



# UglyBot



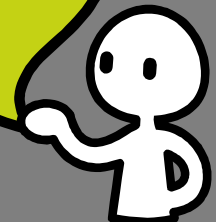
## “라인따라가기”



# 01

## 연결 준비하기

준비하기 상세 설명은  
"파이썬 설정" 문서를  
참고하세요.



# 1. 어글리봇 살펴보기



USB 동글

버튼

적외선  
센서

오른쪽  
모터

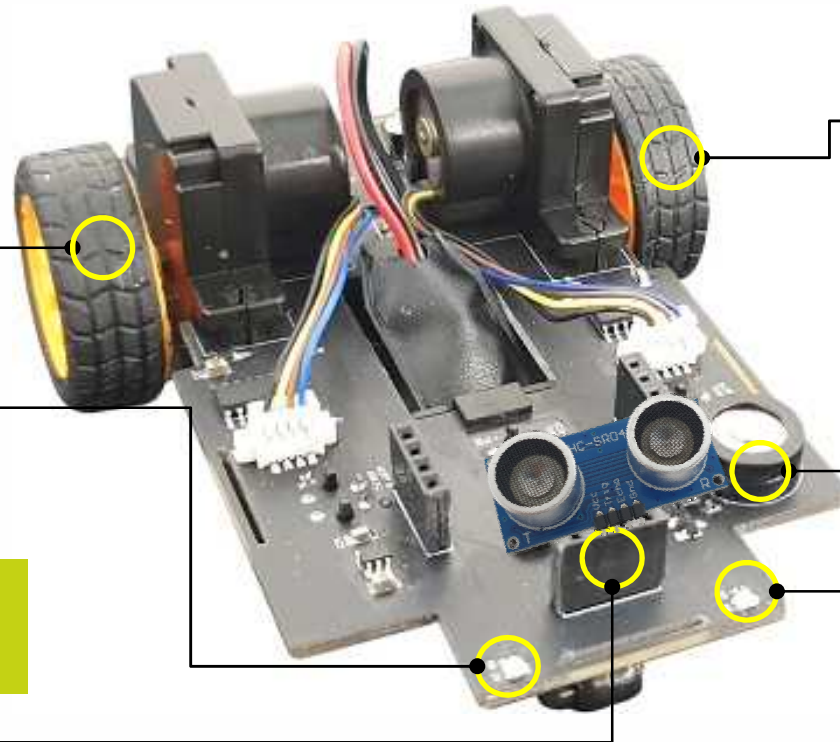
왼쪽  
모터

오른쪽  
LED

부저

초음파

왼쪽  
LED



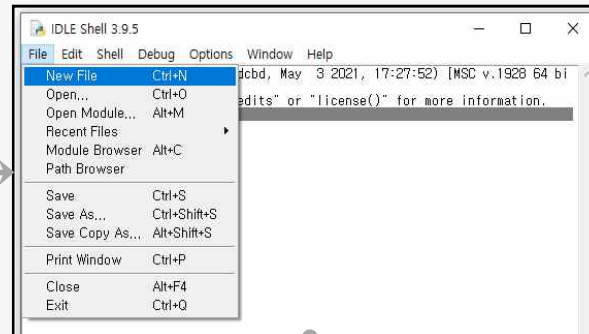
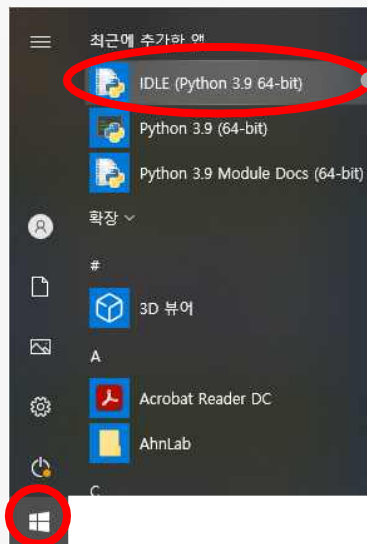
## 2. 연결 하기



1. USB 동글을 컴퓨터에 연결한다.

2. 어글리봇 단말기 전원을 켜다.

3. 파이썬(IDLE shell) 실행한다.



