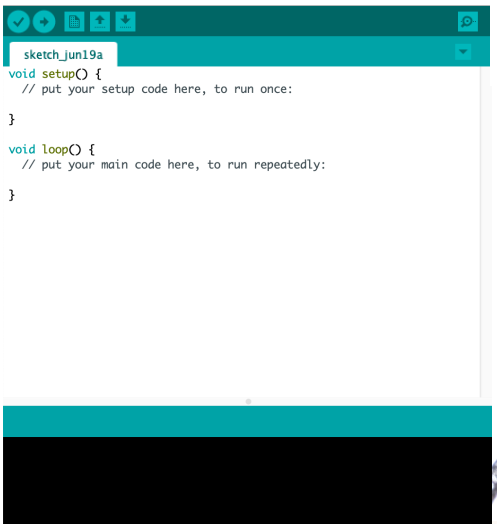
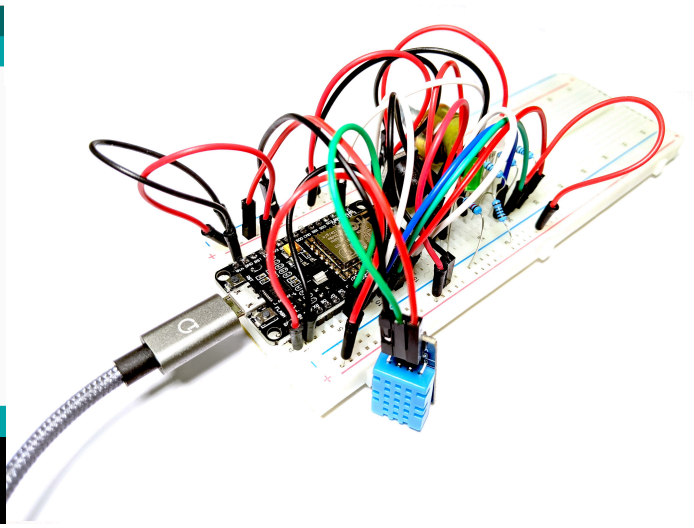
Smart Fridge

Part four: The temperature sensor

Project description:

In this project, you will build on the previous smart fridge exercise and add one more component to your smart fridge circuit. It is common for a fridge to display its internal temperature and humidity level. You will use a digital sensor to output both the temperature and the humidity values of your smart fridge. This is a continuation of the third part of the smart fridge exercise so please refer back to that part if you get lost.



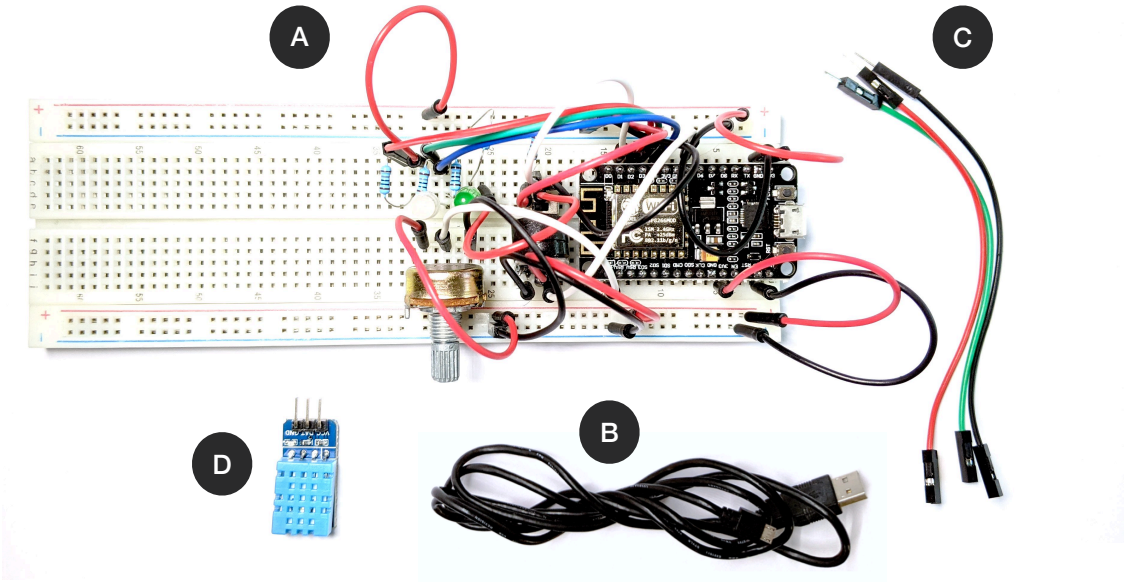


Project objectives:

