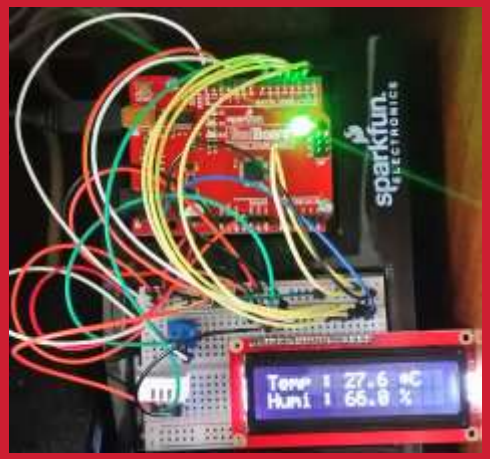


Arduino-IOT

[wk13]

nano33BLE Sensor IoT Project

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

My ID

ID를 확인하고 github에 repo 만들기

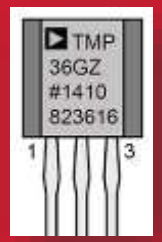
AA01	김준수	AA13	조재윤
AA02	김현서	AA14	고태승
AA03	박영훈	AA15	이한글
AA04	박윤호	AA16	장세진
AA05	성은지	AA17	장태호
AA06	손윤우	AA18	정지원
AA07	오세윤	AA19	진우태
AA08	우승철	AA20	황혁준
AA09	윤현석	AA21	장이제
AA10	이예주	AA22	박상현
AA11	강지환	AA23	정은성
AA12	성인제	AA24	김경영

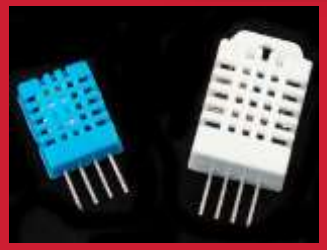
위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, **README.md check**

[Review]





- [wk12]
  - Data Mining of IoT Data
  - Multi-sensor circuits (cds-dht22)
  - 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_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

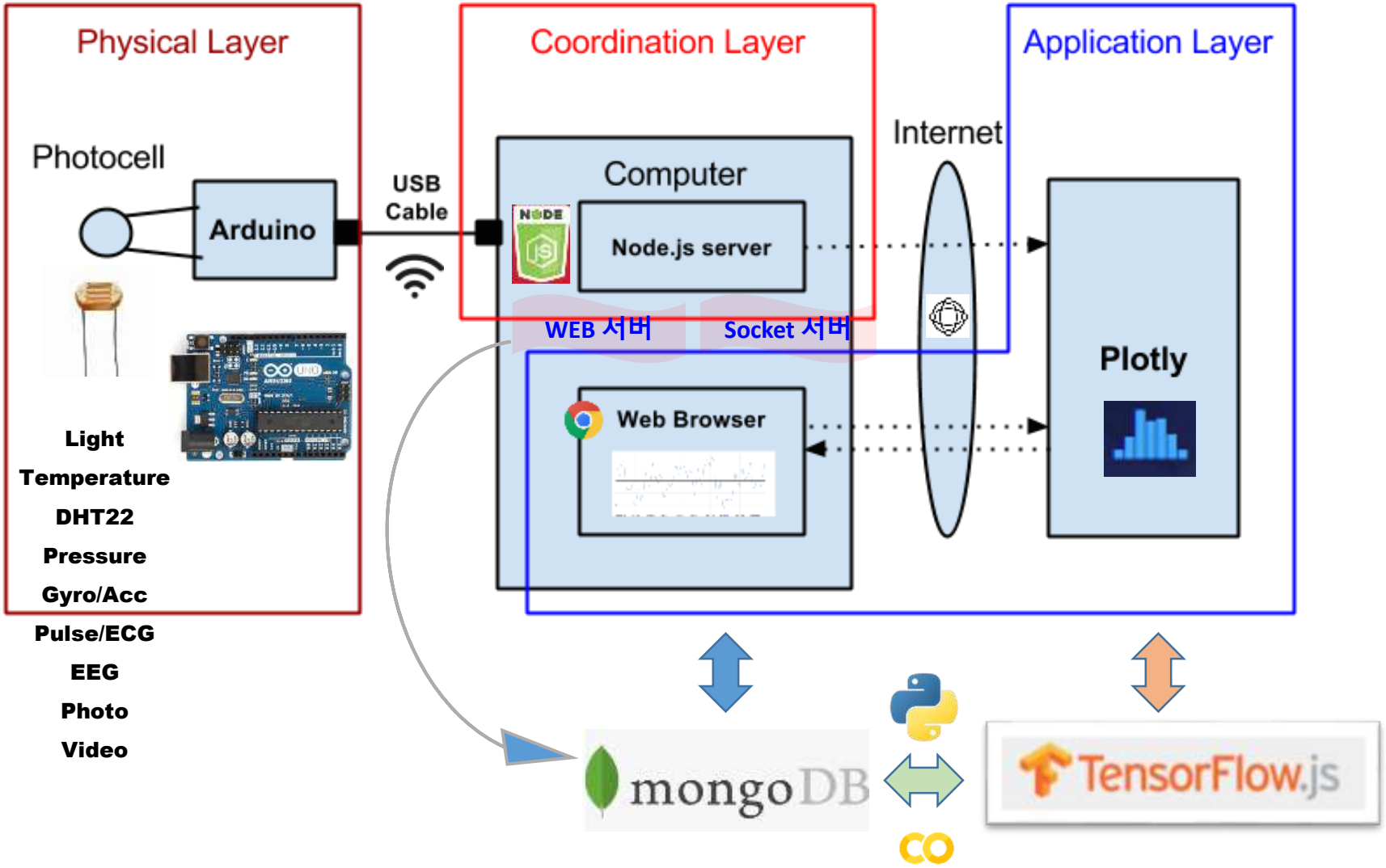
Purpose of AA

주요 수업 목표는 다음과 같다.

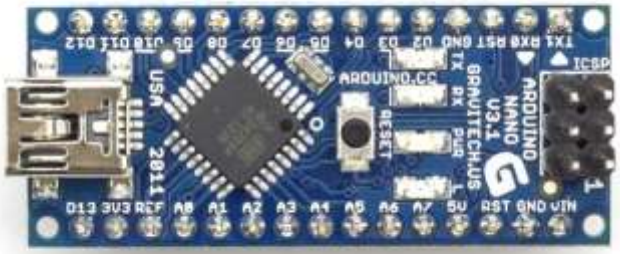
1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]

