**Description and Purpose**

This lab lets us discover some input devices such as a temperature/humidity meter, and also an output device such as an LCD to display the results of the values being read.

Then, you’ll get a chance to experiment by putting these two input-output functions together into one project, and also give the user customizability of the output by letting them select C or F values.

**Learning Objective(s)**

1. Familiarize yourself with libraries
2. Use things like a digital temperature sensor
3. Output values to an LCD

**Steps**

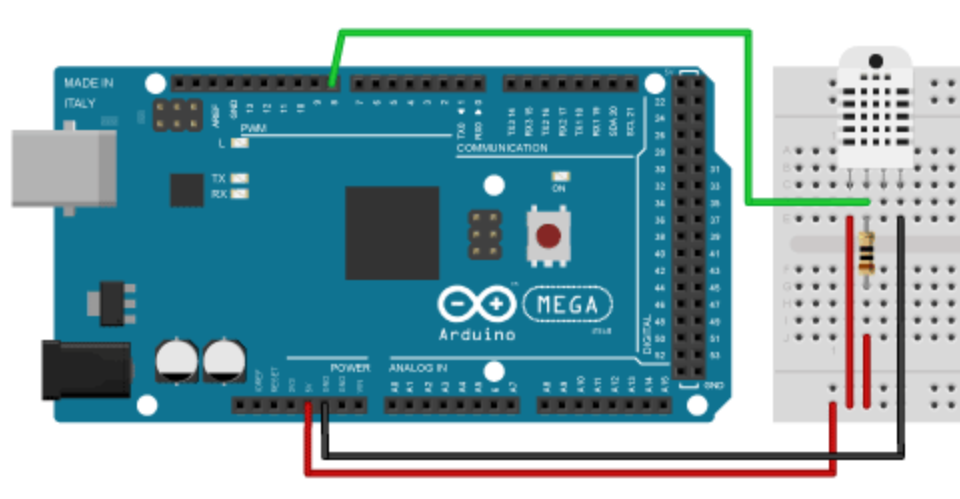
**PART 1**

**Step 1**

Prepare your project using a DHT11 or DHT22 temperature sensor. Create the project using your Arduino Uno and also with a breadboard. Try to create the project in a neat and orderly way, because it will be used for future projects moving forward.

Please refer to the slide “420-N35-LA 99.1 Reference Sheet - DHT11 & DHT22 Digital Temperature Humidity Sensor” for instructions on how to connect the temperature sensor, the links are at the end of this lab.

The following is for a 4-pin sensor ONLY. The resistor should be any value between 5KΩ to 10KΩ. For a 3 pin sensor, connect data to 8, and +V, GND to the pins indicated (no resistor needed for a 3-pin sensor). Use the same pins for whatever board you use (Mega, Uno).



**Step 2**

Using the hints on your “99 slide”, find a way to output the current temperature and humidity setting to your console. Display both the temperature in C and in F.

Copy the values you obtain here, use the “readTemperature” and “readHumidity” methods to help you. Use “readTemperature” to read both C and F (hint: use “true” as a parameter).

Display something like

Current temperature in C is: xx

Current temperature in F is: xx

Current humidity is : xx

Paste the results you get here.

|  |
| --- |
|  |

How do you detect if you can’t read the temperature (some kind of problem with the sensor)? The clue is here -> <https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-temperature-and-humidity-dht-sensors-9e5975>

**Step 3**

Do the above again, but use the built-in helpers called “convertCtoF” or “convertFtoC”. Paste the CODE used to convert your values here.

