

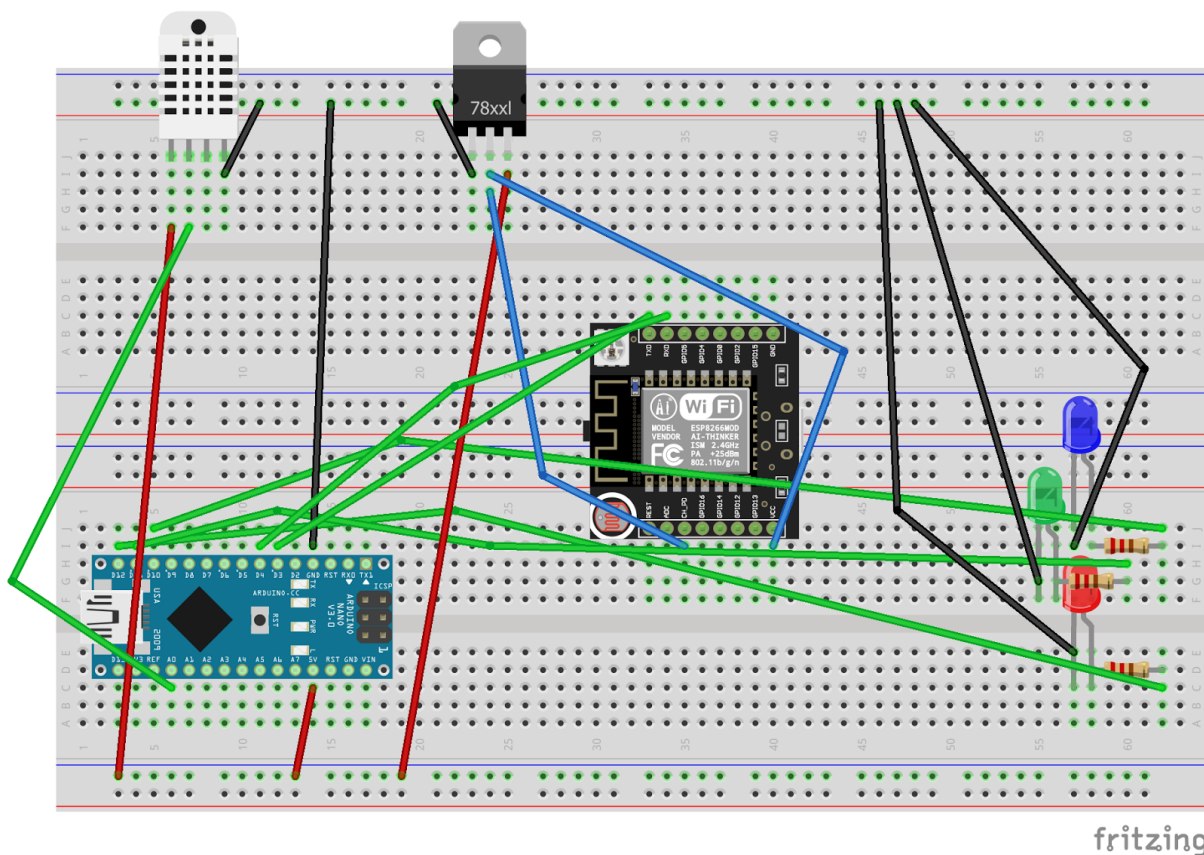
Wiszenko Pawel
Robineau Corentin

The Weather Station

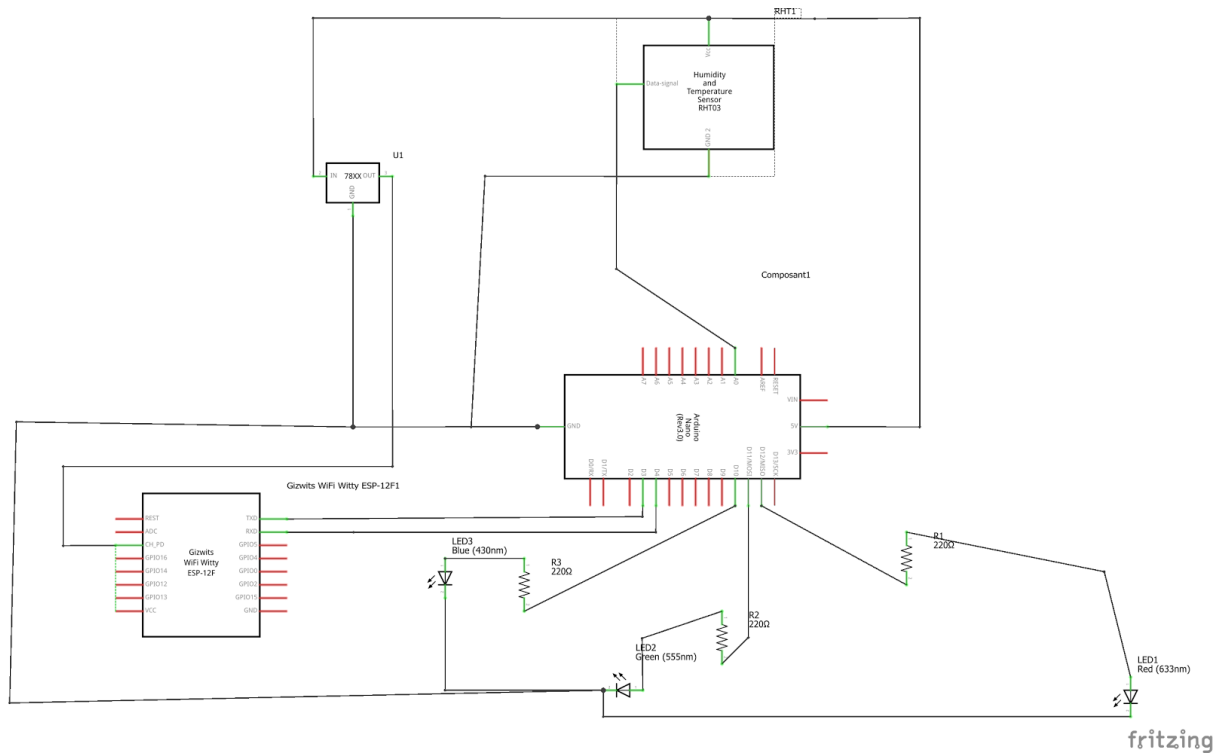
Description:

In Paris and Warsaw, weather is a daily issue that affects us all. Thus, we decided to create our own weather station in order to be able to react faster in case of unplanned hazards. Our station is made of a temperature and humidity sensor and is connected wirelessly, via wifi, to a server to which it sends the data. Therefore, it isn't depend on the computer and can be disposed in any place for better weather data accuracy.

The hardware:



This fritzing image represents the physical part of our weather station. However, the wifi module and the humidity and temperature sensors are slightly different than reality because we couldn't find ours on fritzing.



