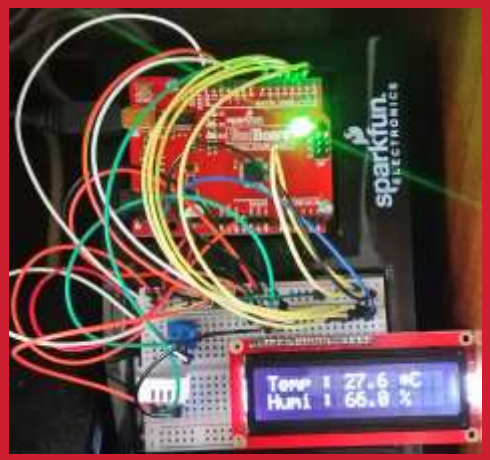


Arduino-IOT

[wk13]

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, 2022

Email : chaos21c@gmail.com

My ID

ID를 확인하고 github에 repo 만들기

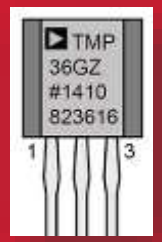
AA01	강대진	AA13	박제홍
		AA14	심준혁
AA03	김성우	AA15	이상혁
AA04	김정헌	AA16	이승무
		AA17	이승준
AA06	김창연	AA18	이준희
AA07	김창욱	AA19	이현준
AA08	김태화	AA20	임태형
AA09	남승현	AA21	정동현
AA10	류재환		
AA11	박세훈	AA23	정희서
AA12	박신영	AA24	최재형

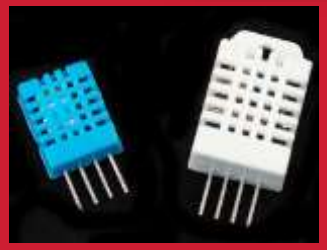
위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, README.md check

[Practice]





- [wk12]
  - IoT Project: nano33ble
  - Multi-sensor circuits
  - Complete your project
  - Upload folder: aann-rpt12
  - Use repo “aann” in github

wk12 : 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_nano33_csv.ipynb** in data_mining folder

② **iot_nano33_json.ipynb** in data_mining folder

③ **All *.js** in nano33 folder

④ **public/All *.html**

⑤ **aann_Iot33_1000.csv** in public/data folder

Email : chaos21c@gmail.com

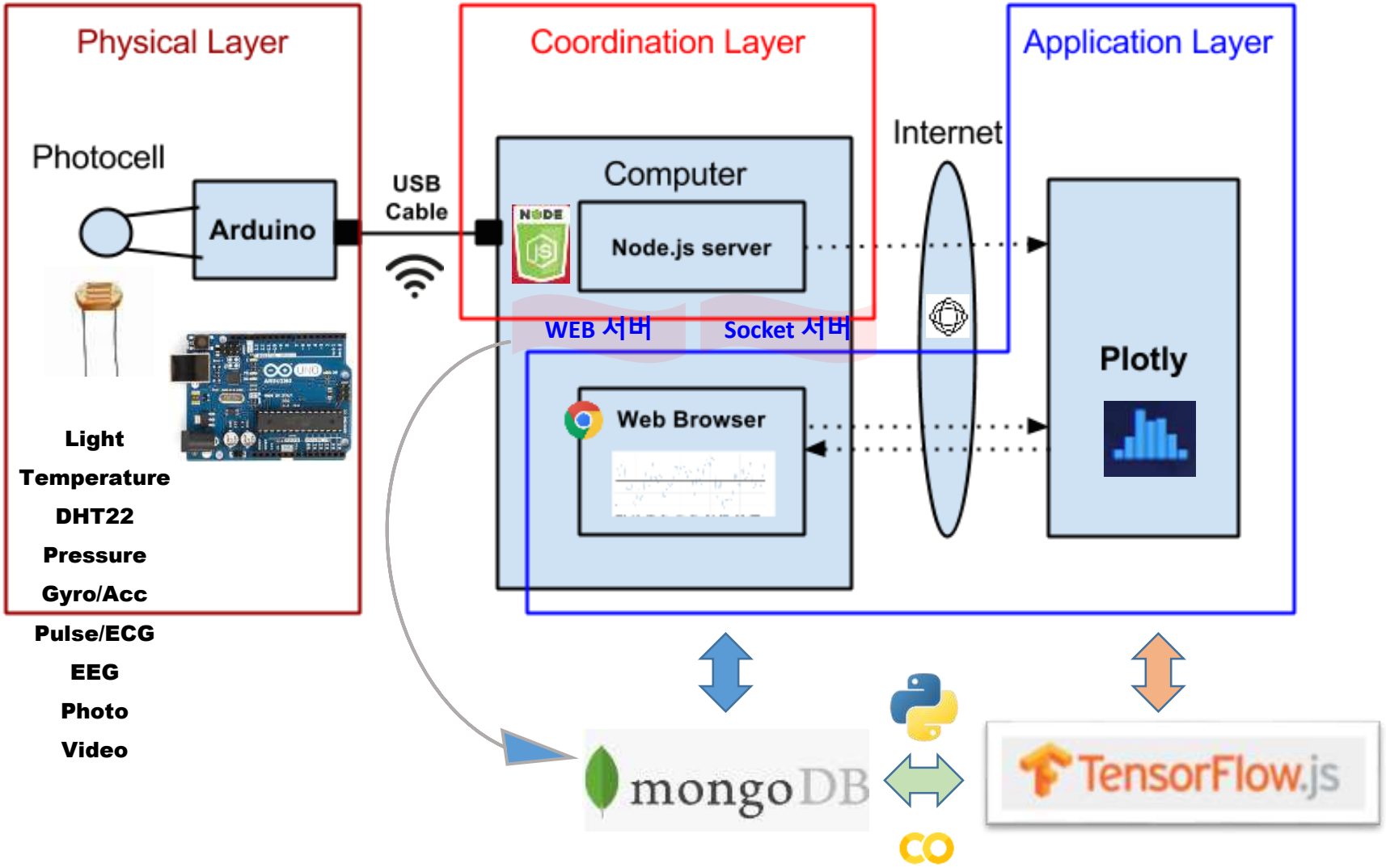
Purpose of AA

주요 수업 목표는 다음과 같다.

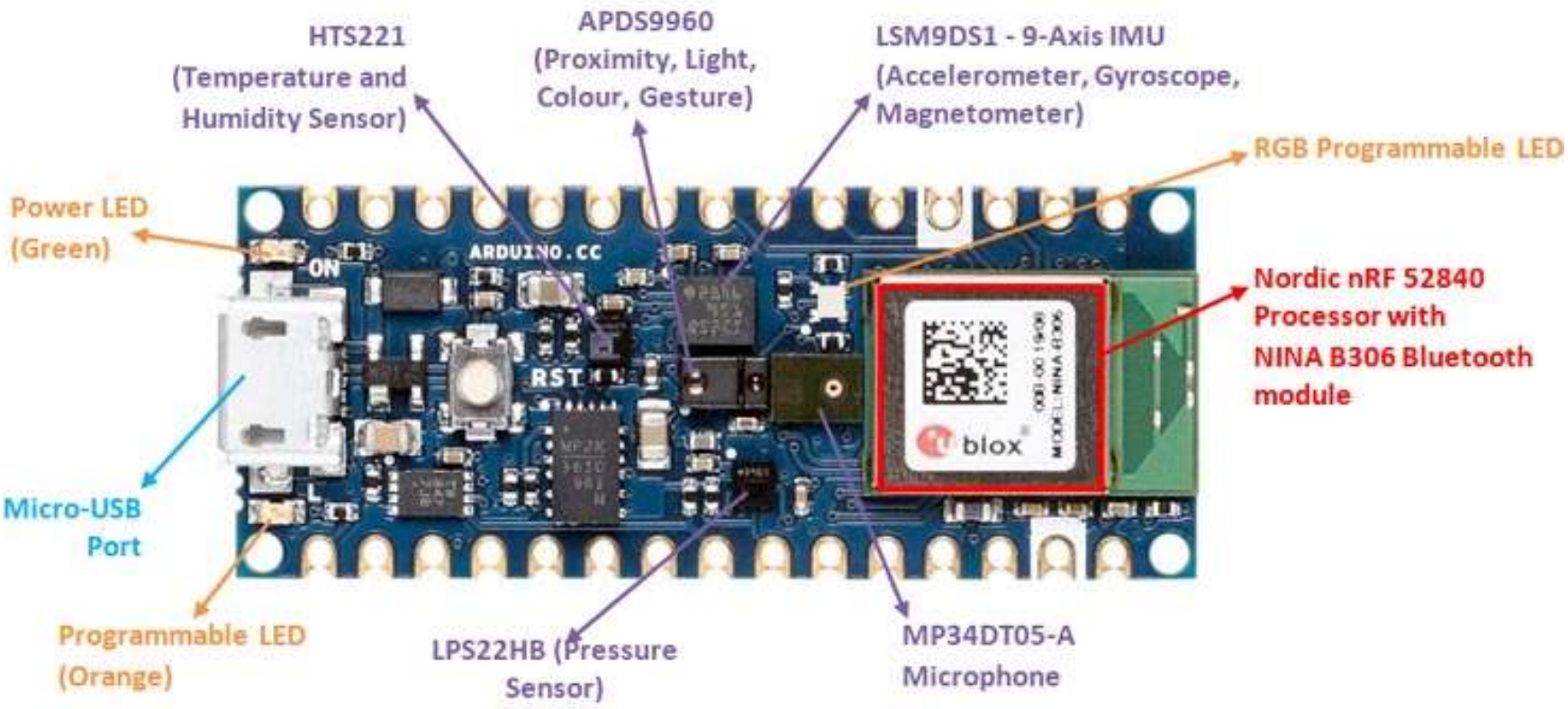
1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]

