



# User Manual & Sensor Tutorials

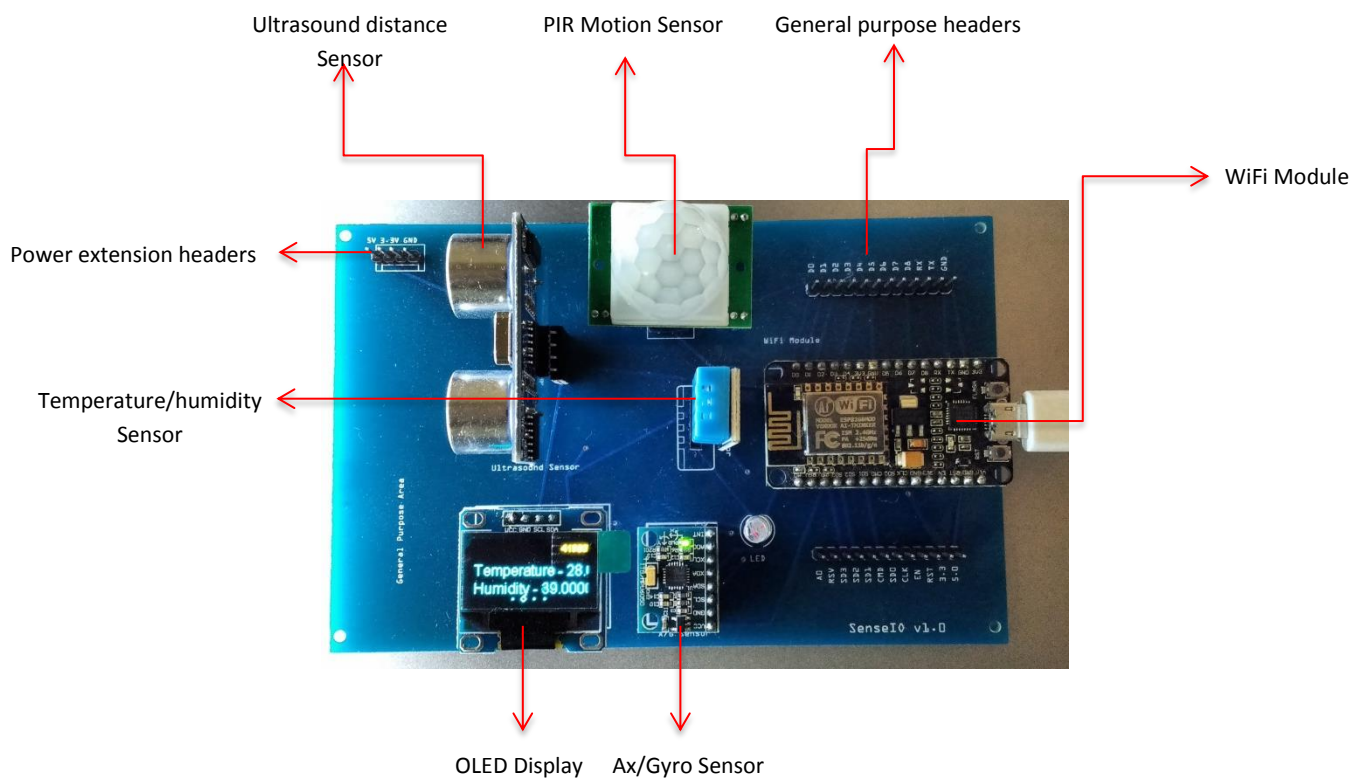
## Contents

1. Drivers & SDK Installation
2. Blink LED test
3. Sensor tests
  - a. Temperature/Humidity Sensor
  - b. PIR Motion Sensor
  - c. Ultrasound distance Sensor
  - d. Accelerometer/Gyroscope Sensor
4. OLED display test
5. Simple Web Server test

Version 1.0

## SenseIO™ Platform consists of following components

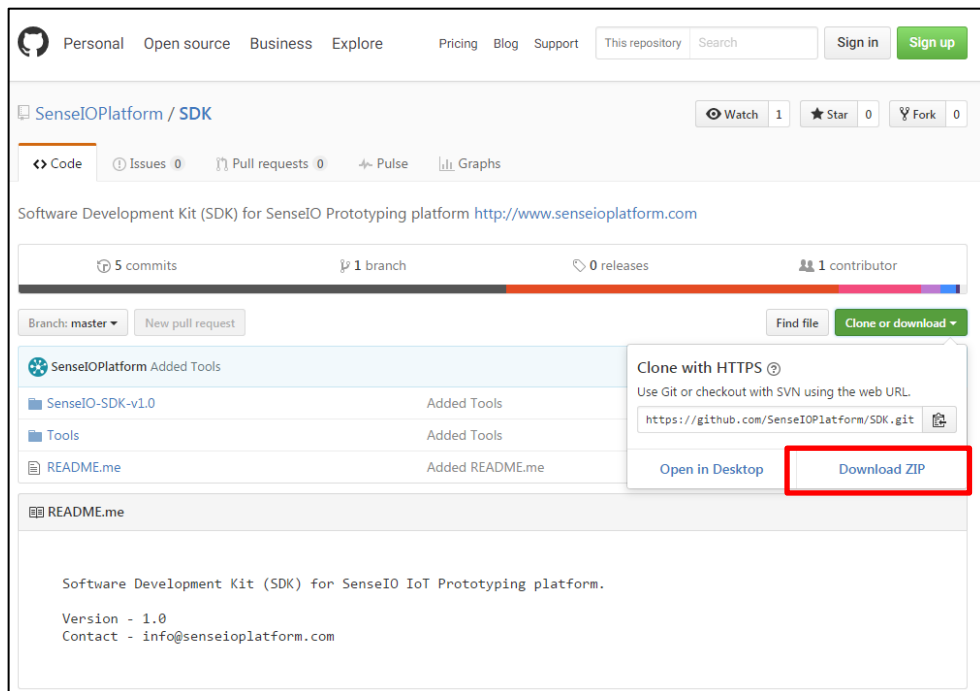
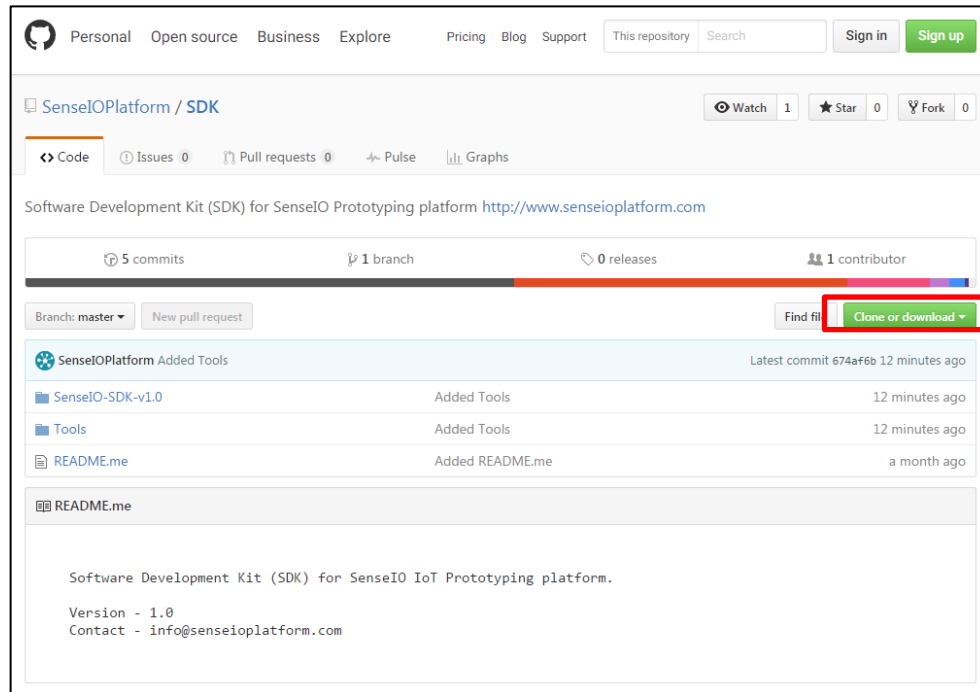
1. WiFi module
2. DHT11 temperature sensor
3. PIR Motion sensor
4. Accelerometer/Gyroscope sensor
5. Ultrasound distance sensor
6. OLED display
7. General purpose headers
8. Power extension header



