Arduino Weather Station Setup

This document is explaining how to create an Weather Station for, Light, Temperature, Air Pressure, Humidity and Pollution.

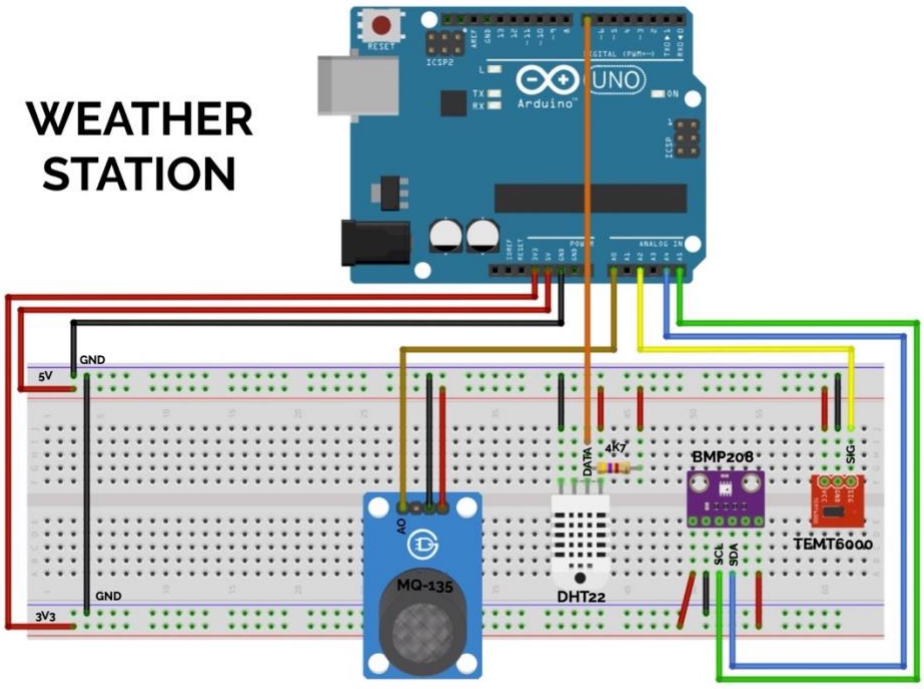
Arduino Weather Station Items

Following items are necessary to make the complete Weather Station:

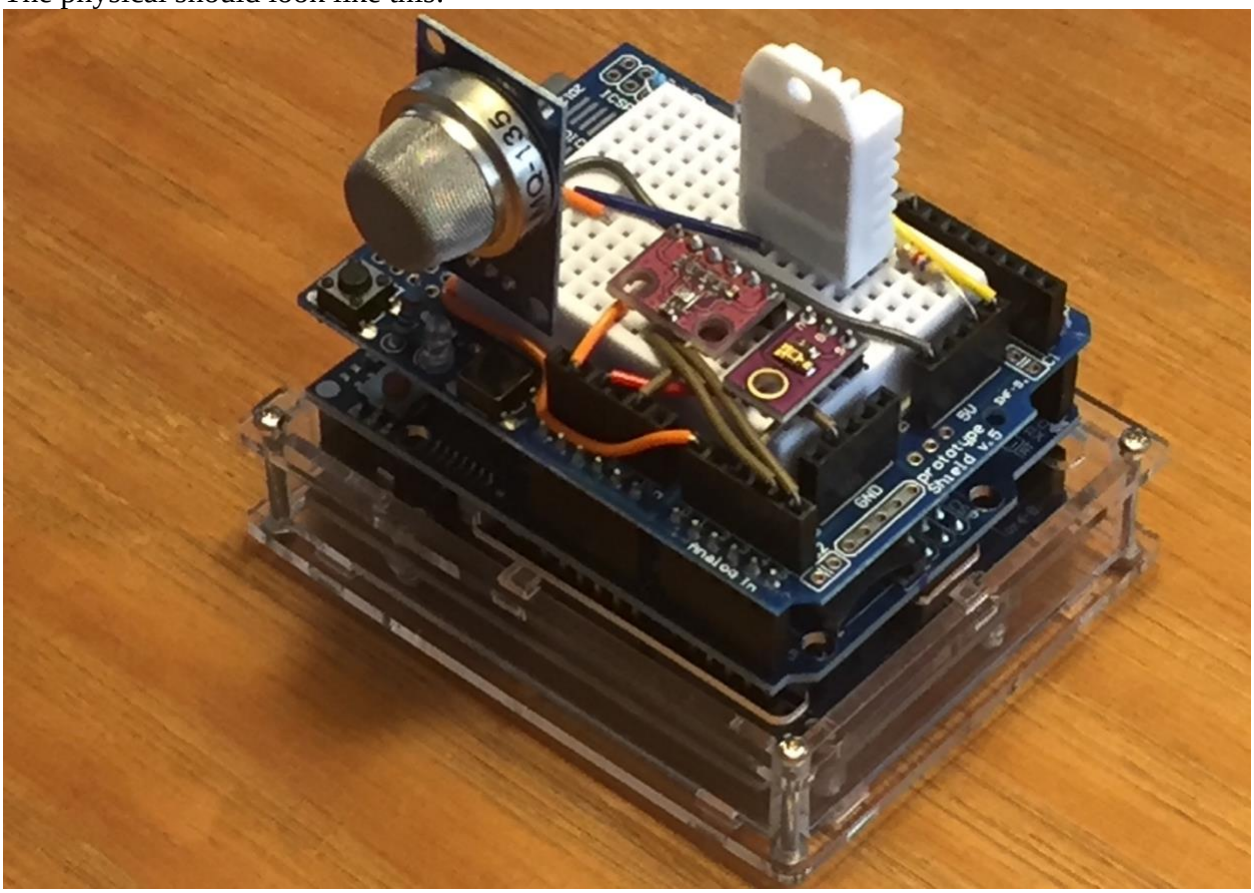
1. **Arduino UNO rev3**
2. **Ethernet-shield W5100**
3. **TEM6000 Light Sensor**
4. **BMP280 Temperature and Air Pressure Sensor**
5. **DHT22 Temperature and Humidity Sensor**
6. **MQ-135 Air Pollution Sensor**
7. **Resistor 4K7 ohm**
8. **Breadboard**

