



IR 센서

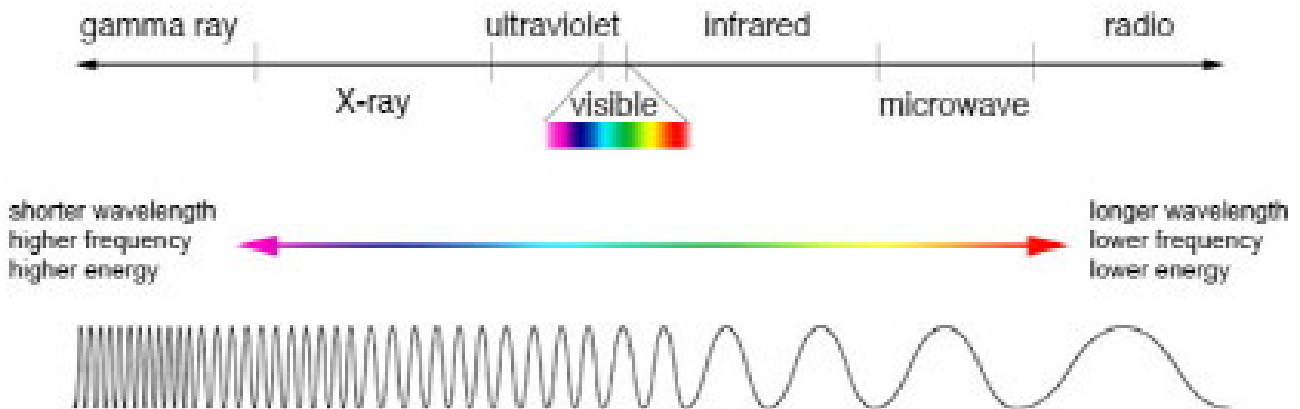
적외선(줄여서 IR)은 사람의 눈으로 볼 수 없는 전자기 파장의 일종입니다.

그것은 에너지의 한 형태이기도 합니다. 이중에 우리가 볼 수 있는 작은 부분이 있는데, 이것을 가시광선이라고 합니다.

우리가 보는 다른 색깔들은 다양한 파장에 달려있습니다. 파장이 길어질수록 색은 인간에게 잘 보이지 않게 됩니다.

하지만, 특수 센서는 이러한 파장을 감지할 수 있습니다.

우리는 주미의 IR 센서를 사용하여 센서에 감지된 물체를 감지할것입니다.



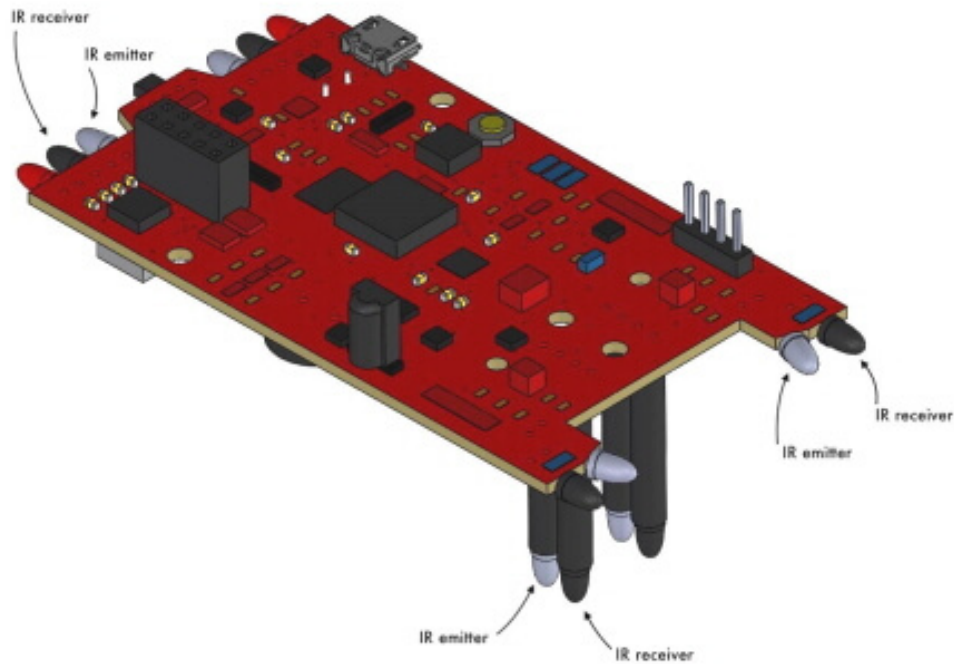
<https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html>

Zumi IR 센서

Zumi에는 전면에 2개, 후면에 2개, 하단에 2개 등 6개의 IR 센서가 장착되어 있습니다. 모두 인덱스가 있으므로 데이터를 더 쉽게 읽을 수 있습니다.

- IR 0 = 전면 우측
- IR 1 = 오른쪽 하단
- IR 2 = 오른쪽 뒤
- IR 3 = 왼쪽 하단
- IR 4 = 왼쪽 뒤

- IR 5 = 전면 왼쪽



화면에 IR 데이터 표시

Zumi의 IR 데이터를 표시할 것입니다. IR 데이터를 얻으려면 Zumi 객체가 필요합니다.

라이브러리 가져오기

In []:

```
from zumi.zumi import Zumi
import time

zumi = Zumi()
```

센서 값은 무엇을 의미할까요?

IR 센서는 적외선 펄스를 방출하고 물체에 반사된 후 되돌아오는 적외선을 측정하여 작동합니다.

이 숫자는 0에서 255 사이입니다. 숫자가 낮을수록 근처에 있는 물체가 IR 광선을 센서로 다시 반사하고 있음을 나타냅니다.