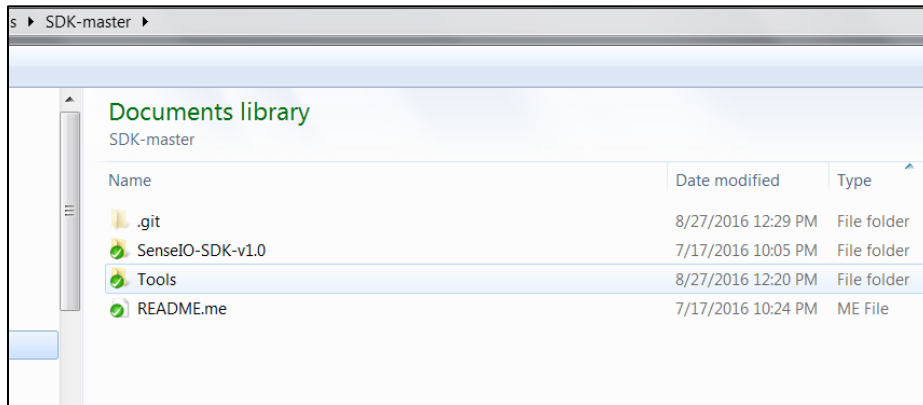
## SDK & Drivers installation

1. Download the SDK from below path:

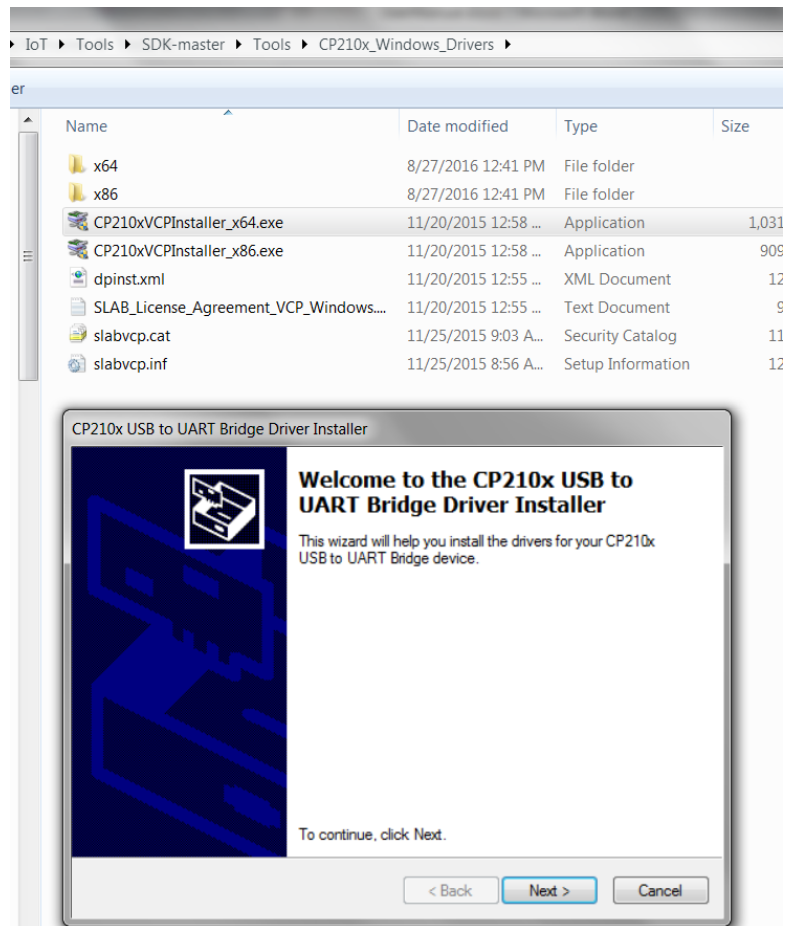
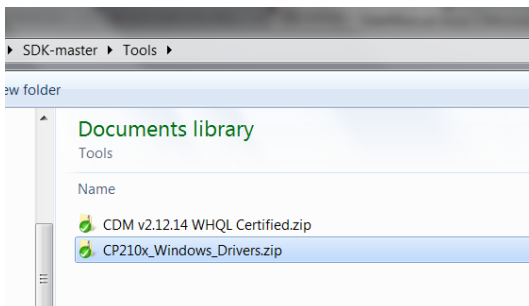
<https://github.com/SenseIOPlatform/SDK>



