Make a Simple Weather Station Using Arduino

In this article, you will learn how to build your own weather station using the Arduino platform. This tutorial is based on the work from Steve Spence from Arduinotronics, and you can find the corresponding article on Instructables where you will find a v

Written By: Feitan
<table>
<thead>
<tr>
<th><strong>TOOLS:</strong></th>
<th><strong>PARTS:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino IDE (1)</td>
<td>SainSmart Uno (1)</td>
</tr>
<tr>
<td>library for the DHT sensor (1)</td>
<td>DHT-22 sensor (1)</td>
</tr>
<tr>
<td>To install the library, just download it and copy it to the /libraries folder inside your Arduino folder.</td>
<td>10K Ohm resistor (1)</td>
</tr>
<tr>
<td></td>
<td>1.8 graphical LCD from SainSmart (1)</td>
</tr>
<tr>
<td></td>
<td>Breadboard and a bunch of jumper wires (1)</td>
</tr>
</tbody>
</table>
Step 1 — Make a Simple Weather Station Using Arduino

There are two things you need to connect in this project: the DHT-22 sensor and the LCD display. For the DHT-22 sensor, you need to connect the signal pin to pin number 2 of the Arduino board.

Then, connect the power supply of the sensor: connect the VCC pin the +5V pin of the Arduino board, and the GND pin to the GND pin of the Arduino board. Finally, connect the 10k Ohm Resistor between the signal pin and the 5V power supply pin.

Then, you need to make the connections for the LCD screen. Connect SCL to Arduino pin 8, SDA to Arduino pin 9, CS to Arduino pin 10, DC to Arduino pin 11, and Reset to Arduino pin 13. Finally, connect the power supply: VCC to the Arduino +5V pin, and Gnd to the Arduino ground pin.

This schematic that summarizes all the connections is in Picture 2
Step 2

Now that everything is in place, we can start by testing the screen. This is the full code:

- `#include <TFT.h> // Arduino LCD library`
- `#include <SPI.h>`
- `// Pins`
- `#define cs 10`
- `#define dc 9`
- `#define rst 8`
- `// Create an instance of the library`

Step 3

- `TFT TFTscreen = TFT(cs, dc, rst);`
- `void setup() {`
  - `// Initialize the screen`
  - `TFTscreen.begin();`
  - `// Clear the screen`
  - `TFTscreen.background(0, 0, 0);`
  - `// Set font color to white`
  - `TFTscreen.stroke(255,255,255);`
Step 4

- // Set the font size
- TFTscreen.setTextSize(2);
- // Write some text on the screen
- TFTscreen.text("The screen is working!",0,0);
- }
- void loop() { }
- The Arduino TFT library really simplifies the task of using the LCD display, but let’s walk through the important parts of the code. First, we have to include the correct libraries:

- #include <TFT.h> // Arduino LCD library
- #include <SPI.h>
Step 5

- Then, we need to create the TFT screen object:

  ```
  TFT TFTscreen = TFT(cs, dc, rst);
  ```

- The next important step is to initialize the LCD:

  ```
  TFTscreen.begin();
  ```

- Finally, you can write some text on the LCD using the `text()` function:

  ```
  TFTscreen.text("The screen is working!",0,0);
  ```

- You can then upload the sketch to the Arduino board, and if you correctly wired the LCD screen, you should see the message “The screen is working!” printed on the screen.
Step 6

- We are now going to integrate the sensor with the screen. The code is quite long, but I will walk you through the essential steps. Here is the complete code:

  /* Arduino TFT text example This example demonstrates how to draw text on the TFT with an Arduino. This example code is in the public domain Created 15 April 2013 by Scott Fitzgerald

- #include <TFT.h> // Arduino LCD library

- #include <SPI.h>

- // pin definition for the Uno

- #define cs 10

- #define dc 9

- #define rst 8
Step 7

- // create an instance of the library
- TFT TFTscreen = TFT(cs, dc, rst);
- // char array to print to the screen
- char tempPrintout[6];
- char humPrintout[6];
- #include <DHT.h>
- #define DHTPIN 2 // what pin we're connected to
- // Uncomment whatever type you're using!

