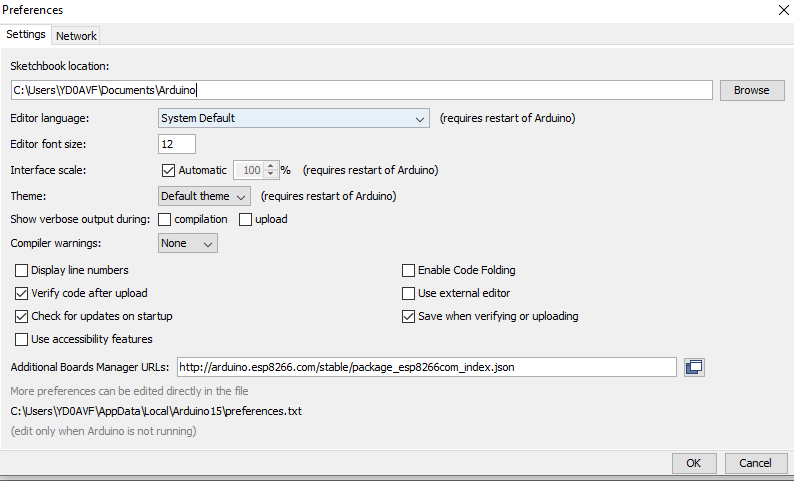
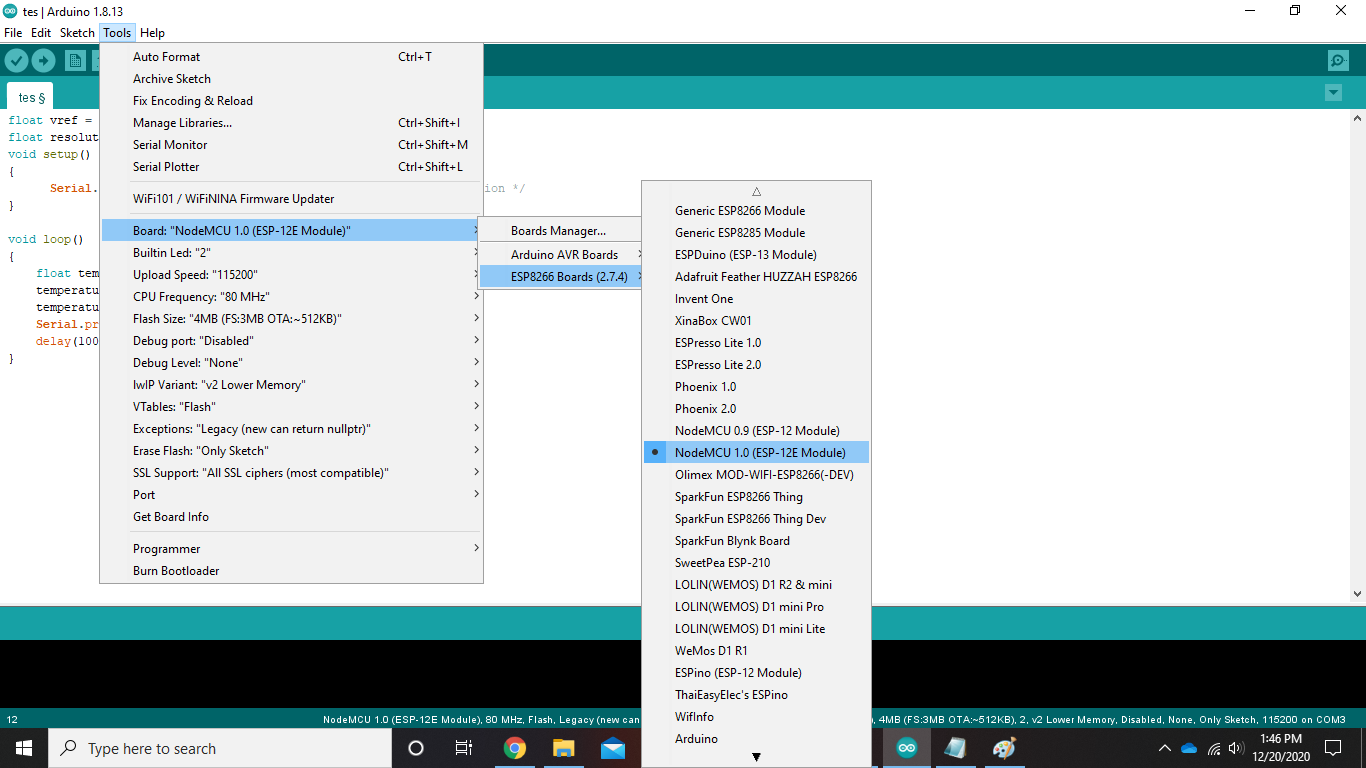


