Term Project Final Presentation

Robotics

| 2021.12.01 |

201533631 김도균
201734935 이혁수
201734939 조석영
201835404 강진호

Ⅰ. Project Objective

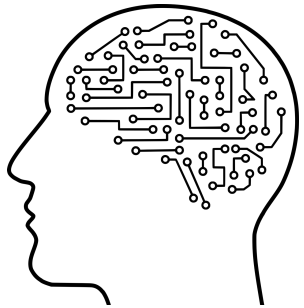
Ⅰ. Project Objective

Motivation

Motion Control

There is a lot of movie scene to control the computer with their motions

Ⅰ. Project Objective



Hardware

Arduino

Sensor to measure the angle of the finger

Software

Machine Learning

Detecting human motions

II . Key Features

II . Key Features

Motion Recognition
Hand gestures

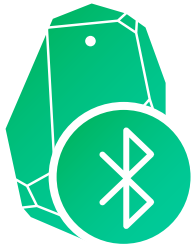
Communication
Bluetooth Low Energy

Control
Able to operate

Ⅲ. Technical Features

Ⅲ. Technical Features

Gyro Sensor
Motion Sensing



BLE Comm.
Arduino to PC

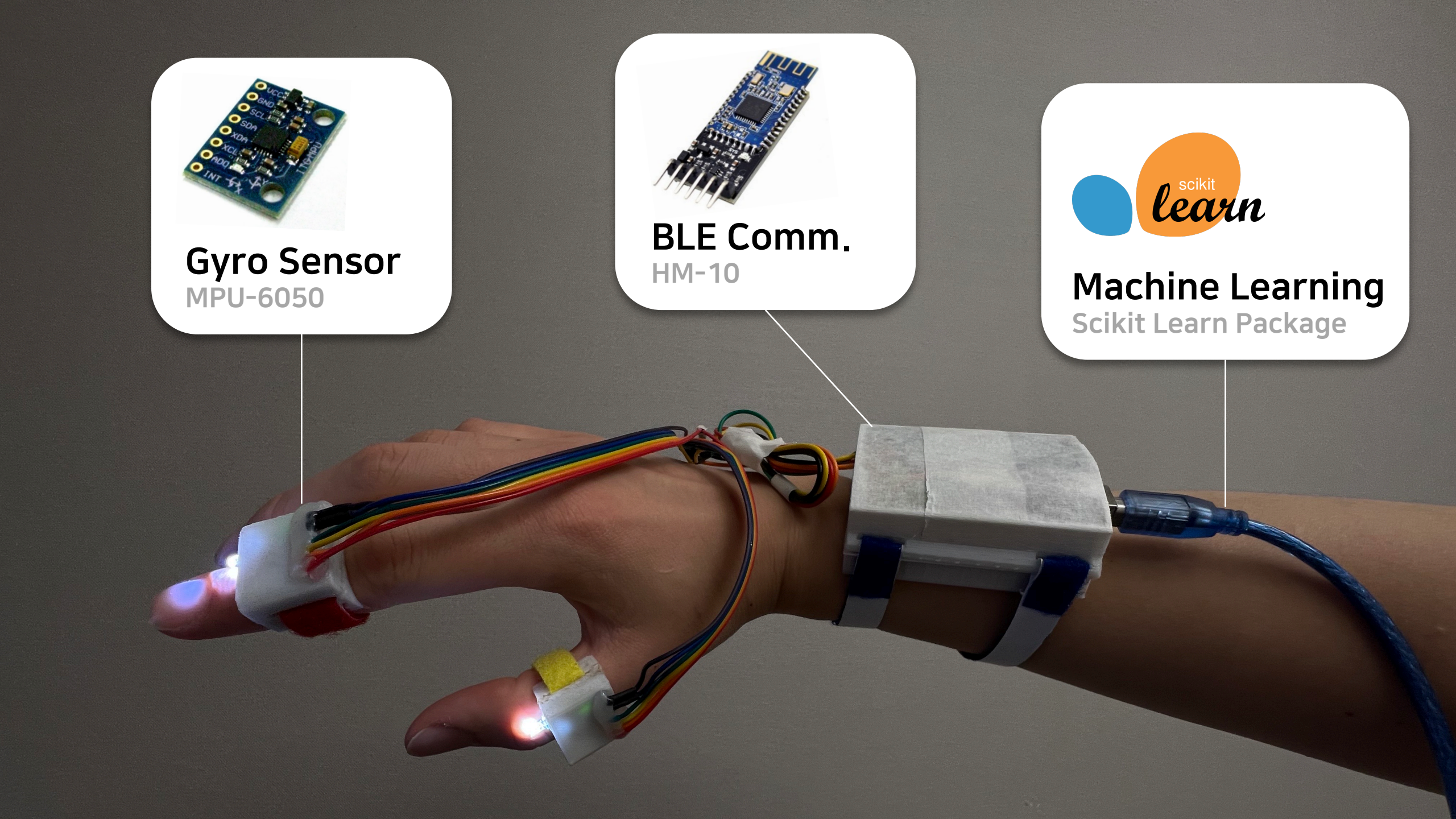
Machine Learning
Predict the specific motion