### 3. 코딩 준비하기-직접 입력하기



어글리봇 파이썬 코딩을 작성 할 경우  
라이브러리 import, 객체 정의를 상단부에 삽입하도록 한다.

```
*untitled+
File Edit Format Run Options Window Help
from time import sleep          # sleep 함수로드
from pyjuni.uglybot import *     # 주니랩 어글리봇 라이브러리 로드
from pyjuni.jkeyevent import *  # 주니랩 키보드 라이브러리 로드

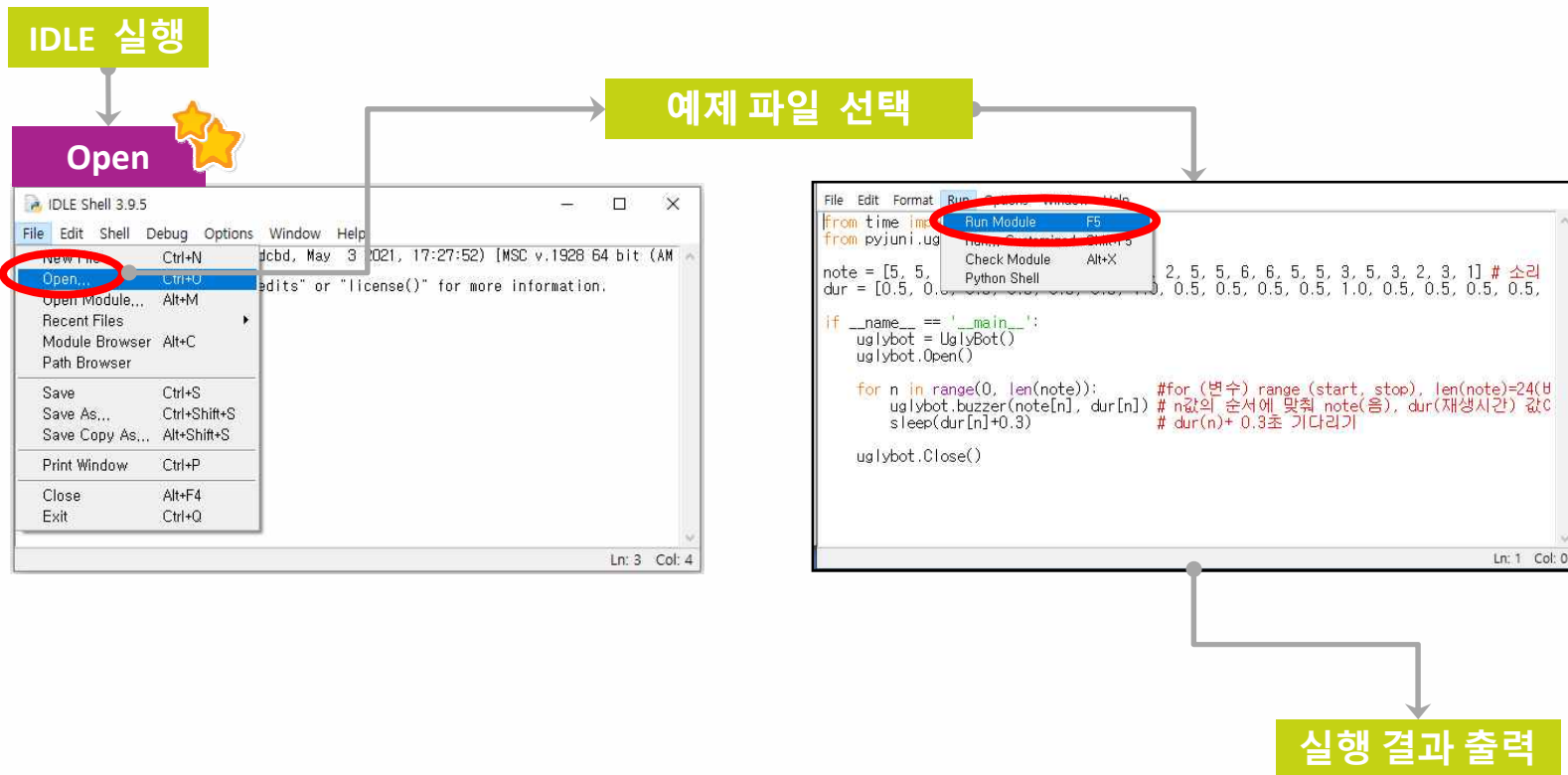
if __name__ == '__main__':
    uglybot = UglyBot()          # uglybot 객체 정의
    uglybot = UglyBot(receiveData) # uglybot callback 데이터 받기
    jkey = JKeyEvent()          # keyboard 이벤트 객체 정의
    uglybot.Open()              # uglybot 객체 열기

Ln: 9 Col: 35
```

## 4. 코딩 준비하기-예제 실행하기



본 교재 파이썬 코딩 실습은 예제 파일이 제공된다.  
하기 방법으로 예제를 선택하여 실행하면 결과를 확인 할 수 있다.





