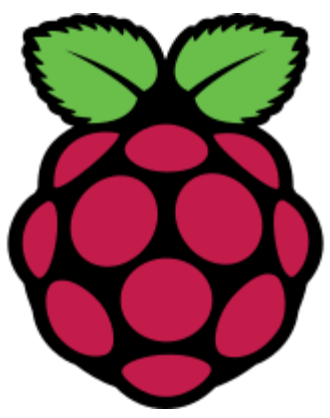


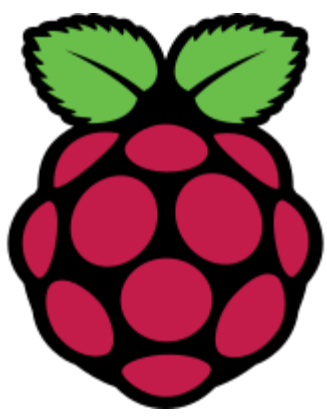
미니프로젝트

조해찬



목차

1. 프로젝트1 개요
2. 기능 구현 및 코드 간단 소개
3. 만들면서 느낀 생각
4. 프로젝트2 개요
5. 기능 구현 및 코드 간단 소개
6. 만들면서 느낀 생각

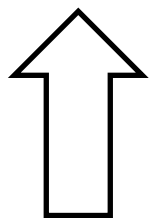
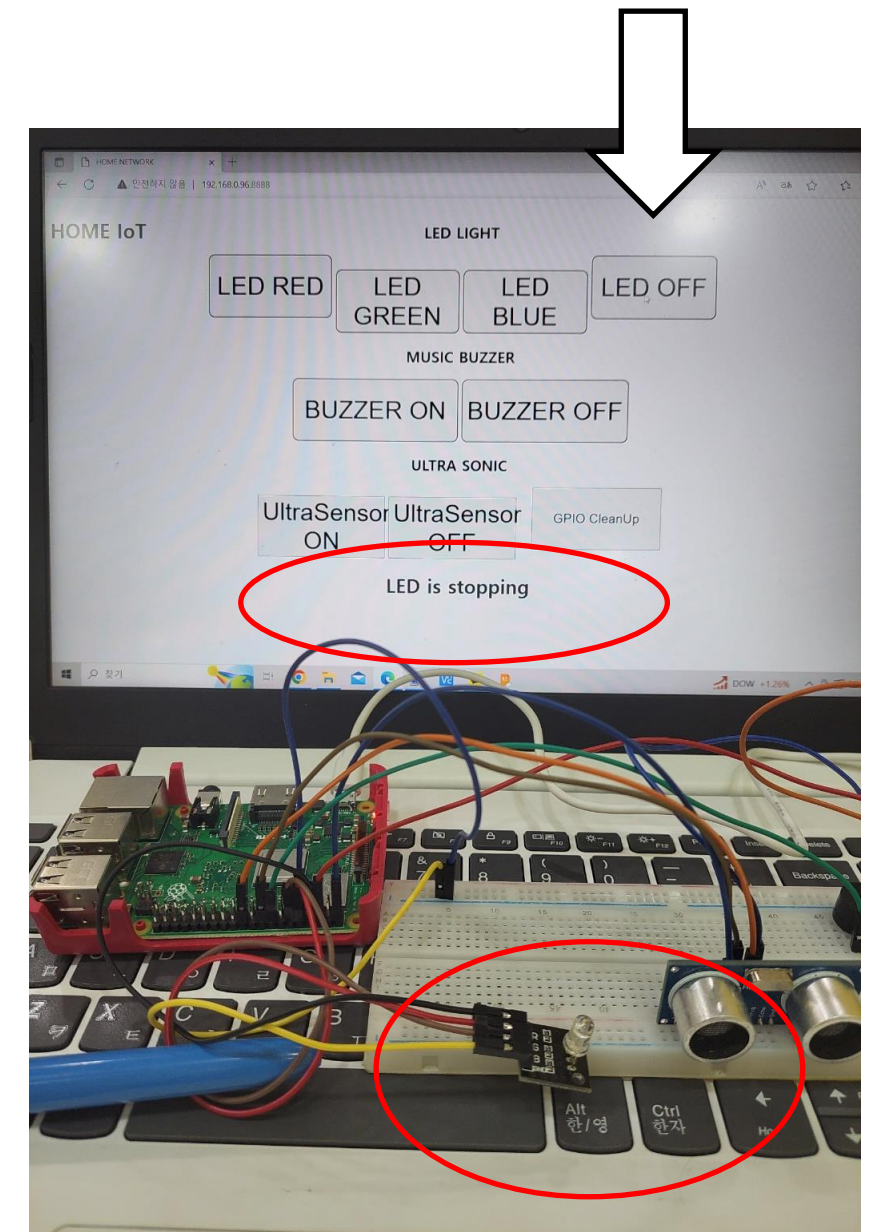
