Arduino Weather Station Project

In this project we are going to make a Weather Station using a variety of sensors. The platform being used is an Arduino board. As we work through the project we will connect up the various sensors using I2C Bus protocol. In this process we create the software sketch that will run the weather station. We could also use a WiFi connection to send the data to the web server.

Overviews

I2C Communication Protocol

I2C (Inter-Integrated Circuit) is a serial bus interface connection protocol. It is also called as TWI (two-wire interface) since it uses only two wires for communication. Those two wires are SDA (serial data) and SCL (serial clock).

I2C is an acknowledgment-based communication protocol i.e. transmitter checks for an acknowledgment from the receiver after transmitting data to know whether data is received by the receiver successfully.

I2Cworks in two modes namely,

- Master mode
- Slave mode

SDA (serial data) wire is used for data exchange between master and slave devices. SCL (serial clock) is used for the synchronous clock in between master and slave devices.

The master device initiates communication with a slave device. The master device requires a slave device address to initiate a conversation with a slave device. The slave device responds to the master device when it is addressed by a master device.

Prerequisites

Before proceeding, make sure you complete the following prerequisites.

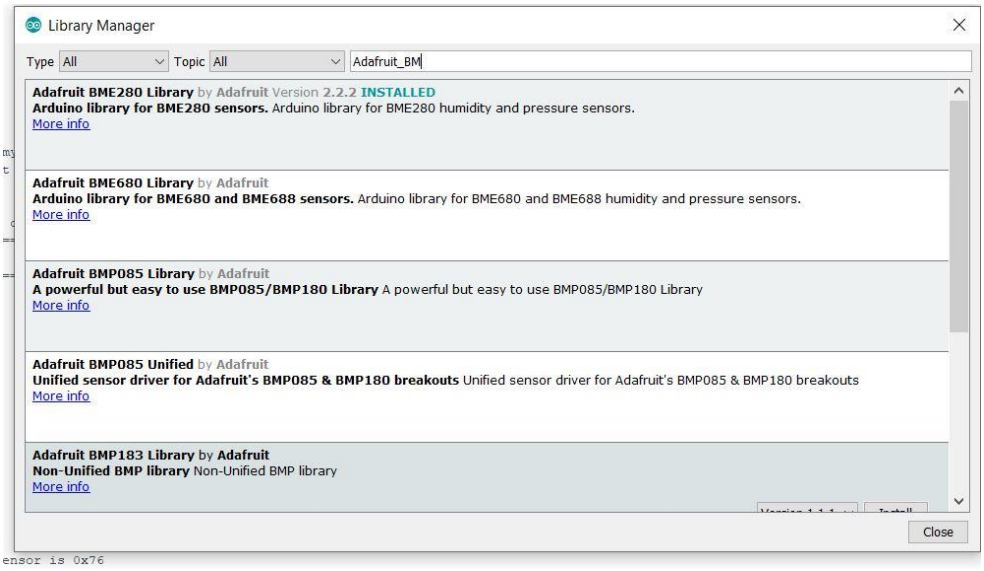
Arduino IDE

We'll program the Arduino UNO board using Arduino IDE

[Installing the Arduino IDE (Windows, Mac OS X, Linux)]

Adafruit_BME280 Library

You also need to install the Adafruit_BME280 , BH1750, Adafruit_Sensor. You can install this library in the Arduino IDE Library Manager. Just go to **Sketch** > **Include Library** > **Manage Libraries** and search for the library name as follows:



Parts Required

For this project you need the following parts:

- Arduino UNO Board
- NodeMCU

- GY-BMP280 sensor
- GY-30 Light intensity sensor
- HW-837 UV index sensor
- Power Supply (5V 2.5A)
- Jumper wires
- Breadboard

About Sensors(GY-BME280, GY-30 Light intensity)

GY-BMP280

The GY-BMP280 Barometer Sensor is a breakout board for Bosch BMP280 high-precision and low-power digital barometer. It can be used to measure temperature and atmospheric pressure accurately. It can be connected to a microcontroller with I2C.

GY-BMP280 Module GY-BMP280 Module Pinout

The **GY-BMP280** module operates from 3.3V so requires 3.3V power and must be driven with 3.3V logic levels. If needed to operate at 5V, it can be done using voltage regulator and level shifters as it doesn't contain one. It is typically recommended to operate it on 3.3V and maximum at 3.6VDC. The module GY-

BMP280 module simply supports both I²C and SPI interfaces and comes with default I²C address of 0x76. The Chip Select (CSB) and Serial Data Output (SDO) pins of the BMP280 are necessary only when SPI-based (four-wire) communication is applied. I2C is a two wire interface SDA SCK.

Leave pin 6 of the module (SDO) unconnected to set the I²C address to 0x76 – the on-board resistor pulls the SDO pin low setting the address to 0x76.

To change the I²C address to 0x77, connect pin 6 of the module (SDO) to Vcc which would typically be the 3.3V supply.