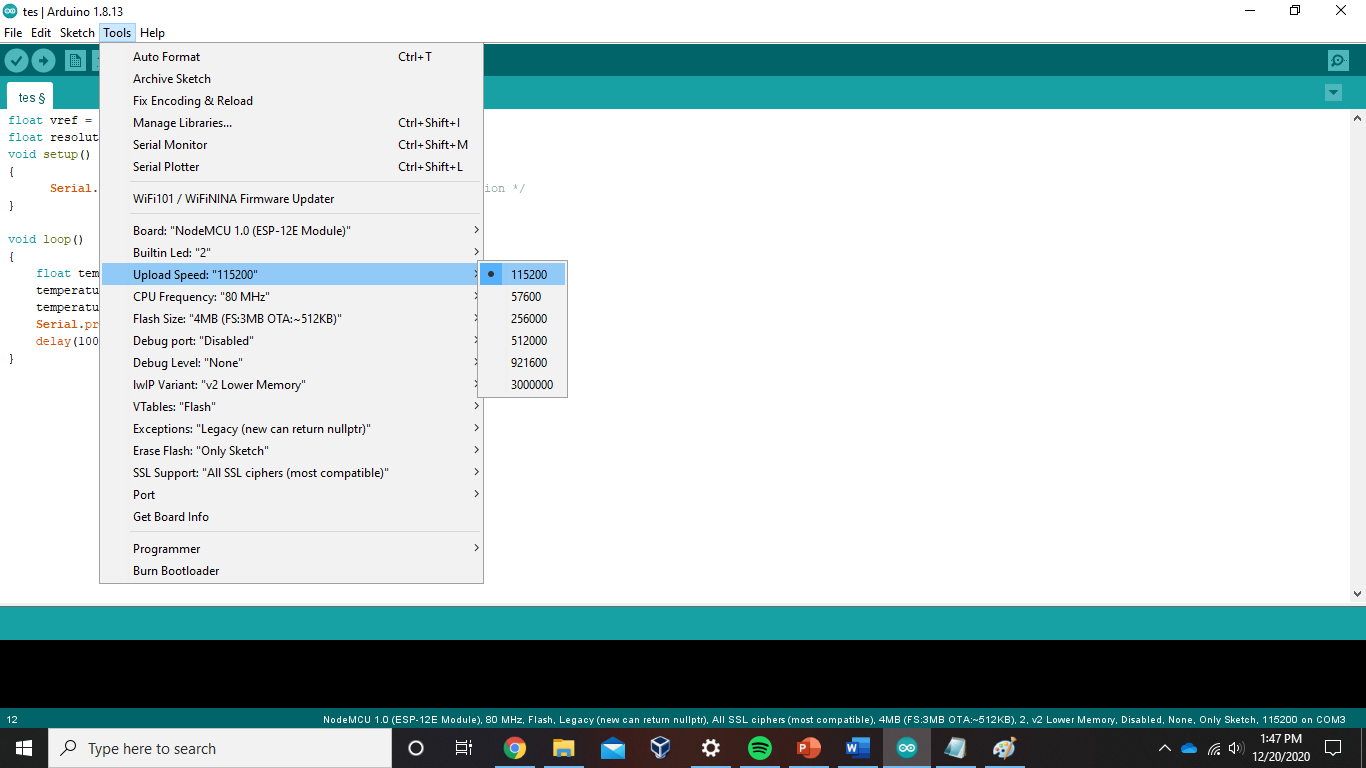
1. **Setting Arduino**
2. **Setting Preferences (File – Preferences)**