Arduino Weather Station Sensor Connections

Here is the complete interconnection schema for the Weather Station.
For more clear visualization of the connection the bread board is separated



Be aware that the Ethernet Shield should be on top of the Arduino board.
I also put a small breadboard shield on to of the Ethernet Shield to make the Weather Station smaller and more mobile.
The physical should look like this:



Arduino Weather Station Libraires

Before coding it is important that following libraires are installed

1. Adafruit_BMP280_Library
2. Adafruit_Unified_Sensor
3. MQUnifiedsensor

If downloaded manually you should put them in the Arduino “libraries” folder.

Arduino Weather Station Code

This code is also in the “WeatherStationEthernet.ino“ sketch.
It has been optimized for the 32K Storage space.
You should verify the your local IP address of you server and also the mac address of the Ethernet Shield. The rest of the code should work as it is.

WeatherStationEthernet.ino

```
/*
Weather
Station
*/
          /* The sketch is using 29720 bytes and 92% of Arduino Uno storage space  of max
          32356 bytes.
          /* The sketch is using 1533 bytes and 74% of Arduino Uno dynamic storage space fo
          max 2028 bytes.
          /* W5100 */
          #include <Ethernet.h>
          byte mac[] = { 0x00, 0xaa, 0xbb, 0xcc, 0xde, 0x02 };  // Ethernet MAC address
          EthernetClient client;
          char qnapserver[] = "192.168.0.4";                    // IP Adress of SQL server
          to dump data to

          /* TEMT6000*/
          int temt6000Pin = 2;                                  // Pin 2 connected to
          TEMT6000

          /* BMP280*/
          #include <Adafruit_BMP280.h>
          Adafruit_BMP280 bmp280;                               // I2C Interface

          /* DHT22 */
          #include <Adafruit_Sensor.h>                          // DHT Sensor library from
          Adafruit
          #include <DHT.h>
          #include <DHT_U.h>
          #define DHTTYPE      DHT22                            // Sensor type is DHT22
          #define DHTPIN 7                                      // Pin 7 coonected to DHT22
          DHT_Unified dht(DHTPIN, DHTTYPE);                     // configurando o Sensor
          DHT - pino e tipo

          /* MQ-135 */
```

```
#include <MQUnifiedsensor.h>
#define placa "Arduino UNO"
#define Voltage_Resolution 5
#define pin A0                                          // Analog input 0 of your
arduino
#define type "MQ-135"                                   // MQ135
#define ADC_Bit_Resolution 10                           // For arduino
UNO/MEGA/NANO
#define RatioMQ135CleanAir 3.6                          //RS / R0 = 3.6 ppm

MQUnifiedsensor MQ135(placa, Voltage_Resolution, ADC_Bit_Resolution, pin, type);

/* Measurement Interval*/
unsigned long interval = 3600000;                       //  1 hour Wait between
dumps <=> 3600000
                                                        //  10 min Wait between
dumps <=>  600000
void setup() {
  Serial.begin(9600);                                   // Serial Monitor 9600

/* W5100 */
  Ethernet.begin(mac);
  Serial.println("The Arduino Weather Station");
  Serial.println("-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-\n");
  Serial.print("IP Address        : ");
  Serial.println(Ethernet.localIP());
  Serial.print("Subnet Mask       : ");
  Serial.println(Ethernet.subnetMask());
  Serial.print("Default Gateway IP: ");
  Serial.println(Ethernet.gatewayIP());
  Serial.print("DNS Server IP     : ");
  Serial.println(Ethernet.dnsServerIP());

/* BMP280*/
  Serial.print(F("BMP280 Test: "));
  if (!bmp280.begin()) {
    Serial.println(F("Could not find the BMP280 sensor, check wiring!"));
    while (1);
  }
  Serial.println("Passed");

/* Default settings from datasheet. */
