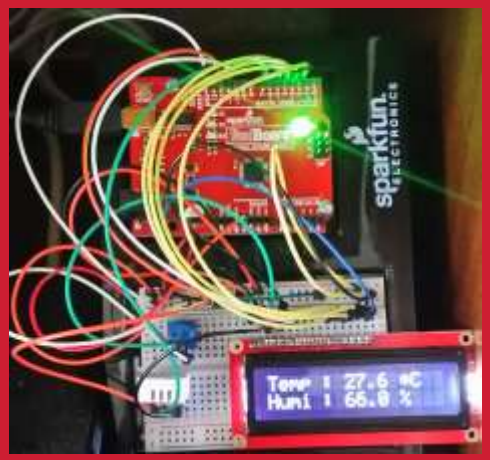
Arduino-IOT

[wk14]



nano33 BLE sensor

T, H, L, P, R, G, B

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining iot data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2023

Email : chaos21c@gmail.com

My ID

ID를 확인하고 github에 repo 만들기

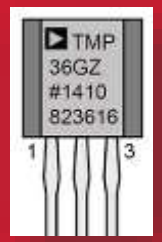
ID	성명
AA01	강동하
AA02	고서진
AA03	김민재
AA04	김예원
AA05	김주호
AA06	김창욱
AA07	김현서
AA08	박종혁
AA09	서명진
AA10	유동기
AA11	
AA12	이근보
AA13	정호기

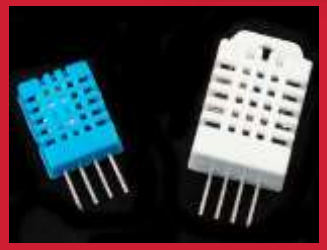
위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, README.md check

[Practice]





- [wk13]
  - Data Mining of IoT Data
  - Multi-sensor circuits (cds-dht22)
  - Complete your project
  - Upload folder: aann-rpt12
  - Use repo “aann” in github

wk13 : Practice : aann-rpt12

[Target of this week]

- Complete your works
- Save your outcomes and upload outputs in github

제출폴더명 : **aann-rpt12**

- 제출할 파일들

① **iot_csv.ipynb**

② **iot_ison.ipynb**

③ **All *.js**

④ **public/All *.html**

⑤ **client_iot.html**

⑥ **public/data/All data (*.csv)**

⑦ AAnn_s1000.csv

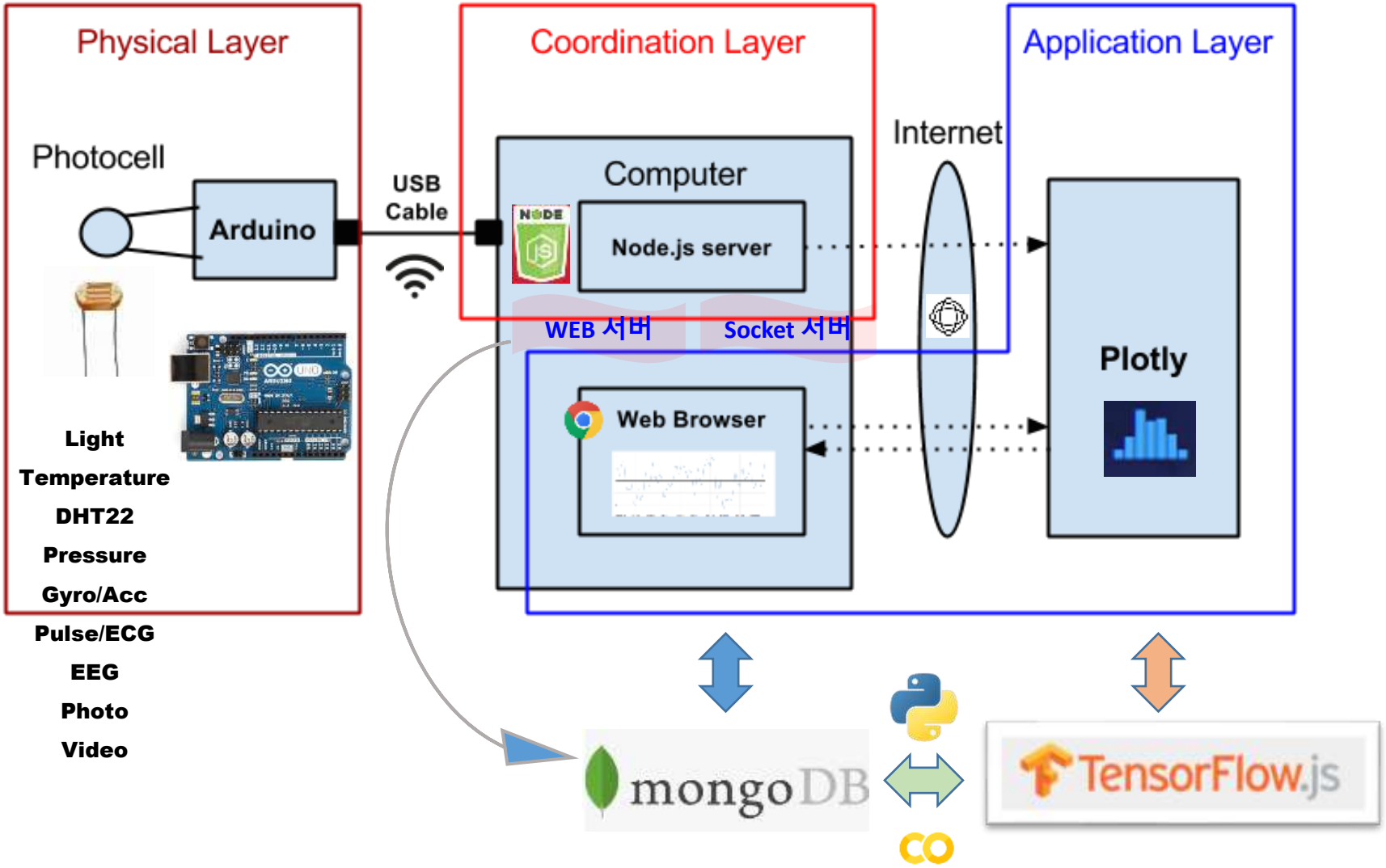
Purpose of AA

주요 수업 목표는 다음과 같다.

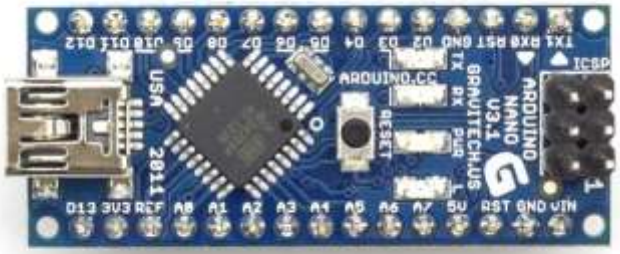
1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]

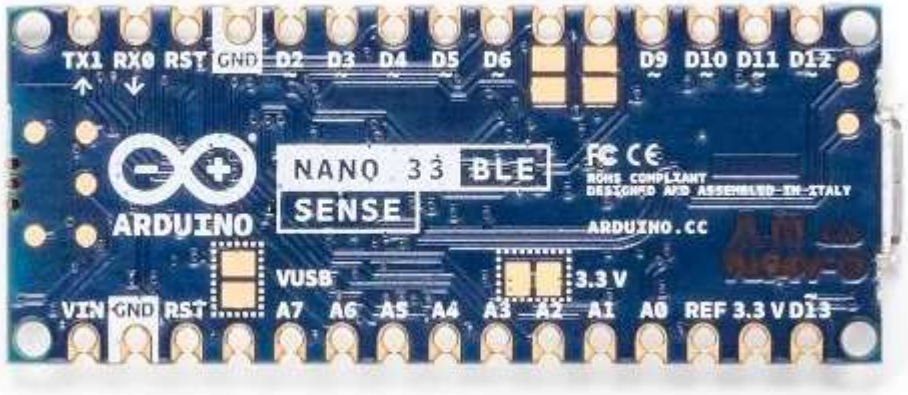

Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

1.2.4 Arduino hardware: nano, nano33



✓ Arduino Pro NANO

· ATmega168/328 microcontroller
· Input voltage: 7~12V
· 14 Digital I/O Pins (6 PWM outputs)
· 8 Analog Inputs
· 16KB Flash Memory
· 16Mhz Clock Speed



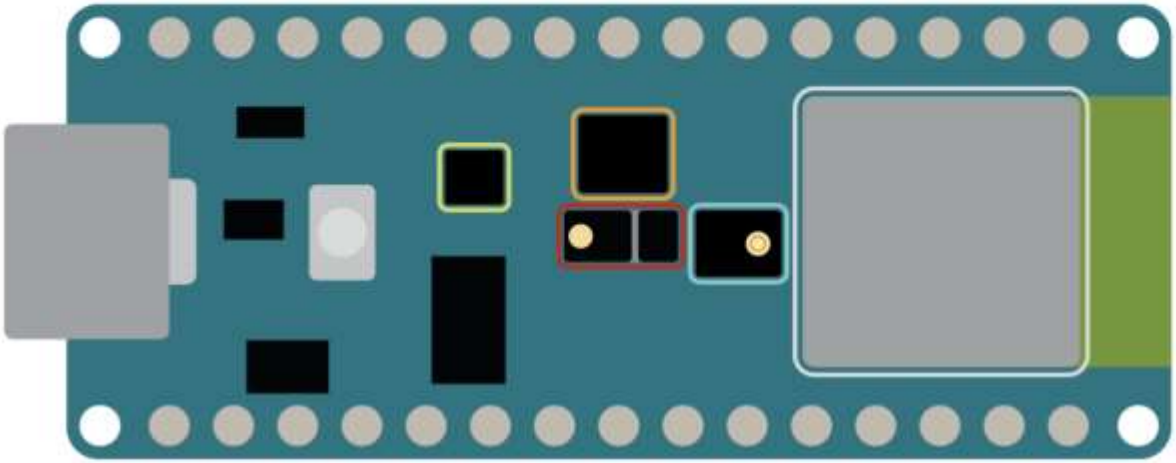
✓ Arduino NANO33

◆ BLE IOT (wifi)

◆ **BLE SENSE**

nano33BLE sensor

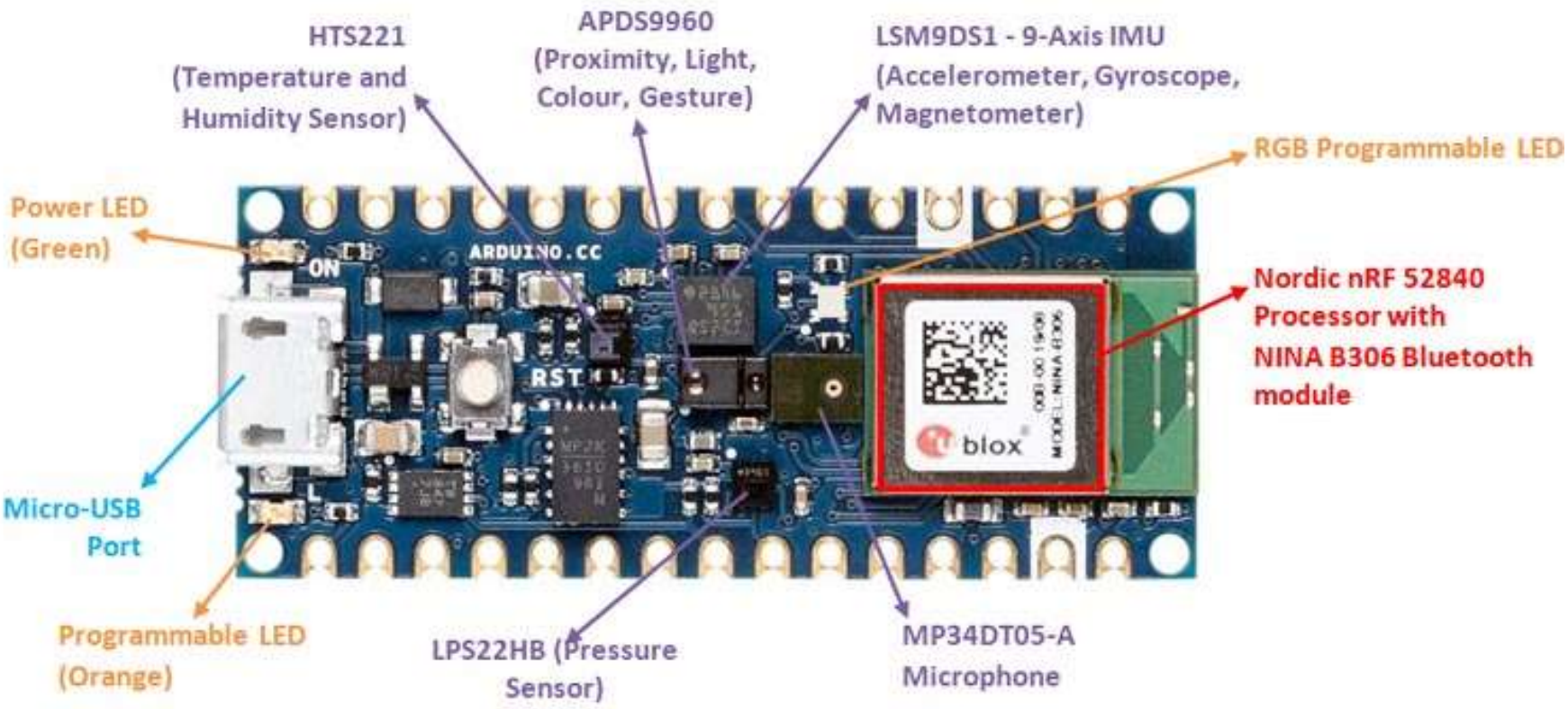
NANO 33 BLE SENSE



- Color, brightness, proximity and gesture sensor
- Digital microphone
- Motion, vibration and orientation sensor
- Temperature, humidity and pressure sensor
- Arm Cortex-M4 microcontroller and BLE module

