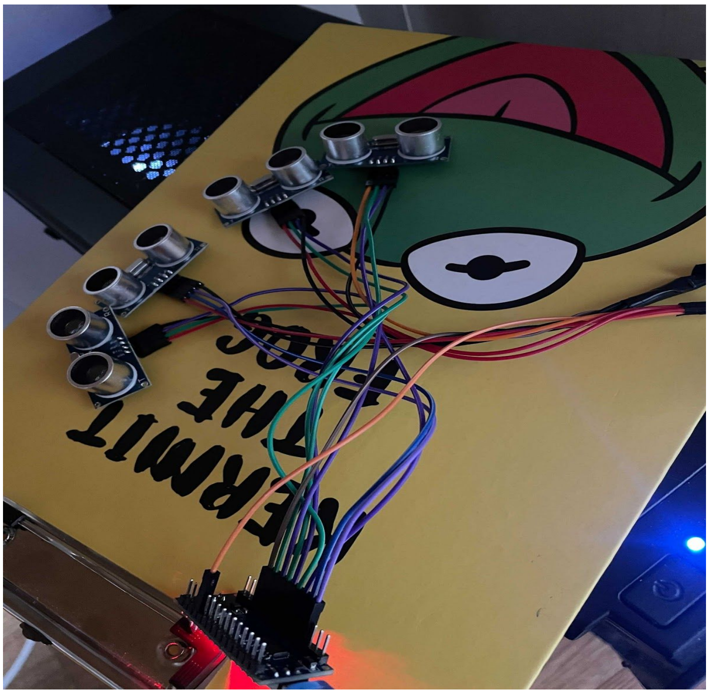
초음파 센서 과제

조정민

1. 아두이노에 4개의 초음파 센서 연결



2. 코드 작성 (ultrasonic_4_fw.ino)
    - 4개의 초음파 센서의 신호를 보낸후 측정한 것을 출력한다.

```
/*
* HC-SR04 ultrasonic FW
*/

#define trig1 2 // define trig pin
#define echo1 3 // define echo pin
#define trig2 4 // define trig pin
#define echo2 5 // define echo pin
#define trig3 6 // define trig pin
#define echo3 7 // define echo pin
#define trig4 8 // define trig pin
#define echo4 9 // define echo pin
void setup()
{
   // start serial with 9600bps speed
   Serial.begin(9600);
   // define trig pin to output
```

```
    pinMode(trig1,OUTPUT);
    // define echopin to input
    pinMode(echo1,INPUT);
    // define trig pin to output
    pinMode(trig2,OUTPUT);
    // define echopin to input
    pinMode(echo2,INPUT);
    // define trig pin to output
    pinMode(trig3,OUTPUT);
    // define echopin to input
    pinMode(echo3,INPUT);
    // define trig pin to output
    pinMode(trig4,OUTPUT);
    // define echopin to input
    pinMode(echo4,INPUT);


}

void loop()
{
    long duration1, distance1; // def var for distance
    long duration2, distance2; // def var for distance
    long duration3, distance3; // def var for distance
    long duration4, distance4; // def var for distance
    // print purse during 10us
    digitalWrite(trig1, LOW);
    delayMicroseconds(2); //2us delay
    digitalWrite(trig1, HIGH);
    delayMicroseconds(10); //10us delay
    digitalWrite(trig1,LOW);
    duration1 = pulseIn(echo1,HIGH);
    distance1 = duration1 * 170 / 1000;

    // print purse during 10us
    digitalWrite(trig2, LOW);
    delayMicroseconds(2); //2us delay
    digitalWrite(trig2, HIGH);
    delayMicroseconds(10); //10us delay
    digitalWrite(trig2,LOW);
    duration2 = pulseIn(echo2,HIGH);
    distance2 = duration2 * 170 / 1000;
    // print purse during 10us
    digitalWrite(trig3, LOW);
    delayMicroseconds(2); //2us delay
    digitalWrite(trig3, HIGH);
    delayMicroseconds(10); //10us delay
    digitalWrite(trig3,LOW);
    duration3 = pulseIn(echo3,HIGH);
    distance3 = duration3 * 170 / 1000;

    // print purse during 10us
    digitalWrite(trig4, LOW);
    delayMicroseconds(2); //2us delay
    digitalWrite(trig4, HIGH);
    delayMicroseconds(10); //10us delay
    digitalWrite(trig4,LOW);
    duration4 = pulseIn(echo4,HIGH);
    distance4 = duration4 * 170 / 1000;

    //pulseIn() read pin sign and convert us


    //Serial.print("Distance(mm): \n");
    Serial.print(distance1);
    Serial.print("mm ");
    Serial.print(distance2);
    Serial.print("mm ");
    Serial.print(distance3);
    Serial.print("mm ");
    Serial.print(distance4);
    Serial.print("mm\n");
    delay(100);
}
```

3. 코드 작성 (ultra4_pub.py)

```
#!/usr/bin/env python2

import serial, time, rospy
from std_msgs.msg import Int32MultiArray

msg = Int32MultiArray()
```

