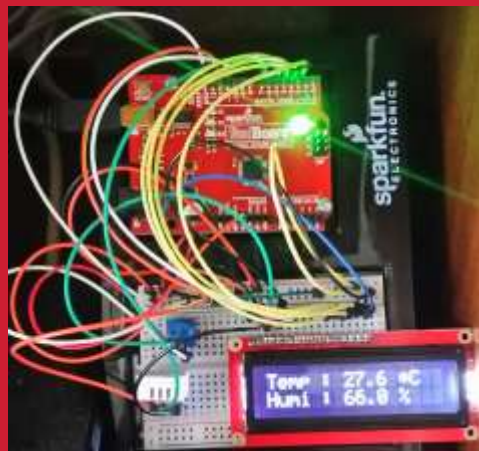# Arduino-IOT

[wk14]

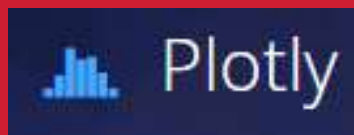## nano33BLE Sensor
IoT Project - IMU

Visualization of Signals using Arduino,

Node.js & storing signals in MongoDB

& mining data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2021

Email : chaos21c@gmail.com

ID를 확인하고 github에 repo 만들기

AA01	김준수	AA13	조재윤
AA02	김현서	AA14	고태승
AA03	박영훈	AA15	이한글
AA04	박윤호	AA16	장세진
AA05	성은지	AA17	장태호
AA06	손윤우	AA18	정지원
AA07	오세윤	AA19	진우태
AA08	우승철	AA20	황혁준
AA09	윤현석	AA21	장이제
AA10	이예주	AA22	박상현
AA11	강지환	AA23	정은성
AA12	성인제	AA24	김경영

위의 **id**를 이용해서 **github**에 **repo**를 만드시오.

Option: 아두이노응용 실습 과제 **– AAnn**

Public,   README.md check

[Review]
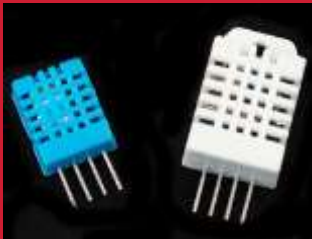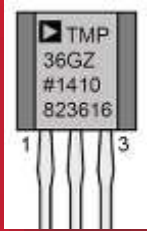
◆ **[wk13]**

➢ **IoT Project: nano33ble**

➢ **Multi-sensor circuits)**

➢ **Complete your project**

➢ **Upload folder: aann-rpt13**

➢ **Use repo "aann" in github**

◆ **[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**

② **iot_nano33_json.ipynb**

③ **All *.js**

④ **public/All *.html**
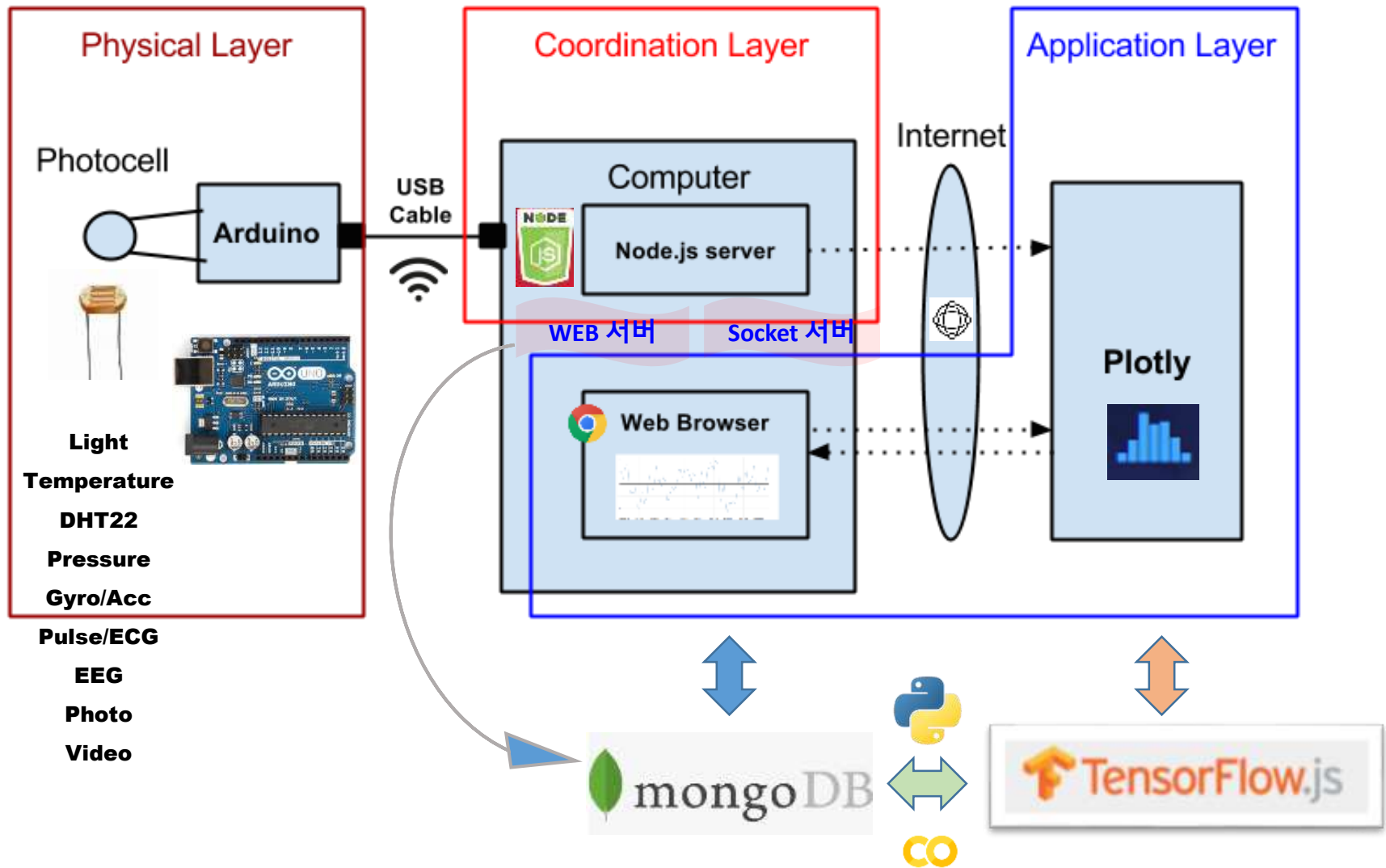
주요 수업 목표는 다음과 같다.
1. Node.js를 이용한 아두이노 센서 신호 처리
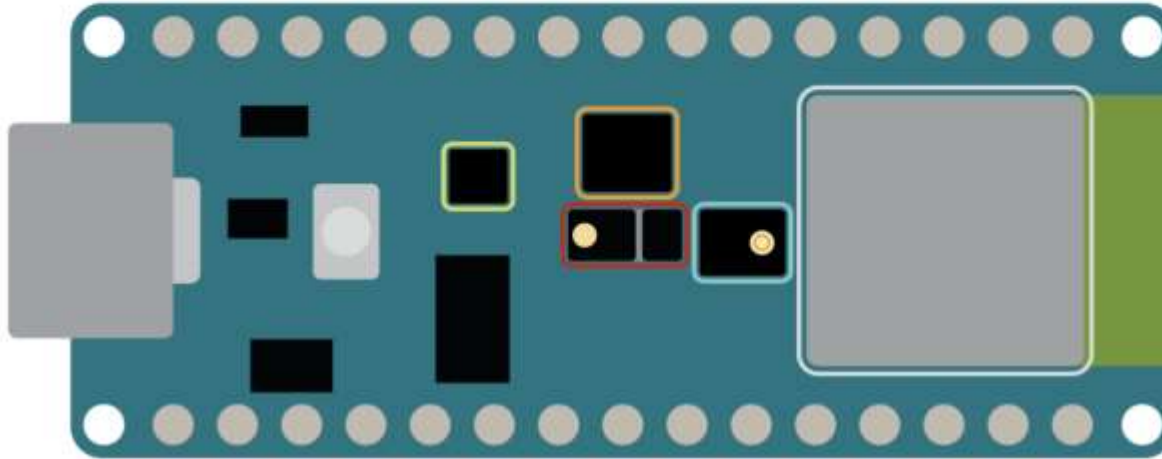2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]