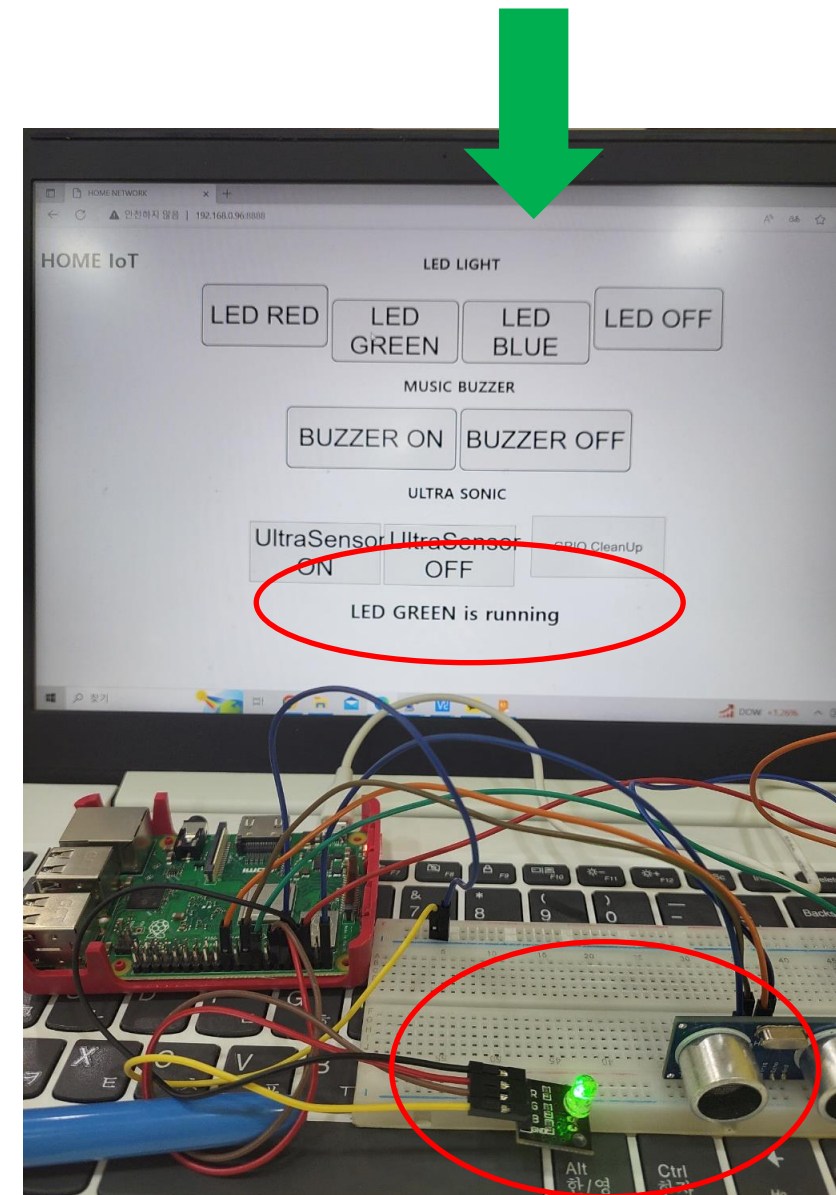
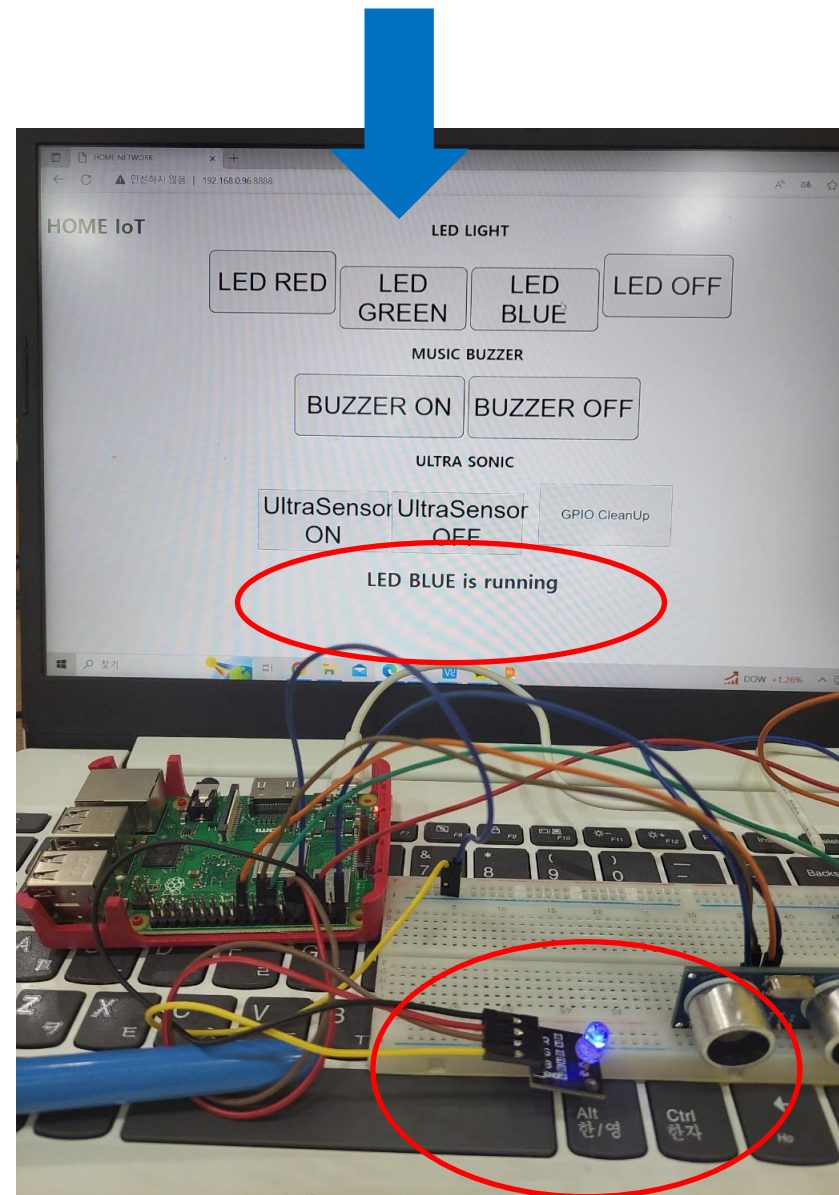
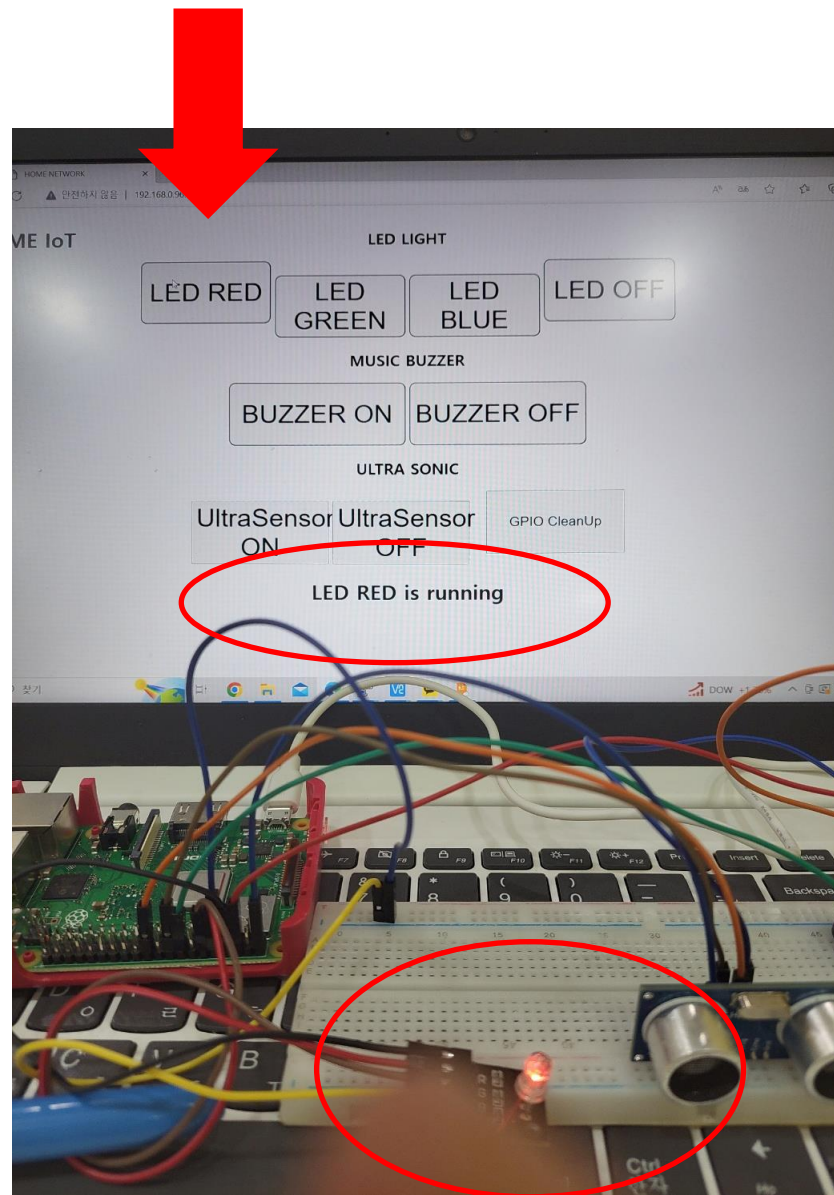


