Making an Arduino Temperature and Humidity Sensor

Inside and outside Model

Knowledge is a process of piling up facts; wisdom lies in their simplification.

Martin H Fischer

Modification Record

Issue	Date	Author	Changes (Including the change authority)
A	05 April 2021	ZizWiz	Original from various notes
B			

Contents

1. Introduction

This project attaches 2 of DH22 Temperature/Humidity sensors to the Arduino Nano allowing us to measure the temperature inside and outside at the same time. We will add code to the Arduino to display the data on the LCD screen.

We also write a Windows app that will allow us to bring back the data to a PC

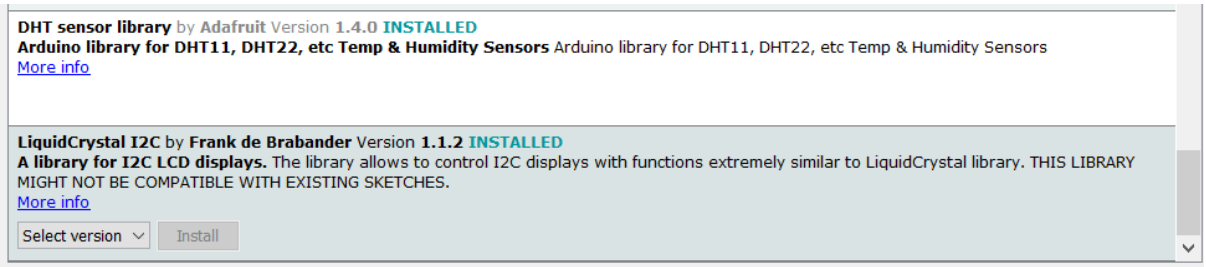
2. Parts list

For this project we will use:

- Arduino Uno or Nano
- 2x DHT22 Temp/Hum module
- I^2C 2x16 LCD screen

3. Libraries

To make this work a few libraries are needed. Using them makes this a lot simpler.



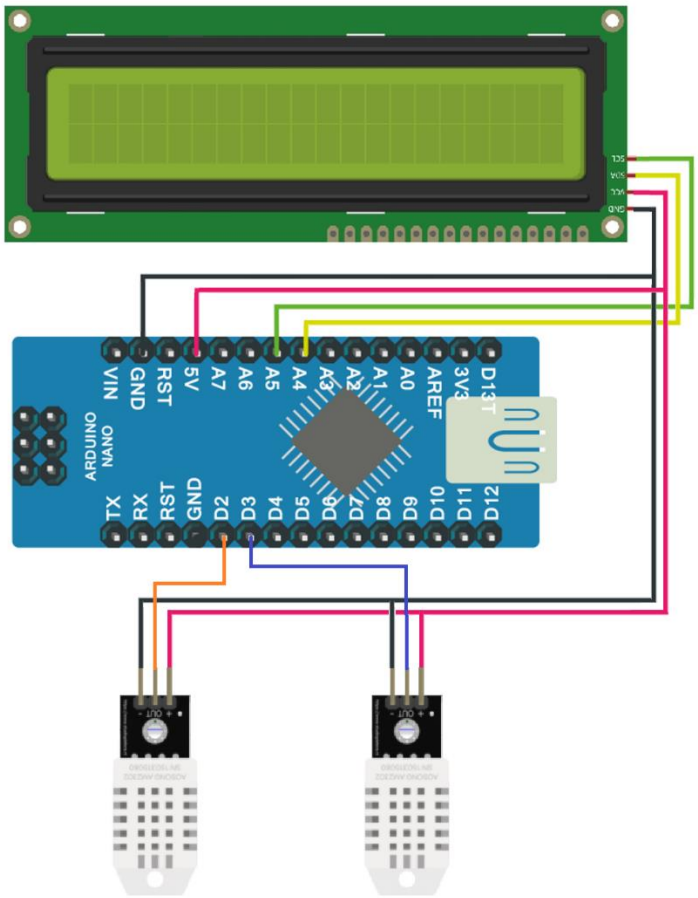
4. Find Modules Addresses

If you want you can find the address of each correctly wired I^2C device on your circuit.

You will find a code sample in Appendix A that you can use to scan for all the I^2C devices on your circuit.

5. DHT22/ AM2302 Temparature and Humidity Module

The unit I am building will have an Arduino Nano, a 2 line of 16 char LCD and two sensors DHT22. You can use DHT11 sensors, but they are less accurate than the DHT22.