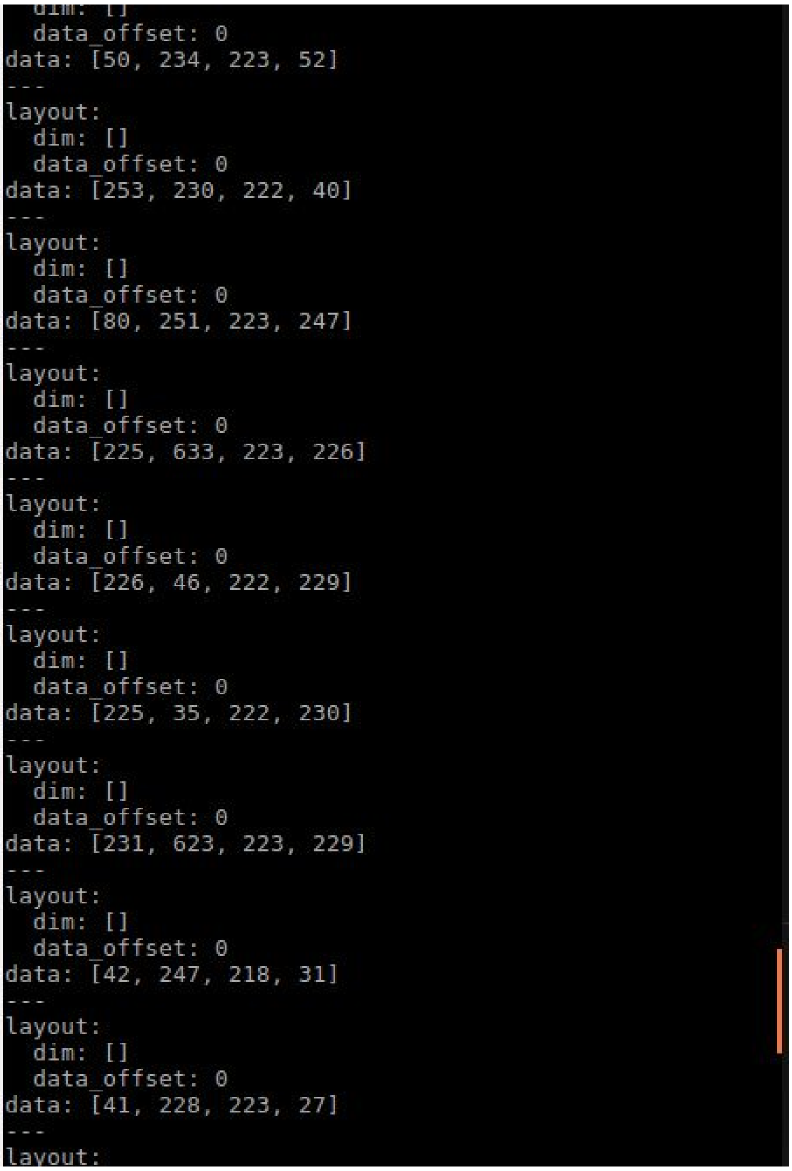
```
ser_front = serial.Serial( \
   port='/dev/ttyUSB0', \
   baudrate=9600,
)

def read_sensor():
   global msg
   serial_data = ''
   serial_data = ser_front.readline()
   # mm을 기준으로 데이터를 쪼갠다.
   ultrasonic_list = serial_data.split("mm")
   msg.data = list(map(lambda x: int(x), ultrasonic_list[:-1]))

if __name__ == '__main__':
   rospy.init_node('ultrasonic4_pub', anonymous=False)
   pub = rospy.Publisher('ultra4', Int32MultiArray, queue_size=1)
   rate = rospy.Rate(10)
   while(not rospy.is_shutdown()):
       read_sensor()
       pub.publish(msg)
       rate.sleep()
   ser_front.close()
```

4. 코드 작성 (ultra4_sub.py)

```
#!/usr/bin/env python

import rospy
from std_msgs.msg import Int32MultiArray
# 구독자 노드 생성 및 데이터 출력 콜백함수 생성
def callback(msg):
   print(msg.data)

rospy.init_node('ultrasonic4_sub')
sub = rospy.Subscriber('ultra4', Int32MultiArray , callback)

rospy.spin()
```

5. 실행 화면

```
roslaunch ultrasonic ultra4.launch
rostopic echo ultra4
```



```
  dim: []
  data_offset: 0
data: [50, 234, 223, 52]
---
layout:
  dim: []
  data_offset: 0
data: [253, 230, 222, 40]
---
layout:
  dim: []
  data_offset: 0
data: [80, 251, 223, 247]
---
layout:
  dim: []
  data_offset: 0
data: [225, 633, 223, 226]
---
layout:
  dim: []
  data_offset: 0
data: [226, 46, 222, 229]
---
layout:
  dim: []
  data_offset: 0
data: [225, 35, 222, 230]
---
layout:
  dim: []
  data_offset: 0
data: [231, 623, 223, 229]
---
layout:
  dim: []
  data_offset: 0
data: [42, 247, 218, 31]
---
layout:
  dim: []
  data_offset: 0
data: [41, 228, 223, 27]
---
layout:
```