Physical Layer

Photocell

Arduino

USB Cable

Coordination Layer

Computer

Node.js server

WEB 서버   Socket 서버

Web Browser

Internet

Application Layer

Plotly

Light
Temperature
DHT22
Pressure
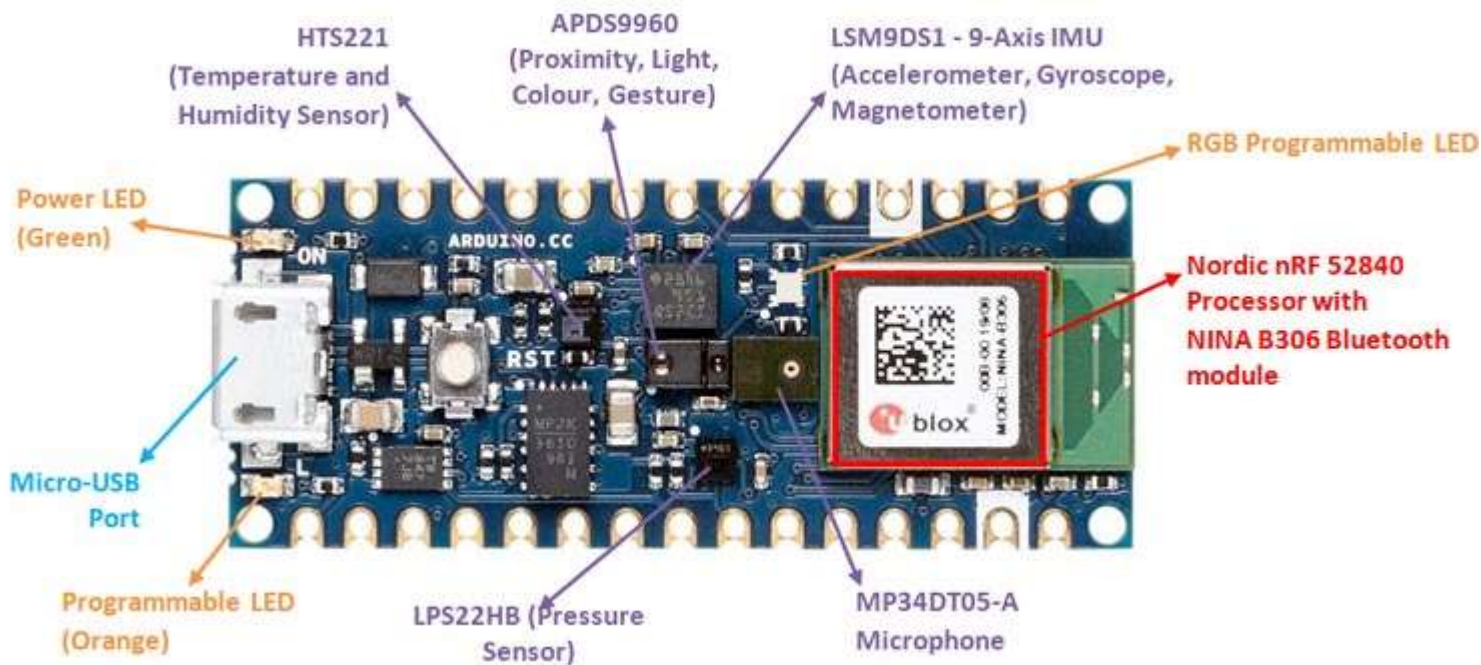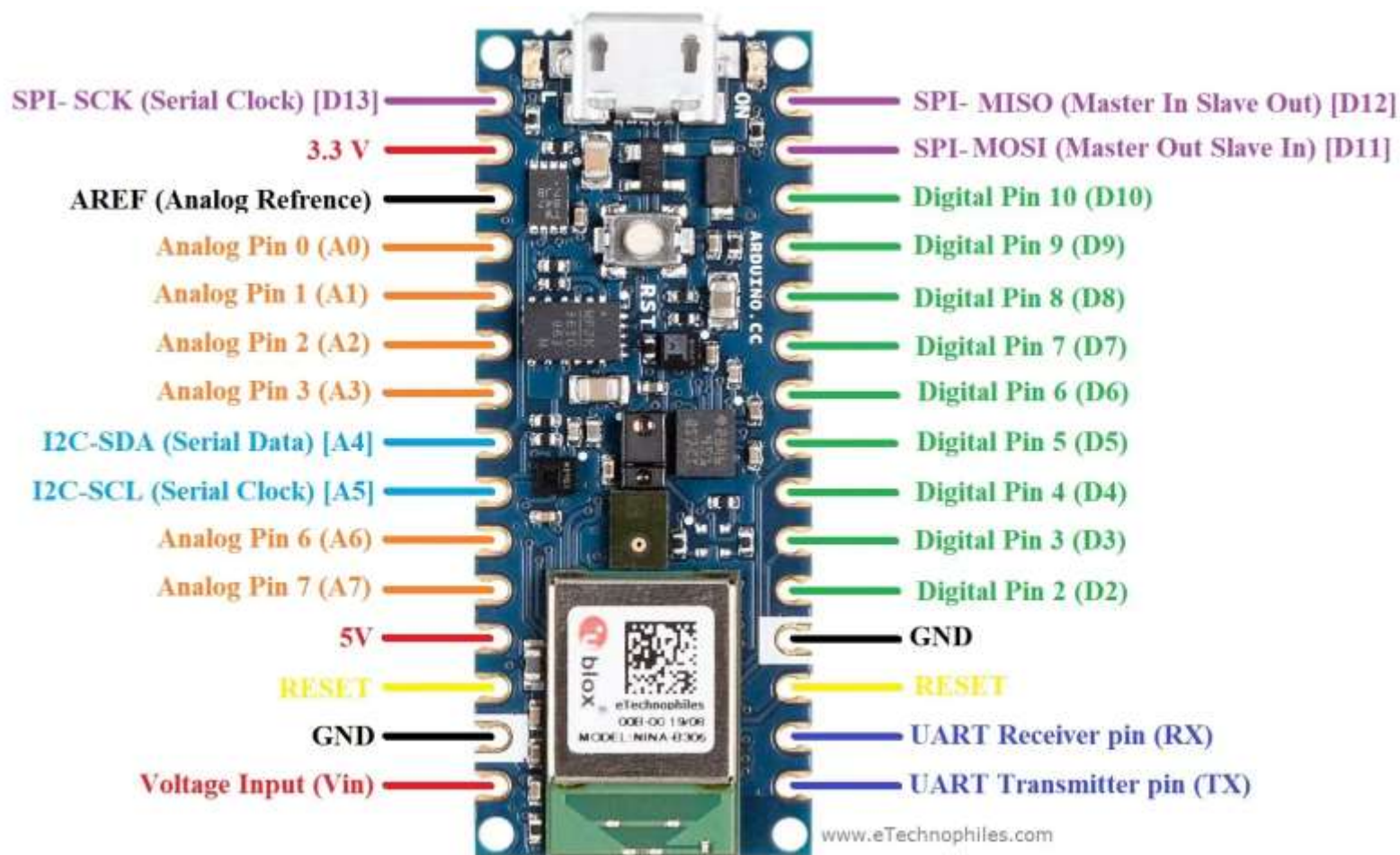Gyro/Acc
Pulse/ECG
EEG
Photo
Video

