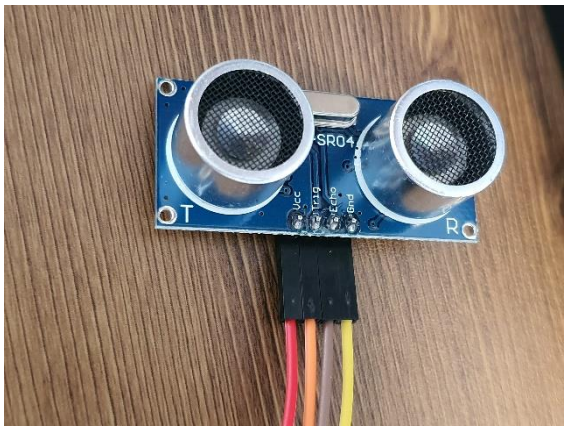
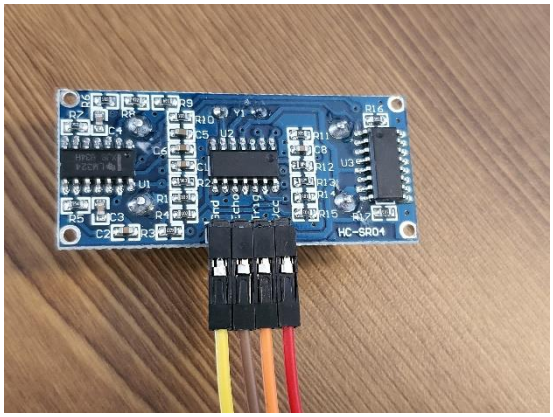
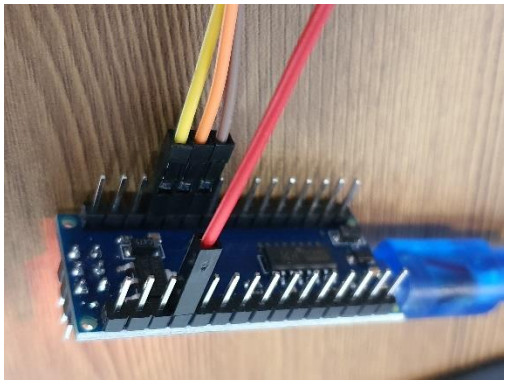
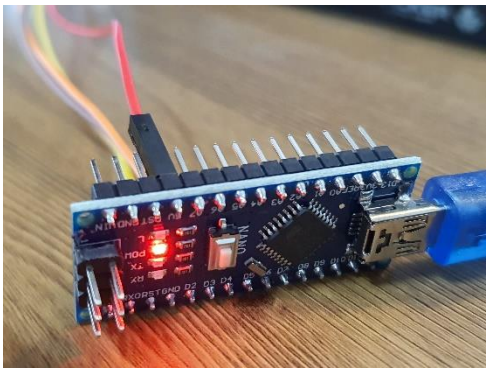
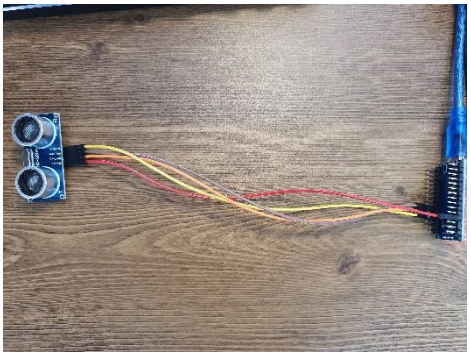
연결 이미지



