

파이썬 프로그래밍

팀 '우리' 최종 프로젝트



소프트웨어학 20214152 우진
멀티미디어학 20172543 우정호



목차

1. 프로젝트 구상

2. 개발환경 구축

3. 사용된 센서 설명

4. 순서도/코드

5. 하드웨어

6. 프로그램 시연

프로젝트 주제 및 개발 환경 선정

프로젝트 주제 선정

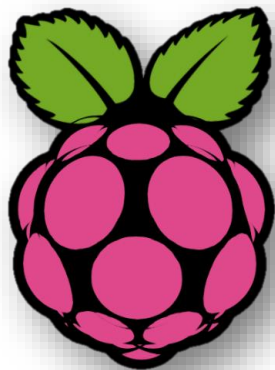
박물관 방문 시, 유물의 일정거리 이내로 접근하는 경우 경고등의 점등과 함께 경보음을 출력하는 시스템.

개발 환경 구상

파이썬과 범용성이 넓고 진입 장벽이 낮은 라즈베리파이를 이용해 프로젝트를 진행함.



프로젝트 개발 단계별 주요 내용



1. 프로젝트 아이디어 구상

파이썬을 이용한 프로젝트의 아이디어 구상 및 구체적 방향성과 툴을 결정.

2. 라즈베리파이 학습

기본적인 라즈베리파이 OS와 센서 등의 기초 활용 능력을 배양.

