Blueprint product

Stefano Igbinosun
500859118

Logbook

# IOT LOGBOOK

## Week 1

This week I set up my GitHub and all necessary things that was needed for it. The blueprint I decided to do is the Weather station and ordered some parts for it.

I ordered a 1.3inch SPI TFT LCD Display RGB to show the data of the Weather station. Ordered a speaker, 8-bit RGB LEDs and as last a micro–DC Motor. Meanwhile I made a sketch for my

Weather station that is based on a windmill. The parts for the Weather station finally came in and tested it, but the LCD Display is not working properly don't know what the problem is.

The DC motors are working fine.

## Week 2

This week I fixed my performance review to know What I want to do for my blueprint. I started with the making of my site thru xampp not mamp for some reason my Apache server did

not start. I got my temperature sensor from Miquel for my Weather station plus I set up my WeMo’s for the Wi-Fi connection and is now successful.

## Week 3

For this week I was mostly busy with creating the site thru some tutorials on YouTube and dlo. Mainly trying to get the connection from my database to my site and WeMo’s.

## week 4

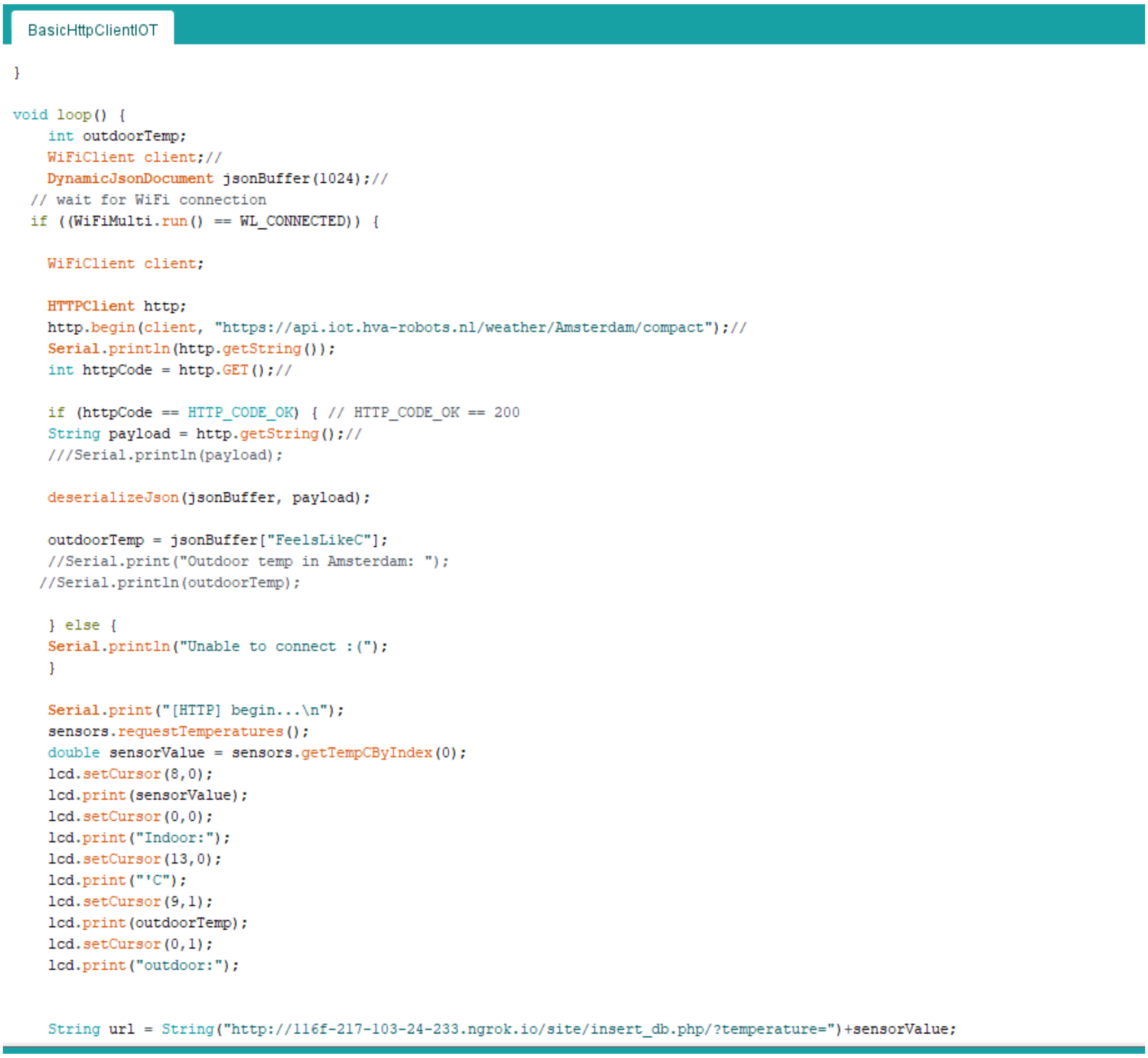
This week I finally have a working database with the connection to the web-application. I can now send data from the temperature sensor to database and with getdb.php file