- Get familiar with the temperature and humidity sensor
- Add the sensor to the smart fridge circuit
- Print both the recorded temperature and humidity values
- Add both values to the smart fridge web dashboard

Project components:

Component Reference	Component Quantity	Component Name	Component Description
A	1	Smart Fridge Part 3	The smart fridge circuit which features the potentiometer, the switch, the buzzer, and the RGB LED components
B	1	Micro USB Cable	A USB cable to power and upload instructions to a microcontroller
C	3	Jumper Wires	Conductive cables frequently used with a breadboard to connect two points in a circuit
D	1	DHT11 Temperature and Humidity Sensor	An electronic device that measures the temperature and humidity of an environment and converts the input data into electronic data



Step One
Introduction and Theory

In the second part of the smart fridge exercise, you encountered some interesting components such as the potentiometer, the buzzer, the RGB LED, and the electrical switch components. Furthermore, you created a web server to visualise and control some of the sensors. This exercise introduces you to one new component: the temperature and humidity sensor.

As a process to exploring the new component, you will build on the latest version of the smart fridge circuit. The idea here is to add the temperature and humidity sensor, read the recorded values, and print them to the Serial Monitor.

You will need both the circuit and the code for the smart fridge part three exercise to complete this project.

Let's briefly discuss the additional component for the circuit:

- **DHT11 Temperature and Humidity Sensor:** The DHT11 is a basic tiny digital module. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air. The values recorded are then sent out as a digital signal on the chosen data pin. The component has usually four leads but only three are necessary for reading the data: the ground (GND), the source (VCC), and the signal (SIG).

Step Two
Adding the component to the Smart Fridge Circuit

The circuit assembly for this exercise requires you to wire up 1 DHT11 Temperature and Humidity Sensor.

You will add the DHT11 component to your existing smart fridge circuit that you have built so far. Refer back to the third part of the smart fridge exercise to view the latest version of the circuit.

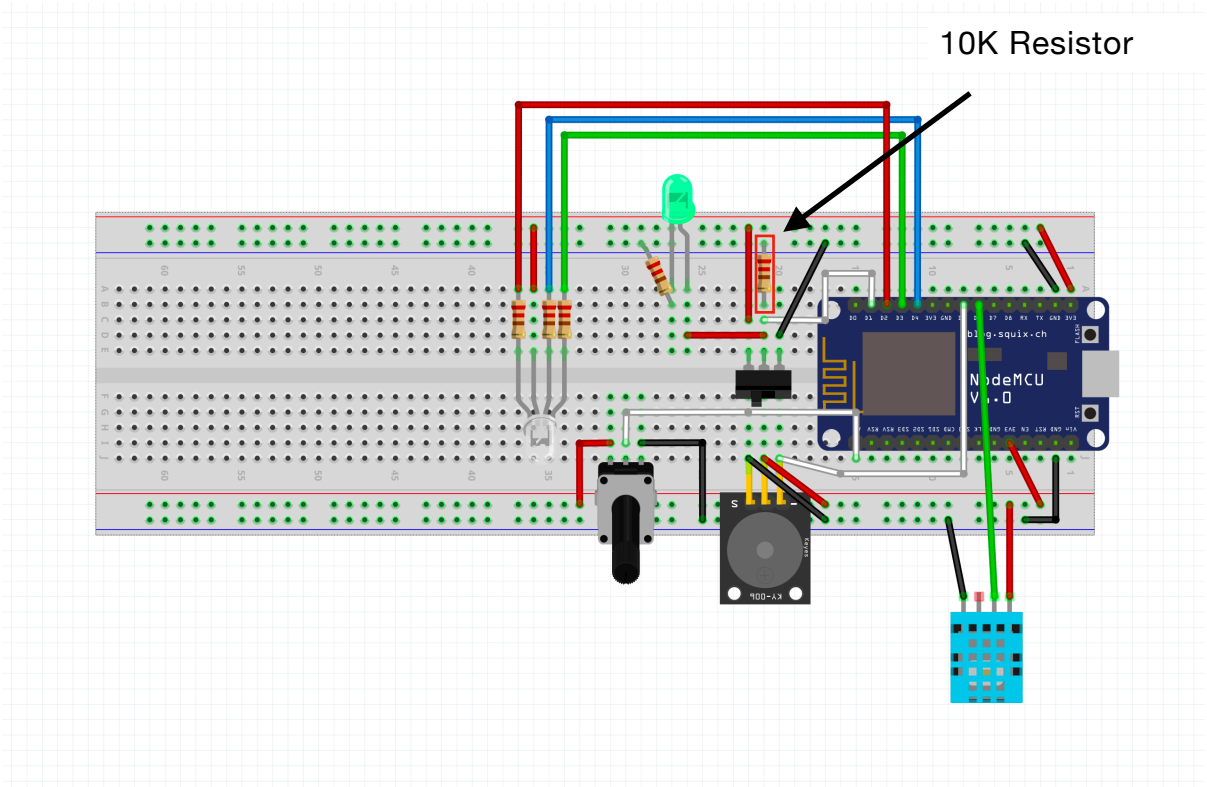
The DHT11 should be connected to digital pin D6 of your ESP board. This component has three leads. You should connect the VCC lead to the source (3.3V) and the GND lead to ground. The middle lead is the signal and it should be connected to digital pin D6.

