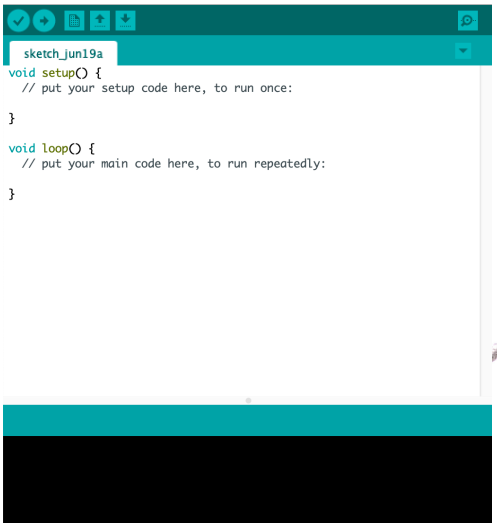
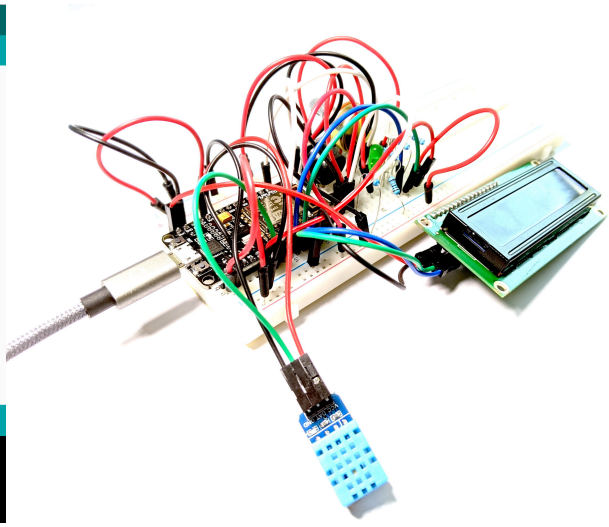
Smart Fridge

Part five: The I2C LCD display

Project description:

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



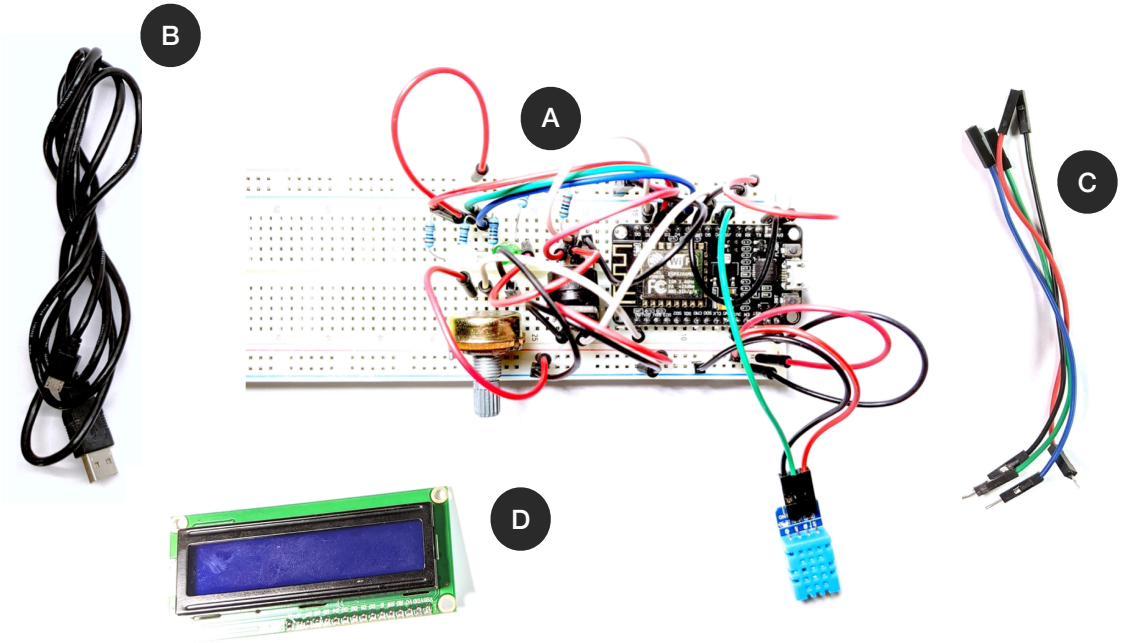


Project objectives:

- Get familiar with the I2C LCD display
- Add the sensor to the smart fridge circuit
- Print both the recorded temperature and humidity values on the I2C LCD display

Project components:

Component Reference	Component Quantity	Component Name	Component Description
A	1	Smart Fridge Part 4	The smart fridge circuit which features the potentiometer, the switch, the buzzer, the DHT11, and the RGB LED components
B	1	Micro USB Cable	A USB cable to power and upload instructions to a microcontroller
C	4	Jumper Wires	Conductive cables frequently used with a breadboard to connect two points in a circuit
D	1	I2C 16x2 LCD display	An electronic device that it is able to display 16x2 characters on 2 lines, white characters on a blue background.



Step One
Introduction and Theory

In the fourth part of the smart fridge exercise, you learnt how to configure the DHT11 temperature and humidity sensor component and included it to your smart fridge circuit. Furthermore, you added the temperature and humidity readings to your web server dashboard. This exercise introduces you to one new component: the I2C LCD display.

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 I2C LCD component to display the temperature and humidity values.

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

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

- **I2C 16x2 LCD display:** The I2C is ideal for displaying text/characters only. The16×2 character LCD has an LED backlight and can display 32 characters arranged into two rows and sixteen columns. This module consists of 4 leads: the GND (ground), the VCC (5V), the SDA (Serial Data), and the SCL (Serial Clock).

The I2C module is deigned to avoid the use of numerous pins on your board. A regular LCD display without the I2C module, in fact, requires up to 7 pins to function correctly.

Step Two
Adding the component to the Smart Fridge Circuit

The circuit assembly for this exercise requires you to wire up one I2C 16x2 LCD display component.

You will add the I2C component to your existing smart fridge circuit that you have built so far. Refer back to the fourth part of the smart fridge exercise to view the latest version of the circuit.

Since the SCL and the SDA leads of the I2C component must be respectively connected to pin D1 and pin D2, you need to first make some small adjustments to the existing circuit configuration.

First, move the signal wire of the switch component from pin D1 to pin D0.
Next, move the RGB LED red signal wire from pin D2 to pin D7.

Remember that RGB LED components might have different configurations. Sometimes the component leads are inverted.

We will modify the code global variables of such pins later in the exercise.