the website can update itself with the data of the database through the getdb.php file. I also change the KY-015 TEMPERATURE AND HUMIDITY SENSOR MODULE for a KY-001 TEMPERATURE SENSOR MODULE

because KY-015 TEMPERATURE AND HUMIDITY SENSOR MODULE is an analog sensor and wasn't given me accurate data, so I change to digital sensor. For the rest I worked on my poster,

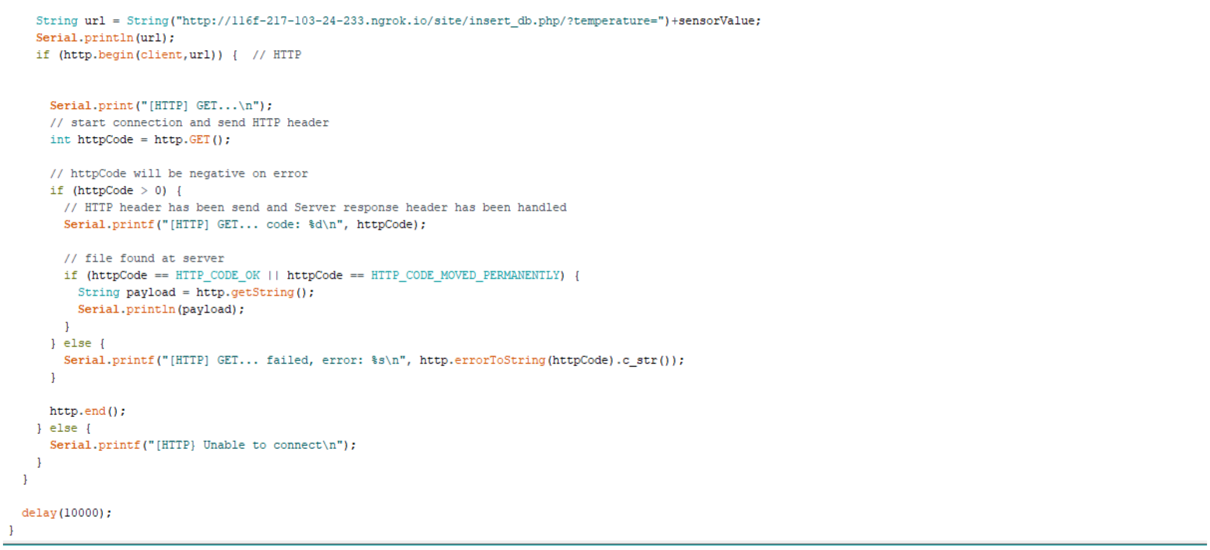
blueprint product, blueprint technical document and user manual.