Here a quick recap of the circuit wiring:

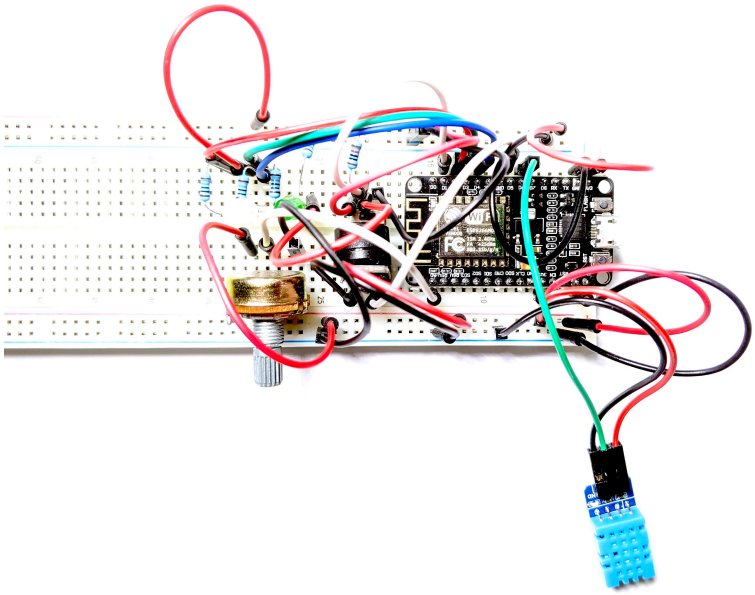
- **The smart fridge circuit :** You should have completed the first three parts of the smart fridge exercise as this component should be added to the latest version of that circuit.
- **DHT11:** The middle lead of the sensor (DAT) should be connected to digital pin D6. The GND lead goes to ground and the VCC lead goes to 3.3V.

It is now your turn to add the additional component to the smart fridge circuit. Use the same half of the longer breadboard to build this circuit. We will reserve the other half for upcoming projects. As the space is quite limited now, use female to male wires so that you can extend the component connection outside of the breadboard.

The diagram below shows you how the circuit should be assembled. It combines the circuit components for the third part of the smart fridge circuit with the additional temperature and humidity sensor component:



Furthermore, see below a picture of the circuit assembled:



Step Three
Writing the Code

Now that you have assembled the circuit, it is time to add the code for the temperature and humidity sensor component. Go ahead and create a new empty sketch from the Arduino IDE.

You should see an empty sketch like the following:

Feel free to save the sketch and rename it to something sensible: **smart_fridge_part4** for instance.

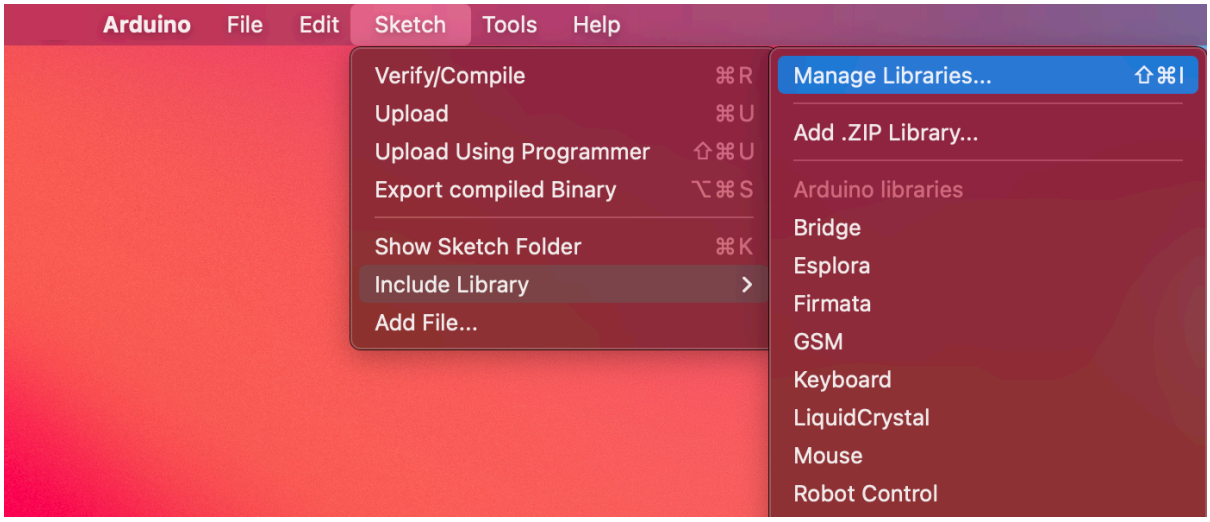
Now copy and paste the code that you wrote for the third part of the smart fridge exercise. You should have the code with the **fridgeTemperature(),** the **fridgeOn(),** the **trigBuzzer(),** the **temperatureStatus(),** and the **get_index()** utility functions. You should also have an additional function to control the buzzer from the web server.