mongoDB

TensorFlow.js

NANO 33 BLE SENSE
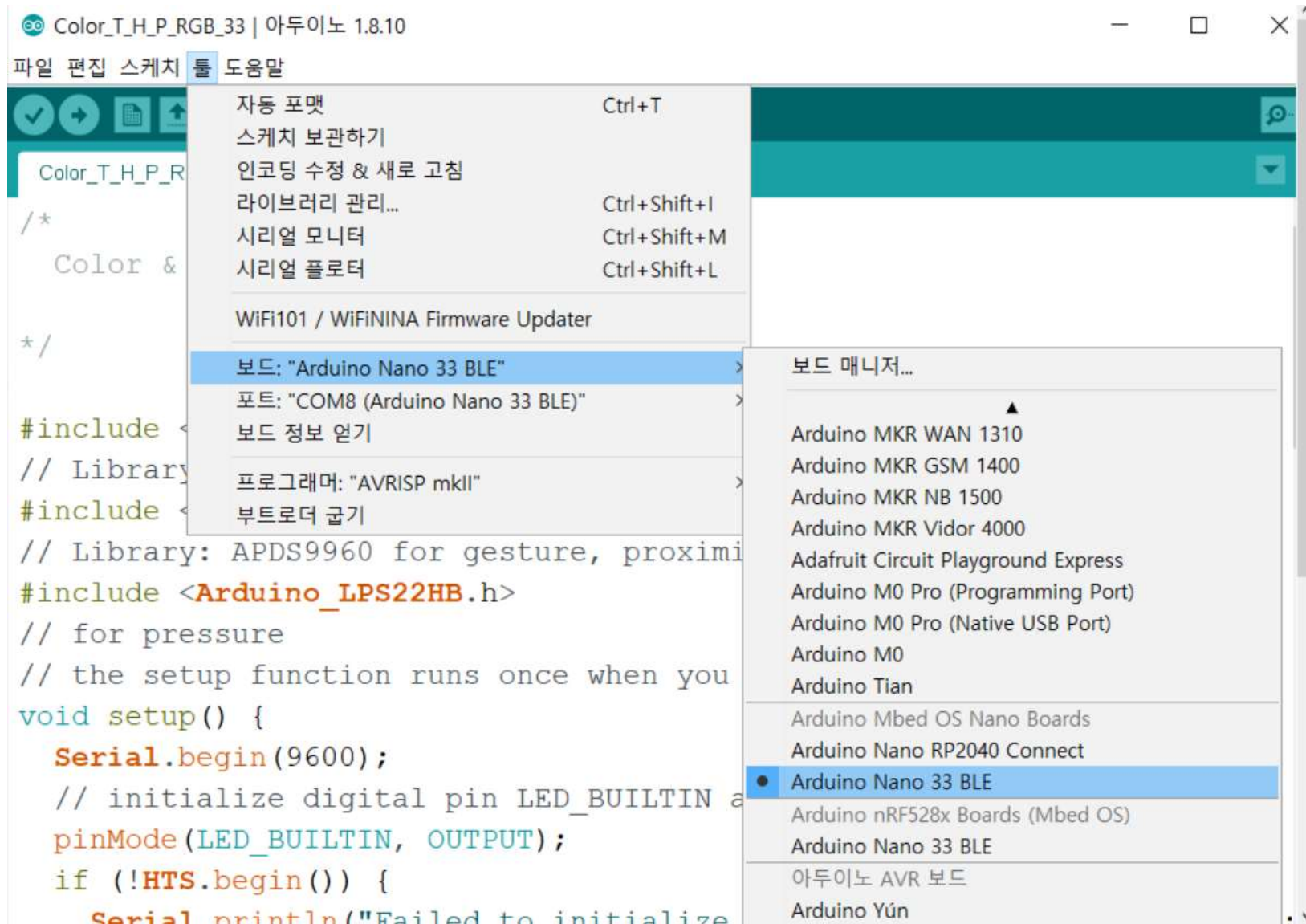

◆ Color, brightness, proximity and gesture sensor

◆ Digital microphone

◆ Motion, vibration and orientation sensor

◆ Temperature, humidity and pressure sensor
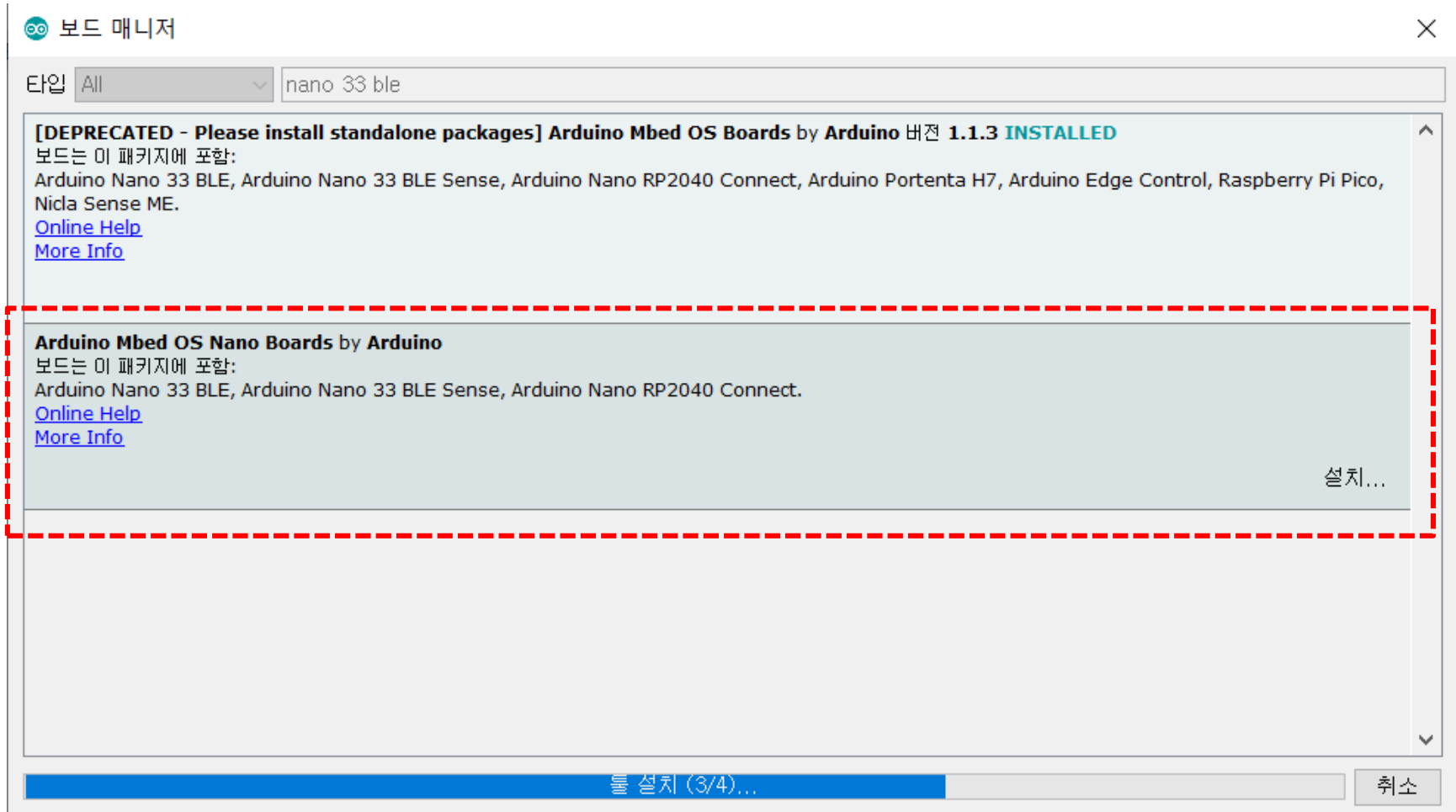
◆ Arm Cortex-M4 microcontroller and BLE module

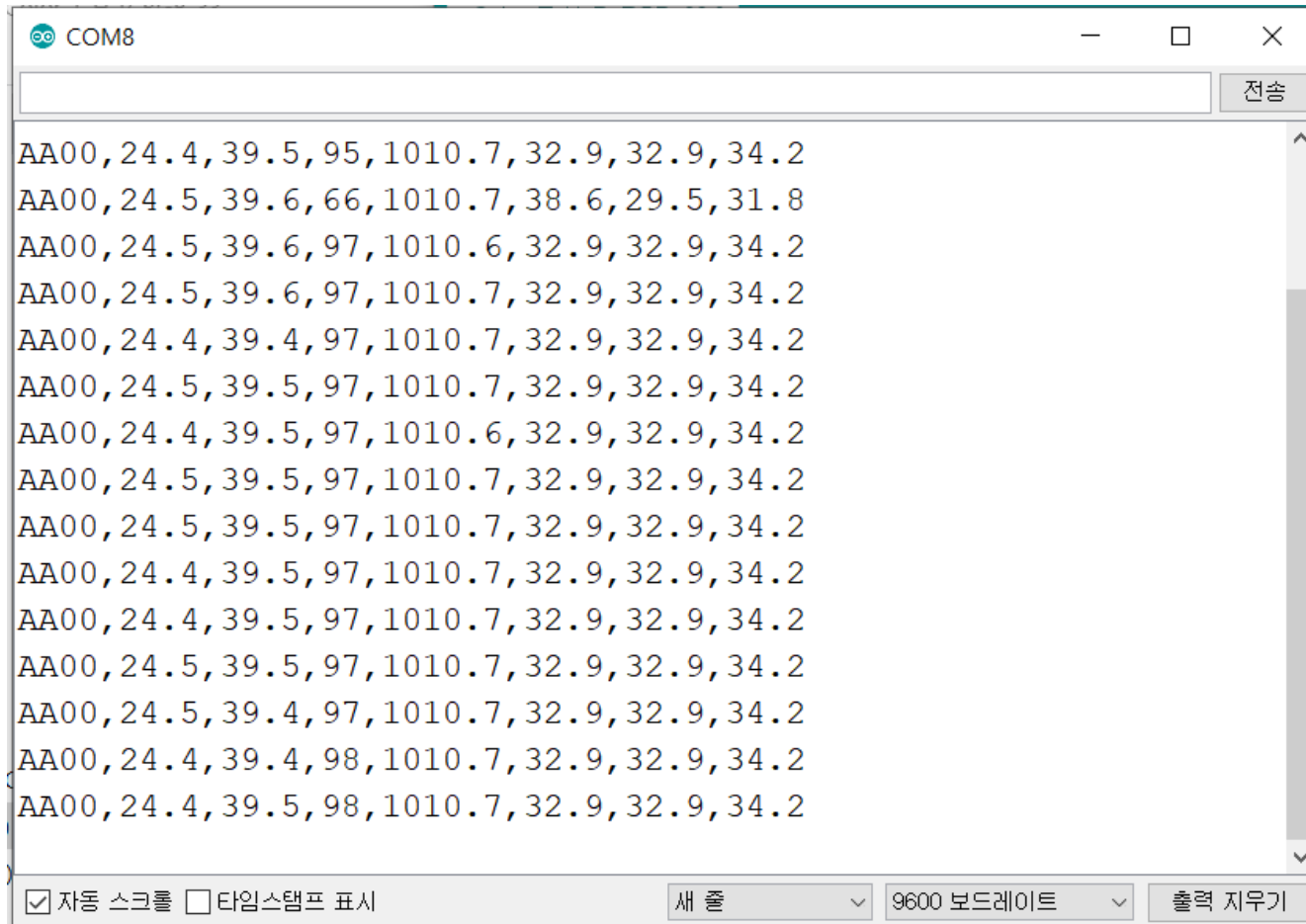https://t1.daumcdn.net/cfile/tistory/99D8E84E5F93D2B109

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

Layout [H S C]

Nono33 board

보드 매니저   ✕

타입 All ▾   nano 33 ble

[DEPRECATED - Please install standalone packages] Arduino Mbed OS Boards by **Arduino** 버전 **1.1.3 INSTALLED**
보드는 이 패키지에 포함:
Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect, Arduino Portenta H7, Arduino Edge Control, Raspberry Pi Pico, Nicla Sense ME.
Online Help
More Info

Arduino Mbed OS Nano Boards by **Arduino**
보드는 이 패키지에 포함:
Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect.
Online Help
More Info

설치...

