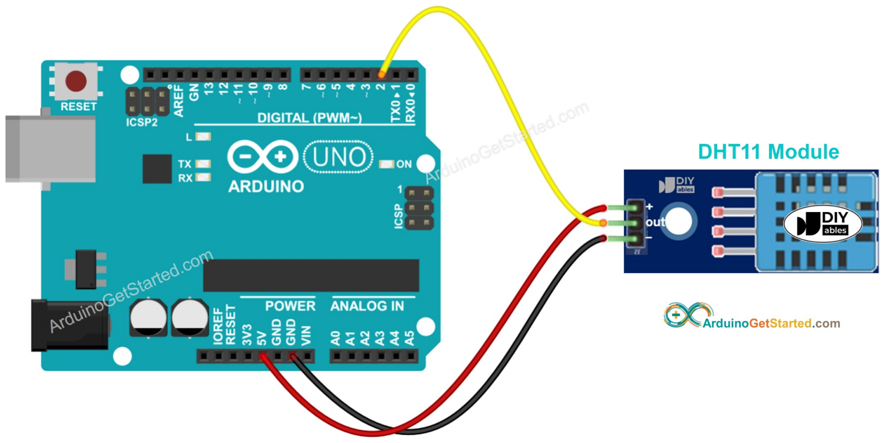

Figure 1: Wiring diagram for DHT11 and Arduino Uno.

DHT11 Pin	Connection	Arduino Uno Pin
VCC / +	Power	5V
DATA / Out	Signal	Digital Pin 2
GND / -	Ground	GND

3 Your First Sketch - Blink

The ”Hello, World!” of microcontrollers is making an LED blink.

1. Go to `File > Examples > 01.Basics > Blink`.
2. Click the **Verify** button (checkmark icon) to compile the code.
3. Click the **Upload** button (right arrow icon) to send the code to your Arduino.
4. Observe the small orange LED on your board. It should start blinking.

Blink Example Code:

```
// the setup function runs once
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off
  delay(1000);                       // wait for a second
}
```

4 Reading Data from the DHT11 Sensor

Now, let’s read data from the sensor and display it on the Serial Monitor.

1. Open a new sketch (`File > New`).
2. Copy and paste the code below into the IDE.
3. **Verify** and **Upload** the code to your Arduino.
4. Open the Serial Monitor (`Tools > Serial Monitor`).
5. Make sure the baud rate in the Serial Monitor is set to **9600**. You should see readings appear.

DHT11 Example Code:

```
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

```

```
void setup() {
  Serial.begin(9600);
  Serial.println("DHT11 Test!");
  dht.begin();
}

void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();

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

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C");
}
```

5 Lab Assignment

Modify the DHT11 example code from the previous section to implement the following features.

Task 1: Convert Celsius to Fahrenheit

Modify your code to calculate and display the temperature in Fahrenheit. The formula is: $F = (C \times \frac{9}{5}) + 32$. Your output should look similar to:

```
Humidity:  55.00 % Temperature:  25.50 *C / 77.90 *F
```

Task 2: Temperature Threshold Alarm

Set a temperature threshold variable (e.g., `float tempThreshold = 20.0;`). If the temperature crosses this threshold, print an alarm message. Example output:

```
Temp:  21.00 *C / 69.80 *F *** TEMPERATURE ALERT! ***
```

Task 3: Serial Commands to Change Threshold

Allow the user to change the `tempThreshold` value from the Serial Monitor. If the user types `T25` and hits Enter, the code should set the threshold to `25.0`. **Hint:** Look into the `Serial.available()` and `Serial.readStringUntil()` functions.

Task 4: Smooth Temperature with a Moving Average

Implement a moving average filter to smooth noisy sensor readings.

1. Define a constant for the number of samples, e.g., `const int NUM_SAMPLES = 10;`.
2. Create an array to store the last `NUM_SAMPLES` readings.

3. In each loop, get a new reading, replace the oldest value in the array, and calculate the average of all values in the array.
4. Print both the raw and the filtered (averaged) temperature. Example output:
   ```
   Raw Temp:  24.50 *C, Filtered Temp:  24.21 *C
   ```

Submission

Submit your final `.ino` Arduino sketch file containing the solutions for all four tasks. Ensure your code is well-commented to explain how you implemented each feature.