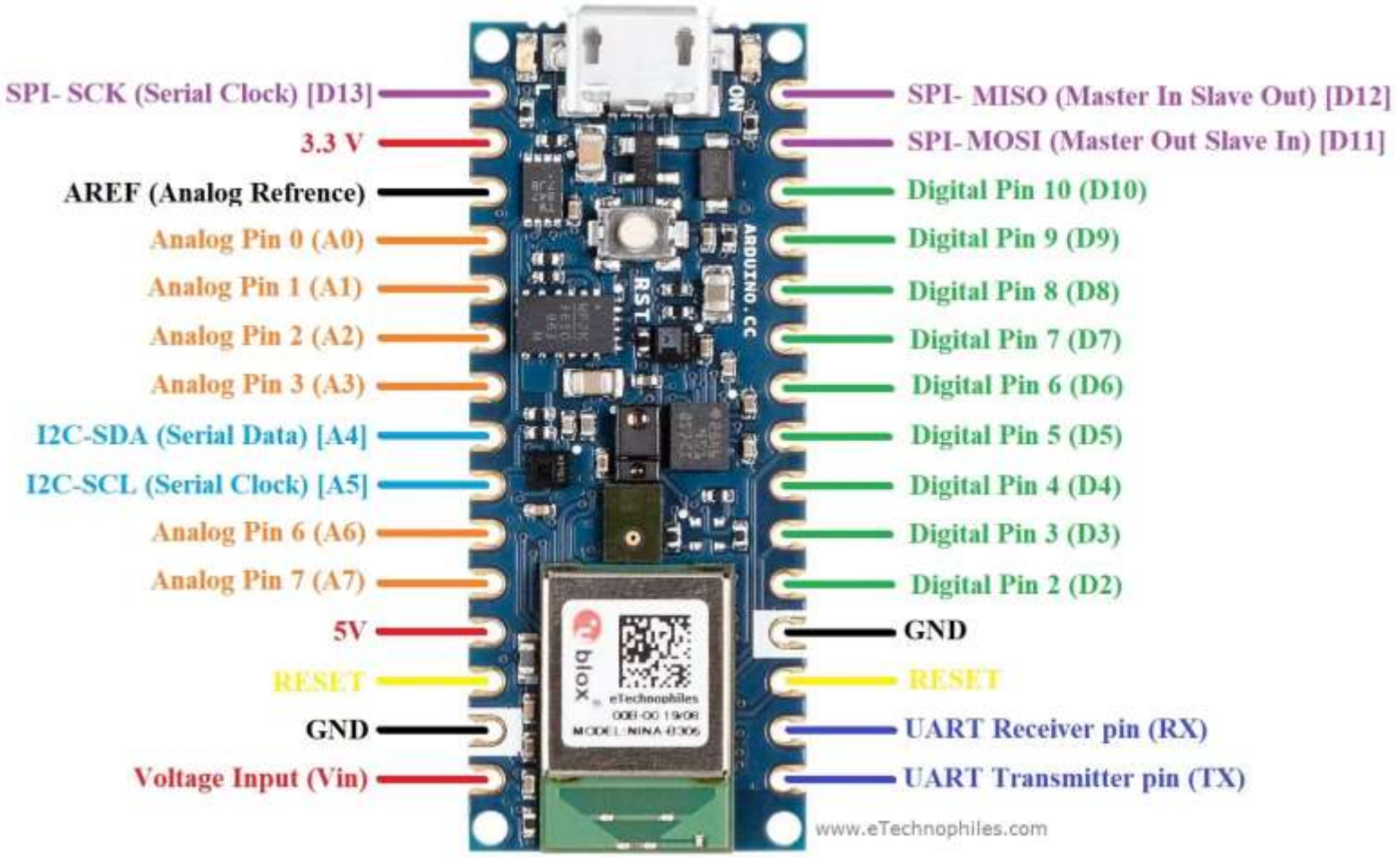
https://t1.daumcdn.net/cfile/tistory/99D8E84E5F93D2B109

nano33BLE sensor



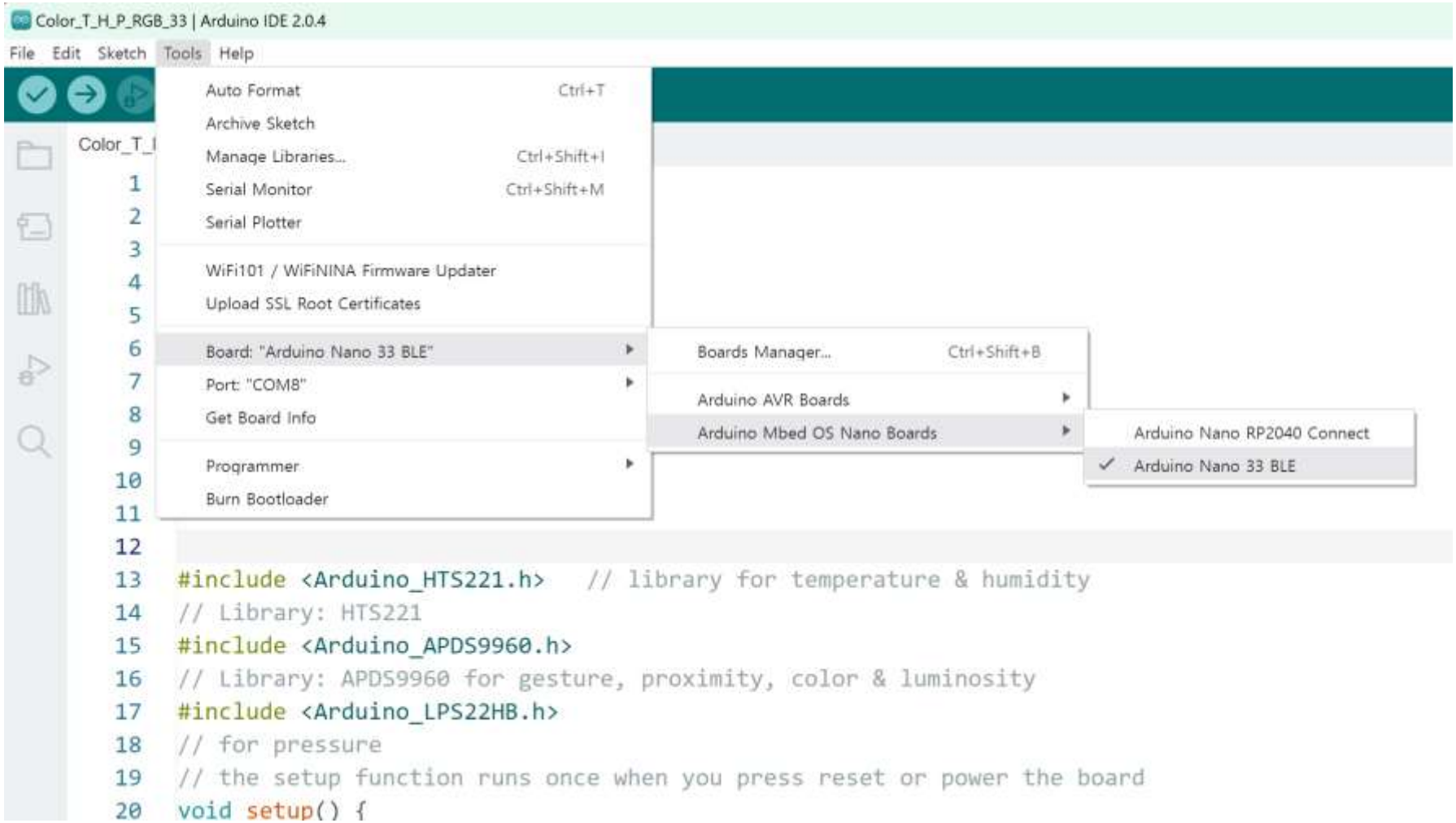
이미지 출처: https://circuitdigest.com/microcontroller-projects/arduino-nano-33-ble-sense-board-review-and-getting-started-guide

nano33BLE sensor - pins



이미지 출처: https://www.etechnophiles.com/arduino-nano-33-ble-sense-pinout-introduction-specifications/

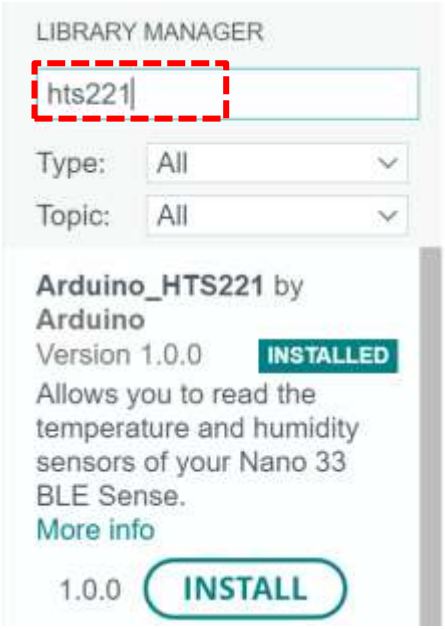
Layout [H S C]

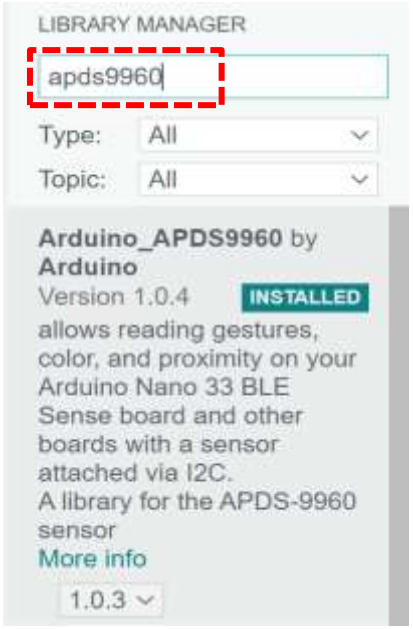

Color_T_H_P_RGB_33 | Arduino IDE 2.0.4
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board: "Arduino Nano 33 BLE"
Port: "COM8"
Get Board Info
Programmer
Burn Bootloader
Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
Arduino Mbed OS Nano Boards
Arduino Nano RP2040 Connect
Arduino Nano 33 BLE
#include <Arduino_HTS221.h> // library for temperature & humidity
// Library: HTS221
#include <Arduino_APDS9960.h>
// Library: APDS9960 for gesture, proximity, color & luminosity
#include <Arduino_LPS22HB.h>
// for pressure
// the setup function runs once when you press reset or power the board
void setup() {

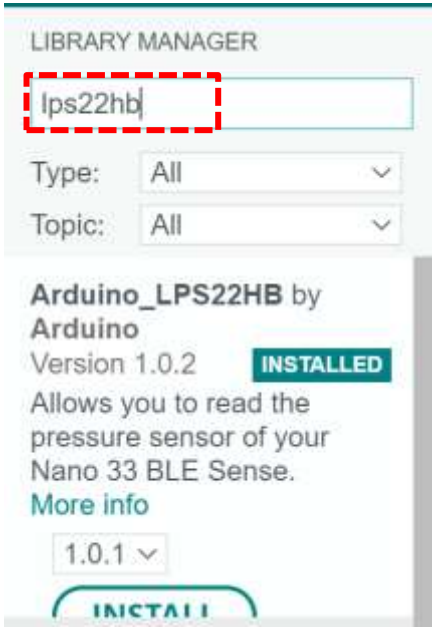
nano33 board

Sensor libraries

```
#include <Arduino_HTS221.h>   // library for temperature & humidity
// Library: HTS221
#include <Arduino_APDS9960.h>
// Library: APDS9960 for gesture, proximity, color & luminosity
#include <Arduino_LPS22HB.h>
// for pressure
```







APDS9960

- gestures
- proximity
- color, light intensity

begin()

end()

gestureAvailable()

readGesture()

colorAvailable()

readColor()

proximityAvailable()

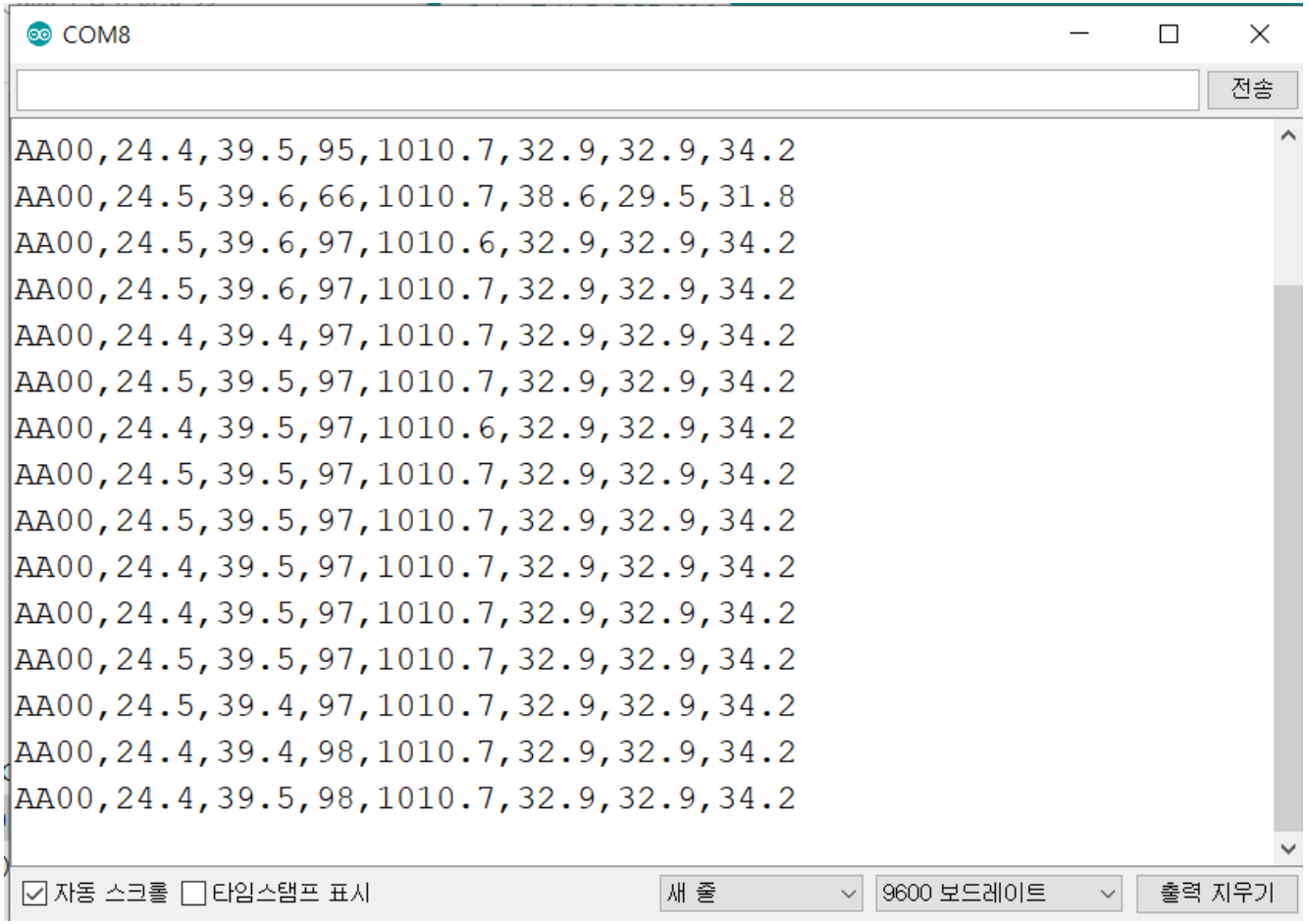
readProximity()

setGestureSensitivity()

setInterruptPin()

setLEDBoost()

Arduino: serial monitor



npm install

```
D:\aann\aann-rpt12\nano33>npm install
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression w
hen used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia
/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression w
hen used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia
/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression w
hen used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia
/debug/issues/797)

added 255 packages, and audited 256 packages in 14s

29 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

D:\aann\aann-rpt12\nano33>
```

Project: nano33BLE sensor

db33rgb.js

```
24  // Schema
25  var iotSchema = new Schema({
26      date : String,
27      temperature : String,
28      humidity : String,
29      luminosity : String,
30      pressure : String,
31      r_ratio : String,
32      g_ratio : String,
33      b_ratio : String
34  });
```

Project: nano33BLE sensor

db33rgb.js

```
86  // process data using parser
87  parser.on('data', (data) => { // call back when data is received
88      readData = data.toString(); // append data to buffer
89      firstcommaidx = readData.indexOf(',');
90      secondcommaidx = readData.indexOf(',',firstcommaidx+1);
91      thirdcommaidx = readData.indexOf(',',secondcommaidx+1);
92      fourthcommaidx = readData.indexOf(',',thirdcommaidx+1);
93      fifthcommaidx = readData.indexOf(',',fourthcommaidx+1);
94      sixthcommaidx = readData.indexOf(',',fifthcommaidx+1);
95
96      // parsing data into signals
97      if (readData.lastIndexOf(',') > firstcommaidx && firstcommaidx > 0) {
98          temp = readData.substring(firstcommaidx + 1, secondcommaidx);
99          humi = readData.substring(secondcommaidx + 1, thirdcommaidx);
100         lux = readData.substring(thirdcommaidx + 1, fourthcommaidx);
101         pres = readData.substring(fourthcommaidx + 1, fifthcommaidx);
102         rr = readData.substring(fifthcommaidx + 1, sixthcommaidx);
103         gg = readData.substring(sixthcommaidx + 1, readData.indexOf(',',sixthcommaidx+1));
104         bb = readData.substring(readData.lastIndexOf(',')+1);
```