# 02

## 라인따라가기 관련 함수

# 1. motor 제어 함수



## 모터 회전 값 제어

```
Uglybot.motor(left, right)
```

모터 제어 값 범위 : -100~100

0 : 모터 정지

1~100 : 모터 정방향으로 제어(직진)

-100~-1 : 모터 역방향으로 제어(후진)



## 2. callback 값



Callback(data) : UglyBot에서 패킷을 받으면 호출되는 값

Data[7] // 버튼 상태 값, 1-버튼 눌림, 0- 버튼 안눌림

Data[8] // 왼쪽 IR 센서 값 (범위: 1~100)

Data[9] // 중간 IR 센서 값 (범위: 1~100)

Data[10] // 오른쪽 IR 센서 값 (범위: 1~100)

Data[11] // 초음파 센서 값 (단위: cm)

## 2. ir 제어 함수



### 적외선 근접 센서(ir) 제어

```
Ug1ybot.ir(status) // 0 bit ( 왼쪽 IR on/off)
                    // 1 bit ( 중앙 IR on/off)
                    // 2 bit ( 오른쪽 IR on/off)
```

```
Status=0           // IR 센서 모두 끄기
Status=1           // 왼쪽 IR 센서 켜기
Status=2           // 중앙 IR 센서 켜기
Status=3           // 오른쪽 IR 센서 켜기
...
Status=7           // IR 센서 모두 켜기
```

