Building A Remote Sensing Weather Station

Harsh Rathee, Partner - Estifa'a Zaid

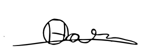
November 30, 2018

Abstract

In this experiment we build a remote sensing Temperature and Humidity Sensor. A Temperature and Humidity sensor is used the acquire the data and connected to a WiFi micro-controller to publish it on a local server. Raspberry Pi is then used to remotely read this data from the server and used to monitor and analyze the data. The data was successfully read and analyzed.

Declaration

I declare that this project and report is my own work.

Signature:  Date: November 30, 2018

Contents

1 Introduction

In this experiment, we use the DHT22 Temperature Humidity sensor to measure the temperature and humidity data. The DHT22 is conncted to the WiFi micro-controller Adafruit HUZZAH ESP8266. The micro-controller takes in the data from the sensor and uploaded on a local server on the LAN connection. This data is then remotely read by the Raspberry Pi by sending a URL request. The Raspberry Pi has to be connected to the same local LAN as the for it to be able to read the data. We then connect an LCD to the Raspberry Pi monitor the data with automatic regular updates of the data. The Raspberry Pi is also used to analyze and plot the values of Temperature and Humidity including using a live graph while the temperature is changed using a blow-dryer and the humidity is changed by blowing on the sensor. The temperature data from the DHT22 is compared against the data from DS18B20 temperature sensor whose data is read using the Raspberry Pi.

2 Theory

The ESP8266 processor from Espressif is an 80 MHz microcontroller with a full WiFi front-end (both as client and access point) and TCP/IP stack with DNS support as well. [1]

The DHT22 uses a capacitive humidity sensor and a thermostat to measure the temperature and humidity of the surrounding air and spits out. The downside to the sensor is that it only sends new data every 2 seconds. It measures humidity with an accuracy of 2-5% and is good for readings between 0-100%. It measures temperature with an accuracy of ± 0.5°C and is good for temperature readings between -40 to 80°C. [2]

The DS18B20 is a digital temperature sensor that provides 9 to 12-bit temperature readings which indicate the temperature of the device. It can measure temperatures between -10 to +85°C with an accuracy of ±.5°C. [3]

The LCD screen used is a 16×2 Character LCD. The LCD requires a total of 9pins to control(6 to control the LCD and 3 for the RGB background. The screen Dimentions are 27mm×71mm. [4]

3 Procedures

Components

The components used in the experiment are:-

- Raspberry Pi - To read the data from the server and monitor it.
- Adafruit HUZZAH ESP8266 WiFi micro-controller - To take the readings from the sensor and host it on a local server.
- DHT22 Temperature Humidity Sensor - The temperature and Humidity Sensor
- DS18B20 Temperature Sensor - Secondary temperature sensor to compare against the DHT22

- DC power supply - To power the micro-controller
- Windows PC - To program the micro-controller
- USB to TTL UART 6PIN CP2102 Module Serial Converter - To connect the micro-controller to the PC for uploading the code.
- Adafruit RGB 16x2 LCD and Keypad Kit - The LCD screen used to monitor the data

Connections

The circuits on the breadboard include connections between the DHT22 sensor and the WiFi micro-controller.1 To set up the micro-controller, the V+ pin on the ESP8266 is connected to the 5V power supply and the GND pin is connected to the ground. The 3.3V and the GND outputs are connected to the rails on the breadboard. The 3.3V and GND output are then connected to the DHT22 sensor to power it up and the data output from the sensor is connected to the GPIO #2 on ESP8266.Fig-3 The USB to Serial cable - Receive, Transmit and Ground Lines (Rx,TX, GND) are used to connect the ESP8266 to the Windows PC.