Ⅲ. Technical Features

Gyro Sensor
MPU-6050

BLE Comm.
HM-10

Machine Learning
Scikit Learn Package


Gyro Sensor
MPU-6050
BLE Comm.
HM-10
scikit
learn
Machine Learning
Scikit Learn Package

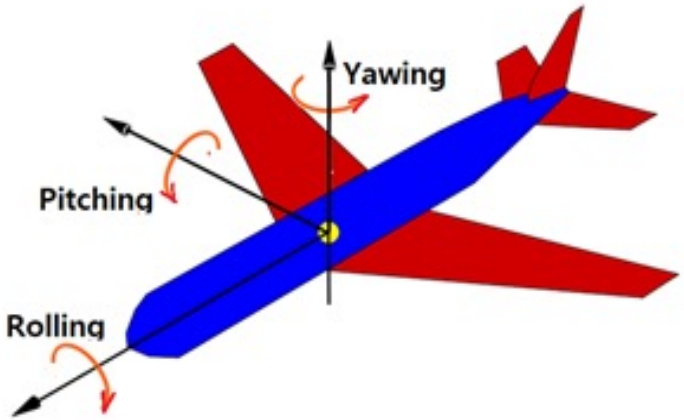
Ⅲ. Technical Features

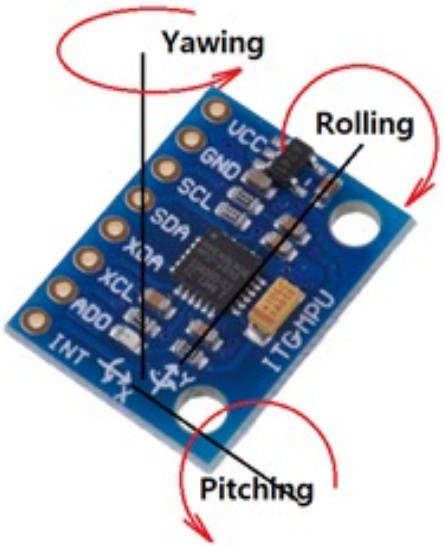
MPU 6050

Gyro Sensor

- Measure how much force based on the Earth's gravitational acceleration.
- Six physical quantities
  - 3accelerations + 3angular velocities