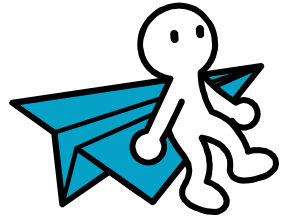


# 02

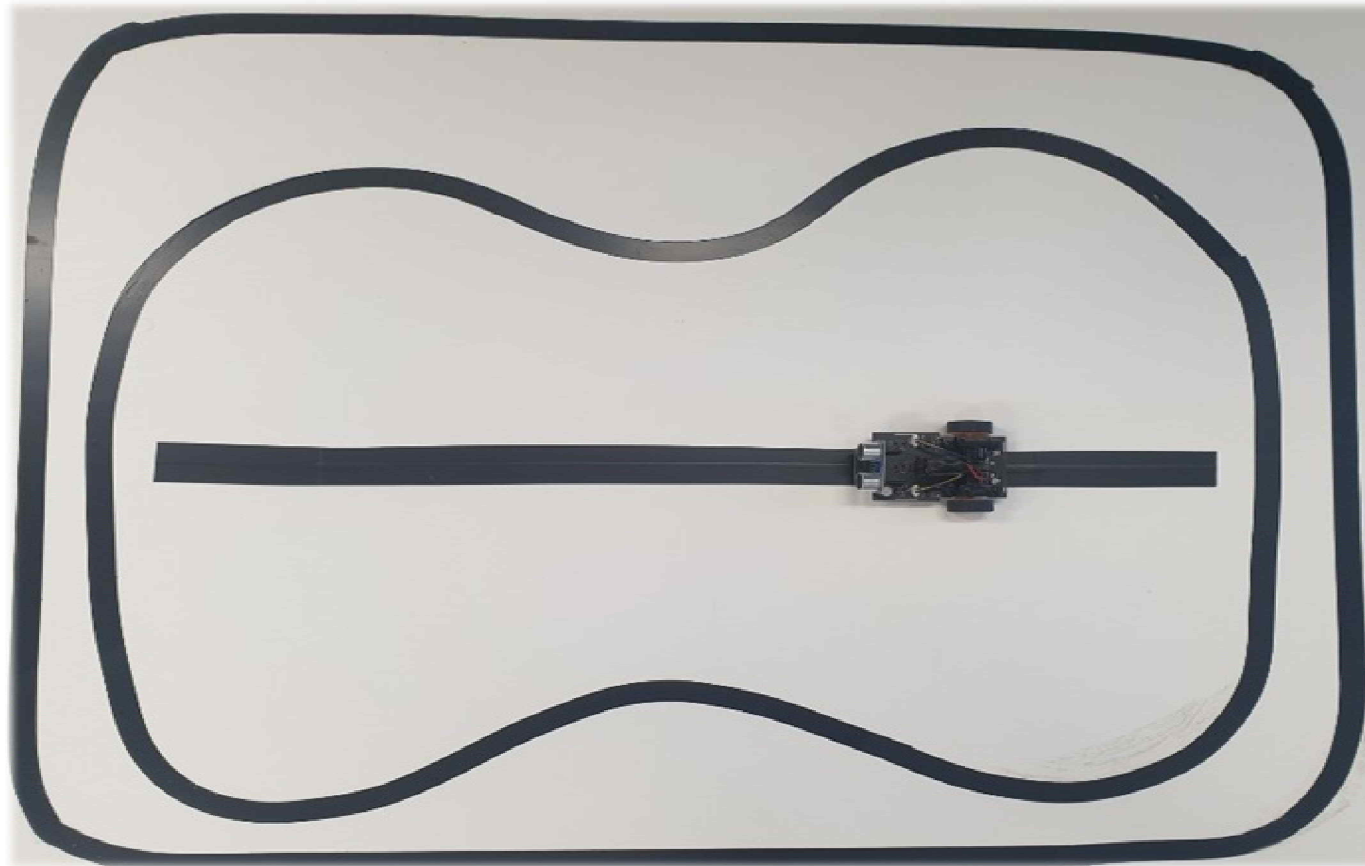
## 적외선으로 선찾기

Contents Subtitle

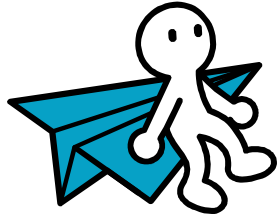
# 라인 따라가기?



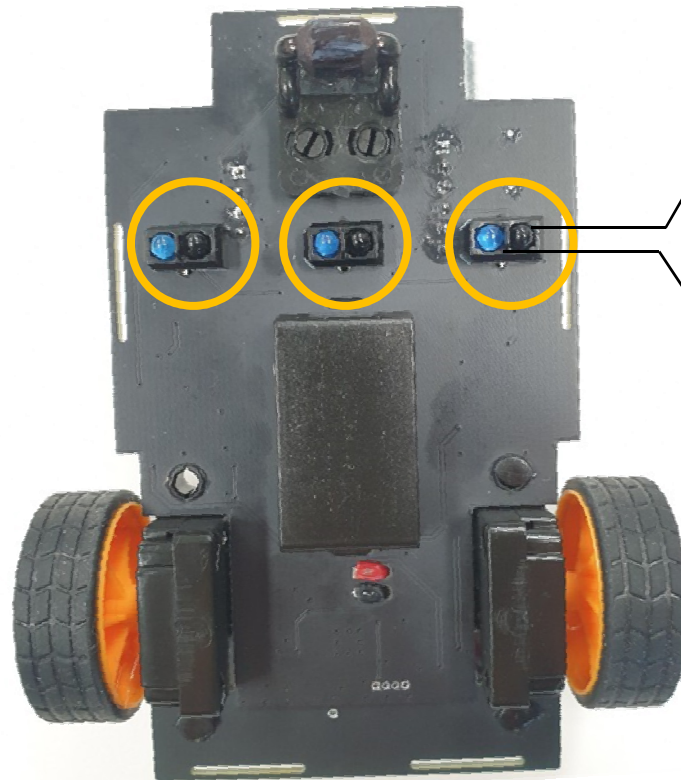
적외선을 이용하여 선을 따라서 움직입니다.



# 적외선(IR)이 선을 찾아줍니다.



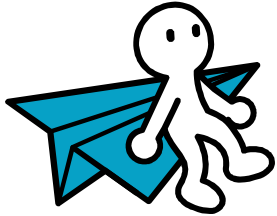
빛의 성질을 이용해 적외선(IR, Infrared Sensor) 송수신으로 흰색과 검정색 등의 선을 구분하여 길(선)을 찾게 도와준다.