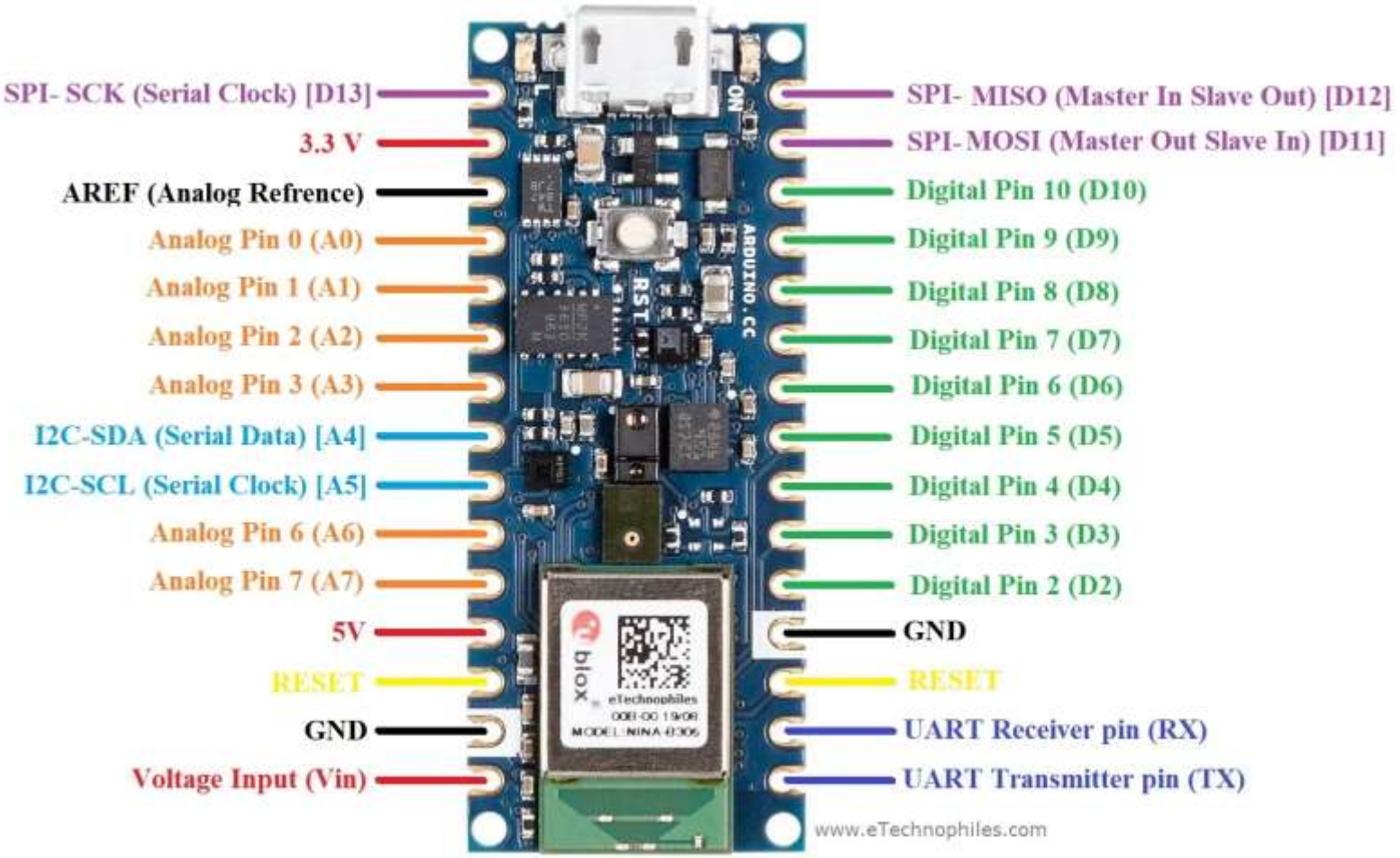

Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

nano33BLE sensor



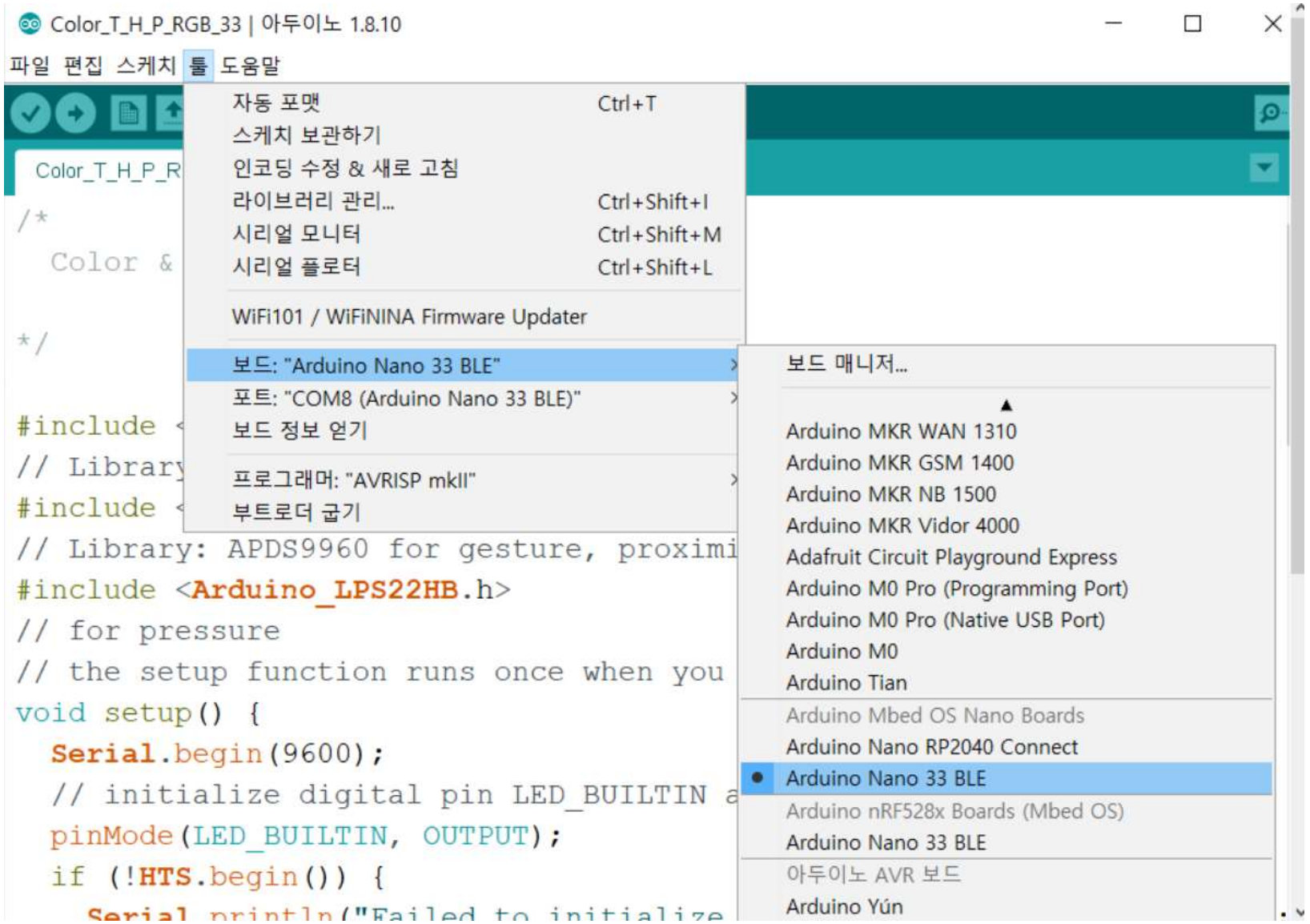
이미지 출처: https://circuitdigest.com/microcontroller-projects/arduino-nano-33-ble-sense-board-review-and-getting-started-guide