Ⅲ. Technical Features





3Angular Velocities

- Pitch
- Roll
- Yaw

Ⅲ. Technical Features

Gyro + Accelerometer Sensing
Get sensor value through I2C Communication

```
for (x = 0; x < num; x++){

  error1 = i2c_read(MPU6050_I2C_ADDRESS1, 0x43, i2cData, 6);
  error2 = i2c_read(MPU6050_I2C_ADDRESS2, 0x43, i2cData, 6);
  if(error1!=0) return;
  if(error2!=0) return;

  xSum += ((i2cData[0] << 8) | i2cData[1]);
  ySum += ((i2cData[2] << 8) | i2cData[3]);
  zSum += ((i2cData[4] << 8) | i2cData[5]);
}

// Set default value (average of 500 samples)
gyrXoffs = xSum / num;
gyrYoffs = ySum / num;
gyrZoffs = zSum / num;
```

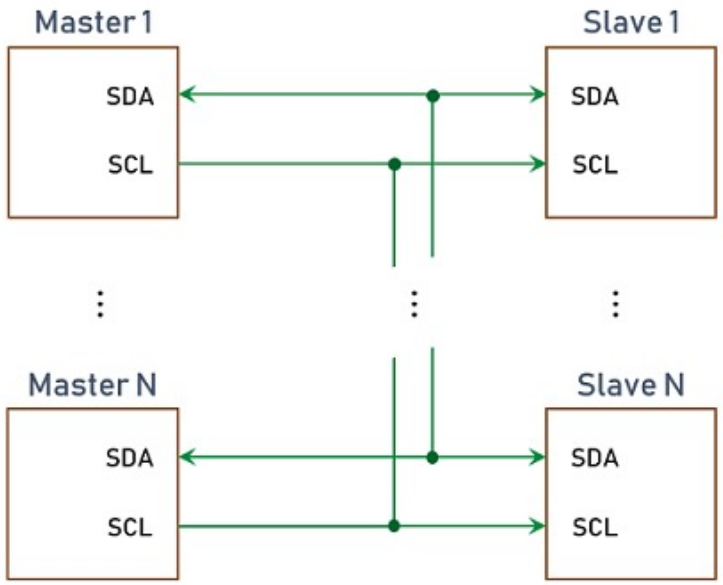
- MPU-6050 Datasheet

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT_H	R	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT_L	R	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT_H	R	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT_L	R	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT_H	R	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT_L	R	ACCEL_ZOUT[7:0]							
41	65	TEMP_OUT_H	R	TEMP_OUT[15:8]							
42	66	TEMP_OUT_L	R	TEMP_OUT[7:0]							
43	67	GYRO_XOUT_H	R	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT_L	R	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT_H	R	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT_L	R	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT_H	R	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT_L	R	GYRO_ZOUT[7:0]							

```
// assemble 16 bit sensor data

//           1       2       3       4       5       6       7       8
// i2cData = axisX_H, axisX_L, axisY_H, axisY_L, axisZ_H, axisZ_L, temp_H, temp_L,
//           gyroX_H, gyroX_L, gyroY_H, gyroY_L, gyroZ_H, gyroZ_L, N/A,    N/A

accX1 = ((i2cData1[0] << 8) | i2cData1[1]);
accY1 = ((i2cData1[2] << 8) | i2cData1[3]);
accZ1 = ((i2cData1[4] << 8) | i2cData1[5]);

accX2 = ((i2cData2[0] << 8) | i2cData2[1]);
accY2 = ((i2cData2[2] << 8) | i2cData2[3]);
accZ2 = ((i2cData2[4] << 8) | i2cData2[5]);

// (actual - offset) / sensitivity
gyrX1 = (((i2cData1[8] << 8) | i2cData1[9]) - gyrXoffs) / gSensitivity;
gyrY1 = (((i2cData1[10] << 8) | i2cData1[11]) - gyrYoffs) / gSensitivity;
gyrZ1 = (((i2cData1[12] << 8) | i2cData1[13]) - gyrZoffs) / gSensitivity;

gyrX2 = (((i2cData2[8] << 8) | i2cData2[9]) - gyrXoffs) / gSensitivity;
gyrY2 = (((i2cData2[10] << 8) | i2cData2[11]) - gyrYoffs) / gSensitivity;
gyrZ2 = (((i2cData2[12] << 8) | i2cData2[13]) - gyrZoffs) / gSensitivity;
```