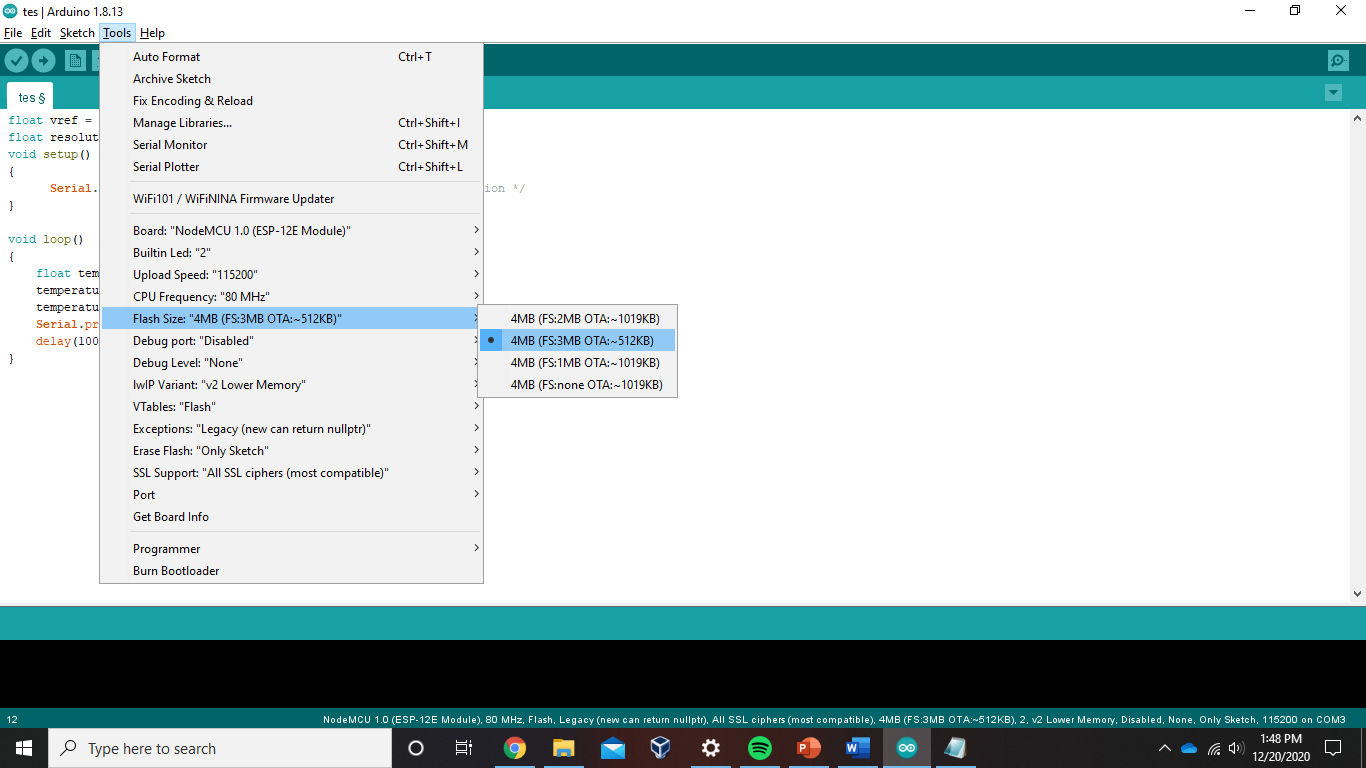
1. **Setting Board (Tools – Board – ESP8266 Boards – NodeMCU 1.0)**



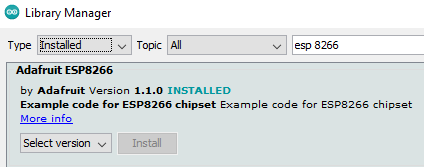
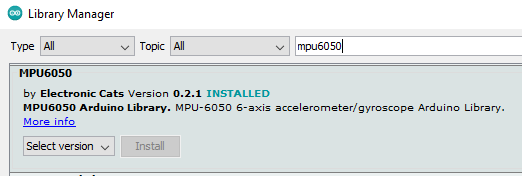
1. **Setting Upload Speed (Tools – Upload Speed – 115200)**