nano33BLE sensor - pins

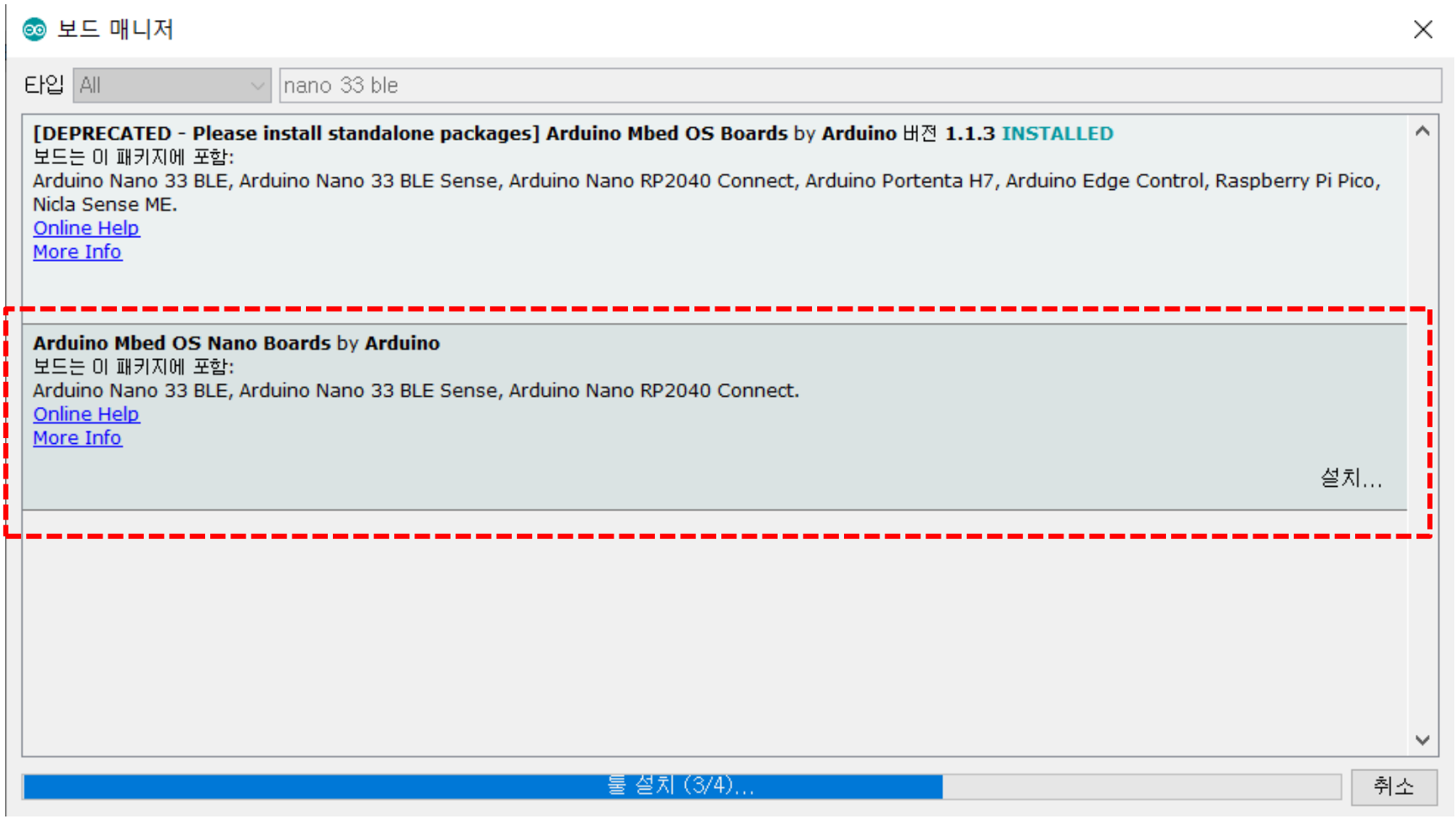


이미지 출처: https://www.etechnophiles.com/arduino-nano-33-ble-sense-pinout-introduction-specifications/

Layout [H S C]

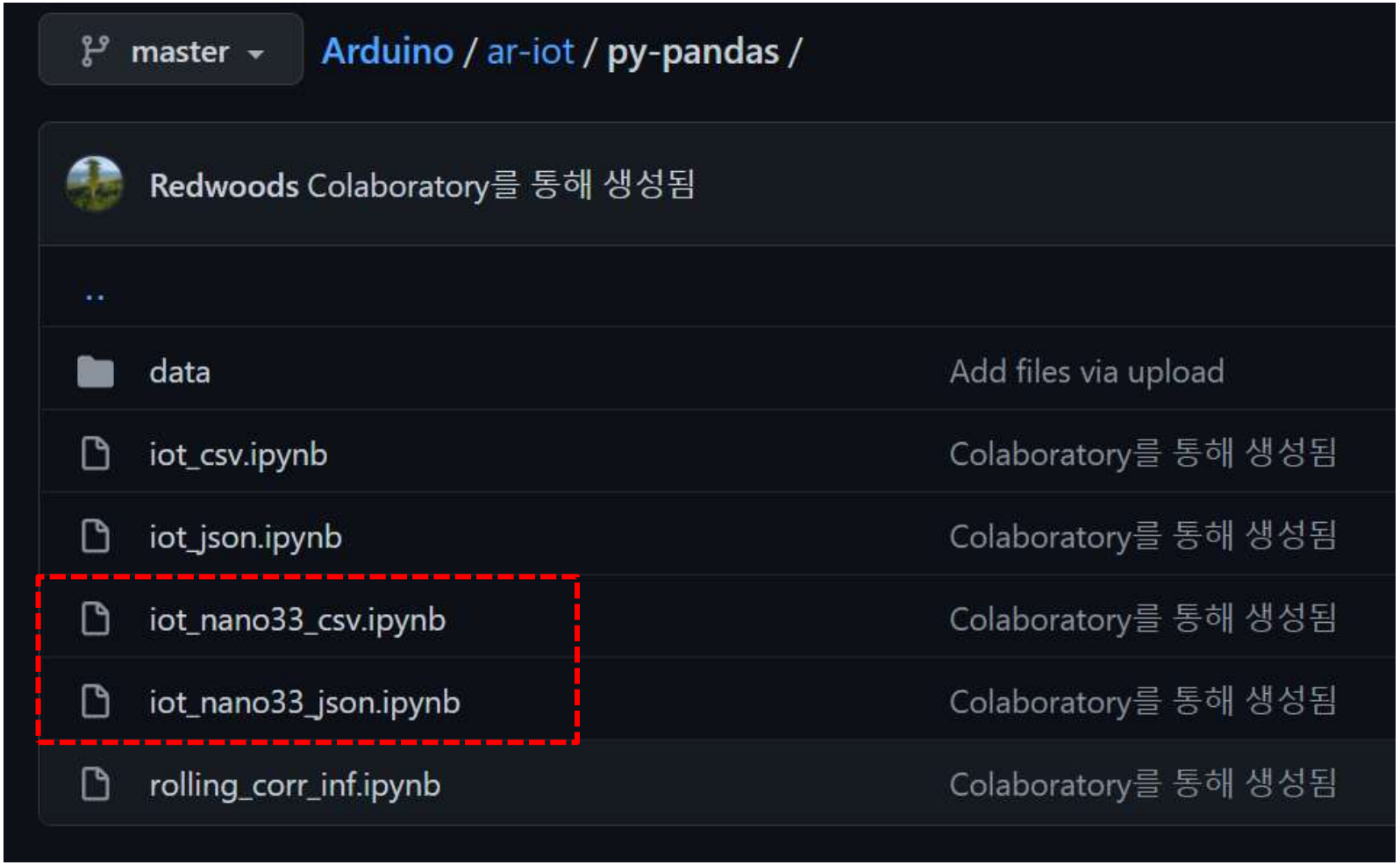

Color_T_H_P_RGB_33 | 아두이노 1.8.10
파일 편집 스케치 툴 도움말
자동 포맷 Ctrl+T
스케치 보관하기
인코딩 수정 & 새로 고침
라이브러리 관리... Ctrl+Shift+I
시리얼 모니터 Ctrl+Shift+M
시리얼 플로터 Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
보드: "Arduino Nano 33 BLE"
포트: "COM8 (Arduino Nano 33 BLE)"
보드 정보 얻기
프로그래머: "AVRISP mkII"
부트로더 굽기
보드 매니저...
Arduino MKR WAN 1310
Arduino MKR GSM 1400
Arduino MKR NB 1500
Arduino MKR Vidor 4000
Adafruit Circuit Playground Express
Arduino M0 Pro (Programming Port)
Arduino M0 Pro (Native USB Port)
Arduino M0
Arduino Tian
Arduino Mbed OS Nano Boards
Arduino Nano RP2040 Connect
Arduino Nano 33 BLE
Arduino nRF528x Boards (Mbed OS)
Arduino Nano 33 BLE
아두이노 AVR 보드
Arduino Yún
Color_T_H_P_R
/*
Color &
*/
#include
// Library
#include
// Library: APDS9960 for gesture, proximi
#include <Arduino_LPS22HB.h>
// for pressure
// the setup function runs once when you
void setup() {
Serial.begin(9600);
// initialize digital pin LED_BUILTIN a
pinMode(LED_BUILTIN, OUTPUT);
if (!HTS.begin()) {

nano33 board



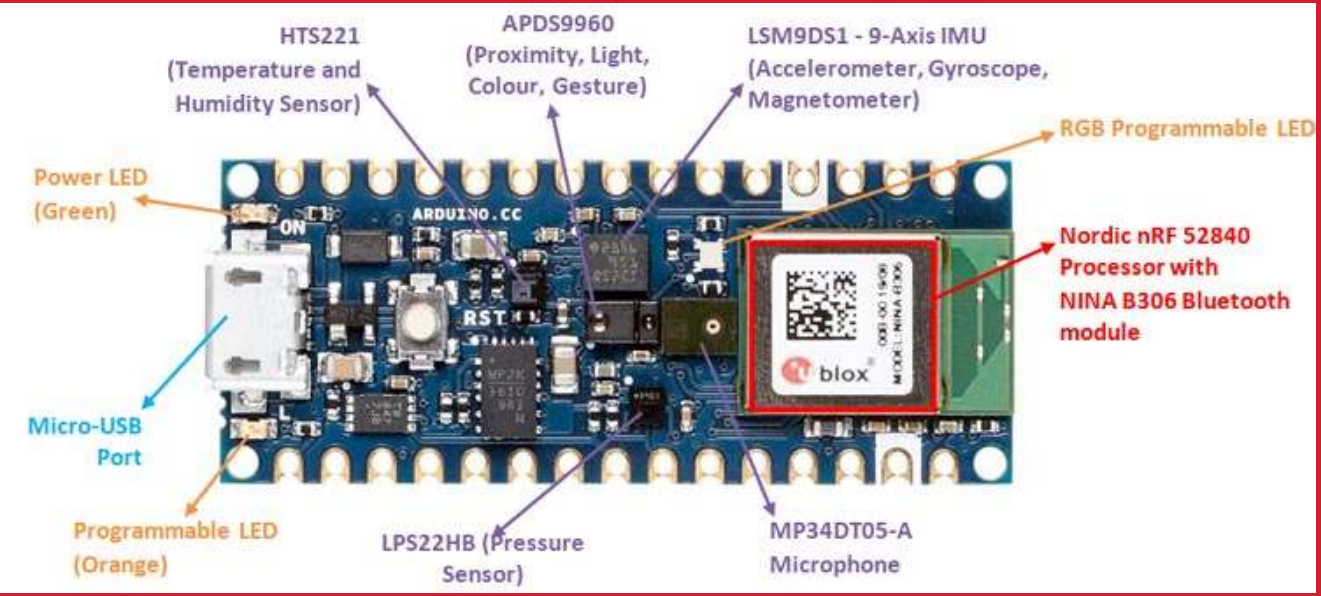
Project: nano33BLE sensor

[2022-project] IoT data mining in Colab



2021/22 AA project

nano33BLE sensor



APDS9960

- gestures
- proximity
- color, light intensity

begin()