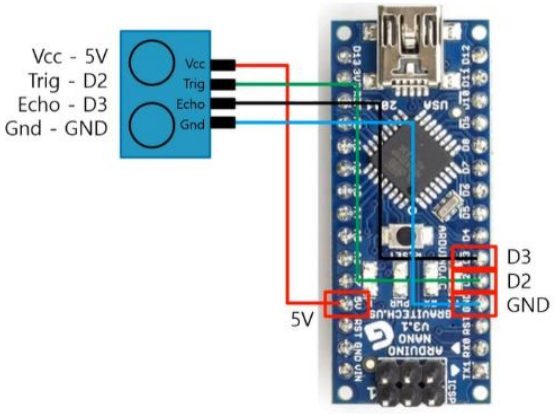


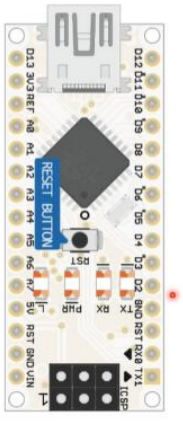




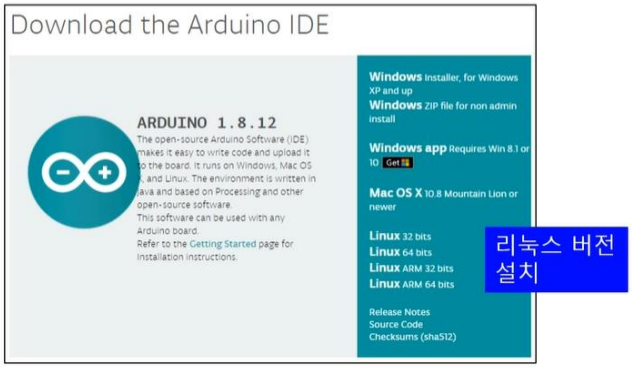


• 각 센서의 Vcc, Trig, Echo, Gnd 를 각 핀에 연결



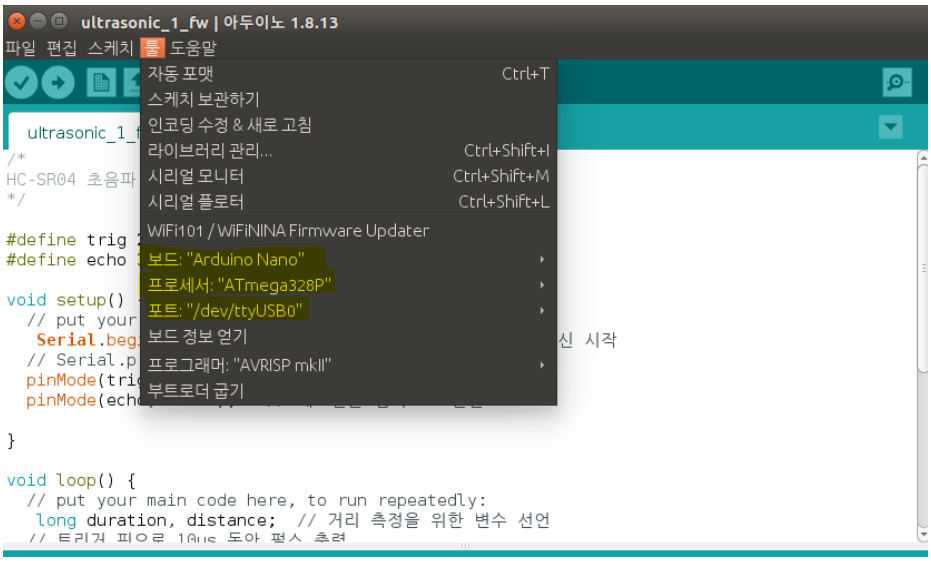


- 아두이노 펌웨어 프로그래밍 도구
  - 아두이노 코드 작성할 때와 제작된 펌웨어를 아두이노에 적어 넣을 때 사용
  - https://www.arduino.cc/en/main/software




```
$ cd ~/Downloads/Arduino-1.8.12
$ sudo ./install.sh
$ sudo arduino
```


```
soorim@soorim-virtual-machine:~$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 005: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
```



1. Python 코드 실행하기

1) cd ~/xycar_ws/src

2) catkin_create_pkg ultrasonic std_msgs rospy

3) mkdir launch (ultrasonic아래 만들기)

4) cm (빌드)

5) gedit ultrasonic_pub (src에서 생성)

6) gedit ultrasonic_sub

7) ultra.launch (launch에서 생성)