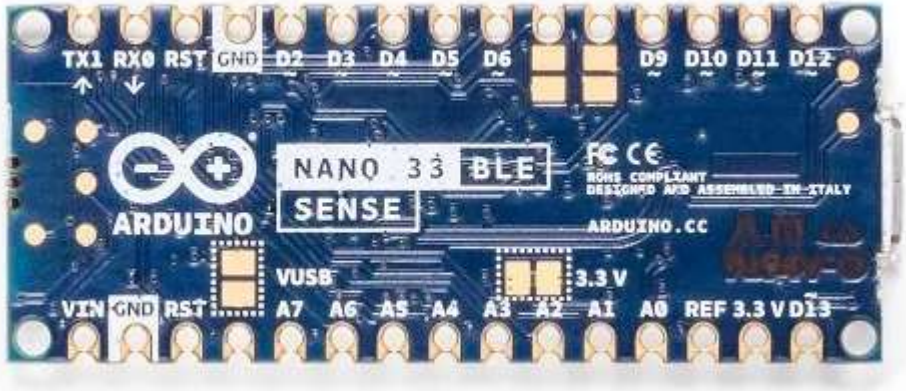

Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

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



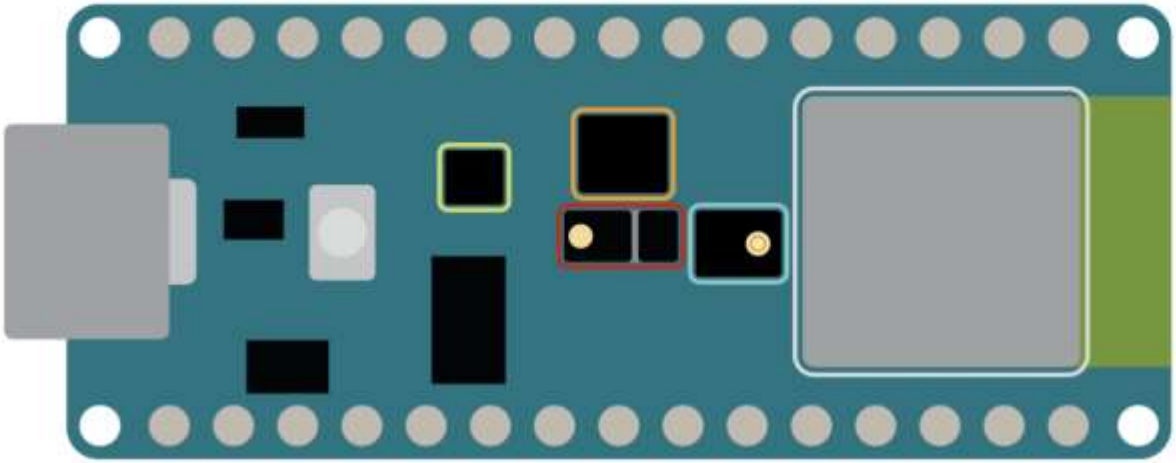
✓ Arduino NANO33

◆ BLE IOT

◆ SENSOR

nano33BLE sensor

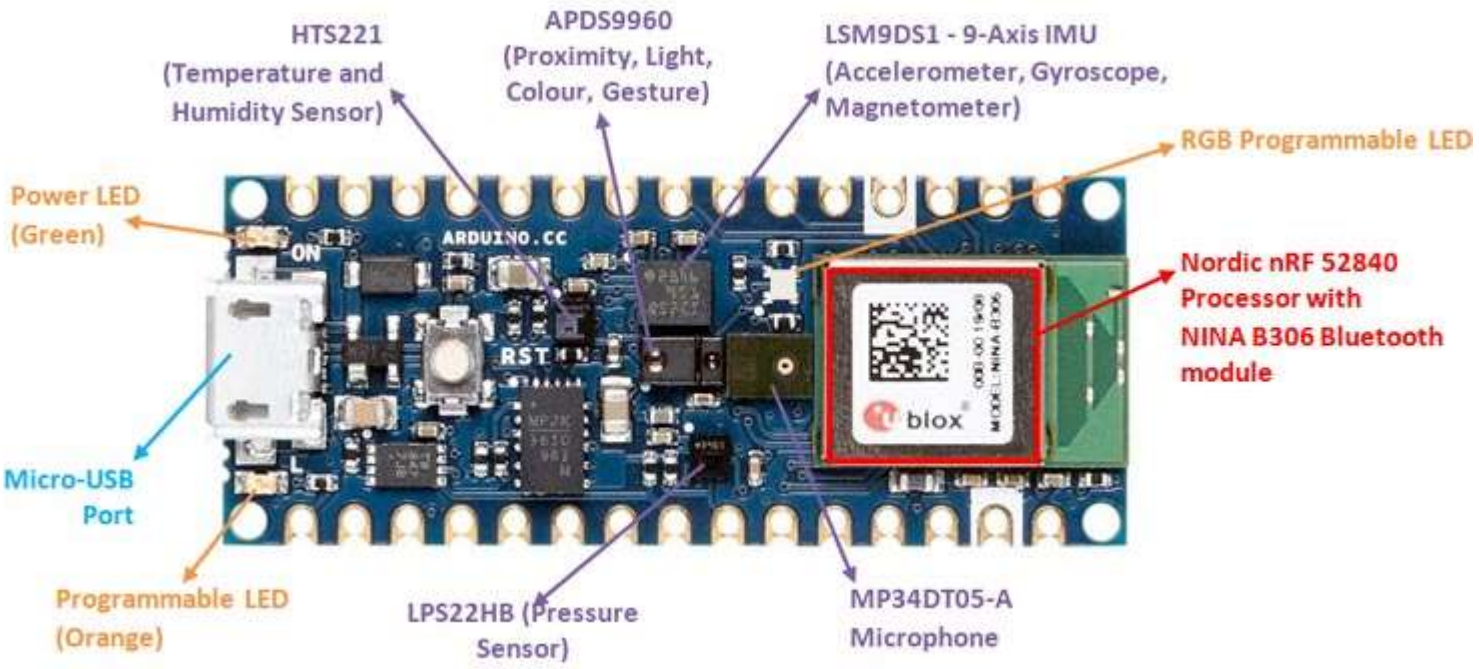
NANO 33 BLE SENSE



- 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

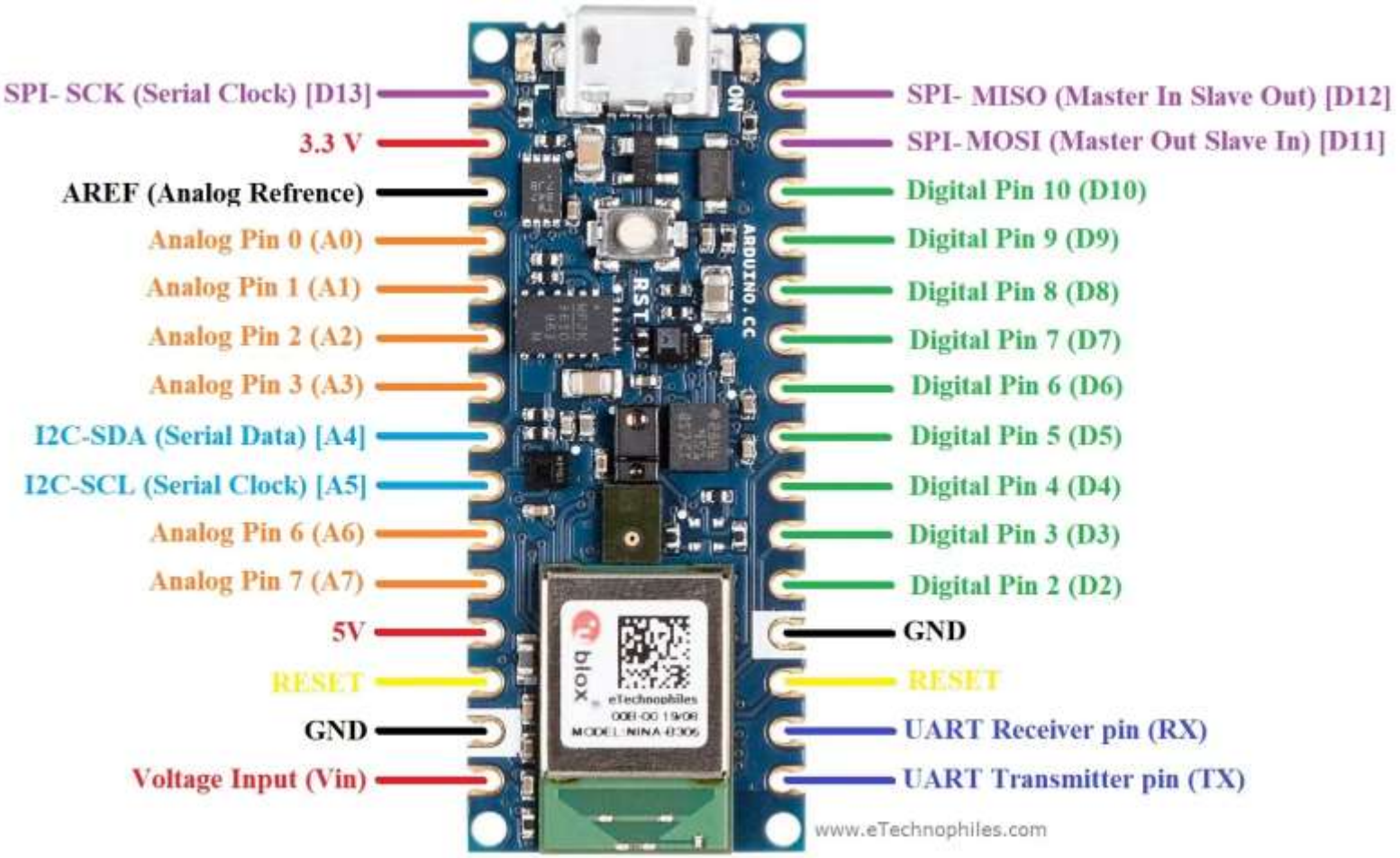
https://t1.daumcdn.net/cfile/tistory/99D8E84E5F93D2B109

nano33BLE sensor



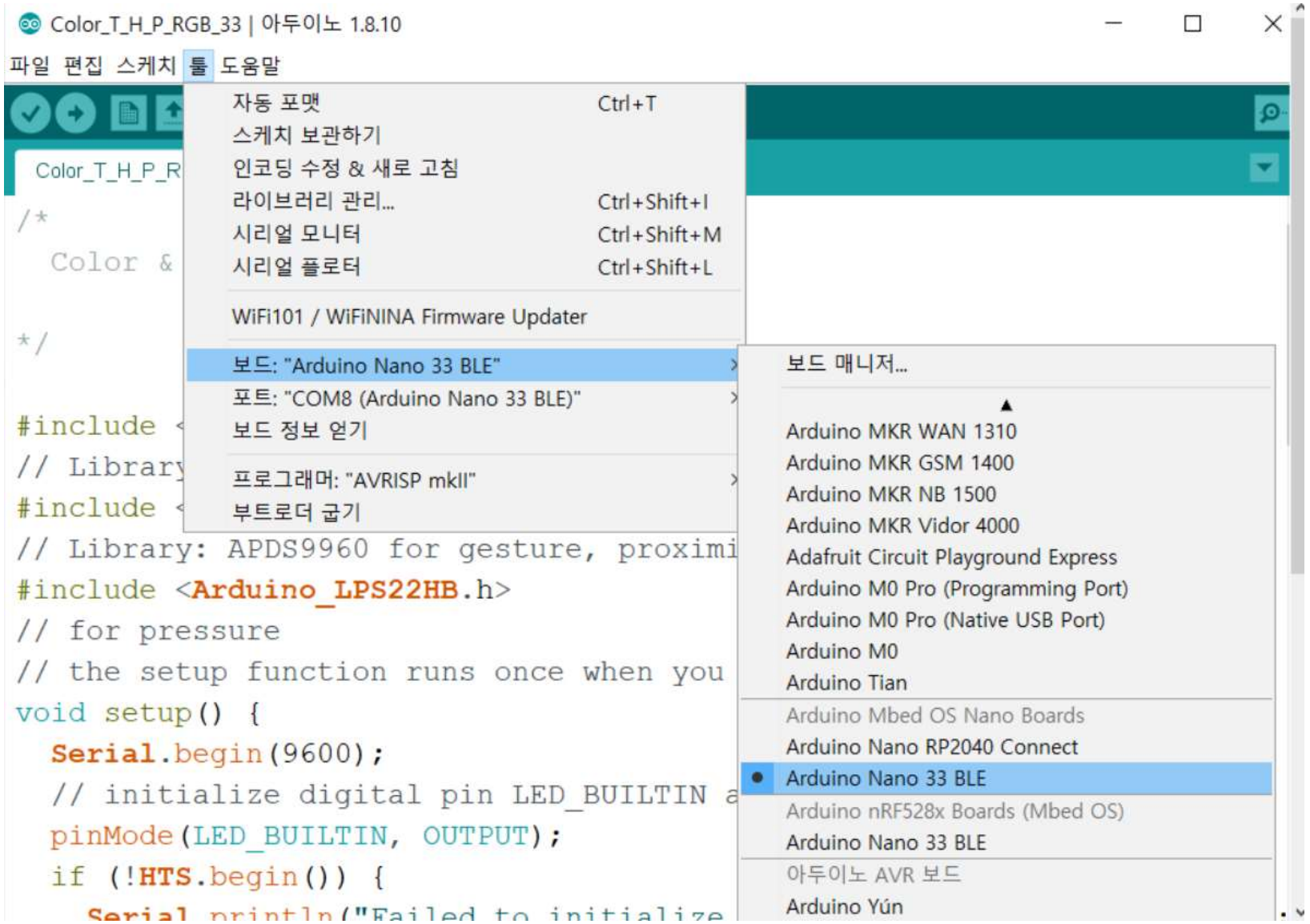
이미지 출처: https://circuitdigest.com/microcontroller-projects/arduino-nano-33-ble-sense-board-review-and-getting-started-guide