툴 설치 (3/4)...   취소

Sensor library

Arduino: serial monitor

npm update

```
D:\aann\aann-rpt13\nano33>npm update

> @serialport/bindings@9.2.7 install D:\aann\aann-rpt13\nano33\node_modu
les\@serialport\bindings
> prebuild-install --tag-prefix @serialport/bindings@ || node-gyp rebuil
d

npm notice created a lockfile as package-lock.json. You should commit th
is file.
npm WARN cds_dht22@1.0.0 No repository field.

+ socket.io@2.4.1
+ express@4.17.1
+ cors@2.8.5
+ mongoose@6.0.13
+ serialport@9.2.7
added 195 packages from 149 contributors and audited 195 packages in 18.
431s

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

found 0 vulnerabilities


D:\aann\aann-rpt13\nano33>
```

db33rgb.js

```js
// Schema
var iotSchema = new Schema({
    date : String,
    temperature : String,
    humidity : String,
    luminosity : String,
    pressure : String,
    r_ratio : String,
    g_ratio : String,
    b_ratio : String
});
```

db33rgb.js

```javascript
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);
```

db33rgb.js

```javascript
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-rpt13\nano33>node db33rgb
mongo db connection OK.
iotInfo: Current date: 2021-11-19 15:55:08.449, Temp: 24.3, Humi: 39.4, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:13.491, Temp: 24.4, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:18.533, Temp: 24.4, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:23.575, Temp: 24.4, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:28.616, Temp: 24.4, Humi: 39.4, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:33.659, Temp: 24.4, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:38.701, Temp: 24.4, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
iotInfo: Current date: 2021-11-19 15:55:43.742, Temp: 24.3, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.9, G: 32.9, B: 34.2
iotInfo: Current date: 2021-11-19 15:55:48.784, Temp: 24.4, Humi: 39.3, Lux: 98, Pres: 1010.8,
R: 32.9, G: 32.9, B: 34.2
iotInfo: Current date: 2021-11-19 15:55:53.826, Temp: 24.4, Humi: 39.3, Lux: 8, Pres: 1010.8, R
: 44.4, G: 33.3, B: 22.2
iotInfo: Current date: 2021-11-19 15:55:58.868, Temp: 24.4, Humi: 39.4, Lux: 98, Pres: 1010.8,
R: 32.5, G: 33.7, B: 33.7
```

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

node db33rgb

Express server : port=3030

node express33rgb

문제 출력 디버그콘솔 터미널

iotInfo: Current date: 2021-11-19 15:57:24.581, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.1, G: 33.3, B: 34.6
iotInfo: Current date: 2021-11-19 15:57:29.623, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.1, G: 33.3, B: 34.6
iotInfo: Current date: 2021-11-19 15:57:34.665, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.1, G: 33.3, B: 34.6
iotInfo: Current date: 2021-11-19 15:57:39.708, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.1, G: 33.3, B: 34.6
iotInfo: Current date: 2021-11-19 15:57:44.749, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.9, G: 32.9, B: 34.2
iotInfo: Current date: 2021-11-19 15:57:49.792, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.9, G: 32.9, B: 34.2
iotInfo: Current date: 2021-11-19 15:57:54.833, Temp: 24.5, Humi: 39.5, Lux: 97, Pres: 1010.8,
R: 32.9, G: 32.9, B: 34.2
iotInfo: Current date: 2021-11-19 15:57:59.876, Temp: 24.4, Humi: 39.4, Lux: 97, Pres: 1010.8,
R: 32.9, G: 32.9, B: 34.2

        382 package.json
2021-11-19 오후 03:35   <DIR>   public
              4개 파일   65,008 바이
트
              4개 디렉터리 2,403,824,119,808 바
이트 남음

D:\aann\aann-rpt13\nano33>node express33rgb
Express_IOT is running at port:3030, CORS powered
!
mongo db connection OK.

mongo shell

```
> show dbs
admin            0.000GB
config           0.000GB
iot              0.015GB
iot10            0.000GB
iot33rgb211119   0.000GB
local            0.000GB
test             0.000GB
> use iot33rgb211119
switched to db iot33rgb211119
> show collections
sensors
> db.sensors.count()
88
> db.sensors.find().sort({_id:-1}).limit(5)
{ "_id" : ObjectId("61974c2139d363e662ef451b"), "date" : "2021-11-19 16:02:57.350", "temperature" : "24.5", "humidity" : "39.6", "luminosity" : "97
", "pressure" : "1010.9", "r_ratio" : "32.9", "g_ratio" : "32.9", "b_ratio" : "34.2", "__v" : 0 }
{ "_id" : ObjectId("61974c1c39d363e662ef4519"), "date" : "2021-11-19 16:02:52.309", "temperature" : "24.4", "humidity" : "39.4", "luminosity" : "97
", "pressure" : "1010.9", "r_ratio" : "32.1", "g_ratio" : "33.3", "b_ratio" : "34.6", "__v" : 0 }
{ "_id" : ObjectId("61974c1739d363e662ef4517"), "date" : "2021-11-19 16:02:47.266", "temperature" : "24.4", "humidity" : "39.5", "luminosity" : "97
", "pressure" : "1010.8", "r_ratio" : "32.9", "g_ratio" : "32.9", "b_ratio" : "34.2", "__v" : 0 }
{ "_id" : ObjectId("61974c1239d363e662ef4515"), "date" : "2021-11-19 16:02:42.225", "temperature" : "24.4", "humidity" : "39.5", "luminosity" : "97
", "pressure" : "1010.8", "r_ratio" : "32.9", "g_ratio" : "32.9", "b_ratio" : "34.2", "__v" : 0 }
{ "_id" : ObjectId("61974c0d39d363e662ef4513"), "date" : "2021-11-19 16:02:37.183", "temperature" : "24.4", "humidity" : "39.6", "luminosity" : "97
", "pressure" : "1010.8", "r_ratio" : "32.9", "g_ratio" : "32.9", "b_ratio" : "34.2", "__v" : 0 }
>
```
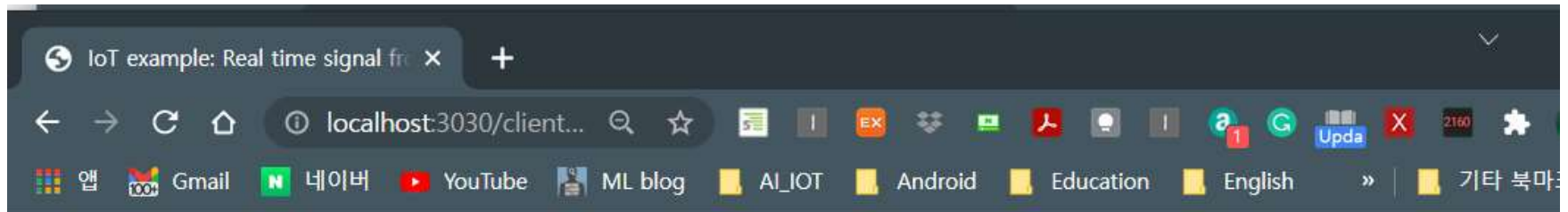
IoT Signal from Arduino

Real-time Signals

on Time: 2021-11-23 16:41:49.935

Signal (온도,습도, 조도, 기압, (R, G, B)) : 23.7,31.8,98,1009.3, (32.9,32.9,34.2)

Real-time Weather Station from nano 33 BLE sensors

on Time: 2021-11-23 16:45:06.581
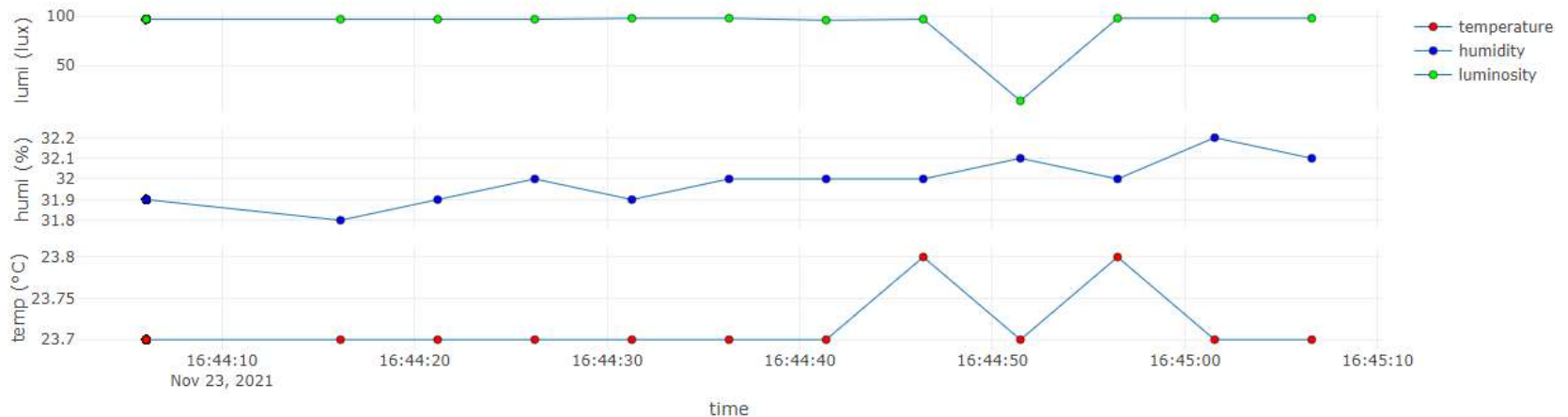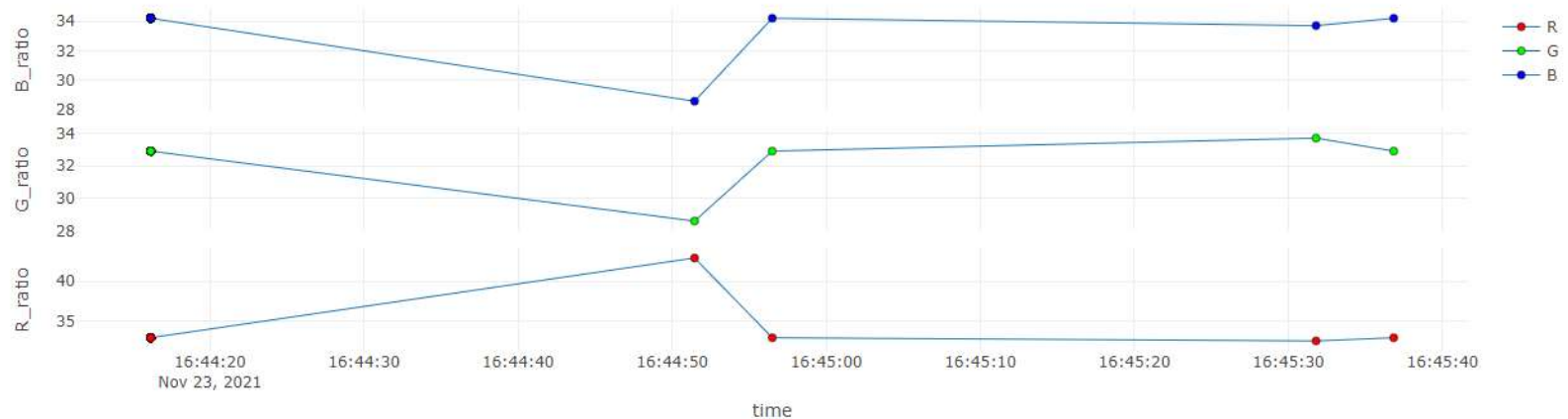
http://localhost:3030/client_33rgb.html