Project: nano33BE sensor

db33rgb.js

```
108  dStr = getDateString();
109  mdata[0]=dStr;     // Date
110  mdata[1]=temp;     // temperature data
111  mdata[2]=humi;     // humidity data
112  mdata[3]=lux;      //  luminosity data
113  mdata[4]=pres;     // pressure data
114  mdata[5]=rr;        // r_ratio
115  mdata[6]=gg;        // g_ratio
116  mdata[7]=bb;        // b_ratio
117  //console.log(mdata);
118  var iotData = new Sensor({date:dStr, temperature:temp, humidity:humi, luminosity:lux, pressure:pres,
119      r_ratio:rr, g_ratio:gg, b_ratio:bb});
120  // save iot data to MongoDB
121  iotData.save(function(err,data) {
122      if(err) return handleEvent(err);
123      data.info();  // Display the information of iot data  on console.
124  })
```

Layout [H S C]

```
D:\aann\aann-rpt12\nano33>node -v
v16.17.0

D:\aann\aann-rpt12\nano33>node db33rgb
mongo db connection OK.
iotInfo: Current date: 2022-11-14 20:18:48.836, Temp: 25.1, Humi: 55.9, Lux: 112, Pres: 1007.6, R: 27.7, G: 46.1, B: 26.2
iotInfo: Current date: 2022-11-14 20:18:53.882, Temp: 25.2, Humi: 55.9, Lux: 113, Pres: 1007.6, R: 27.5, G: 45.8, B: 26.8
iotInfo: Current date: 2022-11-14 20:18:58.926, Temp: 25.2, Humi: 56.0, Lux: 115, Pres: 1007.6, R: 27.6, G: 45.5, B: 26.9
iotInfo: Current date: 2022-11-14 20:19:03.969, Temp: 25.2, Humi: 56.0, Lux: 114, Pres: 1007.6, R: 27.3, G: 46.2, B: 26.6
iotInfo: Current date: 2022-11-14 20:19:09.013, Temp: 25.2, Humi: 56.0, Lux: 113, Pres: 1007.6, R: 27.5, G: 45.8, B: 26.8
iotInfo: Current date: 2022-11-14 20:19:14.059, Temp: 25.1, Humi: 56.0, Lux: 114, Pres: 1007.6, R: 27.3, G: 46.2, B: 26.6
iotInfo: Current date: 2022-11-14 20:19:19.103, Temp: 25.2, Humi: 56.1, Lux: 113, Pres: 1007.6, R: 27.5, G: 45.8, B: 26.8
iotInfo: Current date: 2022-11-14 20:19:24.145, Temp: 25.3, Humi: 56.4, Lux: 113, Pres: 1007.6, R: 27.5, G: 45.8, B: 26.8
iotInfo: Current date: 2022-11-14 20:19:29.192, Temp: 25.3, Humi: 56.4, Lux: 115, Pres: 1007.6, R: 27.8, G: 45.1, B: 27.1
iotInfo: Current date: 2022-11-14 20:19:34.238, Temp: 25.3, Humi: 56.3, Lux: 11, Pres: 1007.6, R: 36.4, G: 36.4, B: 27.3
iotInfo: Current date: 2022-11-14 20:19:39.281, Temp: 25.3, Humi: 56.2, Lux: 112, Pres: 1007.5, R: 27.3, G: 46.0, B: 26.6
iotInfo: Current date: 2022-11-14 20:19:44.326, Temp: 25.3, Humi: 56.1, Lux: 111, Pres: 1007.6, R: 27.3, G: 46.0, B: 26.6
iotInfo: Current date: 2022-11-14 20:19:49.370, Temp: 25.3, Humi: 56.4, Lux: 113, Pres: 1007.6, R: 27.5, G: 45.8, B: 26.8
iotInfo: Current date: 2022-11-14 20:19:54.413, Temp: 25.3, Humi: 56.2, Lux: 93, Pres: 1007.5, R: 27.1, G: 46.6, B: 26.3
iotInfo: Current date: 2022-11-14 20:19:59.459, Temp: 25.5, Humi: 59.6, Lux: 110, Pres: 1007.6, R: 27.5, G: 45.7, B: 26.8
iotInfo: Current date: 2022-11-14 20:20:04.506, Temp: 25.3, Humi: 60.6, Lux: 96, Pres: 1007.5, R: 27.3, G: 46.3, B: 26.4
iotInfo: Current date: 2022-11-14 20:20:09.548, Temp: 25.5, Humi: 63.0, Lux: 110, Pres: 1007.6, R: 27.0, G: 46.0, B: 27.0
iotInfo: Current date: 2022-11-14 20:20:14.593, Temp: 25.4, Humi: 63.1, Lux: 113, Pres: 1007.5, R: 27.0, G: 46.1, B: 27.0
iotInfo: Current date: 2022-11-14 20:20:19.638, Temp: 25.4, Humi: 62.1, Lux: 113, Pres: 1007.5, R: 27.1, G: 45.7, B: 27.1
iotInfo: Current date: 2022-11-14 20:20:24.683, Temp: 25.4, Humi: 61.5, Lux: 111, Pres: 1007.6, R: 27.3, G: 46.0, B: 26.6
```

Project: nano33BLE sensor

express33rgb.js

```
23  // Schema
24  var iotSchema = new Schema({
25      date : String,
26      temperature : String,
27      humidity : String,
28      luminosity : String,
29      pressure : String,
30      r_ratio : String,
31      g_ratio : String,
32      b_ratio : String
33  });
34  var Sensor = mongoose.model("Sensor", iotSchema);  // sensor data model
35
```

Network socket/DB server : port=3000

Express server : port=3030

node db33rgb

node express33rgb

```
D:\aann\aann-rpt12\nano33>node db33rgb
mongo db connection OK.
iotInfo: Current date: 2022-11-14 20:52:35.126, Temp: 25.9, Humi: 54.5, Lux: 109, Pres: 1007.6,
 R: 27.2, G: 46.3, B: 26.5
iotInfo: Current date: 2022-11-14 20:52:40.170, Temp: 26.0, Humi: 54.5, Lux: 111, Pres: 1007.6,
 R: 27.3, G: 46.0, B: 26.6
iotInfo: Current date: 2022-11-14 20:52:45.214, Temp: 26.0, Humi: 54.5, Lux: 109, Pres: 1007.6,
 R: 27.2, G: 46.3, B: 26.5
iotInfo: Current date: 2022-11-14 20:52:50.259, Temp: 26.0, Humi: 54.5, Lux: 109, Pres: 1007.5,
 R: 27.0, G: 46.0, B: 27.0
iotInfo: Current date: 2022-11-14 20:52:55.302, Temp: 26.0, Humi: 54.6, Lux: 110, Pres: 1007.6,
 R: 27.2, G: 46.3, B: 26.5
iotInfo: Current date: 2022-11-14 20:53:00.349, Temp: 26.0, Humi: 54.6, Lux: 108, Pres: 1007.6,
 R: 27.4, G: 45.9, B: 26.7
iotInfo: Current date: 2022-11-14 20:53:05.390, Temp: 26.0, Humi: 54.5, Lux: 111, Pres: 1007.6,
 R: 27.3, G: 46.0, B: 26.6
iotInfo: Current date: 2022-11-14 20:53:10.434, Temp: 26.0, Humi: 54.5, Lux: 110, Pres: 1007.6,
 R: 27.0, G: 46.0, B: 27.0
iotInfo: Current date: 2022-11-14 20:53:15.479, Temp: 26.0, Humi: 54.5, Lux: 108, Pres: 1007.5,
 R: 27.2, G: 46.3, B: 26.5
```

```
D:\aann\aann-rpt12\nano33>node express33rgb
Express_IOT is running at port:3030, CORS power
ed!
mongo db connection OK.
```

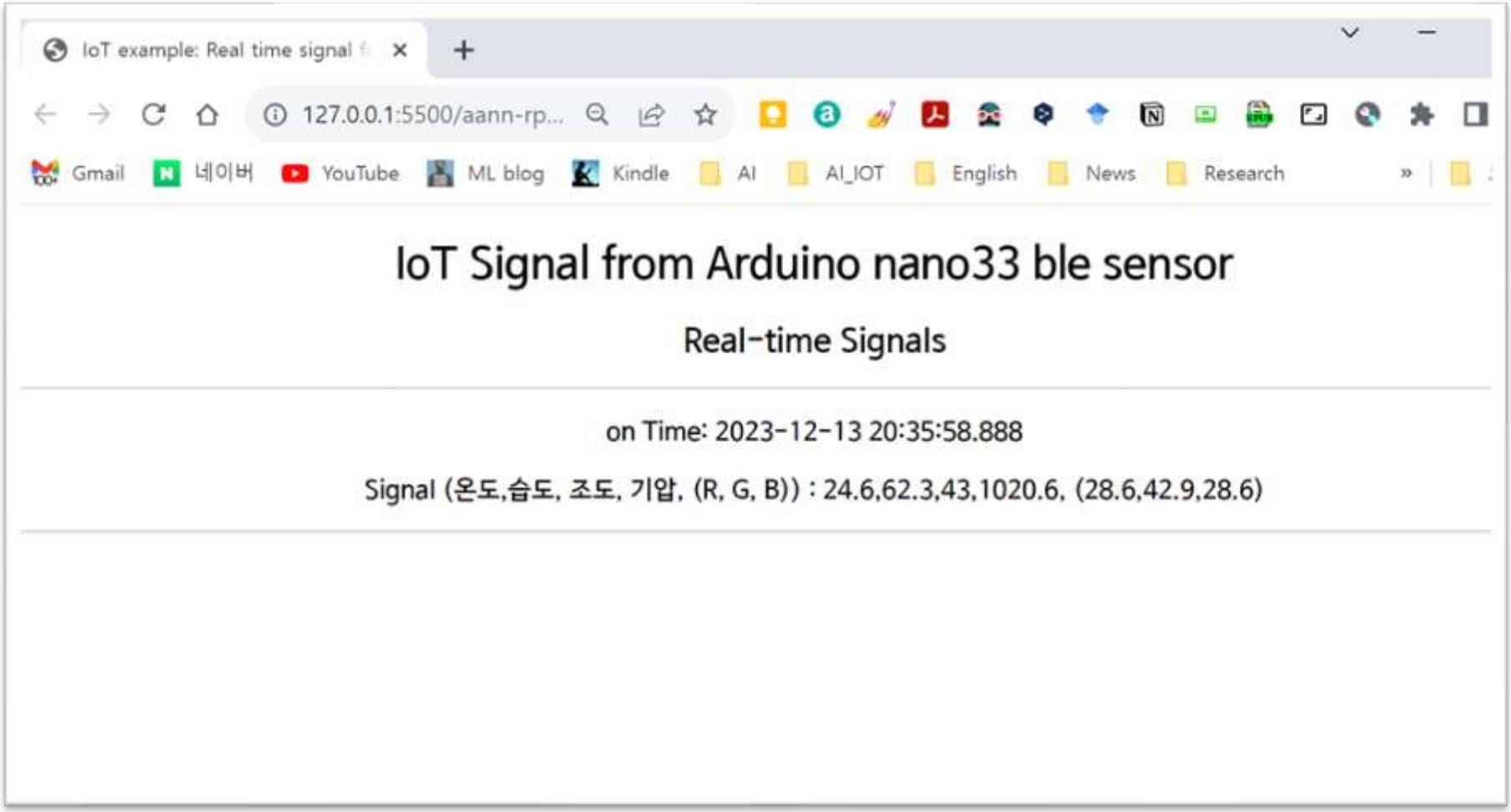
node

node

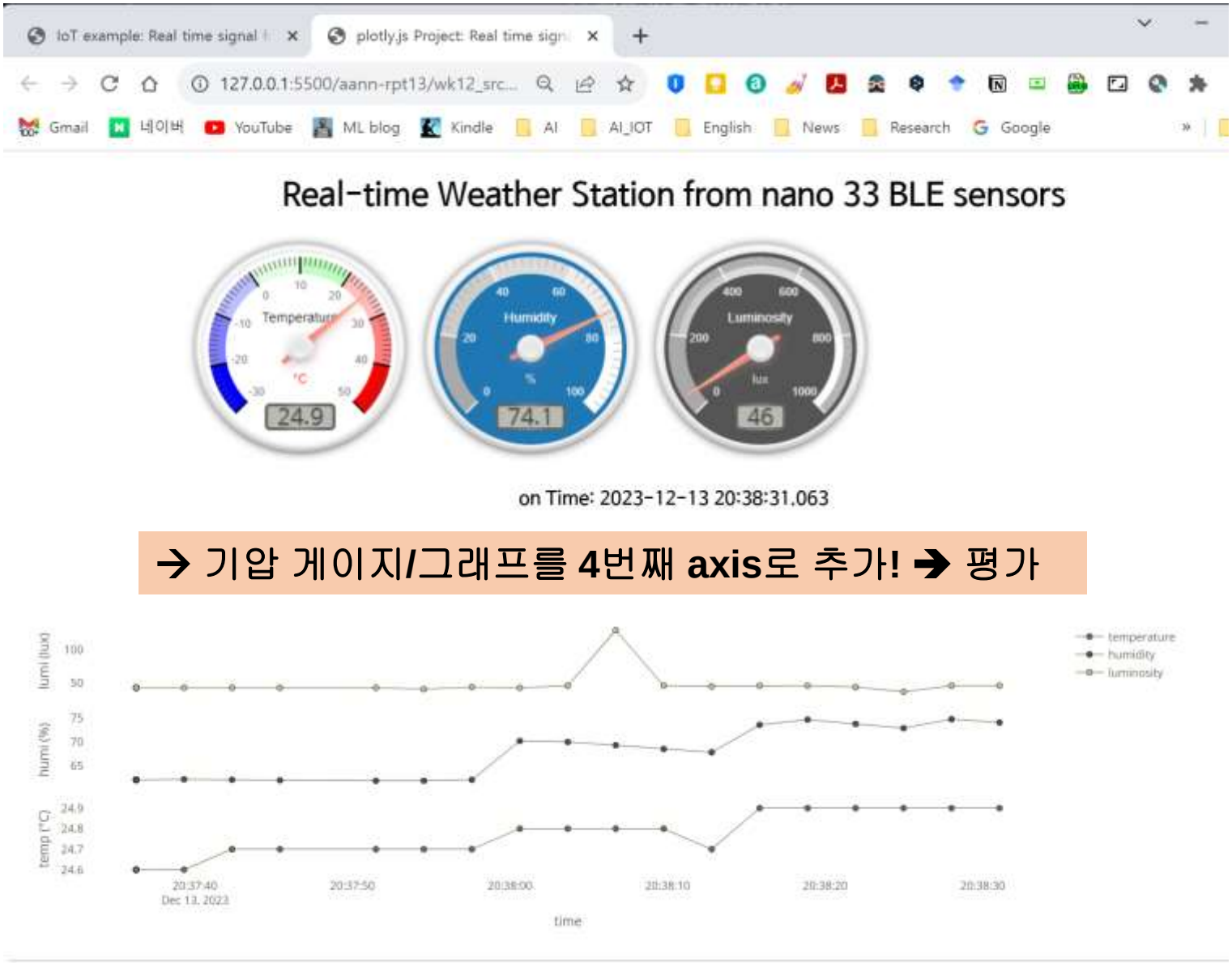
mongo shell

```
> show dbs
admin           0.000GB
config          0.000GB
iot             0.000GB
iot00           0.000GB
iot10           0.000GB
iot33           0.000GB
iot33rgb211119  0.000GB
local           0.000GB
test2           0.000GB
> use iot33
switched to db iot33
> show collections
sensors
> db.sensors.count()
57
> db.sensors.find()
{ "_id" : ObjectId("63722bb7535a16bfb82a8bc1"), "date" : "2022-11-14 20:51:19.470", "temperature" :
 "26.0", "humidity" : "54.6", "luminosity" : "109", "pressure" : "1007.7", "r_ratio" : "27.0", "g_r
atio" : "46.0", "b_ratio" : "27.0", "__v" : 0 }
{ "_id" : ObjectId("63722bbc535a16bfb82a8bc3"), "date" : "2022-11-14 20:51:24.514", "temperature" :
 "26.0", "humidity" : "54.7", "luminosity" : "107", "pressure" : "1007.6", "r_ratio" : "27.4", "g_r
atio" : "45.9", "b_ratio" : "26.7", "__v" : 0 }
{ "_id" : ObjectId("63722bc1535a16bfb82a8bc5"), "date" : "2022-11-14 20:51:29.558", "temperature" :
 "26.0", "humidity" : "54.7", "luminosity" : "107", "pressure" : "1007.6", "r_ratio" : "26.9", "g_r
atio" : "46.3", "b_ratio" : "26.9", "__v" : 0 }
{ "_id" : ObjectId("63722bc6535a16bfb82a8bc7"), "date" : "2022-11-14 20:51:34.602", "temperature" :
 "26.0", "humidity" : "54.6", "luminosity" : "107", "pressure" : "1007.7", "r_ratio" : "26.9", "g_r
atio" : "46.3", "b_ratio" : "26.9", "__v" : 0 }
```