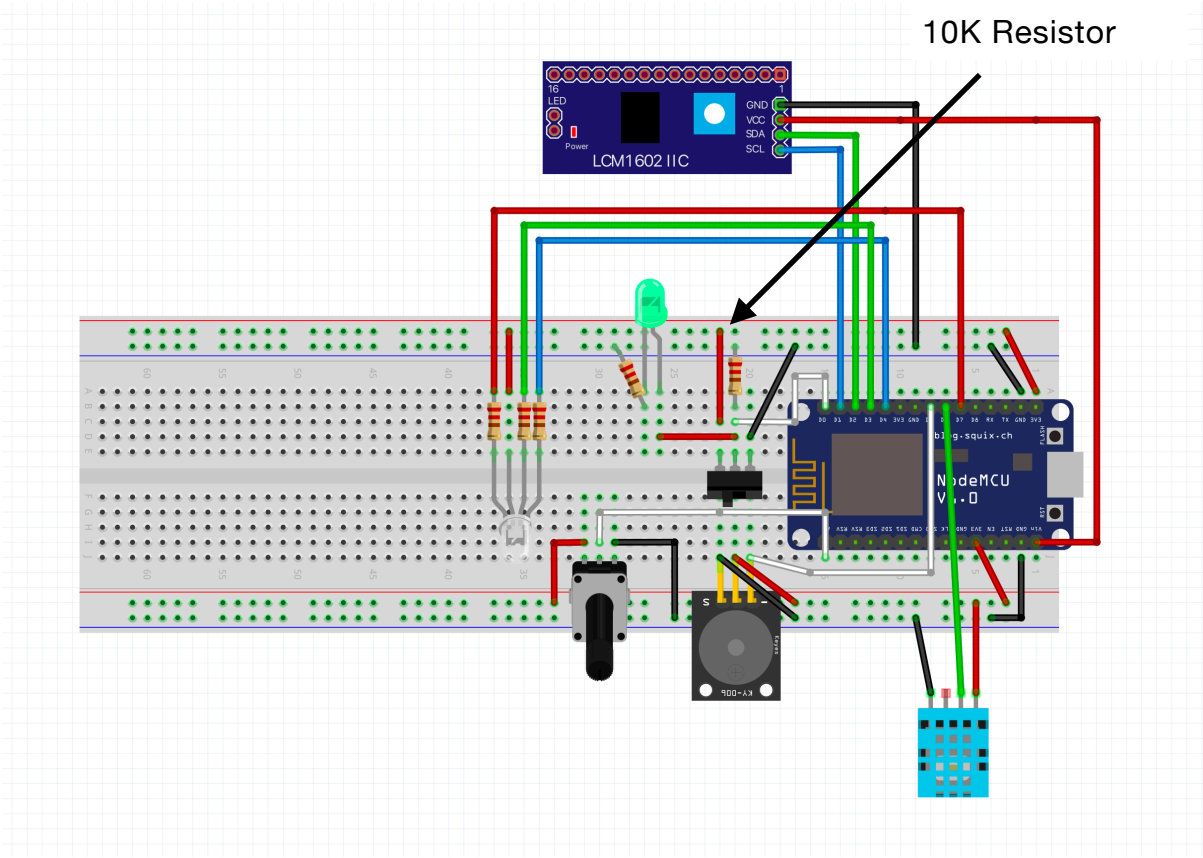
Now that pins D1 and pin D2 are free to use, the SCL and the SDA leads of the LCD display should be connected to digital pins D1 and D2 of your ESP board. You should also connect the VCC lead to the VIN pin (5V) and the GND lead to ground.

Here a quick recap of the circuit wiring:

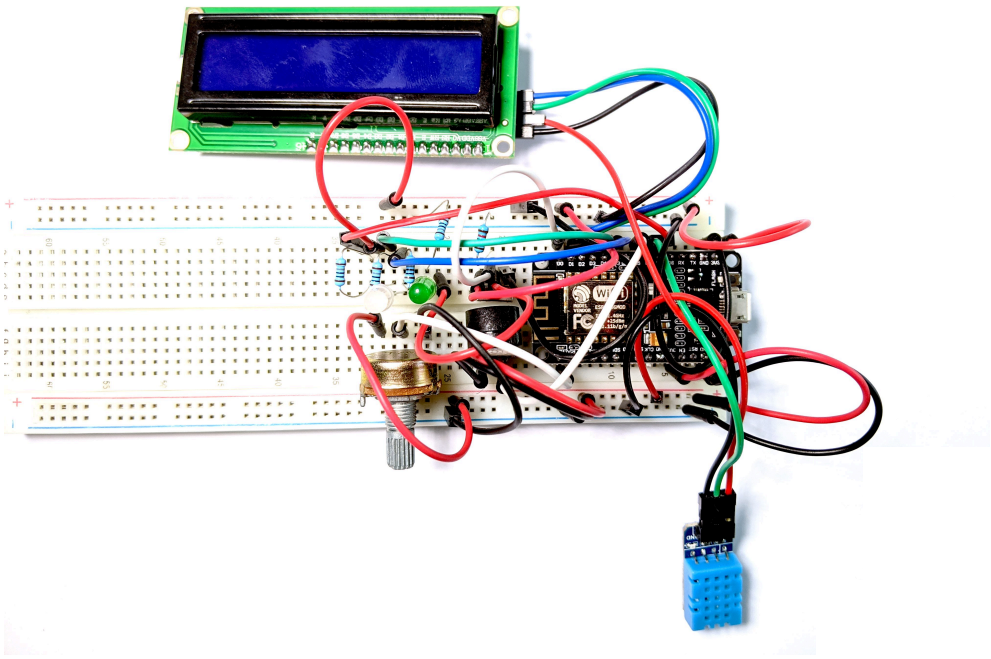
- **The smart fridge circuit:** You should have completed the first four parts of the smart fridge exercise as this component should be added to the latest version of that circuit.
- **Free D1 and D2 pins:** You should have moved the wire on pin D1 to pin D0 and the wire on pin D2 to pin D7.
- **I2C:** The SCL lead of the display should be connected to digital pin D1. The SDA lead of the display should be connected to digital pin D2. The GND lead goes to ground and the VCC lead goes to pin VIN (5V).