nano33BLE sensor

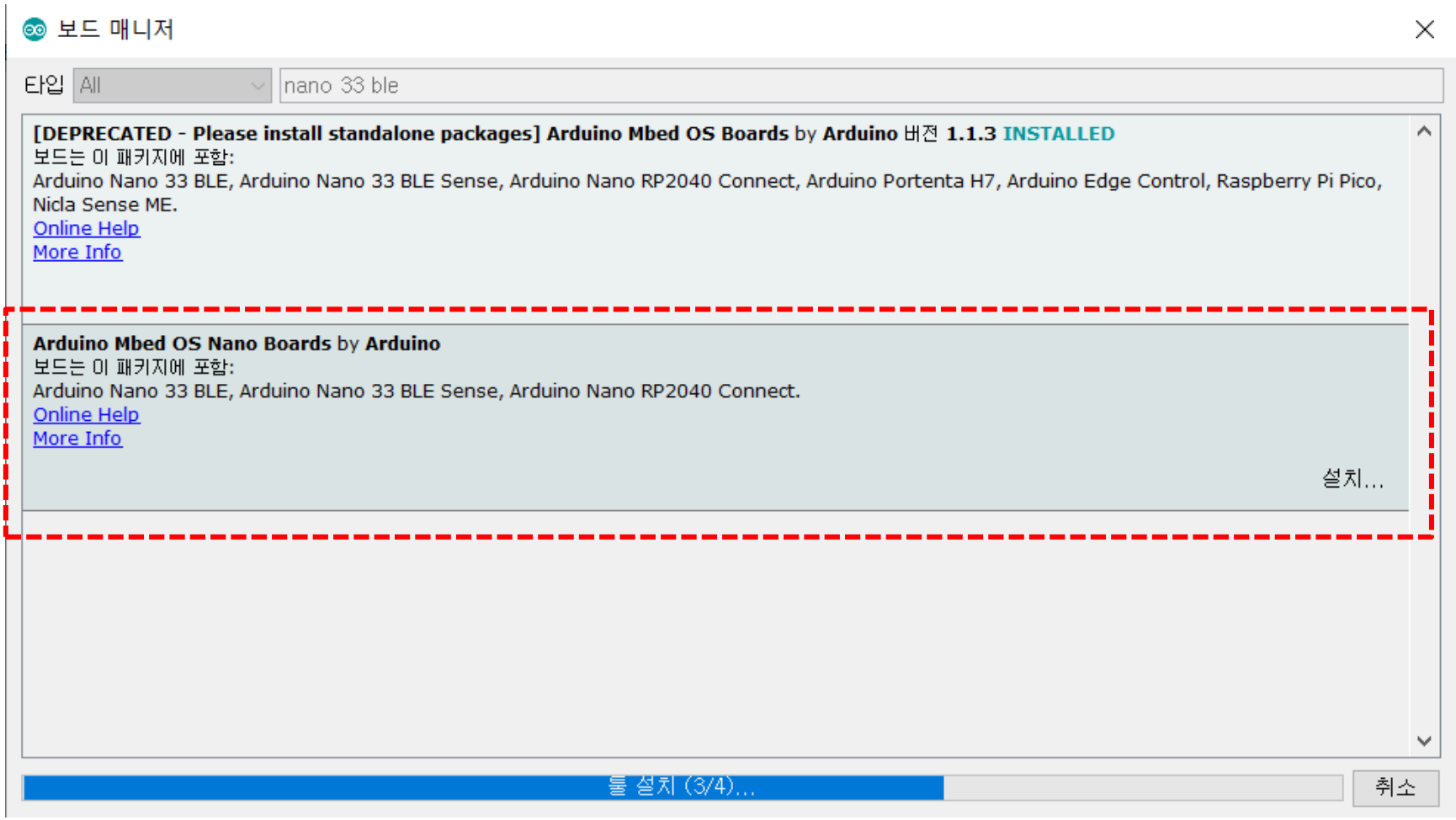


이미지 출처: https://www.etechnophiles.com/arduino-nano-33-ble-sense-pinout-introduction-specifications/

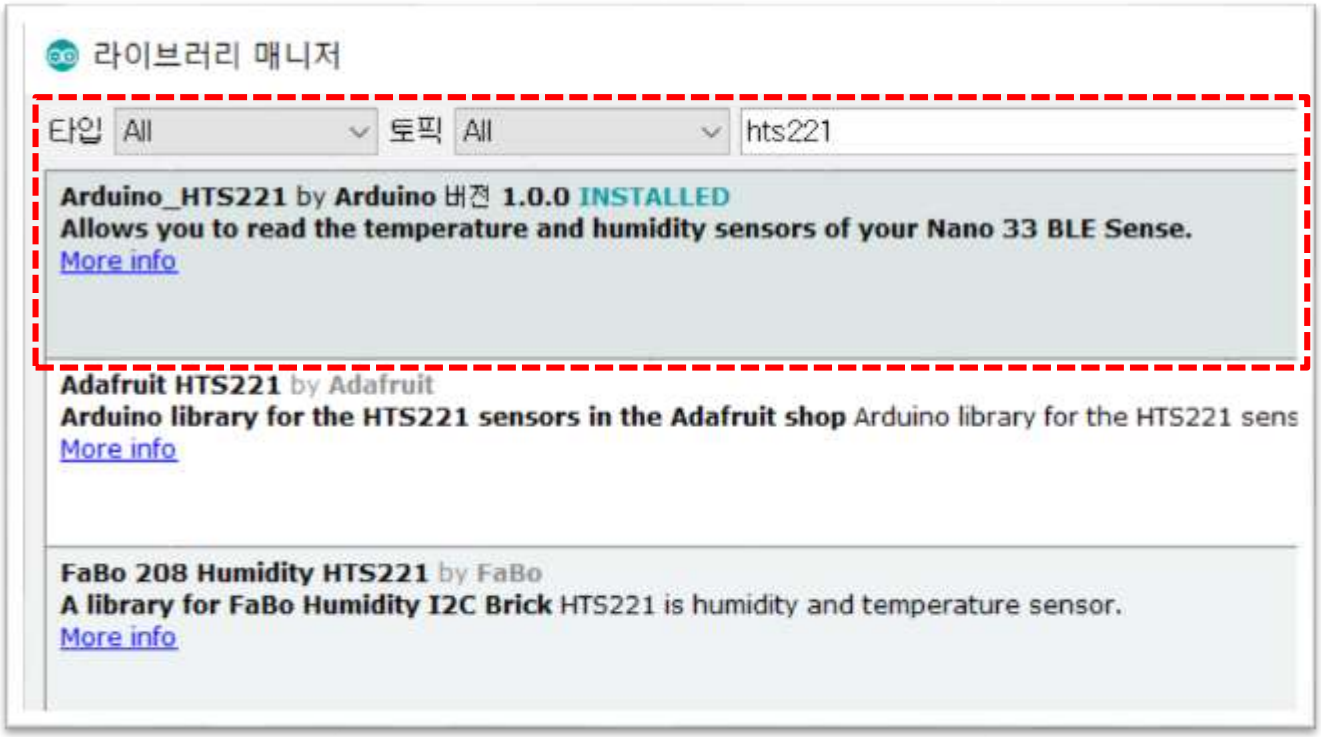
Layout [H S C]