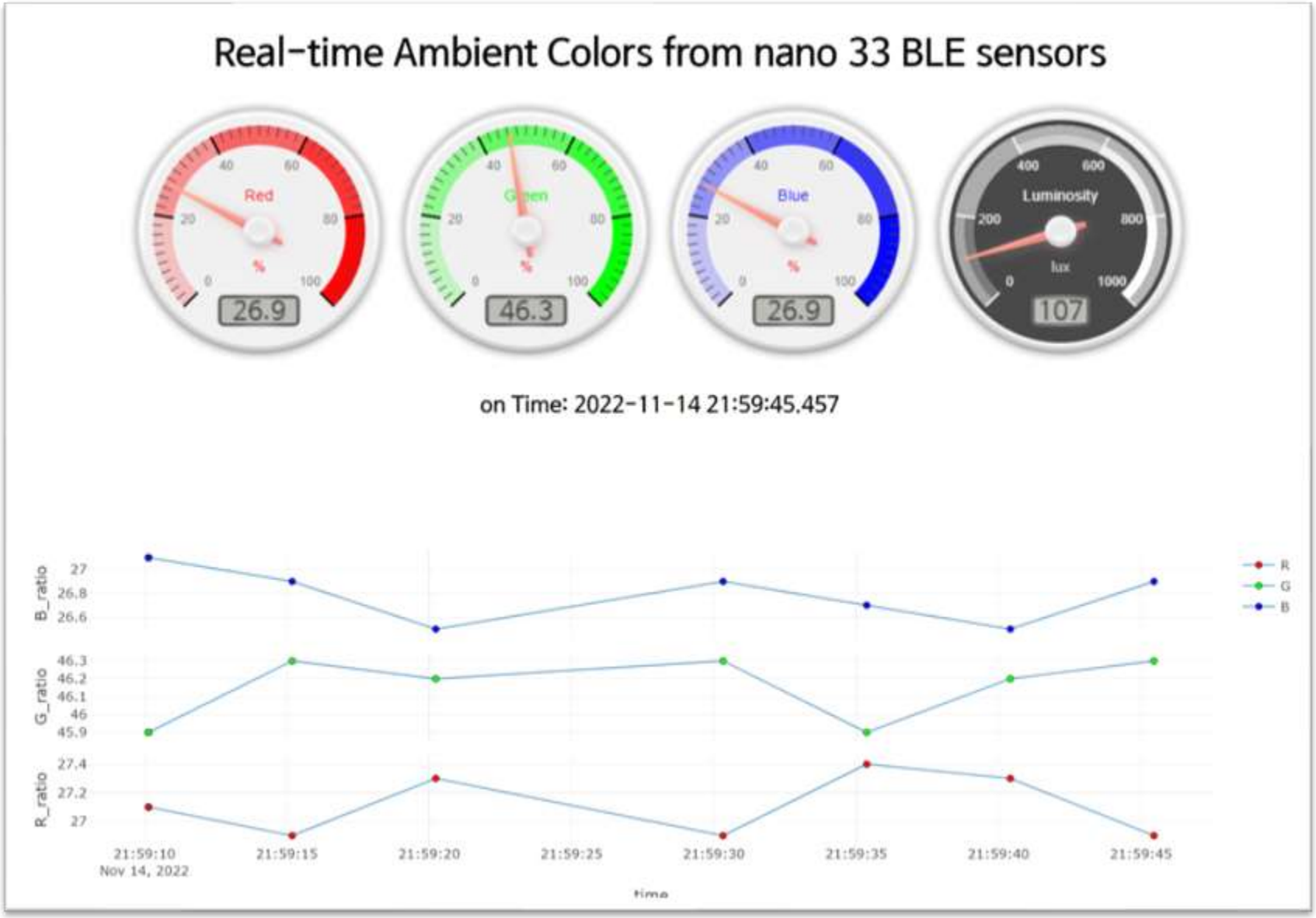
http://localhost:3030/client_signal.html



http://localhost:3030/client_33.html



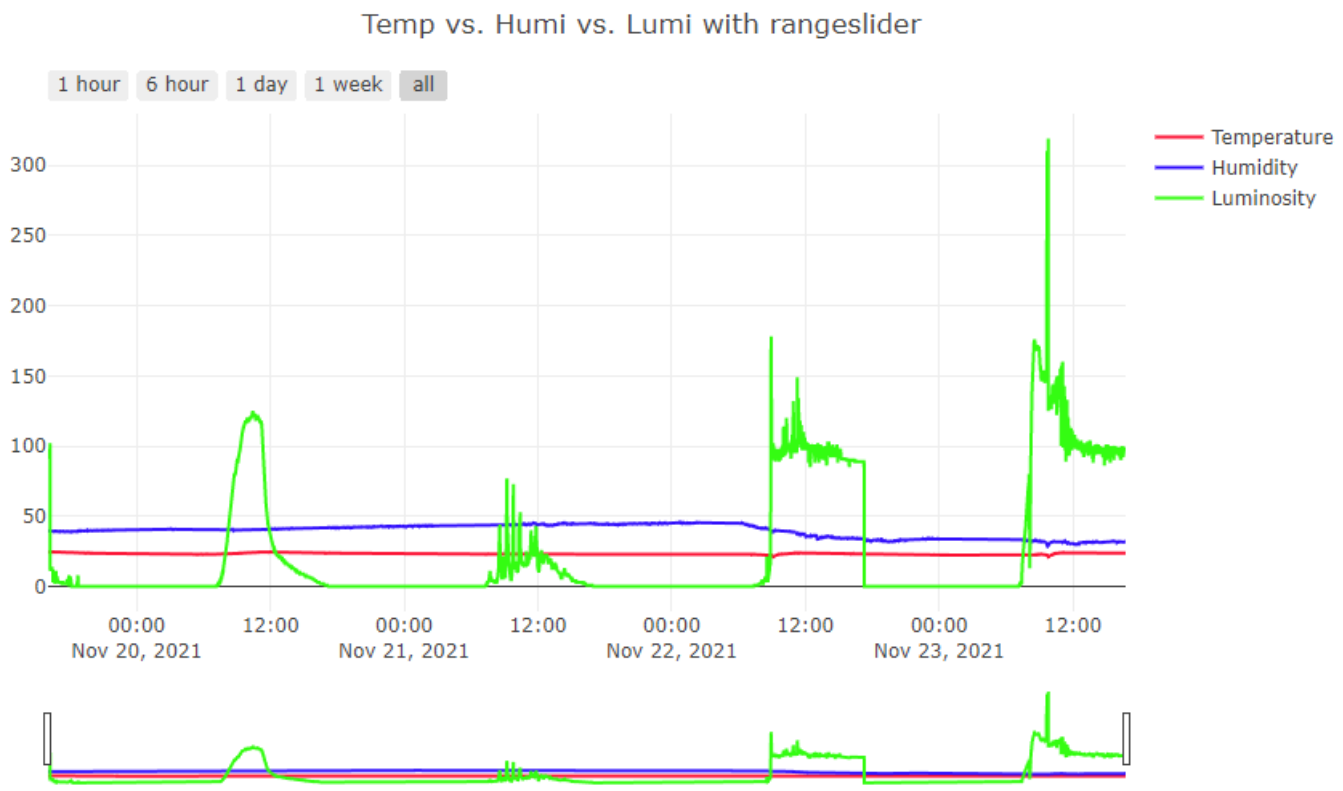
http://localhost:3030/client_33rgb.html



http://localhost:3030/client_33iot.html

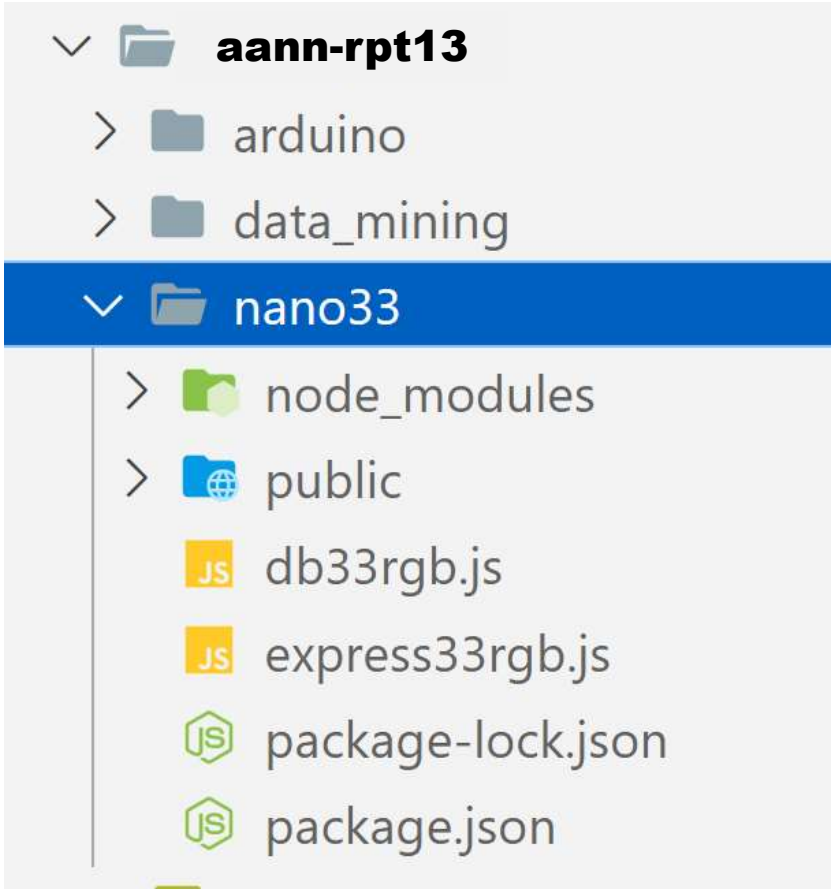
MongoDB database visualization by AA00

Time series : Multiple data from nano 33 ble sensor



Project: nano33BLE sensor

작업 폴더 구조 [2023-nano33-project]

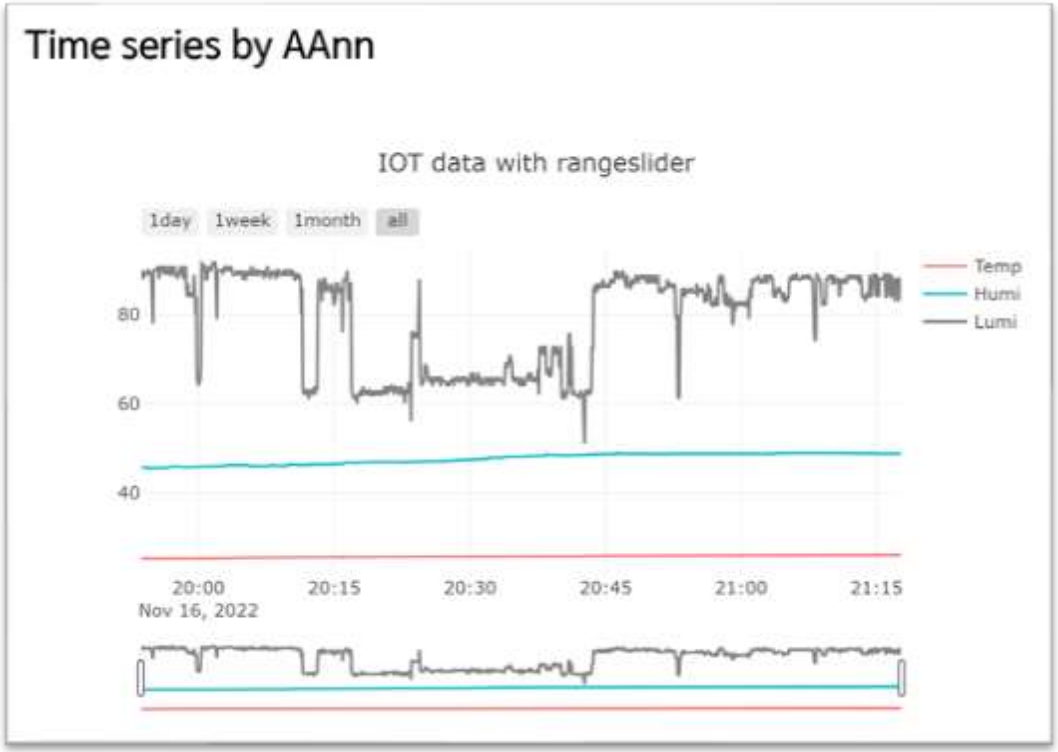


A5.9.8 MongoDB management

[DIY]
1. iot33 db의 최근 데이터 2000개를 csv 파일 (aann_iot33_1000.csv)로 저장하시오.
2. 저장된 aann_iot33_1000.csv 파일을 public/data 폴더에 복사.
3. csv 파일을 이용하는 Rangeslider가 포함된 웹 클라이언트 client_33csv.html 파일을 완성하시오.
4. localhost:3030/client_33csv.html 로 실행하고 확인.