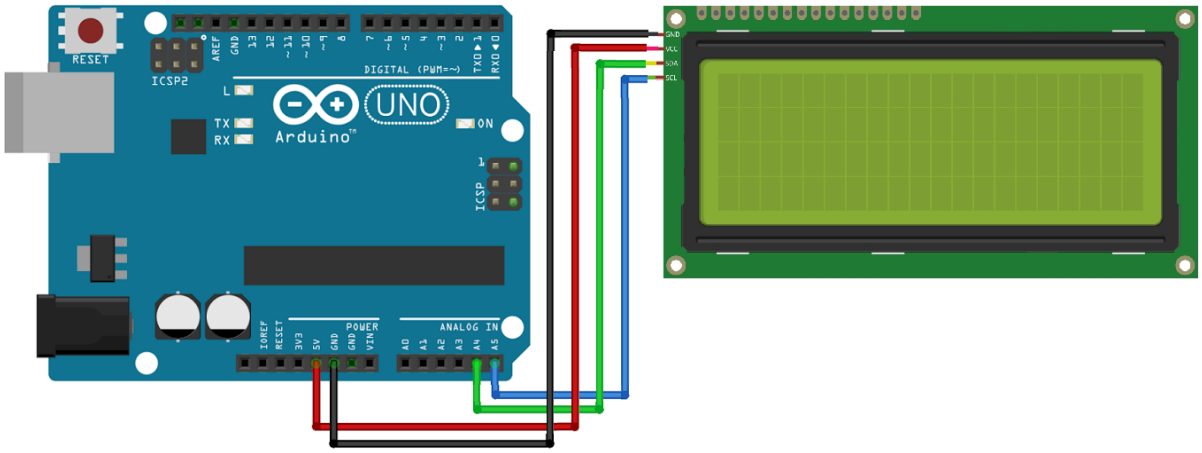
5.1.1. The DHT22 Spec

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)

For this example, we put code onto the Arduino to display the time on a 2-line LCD. We can also use a Windows App to bring back the data from the sensors to display in Windows.

The code to run this module is show in Appendix B. The code is commented to explain some finer points of what we do. The Windows app is also available in the GitHub Folder.

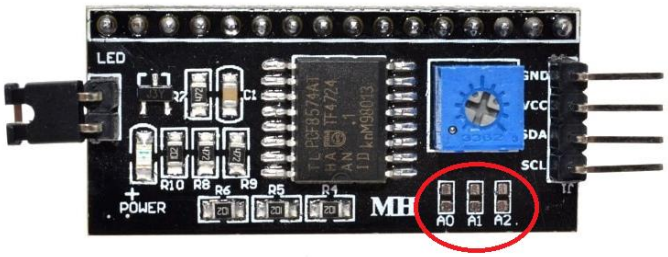
6. I²C LCD



The image is of a 4 line LCD. In my example I used a 2 line one.

6.1. I²C Address

Some I²C LCD module have an address selector solder pad.



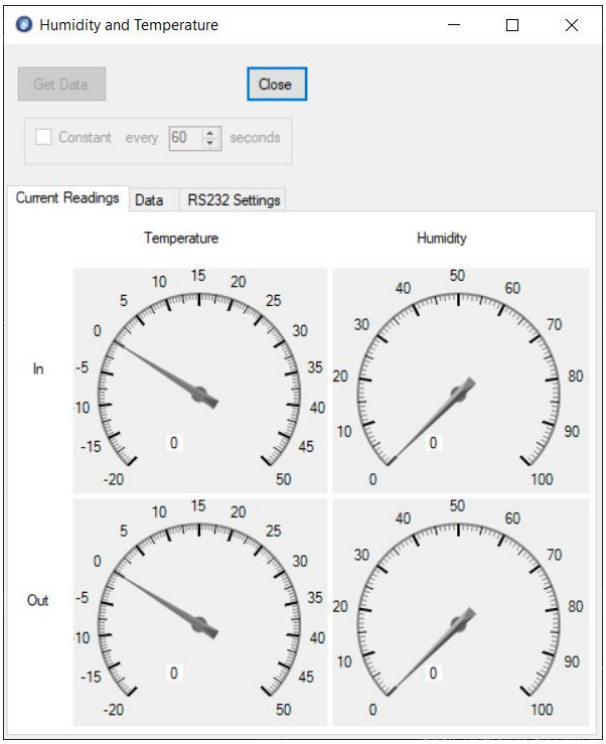
They are usually labelled with A0, A1, and A2. The following table show you how to interpret the selector. “1” = Not Connected, “0” = Connected.