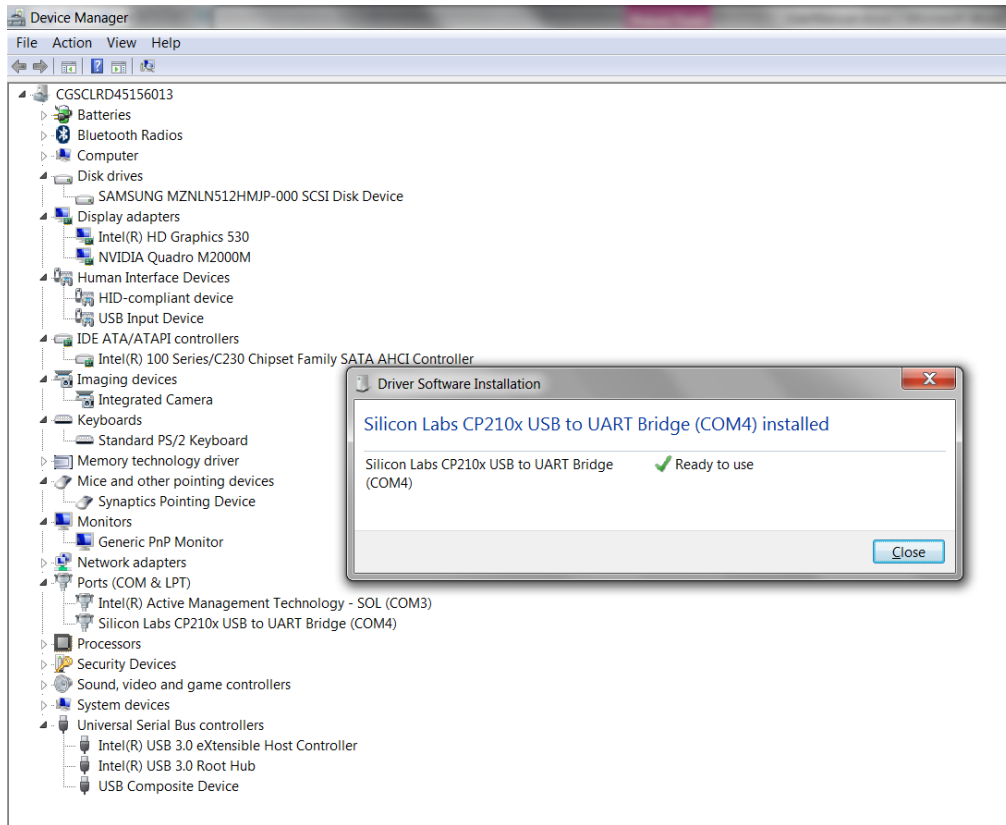
2. Unzip SDK zip file which was downloaded



3. Connect the SenseIO platform to PC using USB cable.
4. Navigate inside SDK-master/Tools folder & unzip the CP210x\_Windows\_Drivers.zip. Install the CP210xVCPInstaller\_x64.exe(if PC is 64 bit) or CP210xVCPInstaller\_x86.exe(if PC is 32 bit).

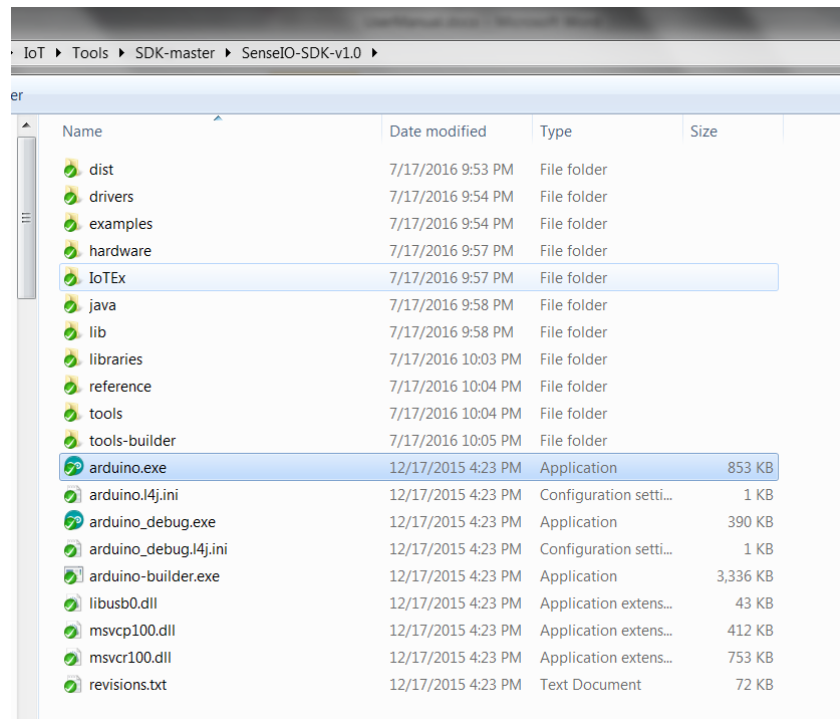


5. A pop will be shown on successful installation of the driver. Please note the COM port at which the SenseIO platform is detected (COM4, in below case).