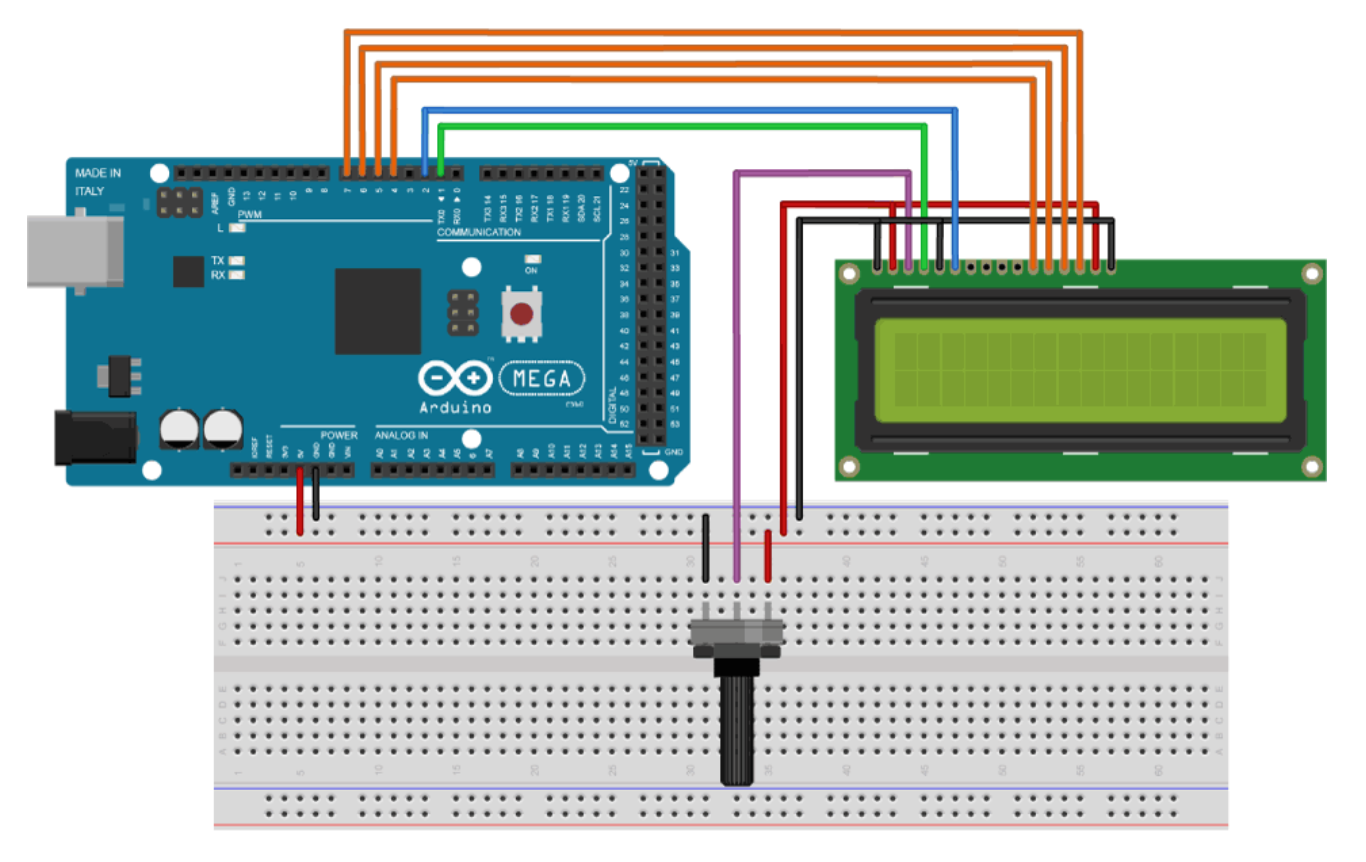
|  |
| --- |
| float currentTemp = sensor.readTemperature();  float currentTempF = sensor.readTemperature(true);  Serial.print("Temperature: ");  Serial.print(currentTemp);  Serial.print("C\n");  Serial.print("Temperature in F: ");  Serial.print(sensor.convertCtoF(currentTemp));  Serial.print("F\n");  Serial.print("Temperature in C again: ");  Serial.print(sensor.convertFtoC(currentTempF));  Serial.print("C\n"); |

**PART 2**

**Step 1**

Prepare your project using an LCD module (not an OLED module).

The variable resistor is 10kohm. (if you don’t know the value of your variable resistor, it’s MOST LIKELY 10K). Later on, if your LCD isn’t working properly, most likely one of these wires are placed incorrectly (there will be lots of wires needed for this one).



**Part 2 : Test your connection to the LCD**

The code necessary to make this LCD function is as follows:

#include <LiquidCrystal.h>

byte slash[8]= { // Array of bytes

B00001, // B stands for binary formatter and the 5 numbers are the pixels

B00010,

B00100,

B01000,

B10000,

B00000,

B00000,

B00000,

};

// Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)

LiquidCrystal lcd(1, 2, 4, 5, 6, 7);

void setup() {

lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display

lcd.createChar(7, slash); // Create a custom character for use on the LCD. Up to eight characters of 5x8 pixels are supported

}

void loop() {

for(int i=0;i<=15;i++) {

lcd.setCursor(i,0); // Sets the location at which subsequent

// text written to the LCD will be displayed

lcd.write(7); // Writes a character to the LCD

delay(1000); // 1 second delay

lcd.clear(); // Write a character to the LCD

}

}

**PART 3: Combine the projects and make them work together**

Create the above two projects, and you must implement the following changes:

**Step 1**

Combine both projects, and both sketches so that the LCD works and the Temperature sensor works. To test if the LCD works at this point, you can post something like “hello” to the screen (for now).

**Step 2**

Print the temperature and humidity to the screen in 2 lines:

Temp ....... : xx C

Humidity ... : xx RH

**Step 3**

Create a custom character on screen, which is a “degree” character.

Do not use the ascii version of it. The code already has a special character defined (in binary) as an example. A good tip, is to take graph paper out and graph the character first.

It should look like this: 

Your screen should look more like this now:

Temp ....... : xx °C

Humidity ... : xx RH

**Step 5**

Answer the following questions:

1. What is the function of the variable resistor on the board?

|  |
| --- |
| The function of the variable resistor is for adjusting the contrast of the LCD screen |

1. What is the difference between he DHT11 and the DHT22?

|  |
| --- |
| The difference between both DHT models is the color of the plastic covering to denote which type it is (Blue for 11 & White for 22) & the DHT22 is more accurate and expensive to buy. |

**Resources, Reference, Please See**

Please refer to course slides, including:

1. Class slides and videos (generally)
2. 420-N35-LA 99.1 Reference Sheet - DHT11 & DHT22 Digital Temperature Humidity Sensor
3. 420-N35-LA 99.2 Reference Sheet - SH-D1602 LCD Display

**Submission Procedure**

To submit your work, you must include the following as one ZIP file:

1. This document filled and completed.
2. The .ino code
3. A video made with your phone, showing the following:
   1. The assembled board with temperature and LCD installed.
   2. The LCD showing the current temperature value and the Celsius symbol.

**Grading Scheme / Rubric**

This will be graded on 10 points.

Everything included (video, document) = 5 points

Accurate answers in the document = 5 points