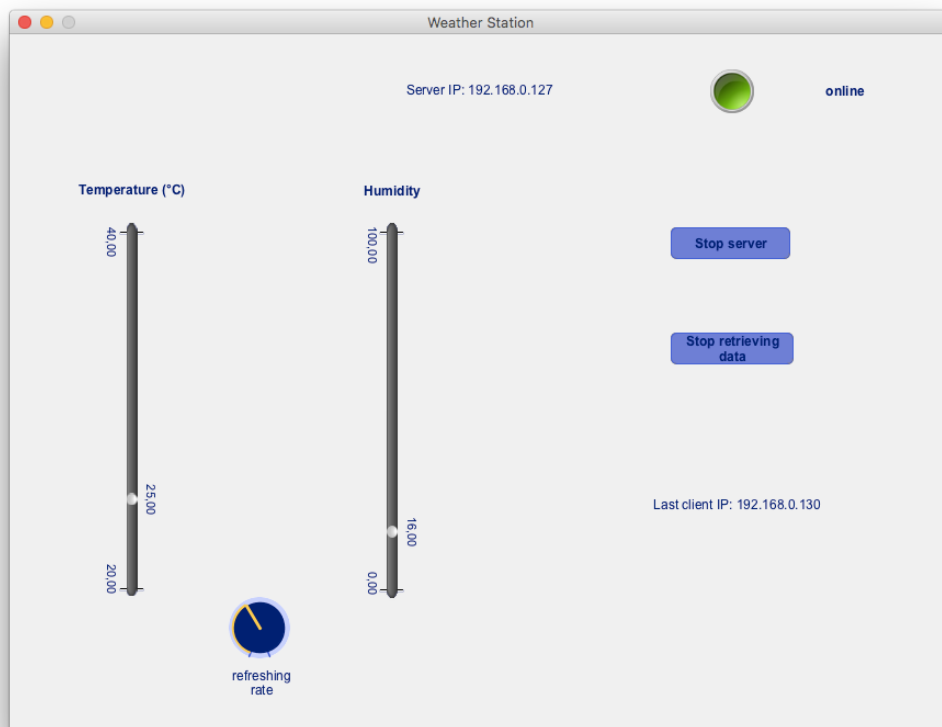
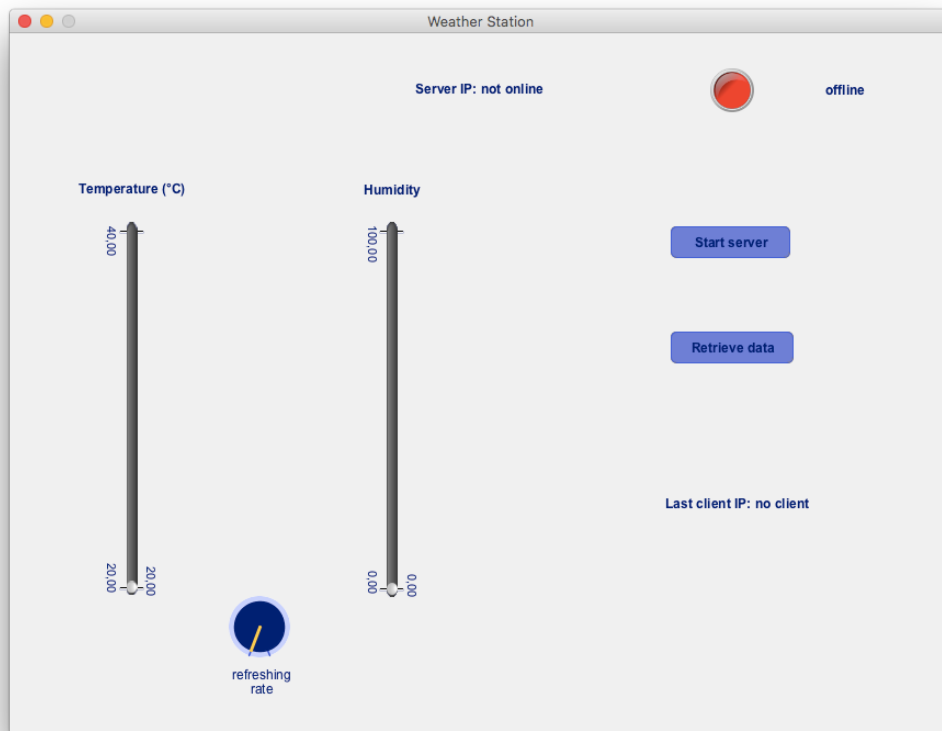
Power Module:

The wifi module needs to be powered with 3.3V tension. However, the arduino module provides a 5V tension. Thus, we had to use a voltage regulator. Usually, capacitors are used with the regulator to fix the current variations in order to provide a stable current. We didn't use capacitors because the tension delivered by the regulator was already stable.

Warning leds:

We're using 3 leds to display information. The red LED is used to indicate that the station isn't connected to a server. The green LED is used to indicate that the station is connected to a server. Finally the blue LED will blink every time data is transmitted to the server.

Server interface:



The server interface was designed with the help of G4P library and G4P GUI Builder.

The button 'Start server' creates a server and updates the top label with the IP of the server. Moreover the light that was red turns green.

Once the server is online, the 'Retrieve data' button can be clicked if a client is connected. From then, the servers starts asking the client for temperature and humidity at a certain rate. This rate can be modified using the GKnob labelled "refreshing rate".

