Lauren Courtney

February, 2020

EN 605.715.81 Embedded Systems

**Project 2 – Temperature Sensor**

**Requirements**

1. The Arduino shall record ambient temperature using a calibrated temperature sensor

a. It shall convert this temperature to Fahrenheit

b. This project shall use the DHT11 Temperature sensor, which is already calibrated

2. The implementation shall use a Round Robin with Interrupts Design

a. This program shall implement this design using a timer interrupt

3. The Arduino shall transmit the time and temperature across a Serial bus (USB) to the host

4. The Arduino shall take temperature readings at a periodic rate of 10 seconds.

5. The demonstration shall show these temperature readings at stabilized room temperature for about 1 minute, then the user shall place the Arduino and temperature sensor into the refrigerator for 5 minutes, continuing to record temperature, and afterwards take the Arduino out of the refrigerator and record the temperature of the room for another 5 minutes.

a. The total recording time for the demonstration shall be no more than 700 seconds.

6. The program shall plot the results of these temperature recordings in real time.

a. A Jupyter notebook will allow this graph to be displayed as the temperature is recorded.

7. The user may export the temperature and time data as a CSV after the recordings are finished.

**Design**

**Hardware**

1 Arduino – Mega 2650 (timer interrupt code would need to be revised to run on a different Arduino)

1 DHT11 Temperature and Humidity Sensor

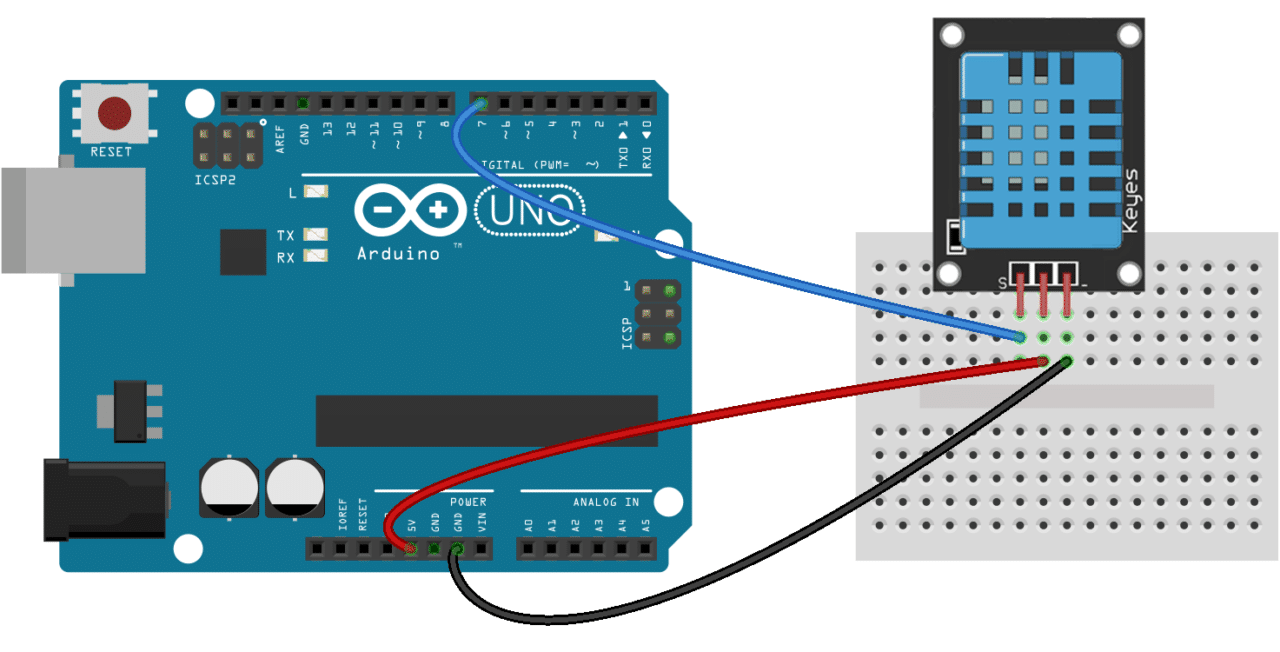
3 jumper cables

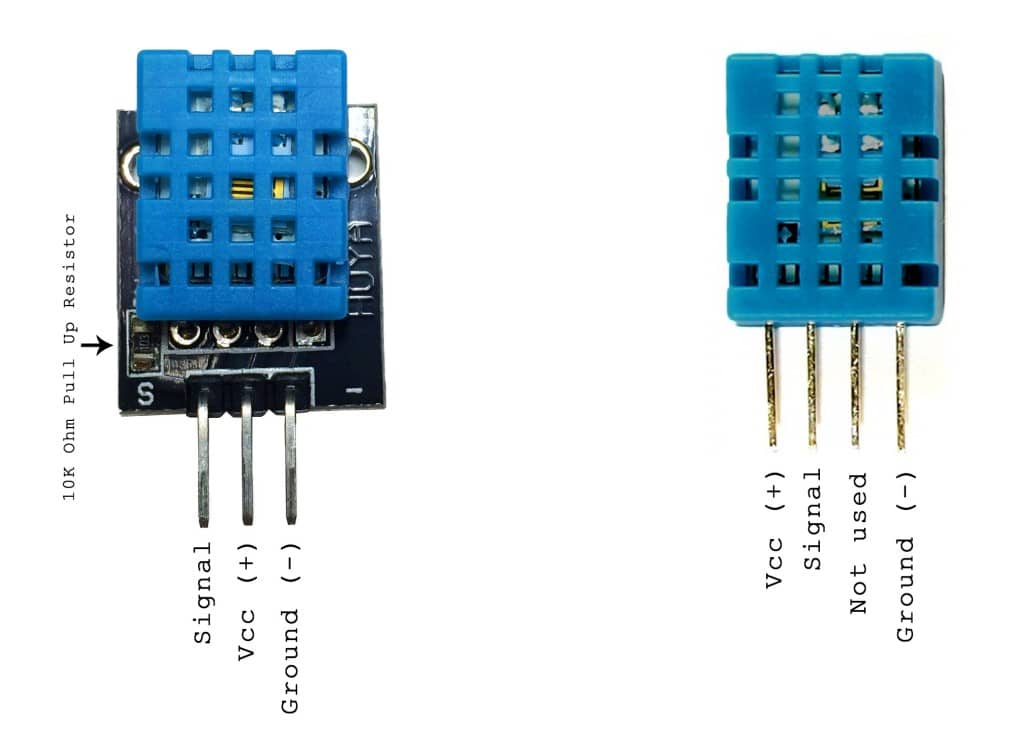
1 Breadboard

1 Laptop

1 USB cable

Design schematics and images from [Source 1](http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/)





Use the breadboard and the jumper cables to connect the Signal pin to digital pin 7 on the Arduino. Connect the VCC pin to the 5V power pin, and the Ground pin to GND. This sensor already has a resistor, so it is not necessary to worry about that.

**Software**

* The code will be written in C using Arduino Sketch for the temperature recordings and conversion. Reading the Serial data and plotting the real-time graph will be written in Python using Jupyter notebooks.
* DHT11.zip (Download available from [Source 1](http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/)). To import in Arduino Sketch: Sketch->Include Library->Add ZIP library -> choose downloaded DHT11.zip

I. Initialization