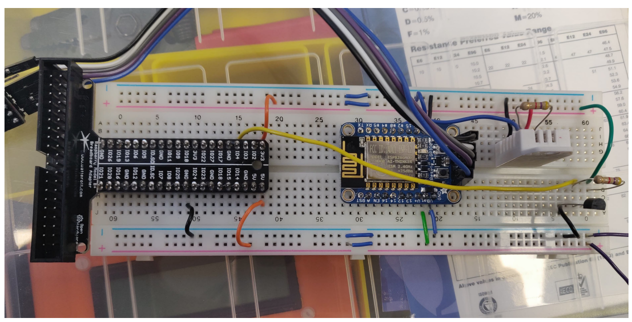


Figure 1: Figure showing connection on the breadboard

For monitoring the data on the LCD screen, the Raspberry pi is connected to the LCD screen using the GPIO extender plug.4 For comparing the DTH22 temperature sensor against the DS18B20 sensor, the Raspberry Pi is connected to the breadboard to connect the DS18B20 to it. For the connections of DS18B20, refer to Fig:2

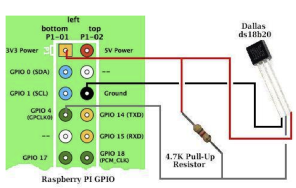


Figure 2: Figure showing connection of DS18B20 to the Raspberry Pi

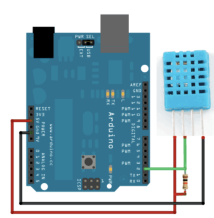


Figure 3: Figure showing LCD screen being connected to the PI

Setting up the WiFi server

To setup the Temprature and Humidity WiFi server on ESP8266, it is first connected to a windows PC using the USB to Serial cable. Then, the libraries and packages for the board and the DTH server are installed. Then the code for setting up the server is used from [5], replacing the SSID and password to the WiFi we want broadcast the data on. The code connects the micro-controller to the WiFi and when the conection is established, it starts a HTTP server. It then fetches the data from the sensor every two seconds and updates it on the server.

Setting up the LCD Screen

The LCD screen is connected to the Raspberry Pi, Fig:4. The LCD screen has 5 programmable buttons on it.

In the main thread of the display, we run an infinite loop to check if a button is pressed and if yes, we perform the function associated with that button. Out of the 5 buttons on the display, the select is used to return to the main display, two of the buttons are used to switch to the temperature and the other two to switch to humidity data.

Figure 4: Figure showing LCD screen being connected to the PI

Multi-Threading

After programming the the LCD to update the values of temperature and humidity when a button is pressed, the program is updated such that the Raspberry Pi continuously fetches new data at regular intervals and updates the LCD screen accordingly while also keeping track of any new button presses. This is achieved by using multi-threading. Three threads are created to update the value of temperature, humidity and the main display. Whenever a button is pressed, it starts the thread corresponding to the button which continuously fetches and updates the data on the LCD until another button is pressed.

Testing

The LCD display is then tested by varying the values of temperature by using a hair-dryer near the sensor and the values of Humidity is varied by blowing on the sensor.Fig-5 The varying values of temperature and humidity are plotted against time and against each other as well.

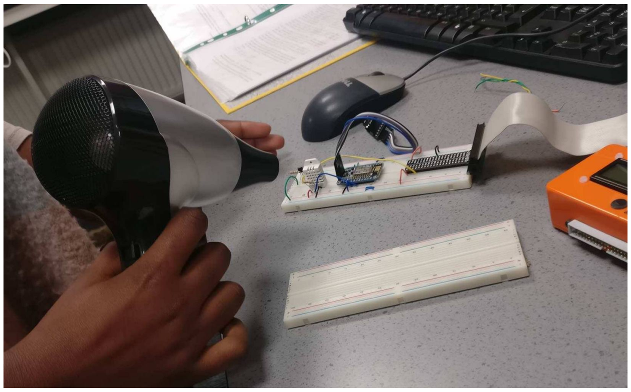


Figure 5: Varying the Temperature using blow-dryer

4 Analysis

First, the temperature readings from the DHT22 are recorded on a graph as it is varied using the hairdryer.Fig-6,7

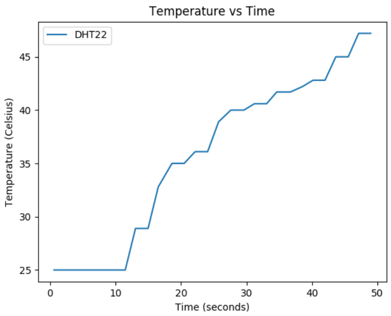


Figure 6: Temp(°C)vsTime(sec)