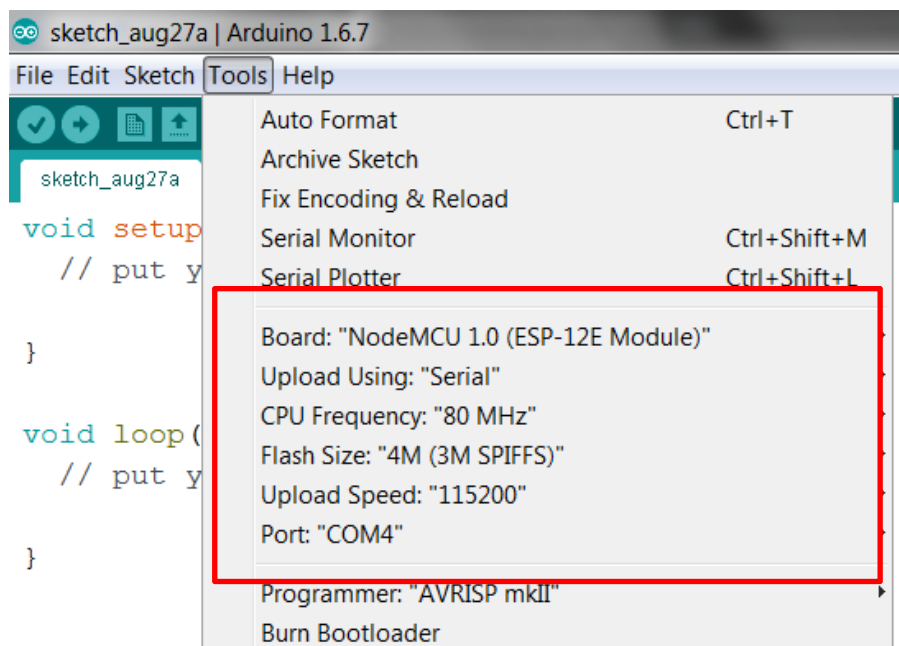


## Basic Compile & Run test

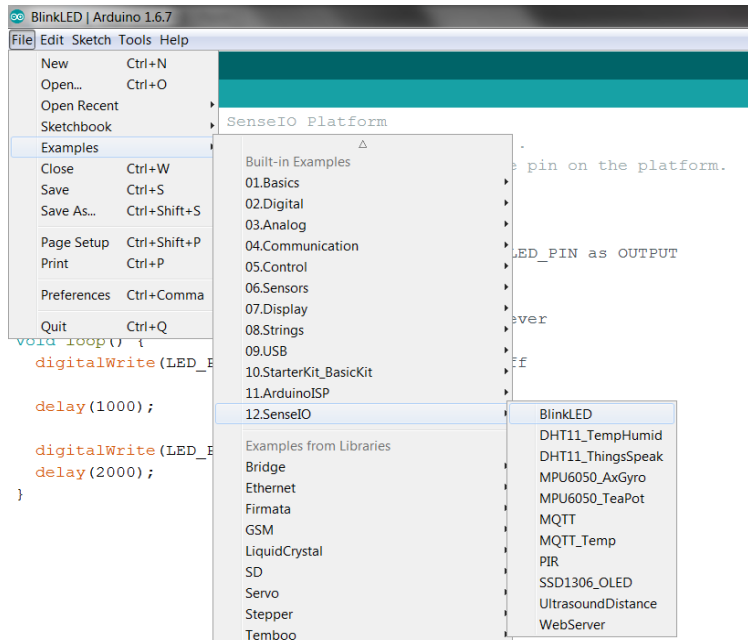
1. Open Arduino IDE application.

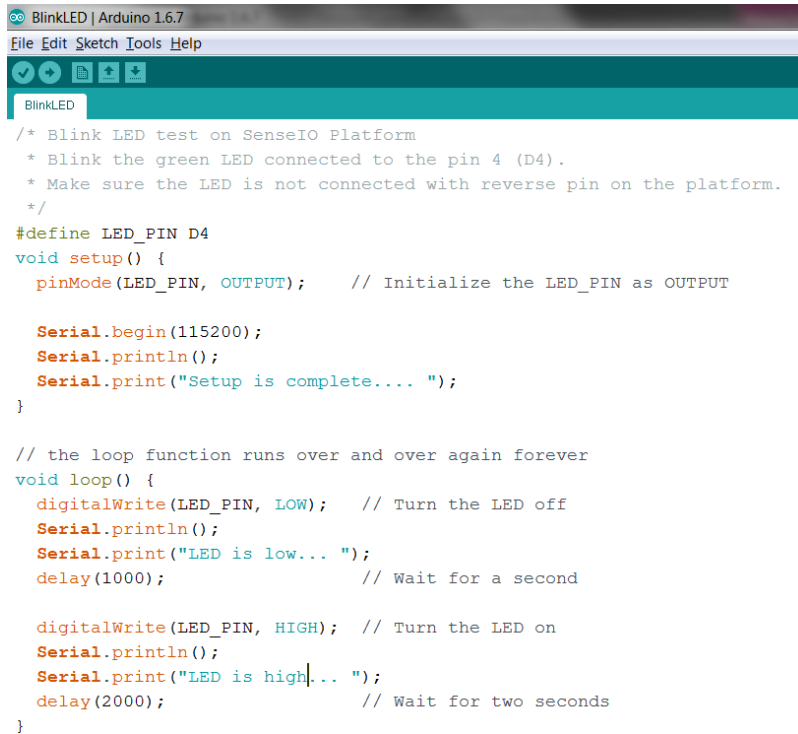


2. Select Board as **"NodeMCU 1.0(ESP-12E Module)";** make sure other configuration parameters are set as below. COM Port will be the PC's port at which device is currently connected; it may be different than it is shown below.