Code



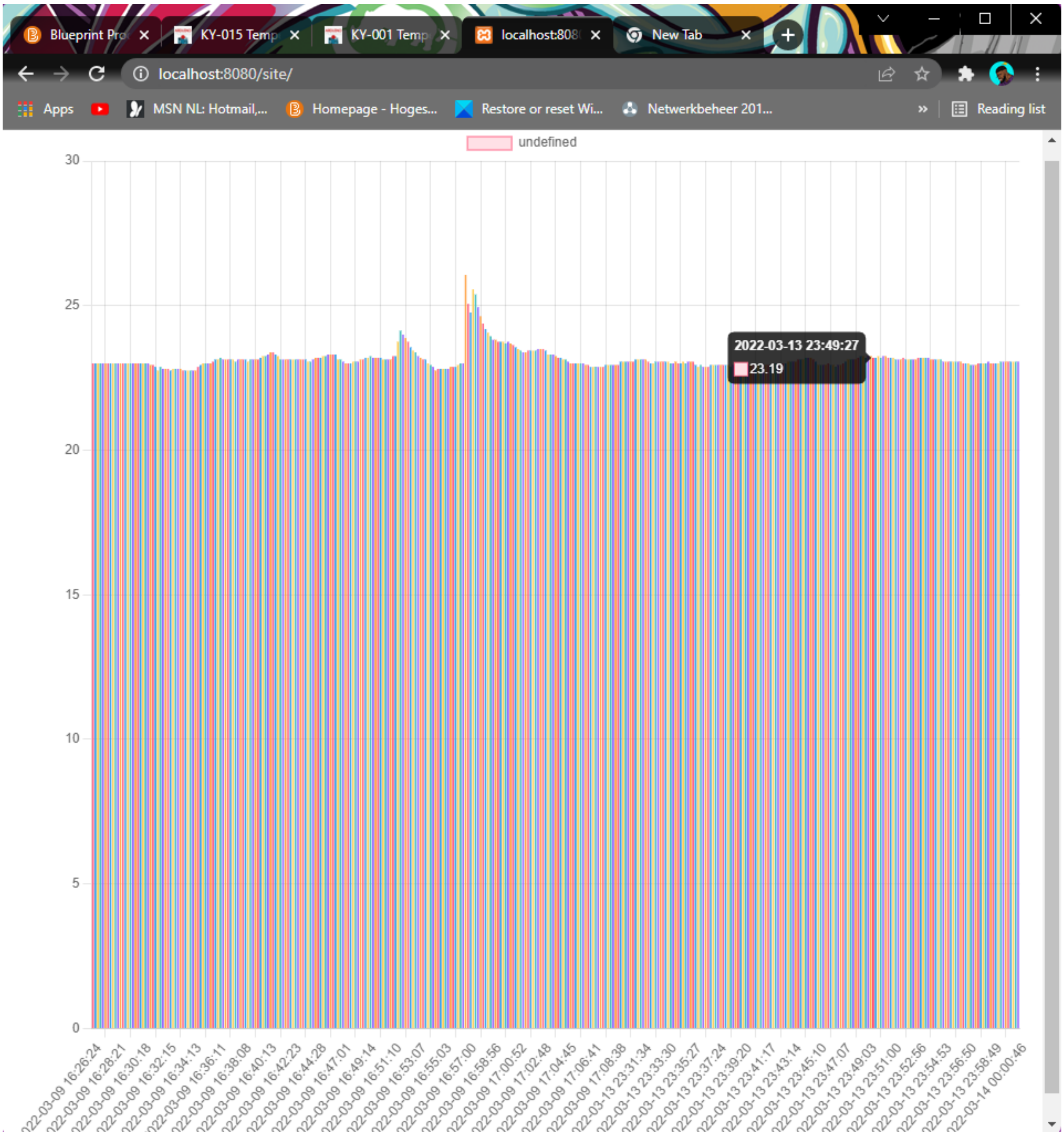
BasicHttpClientIOT

```
}

void loop() {
    int outdoorTemp;
    WiFiClient client;//
    DynamicJsonDocument jsonBuffer(1024);//
  // wait for WiFi connection
  if ((WiFiMulti.run() == WL_CONNECTED)) {

    WiFiClient client;

    HTTPClient http;
    http.begin(client, "https://api.iot.hva-robots.nl/weather/Amsterdam/compact");//
    Serial.println(http.getString());
    int httpCode = http.GET();//

    if (httpCode == HTTP_CODE_OK) { // HTTP_CODE_OK == 200
    String payload = http.getString();//
    ///Serial.println(payload);

    deserializeJson(jsonBuffer, payload);

    outdoorTemp = jsonBuffer["FeelsLikeC"];
    //Serial.print("Outdoor temp in Amsterdam: ");
   //Serial.println(outdoorTemp);

    } else {
    Serial.println("Unable to connect :(");
    }

    Serial.print("[HTTP] begin...\n");
    sensors.requestTemperatures();
    double sensorValue = sensors.getTempCByIndex(0);
    lcd.setCursor(8,0);
    lcd.print(sensorValue);
    lcd.setCursor(0,0);
    lcd.print("Indoor:");
    lcd.setCursor(13,0);
    lcd.print("'C");
    lcd.setCursor(9,1);
    lcd.print(outdoorTemp);
    lcd.setCursor(0,1);
    lcd.print("outdoor:");


    String url = String("http://116f-217-103-24-233.ngrok.io/site/insert_db.php/?temperature=")+sensorValue;
```