[hint] iot33 db의 최근 데이터 500개를 csv 파일 (iot_500.csv)로 저장할 때,

- mongoexport /db:iot33 /collection:sensors /sort:"{_id: -1}" /limit:500 /fields:date,temperature,humidity,luminosity /type:csv /out:iot_500.csv



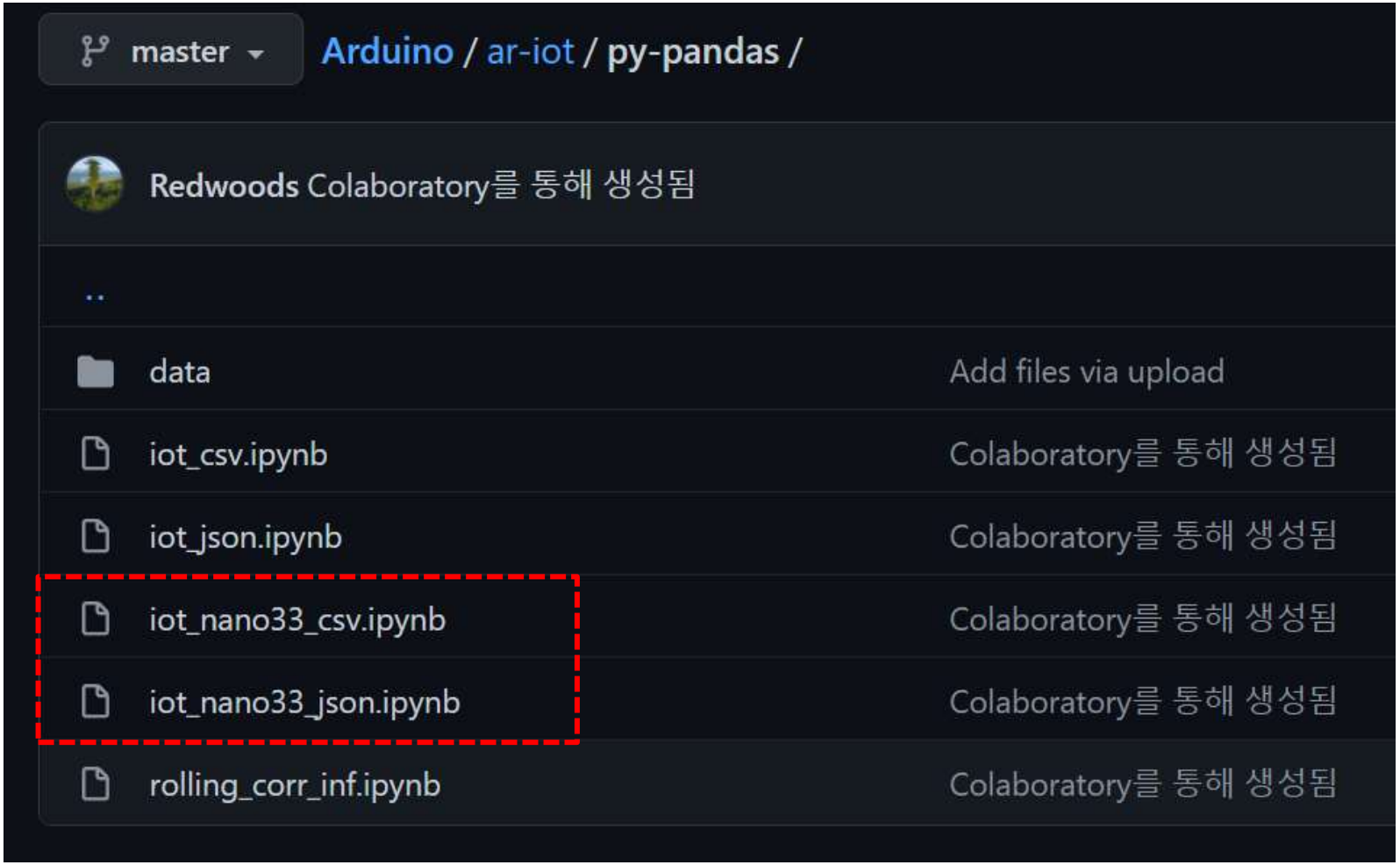
client_33csv.html

코드를 완성하시오.

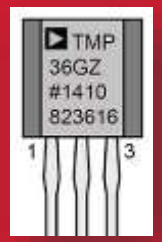
public 폴더에 저장

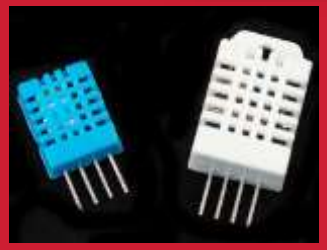
Project: nano33BLE sensor

[2023-project] IoT data mining in Colab



[Practice]





- [wk14]
  - IoT Project: nano33ble
  - Multi-sensor circuits)
  - Complete your project
  - Upload folder: aann-rpt13
  - Use repo “aann” in github

wk15 : Practice : aann-rpt13

- [Target of this week]
  - Complete your works
  - Save your outcomes and upload outputs in github

제출폴더명 : **aann-rpt13**

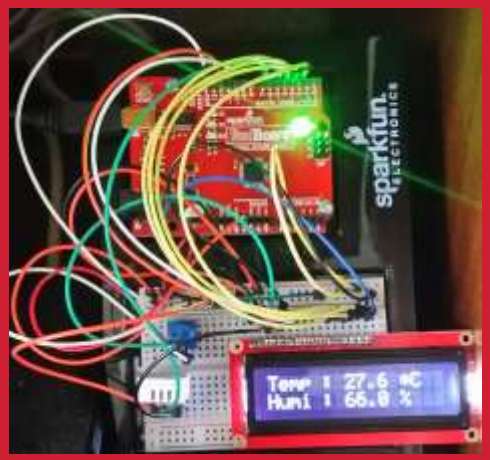
- 제출할 파일들

① **iot_nano33_csv.ipynb** in data_mining folder

② **All *.js** in nano33 folder

③ **public/All *.html**

④ aann_Iot33_1000.csv in public/data folder



Arduino-IOT

[wk14]

nano33 BLE sensor Gesture, Motion

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining iot data using Python

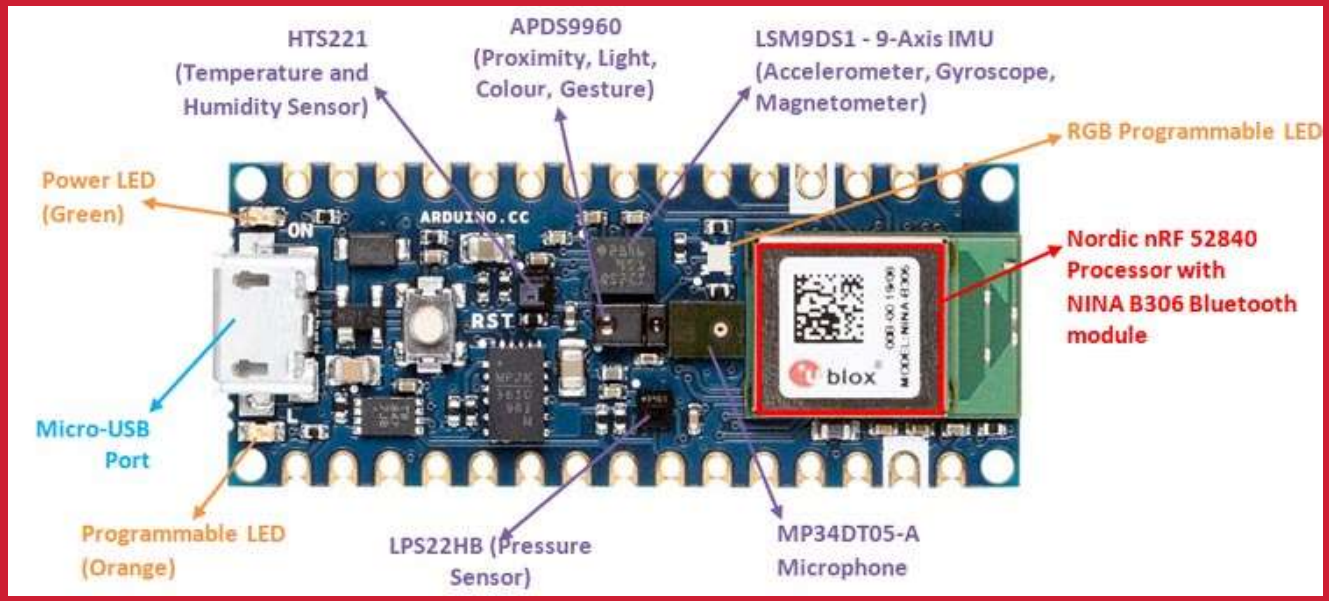
Drone-IoT-Comsi, INJE University

2nd semester, 2023

Email : chaos21c@gmail.com

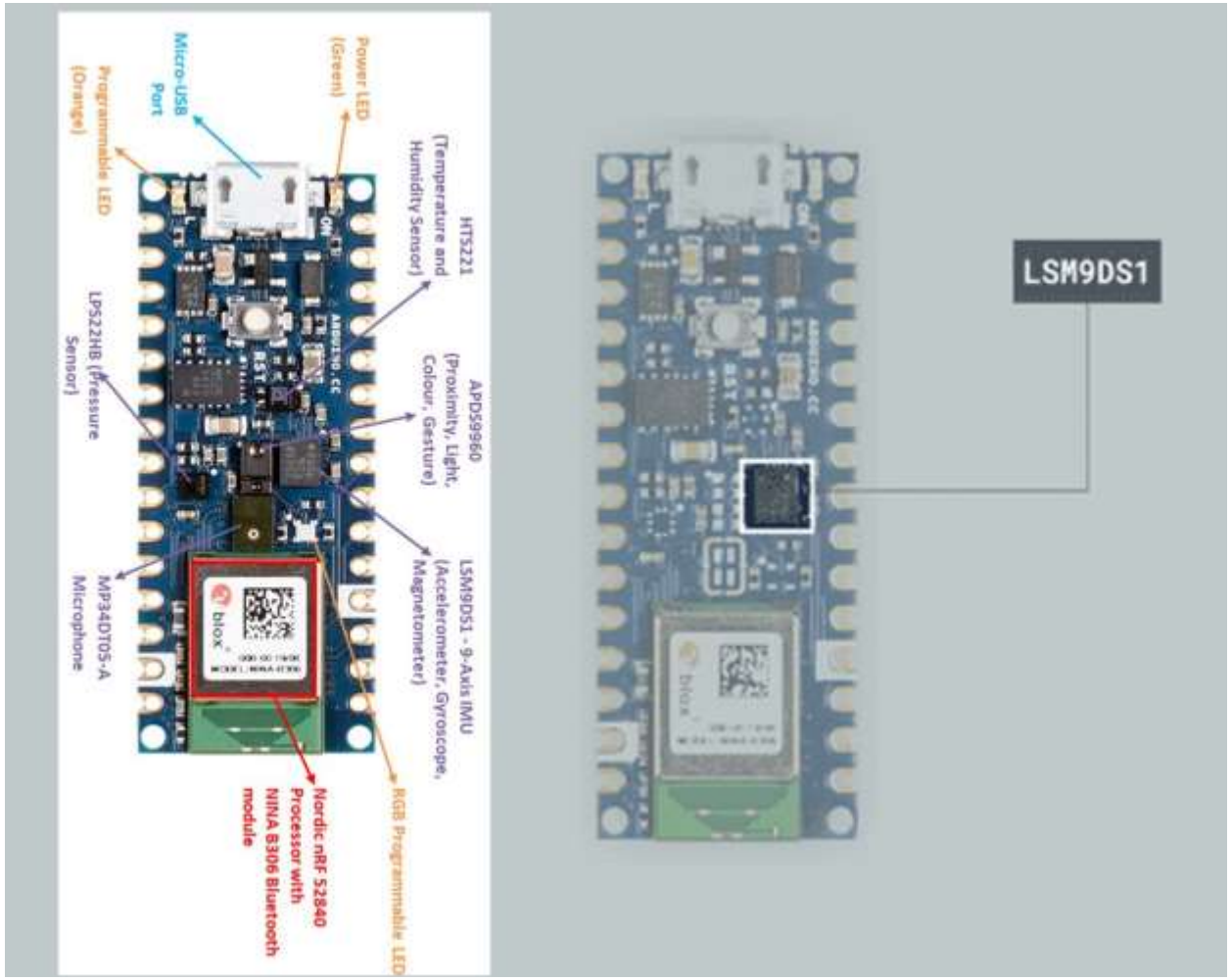
IMU

nano33BLE sensor



IMU (Inertial Measurement Unit)