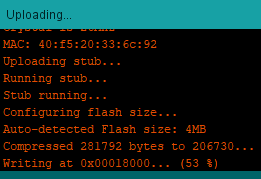


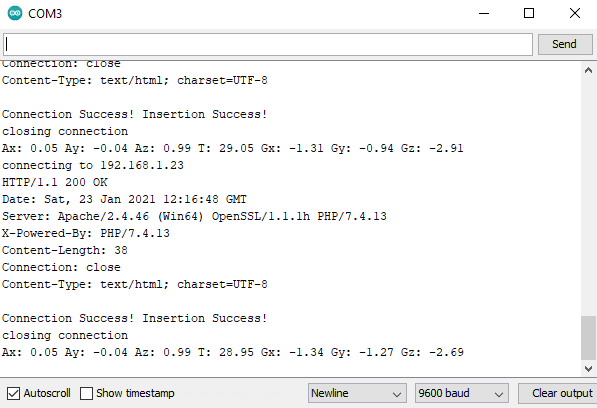
1. **Setting Flash Size (Tools – Flash Size – 4MB FS:3MB OTA:~512KB)**

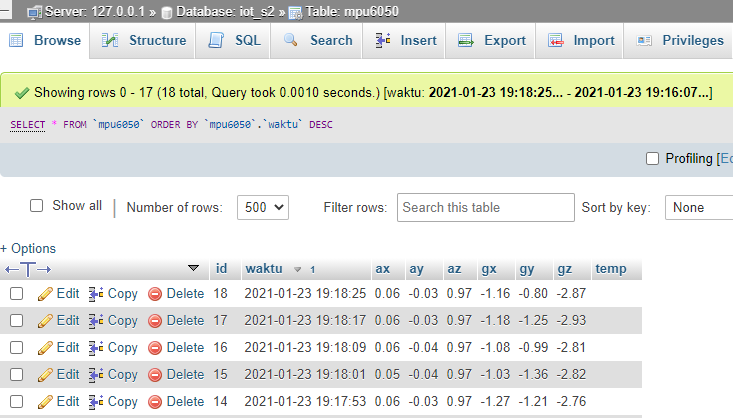


1. **Setting Libraries (Tools – Manage Libraries…)**

* **Install Library Board of Adafruit ESP8266**
* **Install Library of MPU6050 Sensor**

1. **Upload and Process Scripts**
2. **Data Output**



1. **Data Output Storing and Result on MySQL**

**Source:**

[**https://www.electronicwings.com/nodemcu/mpu6050-interfacing-with-nodemcu**](https://www.electronicwings.com/nodemcu/mpu6050-interfacing-with-nodemcu)

Wire.begin(sda, scl);

MPU6050\_Init();

}

void loop() {

double Ax, Ay, Az, T, Gx, Gy, Gz;

Read\_RawValue(MPU6050SlaveAddress, MPU6050\_REGISTER\_ACCEL\_XOUT\_H);

//divide each with their sensitivity scale factor

Ax = (double)AccelX/AccelScaleFactor;

Ay = (double)AccelY/AccelScaleFactor;

Az = (double)AccelZ/AccelScaleFactor;

T = (double)Temperature/340+36.53; //temperature formula

Gx = (double)GyroX/GyroScaleFactor;

Gy = (double)GyroY/GyroScaleFactor;

Gz = (double)GyroZ/GyroScaleFactor;

Serial.print("Ax: "); Serial.print(Ax);

Serial.print(" Ay: "); Serial.print(Ay);

Serial.print(" Az: "); Serial.print(Az);

Serial.print(" T: "); Serial.print(T);

Serial.print(" Gx: "); Serial.print(Gx);

Serial.print(" Gy: "); Serial.print(Gy);

Serial.print(" Gz: "); Serial.println(Gz);

delay(3000);

Serial.print("connecting to ");

Serial.println(host);

// Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}

// This will send the request to the server

client.print(String("GET http://192.168.1.23/iot-s2/mpu6050.php?") +

("ax=") + Ax +

("&ay=") + Ay +

("&az=") + Az +

("&gx=") + Gx +

("&gy=") + Gy +

("&gz=") + Gz +

" HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

unsigned long timeout = millis();

while (client.available() == 0) {

if (millis() - timeout > 1000) {

Serial.println(">>> Client Timeout !");

client.stop();

return;

}

}

C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_USER\_CTRL, 0x00);

}

//https://www.electronicwings.com/nodemcu/mpu6050-interfacing-with-nodemcu

#include <Wire.h>

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

const char\* ssid = "Azzam";

const char\* password = "azzam53a";

const char\* host = "192.168.1.23";

WiFiClient client;

// MPU6050 Slave Device Address

const uint8\_t MPU6050SlaveAddress = 0x68;

// Select SDA and SCL pins for I2C communication

const uint8\_t scl = D1;

const uint8\_t sda = D2;

// sensitivity scale factor respective to full scale setting provided in datasheet

const uint16\_t AccelScaleFactor = 16384;

const uint16\_t GyroScaleFactor = 131;

// MPU6050 few configuration register addresses

const uint8\_t MPU6050\_REGISTER\_SMPLRT\_DIV = 0x19;

const uint8\_t MPU6050\_REGISTER\_USER\_CTRL = 0x6A;

const uint8\_t MPU6050\_REGISTER\_PWR\_MGMT\_1 = 0x6B;

const uint8\_t MPU6050\_REGISTER\_PWR\_MGMT\_2 = 0x6C;

const uint8\_t MPU6050\_REGISTER\_CONFIG = 0x1A;

const uint8\_t MPU6050\_REGISTER\_GYRO\_CONFIG = 0x1B;

const uint8\_t MPU6050\_REGISTER\_ACCEL\_CONFIG = 0x1C;

const uint8\_t MPU6050\_REGISTER\_FIFO\_EN = 0x23;

const uint8\_t MPU6050\_REGISTER\_INT\_ENABLE = 0x38;

const uint8\_t MPU6050\_REGISTER\_ACCEL\_XOUT\_H = 0x3B;

const uint8\_t MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET = 0x68;

int16\_t AccelX, AccelY, AccelZ, Temperature, GyroX, GyroY, GyroZ;

void setup() {

Serial.begin(9600);

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

// Read all the lines of the reply from server and print them to Serial

while(client.available()) {

String line = client.readStringUntil('\r');

Serial.print(line);

}

Serial.println();

Serial.println("closing connection");

}

void I2C\_Write(uint8\_t deviceAddress, uint8\_t regAddress, uint8\_t data){

Wire.beginTransmission(deviceAddress);

Wire.write(regAddress);

Wire.write(data);

Wire.endTransmission();

}

// read all 14 register

void Read\_RawValue(uint8\_t deviceAddress, uint8\_t regAddress){

Wire.beginTransmission(deviceAddress);

Wire.write(regAddress);

Wire.endTransmission();

Wire.requestFrom(deviceAddress, (uint8\_t)14);

AccelX = (((int16\_t)Wire.read()<<8) | Wire.read());

AccelY = (((int16\_t)Wire.read()<<8) | Wire.read());

AccelZ = (((int16\_t)Wire.read()<<8) | Wire.read());

Temperature = (((int16\_t)Wire.read()<<8) | Wire.read());

GyroX = (((int16\_t)Wire.read()<<8) | Wire.read());

GyroY = (((int16\_t)Wire.read()<<8) | Wire.read());

GyroZ = (((int16\_t)Wire.read()<<8) | Wire.read());

}

//configure MPU6050

void MPU6050\_Init(){

delay(150);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SMPLRT\_DIV, 0x07);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_PWR\_MGMT\_1, 0x01);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_PWR\_MGMT\_2, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_CONFIG, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_GYRO\_CONFIG, 0x00);//set +/-250 degree/second full scale

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_ACCEL\_CONFIG, 0x00);// set +/- 2g full scale

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_FIFO\_EN, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_INT\_ENABLE, 0x01);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_SIGNAL\_PATH\_RESET, 0x00);

I2C\_Write(MPU6050SlaveAddress, MPU6050\_REGISTER\_USER\_CTRL, 0x00);

}