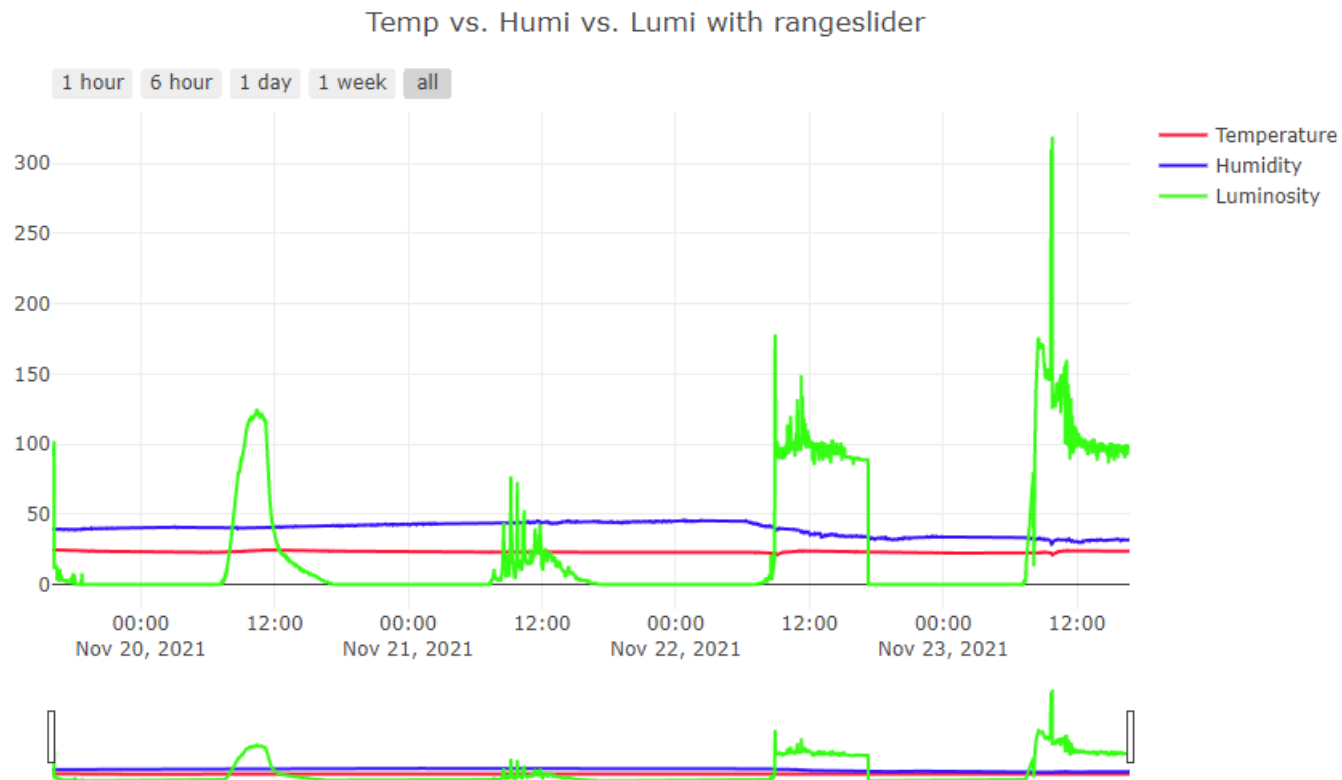Real-time Ambient Colors from nano 33 BLE sensors

MongoDB database visualization by AA00

Time series : Multiple data from nano 33 ble sensor

Temp vs. Humi vs. Lumi with rangeslider

작업 폴더 구조 [2021-project]
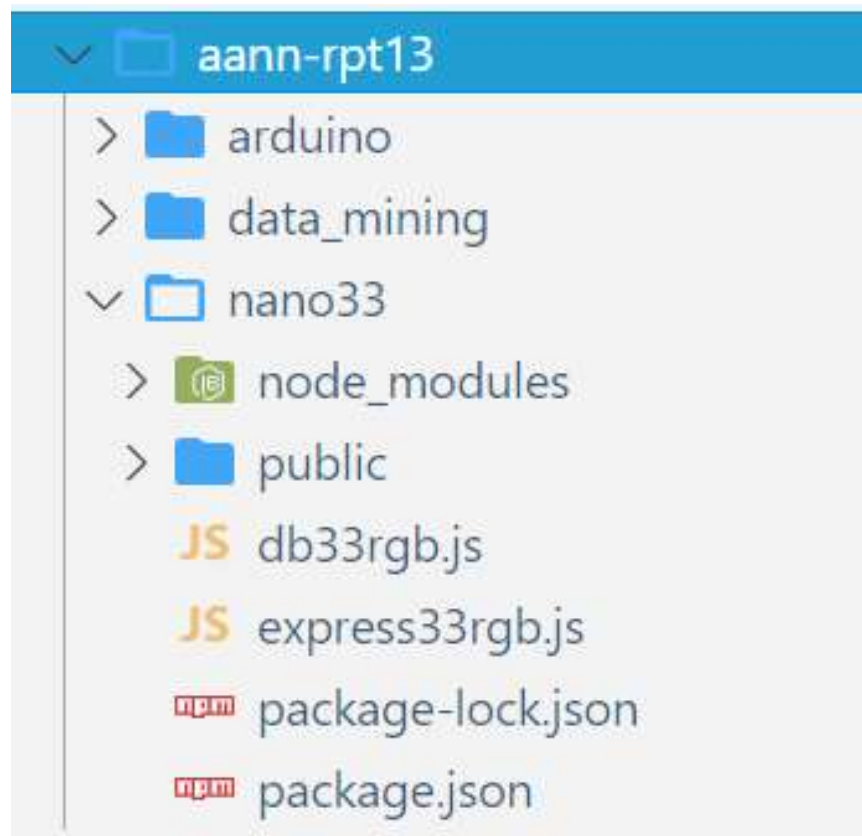
Project: nano33BLE sensor