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()) {

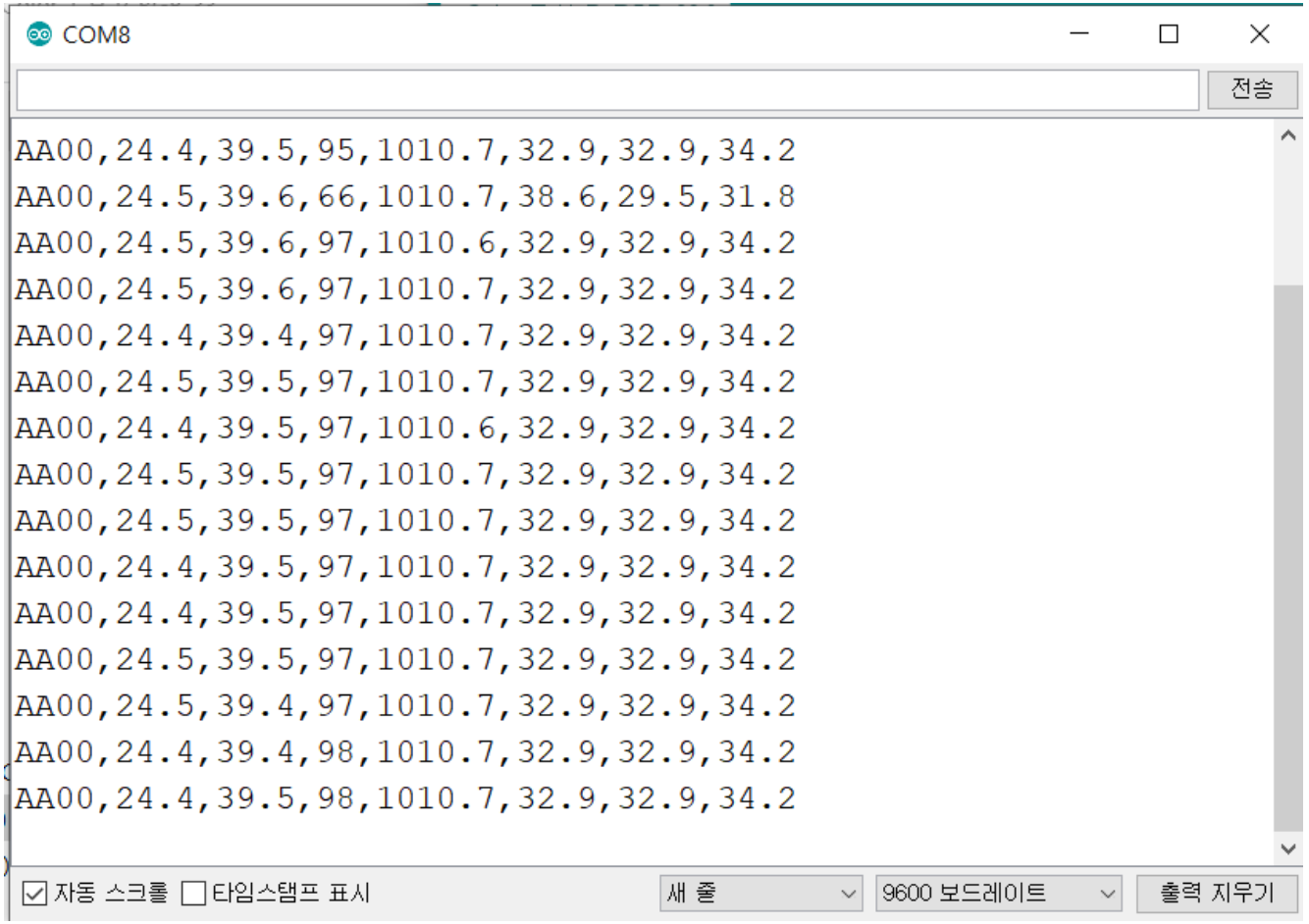
Nono33 board



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>
```

Project: nano33BE 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: nano33BE 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-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
```

Project: nano33BE 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

문제 출력 디버그 콘솔 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.
```

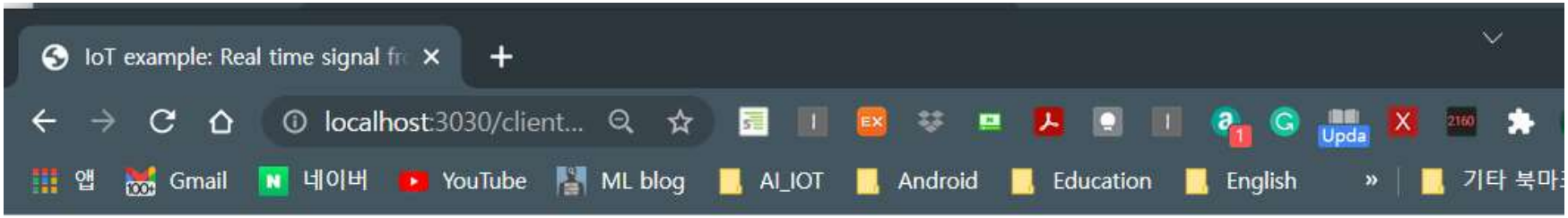
node

node

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 }
>
```

http://localhost:3030/client_signal.html



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)

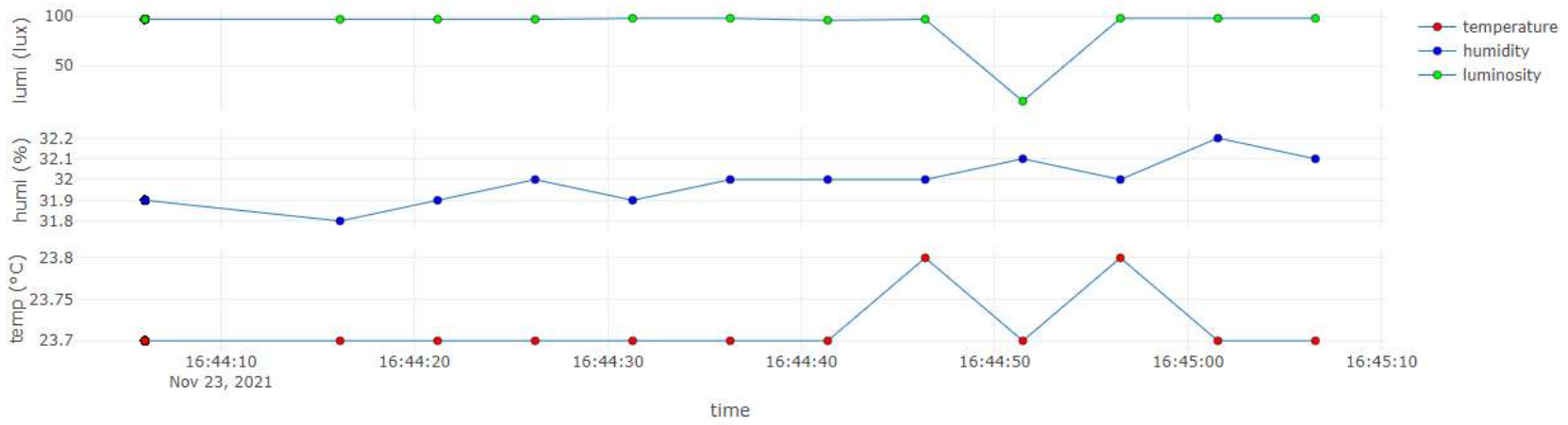
---

http://localhost:3030/client_33.html

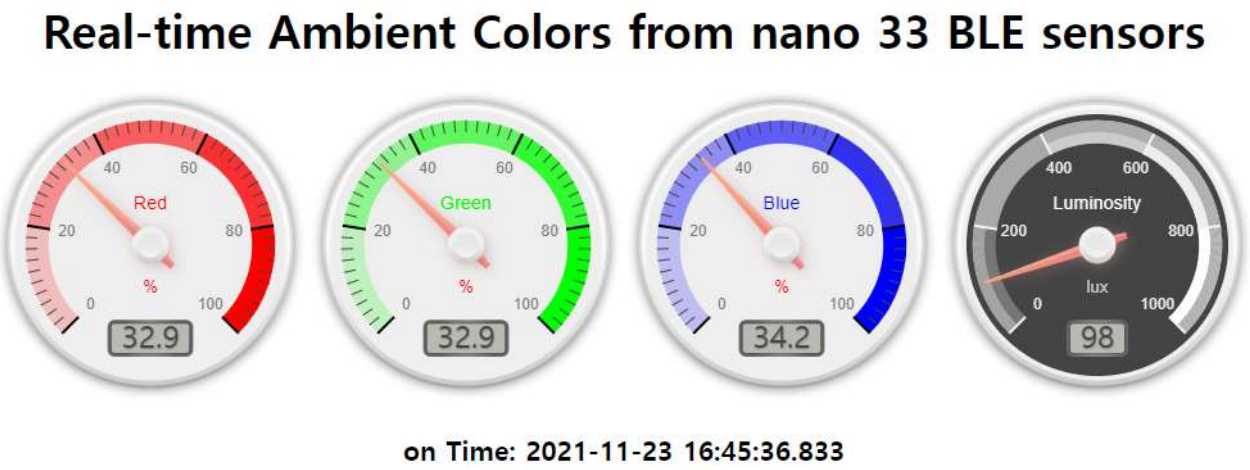
Real-time Weather Station from nano 33 BLE sensors