There is one core functionality that we want to add to the program here:

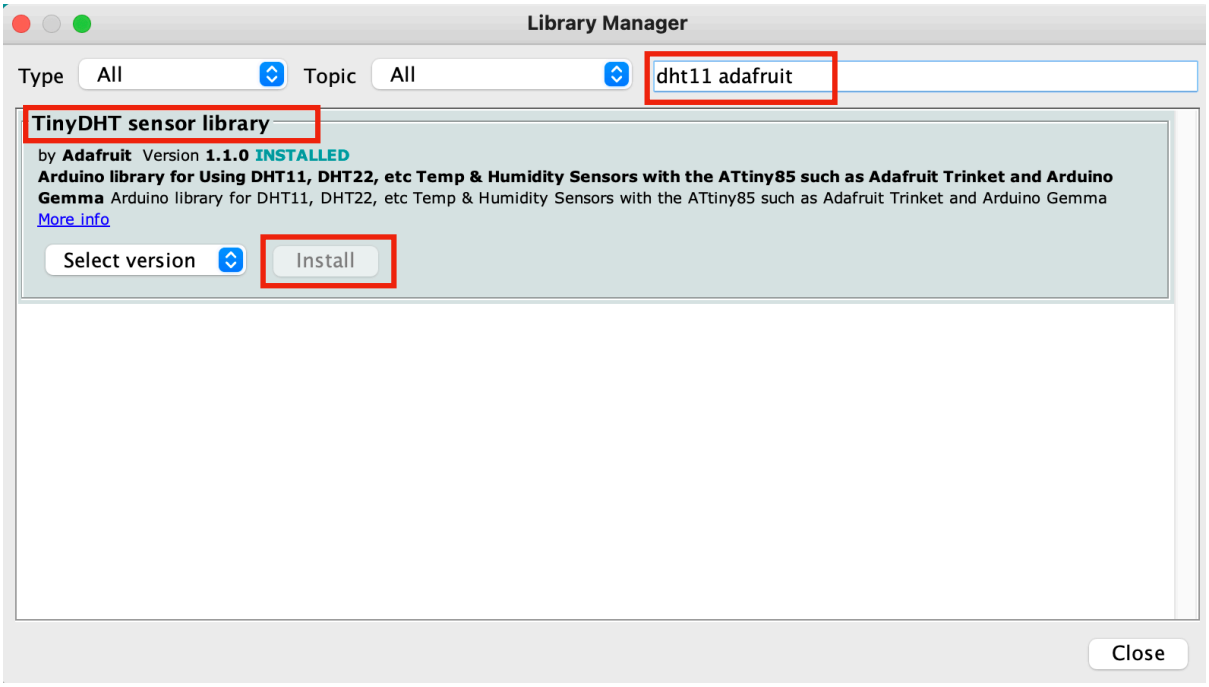
- Use the DHT11 sensor to print both the temperature and the humidity values to the Serial Monitor.

Let's begin by first including the DHT11 library.

With your Arduino sketch opened, click on **Sketch -> Include Library -> Manage Libraries...** like shown below:



Next, search for “dht11 adafruit” and install the latest version of the **TinyDHT sensor library** as shown below:



The **TinyDHT sensor library** allows you to read data from the temperature and humidity component easily.