Step 8

- #define DHTTYPE DHT22 // DHT 22 (AM2302)
- int cycleTime = 2000;
- DHT dht(DHTPIN, DHTTYPE);
- float h;
- float t;
- void setup() {
- // Put this line at the beginning of every sketch that uses the GLCD:
  TFTscreen.begin();
Step 9

- // clear the screen with a black background
  TFTscreen.background(0, 0, 0);
- // write the static text to the screen
  TFTscreen.stroke(255, 255, 255);
- // set the font size
  TFTscreen.setTextSize(2);
- // write the text to the top left corner of the screen
  TFTscreen.text("Temp (F)", 0, 0);
- TFTscreen.text("Humidity (%)", 0, 60);
- // set the font size very large for the loop
  TFTscreen.setTextSize(4);
  dht.begin();

Step 11

```cpp
void loop() {
  // Read the value of the temp/humidity sensor on D2

  // Reading temperature or humidity takes about 250 milliseconds!

  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

  h = dht.readHumidity();
  t = dht.readTemperature();

  t = (t*1.8)+32; //C to F conversion

  String tempVal = doubleToString(t, 2);
}
```
Step 12

CONVERT THE READING TO CHAR ARRAY

- String humVal = doubleToString(h, 0);
- // String sensorVal = String(1.234);
- // convert the reading to a char array
- tempVal.toCharArray(tempPrintout, 6);
- humVal.toCharArray(humPrintout, 6);
- // set the font color
- TFTscreen.stroke(255,255,255)
- // print the sensor value

Step 13

PRINT THE SENSOR VALUE

- TFTscreen.text(tempPrintout, 0, 25);
- TFTscreen.text(humPrintout, 0, 85);
- // wait for a moment
- delay(cycleTime);
- // erase the text you just wrote
- TFTscreen.stroke(0,0,0);
- TFTscreen.text(tempPrintout, 0, 25);
- TFTscreen.text(humPrintout, 0, 85);
}
Step 14

//Rounds down (via intermediary integer conversion truncation)

String doubleToString(double input, int decimalPlaces) {
    if (decimalPlaces != 0) {
        String string = String((int) (input * pow(10, decimalPlaces)));
        if (abs(input) < 1) {
            if (input > 0)
                string = "0" + string;
            else if (input < 0)
                string = string.substring(0, 1) + "0" + string.substring(1);
        }
        return string.substring(0, string.length() - decimalPlaces) + "." + string.substring(string.length() - decimalPlaces);
    } else {
        return String((int) input);
    }
}

Step 15

string = string.substring(0, 1) + "0" + string.substring(1); }

return string.substring(0, string.length() - decimalPlaces) + "." + string.substring(string.length() - decimalPlaces); }

else { return String((int)input); } }
Let's see what was added compared to the previous part where we just tested the LCD screen. We first need to include the library for the DHT sensor, define the correct sensor type, and create the instance for the sensor:

- #include <DHT.h>
- #define DHTPIN 2 // what pin we're connected to
- // Uncomment whatever type you're using!
- // #define DHTTYPE DHT11 // DHT 11
- #define DHTTYPE DHT22 // DHT 22 (AM2302)
- // #define DHTTYPE DHT21 // DHT 21 (AM2301)
- DHT dht(DHTPIN, DHTTYPE);
Step 17

Still in the setup() function, we need to initialize the sensor:

- dht.begin();

In the loop() part of the sketch, we can then read out the temperature and the humidity:

- h = dht.readHumidity();
- t = dht.readTemperature();

We then need to convert these two measurements into Strings, using the custom doubleToString() function. Before actually printed the value on the LCD screen, we need to convert these two Strings to char arrays:

- tempVal.toCharArray(tempPrintout, 6);
- humVal.toCharArray(humPrintout, 6);
Finally, we can print the measurements on the LCD screen with these two lines:

- TFTscreen.text(tempPrintout, 0, 25);
- TFTscreen.text(humPrintout, 0, 85);

You can now upload the sketch, and if everything is wired correctly, the temperature and the humidity should now be displayed on the LCD screen!

Finally, thanks again to Steve Spence from Arduinotronics for this excellent guide!