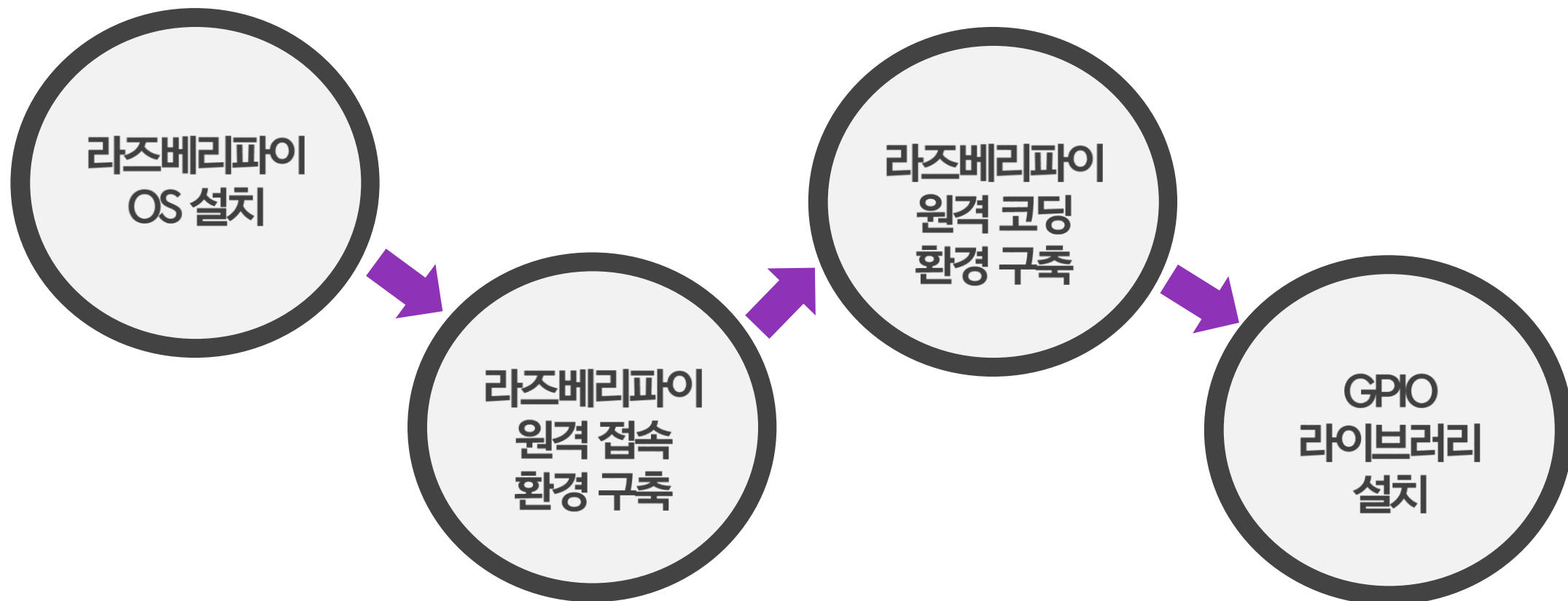
3. 센서의 구현

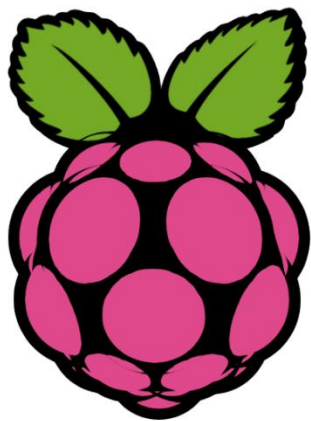
프로젝트에서 사용되는 센서와 코드를 유기적으로 연결 및 구현.

4. 프로젝트 완성

완성된 센서 설계와 코드로 하드웨어적 요소까지 완성시켜 하나의 프로젝트로 완성

라즈베리파이 사용을 위한
OS 설치부터 GPIO 라이브러리 설정까지.





Raspberry Pi

1. 라즈베리파이 이미지 설치
(Raspberry Pi imager)
2. 컴퓨터에 sd카드 연결
3. 이미지 실행 후, 라즈베리파이
os를 sd카드에 설치