검정 : 적외선(IR) 수신

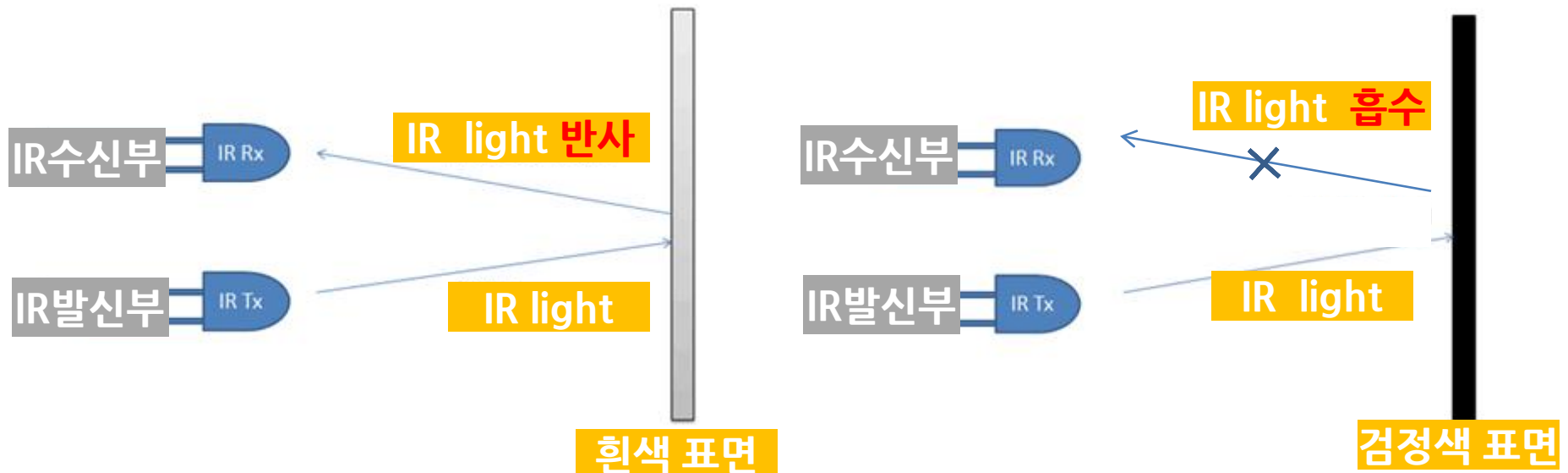
파랑 : 적외선(IR) 송신

# 적외선이 어떻게 선을 찾을까요?

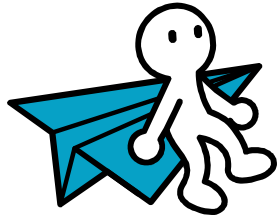


검정색은 빛을 흡수, 그 외 색상은 빛을 반사하여 눈을 통해 고유의 색으로 구분하게 된다.

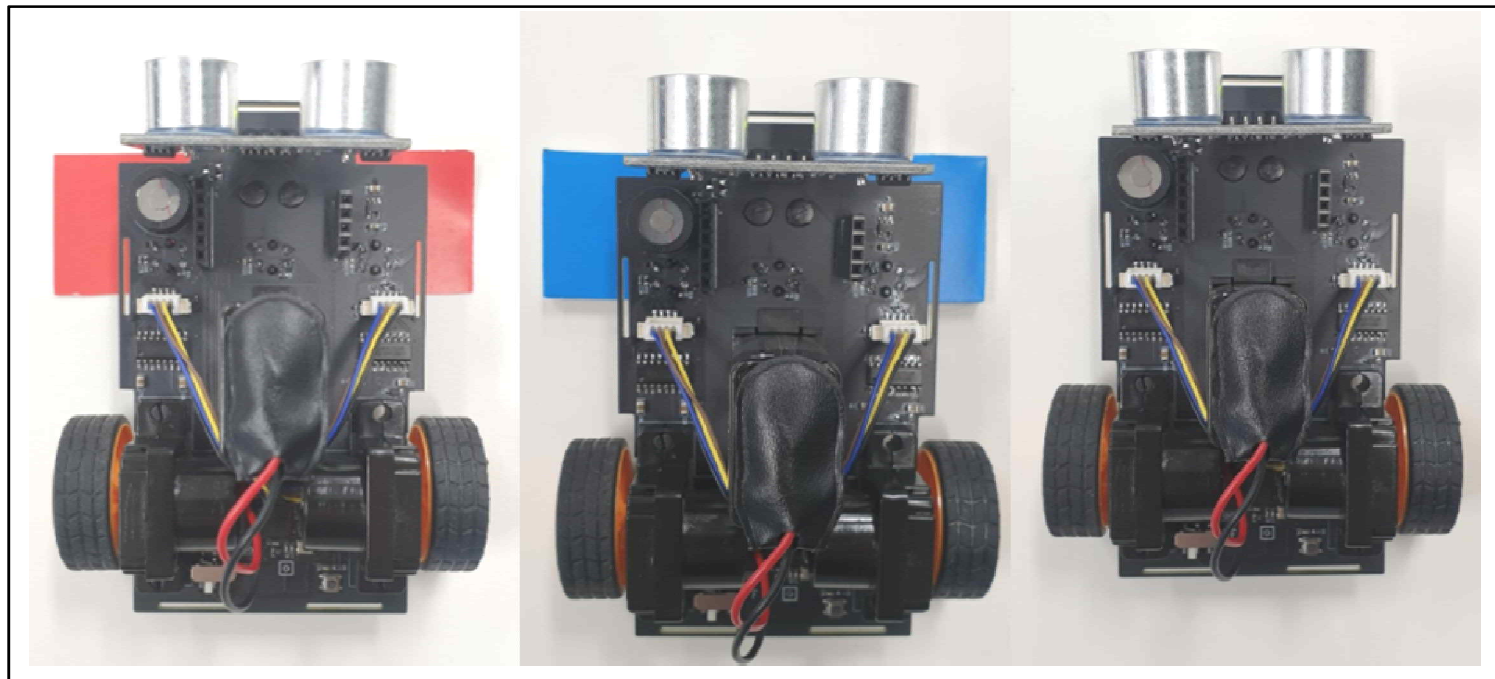
이런 빛의 특성을 이용하여 적외선의 송수신으로 빛의 반사/흡수를 통해 라인을 찾아 주행하게 된다.