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 fourth part of the smart fridge circuit with the additional LCD display component:



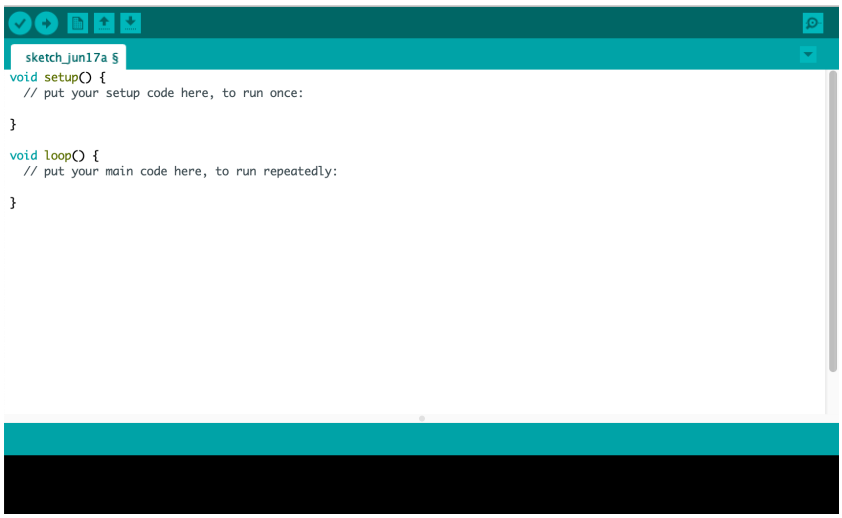
Furthermore, see below a picture of the circuit assembled:



Step Three
Writing the Code

Now that you have assembled the circuit, it is time to add the code for the LCD display 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_part5 for instance.

Now copy and paste the code that you wrote for the fourth part of the smart fridge exercise. You should have the code with the **fridgeTemperature(),** the **fridgeOn(),** the **trigBuzzer(),** the **temperatureStatus(),** the **readTempHum(),** 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 I2C display to print both the temperature and the humidity values registered by the DHT11 sensor.

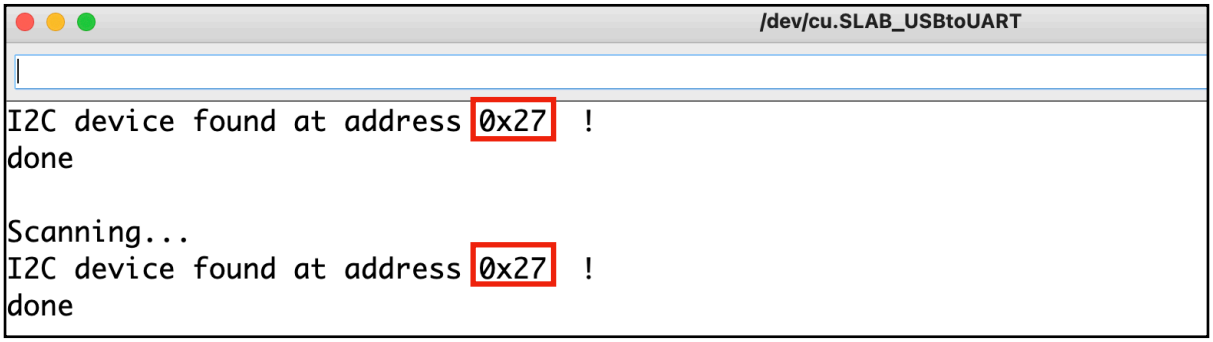
Let's begin by first identifying the address of the LCD display.
This will be necessary to control the LCD component using the official library.