프로젝트1 개요



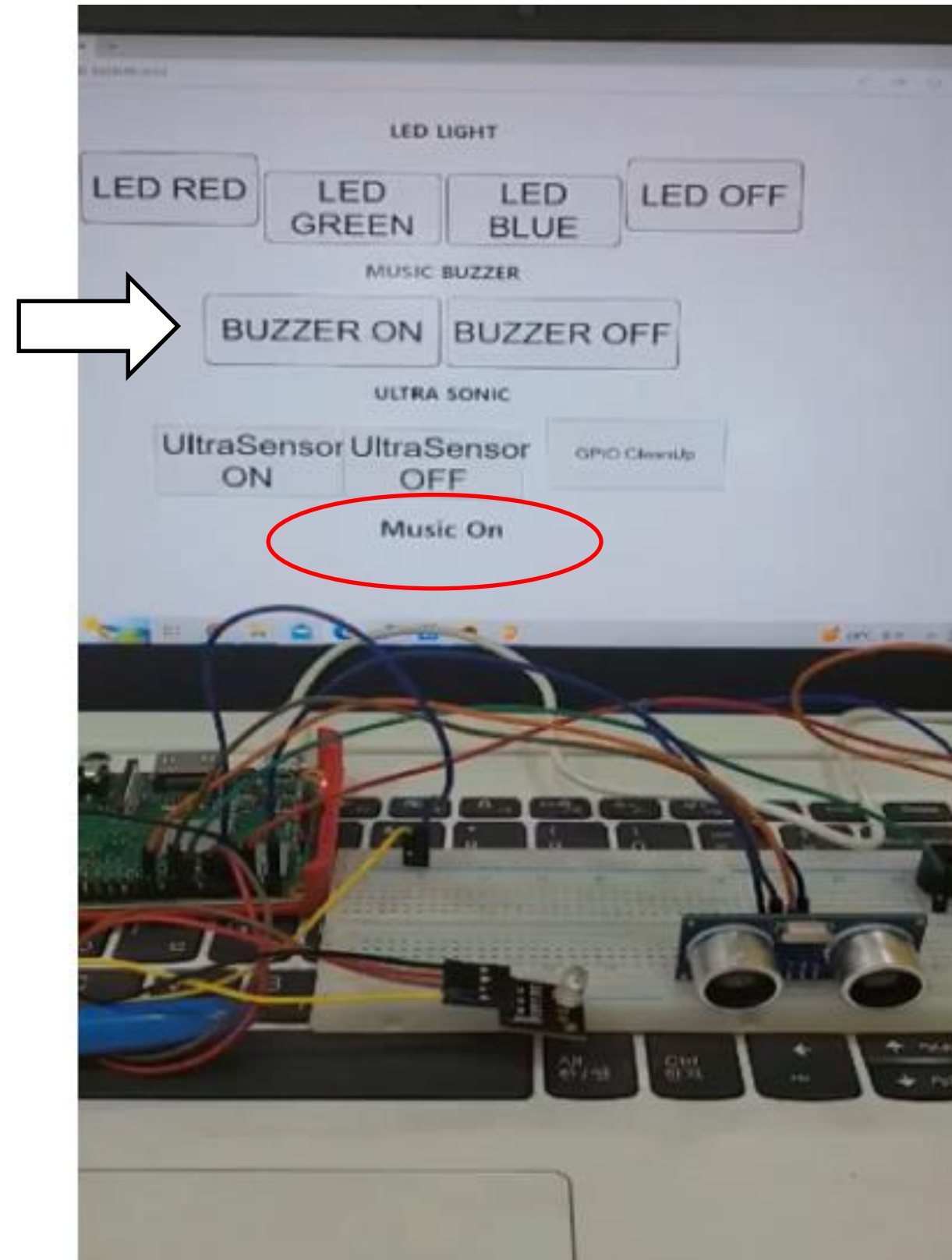
- 개발인원: 1명
- 개발 기간: 2023.06.14 ~ 2023.06.16
- 주요 기능:
 - 3색 LED 켜기 및 끄기
 - 부저 음악 켜기 및 끄기
 - 울트라소닉으로 사물과의 거리 측정(미구현)
- 개발 언어: Python
HTML
CSS
JavaScript
- 간단 소개: 웹에서 버튼을 클릭하여 원격 제어가 가능

기능 구현: 3색 LED RED 켜기



기능 구현: 부저 음악 켜기 및 끄기

영상



영상은 여기서 확인 가능합니다.