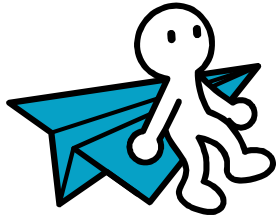
# 검정색만을 따라 갈 수 있을까요?



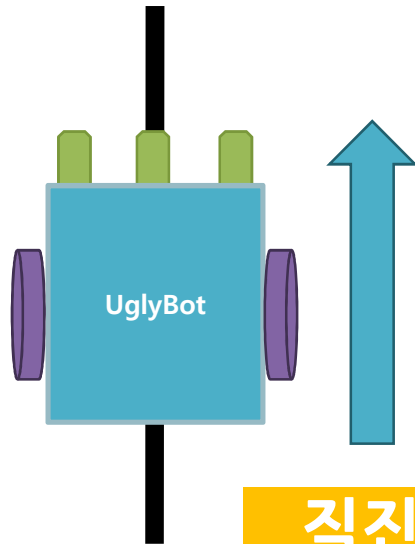
적외선 디지털 신호 값을 색상별로 확인 하고 선을 찾는 원리를 이해한다. 검정색 외의 반사값이 낮은 색상을 찾아 또다른 길(선)을 만들 수 있다.



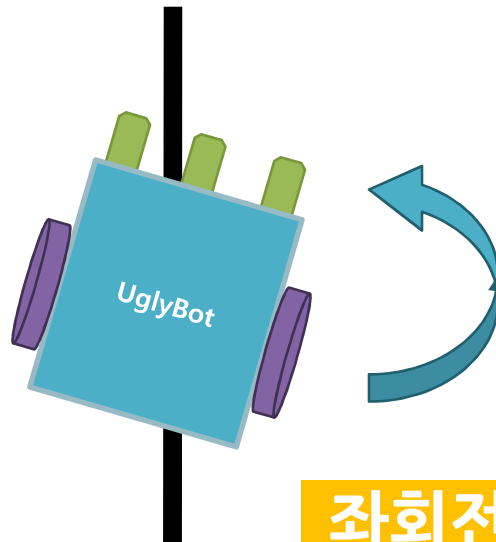
# 선을 넘지않는 어글리봇



3개의 적외선 센서 모듈이 탑재되어 중앙의 센서를 기준으로 선(검정색)을 트래킹하게 되며, 선을 이탈 할 경우에 좌/우 센서의 적외선 반사 값에 의해 선을 찾도록 자세 보정하여 트래킹한다.