  bmp280.setSampling(Adafruit_BMP280::MODE_NORMAL,      // Operating Mode
```

```
                  Adafruit_BMP280::SAMPLING_X2,         // Temp. oversampling
                  Adafruit_BMP280::SAMPLING_X16,        // Pressure oversampling
                  Adafruit_BMP280::FILTER_X16,          // Filtering
                  Adafruit_BMP280::STANDBY_MS_500);     // Standby time

/* DHT22 */
  dht.begin();
  sensor_t dht22sensor;

/* MQ-135 */
  MQ135.setRegressionMethod(1); //_PPM =  a*ratio^b
  MQ135.setA(110.47);
  MQ135.setB(-2.862);
  MQ135.init();
  float calcR0 = 0;
  for(int i = 1; i<=10; i ++)
  {
    MQ135.update(); // Update data, the arduino will be read the voltage on the
analog pin
    calcR0 += MQ135.calibrate(RatioMQ135CleanAir);
    Serial.print(".");
  }
  MQ135.setR0(calcR0/10);
  Serial.println("-----------------------------------");
}

void loop() {

/* TEMT6000 */
    float lightreading = analogRead(temt6000Pin);     // Read Light level from
TEMT6000
    float lightvalue = (100*lightreading)/1023;
    Serial.print(F("TEMT6000 Light is:       "));
    Serial.print(lightvalue);
    Serial.println(" %");

/* BMP280 */
    Serial.print(F("BMP280 Temperature:      "));
    Serial.print(bmp280.readTemperature());          // Read Temperature from
BMP280
    Serial.println(" *C");
    Serial.print(F("BMP280 Air Pressure:     "));
```

```
    Serial.print(bmp280.readPressure()/100);          // Read Air Pressure from
BMP280 and print in hPa
    Serial.println(" hPa");
/* DHT22 */
    sensors_event_t dht22;                            // inicializa o evento da
Temperatura
    dht.temperature().getEvent(&dht22);
    Serial.print(F("DHT22 Temperature:       "));
    Serial.print(dht22.temperature);                  // Read Temperature from
DHT22
    Serial.println(" *C");
    dht.humidity().getEvent(&dht22);
    Serial.print(F("DHT22 Humidity:          "));
    Serial.print(dht22.relative_humidity);            // Print Temperature of
DHT22
    Serial.println(" %");
/* MQ-135 */
    Serial.print(F("MQ135 Air Quality:       "));
    MQ135.update();
    Serial.print(100*MQ135.readSensor());             // Read ppm values from
MQT135
    Serial.println(" ppm");
    Serial.println();
/* SQL Server */
  if (client.connect(qnapserver, 80)) {
    Serial.println("QNAP Database Connected:"); // Connection successfull to SQL
Server
    // Make a HTTP request:
    client.print( "GET /phpMyAdmin/addweatherdatatoqnapdb.php?");
    client.print("Microcontroller=");
    client.print( "ArduinoEthernet" );
    client.print("&&");
    client.print("Temt6000Light=");
    client.print(lightvalue);
    client.print("&&");
    client.print("Bmp280Temperature=");
    client.print(bmp280.readTemperature());
    client.print("&&");
    client.print("Bmp280Pressure=");
    client.print(bmp280.readPressure()/100);
    client.print("&&");
    dht.temperature().getEvent(&dht22);
    client.print("Dht22Temperature=");
```

```
    client.print(dht22.temperature);
    client.print("&&");
    dht.humidity().getEvent(&dht22);
    client.print("Dht22Humidity=");
    client.print(dht22.relative_humidity);
    client.print("&&");
    client.print("Mq135AirQuality=");
    client.print(100*MQ135.readSensor());
    client.println( " HTTP/1.1");
    client.print( "Host: " );
    client.println(qnapserver);
    client.println( "Connection: close" );
    client.println();
    client.println();
    client.stop();
    Serial.println("New SQL record created");
  }
  else {
    Serial.println("QNAP Database Connection Failed/n");          // Connection
failed to SQL Server
  }
  Serial.println("-----------------------------------");
  delay(interval);
}
```

Final Words

If you have followed all the steps and setup your QNAP MariaDB SQL Server Data Base, you should now be able to see the data in the serial monitor and also be able to send and see the data in the QNAP MariaDB SQL Server Data Base.