end()

gestureAvailable()

readGesture()

colorAvailable()

readColor()

proximityAvailable()

readProximity()

setGestureSensitivity()

setInterruptPin()

setLEDBoost()

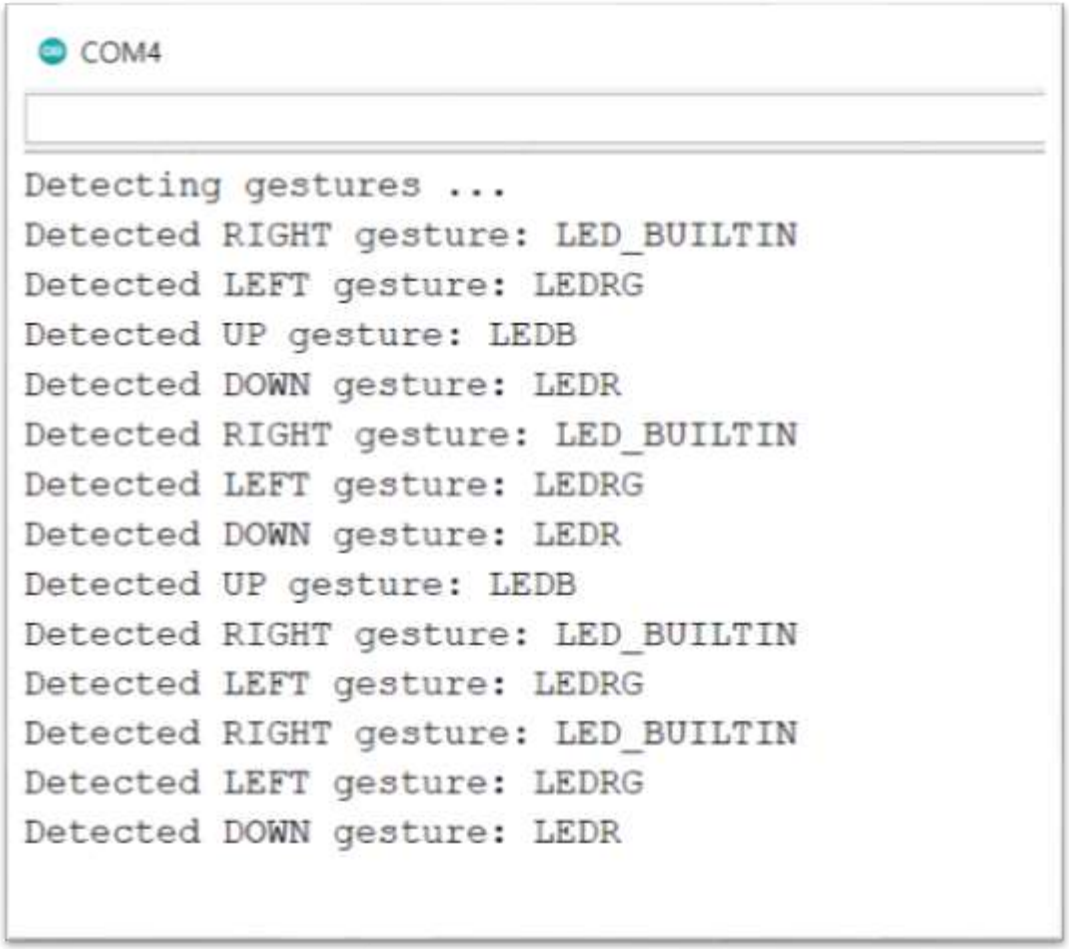
Arduino: gesture.ino

```
15  #include <Arduino_APDS9960.h>
16
17  void setup() {
18    Serial.begin(9600);
19    //in-built LED
20    pinMode(LED_BUILTIN, OUTPUT);
21    //Red
22    pinMode(LEDR, OUTPUT);
23    //Green
24    pinMode(LEDG, OUTPUT);
25    //Blue
26    pinMode(LEDB, OUTPUT);
27
28    while (!Serial);
29    if (!APDS.begin()) {
30      Serial.println("Error initializing APDS9960 sensor!");
31    }
32    // for setGestureSensitivity(..) a value between 1 and 100 is required.
33    // Higher values makes the gesture recognition more sensible but less accurate
34    // (a wrong gesture may be detected). Lower values makes the gesture recognition
35    // more accurate but less sensible (some gestures may be missed).
36    // Default is 80
37    //APDS.setGestureSensitivity(80);
38    Serial.println("Detecting gestures ...");
39    // Turining OFF the RGB LEDs
40    digitalWrite(LEDR, HIGH);
41    digitalWrite(LEDG, HIGH);
42    digitalWrite(LEDB, HIGH);
43  }
```

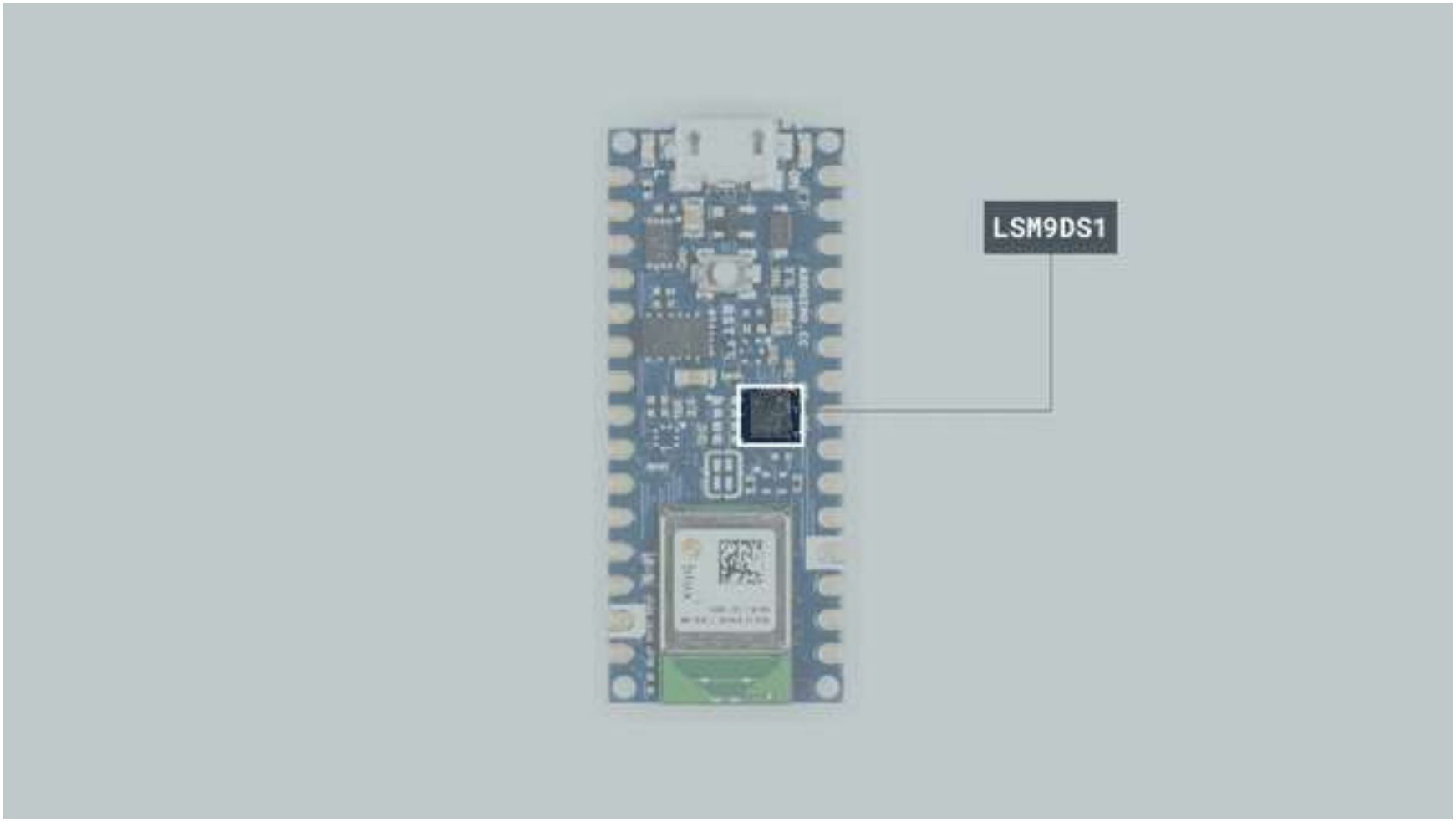
Arduino: gesture.ino

```
44  void loop() {
45    if (APDS.gestureAvailable()) {
46      // a gesture was detected, read and print to serial monitor
47      int gesture = APDS.readGesture();
48      switch (gesture) {
49        case GESTURE_UP:
50          Serial.println("Detected UP gesture");
51          digitalWrite(LEDR, LOW);
52          delay(1000);
53          digitalWrite(LEDR, HIGH);
54          break;
55        case GESTURE_DOWN:
56          Serial.println("Detected DOWN gesture");
57          digitalWrite(LEDG, LOW);
58          delay(1000);
59          digitalWrite(LEDG, HIGH);
60          break;
61        case GESTURE_LEFT:
62          Serial.println("Detected LEFT gesture");
63          digitalWrite(LEDB, LOW);
64          delay(1000);
65          digitalWrite(LEDB, HIGH);
66          break;
67        case GESTURE_RIGHT:
68          Serial.println("Detected RIGHT gesture");
69          digitalWrite(LED_BUILTIN, HIGH);
70          delay(1000);
71          digitalWrite(LED_BUILTIN, LOW);
72          break;
73        default:
74          break;
75      }
76    }
77  }
```