Ⅲ. Technical Features

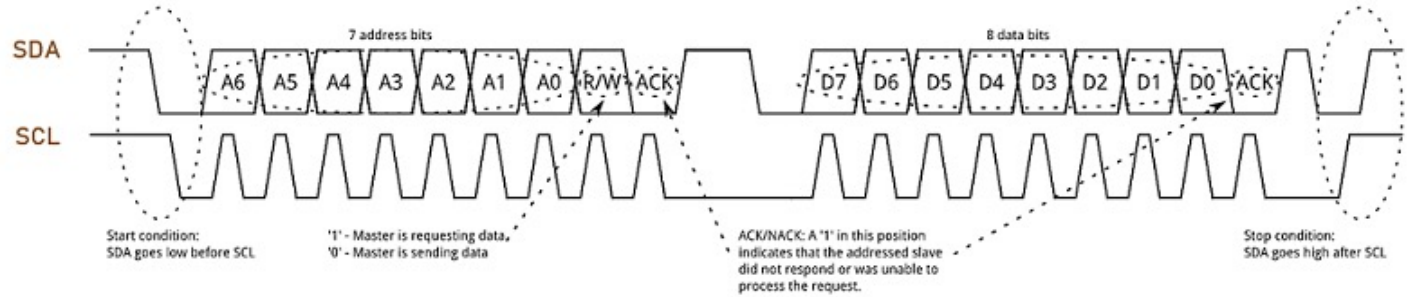
I2C Communication
Inter-Intergrated Circuit



<I2C wiring example>

```
#define MPU6050_I2C_ADDRESS1 0x68
#define MPU6050_I2C_ADDRESS2 0x69
```

```
accX1 = ((i2cData1[0] << 8) | i2cData1[1]);
accY1 = ((i2cData1[2] << 8) | i2cData1[3]);
accZ1 = ((i2cData1[4] << 8) | i2cData1[5]);

accX2 = ((i2cData2[0] << 8) | i2cData2[1]);
accY2 = ((i2cData2[2] << 8) | i2cData2[3]);
accZ2 = ((i2cData2[4] << 8) | i2cData2[5]);
```



<I2C data stream example>

Ⅲ. Technical Features

Gyro + Accelerometer Sensing
Complementary Filter


```
// complementary filter
gx1 = gx1 * 0.96 + ax1 * 0.04;
gy1 = gy1 * 0.96 + ay1 * 0.04;

gx2 = gx2 * 0.96 + ax2 * 0.04;
gy2 = gy2 * 0.96 + ay2 * 0.04;
```