<https://github.com/Haecms/My-Portfolio/tree/78e0bf0f99886ad814ae18521841179a66bdee66/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC%ED%8C%8C%EC%9D%B4/Video>

기능 미구현: ULTRA SONIC

```
trigPin = 24
echoPin = 25
GPIO.setup(trigPin, GPIO.OUT)
GPIO.setup(echoPin, GPIO.IN)
GPIO.setup(trigPin, False)

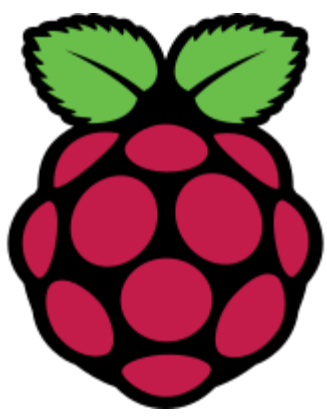
@app.route("/ultra/on")
def ultra_on():
    global stop
    stop = True
    try:
        while stop:
            GPIO.output(trigPin, True)
            time.sleep(0.00001)
            GPIO.output(trigPin, False)

            while GPIO.input(echoPin)==0:
                #print("0")
                start = time.time()
            while GPIO.input(echoPin)==1:
                #print("1")
                end = time.time()

            check_time = end-start
            distance = check_time * 34300/2
            print("Distance: %.1f cm" %distance)
            time.sleep(0.4)
            return "ok"
    except Exception as identifier:
        print("except")
        return "fail"
```

```
pi@raspberrypi: ~/Work/Python/Flask
dy in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.
GPIO.setup(trigPin, GPIO.OUT)
* Serving Flask app "mProject" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8888/ (Press CTRL+C to quit)
192.168.0.112 - - [16/Jun/2023 09:35:06] "GET / HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:35:06] "GET /favicon.ico HTTP/1.1" 404 -
192.168.0.112 - - [16/Jun/2023 09:35:52] "GET /led/r HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:36:09] "GET /led/g HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:36:18] "GET /led/b HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:36:26] "GET /led/off HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:37:07] "GET /buzzer/off HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:37:07] "GET /buzzer/on HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:37:36] "GET /buzzer/off HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:37:38] "GET /buzzer/on HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:38:14] "GET /led/r HTTP/1.1" 200 -
192.168.0.112 - - [16/Jun/2023 09:38:24] "GET /gpio/cleanup HTTP/1.1" 200 -
Distance: 8.3 cm
192.168.0.112 - - [16/Jun/2023 10:43:00] "GET /ultra/on HTTP/1.1" 200 -
```

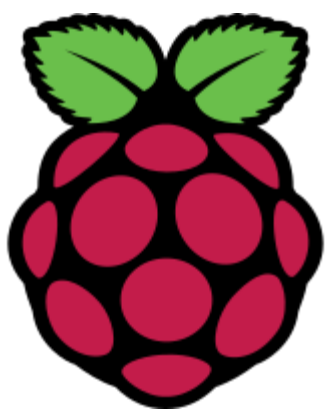
계속 해서 Distance를 받아오고 싶었지만 html에 return "ok"를 보내는 순간 로직이 끝나서 받아올 수가 없음



프로젝트1 만들면서 느낀 생각



1. 3색 LED는 결국 redPin, greenPin, bluePin 으로만 작동하기에 함수로 만들어서 코드를 줄일 수 있었다.
2. 부저 음악 키는것과 끄는것은 배운것을 바탕으로 하였기에 별 무리가 없었다.
3. UltraSonic으로 계속해서 거리의 값을 받아오고 웹페이지 화면에 표현하고 싶었다. 하지만 파이썬 UltraSonic 코드에서 return "ok"를 html 코드로 던져버리면 while문이 끝나기 때문에 거리를 한 번 밖에 받아올 수가 없었다. 생각을 더 해야할 문제인 것 같다.
4. html과 css의 지식이 부족해서 원하는 디자인의 결과가 나오지 않았기에 공부의 필요성을 느꼈다.