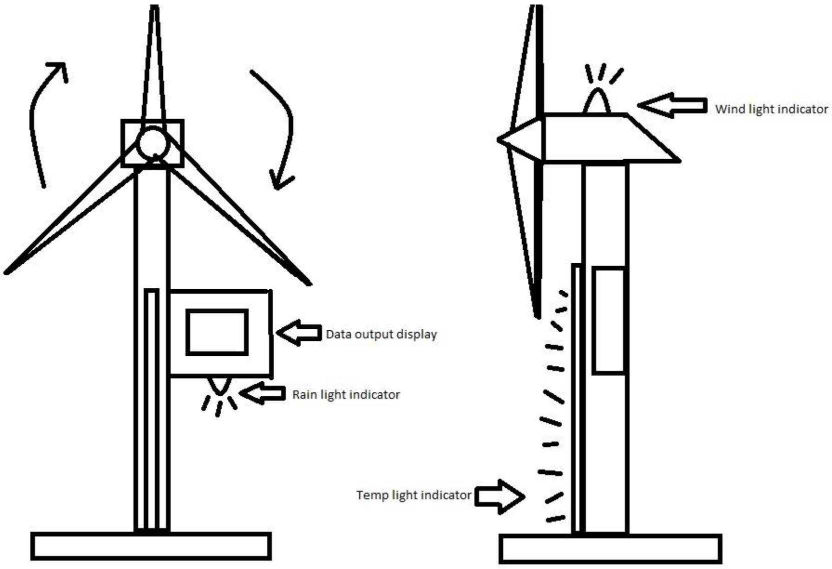
```
    String url = String("http://116f-217-103-24-233.ngrok.io/site/insert_db.php/?temperature=")+sensorValue;
    Serial.println(url);
    if (http.begin(client,url)) {  // HTTP


      Serial.print("[HTTP] GET...\n");
      // start connection and send HTTP header
      int httpCode = http.GET();

      // httpCode will be negative on error
      if (httpCode > 0) {
        // HTTP header has been send and Server response header has been handled
        Serial.printf("[HTTP] GET... code: %d\n", httpCode);

        // file found at server
        if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
          String payload = http.getString();
          Serial.println(payload);
        }
      } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
      }

      http.end();
    } else {
      Serial.printf("[HTTP} Unable to connect\n");
    }
  }

  delay(10000);
}
```

This the Code and documentation showing an embedded device (WeMo's) with an input (sensor) and an output (LCD Display). Also, with the use of an API as you can see in the code.



This the start of a web interface to the Weather station.

Design



Weather station windmill blueprint

The windmill station has a display (LCD) where the temperature data show. There are also LEDs that give different indicator of the weather. The rain led indicator will only blink blue if there is rain outside. The wind led indicator will blink with also the windmill propellers moving based on the wind speed outside. And finally, for more data output the temperature led indicator will light up by matching the temperature data on the display (LCD). The whole will basically model after a regular windmill.