Arduino: serial monitor



LSM9DS1, 9축 IMU센서: acc, gyro, mag



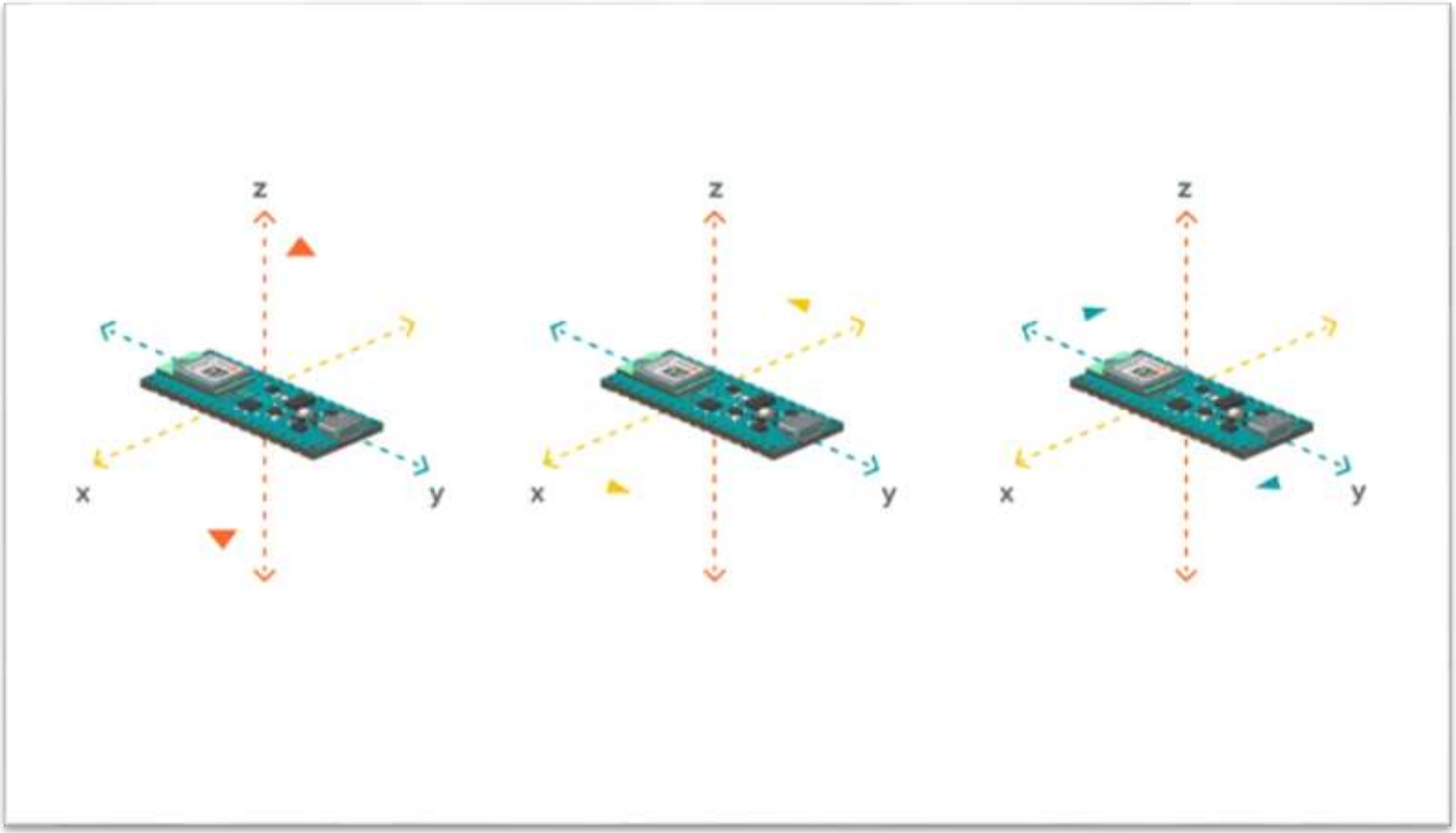
LSM9DS1, 9축 IMU센서: acc, gyro, mag

The LSM9DS1 Library

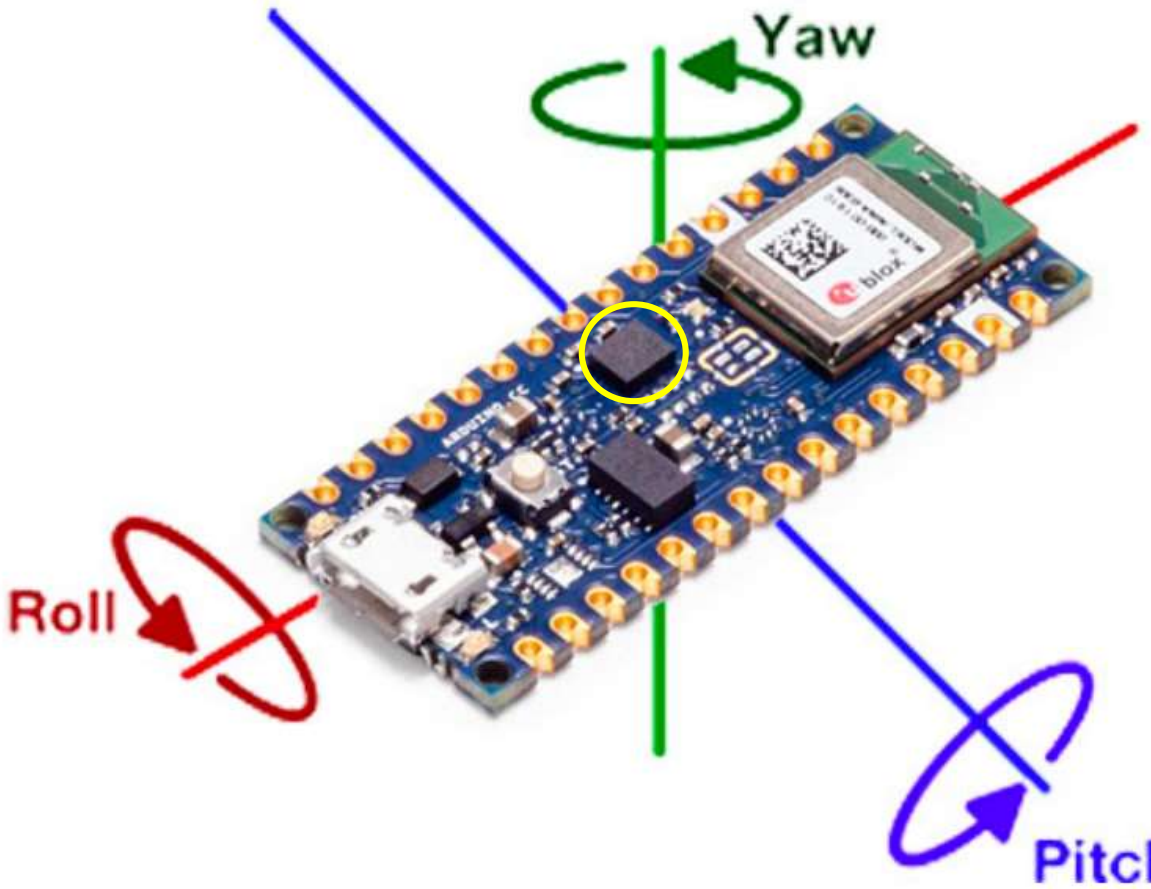
The Arduino LSM9DS1 library allows us to use the Arduino Nano 33 BLE IMU module without having to go into complicated programming. The library takes care of the sensor initialization and sets its values as follows:

- **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

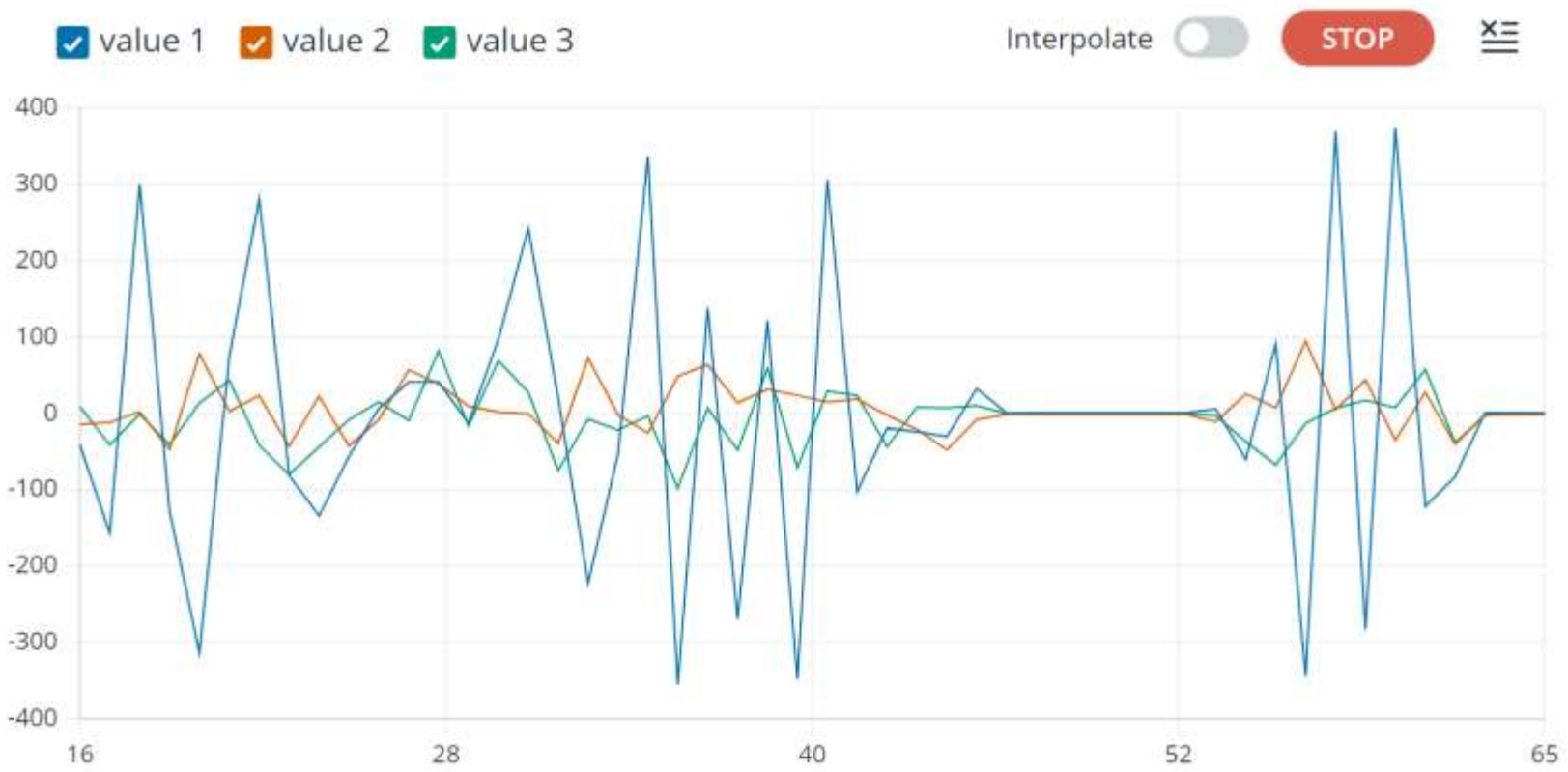
```
-0.02,0.76,0.48,108.58,-7.26,15.08,-19.34,-10.67,9.59
-0.43,-1.02,-0.49,103.88,-49.99,-3.30,-26.07,25.98,18.08
0.10,0.82,0.74,-228.39,49.38,-27.10,-16.94,-10.21,5.30
0.38,-0.11,0.33,70.62,112.61,-47.85,-9.86,9.01,-9.19
1.12,0.27,0.61,-91.31,-158.75,49.87,1.83,-3.25,-3.11
0.39,0.92,-0.30,153.75,-18.55,0.37,-6.21,-15.71,23.00
-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.

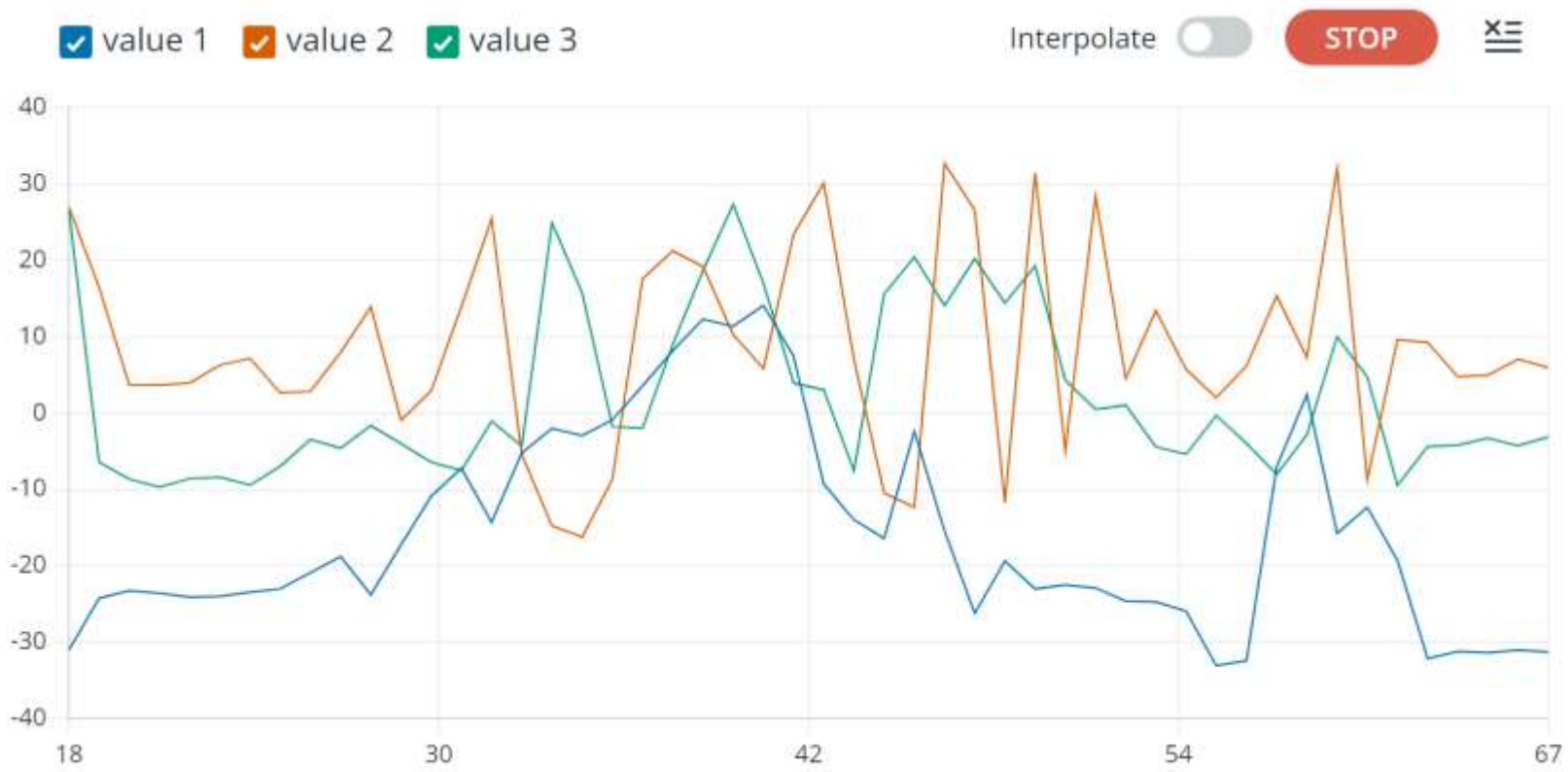
LSM9DS1, 9축 IMU센서: acc, gyro, mag

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



LSM9DS1, 9축 IMU센서: acc, gyro, mag

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

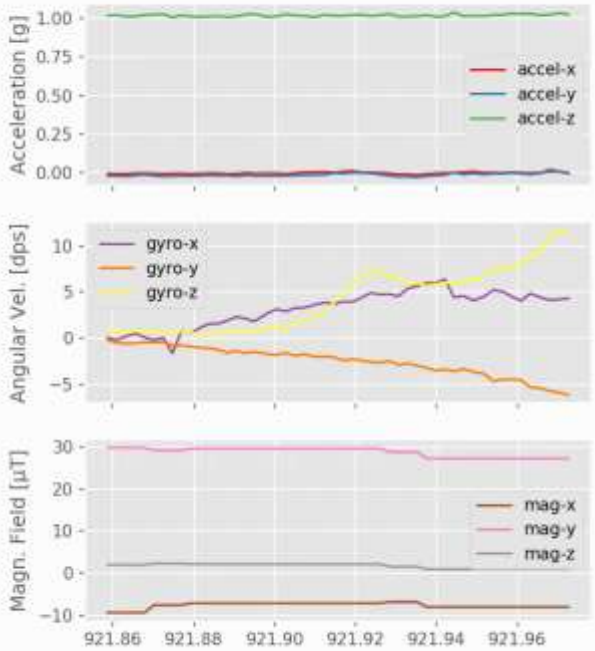


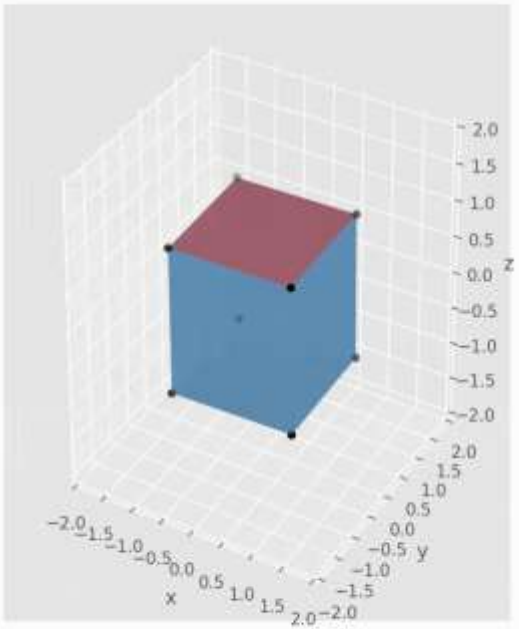
LSM9DS1, 9축 IMU센서: acc, gyro, mag

* 9축 IMU 센서 신호 마이닝

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

모션 인식(9-축 IMU)

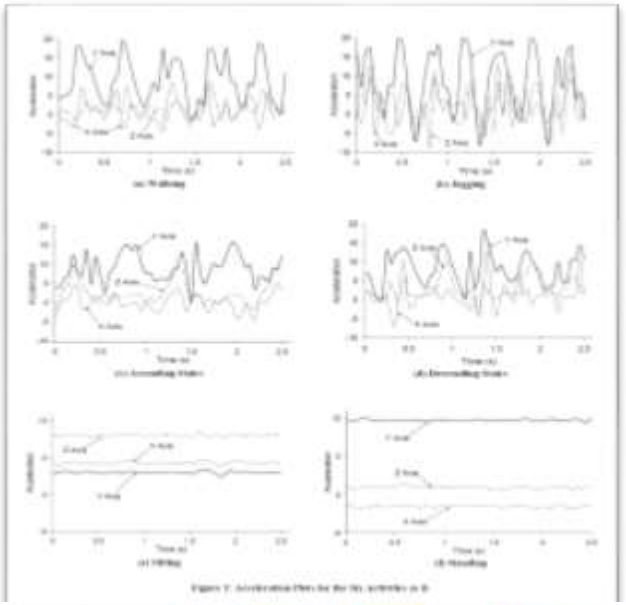


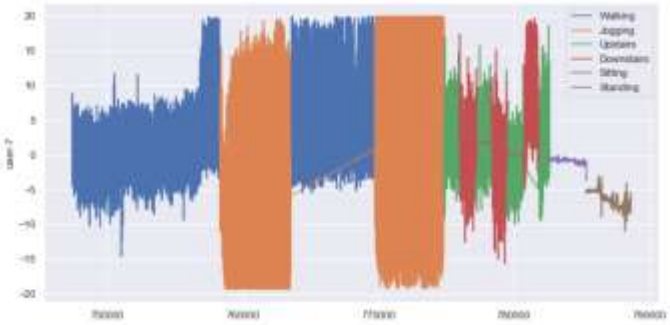


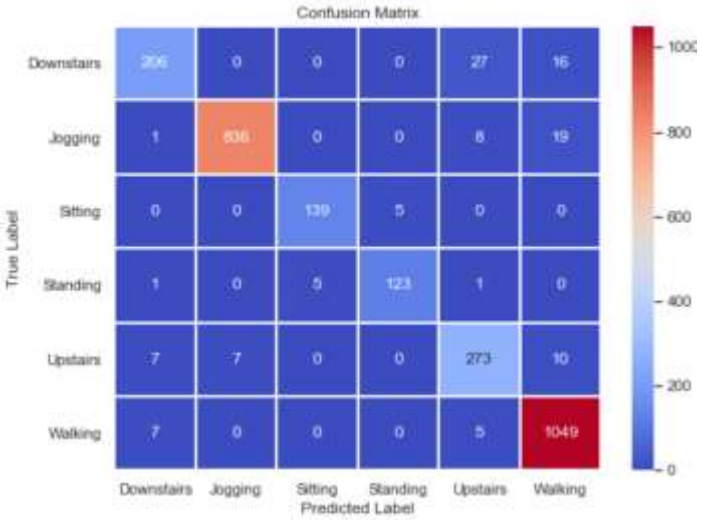