[2021-project] IoT data mining in Colab
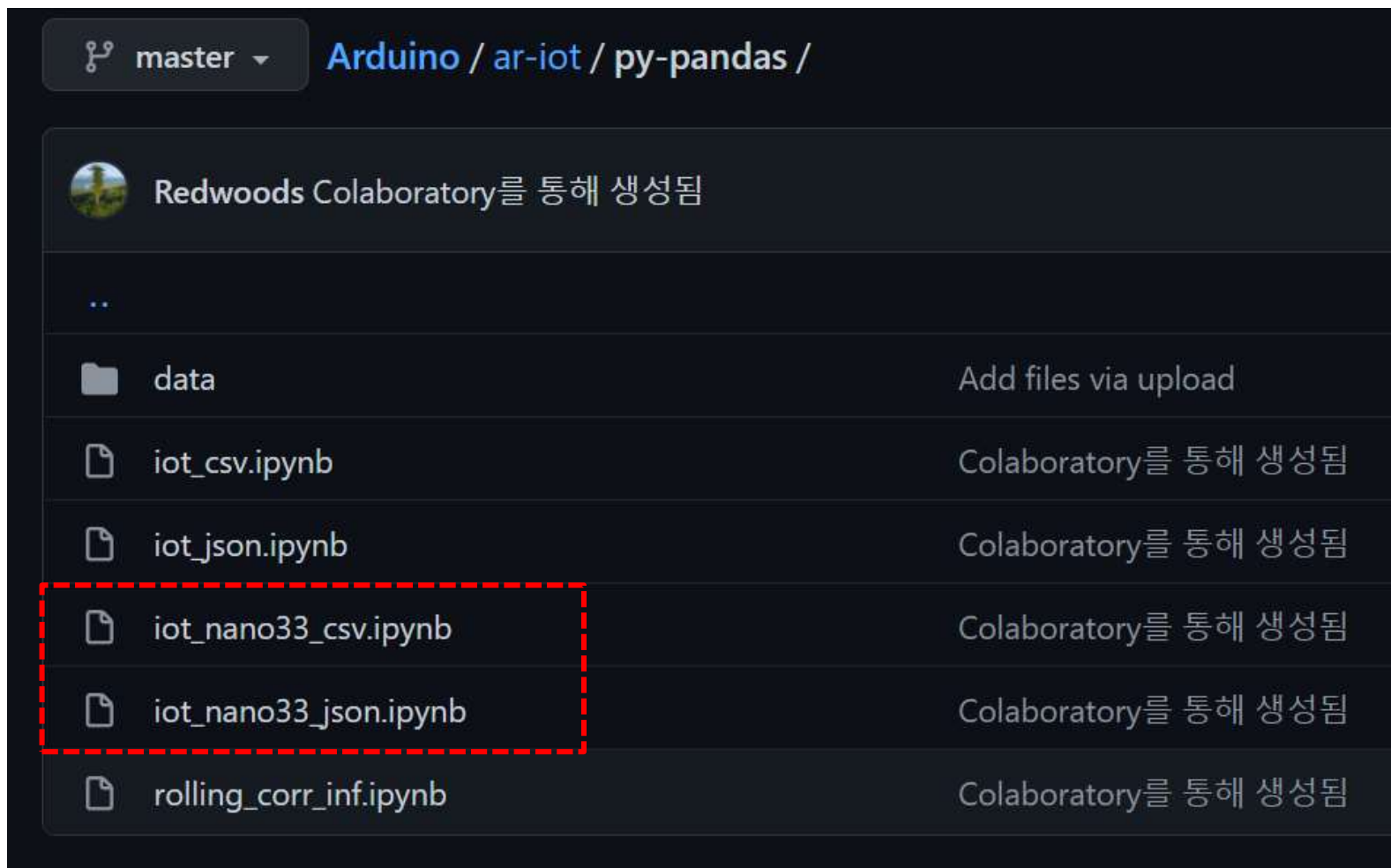
2021 AA team project

nano33BLE

sensor

HTS221
(Temperature and
Humidity Sensor)

APDS9960
(Proximity, Light,
Colour, Gesture)

LSM9DS1 - 9-Axis IMU
(Accelerometer, Gyroscope,
Magnetometer)

RGB Programmable LED

Power LED
(Green)

ARDUINO.CC

Nordic nRF 52840
Processor with
NINA B306 Bluetooth
module

Micro-USB
Port

Programmable LED
(Orange)

LPS22HB (Pressure
Sensor)

MP34DT05-A
Microphone
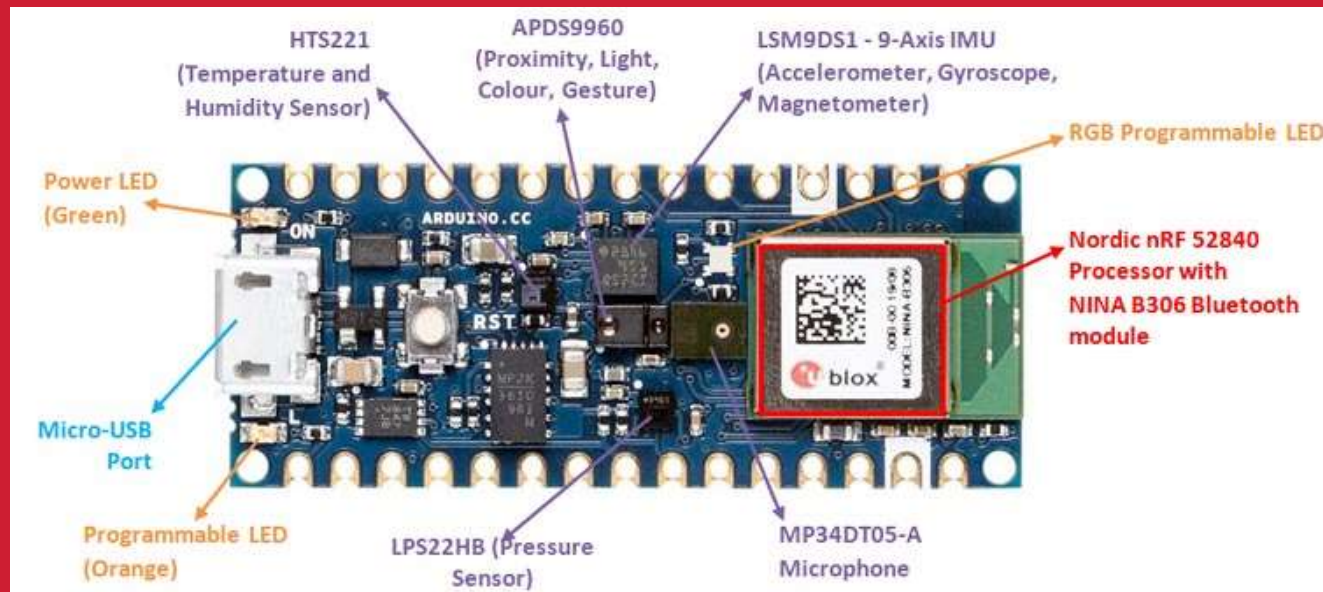
LSM9DS1, 9축 IMU센서: acc, gyro, mag

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

* 9축 IMU 센서 신호 마이닝

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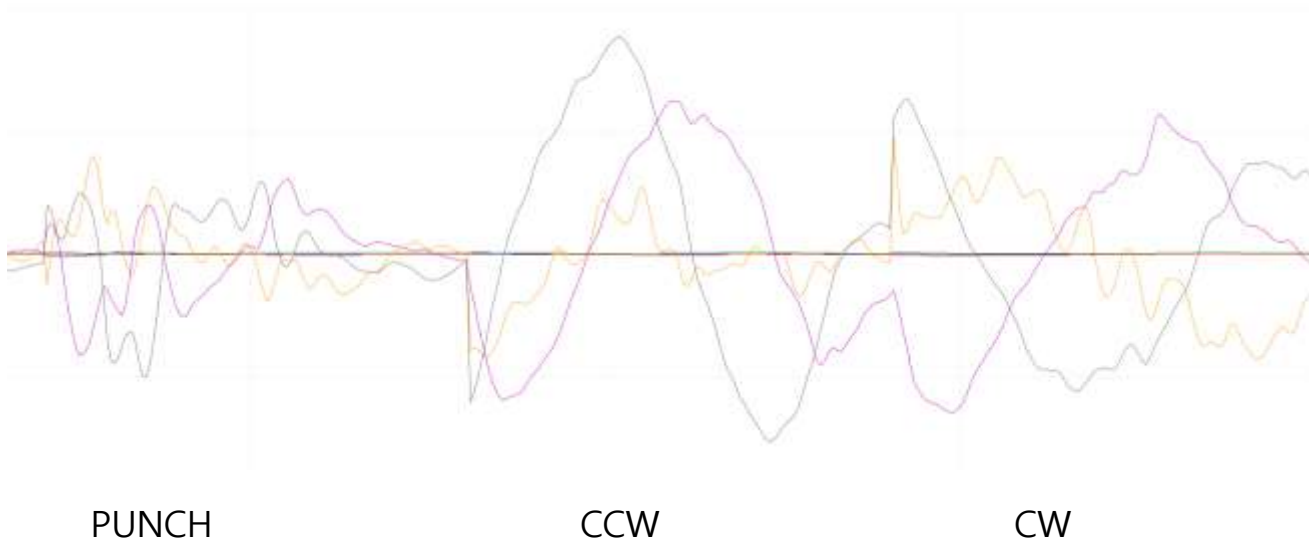   - 실시간 모니터링
       - DB 모니터링
    3. data mining using Colab
    4. Deep learning ?

모션 인식(9-축 IMU)

Data 모으기

PUNCH   CCW   CW

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

Useful sites

- https://docs.arduino.cc/tutorials/nano-33-ble-sense/imu_accelerometer
- https://fishpoint.tistory.com/4419?category=626454
- https://www.tensorflow.org/lite/microcontrollers?hl=ko
- https://github.com/arduino/ArduinoTensorFlowLiteTutorials

Plan for the good team project

1. Real-time monitoring IMU data while saving data in MongoDB

2. Monitoring IMU data saved in MongoDB and mining data.

3. Search web to get idea how to use IMU data.

4. Make ppt to show what your team have done.

5. Try ML or DL using IMU data

[Practice]
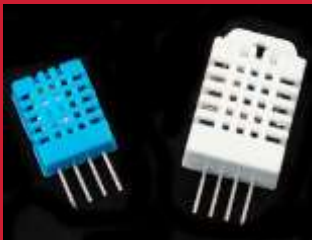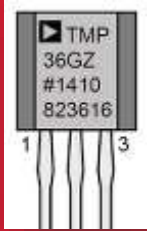
◆ **[wk14]**