02

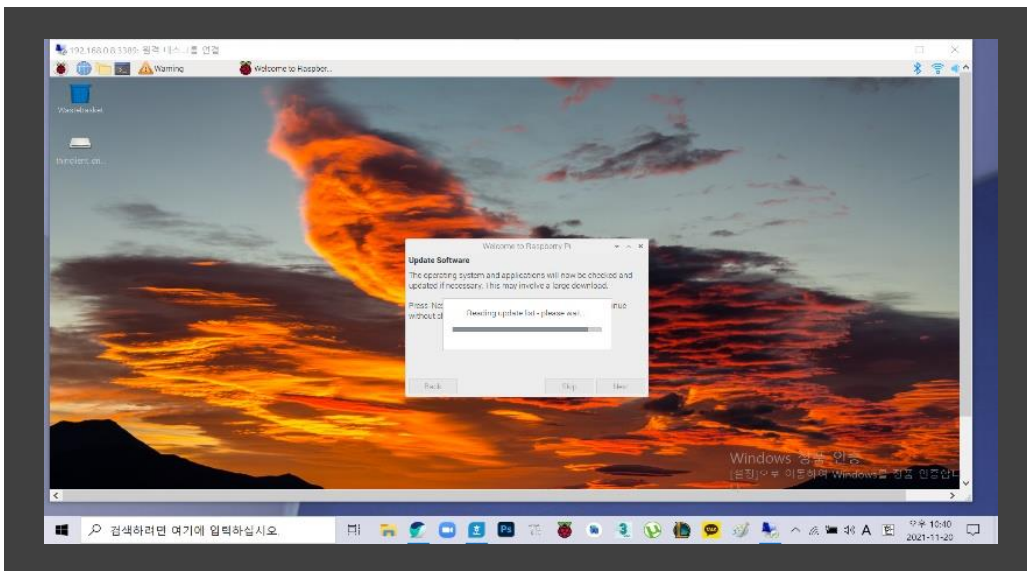
개발환경 구축 – 라즈베리파이 원격 접속 환경 구축

라즈베리파이 '원격 접속' 환경 설정하기

1. 원격 로그인을 위해 'ssh'기능 켜기

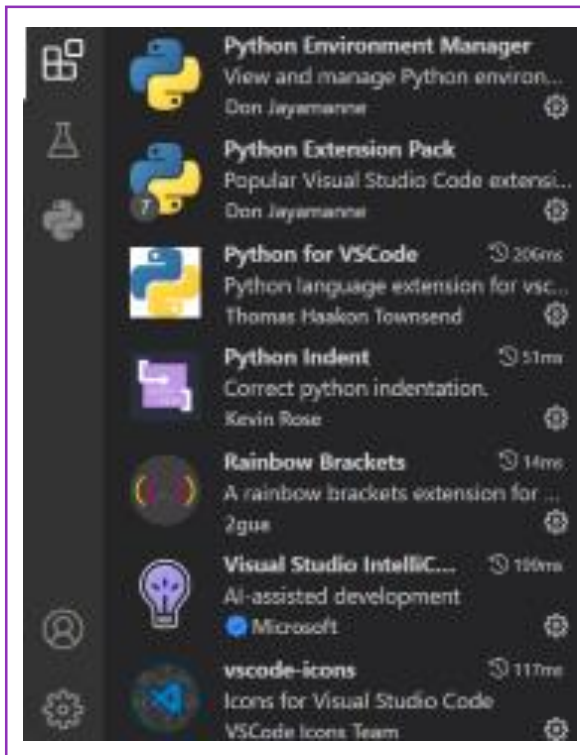
2. puTTY로 원격 접속 설정하기

3. 라즈베리파이 OS에 연결

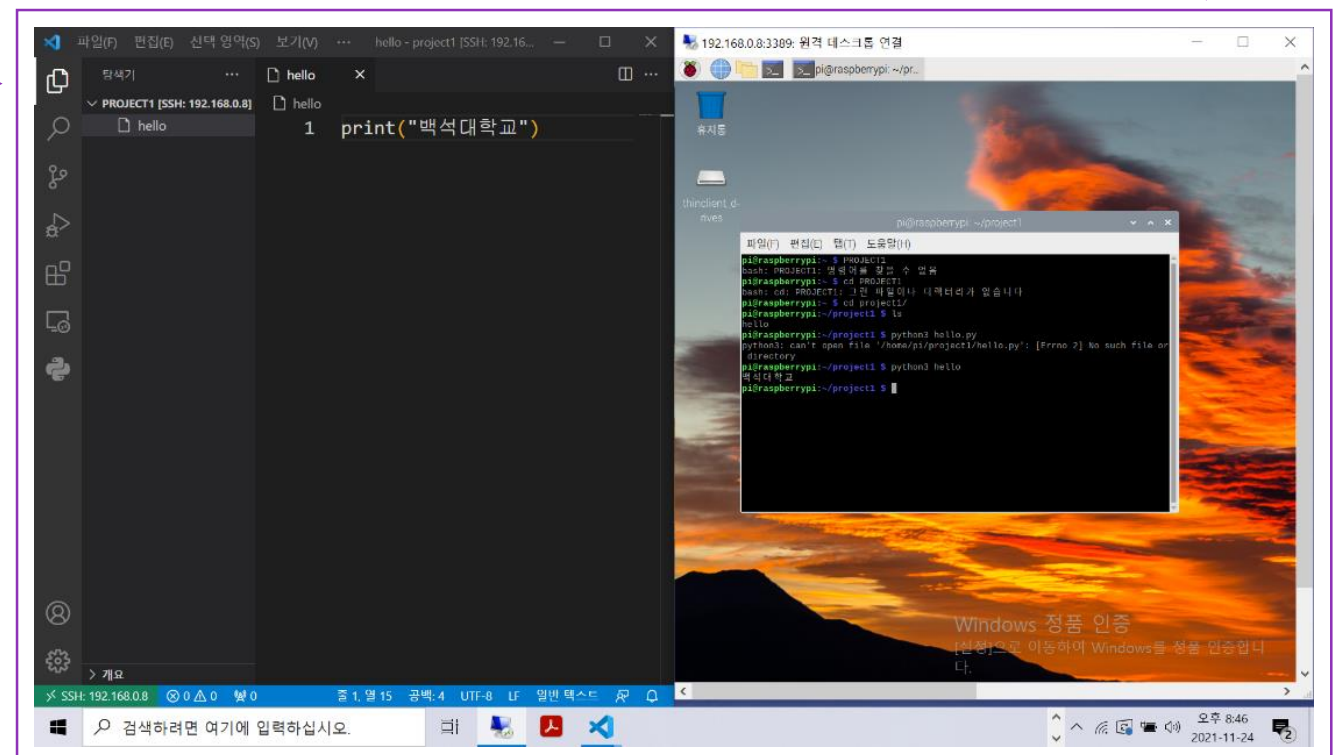




1. 파이썬, Vs Code 설치



2 원격 영역 연결 확인



3. 가상 환경 구축

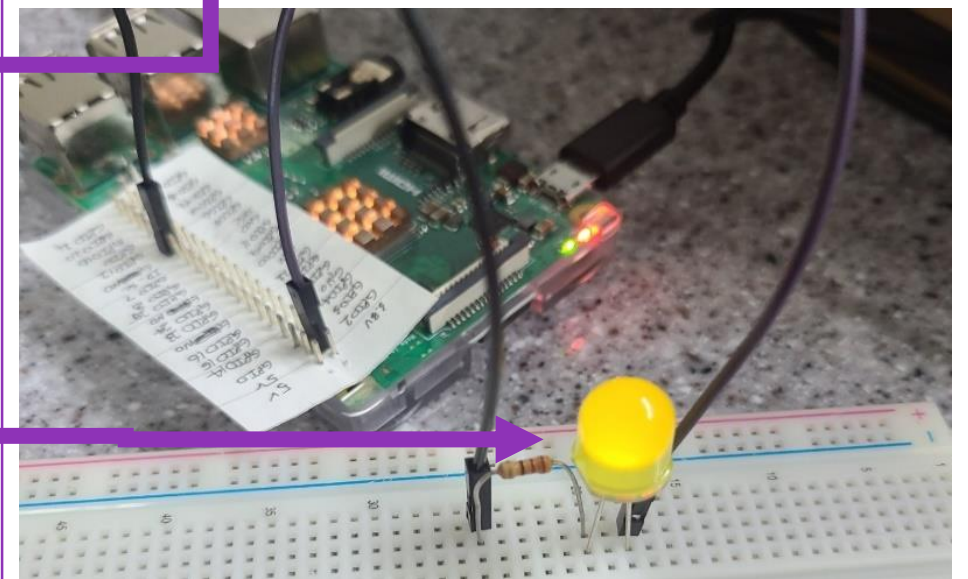
02

개발환경 구역 - GPIO 라이브러리 설치

1. GPIO 라이브러리 설치

2. GPIO 핀 표시

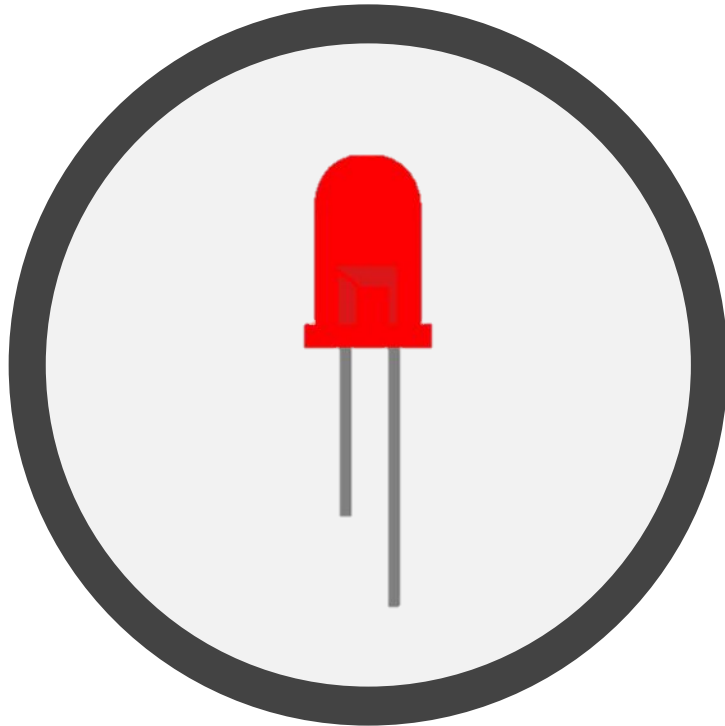
3. LED 점등시켜서 GPIO 체크



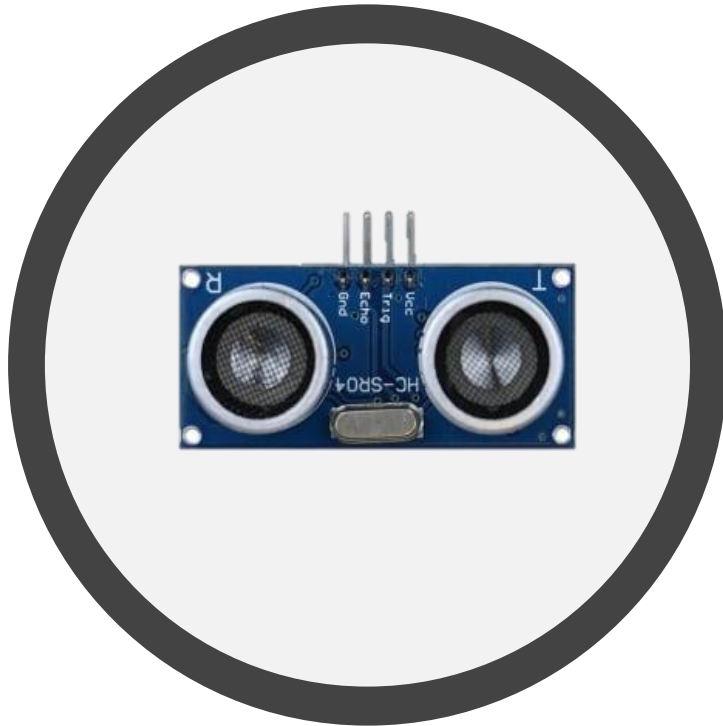