Now that the library has been installed, let’s include it in the sketch.
Add the following line of code just below the WiFi and WebServer libraries:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "DHT.h"
```

As always, let’s also add other useful variables just before the setup function:

```
// Initialise the Temperature and Humidity pin
const int temp_hum_pin = D6;

// Initialise variables to store the temperature and humidity values
int temperature = 0;
int humidity = 0;

// Initialise the DHT11 component
DHT dht(temp_hum_pin, DHT11);
```

Type the above code just before the setup() function.

Here we initialise all the variables needed to measure the temperature and the humidity values using the DHT11 sensor. **temp_hum_pin** is a reference to the digital pin D6. **temperature** and **humidity** store the values read by the DHT11 sensor. **DHT dht(temp_hum_pin, DHT11)** creates an object of type DHT and configures it with the pin number (D6) and the sensor type (DHT11).

Next we want to add some code to the setup() function:

```
  // Start the dht component reading
  dht.begin();
```

Type the above code at the end, but inside, the setup function.

dot.begin() initialises the readings of both the temperature and the humidity values.

At this point, we have configured the DHT11 sensor and initialised the readings inside the setup function. We can now use one additional utility function to code the temperature and the humidity readings.

The **readTempHum()** function:

```
// Read the temperature and humidity values
void readTempHum(){

  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  Serial.println(temperature);
  Serial.println(humidity);

}
```

Type the above code at the end of the sketch, together with the already existing utility functions.

The **readTempHum()** utility function is relatively easy. It first registers the temperature and humidity values using the library built in functions **readTemperature()** and **readHumidity(),** and successively stores the readings inside the temperature and humidity variables. Finally, it prints both values on the Serial Monitor.

Now that we have the utility function in place, it is time to put the code together in the **loop()** function to read the temperature and humidity values from the DHT11 sensor.

The **loop()** function:

```
// Only execute if the fridge is ON
if (fridgeOn()){

  // Print the fridge temperature on the serial monitor
  Serial.println(fridgeTemperature());

  // Signal temperature status
  temperatureStatus(fridgeTemperature(), fridgeOn());

  // Signal critical temperatures
  trigBuzzer();

  // Read the temperature and humidity
  readTempHum();
```

Add the call to the **readTempHum()** function inside the loop() function as shown above. Notice that the call to the **readTempHum()** function only happens when the fridge in ON (fridgeOn) as we only want to read the temperature and the humidity when the fridge is ON.

Hurray, you have successfully completed the code section of the circuit.
Now you have a more advanced smart fridge circuit.
Go ahead, compile your code, and upload it to your ESP board.

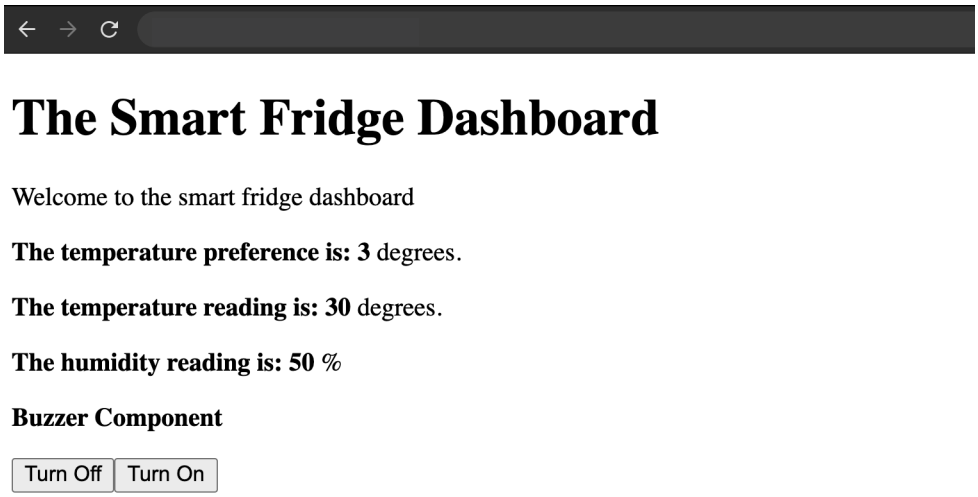
When you open the Serial Monitor, you should see the temperature and the humidity values printed.

Step Five
Additional Tasks

In upcoming exercises, you will add one more component to the smart fridge circuit.

Meantime, try the following:

- Tidy up the Serial Monitor print of the temperature and humidity values so that it is more clear. What about “the temperature is: <>” and “the humidity is: <>”?
- Add the DHT11 readings to your smart fridge web server dashboard. Update the **get_index()** function to replicate the following dashboard:



Just a note that the first line of the dashboard refers to the potentiometer reading while the second one refers to the DHT11 reading.