각 셀을 실행하고 센서를 가릴 때 화면의 숫자가 변경되는 것을 지켜보세요.

전면 IR 센서

In []:

```
for i in range(0,20):
    ir_readings = zumi.get_all_IR_data()
    front_right_ir = ir_readings[0]
    front_left_ir = ir_readings[5]

    message = "    IR readings    "
    message = message + str(front_right_ir) + ", " + str(front_left_ir)

    print(message)
    time.sleep(0.1)
print("완료")
```

후면 IR 센서

In []:

```
for i in range(0,20):
    ir_readings = zumi.get_all_IR_data()
    back_right_ir = ir_readings[2]
    back_left_ir = ir_readings[4]

    message = "    IR readings    "
    message = message + str(back_right_ir) + ", " + str(back_left_ir)

    print(message)
    time.sleep(0.1)
print("완료")
```

하단 IR 센서

하단 IR 센서는 검정과 흰색의 라인 추적 또는 감지에 적합합니다.

In []:

```
for i in range(0,20):
    ir_readings = zumi.get_all_IR_data()
    bottom_right_ir = ir_readings[1]
    bottom_left_ir = ir_readings[3]

    message = "    IR readings    "
    message = message + str(bottom_right_ir) + ", " + str(bottom_left_ir)

    print(message)
    time.sleep(0.1)
print("완료")
```

리스트

IR 데이터에서 아직 모르는 내용을 발견할 수 있습니다.
위의 예에서는 `get_all_IR_data()` 라는 함수를 사용합니다.

함수에서는 한번에 6개의 센서의 값을 모두 알려줍니다. 6개의 센서 값이 어떻게 모두 같은 변수에 있을 수 있을까요?

이 모든 데이터는 실제로 **리스트**에 저장됩니다.

파이썬에서 목록은 여러 변수를 생성하지 않고도 동일한 유형의 데이터를 저장하고 구성하는 데 사용할 수 있습니다.

값 접근

`zumi.get_all_IR_data()`를 호출하면 프로그램은 모든 센서 데이터를 요청하고 6개 항목으로 리스트에 저장합니다.

이 예에서 리스트 이름은 `ir_readings`입니다.

일반적으로 세로로 표시되는 리스트와 달리 Python의 리스트는 가로로 작성되고 [대괄호]로 묶입니다.

다음 셀에서 값을 읽고 출력해봅시다.

In []:

```
ir_readings = zumi.get_all_IR_data()
print(ir_readings)
```

리스트 이름을 출력하면 모든 값이 출력되지만 하나 또는 두 개의 값만 확인하려는 경우에는 어떻게 해야 할까요?

인덱스로 데이터 또는 각 **요소**에 접근할 수 있습니다.

인덱스는 목록에 있는 집 주소와 같습니다. for 루프와 마찬가지로 파이썬 목록은 0에서 시작합니다.

이 리스트를 예로 들어 보겠습니다. 첫 번째 요소의 값은 무엇입니까? 마지막은 어떻습니까?

```
ir_readings = [195, 12, 171, 13, 142, 240]
```

index	0	1	2	3	4	5
value	195	12	171	13	142	240

다음은 인덱스 0의 첫 번째 요소에 접근하는 예입니다.

모든 값을 출력하는 연습을 합니다.

IR 값을 알아보기 쉬운 이름으로 변수에 저장했는지 확인하세요!

프로그램이 점점 더 복잡해지기 시작하면 어떤 변수가 어떤 센서를 나타내는지 잊어버릴 수도 있습니다.

In []:

```
ir_readings = zumi.get_all_IR_data()
front_right_ir = ir_readings[0]
print(front_right_ir)
```

#여기에 나머지 센서들도 출력해보세요.

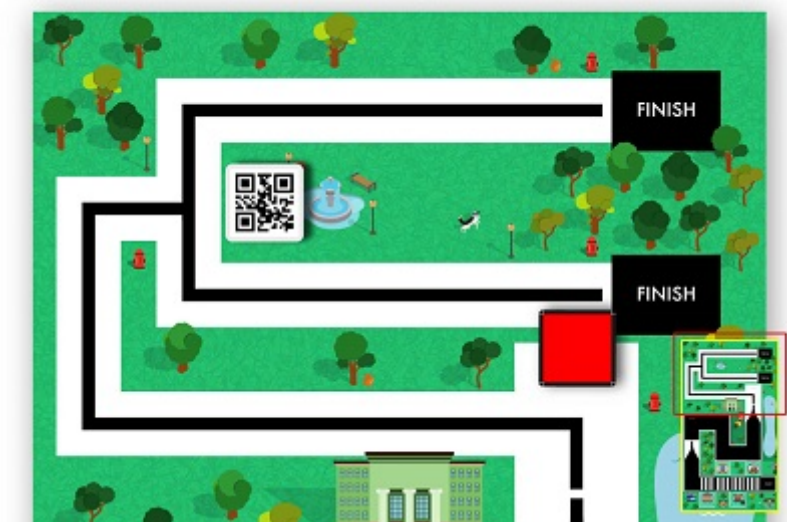
해결 방안

[해결 방안을 보려면 클릭하세요!](#)

대회 : 어떻게 사용되나요?



맵에는 벽으로 막혀있는 구간이 있습니다. 주미는 벽을 피해서 통과할 필요가 있습니다. 주미의 IR 센서를 사용하여 벽을 회피하세요. 자세한 방법은 장애물 회피 챕터를 확인하세요



대회에서 일부 구간에는 선 모양의 경로가 있는 것을 보았을 것입니다.
이 구간의 경우 Zumi의 IR 센서를 사용하여 Zumi를 안내할 수 있습니다. 그 방법이 궁금하다면 IR
센서로 선을 따라가도록 하세요!
자세한 방법은 라인따라가기 챕터를 확인하세요.