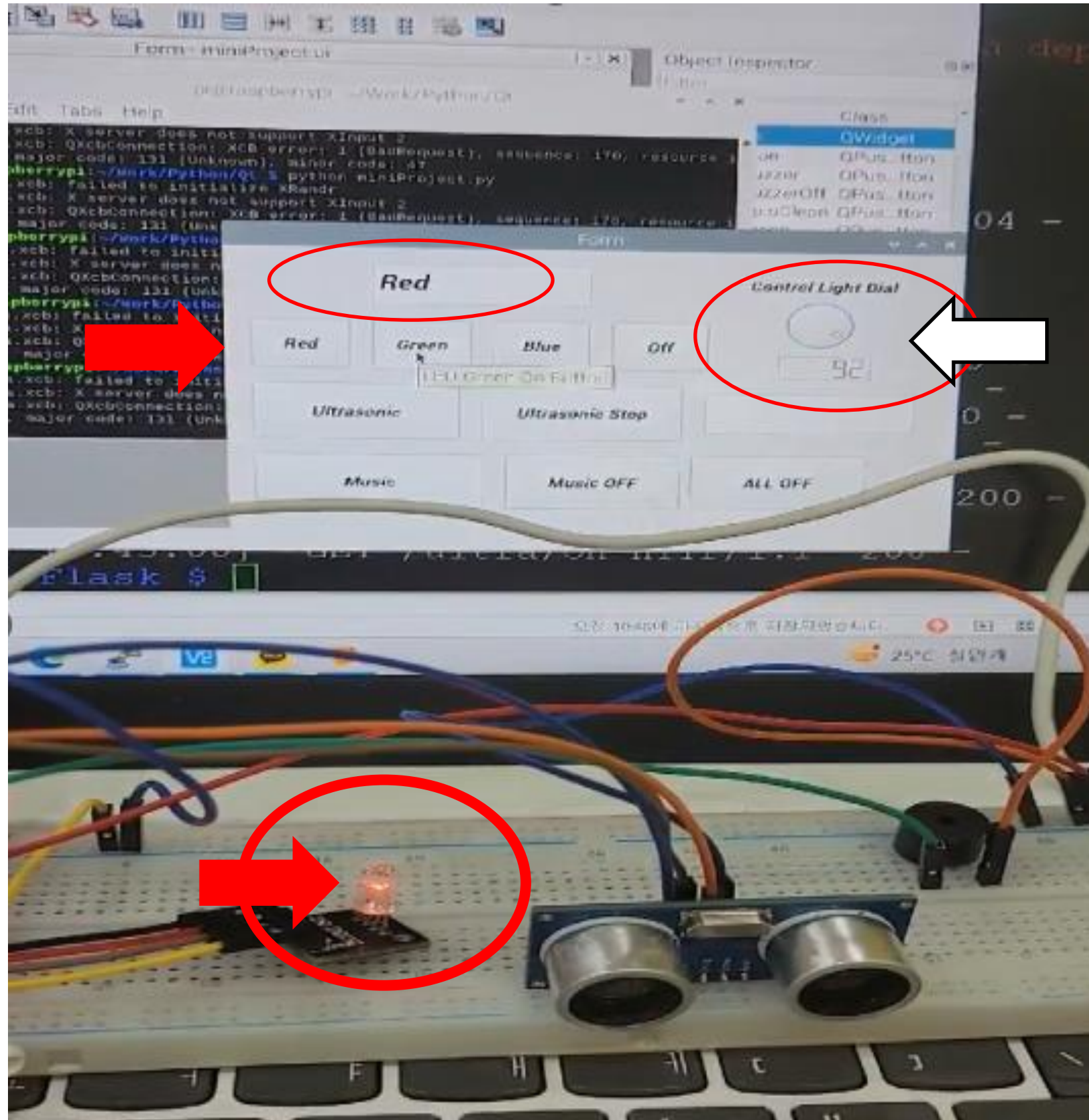
프로젝트2 개요



- 개발인원: 1명
- 개발 기간: 2023.06.14 ~ 2023.06.16
- 주요 기능:
 - 3색 LED 켜기 및 끄기
 - 부저 음악 켜기 및 끄기
 - 울트라소닉으로 사물과의 거리 측정
- 개발 언어: Python
- 간단 소개: PyQt를 이용하여 버튼을 클릭하여 원격 제어가 가능