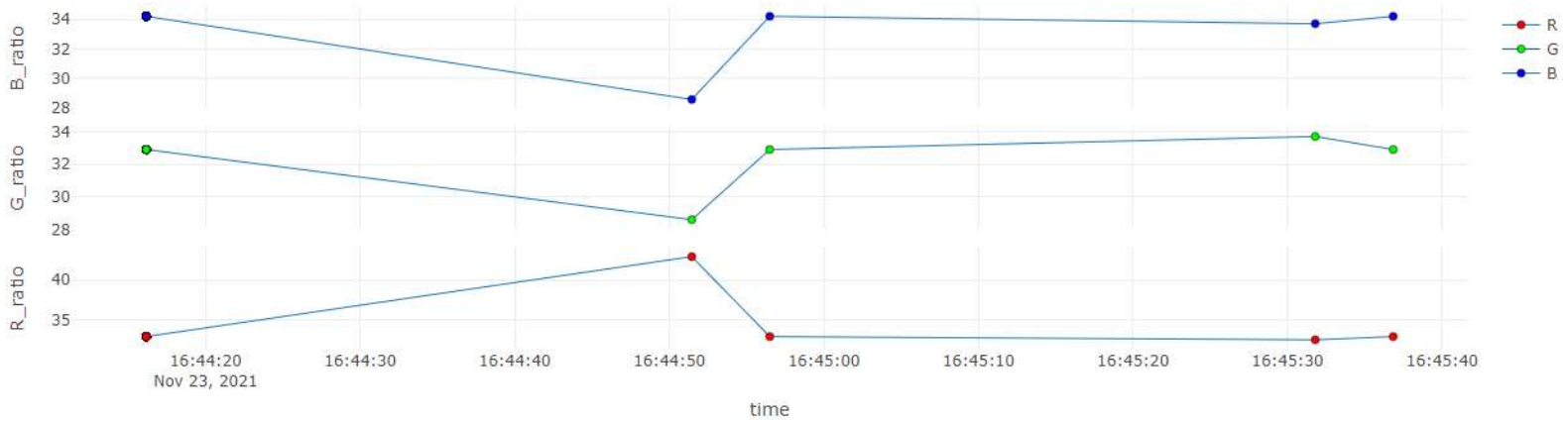
on Time: 2021-11-23 16:45:06.581

→ 압력 그래프를 4번째 axis로 추가!