https://images.squarespace-cdn.com/content/v1/59b037304c0dbfb092fbe894/1573836927118-IS5CS61OW9XH9HSRCMA1/ke17ZwdGBTodd8pDm48kGbFogdxZzB1B7PQq3zm9xl7gQa3H78H3Y0txjaiv_0fDoOvxcdMmMKkDsyUqMSsMWxHk725yiiHCCLfrh8O1z5QPOohDIaIeljMHgDF5CVlOqpeNLcJ80NK65_fV7S1UQupMlr7Z9cq9PZkRYtzEu3SbZmkCxOjksrEup4_K2kPH3bqxw7fF48mhrq5Ulr0Hg/mpu9250_cube_rotation_compressed.gif

일상활동 인식(3축 가속도)

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

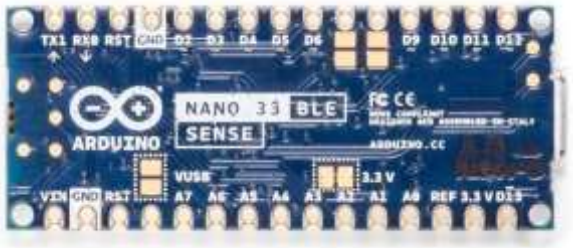






--- ACC_XYZ, 4s: classification report for test data ---

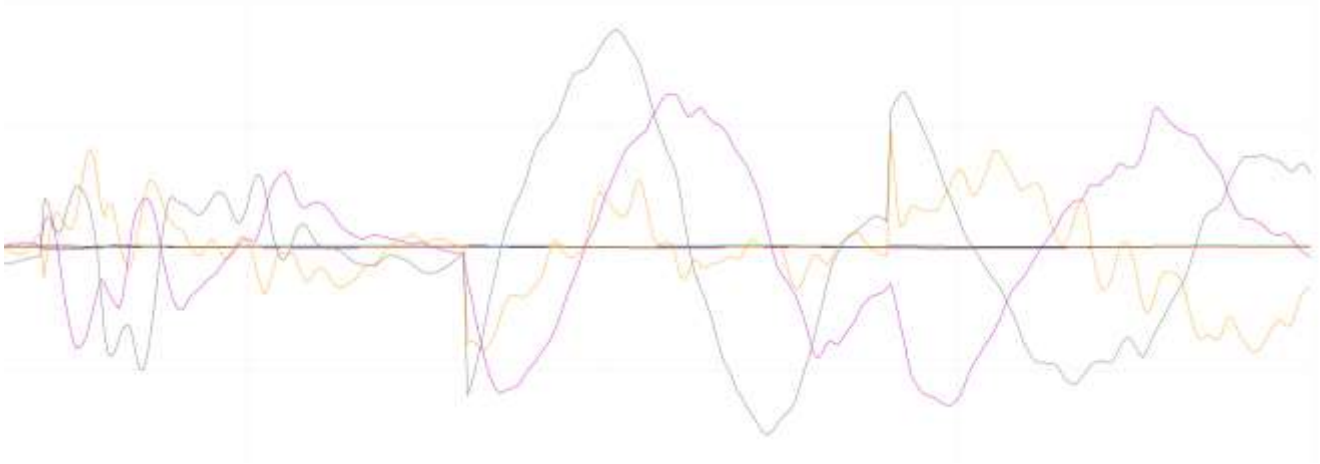
	precision	recall	f1-score	support
0	0.93	0.83	0.87	249
1	0.99	0.97	0.98	864
2	0.97	0.97	0.97	144
3	0.96	0.95	0.95	130
4	0.87	0.92	0.89	297
5	0.96	0.99	0.97	1061
accuracy			0.96	2745
macro avg	0.95	0.94	0.94	2745
weighted avg	0.96	0.96	0.96	2745



Arduino nano33 BLE

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

Data 모으기



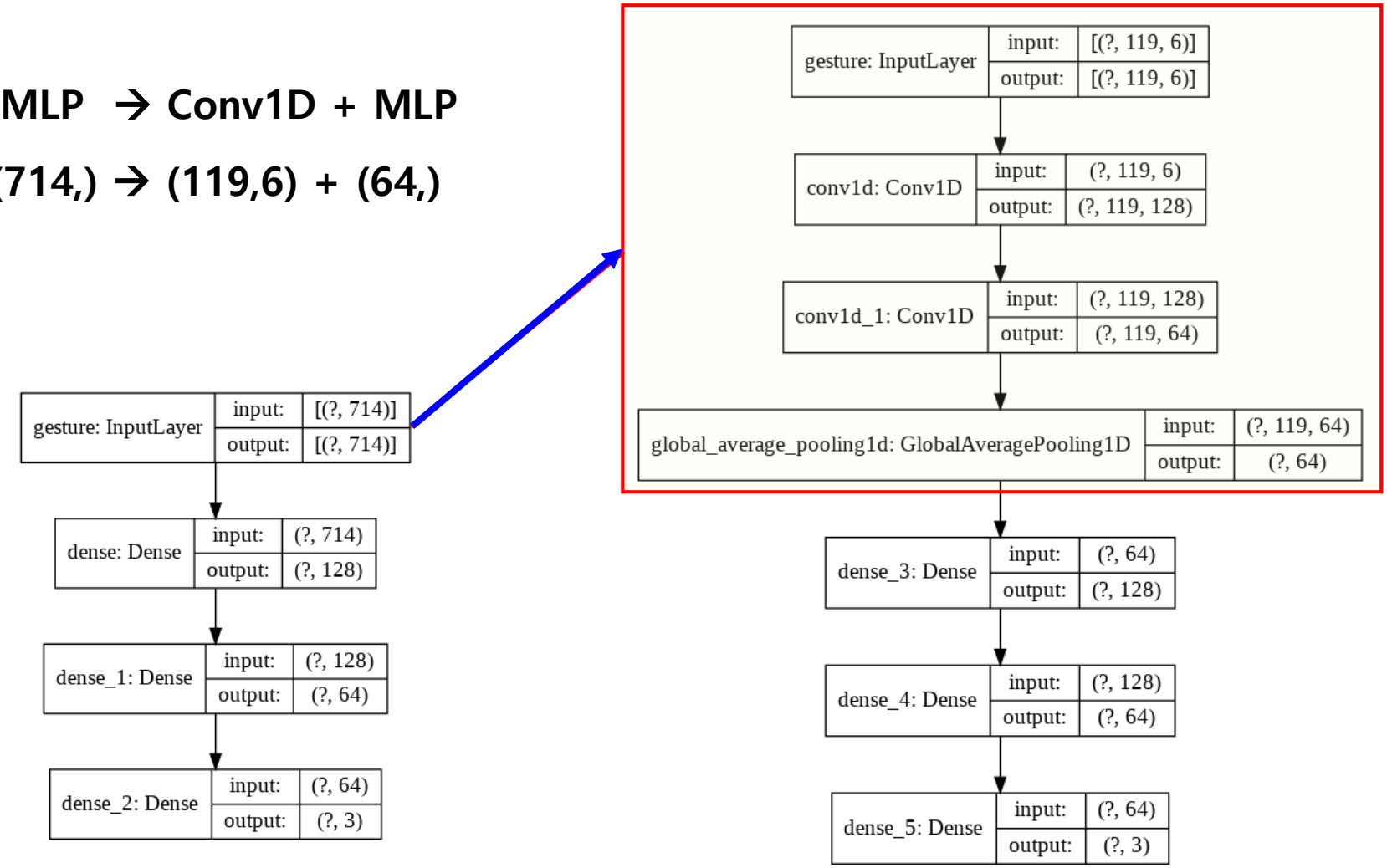
PUNCH CCW CW

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'])
```