기능 구현: 3색 LED 켜기 및 끄기

영상



다이얼을 사용하여 led밝기를 표현

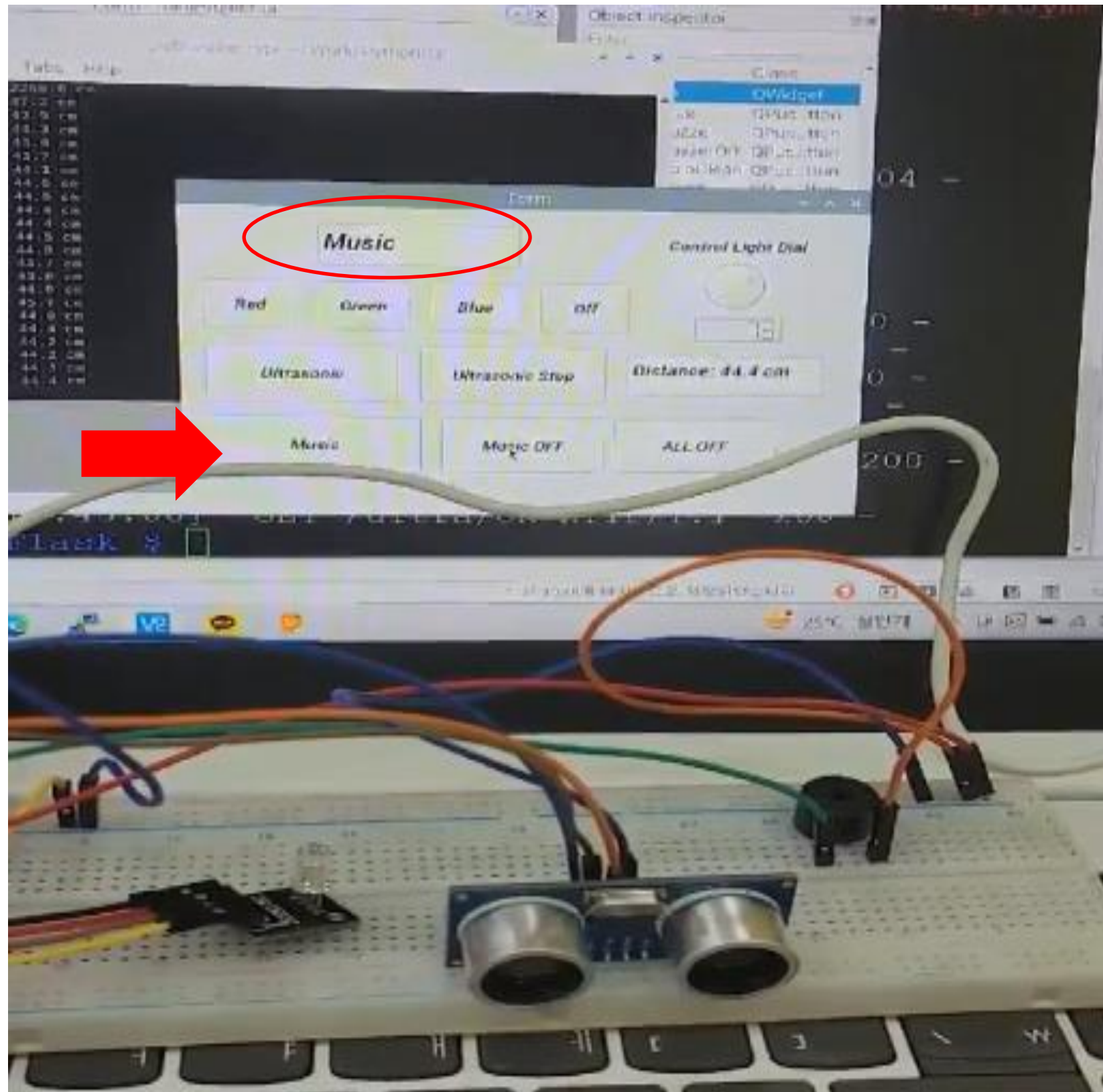
- 다이얼을 돌렸을 때 어떤 색의 led밝기가 달라져야 하는지 알아야함으로 변수 state를 선언
- Red 등의 색상 버튼을 클릭하면 다른 색의 LED는 꺼지게 하고싶음
- 버튼을 클릭 할 때마다 Red, Green, Blue 중 하나는 켜주고 나머지 두 개는 꺼야 하기에 3줄이 계속해서 사용됨
- 함수를 사용해서 코드를 줄이고 깔끔하게 함

영상은 여기서 확인 가능합니다.

<https://github.com/Haecms/My-Portfolio/tree/78e0bf0f99886ad814ae18521841179a66bdee66/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC%ED%8C%8C%EC%9D%B4/Video>

기능 구현: 부저 음악 켜기 및 끄기

영상



Music 버튼을 눌렀을 때 음악 켜기

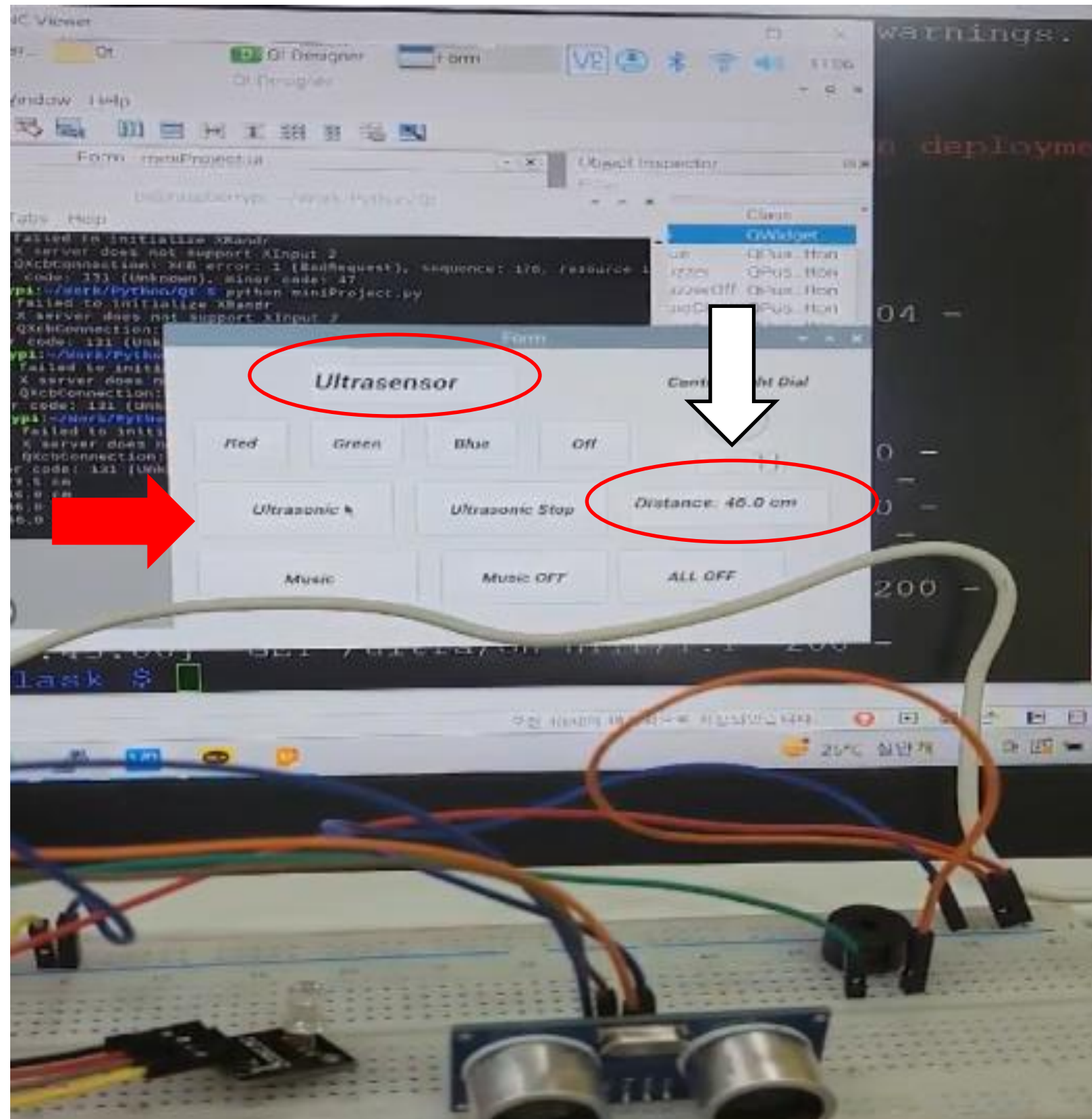
- 버튼을 누르면 음악이 흘러나옴
- Music OFF 버튼을 누르면 음악이 꺼짐
- 음악이 나오는 로직을 for문을 사용했기 때문에 음악이 흘러 나오는 중에 Music OFF 버튼을 누르면 탈출 가능해야 함
- QApplication.processEvents()를 사용하여 음악이 흘러나오는 중에 혹시라도 다음 버튼클릭 이벤트가 있다면 그 로직도 실행하게 함