### 3. Navigate to **File->Examples->SenseIO->BlinkLED**





```

BlinkLED | Arduino 1.6.7
File Edit Sketch Tools Help

BlinkLED

/* Blink LED test on SenseIO Platform
 * Blink the green LED connected to the pin 4 (D4).
 * Make sure the LED is not connected with reverse pin on the platform.
 */

#define LED_PIN D4
void setup() {
  pinMode(LED_PIN, OUTPUT);    // Initialize the LED_PIN as OUTPUT

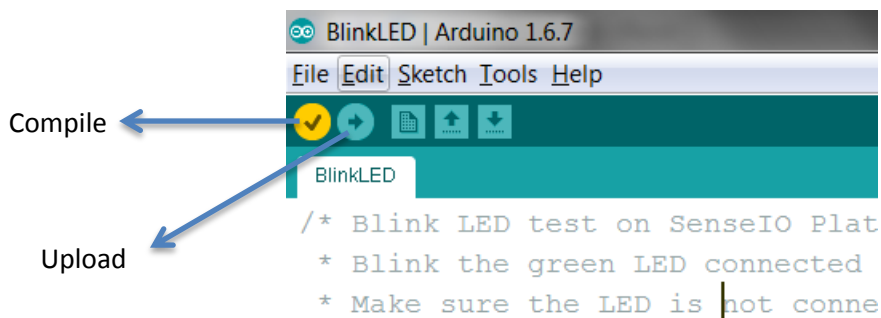
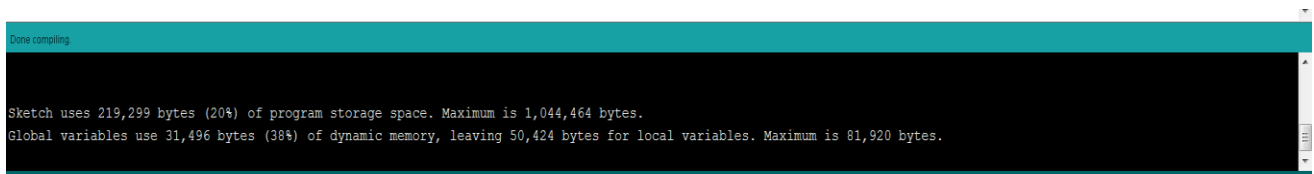
  Serial.begin(115200);
  Serial.println();
  Serial.print("Setup is complete.... ");
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_PIN, LOW);  // Turn the LED off
  Serial.println();
  Serial.print("LED is low... ");
  delay(1000);                 // Wait for a second

  digitalWrite(LED_PIN, HIGH); // Turn the LED on
  Serial.println();
  Serial.print("LED is high... ");
  delay(2000);                 // Wait for two seconds
}

```

4. Press on compile button (Program will only be compiled & not uploaded on platform). Press on upload button to upload the program on platform.

Done compiling

Sketch uses 219,299 bytes (20%) of program storage space. Maximum is 1,044,464 bytes.  
Global variables use 31,496 bytes (38%) of dynamic memory, leaving 50,424 bytes for local variables. Maximum is 81,920 bytes.

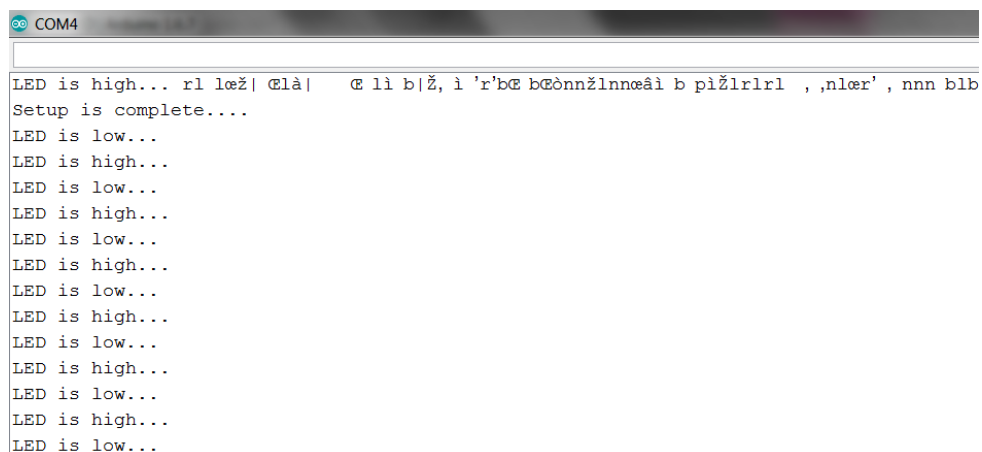
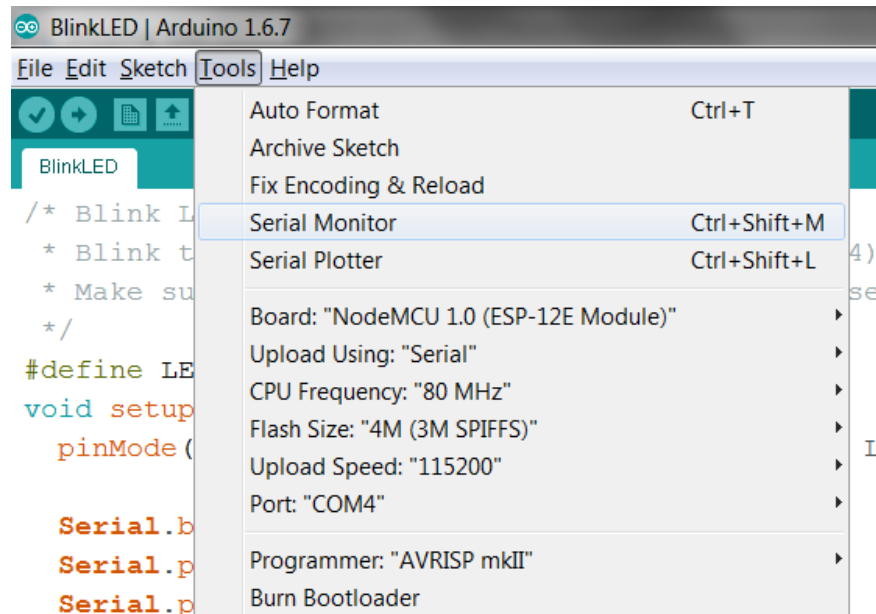
5. Observe above compile message, when there are no errors in the code.



- On upload button, binary will be flashed on the platform. Check the status of upload as progress bar.

```
Uploading...
Sketch uses 219,299 bytes (20%) of program storage space. Maximum is 1,044,464 bytes.
Global variables use 31,496 bytes (38%) of dynamic memory, leaving 50,424 bytes for local variables. Maximum is 81,920 bytes.
Uploading 223440 bytes from C:\p\builda54e62155409d6289c78de5e587b37db.tmp/BlinkLED.ino.bin to flash at 0x00000000
.....
```