LSM9DS1, 9축 IMU센서: acc, gyro, mag



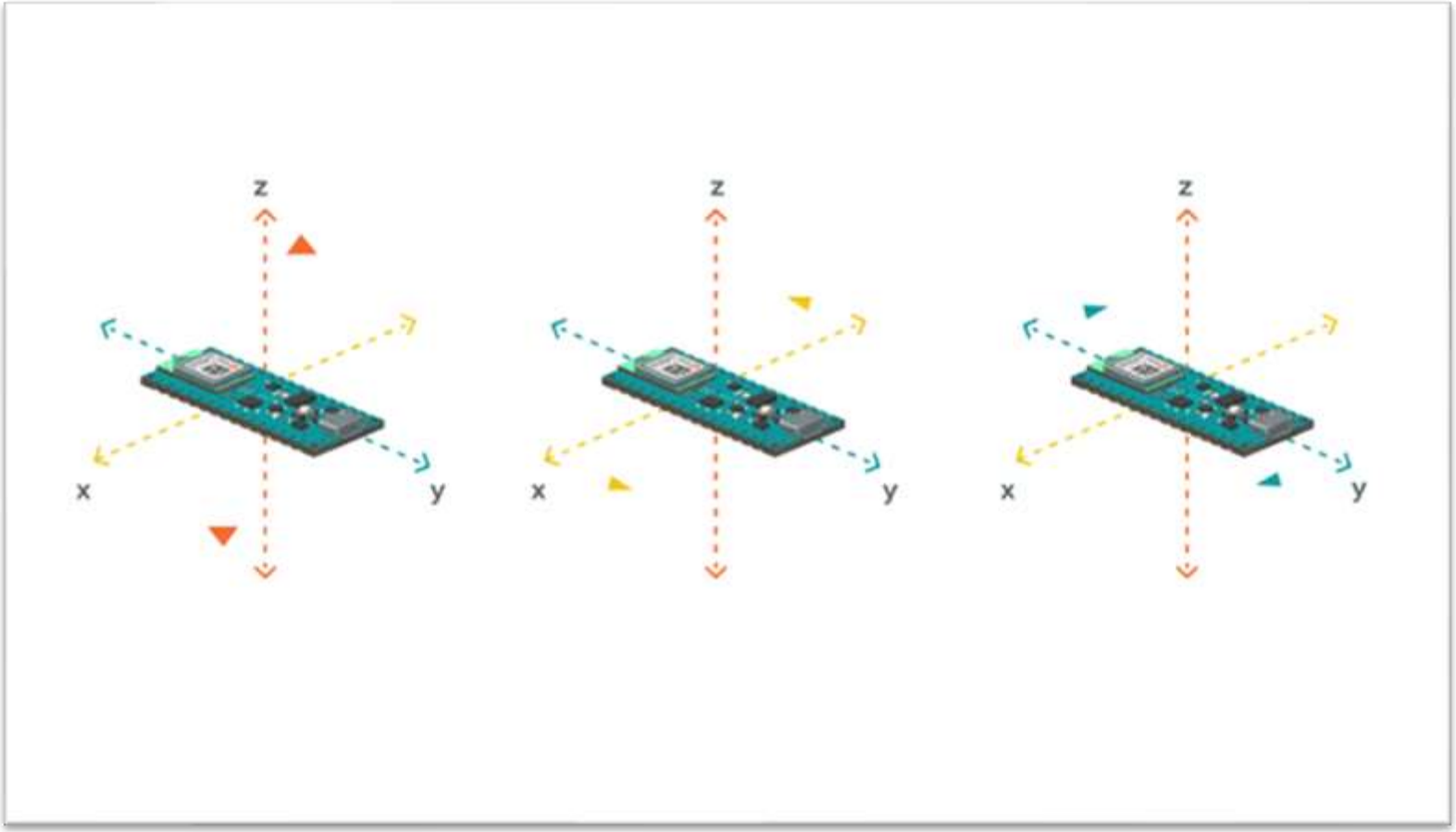
LSM9DS1, 9축 IMU센서: acc, gyro, mag

The LSM9DS1 Library

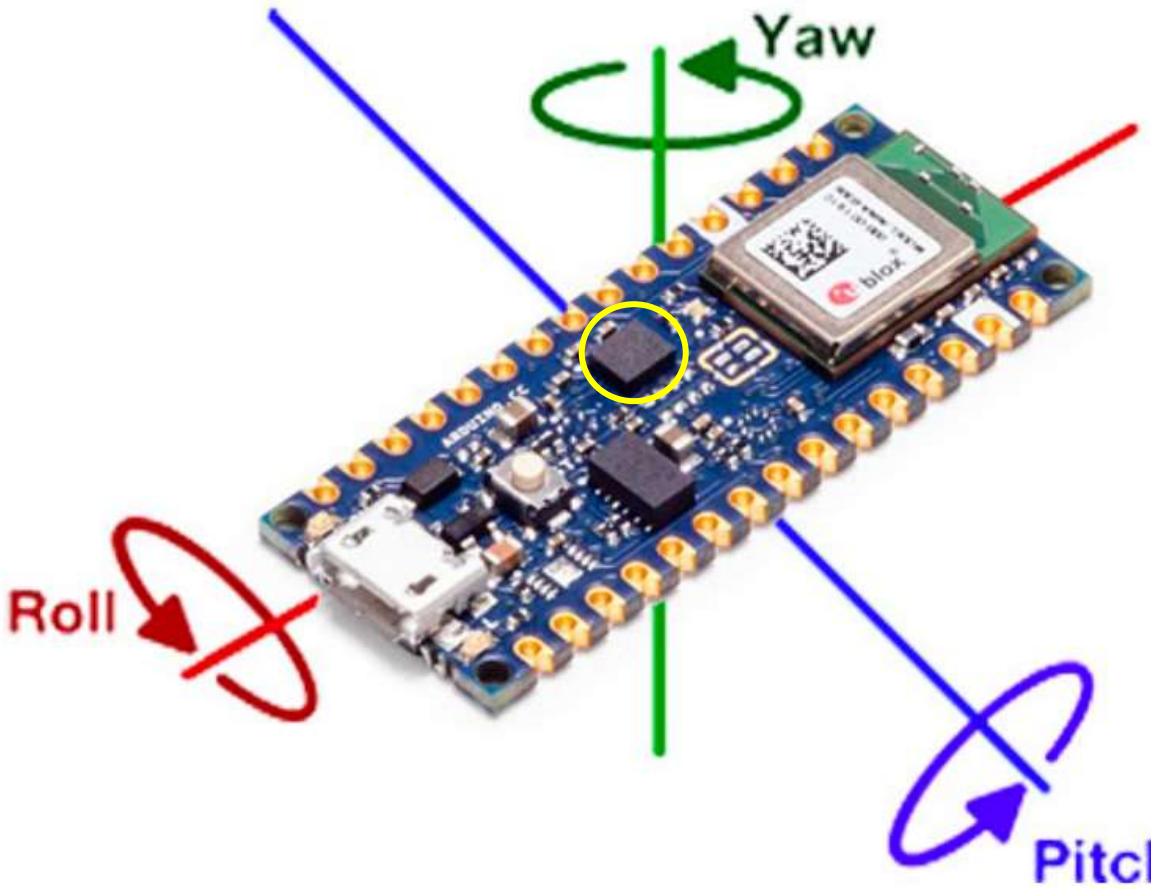
The Arduino LSM9DS1 library allows us to use the Arduino Nano 33 BLE IMU module without having to go into complicated programming. The library takes care of the sensor initialization and sets its values as follows:

- **Accelerometer** range is set at [-4, +4]g -/+0.122 mg.
- **Gyroscope** range is set at [-2000, +2000] dps +/-70 mdps.
- **Magnetometer** range is set at [-400, +400] uT +/-0.014 uT.
- **Accelerometer** output data rate is fixed at 104 Hz.
- **Gyroscope** output data rate is fixed at 104 Hz.
- **Magnetometer** output data rate is fixed at 20 Hz.

LSM9DS1, 9축 IMU센서: acc , gyro, mag



LSM9DS1, 9축 IMU센서: acc, gyro, mag



자이로 스코프 측정, 이미지 출처 https://www.mauroalfieri.it/elettronica/arduino-nano-33-ble-giroscopio-lsm9ds1.html

https://fishpoint.tistory.com/4438

LSM9DS1, 9축 IMU센서: acc, gyro, mag

LSM9DS1_Basic.ino

```
#include <Arduino_LSM9DS1.h>

void setup() {
  Serial.begin(9600);
  while (!Serial);  // 직렬통신 연결될 때 까지 대기
  Serial.println("Started");
  if (!IMU.begin()) { // IMU센서를 초기화합니다. 초기화중 문제가 발생하면 오류를 발생시킵니다.
    Serial.println("Failed to initialize IMU!");
    while (1);
  }
}
float ax, ay, az; // 가속도 센서의 XYZ값을 저장할 변수입니다.
float gx, gy, gz; // 자이로 센서의 XYZ값을 저장할 변수입니다.
float mx, my, mz; // 지자기 센서의 XYZ값을 저장할 변수입니다.

void loop() {
  delay(500);
  if (IMU.accelerationAvailable()) { // 가속도 센서의 값을 출력합니다.
    IMU.readAcceleration(ax, ay, az); // x, y, z에 각 축별 데이터를 넣습니다.
    Serial.print("ACC 센서 - ");
    Serial.print(ax);
    Serial.print(',');
    Serial.print(ay);
    Serial.print(',');
    Serial.print(az);
    Serial.print(" G's");
    Serial.print(',');
  }
```

LSM9DS1, 9축 IMU센서: acc, gyro, mag

```
30    if (IMU.gyroscopeAvailable()) { // 자이로 센서의 값을 출력합니다.
31      IMU.readGyroscope(gx, gy, gz);
32      Serial.print("GYRO 센서 - ");
33      Serial.print(gx);
34      Serial.print(',');
35      Serial.print(gy);
36      Serial.print(',');
37      Serial.print(gz);
38      Serial.print(" degrees/second");
39      Serial.print(',');
40    }
41
42    if (IMU.magneticFieldAvailable()) { // 지자기 센서의 값을 출력합니다.
43      IMU.readMagneticField(mx, my, mz);
44      Serial.print("MAG 센서 - ");
45      Serial.print(mx);
46      Serial.print(',');
47      Serial.print(my);
48      Serial.print(',');
49      Serial.print(mz);
50      Serial.println(" uT");
51    }
52    // Serial.println();
53  }
54
```

LSM9DS1, 9축 IMU센서: acc, gyro, mag

```
Started
ACC 센서 - 0.02,-0.04,0.96 G's,GYRO 센서 - 1.10,-0.79,-0.12 degrees/second,MAG 센서 - -24.40,2.99,-13.93 uT
ACC 센서 - 0.02,-0.04,0.96 G's,GYRO 센서 - 1.22,-0.92,-0.12 degrees/second,MAG 센서 - -24.58,2.11,-13.27 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.28,-0.98,-0.06 degrees/second,MAG 센서 - -23.83,1.90,-13.56 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.04,-0.92,-0.12 degrees/second,MAG 센서 - -24.41,2.39,-12.95 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.28,-1.04,0.00 degrees/second,MAG 센서 - -24.28,4.21,-13.21 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.16,-1.04,-0.12 degrees/second,MAG 센서 - -24.72,1.66,-11.68 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.46,-0.55,0.00 degrees/second,MAG 센서 - -24.34,2.48,-12.59 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.04,-1.16,0.00 degrees/second,MAG 센서 - -24.65,2.88,-13.54 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.16,-1.10,-0.06 degrees/second,MAG 센서 - -24.21,2.72,-13.13 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 0.98,-1.16,-0.18 degrees/second,MAG 센서 - -24.13,2.39,-12.68 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 0.98,-0.92,0.00 degrees/second,MAG 센서 - -24.06,2.03,-13.05 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 1.10,-1.10,0.06 degrees/second,MAG 센서 - -23.96,6.35,-13.18 uT
ACC 센서 - 0.02,-0.04,0.97 G's,GYRO 센서 - 0.92,-0.67,-0.06 degrees/second,MAG 센서 - -24.05,2.17,-12.16 uT
```

LSM9DS1, 9축 IMU센서: acc, gyro, mag

IoT 데이터 수집 형태로 출력을 변경하시오.

LSM9DS1_IoT.ino 로 저장

```
-0.02,0.76,0.48,108.58,-7.26,15.08,-19.34,-10.67,9.59
-0.43,-1.02,-0.49,103.88,-49.99,-3.30,-26.07,25.98,18.08
0.10,0.82,0.74,-228.39,49.38,-27.10,-16.94,-10.21,5.30
0.38,-0.11,0.33,70.62,112.61,-47.85,-9.86,9.01,-9.19
1.12,0.27,0.61,-91.31,-158.75,49.87,1.83,-3.25,-3.11
0.39,0.92,-0.30,153.75,-18.55,0.37,-6.21,-15.71,23.00
-0.16,-1.15,-0.45,-24.17,-51.57,-51.94,-16.20,31.29,9.70
0.16,0.70,0.52,-162.05,-14.34,11.05,-16.25,-10.57,6.09
-0.13,-0.90,0.34,368.29,-10.25,-137.94,-17.02,24.33,1.95
0.10,0.33,0.67,-99.61,7.08,34.91,-19.93,3.16,-2.32
0.17,0.11,0.81,-12.88,1.89,-4.58,-20.63,1.93,-14.48
0.02,-0.04,0.97,1.04,-0.73,-0.12,-25.82,3.49,-13.90
0.02,-0.04,0.97,1.10,-1.04,-0.18,-25.20,2.51,-14.53
0.02,-0.04,0.97,1.04,-1.04,-0.12,-25.21,2.17,-13.78
0.03,-0.04,0.98,0.98,-1.16,-0.12,-25.56,2.22,-14.06
```

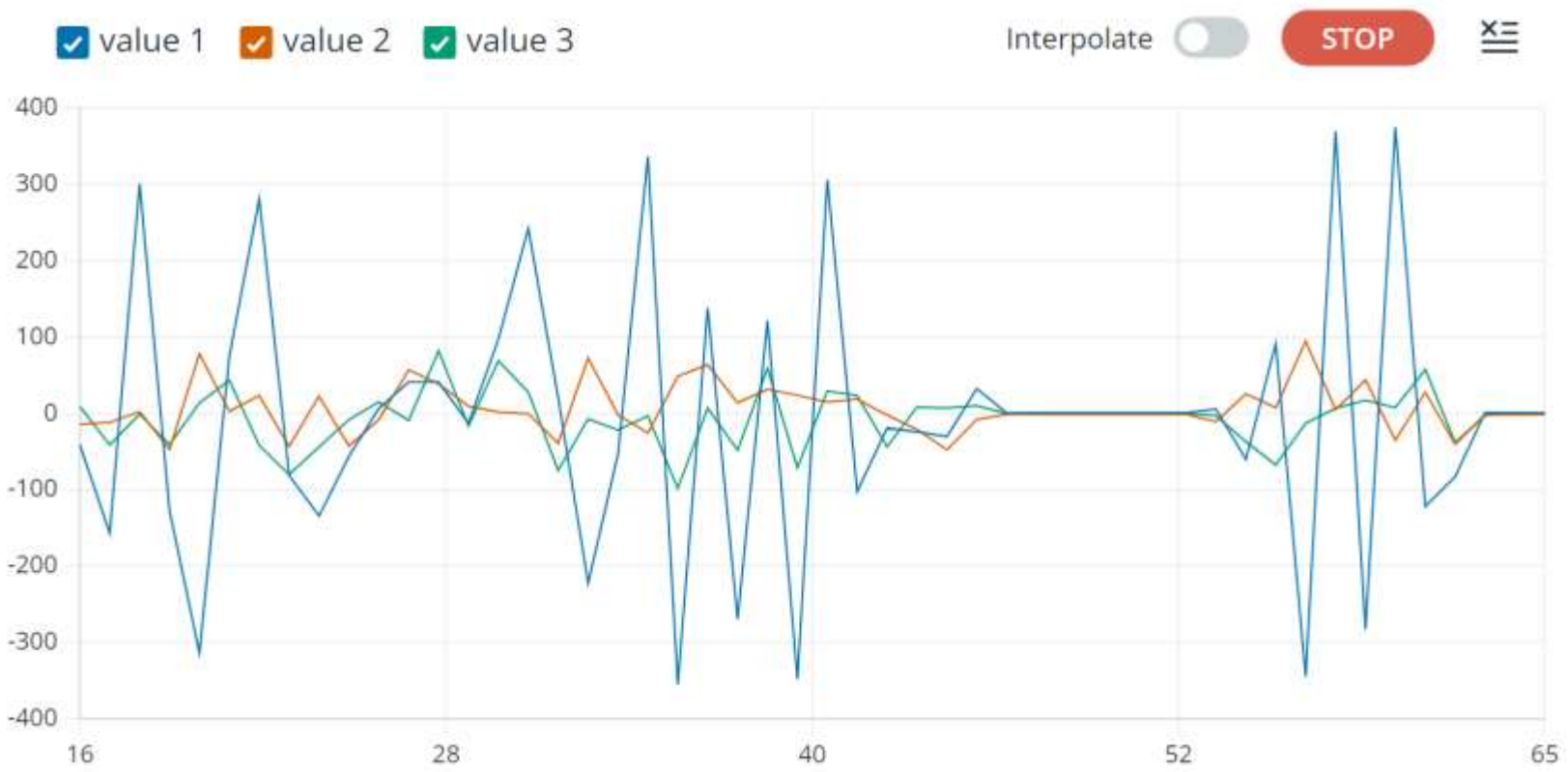
LSM9DS1, 9축 IMU센서: acc, gyro, mag

Accelerometer range is set at [-4, +4]g -/+0.122 mg.