03

사용된 센서 설명

불빛이 들어왔다 나가는 LED

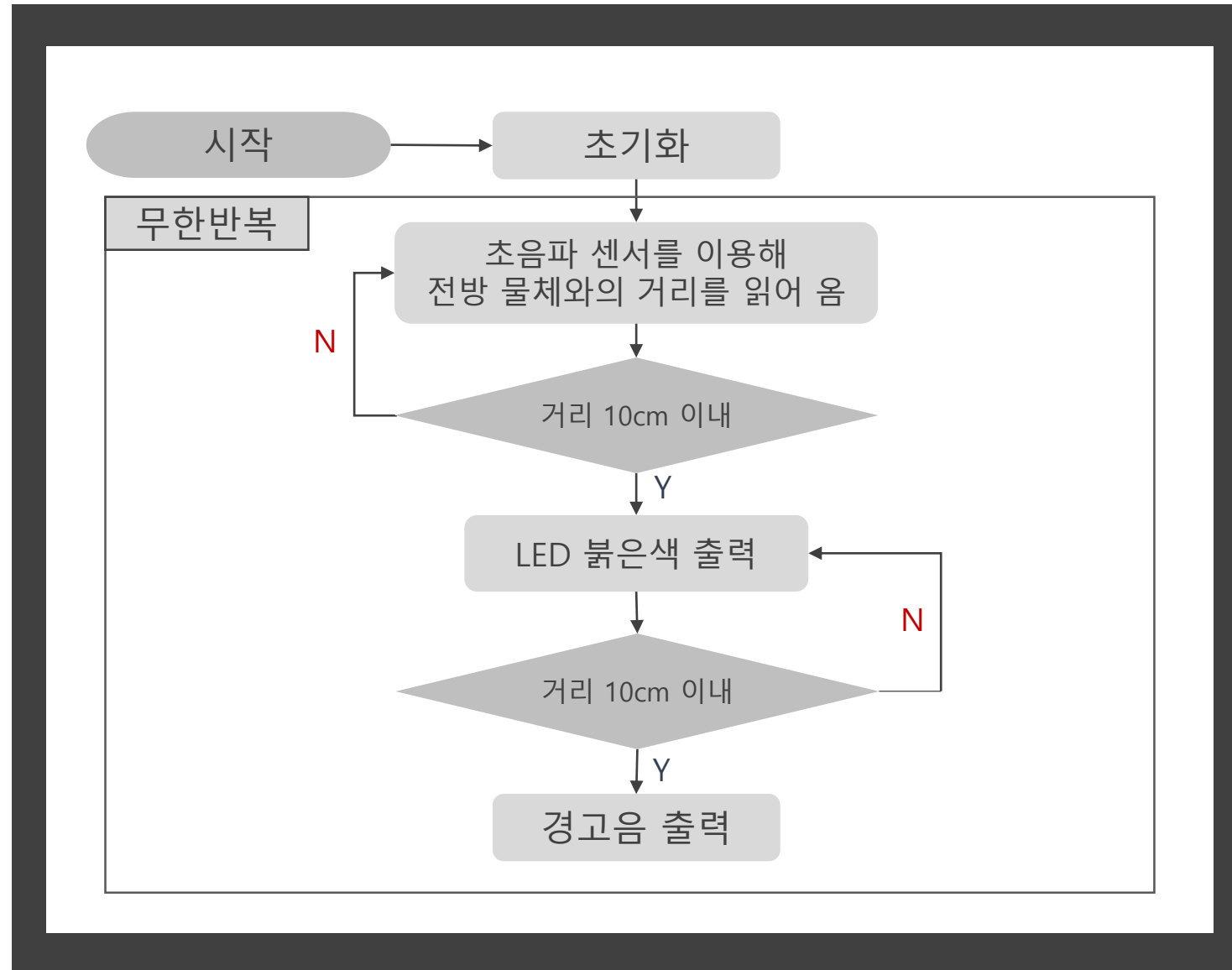


초음파 센서를 감지하는 HC-SR04



소리를 내는 부저 Buzzer





04

순서도/코드 - 코드

```
import RPi.GPIO as GPIO    #라즈베리파이 GPIO 삽입
import time                #시간 모듈 삽입

led = 5    #LED핀을 GPIO 5번핀으로 설정
buzzer = 11    #부저핀을 GPIO 11번핀으로 설정
trig = 3    #TRIG핀을 GPIO 3번핀으로 설정
echo = 2    #ECHO핀을 GPIO 2번 핀으로 설정

print('start')    #'start' 문구 출력

GPIO.setmode(GPIO.BCM)    #핀 입출력 모드를 BCM방식으로 지정(핀 번호를 GPIO모듈 번호로 사용)
GPIO.setwarnings(False)    #setwarning false 오류 제거