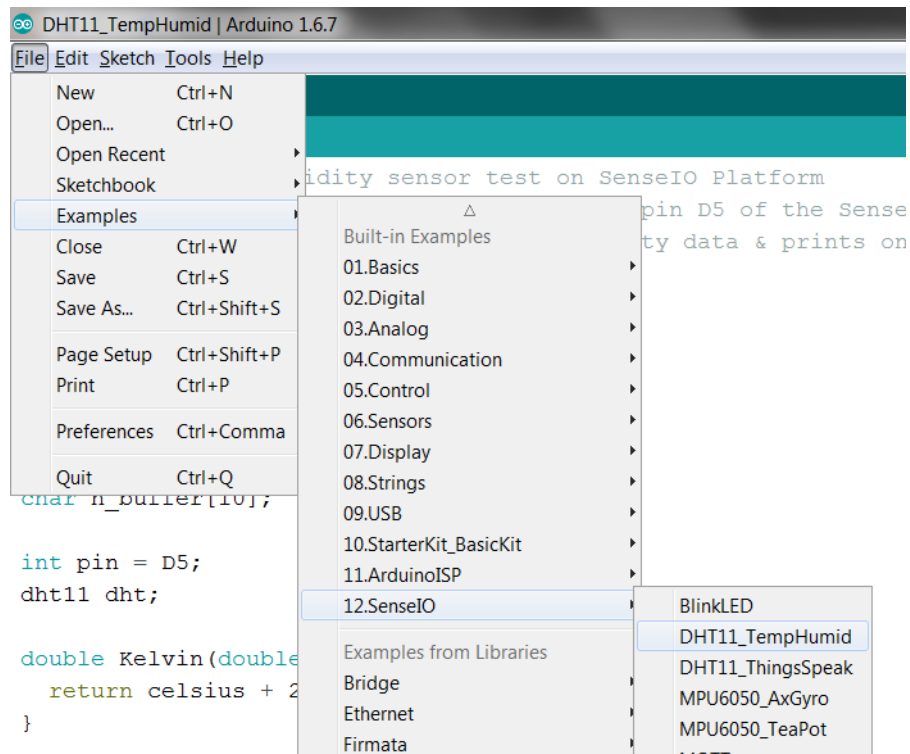
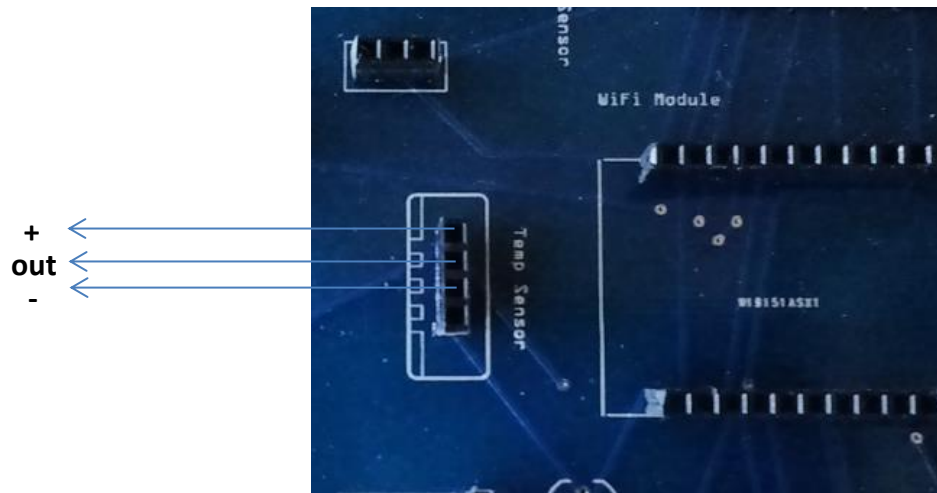
- Open the Serial Monitor to view the messages.



# Sensor Tutorials

## 1. Temperature/Humidity Sensor

- a. Test file: **Files->Examples->SenseIO->DHT11\_TempHumid**

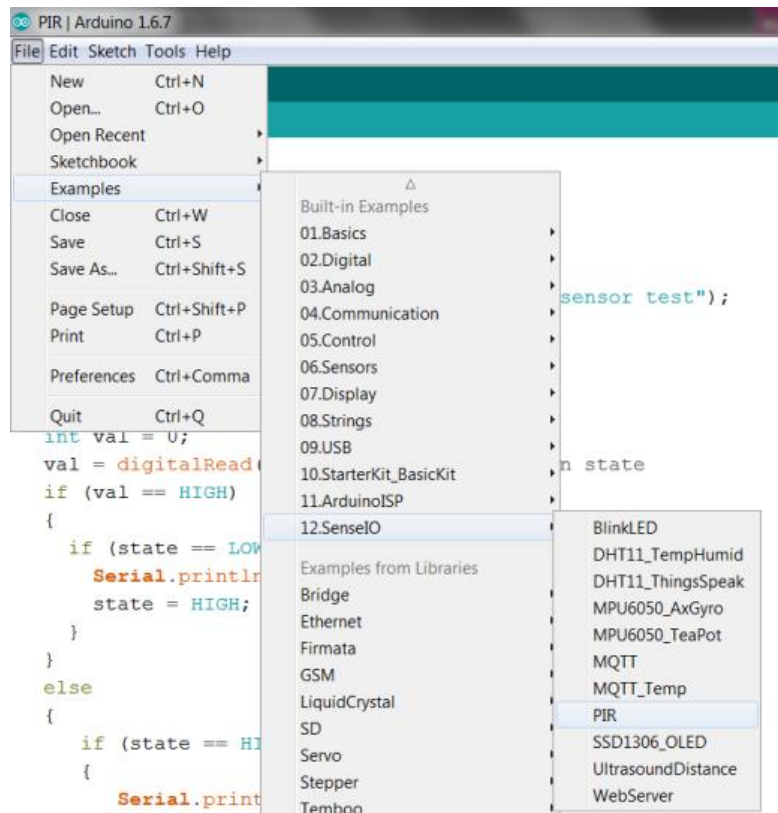


```
COM4
rl lœž| Elà|  E lì b|ž, ì 'r'bE bEònnžlñnœâì b pEžlrlrlpònà , l Eœ

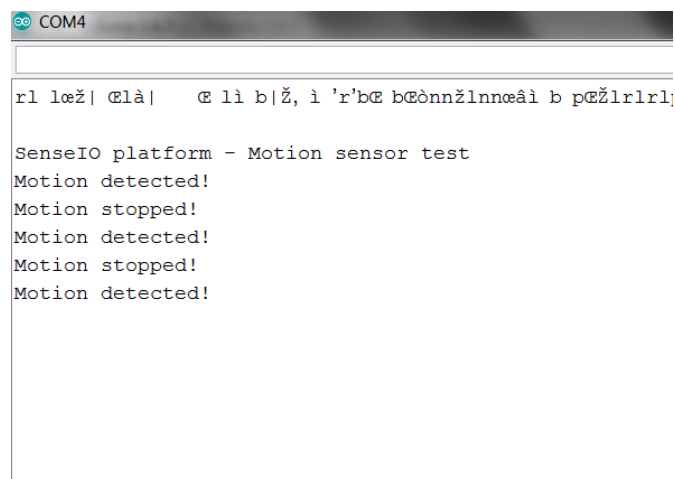
SenseIO platform - Temp/Humidity sensor test
temperature:28.00 humidity:47.00
temperature:28.00 humidity:47.00
temperature:28.00 humidity:49.00
temperature:28.00 humidity:47.00
temperature:28.00 humidity:47.00
```