➢ **IoT Project: nano33ble**

➢ **Multi-sensor circuits)**

➢ **Complete your project**

➢ **Upload folder: aann-rpt14-project**

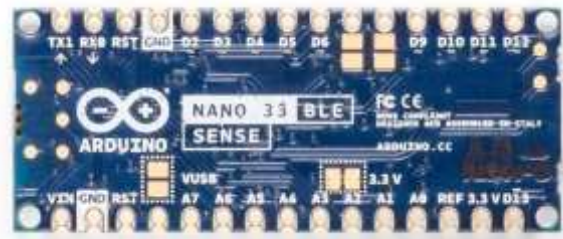➢ **Use repo "aann" in github**

◆ **[Target of this week]**

- **Complete your works**

- **Save your outcomes and upload outputs in github**
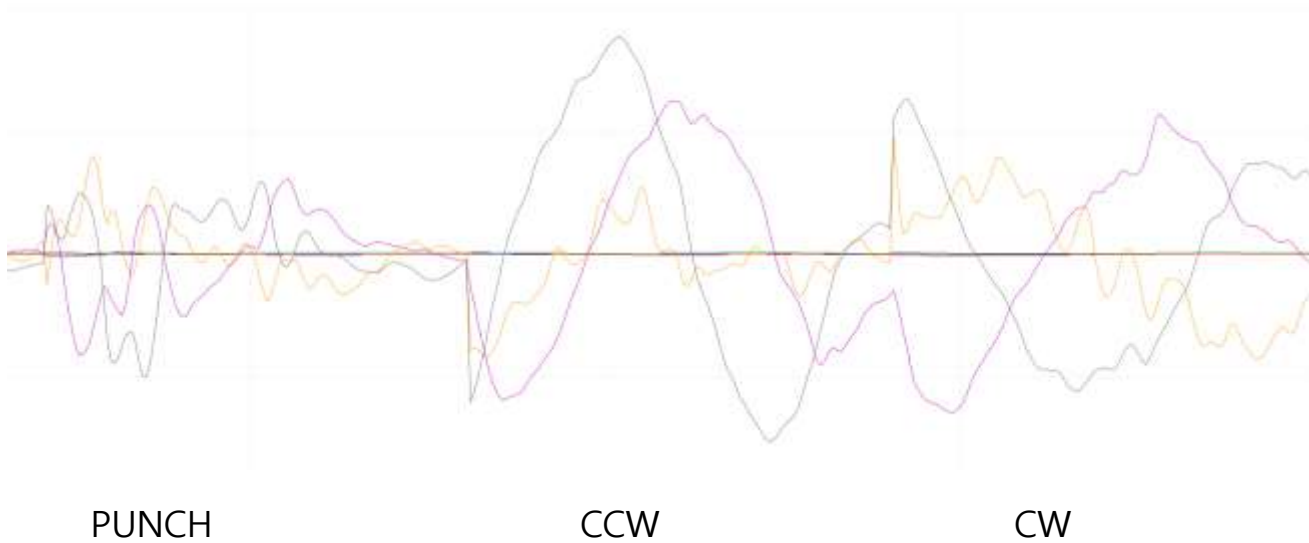
제출폴더명 : **aann-rpt14-project**

- 제출할 파일들

  ① **Code – Arduino**

  ② **Code – Node**

  ③ **Code – Client html files in public folder**

  ④ **aamn-aaxy-project.pptx**

Arduino nano33 BLE

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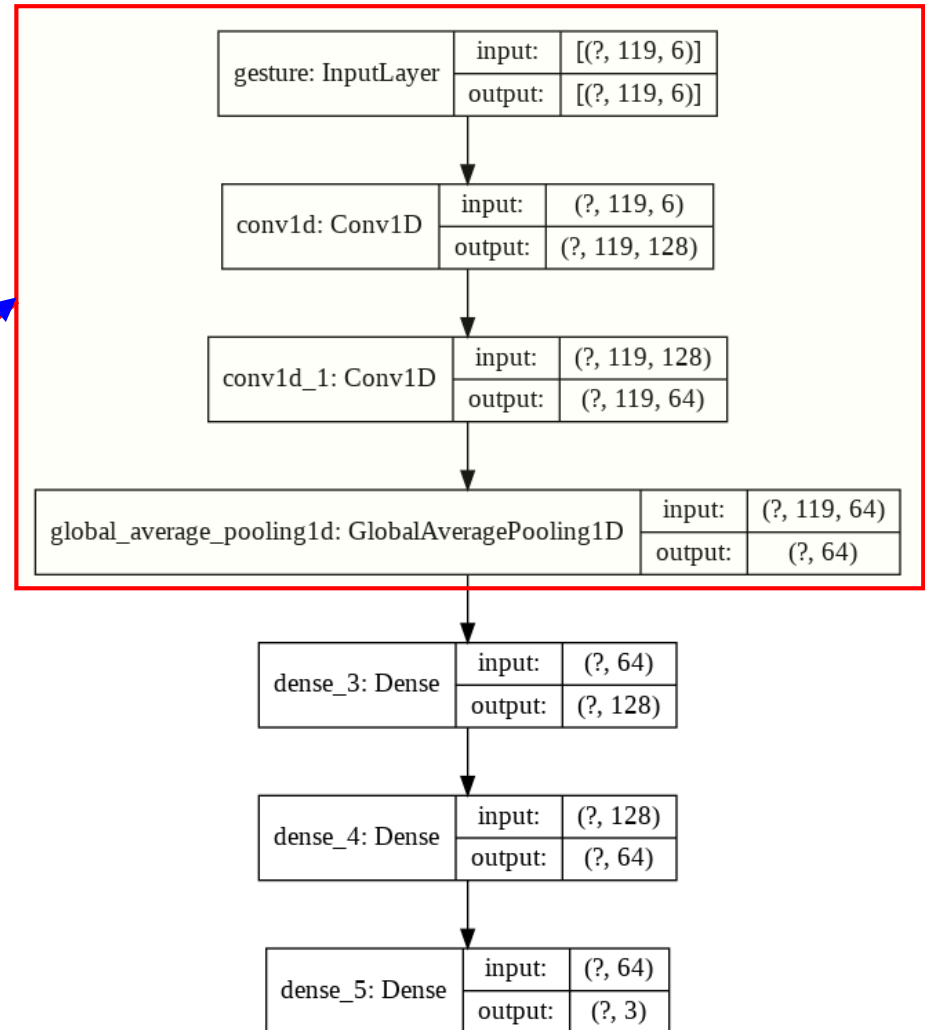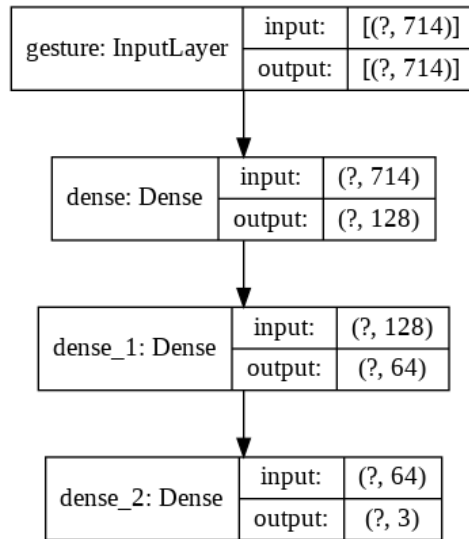## using ACC
in Tensorflow 2.x
& TinyML/TF-Lite

Data 모으기

PUNCH   CCW   CW

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

DL architecture

MLP → Conv1D + MLP

(714,) → (119,6) + (64,)

gesture: InputLayer	input:	[(?, 714)]
	output:	[(?, 714)]

dense: Dense	input:	(?, 714)
	output:	(?, 128)

dense_1: Dense	input:	(?, 128)
	output:	(?, 64)

dense_2: Dense	input:	(?, 64)
	output:	(?, 3)

gesture: InputLayer	input:	[(?, 119, 6)]
	output:	[(?, 119, 6)]

conv1d: Conv1D	input:	(?, 119, 6)
	output:	(?, 119, 128)

conv1d_1: Conv1D	input:	(?, 119, 128)
	output:	(?, 119, 64)

global_average_pooling1d: GlobalAveragePooling1D	input:	(?, 119, 64)
	output:	(?, 64)

dense_3: Dense	input:	(?, 64)
	output:	(?, 128)

dense_4: Dense	input:	(?, 128)
	output:	(?, 64)

dense_5: Dense	input:	(?, 64)
	output:	(?, 3)