GPIO.setup(buzzer,GPIO.OUT)    #부저핀을 출력으로 설정
GPIO.setup(led,GPIO.OUT)    #LED핀을 출력으로 설정
GPIO.setup(trig,GPIO.OUT)    #TRIG핀을 출력으로 설정
GPIO.setup(echo,GPIO.IN)    #ECHO핀을 출력으로 설정
```

```
try:    #try문 수행
    GPIO.output(buzzer, False)    #부저를 출력하지 않는다고 선언
    GPIO.output(led, False)    #LED를 출력하지 않는다고 선언
while(True):    #while문 무한루프 수행
    GPIO.output(trig, False)    #TRIG을 출력하지 않는다고 선언
    time.sleep(0.5) #0.5초간 대기

    GPIO.output(trig, True)    #TRIG 출력
    time.sleep(0.00001)    #0.00001초 대기
    GPIO.output(trig, False)    #TRIG 출력 종료

    while GPIO.input(echo) == 0:    #ECHO가 0일 경우
        pulse_start = time.time()    #pulse_start에 시간을 저장

    while GPIO.input(echo) == 1:    #ECHO가 1일 경우
        pulse_end = time.time()    #pulse_end에 시간을 저장
```

```
pulse_duration = pulse_end - pulse_start    #pulse_end와 pulse_start의 차를 이용해 물체까지의 도달 시간(pulse_duration)을 계산
distance = pulse_duration * 17000            #거리(distance)는 도달 시간 * 17000
distance = round(distance, 2)                #소수점 둘째 자리까지 출력을 위한 ROUND함수 사용