## 2. PIR Motion Sensor:

- Test file: **Files->Examples->SenseIO->PIR**
- Motion sensor will detect the human motion, the sensor output remains at detect level for few seconds before going back to not-detected state.

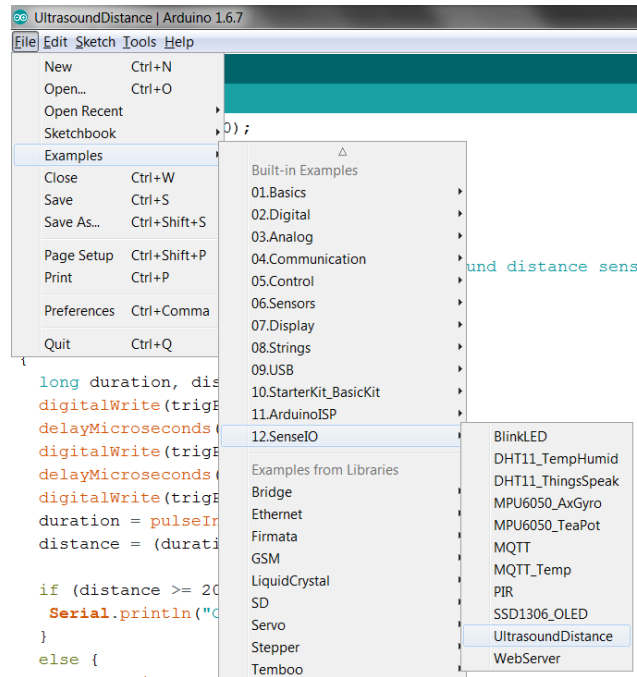


- Make a motion in front of motion sensor to test this sensor.

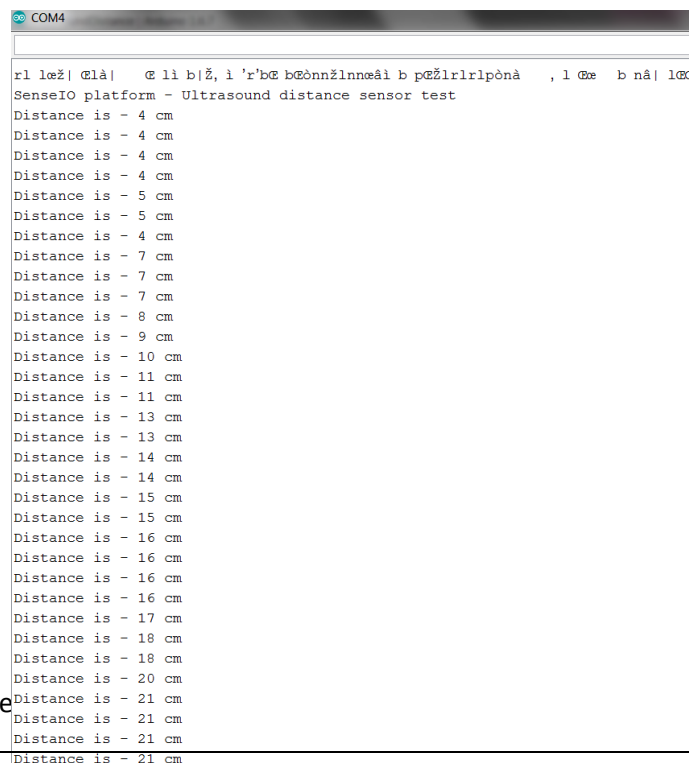


### 3. Ultrasound Distance Sensor

- Test file: **Files->Examples->SenselIO->UltrasoundDistance**
- Ultrasound sensor detects the distance at which the obstacle is present in front of it.

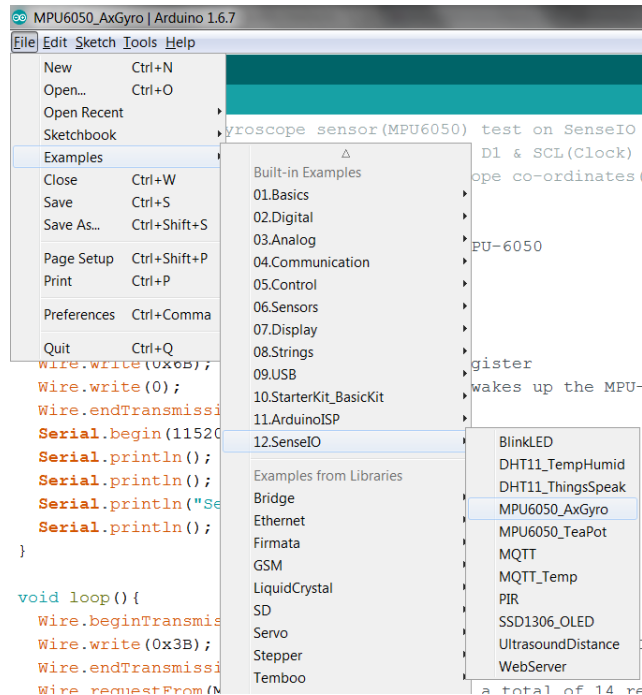


- Distance detection test can be performed by varying the distance between an obstacle & the sensor.



#### 4. Accelerometer/Gyroscope sensor test

- Test file: **Files->Examples->SenseIO->MPU6050\_AxGyro**
- This sensor detects acceleration & gyroscope property in 3 directions.