aann_acc.png 로 저장

LSM9DS1, 9축 IMU센서: acc, gyro, mag

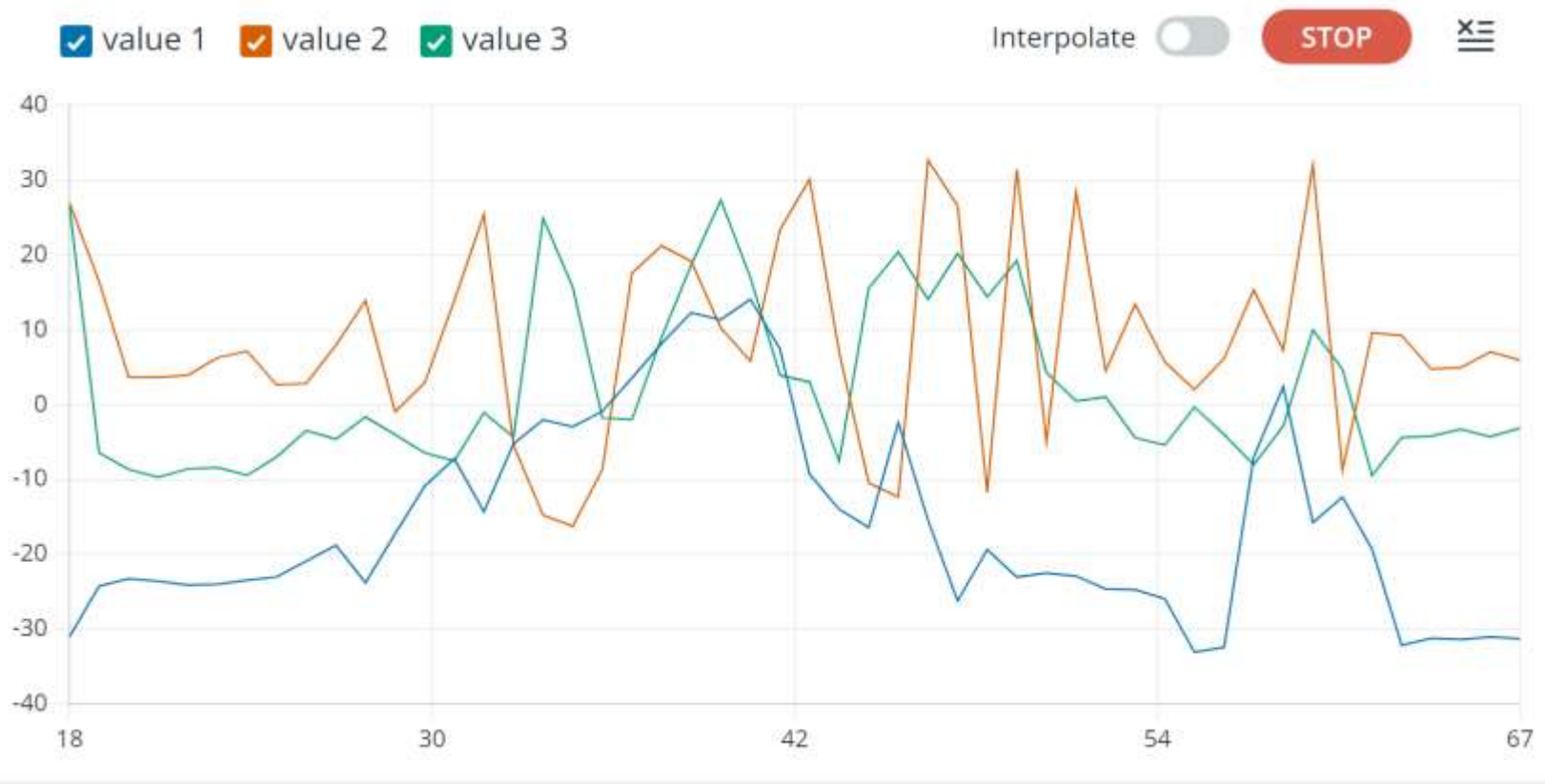
Gyroscope range is set at [-2000, +2000] dps +/-70 mdps.



aann-gyro.png 로 저장

LSM9DS1, 9축 IMU센서: acc, gyro, mag

Magnetometer range is set at [-400, +400] uT +/-0.014 uT.



aann-mag.png 로 저장

IoT Signals from nano33ble IMU sensor

Real-time Signals

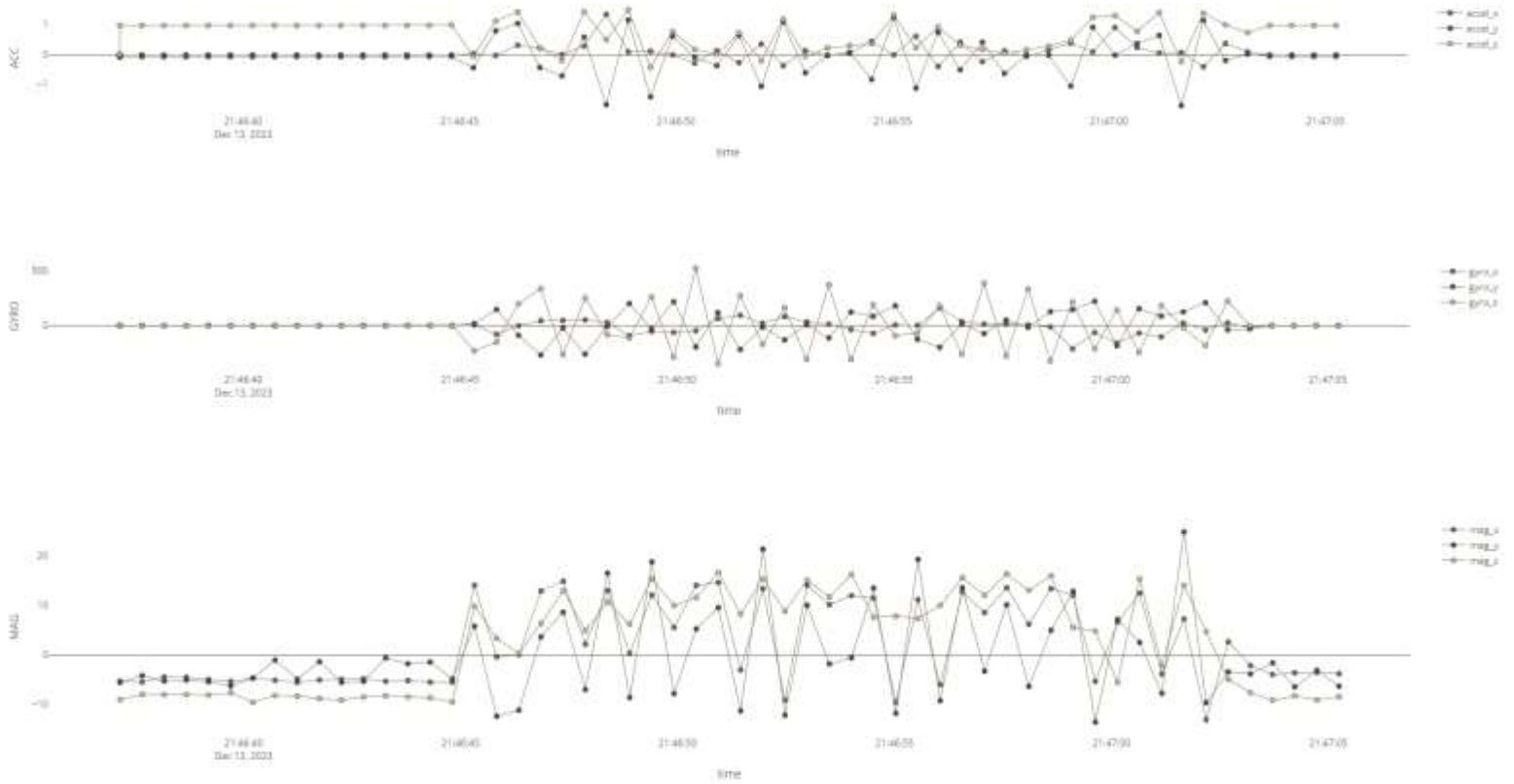
on Time: 2023-12-13 21:49:20.517

IMU signals (ax,ay,az),(gx,gy,gz),(mx,my,mx) : (-0.00,-0.07,0.98) , (1.53,-0.24,-0.06) , (-3.91,-6.34,-7.84)

RMS of IMU signals (ma, mg, mm) : 0.98,1.55,10.81

Real-time IMU from nano 33 BLE sensor

on Time: 2023-12-13 21:47:05.246

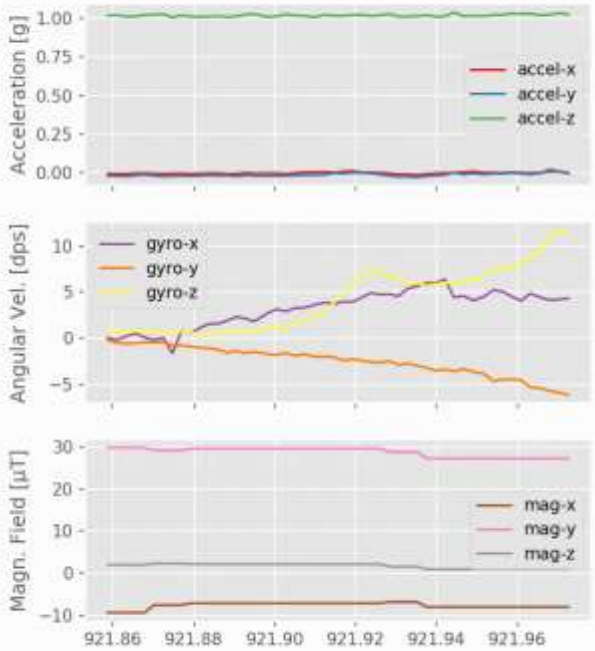


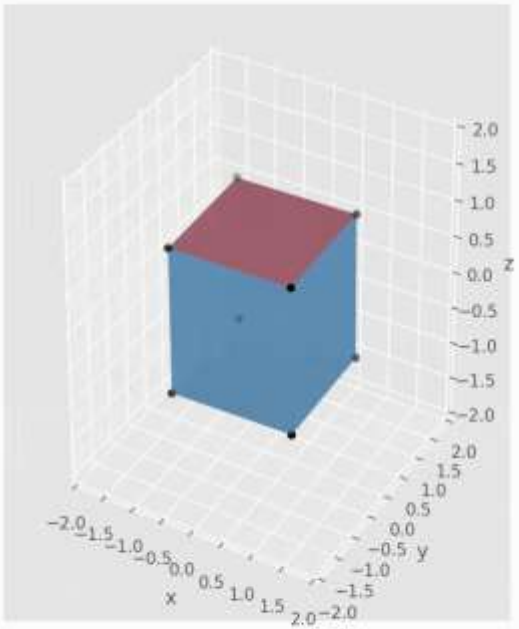
LSM9DS1, 9축 IMU센서: acc, gyro, mag

* 9축 IMU 센서 신호 마이닝 (딥러닝)

1. **MongoDB**
2. **Express server**
   - 실시간 모니터링
   - **DB** 모니터링
3. **data mining using Colab**
4. **Deep learning ?**

모션 인식(9-축 IMU)

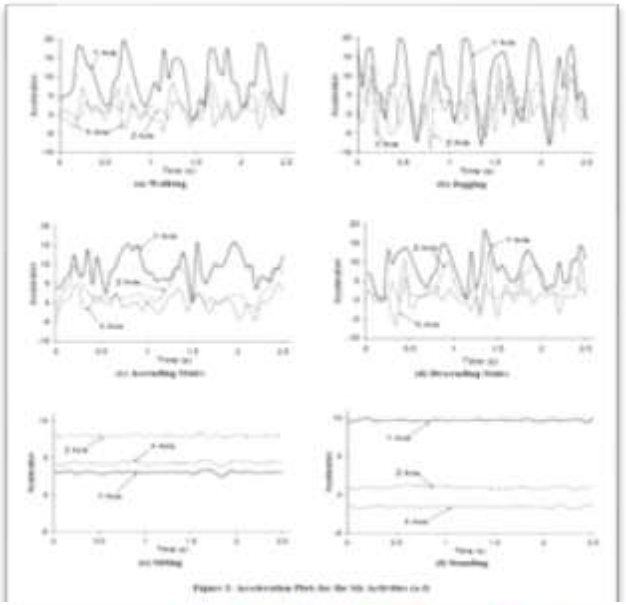


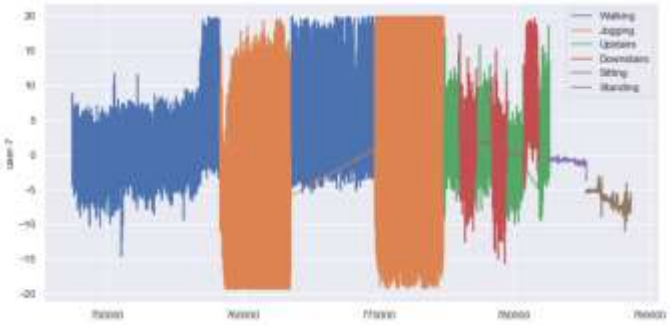


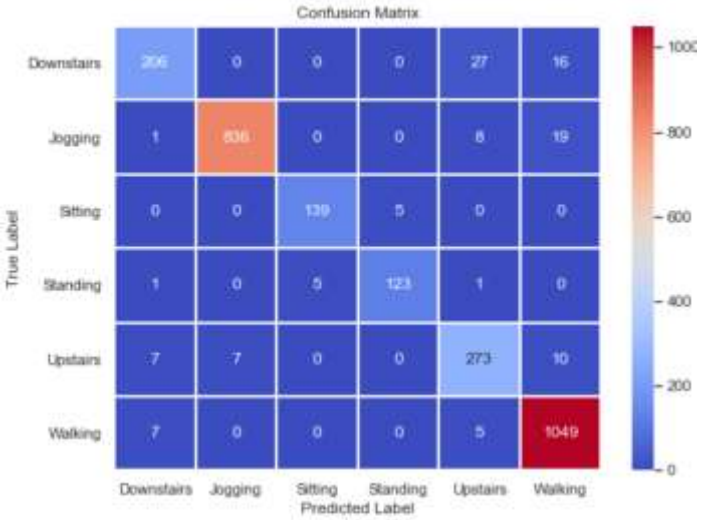
https://images.squarespace-cdn.com/content/v1/59b037304c0dbfb092fbe894/1573836927118-IS5CS61OW9XH9HSRCMA1/ke17ZwdGBTodd8pDm48kGbFogdxZzB1B7PQq3zm9xl7gQa3H78H3Y0txjaiv_0fDoOvxcdMmMKkDsyUqMSsMWxHk725yiiHCCLfrh8O1z5QPOohDIaIeljMHgDF5CVlOqpeNLcJ80NK65_fV7S1UQupMlr7Z9cq9PZkRYtzEu3SbZmkCxOjksrEup4_K2kPH3bqxw7fF48mhrq5Ulr0Hg/mpu9250_cube_rotation_compressed.gif

일상활동 인식(3축 가속도)

['Downstairs',
'Jogging',
'Sitting',
'Standing',
'Upstairs',
'Walking']