DL-model

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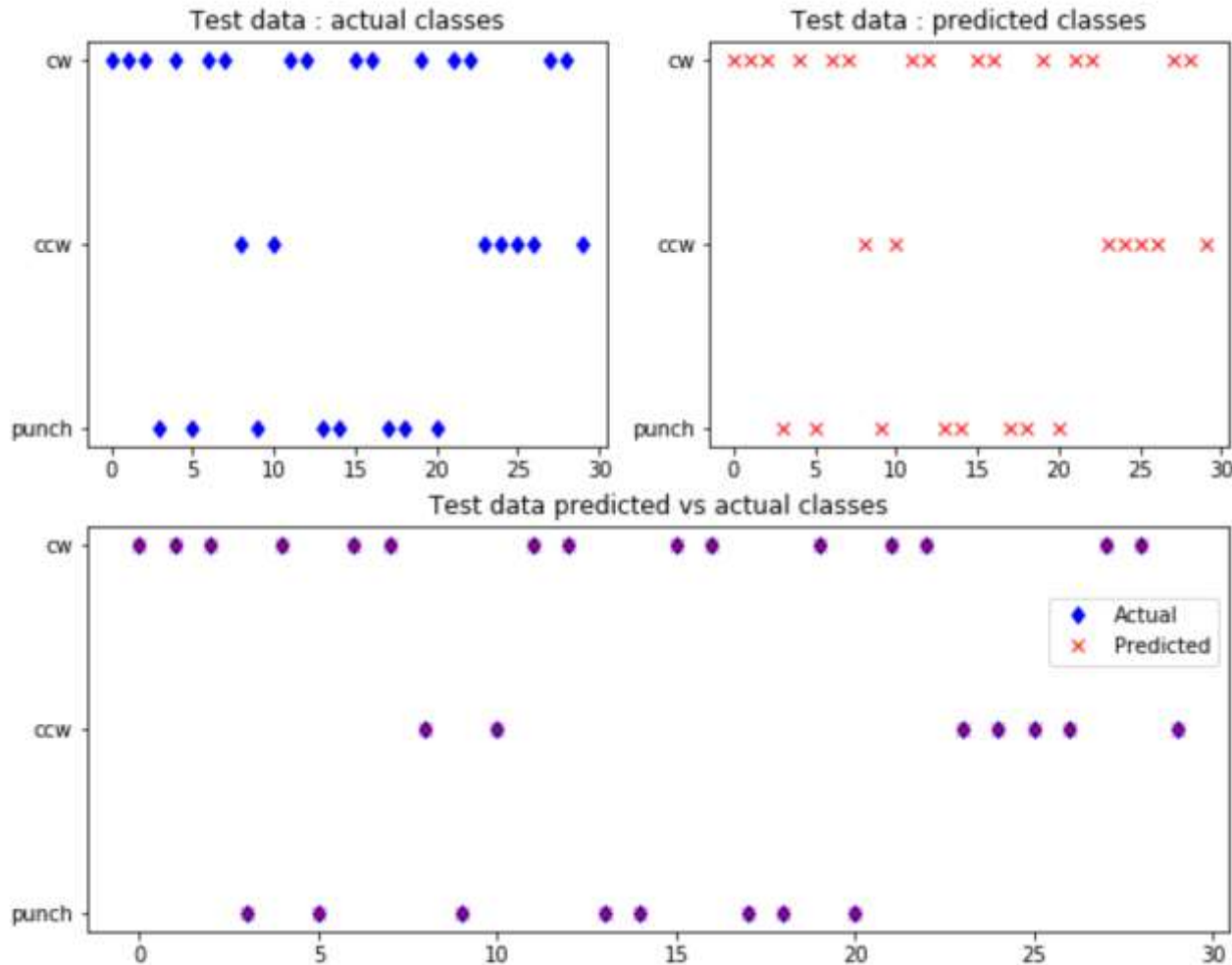
predictions =
[[0.   0.001 0.999]
[0.001 0.001 0.998]
[0.005 0.001 0.994]
[0.999 0.   0.001]
[0.   0.001 0.999]
[1.   0.   0.]
[0.   0.001 0.999]
[0.002 0.001 0.997]
[0.   1.   0.]
[1.   0.   0.]
[0.   1.   0.]
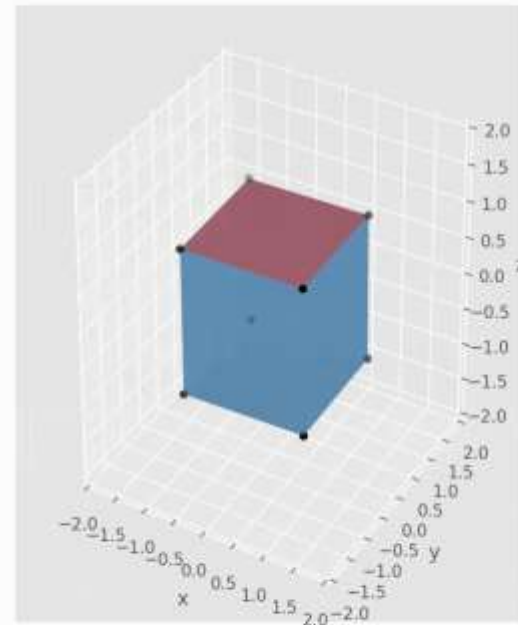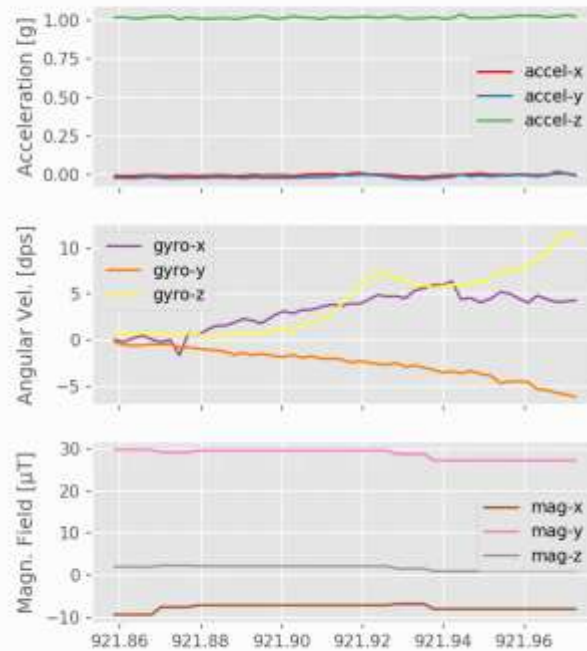[0.   0.001 0.999]
[0.001 0.   0.999]
[0.997 0.   0.003]

Real-time testing

PUNCH   CCW   CW
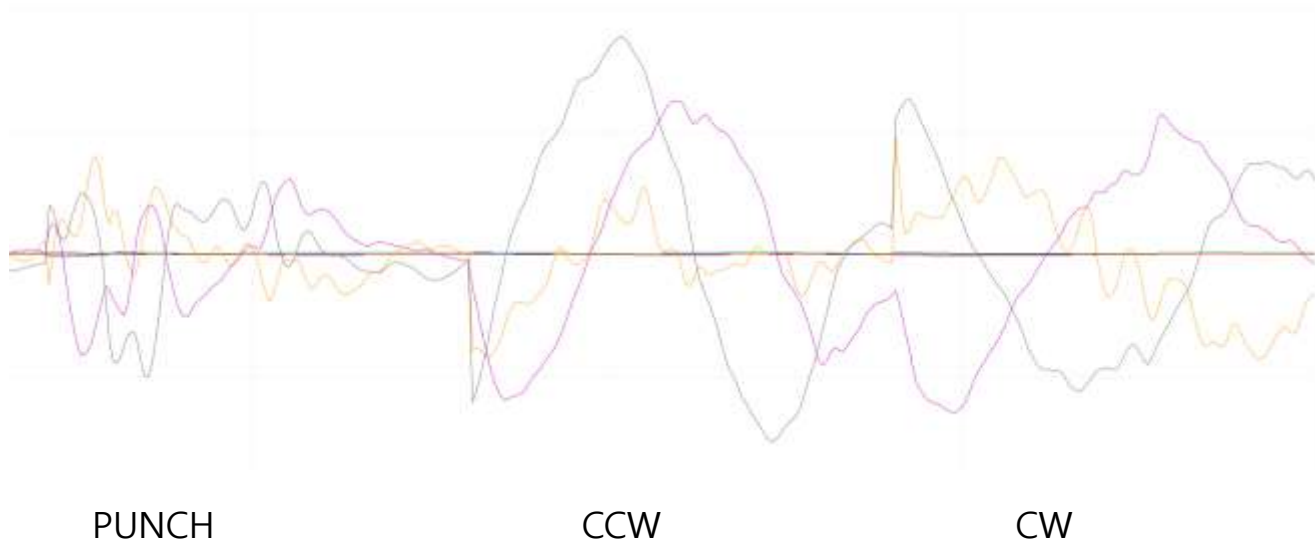
COM8

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
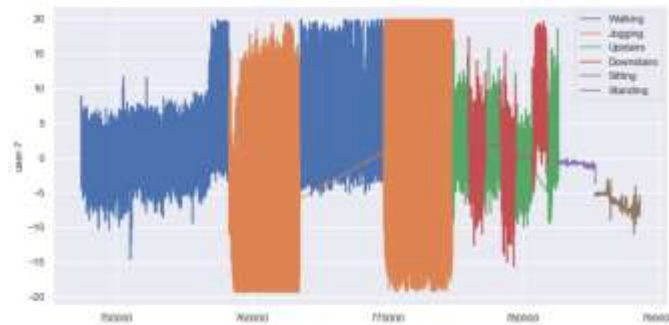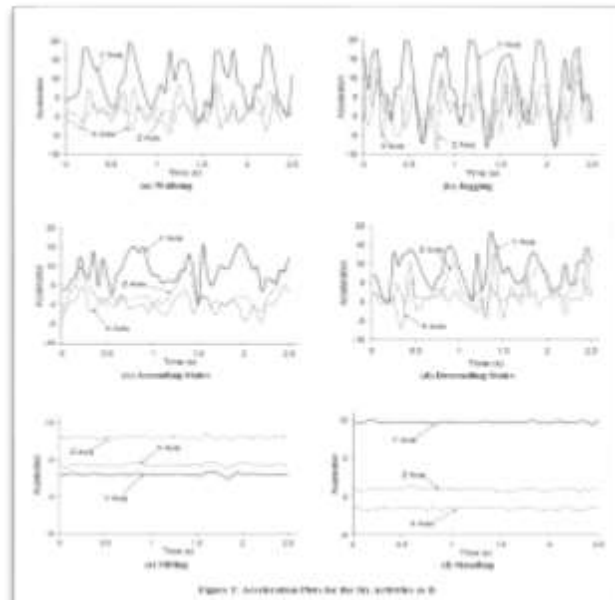13:24:13.193 -> ccw: 1.000000
13:24:13.193 -> cw: 0.000000

모션 인식(9-축 IMU)

일상활동 인식(3축 가속도)

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

- **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**

Real-time Weather Station from nano 33 BLE sensors