Create a new temporary Arduino sketch and copy the **I2C_scanner** code that you find at the end of the following link:

https://playground.arduino.cc/Main/I2cScanner/

Save the sketch and upload it to the ESP smart fridge circuit.
Once uploaded, open the Serial Monitor.

You should see the address of the I2C LCD display:



In the above case the address is 0x27, make a note of it.

If the scan fails with an error message, simply unplug your ESP board from the USB cable and power it up again.

Now back to the **smart_fridge_part5** sketch.

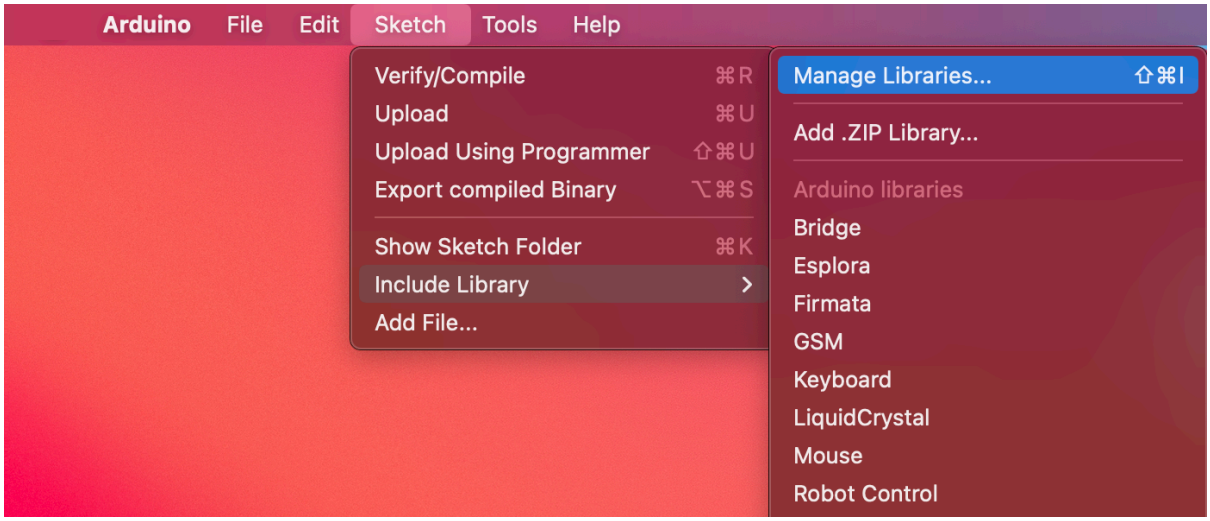
First thing first, let's change the global variables to free pins D1 and D2:

```
const int switch_pin = D0;
```
→ This was D1 originally

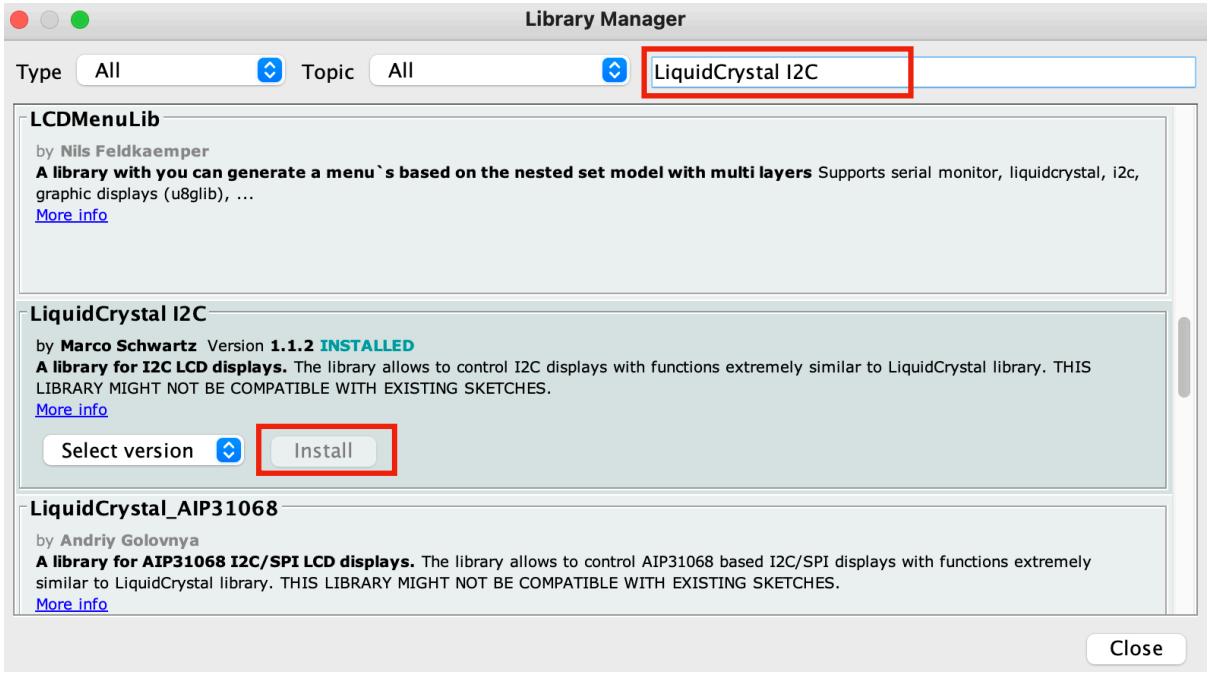
```
// Initialise the RGB pins
const int red_led_pin = D7;
const int green_led_pin = D3;
const int blue_led_pin = D4;
```
→ This was D2 originally