영상은 여기서 확인 가능합니다.

<https://github.com/Haecms/My-Portfolio/tree/78e0bf0f99886ad814ae18521841179a66bdee66/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC%ED%8C%8C%EC%9D%B4/Video>

기능 구현: ULTRA SONIC

영상



UltraSonic 버튼을 눌렀을 때 거리 표현

- 버튼을 누르면 맞은편 물체와의 거리 표현

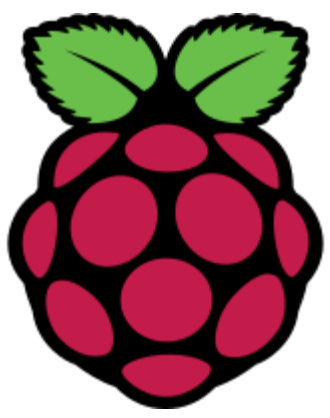
- UltraSonic Stop 버튼을 누르면 정지함

- 계속해서 거리를 나타내야 하기 때문에 while문을 사용했음 그래서 stop버튼을 눌렀을 때 탈출 로직을 생각해야 함

- 이것도 QApplication.processEvents()를 사용하여 다음 버튼클릭 이벤트가 있다면 그 로직도 실행하게 함

영상은 여기서 확인 가능합니다.

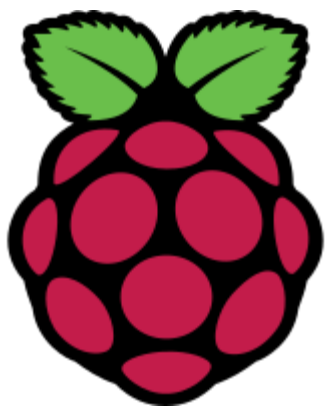
<https://github.com/Haecms/My-Portfolio/tree/78e0bf0f99886ad814ae18521841179a66bdee66/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC%ED%8C%8C%EC%9D%B4/Video>



프로젝트2 만들면서 느낀 생각



1. 3색 LED는 결국 redPin, greenPin, bluePin 으로만 작동하기에 함수로 만들어서 코드를 줄일 수 있었다.
2. UltraSonic과 Buzzer에서 QApplication.processEvents()를 사용하였는데 이 함수는 다음 이벤트가 있으면 그것도 실행시키는 것이기에 과부하가 올 수 있다. 가급적이면 사용하지 않는것이 좋겠다.



감사드립니다 !

조해찬 : <https://github.com/Haecms>

라즈베리파이 : <https://github.com/Haecms/My-Portfolio/tree/main/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC%ED%8C%8C%EC%9D%B4>

영상 주소:
<https://github.com/Haecms/My-Portfolio/tree/78e0bf0f99886ad814ae18521841179a66bdee66/%EB%9D%BC%EC%A6%88%EB%B2%A0%EB%A6%AC%ED%8C%8C%EC%9D%B4/Video>