직진



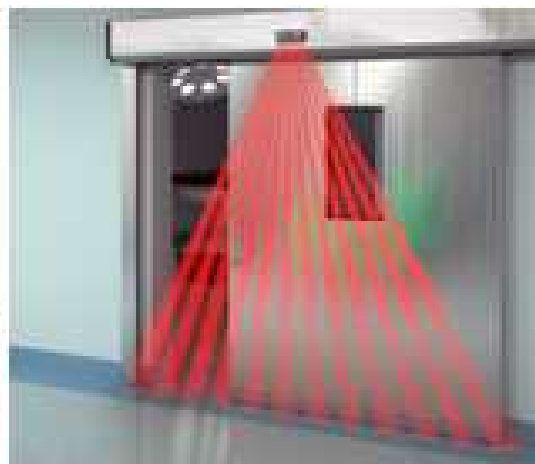
좌회전



우회전



# 적외선 센서 응용 제품 찾아보기





03

코딩 실습하기

# 1단계 : 변수 설정



## 적외선 센서 변수 설정과 데이터 정의

```
leftir = 0
midir = 0
rightir = 0
preState = 0
```

왼쪽 IR센서

중간 IR센서

오른쪽 IR센서

이전상태

변수 정의

```
def receiveData(packet):
```

```
    global leftir
```

```
    global midir
```

```
    global rightir
```

```
    leftir = packet[8];
```

```
    midir = packet[9];
```

```
    rightir = packet[10];
```

글로벌 변수 선언

변수에 들어갈 패킷 번호 정의



## 2단계 : 센서 켜기&모터변수 값 정의



적외선 센서를 켜고, 기본 모터 세기 값 정의



```
if __name__ == '__main__':
```

```
    uglybot = UglyBot(receiveData)
```

```
    jkey = JKeyEvent()
```

```
    uglybot.Open()
```

```
    uglybot.ir(7)
```

} 적외선 센서 모두 켜기

```
    while not jkey.isKeyEscPressed(): ESC키를 누를 때까지 무한 반복
```

```
        leftMotor = 30
```

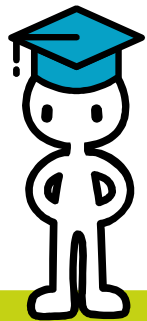
```
        rightMotor = 30
```

} 직진 할 수 있도록  
왼쪽/오른쪽 모터 값 30으로 정의

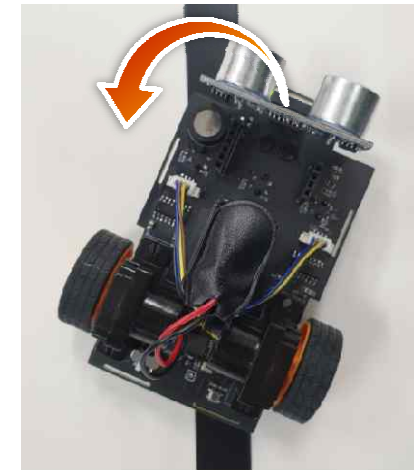
### 3단계 : 왼쪽센서가 라인 위에 있을 때



왼쪽센서가 선 위에 있을 때 왼쪽모터를 감속하여 오른쪽으로 회전하도록 한다.



왼쪽센서	0
중간센서	100
오른쪽센서	100



if leftir < 50:

leftMotor -= 15

preState = 1

왼쪽센서 값이 50이하 일 경우

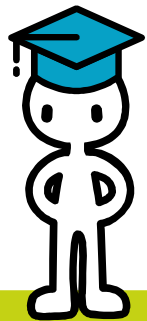
왼쪽 모터 회전 감속으로 우회전

오른쪽 치우칠 때  
이전상태 변수값 "1"로 정의

## 4단계 : 오른쪽 센서가 라인위에 있을 때



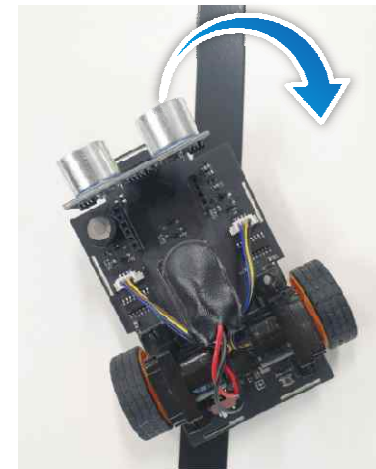
오른쪽센서가 선 위에 있을 때 오른쪽모터를 감속하여 왼쪽으로 회전하도록 한다.