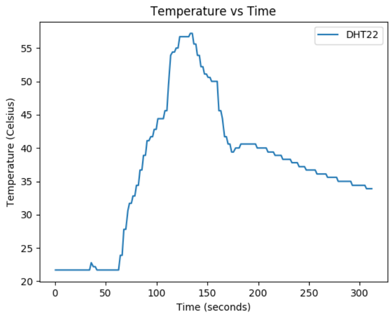


Figure 7: Temp(°C)vsTime(sec)

Then, the humidity readings from DHT22 are recorded on a graph as they are varied by blowing on the sensor.Fig-8

Then the values of temperature and humidity are plotted against each other as a blow-dryer is used to manipulate both of them.Fig-9 It can be seen that as the temperature due to the hairdryer increases, the humidity goes down, just like we would expect.

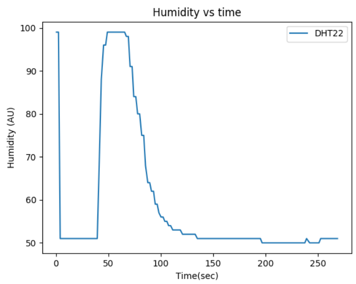


Figure 8: Humidity(A.U)vsTime(sec)

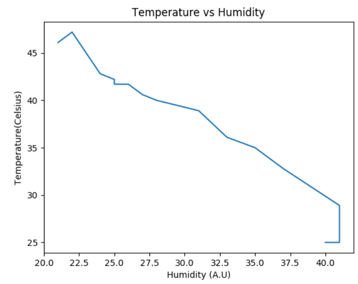


Figure 9: Tempe(°C)vsHumidity(A.U)

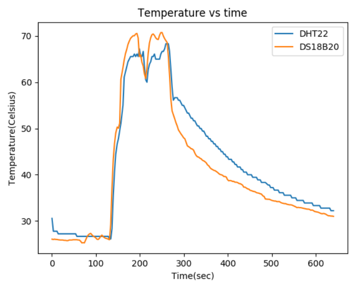


Figure 10: Temp(°C)vsTime(sec) for two sensors

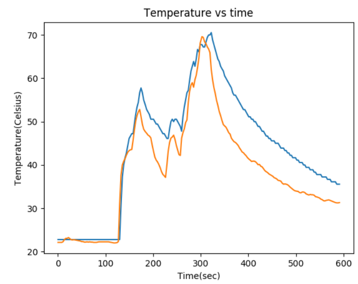


Figure 11: Temp(°C)vsTime(sec) for two sensors

Finally, the temperature readings from the two sensors are plotted on the same graph as they are both subjected to the same blow-dryer at the same time. Fig-10,11. It can be seen that DS18B20 heats up and cools down faster than the DHT22. This might be due to its smaller size that it can loose heat faster.

Finally, some of the values of temperature and humidity for normal room conditions are listed in Table-1

Table 1: Temp and Humidity data for Room Conditions.

T (°C)	H (A.U.)
21.11111111111111	51.0
21.11111111111111	52.0
21.11111111111111	52.0
21.11111111111111	52.0
21.11111111111111	52.0
21.11111111111111	51.0

5 Conclusions

The WiFi micro-controller was successfully used to read temperature and humidity data from the sensor and broadcast-ed on a local server. Raspberry Pi was then successfully used to fetch the data and monitor it. Graphs to study the variations in data were successfully plotted. In the future, this system can be used to measure the temperature data over several days and then can be used to predict the temperature for coming days. The code for predicting the temperature can be made better using regression techniques from machine learning.

References

[1] *More Information*, available at https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout?view=all#assemblye.

[2] *More Information*, available at https://learn.adafruit.com/dht.

[3] *More Information and Datasheet*, available at https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS

[4] *More Information and Datasheet*, available at https://learn.adafruit.com/adafruit-16x2-character-lcd-plus-keypad-for-raspberry-pi/overview.

[5] *Code for setting up WiFi server*, available at https://learn.adafruit.com/esp8266-temperature-slash-humidity-webserver/code.