- set the DHT11\_PIN to pin 7 (where serial output of sensor is connected

- set Boolean get\_temp to True to get the first temperature at time 0

- ensure anything that will be used by the interrupt is volatile

II. Setup:

- Open the Serial port.

- Disable interrupts with cli()

- Set up CTC (clear timer on compare match) timer interrupt for timer 4 (16 bit) according to instructions in [Source 3](https://forum.arduino.cc/index.php?topic=625904.0)

- set the compare match register to 1 Hz (meaning the interrupt will match at a rate of 1 cycle per second, which will allow us a 1 second interrupt timer)

- Reenable interrupts

III. ISR:

- The interrupt service routine for the timer 4 interrupt

- Increment seconds (the interrupt will happen each second)

- If seconds are at a multiple of 10 (or whatever value we want to test at):

-Set get\_temp to True

-Otherwise

- Set get\_temp to False

III. Loop:

1. Check if get\_temp is true:

- This means it is time to take a temperature reading

- Use the time in seconds from our interrupt routine

- Read from the DHT\_11 to get the temperature in Celcius

- Convert the temperature to Fahrenheit

- Print the time and temperature to the Serial port separated by a comma

- The interrupt handler will ensure the temperature reading always arrives at 10 sec intervals

2. If get\_temp is false, continue looping until get\_temp is true

This processing guarantees that the Arduino will continuously transmit temperature data and time.

IV. Python notebook

- Set-up a serial connection to the Arduino through COM3

- Initialize a graph with time on the x-axis and temperature on the y-axis.

- Make the maximum size of the live graph 700 seconds

- While the demo is running:

- Read in time and temperature data over the serial connection

- Format the data so it can be displayed on the graph

- Update the graph as each new data point arrives as outlined in [Source 4](https://gist.github.com/brandoncurtis/33a67d9d402973face8d)

- After the demo, plot another graph with all the data

- Write the data to a CSV so it can be exported and loaded into another graphing program

**Demo**

Demo can be found on YouTube at https://youtu.be/2RLxJkFv3rY

**References**

Source 1: <http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>

Source 2: <https://www.instructables.com/id/Arduino-Timer-Interrupts/>

Source 3: <https://forum.arduino.cc/index.php?topic=625904.0>

Source 4: <https://gist.github.com/brandoncurtis/33a67d9d402973face8d>