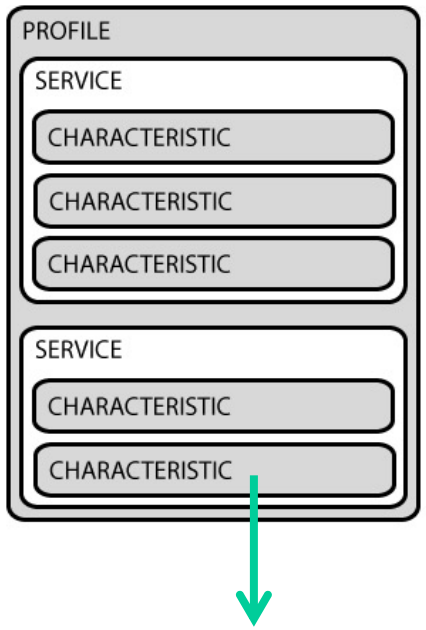
Ⅲ. Technical Features

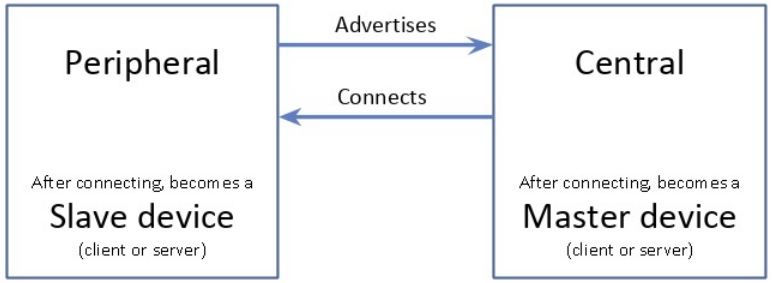
HM-10

BLE Comm.

- Communicate other devices with Bluetooth Low Enery method
- BLE = Simplicity, low power communication

III. Technical Features






```
# Device preset
device_name = "BLETEST"
device_service = "0000ffe0"
device_characteristic = "0000ffe1"
```

R / W / Notify

```
String rtn = String(round(gx1)) + "/"

Serial.println(rtn);
BTSerial.println(rtn);
```

```
# Find target device
devices = await discover()

target_device = None
target_service = None
target_characteristic = None

for d in devices:
    tmp_name = str(d.name)
    if len(tmp_name) != 0:
        if tmp_name.count(device_name) > 0:
            target_device = d

print(target_device)


# Find target BLE service
client = BleakClient(target_device.address)
print(client)
await client.connect()

services = await client.get_services()
for s in services:
    if str(s.uuid).count(device_service) > 0:
        target_service = s
        break

print(target_service)


# Find target characteristics
for c in target_service.characteristics:
    if str(c.uuid).count(device_characteristic) > 0:
        target_characteristic = c
        break

print(target_characteristic)
```

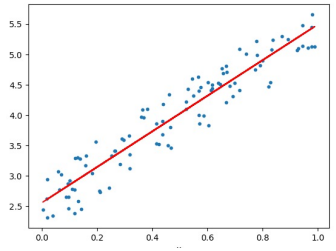
Ⅲ. Technical Features

Machine Learning

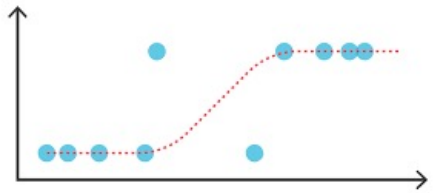
- Motion detecting with pattern of angle of fingers
- Various model was used for evaluating high score model

III. Technical Features

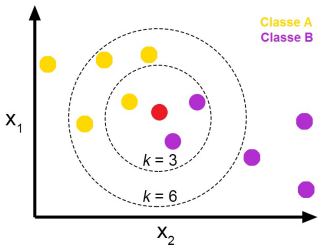
Linear Regression
50.04%

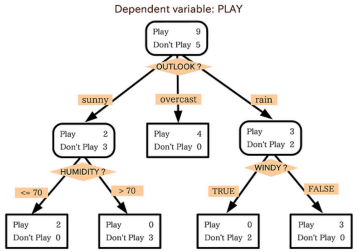


Logistic Regression
96.78%



k Nearest Neighbors
99.76%





Decision Tree Classification
100.00% (Gini)

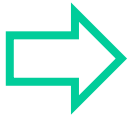
Decision Tree Classification
100.00% (Entropy)

Maybe overfitted
because of lack of sample data

Ⅲ. Technical Features

Training Pattern Data
Label: 1,2 (for each motion)