Next, let's add the **LiquidCrystal_I2C** library. Such library will facilitate the LCD component control.

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



Next, search for "LiquidCrystal I2C" and install the latest version of the **LiquidCrystal I2C** library as shown below:



Now that the library has been installed, let's include it in the sketch.
Add the following line of code just below the DHT library:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "DHT.h"
#include <LiquidCrystal_I2C.h>
```

Let's now initialise the lcd component just before the setup function:

```
// I2C address 0x27, 16 column and 2 rows
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

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

Here we initialise the lcd object using the LiquidCrystal library. The constructor takes three arguments: the lcd address (0x27), the number of columns of the lcd display (16), and the number of rows (2). Change the address to the one found during the **I2C_scanner** process. This might well be 0x27.

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

```
lcd.init(); // initialize the lcd
lcd.backlight(); // initialise the lcd screen light
```

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

lcd.init() initialises the lcd object so that we can work with it in the code
lcd.backlight() initialises the light on the lcd screen so that we can print on it

At this point, we have configured the I2C LED component and initialised it. We can now use one additional utility function to display the temperature and humidity values on the LCD display.

The **displayData()** function:

```
// Display the the temperature and humidity values on the LCD
void displayData(){

  String tmp ="Temp: ";
  tmp += temperature;
  tmp += " degrees";

  String hum ="Hum : ";
  hum += humidity;
  hum += " %";

  lcd.setCursor(0, 0);
  lcd.print(tmp);
  lcd.setCursor(0, 1);
  lcd.print(hum);

}
```

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

The **displayData()** utility function is relatively simple. It first constructs two strings **tmp** and **hum** to store the temperature and humidity text that we want to display. **lcd.setCursor()** is a function to specify the coordinates of where we would like to print the text and **print()** is the function which executes the printing. For instance, **setCursor(0,0)** prepares the printing to be done on the first column and first row of the LCD display. Here, we want to print the temperature on the first row and the humidity on the second row, just below the temperature.

Now that we have the utility function in place, it is time to put the code together in the **loop()** function to print the temperature and humidity values on the LCD display.

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();

  // Display the temperature and humidity on the LCD
  displayData();
```

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

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

You should now see that the LCD display prints the temperature and the humidity values. If you have trouble seeing the values, there is a small screw at the back of the I2C module to adjust the contrast of the display.

Step Five
Additional Tasks

This is the final version of the smart fridge exercise and no additional tasks are required. In upcoming exercises, you will keep exploring components on a brand new project: the smart door.

Meantime, revisit all of the smart fridge coding exercises and see if you have missed anything. If you really want to work on an extension task for this exercise, what about creating a function to display the readings of the other smart fridge components in turn on the LCD display?