8) roslaunch ultrasonic ultra.launch

결과

```
setting /run_id to 38c81ff8-5617-11eb-a4ee-000c294531f4
process[rosout-1]: started with pid [2539]
started core service [/rosout]
process[ultrasonic_pub-2]: started with pid [2542]
process[ultrasonic_sub-3]: started with pid [2543]
11915
11914
11913
11916
11914
11915
11914
11915
11915
31
183
461
651
824
```

-에러가 발생할 경우

```
Traceback (most recent call last):
  File "/home/selfdriving/xycar_ws/src/ultrasonic/src/ultrasonic_pub.py", line 8
, in <module>
    baudrate=9600,
  File "/usr/lib/python2.7/dist-packages/serial/serialutil.py", line 180, in __i
nit__
    self.open()
  File "/usr/lib/python2.7/dist-packages/serial/serialposix.py", line 294, in op
en
    raise SerialException(msg.errno, "could not open port %s: %s" % (self._port,
 msg))
serial.serialutil.SerialException: [Errno 13] could not open port /dev/ttyUSB0:
[Errno 13] Permission denied: '/dev/ttyUSB0'
[ultrasonic_pub-2] process has died [pid 3657, exit code 1, cmd /home/selfdrivin
g/xycar_ws/src/ultrasonic/src/ultrasonic_pub.py __name:=ultrasonic_pub __log:=/h
ome/selfdriving/.ros/log/5934b3ac-55b1-11eb-bff6-000c29c2fee2/ultrasonic_pub-2.l
og].
log file: /home/selfdriving/.ros/log/5934b3ac-55b1-11eb-bff6-000c29c2fee2/ultras
onic_pub-2*.log
```

sudo usermod -aG dialout <사용자이름>

reboot

pub파일

```
#!/usr/bin/env python
import serial, time, rospy, re
from std_msgs.msg import Int32

ser_front = serial.Serial(
        port='/dev/ttyUSB0',
        baudrate=9600,
        )
def read_sensor():
        serial_data = ser_front.readline()
        ser_front.flushInput()
        ser_front.flushOutput()
        ultrasonic_data = int(filter(str.isdigit, serial_data))
        msg.data = ultrasonic_data

if __name__ == '__main__':
        rospy.init_node('ultrasonic_pub', anonymous=False) # initialize node
        pub = rospy.Publisher('ultrasonic', Int32, queue_size=1)
        msg = Int32() # message type
        while not rospy.is_shutdown():
                read_sensor()
                pub.publish(msg) # publish a message
                time.sleep(0.2)
        ser_front.close()
```

sub파일

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


import rospy

from std_msgs.msg import Int32


def callback(msg):

    print(msg.data)


rospy.init_node('ultrasonic_sub')

sub = rospy.Subscriber('ultrasonic', Int32, callback)


rospy.spin()
```

launch 파일

```
<launch>

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

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

</launch>
```

Arduino 파일

```
/*
HC-SR04 초음파 센서
*/
#define trig 2   // 트리거 핀 선언
#define echo 3   // 에코 핀 선언

void setup()
{
   Serial.begin(9600);        // 통신속도 9600bps로 시리얼 통신 시작
   // Serial.println("Start... Ultrasonic Sensor");
   pinMode(trig, OUTPUT);   // 트리거 핀을 출력으로 선언
   pinMode(echo, INPUT);    // 에코핀을 입력으로 선언
}
void loop() {
   long duration, distance;   // 거리 측정을 위한 변수 선언
   // 트리거 핀으로 10us 동안 펄스 출력
   digitalWrite(trig, LOW);   // Trig 핀 Low
   delayMicroseconds(2);        // 2us 딜레이
   digitalWrite(trig, HIGH); // Trig 핀 High
   delayMicroseconds(10);      // 10us 딜레이
   digitalWrite(trig, LOW);   // Trig 핀 Low

   // pulseIn() 함수는 핀에서 펄스신호를 읽어서 마이크로초 단위로 반환
   duration = pulseIn(echo, HIGH);
   distance = duration * 170 / 1000; // 왕복시간이므로 340m를 2로 나누어 170 곱하
```

```
    Serial.print("Distance(mm): ");
    Serial.println(distance); // 거리를 시리얼 모니터에 출력
    delay(100);
}
```