http://localhost:3030/client_33rgb.html

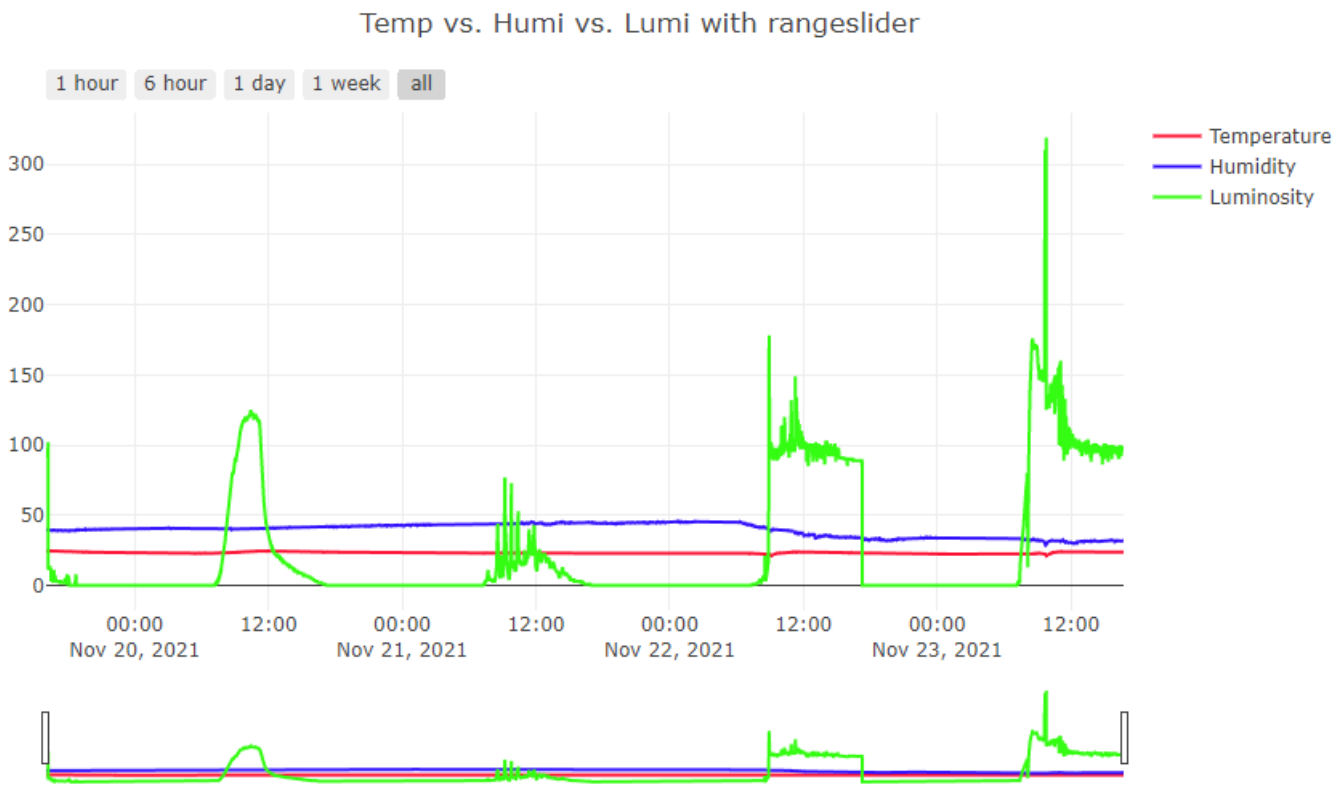




http://localhost:3030/client_33iot.html

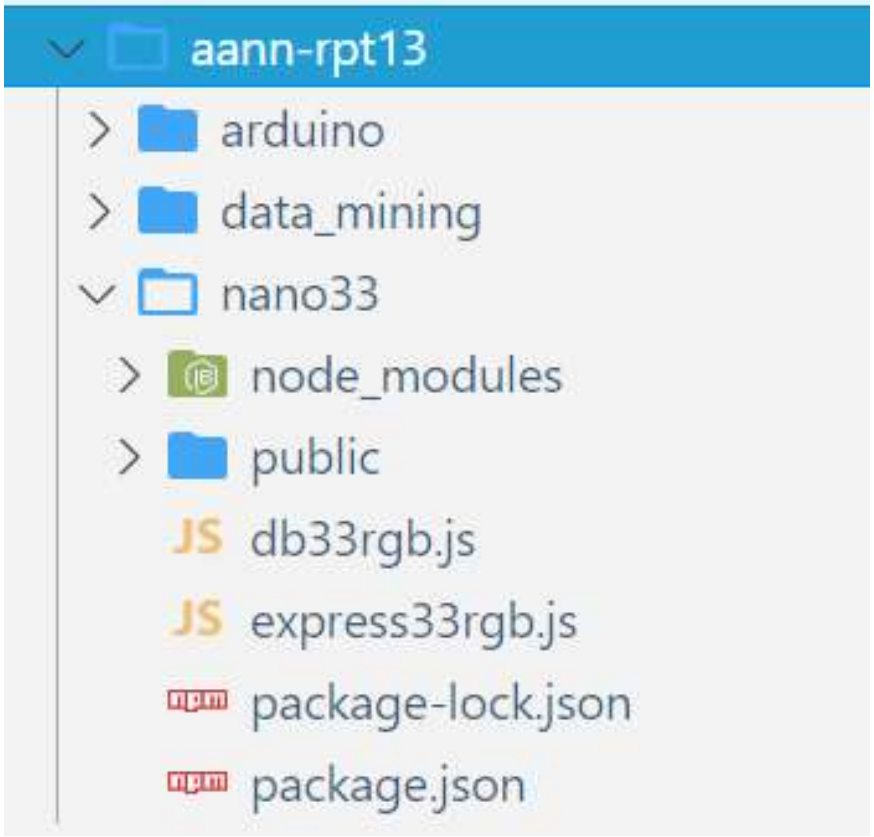
MongoDB database visualization by AA00

Time series : Multiple data from nano 33 ble sensor



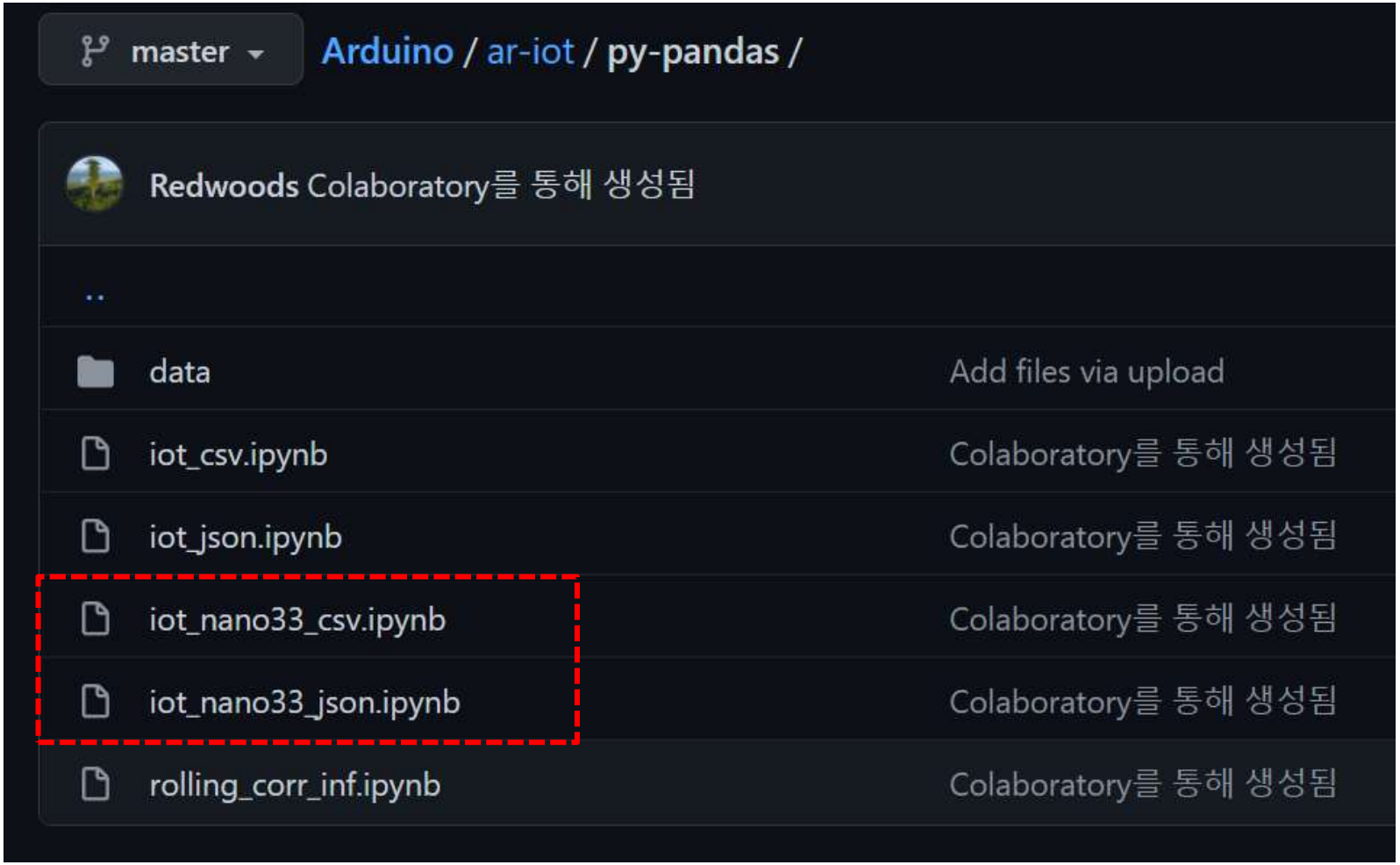
Project: nano33BLE sensor

작업 폴더 구조 [2021-project]

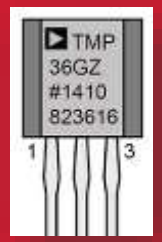


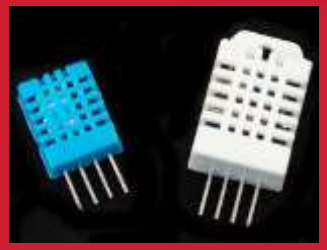
Project: nano33BLE sensor

[2021-project] IoT data mining in Colab



[Practice]





- [wk13]
  - IoT Project: nano33ble
  - Multi-sensor circuits)
  - 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 **iot_nano33_csv.ipynb**