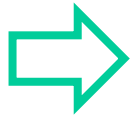
Dummy Data
Label: 0



Training model
Decision Tree Classification

Extract Model Data
to .pkl file

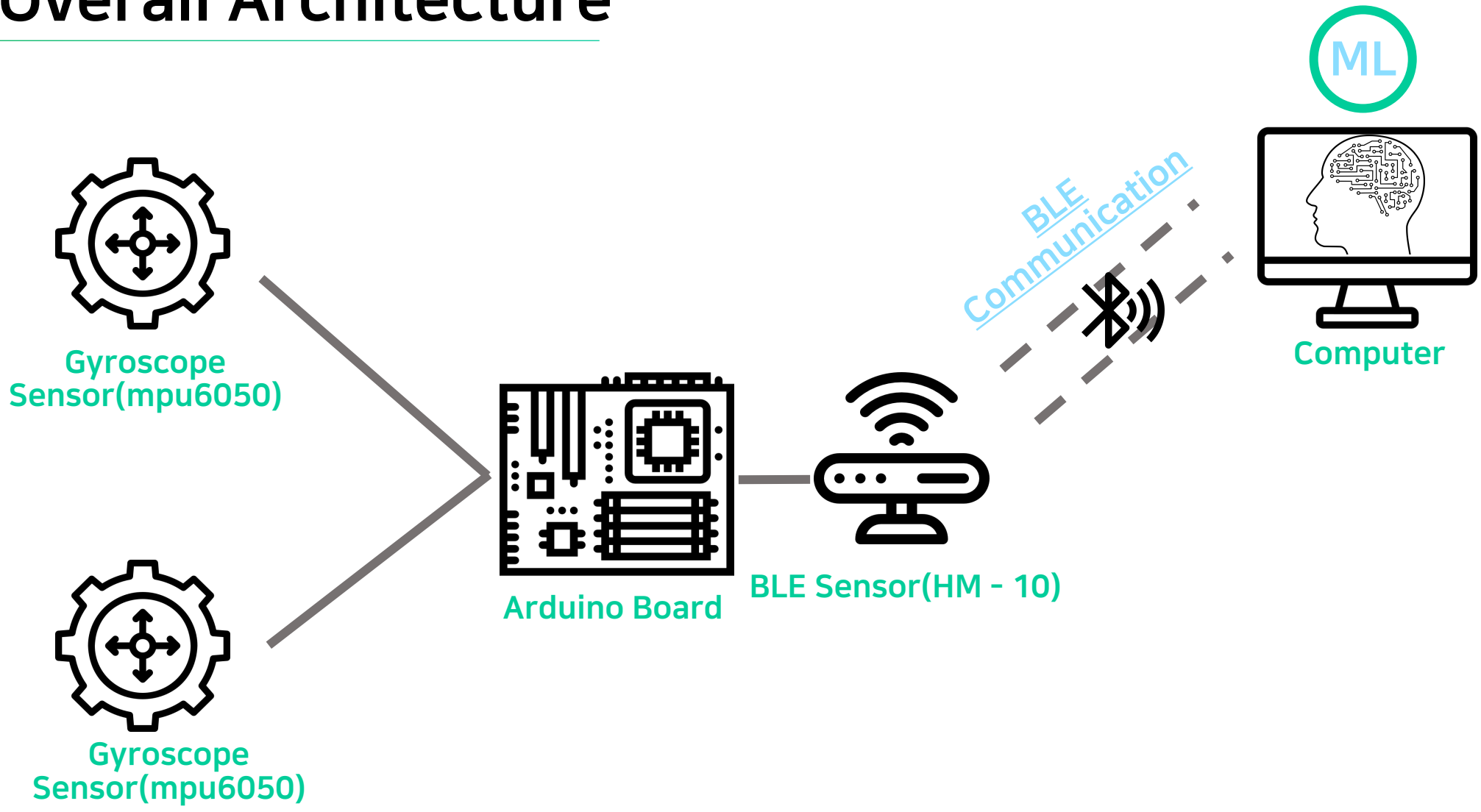
```
dtcg = DecisionTreeClassifier(criterion="gini")
dtcg.fit(X, y)

# Save model to serialized file
saved_model = joblib.dump(dtcg, "savedmodel.pkl")
print(saved_model)
```