print('Distance : ', distance, 'cm')         #거리를 출력(단위: cm)

if distance <= 10:    #거리가 10cm 이하일 경우
    GPIO.output(led, True)    #LED 실행
    GPIO.output(buzzer, True)    #부저 실행
    time.sleep(0.5)    #0.5초 간 대기
    GPIO.output(buzzer, False)    #부저 종료

else:
    GPIO.output(buzzer, False)    #부저 출력하지 않음
    time.sleep(0.3)    #0.3초 간 대기
    GPIO.output(led, False)    #LED 출력하지 않음

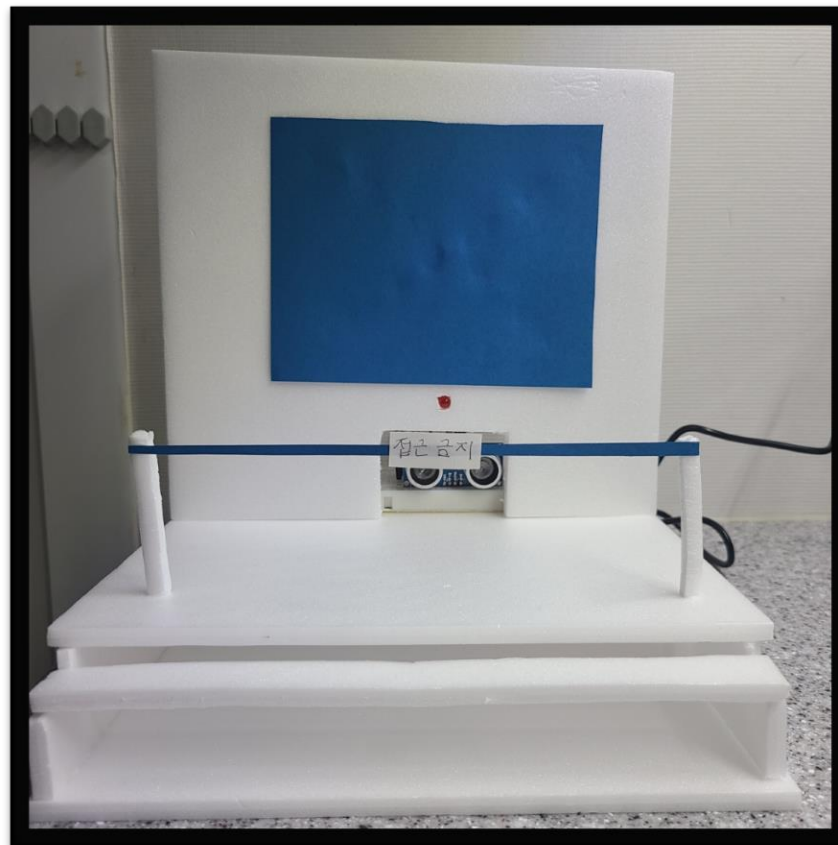
except:
    GPIO.cleanup()    #모듈 간 충돌 방지를 위해 GPIO핀을 초기화
```