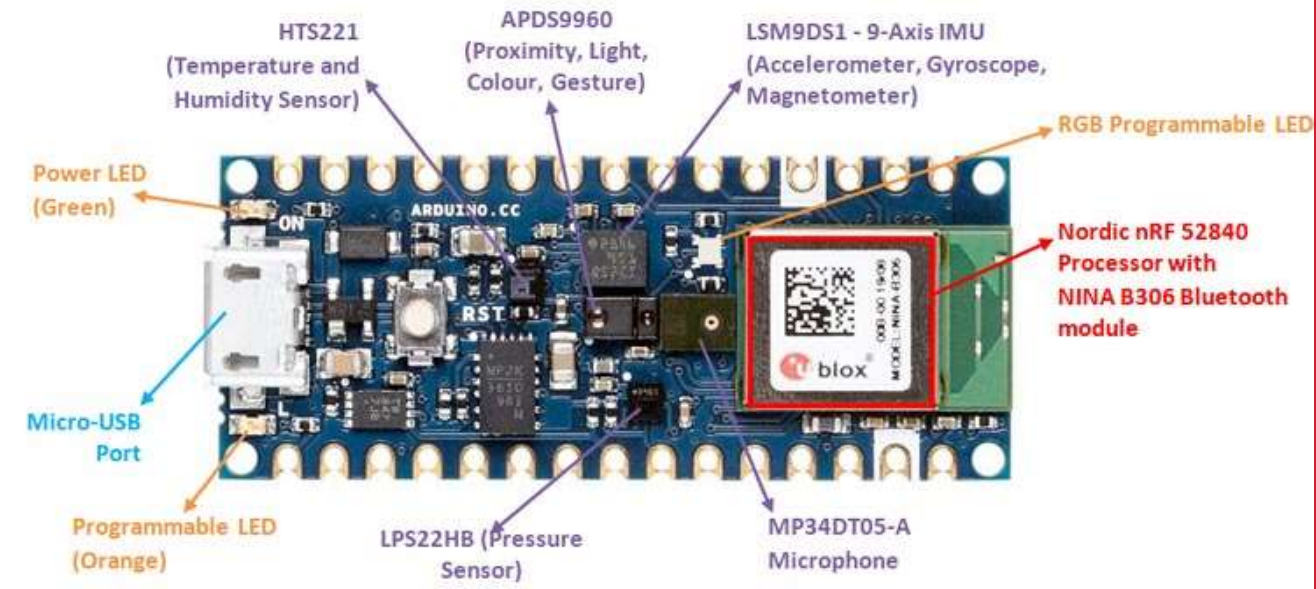
② **iot_nano33_json.ipynb**

③ **All *.js**

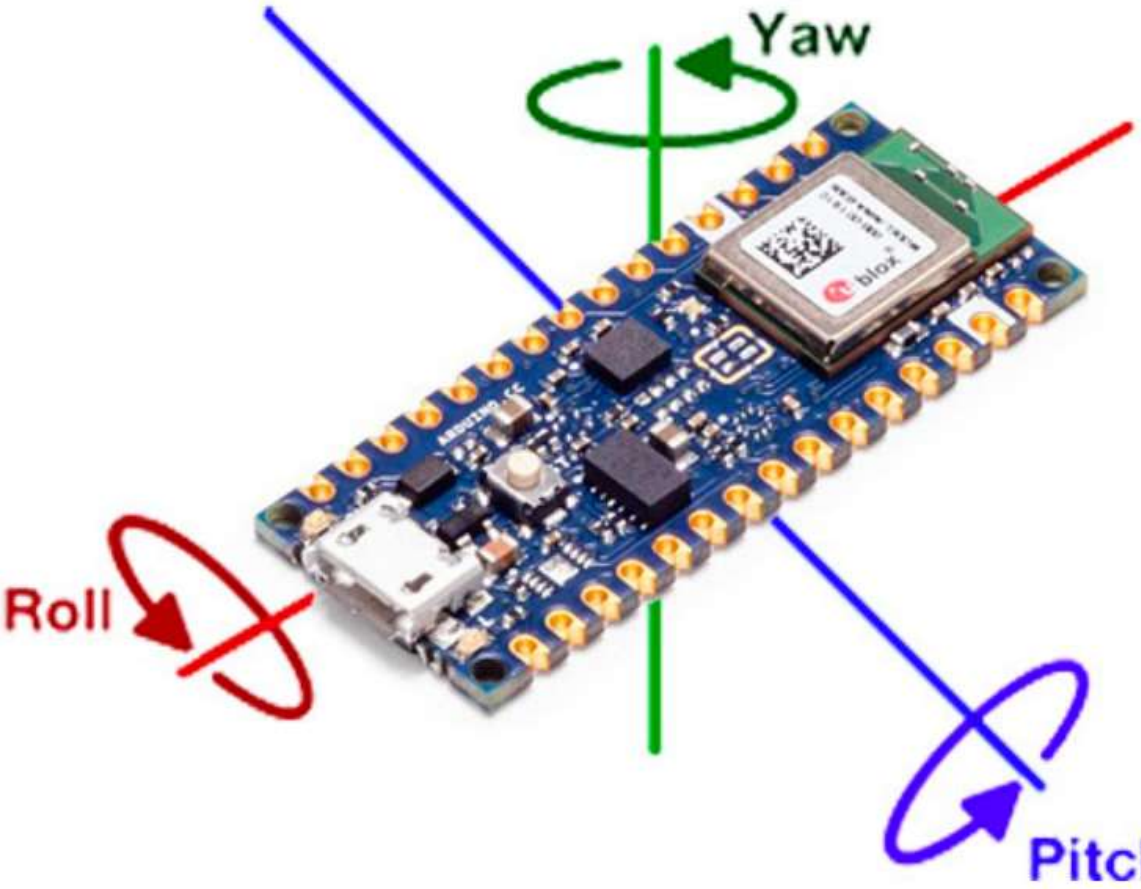
④ **public/All *.html**

2021 AA team project

nano33BLE sensor



LSM9DS1, 9축 IMU센서: acc, gyro, mag



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

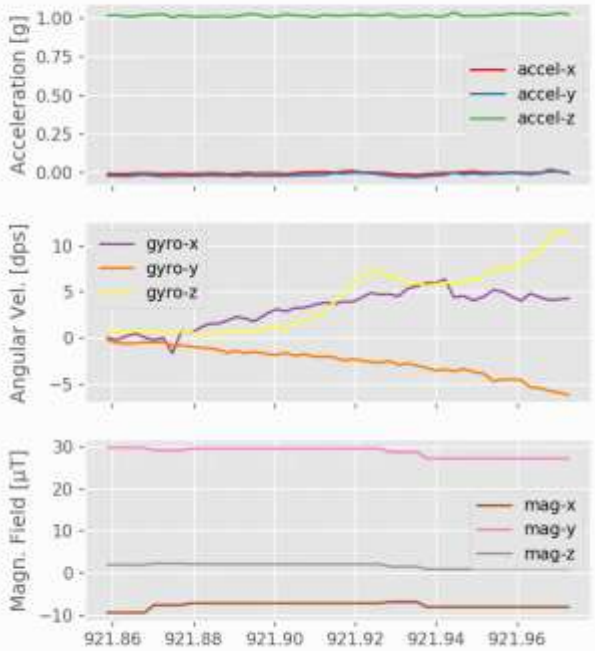
https://fishpoint.tistory.com/4438

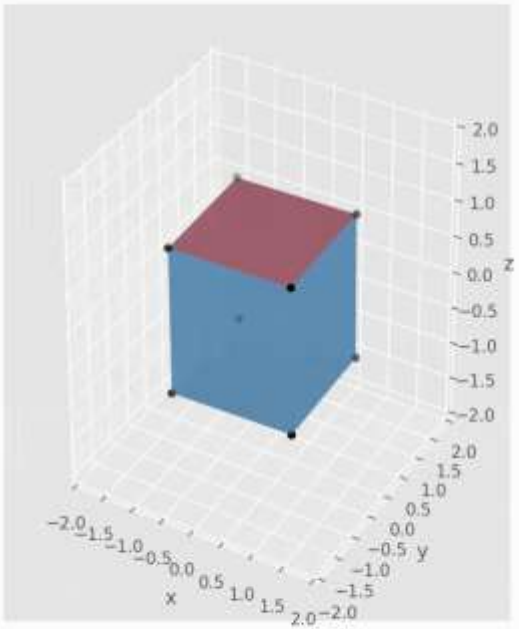
LSM9DS1, 9축 IMU센서: acc, gyro, mag

* 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)

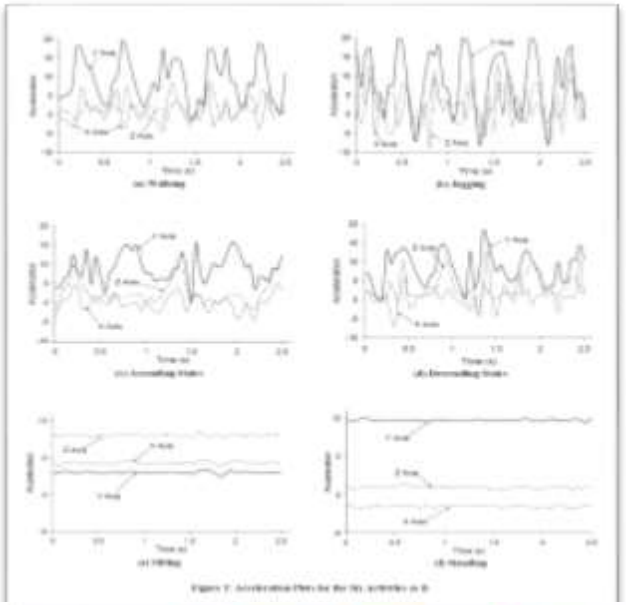


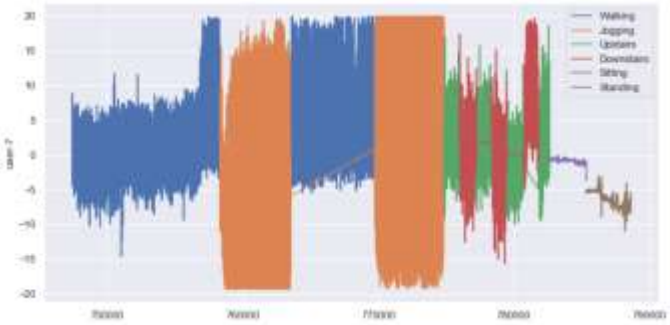


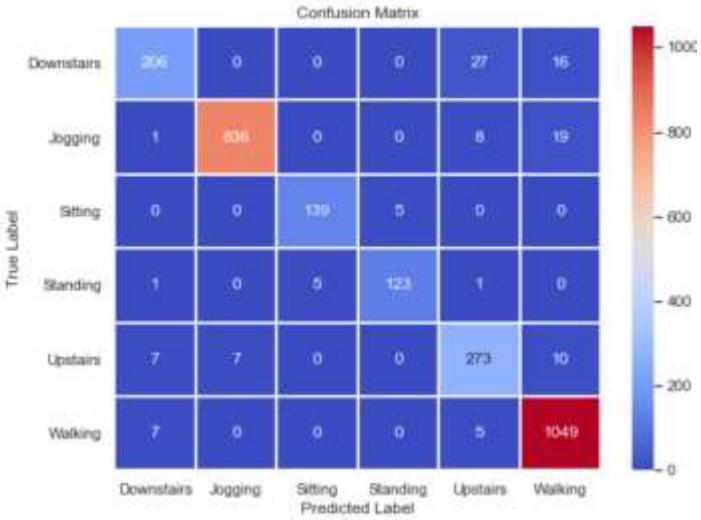
https://images.squarespace-cdn.com/content/v1/59b037304c0dbfb092fbe894/1573836927118-IS5CS61OW9XH9HSRCMA1/ke17ZwdGBTodd8pDm48kGbFogdxZzB1B7PQq3zm9xl7gQa3H78H3Y0txjaiv_0fDoOvxcdMmMKkDsyUqMSsMWxHk725yiiHCCLfrh8O1z5QPOohDIaIeljMHgDF5CVlOqpeNLcJ80NK65_fV7S1UQupMlr7Z9cq9PZkRYtzEu3SbZmkCxOjksrEup4_K2kPH3bqxw7fF48mhrq5Ulr0Hg/mpu9250_cube_rotation_compressed.gif

일상활동 인식(3축 가속도)

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

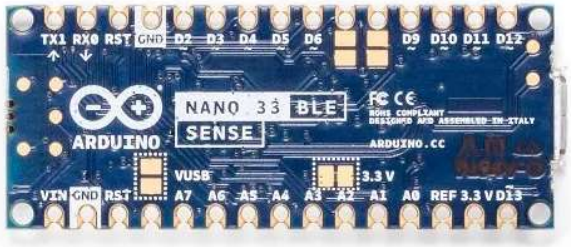






--- ACC_XYZ, 4s: classification report for test data ---

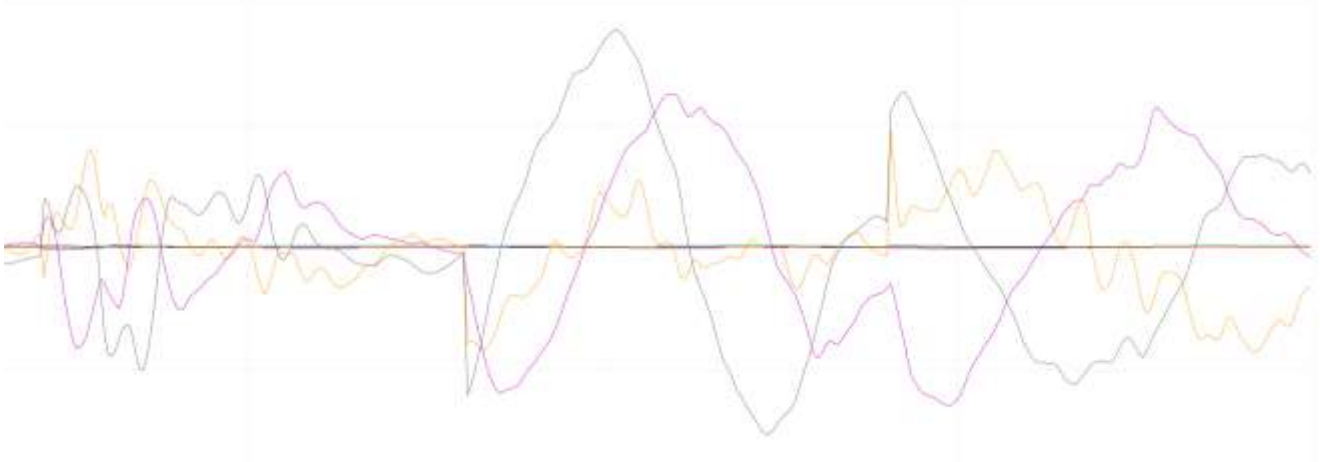
	precision	recall	f1-score	support
0	0.93	0.83	0.87	249
1	0.99	0.97	0.98	864
2	0.97	0.97	0.97	144
3	0.96	0.95	0.95	130
4	0.87	0.92	0.89	297
5	0.96	0.99	0.97	1061
accuracy			0.96	2745
macro avg	0.95	0.94	0.94	2745
weighted avg	0.96	0.96	0.96	2745