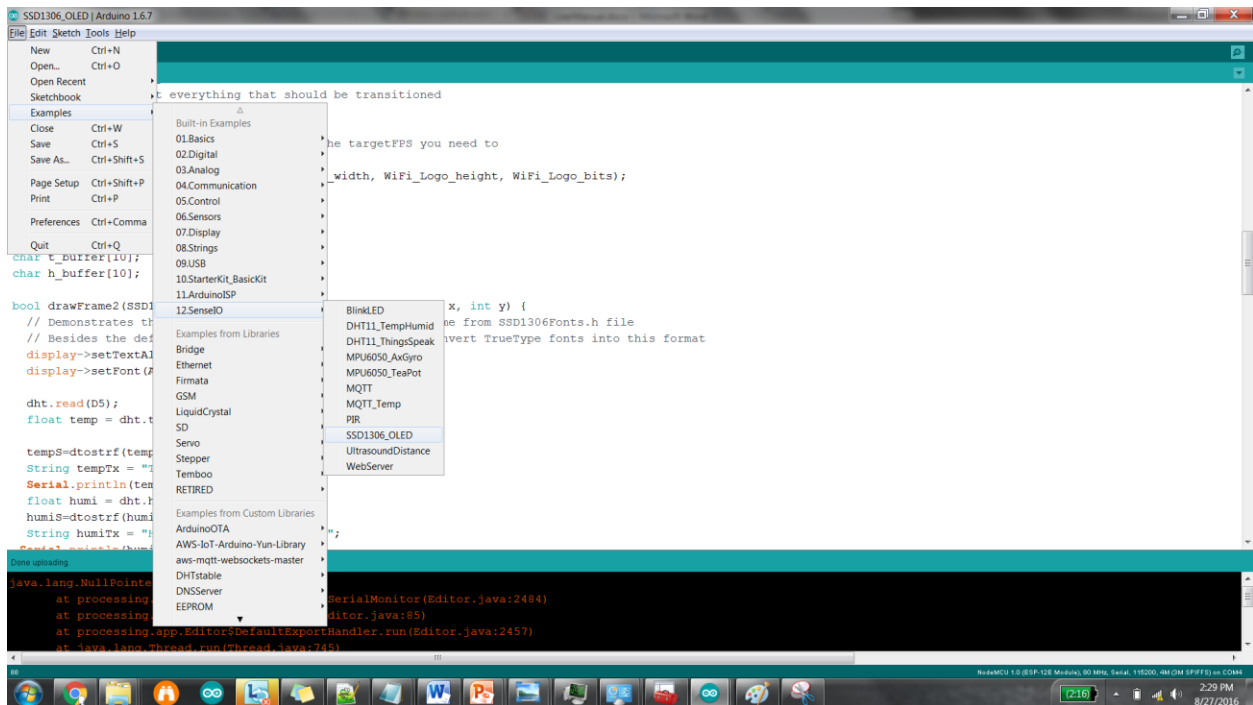
- Change in values of Ax & Gyro can be observed by moving the platform in different axis (x, y & z).

The screenshot shows the Arduino Serial Monitor window. The title bar indicates 'COM4'. The output text is as follows:

```
rl lœž| @lâ|  @ li b|ž, i 'r'b@ bœönnžlñnœâi b pœžlrlrlpônâ , l @œ b nâ| lœœžžbœöi  
SenseIO platform - Accelerometer/Gyroscope sensor test  
AcX = 4648 | AcY = -1204 | AcZ = 14352 | Tmp = 30.69  
| GyX = -4117 | GyY = -210 | GyZ = -498  
AcX = 4696 | AcY = -1040 | AcZ = 14328 | Tmp = 30.74  
| GyX = -4119 | GyY = -201 | GyZ = -489  
AcX = 4672 | AcY = -1204 | AcZ = 14268 | Tmp = 30.65  
| GyX = -4134 | GyY = -185 | GyZ = -474  
AcX = 4820 | AcY = -1136 | AcZ = 14308 | Tmp = 30.69  
| GyX = -4106 | GyY = -221 | GyZ = -513  
AcX = 4664 | AcY = -1092 | AcZ = 14304 | Tmp = 30.69  
| GyX = -4116 | GyY = -206 | GyZ = -491  
AcX = 4688 | AcY = -1196 | AcZ = 14324 | Tmp = 30.69  
| GyX = -4119 | GyY = -203 | GyZ = -482  
AcX = 4748 | AcY = -988 | AcZ = 14368 | Tmp = 30.69  
| GyX = -4111 | GyY = -214 | GyZ = -482  
AcX = 4700 | AcY = -1096 | AcZ = 14320 | Tmp = 30.74  
| GyX = -4093 | GyY = -252 | GyZ = -506
```

## 5. OLED display test

### a. Test file: Files->Examples->SenselIO->SSD1306\_OLED



## 6. Simple Web Server

### a. Test file: Files->Examples->SenselIO->WebServer

- b. This simple webserver controls the provides an url through which the LED connected to SenselIO can be controlled. LED can be controlled from a browser on a PC connected to the same network as SenselIO platform.

### c. URL will be as follows

`<server_ip_address>/gpio/1 → to turn LED on`

`<server_ip_address>/gpio/0 → to turn LED off`

```

WebServer | Arduino 1.6.7
File Edit Sketch Tools Help

WebServer

/*
 * A Simple WebServer example on SenseIO Platform
 * The server will set a pin depending on the request
 * http://server_ip/gpio/0 will set the D4 low,
 * http://server_ip/gpio/1 will set the D4 high
 * server_ip is the IP address of the SenseIO platform, will
 * printed to Serial when the module is connected.
 */

#include <ESP8266WiFi.h>

const char* ssid = "wifi ssid";
const char* password = "wifi_password";

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

```

Use your WiFi SSID & password

```

COM4

r1 læž| Èlà|   È lì b|Ž, ì 'r'bÈ bÈðnnŽlñnœâ

Connecting to AirtelAp
.....
WiFi connected
Server started
192.168.43.244
new client
GET /gpio/1 HTTP/1.1
Client disonnected

```

- d. LED on the platform will be turned ON/OFF based on the URL; HTTP response from the platform will be displayed on the webpage.

```

192.168.43.244/gpio/ x

192.168.43.244/gpio/1

Apps Embedded Softw Embedded Softw CW 12 must-have And New T

GPIO is now high

```