The effect of changing the rate can be observed both on the server interface as the vertical sliders will move more or less often and on the arduino as the blue led will blink more or less often.

The vertical sliders 'enable' attribute has been set to false so that the user can't modify the value.

Server:

The server communicates with Arduino client and displays weather data as well as other indicators. It sends requests to clients for the temperature and humidity data every n milliseconds (where n is defined by user and is between 250ms and 5000ms).

Below is a part of code which is responsible for sending requests. It is executed in main server loop.

```
void retrieveData()
{
  if(!isRetrievingData)
    return;

  if(millis() - lastRetrievingTime > refreshingRate)
  {
    server.write("DHT");
    lastRetrievingTime = millis();
  }
}
```

Client:

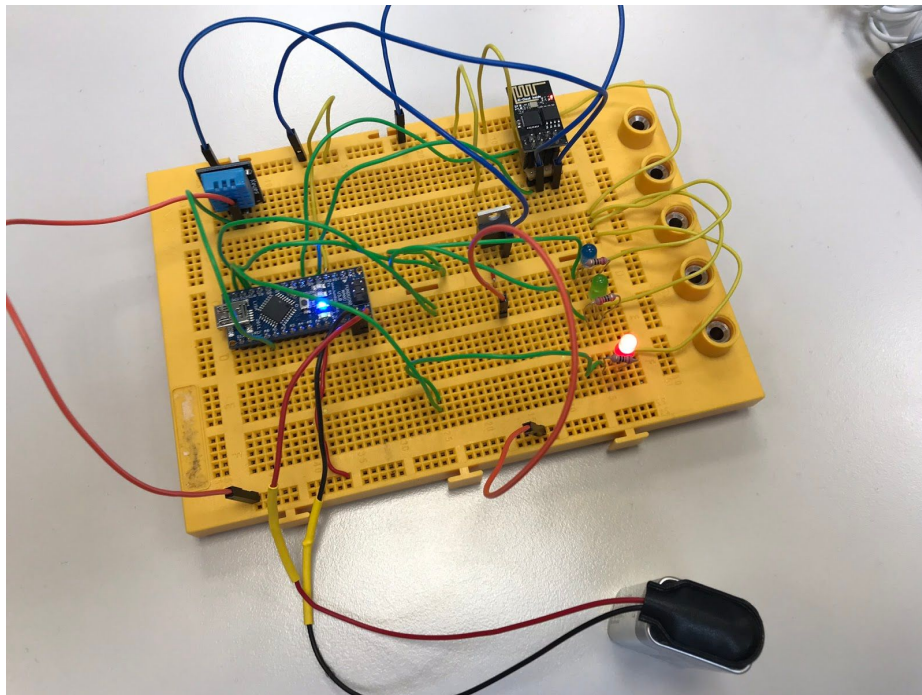
The arduino client waits for a request from the server and then immediately sends weather data from DHT sensor. Also every 10 seconds the client will test connection to the server and react accordingly.

Below is a part of code which interprets messages from the server and then sends a response containing temperature and humidity data to the server.