Import Model File
.pkl file to predict motion

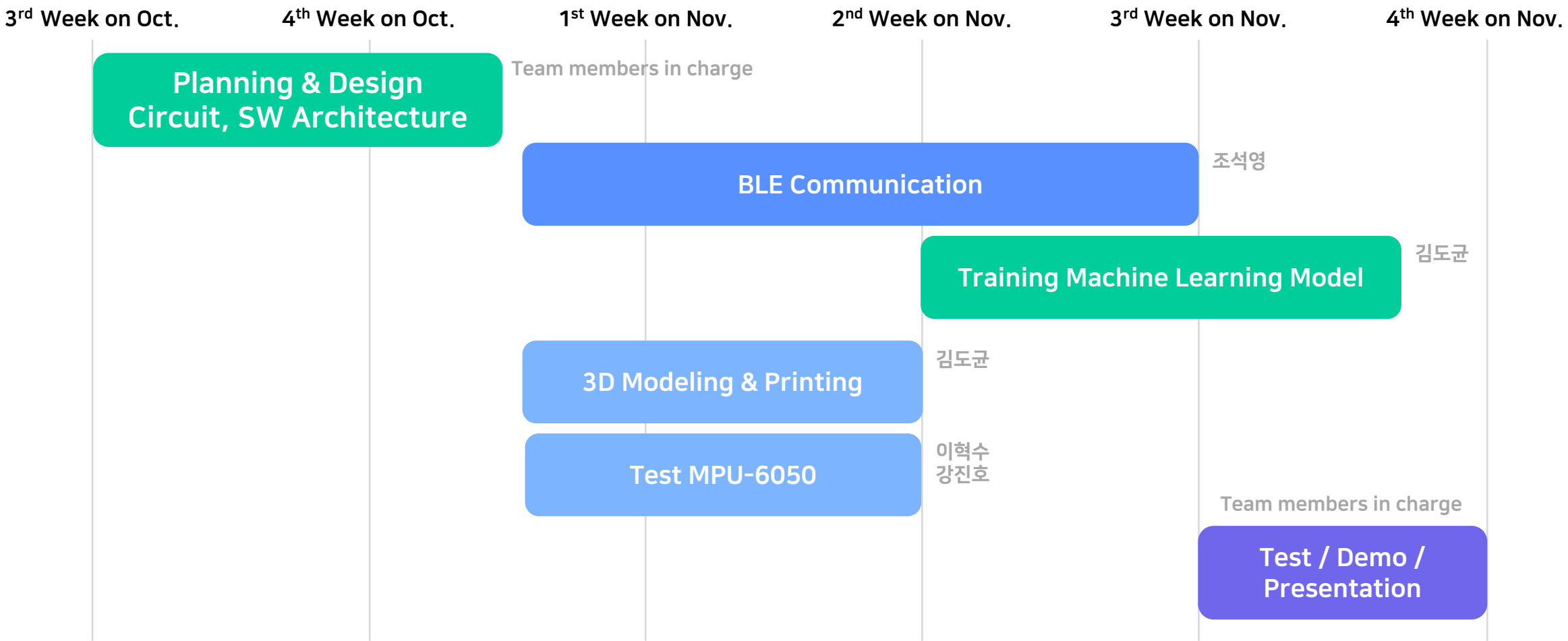
Keyboard Control
Through pynput package

Ⅲ. Overall Architecture



Ⅳ. Technical Features

Ⅳ. Progress of Project



V. Team Information

V. Team Members

Dokyoon Kim

Implement Auto ML, Program Structure

uhug@gachon.ac.kr

Hyeoksu Lee

Implement & Classification & Clustering

sg9806000@gachon.ac.kr

Seokyoung Cho

Preprocessing, Analyze Model Result

whtjrdud12@gachon.ac.kr

Jinho Kang

Clustering, Documentation

----@gachon.ac.kr

Term Project Final Presentation

Robotics

| 2021.12.01 |

201533631 김도균
201734935 이혁수
201734939 조석영
201835404 강진호