A0	A1	A2	HEX Address
1	1	1	27
0	1	1	26
1	0	1	25
0	0	1	24
1	1	0	23
0	1	0	22
1	0	0	21
0	0	0	20

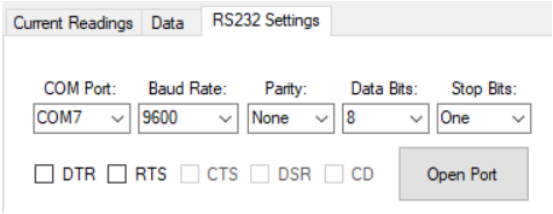
7. Humidity and Temperature Windows App

The aim of this app is to allow you to get the Inside and Outside humidity and temperature and bring it back to your PC/Laptop from where you can display it and/or save it to file. This is only a small app to show you how I did it. There are many ways to do it and the app can be updated to save the data or show it in graphs.

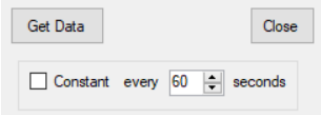
While doing this App I wanted to look at using some gauges that I had seen. I used this in the App and did modify it to how I wanted it to look. If you want the original code or to know more about it then I got it from http://www.ucancode.net/CSharp_Tutorial_GDI+_Gauge_Source_Code.htm. Thanks to the person who originally wrote it.



First to open the RS232 port to the Arduino. Do this in the RS232 tab

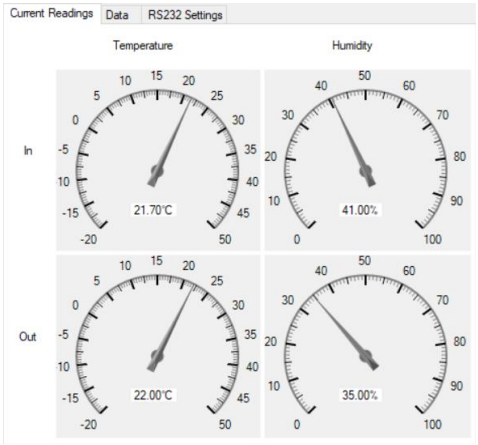


After this you will see that some of the buttons are now visible. Here you can get the data when you press the button or set it to run constantly if you tick the box and set the time in the box.



You will now see the data that is coming from the Arduino in the Data tab

The data will also be show in the gauges and written in the box below the Gauges.



The C# .Net solutions code is available should you need to adjust it to your needs.

8. Appendix A – Find I²C addresses

```
#include <Wire.h>

void setup() {
  Serial.begin (115200);

  // wait for serial port to connect
  while (!Serial)
    {
    }

  Serial.println ();
  Serial.println ("I2C scanner. Scanning ...");
  byte count = 0;

  Wire.begin();
  for (byte i = 8; i < 120; i++)
  {
    Wire.beginTransmission (i);
    if (Wire.endTransmission () == 0)
      {
      Serial.print ("Found address: ");
      Serial.print (i, DEC);
      Serial.print (" (0x");
      Serial.print (i, HEX);
      Serial.println (")");
      count++;
      delay (1);  // maybe unneeded?
      } // end of good response
  } // end of for loop
  Serial.println ("Done.");
  Serial.print ("Found ");
  Serial.print (count, DEC);
  Serial.println (" device(s).");
}  // end of setup

void loop() {}
```

10. Appendix B – Arduino Code

```
#include <DHT.h>;
#include <LiquidCrystal_I2C.h>

#define DHTPIN1 2 //define the pins we will use to get the data from.
#define DHTPIN2 3

LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2 line display

//create an array of sensors
DHT dht[] = {
  {DHTPIN1, DHT22},
  {DHTPIN2, DHT22},
};

//create arrays to put the data into
float humidity[2];
float temperature[2];

char SerialInput = '0';

void setup()
{
          lcd.init();  // initialise the class
    lcd.backlight(); //switch on the backlight

  Serial.begin(9600);  //open the RS232 port at 9600 baud
  for (auto& sensor : dht) {
    sensor.begin();
  }

//Write the non-changing data
   lcd.setCursor(0,0);
   lcd.print("I");
   lcd.setCursor(0,1);
   lcd.print("O");

}

void loop()
{


  //Read the sensor data and put into array
  for (int i = 0; i < 2; i++) {
    temperature[i] = dht[i].readTemperature();
    humidity[i] = dht[i].readHumidity();
```

```
  }

  int line = 0;
  String output = "";

// write the data to the LCD screen
  for (int i = 0; i < 2; i++) {

    lcd.setCursor(1,line);
    lcd.print(" ");
    lcd.print(temperature[i]);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(10,line);
    lcd.print(humidity[i]);
    lcd.print("%");

    // construct the data to send out the RS232 if required
    output += String(temperature[i]) + "," + String(humidity[i]) + ",";


    line = 1;
    }

    //read from serial port and send to Windows App

    if (Serial)
    {
    SerialInput = Serial.read();
    if (SerialInput=='1')
    {
      Serial.println(output); //send output to the App
    }
    SerialInput = '0';
    }


  delay(5000);
}
```