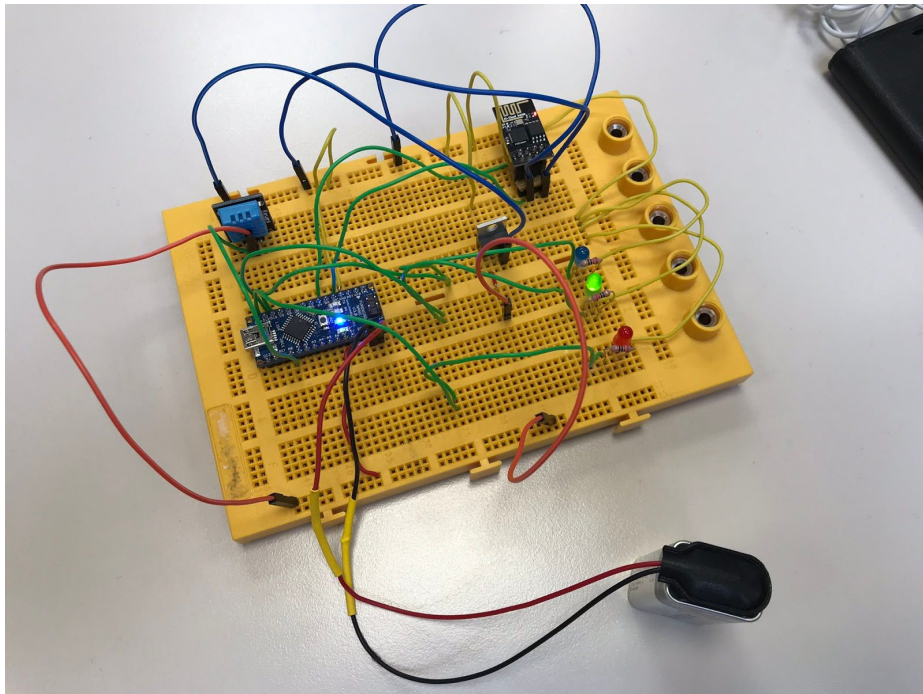
```
if (buffer_str.equals("DHT")) {  
    DHT.read11(DHT_PIN);  
    String data = "data" + delimiter + String(DHT.temperature) + delimiter +  
String(DHT.humidity) + "\n";  
    Serial.println(data);  
    char *msg = data.c_str();  
    wifi.send((const uint8_t*)msg, strlen(msg));  
}
```

Arduino photos:

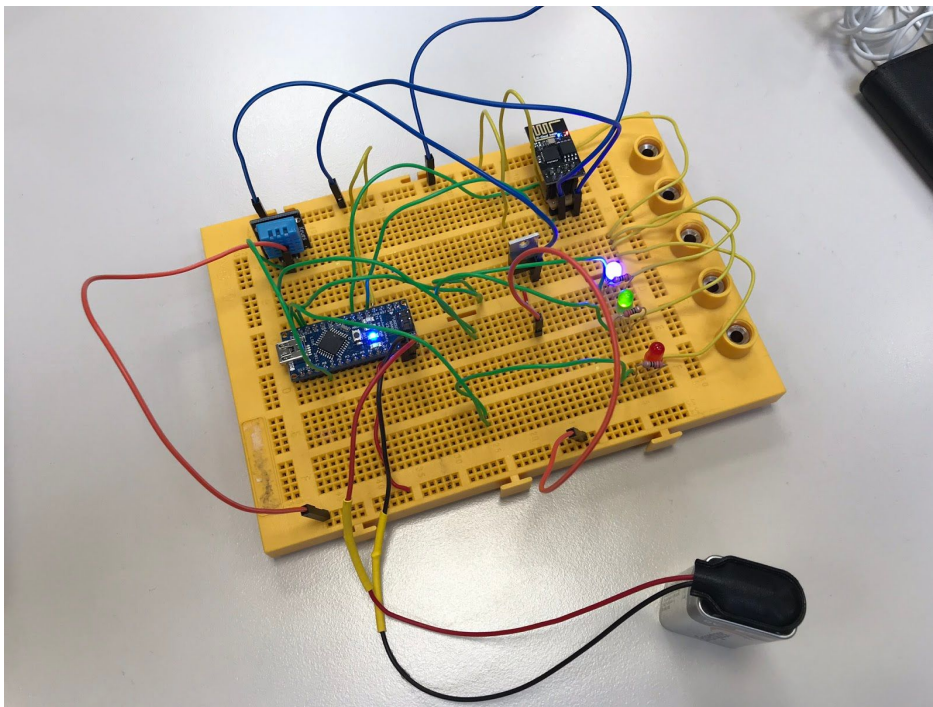
On our arduino board, red wires are connected to 5V, yellow wires to ground and blue wires to 3.3 V. Green wires are used to connect the different elements to digital and analog ports.



Here we can see that the red light is on which means that the station is on but is not connected to the server because it is offline.



On this picture, the station has successfully connected to the server because the green light is on



On this picture, we can see that the blue light is on which means that the server asked for data and the weather station is responding.