05

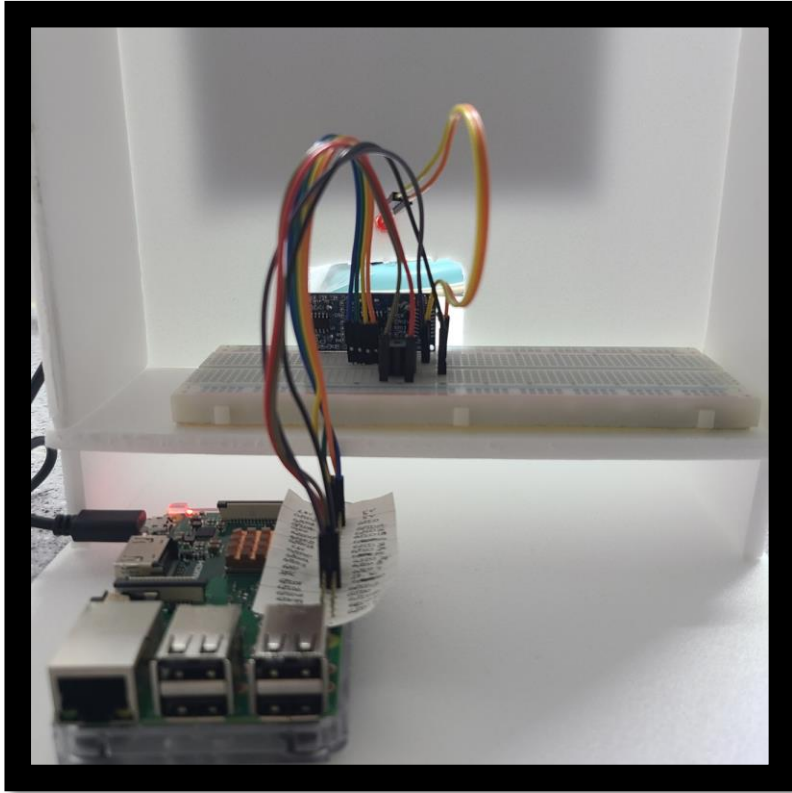
하드웨어



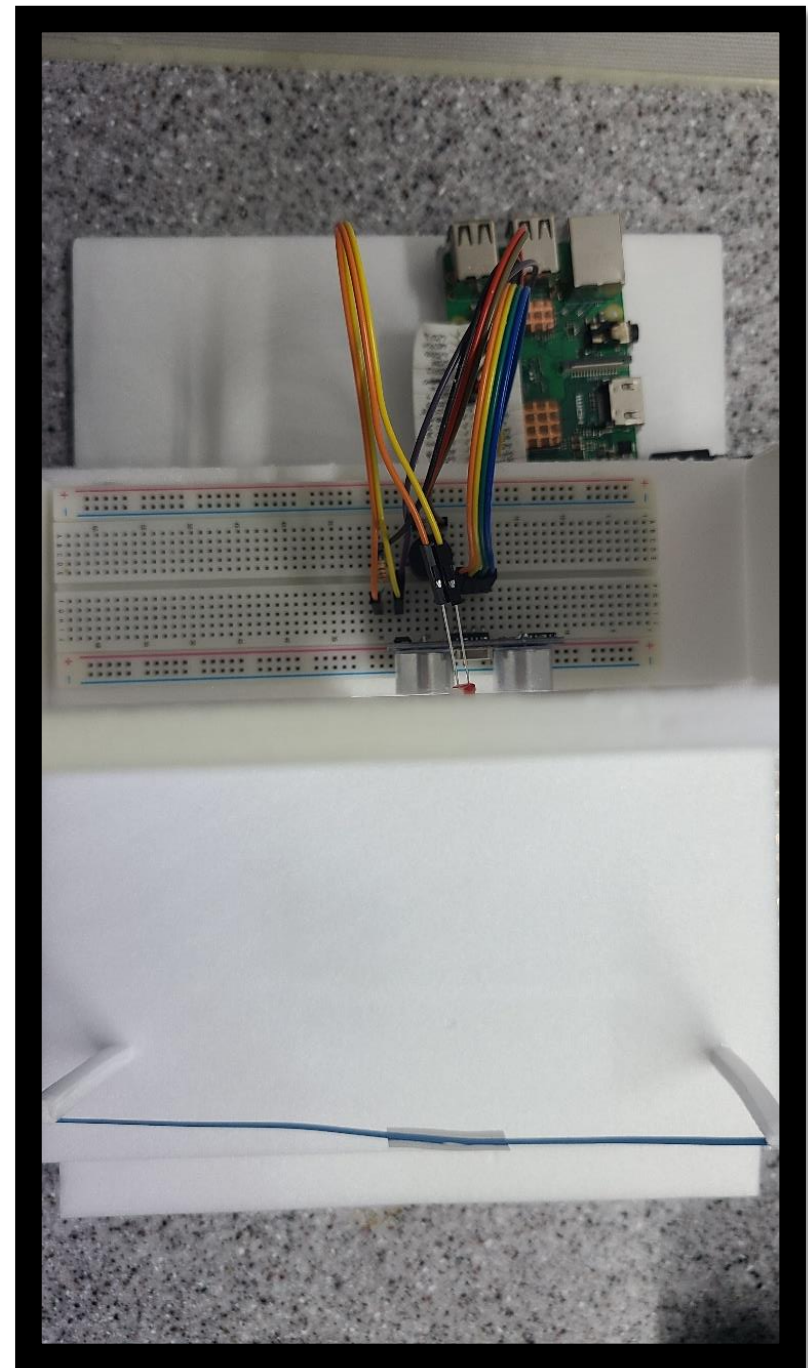
정면



05 하드웨어



후면

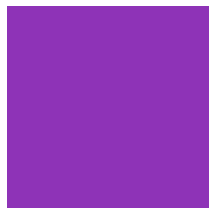


앞면

06

프로그램 시연





팀 '우리' 최종 프로젝트

발표를 들어주셔서 감사합니다.



소프트웨어학 20214152 우진
멀티미디어학 20172543 우정호