왼쪽센서 100

중간센서 100

오른쪽센서 0



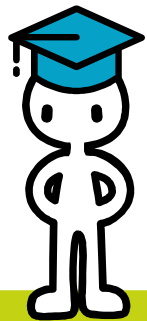
```
elif rightir < 50:  
    rightMotor -= 15  
    preState = 2
```

- > 오른쪽 IR 센서가 50이하 일 경우
- > 오른쪽 모터 회전 감속으로 좌회전
- > 왼쪽 치우칠 때  
이전상태 변수값 "2"로 정의

## 5단계 : 중앙 센서가 라인위에 있을 때



중앙센서가 선 위에 있을 때 양쪽모터 감속없이 직진한다.



왼쪽센서 100

중간센서 100

오른쪽센서 100



`elif midir < 50:`

`preState = 0`



오른쪽 IR 센서가 50이하 일 경우

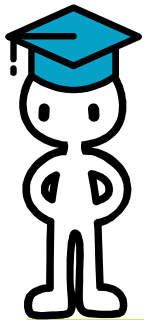


중앙 센서 위에 있을 때(치우침 없음)  
이전상태 변수값 "0"로 정의

## 6단계 : 선을 이탈 했을 때



왼쪽, 중간, 오른쪽 센서 모두 선을 이탈했을 경우에  
어글리봇 치우침 방향에 따라 모터 값을 정한다.



"이전상태" 변수는  
2단계, 3단계 참고

왼쪽센서

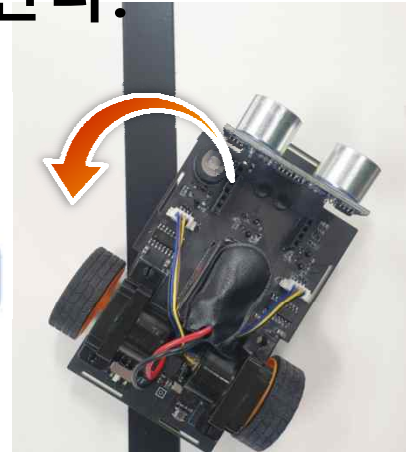
100

중간센서

100

오른쪽센서

100



if leftir>60 and midir>60 and rightir>60:

if preState == 1:

leftMotor = 0

if preState == 2:

rightMotor = 0

오른쪽으로 치우칠 때  
좌회전

왼쪽으로 치우칠 때  
우회전



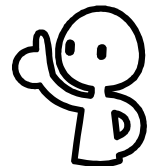
## 7단계 : 바퀴 회전하기



1단계~4단계에서 조건에 따라 설정한 왼쪽/오른쪽모터 변수 값으로 모터 회전 세기로 Input하여 바퀴를 움직여 어글리봇이 작동한다.

```
uglybot.motor(leftMotor, rightMotor)  
sleep(0.05)
```

앞에서 작성한 코드에서  
왼쪽모터 / 오른쪽 모터  
변수 값을 단계별 찾아보고  
코드의 흐름을 읽어보자.



# 1~7단계 : 모아보기



```
from time import sleep
from pyjuni.uglybot import *
from pyjuni.jkeyevent import *
```

```
leftir = 0
midir = 0
rightir = 0
preState = 0
```

```
def receiveData(packet):
    global leftir
    global midir
    global rightir
    leftir = packet[8];
    midir = packet[9];
    rightir = packet[10];
```

```
if __name__ == '__main__':
    uglybot = UglyBot(receiveData)
    jkey = JKeyEvent()
    uglybot.Open()
```

```
    uglybot.ir(7)
    while not jkey.isKeyEscPressed():
        leftMotor = 20
        rightMotor = 20
        if leftir < 50:
            leftMotor -= 15
            preState = 1
        elif rightir < 50:
            rightMotor -= 15
            preState = 2
        elif midir < 50:
            preState = 0
        if leftir > 60 and midir > 60 and rightir > 60:
            if preState == 1:
                leftMotor = 0
            if preState == 2:
                rightMotor = 0
        uglybot.motor(leftMotor, rightMotor)
        sleep(0.05)
```

```
    uglybot.Close()
```

## 1단계

센서 선언, 변수 설정, 변수 정의

## 2단계

센서 켜기 선언  
왼쪽모터, 오른쪽 모터 변수 값 정의

## 3단계

어글리봇이 오른쪽으로 치우칠 때

## 4단계

어글리봇이 왼쪽으로 치우칠 때

## 5단계

어글리봇이 치우침 없이 갈 때

## 6단계

3개센서 선을 모두 이탈 했을 때

## 7단계

왼쪽/오른쪽 모터 회전 하기



**Thank You**