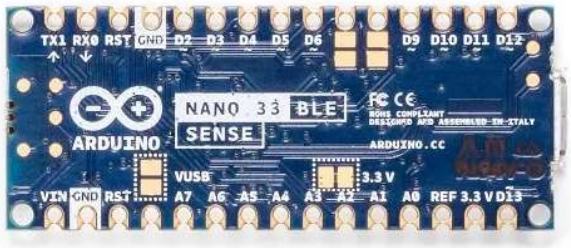






--- ACC_XYZ, 4s: classification report for test data ---

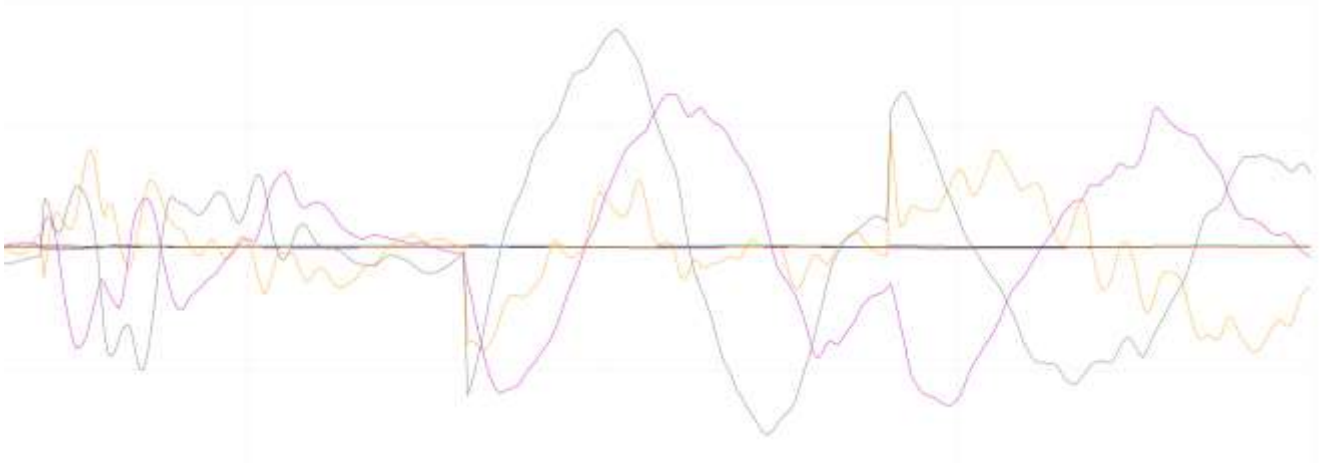
	precision	recall	f1-score	support
0	0.93	0.83	0.87	249
1	0.99	0.97	0.98	864
2	0.97	0.97	0.97	144
3	0.96	0.95	0.95	130
4	0.87	0.92	0.89	297
5	0.96	0.99	0.97	1061
accuracy			0.96	2745
macro avg	0.95	0.94	0.94	2745
weighted avg	0.96	0.96	0.96	2745



Arduino nano33 BLE

Classification of gestures using ACC in Tensorflow 2.x & TinyML/TF-Lite

Data 모으기



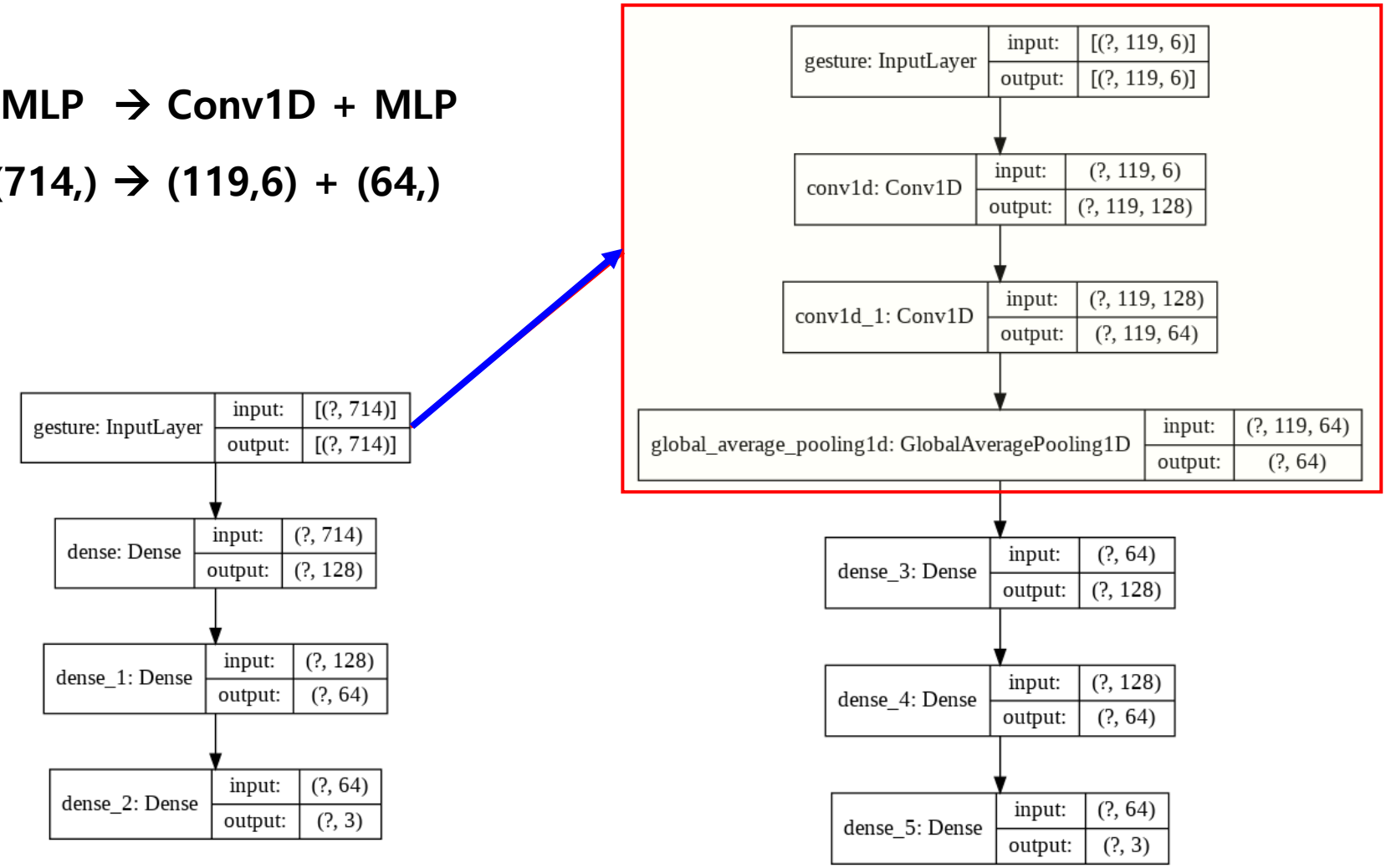
PUNCH CCW CW

119 X 6 (ax,ay,az, gx,gy,gz)

DL architecture

MLP → Conv1D + MLP

(714,) → (119,6) + (64,)



DL-model

```
from tensorflow.keras import layers
```

```
# TF2 functional API
# CONV1D & MLP
inputs = keras.Input(shape=(119,6), name='gesture')
x = layers.Conv1D(128, 3, padding='causal',activation='relu')(inputs)  # 32,
x = layers.Conv1D(64, 3, padding='causal', activation='relu')(x)      # 16
x = layers.GlobalAveragePooling1D()(x)  # New features (714 => 16 or 64)
x = layers.Dense(128, activation='relu')(x)
x = layers.Dense(64, activation='relu')(x)
outputs = layers.Dense(NUM_GESTURES, activation='softmax')(x)

model_conv = keras.Model(inputs=inputs, outputs=outputs, name='gesture_model2')

model_conv.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
```

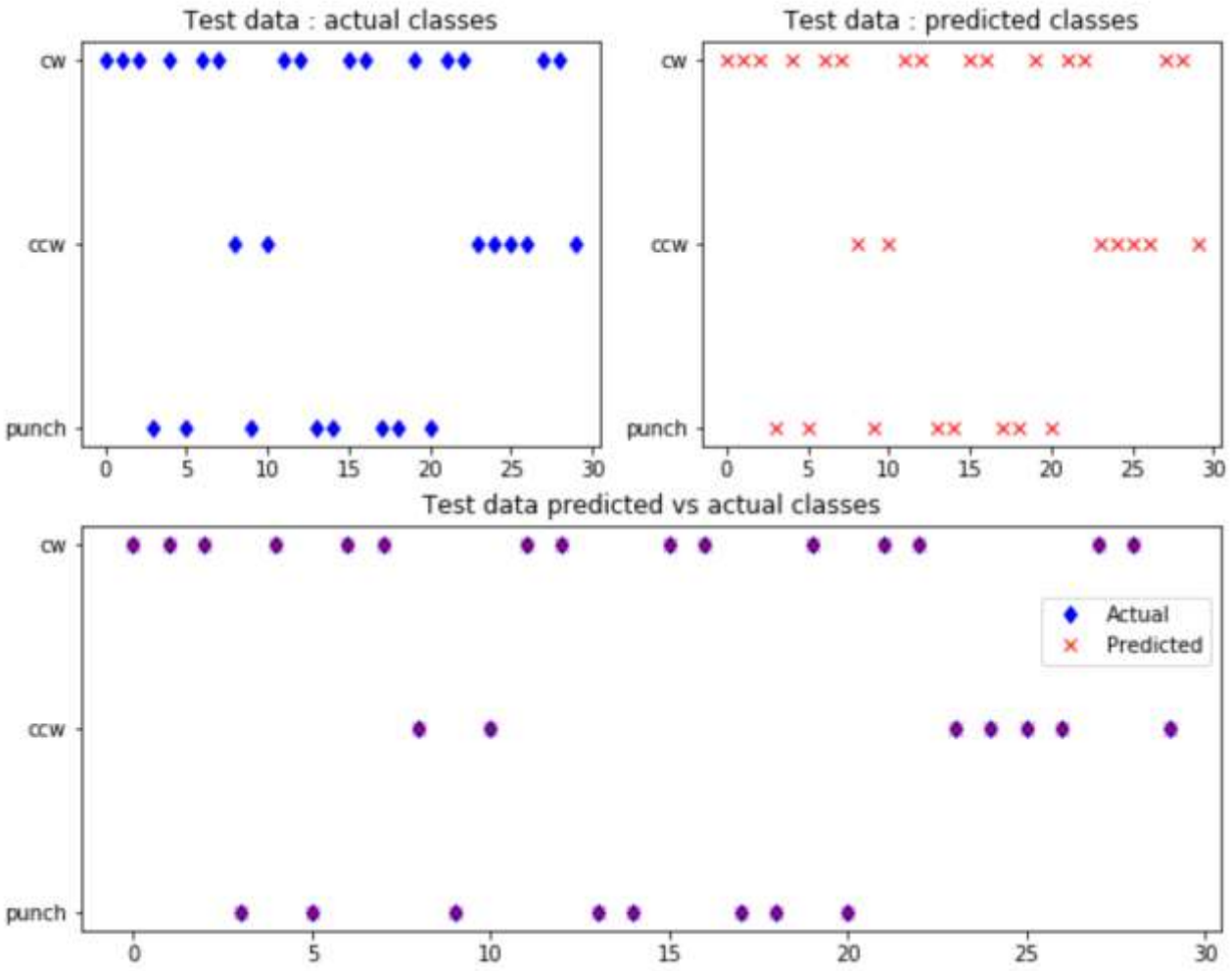
```
# train the model
history = model_conv.fit(inputs_train2, outputs_train, epochs=500, batch_size=16
                         validation_data=(inputs_validate2, outputs_validate))
```

```
model_conv.summary()
```

```
Model: "gesture_model2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gesture (InputLayer)         [(None, 119, 6)]          0
_________________________________________________________________
conv1d (Conv1D)              (None, 119, 128)          2432
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 119, 64)           24640
_________________________________________________________________
global_average_pooling1d (Gl (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               8320
_________________________________________________________________
dense_4 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_5 (Dense)              (None, 3)                 195
=================================================================
Total params: 43,843
Trainable params: 43,843
Non-trainable params: 0
```

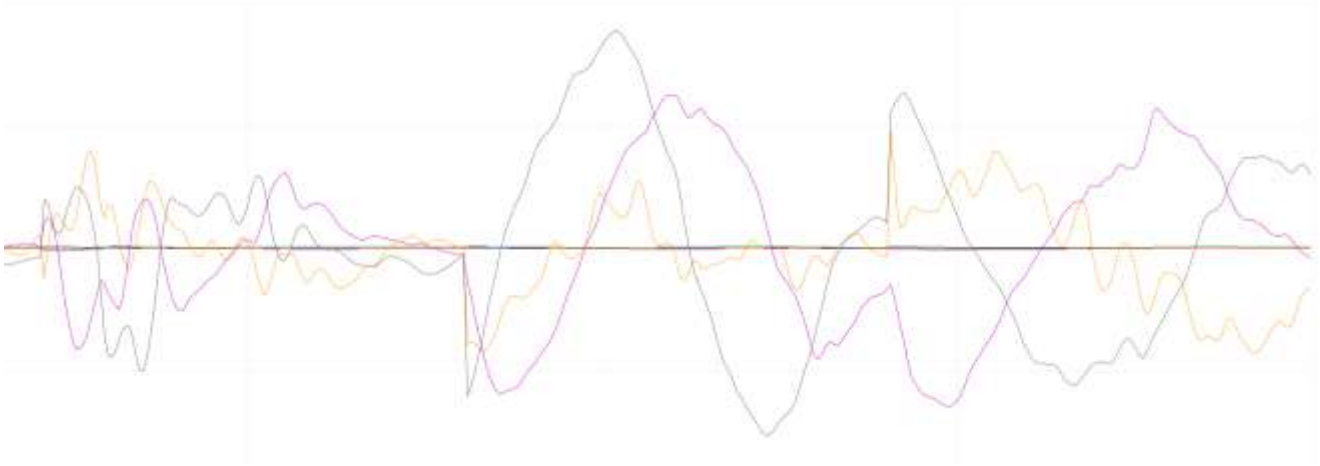
DL-model testing

actual =
[[0. 0. 1.]
[0. 0. 1.]
[0. 0. 1.]
[1. 0. 0.]
[0. 0. 1.]
[1. 0. 0.]
[0. 0. 1.]
[0. 0. 1.]
[0. 1. 0.]
[1. 0. 0.]
[0. 1. 0.]
[0. 0. 1.]
[0. 0. 1.]
[1. 0. 0.]



predictions =
[[0. 0.001 0.999]
[0.001 0.001 0.998]
[0.005 0.001 0.994]
[0.999 0. 0.001]
[0. 0.001 0.999]
[1. 0. 0.]
[0. 0.001 0.999]
[0.002 0.001 0.997]
[0. 1. 0.]
[1. 0. 0.]
[0. 1. 0.]
[0. 0.001 0.999]
[0.001 0. 0.999]
[0.997 0. 0.003]

Real-time testing

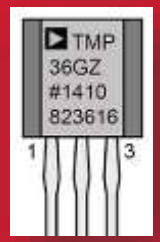


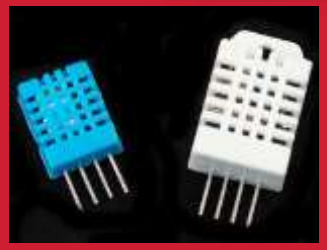
PUNCH CCW CW

COM8

```
13:24:06.140 -> punch: 0.996470
13:24:06.140 -> ccw: 0.000000
13:24:06.140 -> cw: 0.003530
13:24:06.140 ->
13:24:10.197 -> punch: 0.000095
13:24:10.197 -> ccw: 0.000000
13:24:10.197 -> cw: 0.999905
13:24:10.197 ->
13:24:13.193 -> punch: 0.000000
13:24:13.193 -> ccw: 1.000000
13:24:13.193 -> cw: 0.000000
```

[Practice]





- [wk14]
  - IoT Project: nano33ble
  - Multi-sensor circuits : IMU
  - Complete your project
  - Upload folder: aann-rpt13
  - Use repo “aann” in github

wk14 : Practice : aann-rpt13

- [Target of this week]
  - Complete your works
  - Save your outcomes and upload outputs in github

제출폴더명 : **aann-rpt13**

- 제출할 파일들

① **LSM9DS1_Basic.ino**

② **LSM9DS1_IoT.ino**

③ **aann_acc.png**

④ **aann_gyro.png**

⑤ **aann_mag.png**

Lecture materials

References & good sites

- ✓ http://www.arduino.cc Arduino Homepage
- ✓ http://www.nodejs.org/ko Node.js
- ✓ https://plot.ly/ plotly
- ✓ https://www.mongodb.com/ MongoDB
- ✓ http://www.w3schools.com By w3schools
- ✓ http://www.github.com GitHub

Target of this class