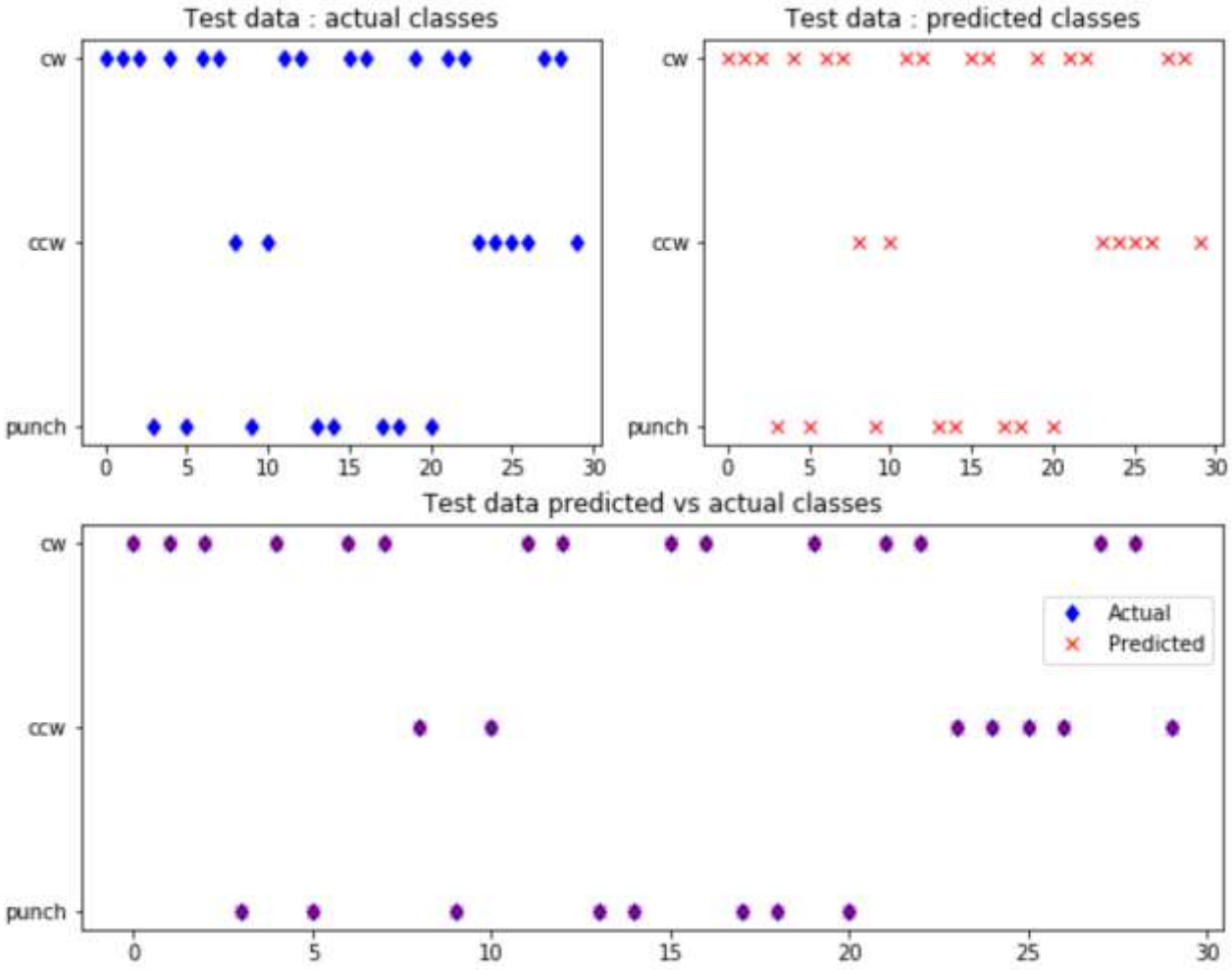
```
# train the model
history = model_conv.fit(inputs_train2, outputs_train, epochs=500, batch_size=16
                         validation_data=(inputs_validate2, outputs_validate))
```

```
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
```

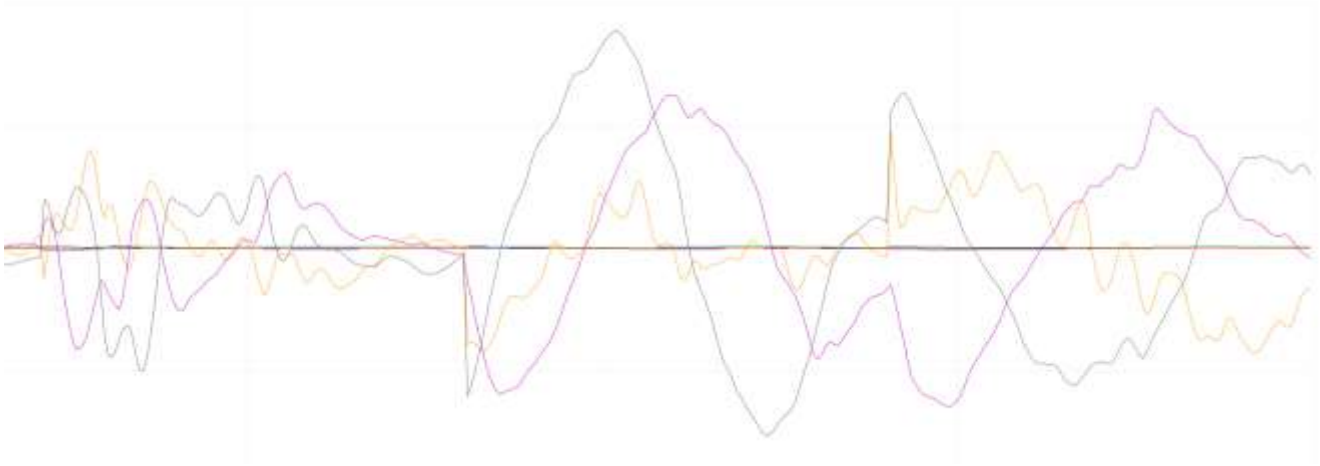
DL-model testing

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

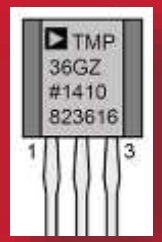


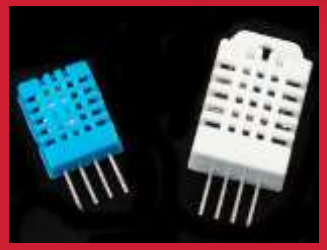
PUNCH CCW CW

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]





- [wk13]
  - IoT Project: nano33ble
  - Multi-sensor circuits : IMU
  - Complete your project
  - Upload folder: aann-rpt13
  - Use repo “aann” in github

wk13 : Practice : aann-rpt13

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

제출폴더명 : **aann-rpt13**

- 제출할 파일들

① **gesture.ino**

② **LSM9DS1_Basic.ino**

③ **LSM9DS1_IoT.ino**

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