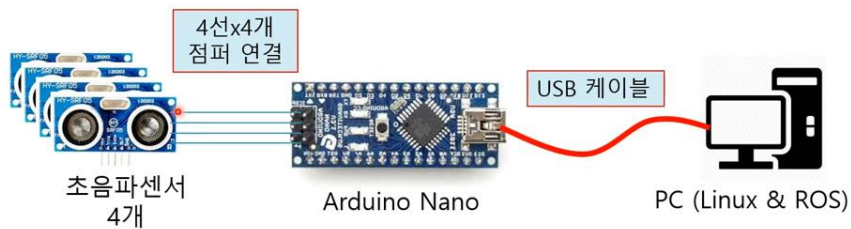


과제 설명

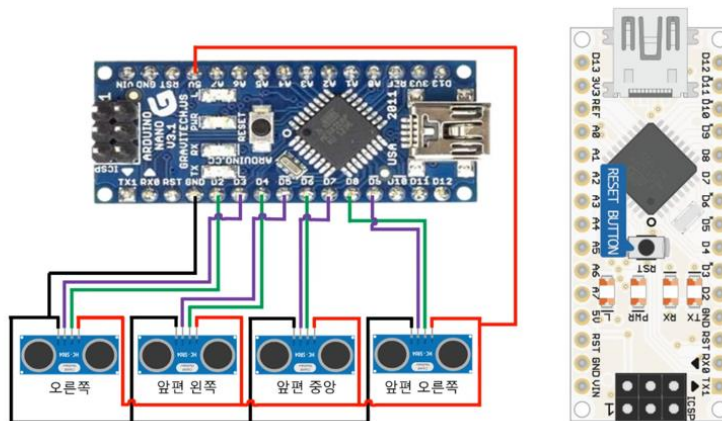
- 초음파 센서 4개를 아두이노에 연결 후 리눅스로 출력하기

- 초음파센서 4개를 지원하는 ROS 패키지 제작하기



- 아두이노 보드에 연결하기(아두이노 코드로 실행 확인 후 리눅스 실행)

- 각 센서의 Vcc, Trig, Echo, Gnd 를 각 핀에 연결



아래 코드 실행 결과

\$ roslaunch ultrasonic ultra4.launch

```
sungmin@machine: ~/catkin_ws/src/ultrasonic/src
process[ultrasonic_pub-2]: started with pid [4543]
process[ultrasonic_sub-3]: started with pid [4546]
(38, 156, 116, 0)
(38, 0, 35, 0)
(39, 0, 35, 0)
(38, 0, 35, 0)
(38, 0, 35, 0)
(38, 0, 35, 0)
```

1. ultra4_pub.py

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

```
import serial, time, rospy, re
```

```
from std_msgs.msg import Int32MultiArray
```

```
#아두이노 연결 포트 지정
```

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

```
def read_sensor():
```

```
    # 시리얼 포트에 들어온 데이터 받아오기
```

```
    serial_data = ser_front.readline()
```

```
    ser_front.flushInput()
```

```
    ser_front.flushOutput()
```

```
    # 입력 받은 한줄을 공백 기준으로 나누기
```

```
    ultra4=serial_data.split(" ")
```

```
    data=[] # 4개의 거리 값을 담은 데이터 리스트
```

```
    #4개로 분리된 값에서 mm을 제거한 나머지 숫자만 int로 변환 후 data리스트에 담기
```

```
    for i in range(4):
```

```
        ultrasonic_data = int(filter(str.isdigit, ultra4[i]))
```

```
        data.append(ultrasonic_data)
```

#발행할 최종 값

msg.data=data

if __name__ == '__main__':

rospy.init_node('ultra4_pub', anonymous=False) # initialize node

pub = rospy.Publisher('ultra4', Int32MultiArray, queue_size=1)

msg = Int32MultiArray() # message type

while not rospy.is_shutdown():

read_sensor() #센서가 보내준 문자열 읽고 거리정보 추출

pub.publish(msg) # 토픽에 담아서 발행

time.sleep(0.2)

ser_front.close() #끝나면 시리얼 포트 닫기

2. ultra4_sub.py

#!/usr/bin/env python

import rospy

from std_msgs.msg import Int32MultiArray

#구독할 때 마다 메시지 출력

def callback(msg):

print(msg.data)

rospy.init_node('ultra4_sub')

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

rospy.spin()

3. ultra4.launch

#ultra4_pub.py, ultra4_sub.py를 실행

```
<launch>
```

```
  <node pkg="ultrasonic" type="ultra4_pub.py" name="ultra4_pub"/>
```

```
  <node pkg="ultrasonic" type="ultra4_sub.py" name="ultra4_sub" output="screen"/>
```

```
</launch>
```

4. ultrasonic_4_fw

//4개의 트리거 핀과 에코 핀을 배열에 담아서 값 저장

```
int trig[4]={2,4,6,8};
```

```
int echo[4]={3,5,7,9};
```

```
long distance[4]; //4개 거리의 정보 담는 배열
```

```
void setup() {
```

```
  Serial.begin(9600);    // 통신속도 9600bps로 시리얼 통신 시작
```

```
  //4개의 트리거 및 에코 핀을 출력과 입력으로 선언
```

```
  for(int i=0; i<4; i++){
```

```
    pinMode(trig[i], OUTPUT);
```

```
    pinMode(echo[i], INPUT);
```

```
  }
```

```
}
```

//4개의 초음파센서에 대해 신호 내보내기

```
void connect_pin(){
```

```
    long duration;
```

```
    for(int i=0; i<4; i++){
```

```
        digitalWrite(trig[i], LOW); // Trig 핀 Low
```

```
        delayMicroseconds(2);      // 2us 딜레이
```

```
        digitalWrite(trig[i], HIGH); // Trig 핀 High
```

```
        delayMicroseconds(10);     // 10us 딜레이
```

```
        digitalWrite(trig[i], LOW); // Trig 핀 Low
```

```
        //거리 계산
```

```
        duration = pulseIn(echo[i], HIGH);
```

```
        distance[i] = duration * 170 / 1000; // 왕복시간이므로 340m를 2로 나누어 170 곱하기
```

```
    }
```

```
}
```

```
void loop() {
```

```
    //한번씩 회전 할 때마다 업데이트 된 거리 값 가져오기
```

```
    connect_pin();
```

```
    // 거리를 시리얼 모니터에 출력
```

```
    Serial.print(distance[0]);
```

```
    Serial.print("mm ");
```

```
    Serial.print(distance[1]);
```

```
    Serial.print("mm ");
```

```
    Serial.print(distance[2]);
```

```
    Serial.print("mm ");
```

```
    Serial.print(distance[3]); // 거리를 시리얼 모니터에 출력
```

```
Serial.println("mm ");
```

```
delay(100);
```

```
}
```