Arduino nano33 BLE

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

Data 모으기



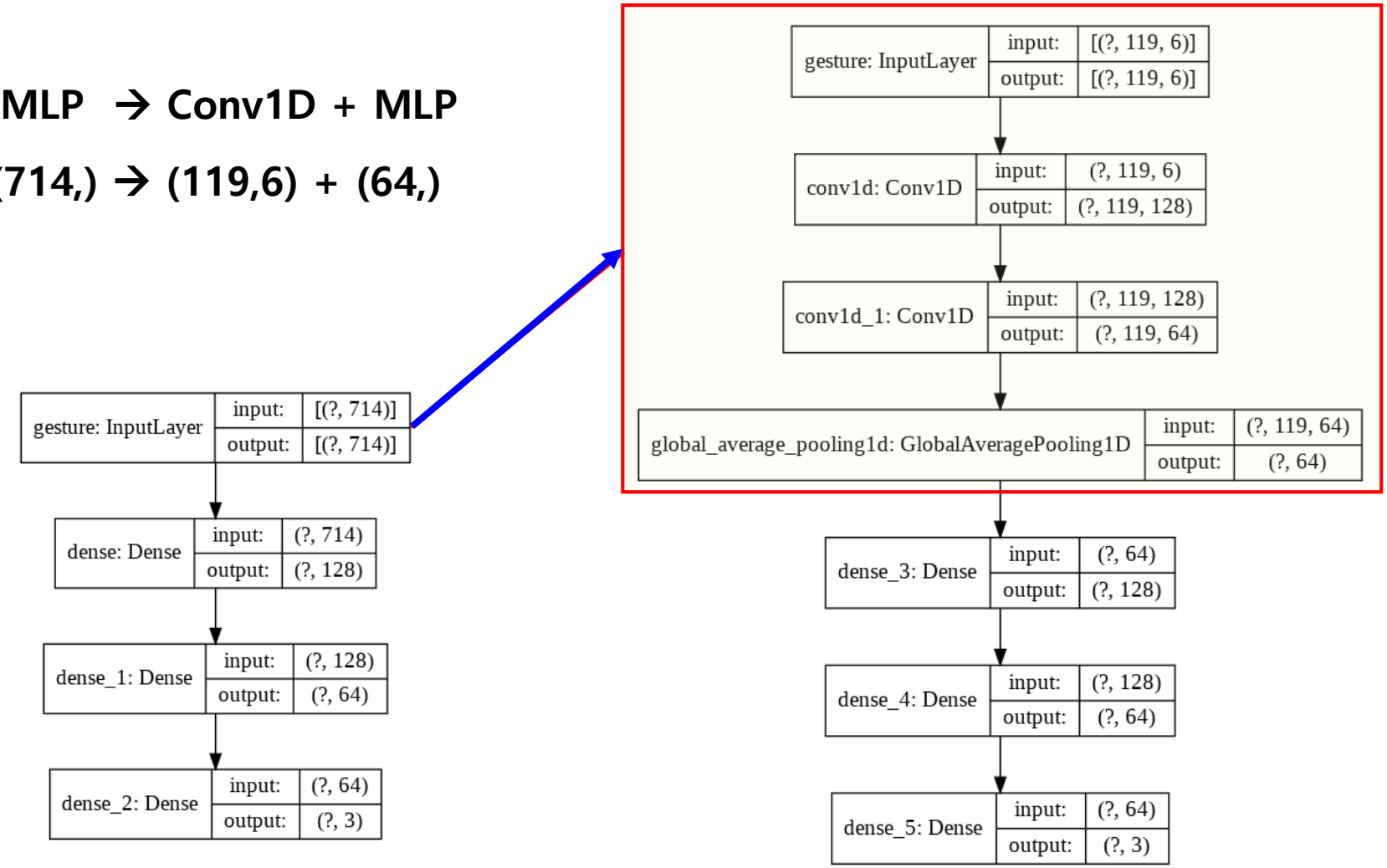
PUNCH CCW CW

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'])
```

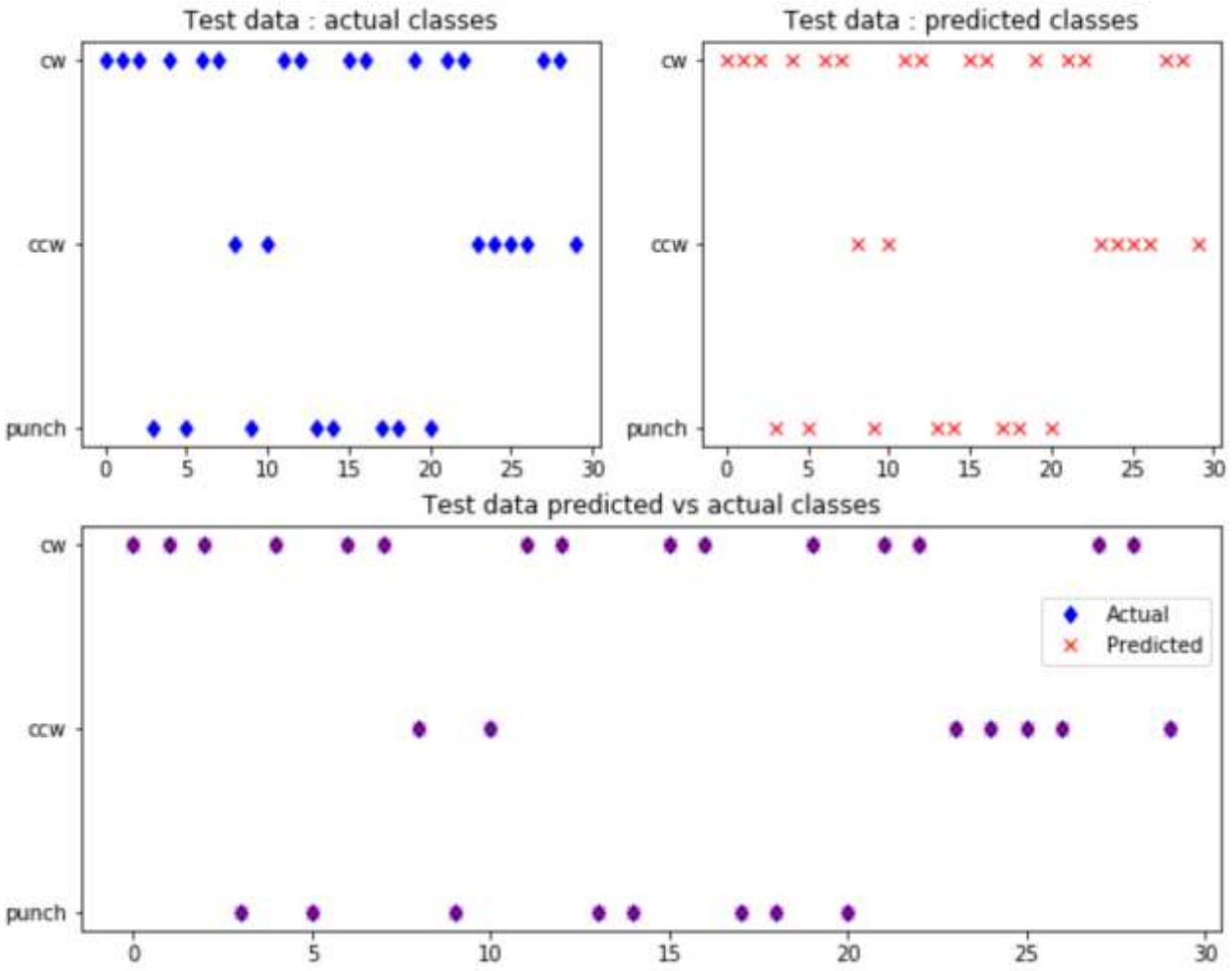
```
# train the model
history = model_conv.fit(inputs_train2, outputs_train, epochs=500, batch_size=16
                         validation_data=(inputs_validate2, outputs_validate))
```

```
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
```

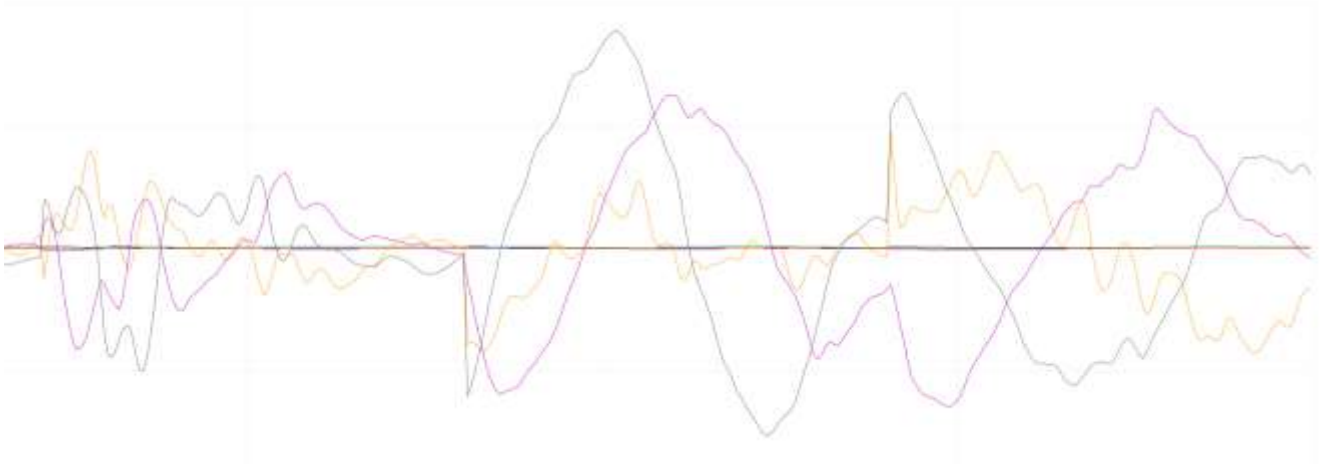
DL-model testing

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
```

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