GY-30 BHI1750 Light Intensity

The BHI1750FVI is a digital ambient light intensity sensor with I2C interface and output directly readable in Lux. The sensor is designed to detect the light spectrum visible to the human eye with peak sensitivity at 560nm which is in the green spectrum. The full range spans approximately 400nm to 700nm.

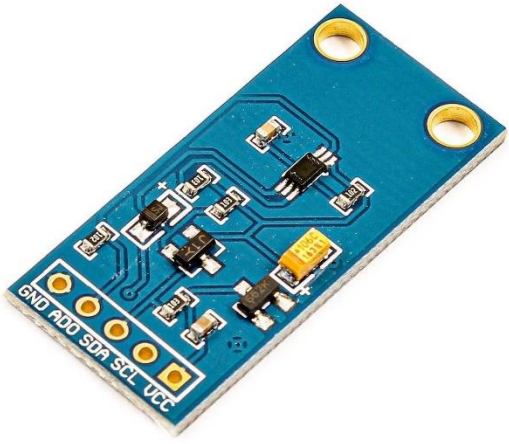


Photo by ElectroPeak

The sensor works well with most white visible light including the sun, incandescent, fluorescent, halogen and white LED lighting. It has good UV and IR rejection.

The GY-30 module incorporates a 3.3V regulator and can be powered from either 3.3V or 5V.

The I2C has logic level shifting circuitry to make the module compatible with both 3.3V and 5V logic. The SDA pin uses a MOSFET for level translation since it is a bi-directional signal. the SCL pin uses a diode / resistor to level shift since it is uni-directional.

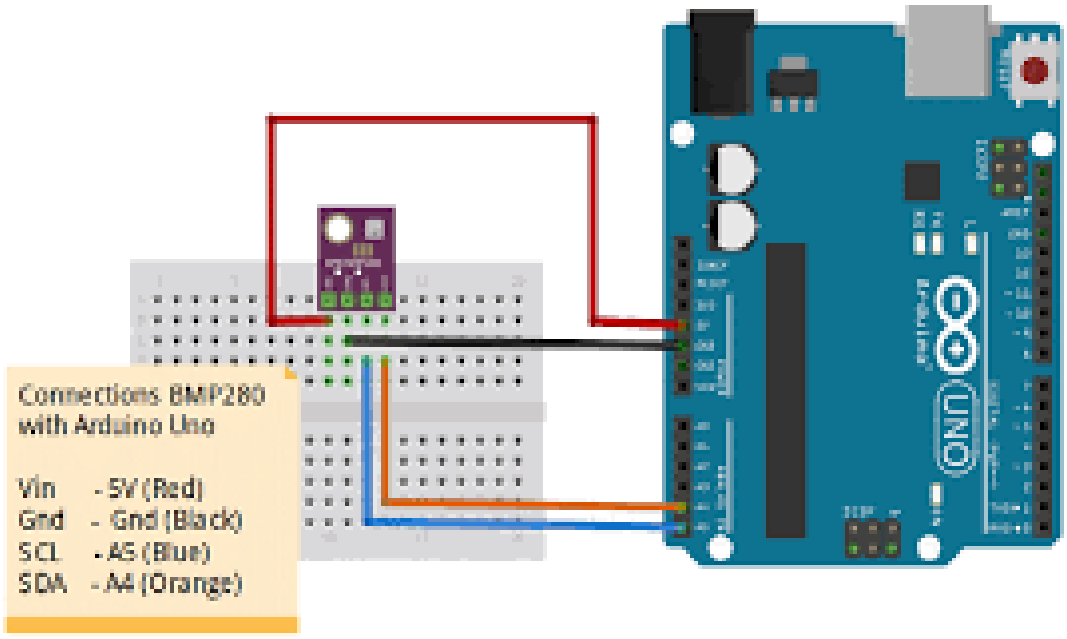
How to Make Weather Station

Step 1 : Connecting GY-BMP280 with Arduino UNO

Connecting the GY-BMP 280 module to an Arduino is very easy being that there are only four wires between them:

- VCC > Arduino VCC (3.3 V)
- GND > Arduino GND
- SCL > Arduino A5
- SDA > Arduino A4

Also see the hardware setup diagram shown below:

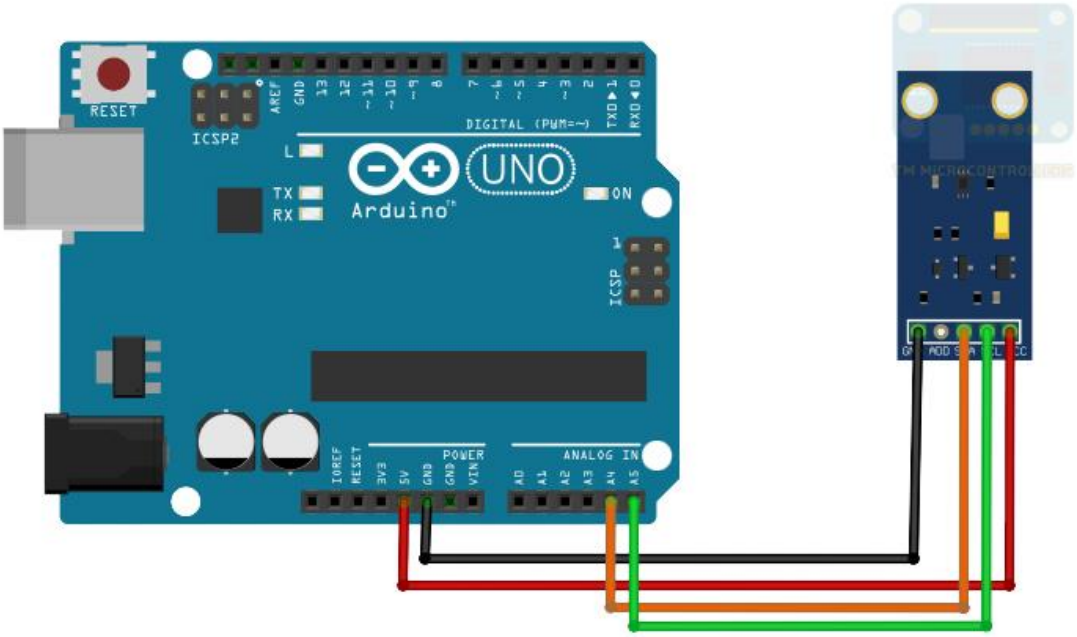


Step 2 : Connect GY-30 & HW-837 with Arduino UNO

Connection of GY-30 is also an eassy task that that there are only four wires between them:

- VCC > Arduino VCC (3.3 V)
- GND > Arduino GND
- SCL > Arduino A5
- SDA > Arduino A4

Also see the hardware setup diagram shown below:

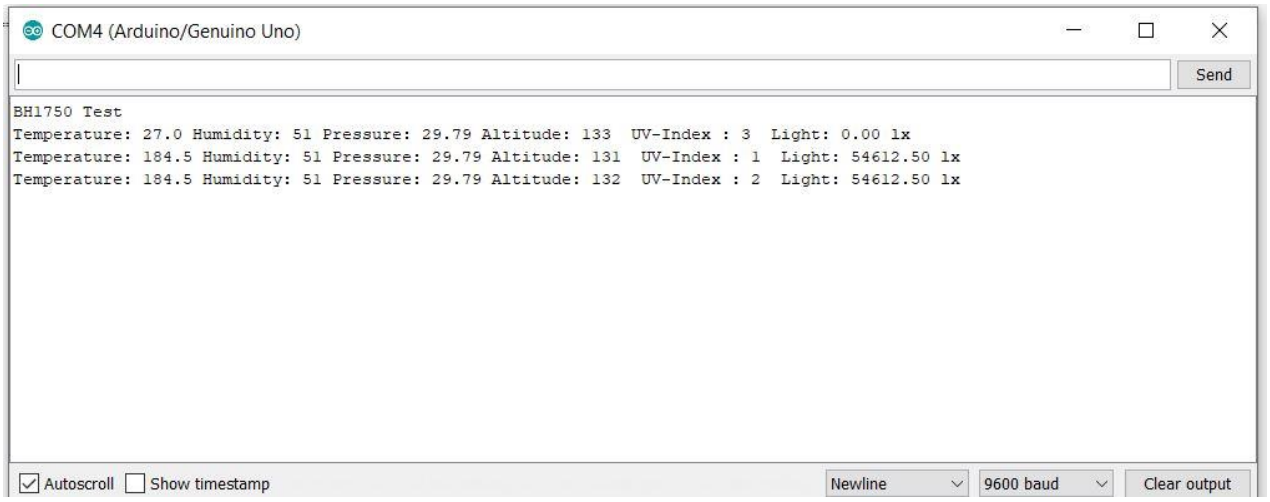

RESET
ICSP2
DIGITAL (PWM=~)
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN

Step 3 : Code

Weather_Station

```
#include <LiquidCrystal_I2C.h>
#include <BH1750.h>
#define I2C_address 0x23
BH1750 lightMeter;

float temperature;
float humidity;
float pressure;
float altitude;

float const ALTITUDE = 4.0;              // Altitude at my location in meters
float const SEA_LEVEL_PRESSURE = 1013.25;  // Pressure at sea level
Adafruit_BME280 bme; // I2C

LiquidCrystal_I2C lcd(0x27, 20, 4);   // I2C address, 20 char x 4 lines
//=========================================================================
//  Initialization
//=========================================================================
void setup(void) {

  Wire.begin();
  lcd.begin();
  Serial.begin(9600);
  lightMeter.begin();
  //Serial.println(F("BH1750 Test"));
  lcd.clear();          // Clear display
  lcd.backlight();      // Make sure backlight is on
  lcd.print("Reading sensor");
  //Serial.println(" Reading sensor");

  bool status;

  // default settings
  status = bme.begin(0x76);  // The I2C address of the sensor is 0x76
 /* if (!status) {             // Loop if sensor not found
    lcd.clear();
    lcd.print("Error. Check");
    lcd.setCursor(0, 1);
    lcd.print("connections");
    while (1);
  }*/
  // Print non changing info on LCD once
```

Step 4 : Console Output



Step 5 : Web Output