

Copyright 2021. (Univ. of Seoul) All rights reserved.

- 수강생만 시청, 시청 후 삭제**
- 변경, 복사, 배포 절대 금지**

라즈베리파이 실습

- Python과 GPIO



Rpi 원격데스크탑 접속

- 사이트 문서 참고
 - <https://m.blog.naver.com/tipsware/220991540922>
- Rpi 패스워드 설정
 - 기본설정->Raspberry Pi Configuration->Change Password
- Rpi에서 프로그램 설치
 - # sudo apt-get purge realvnc-vnc-vncserver
 - # sudo apt-get install tightvncserver
 - # sudo apt-get install xrdp
 - # ifconfig // Rpi IP 주소 확인할 것
- 윈도우에서 원격 데스크탑 접속
 - Windows 보조프로그램 -> 원격데스크탑연결-> IP 주소 입력
 - Username: pi, Password: 변경한 암호 입력



Raspberry Pi

- 라즈베리 파이에서 주로 사용 되는 프로그래밍 언어
 - Python
 - C

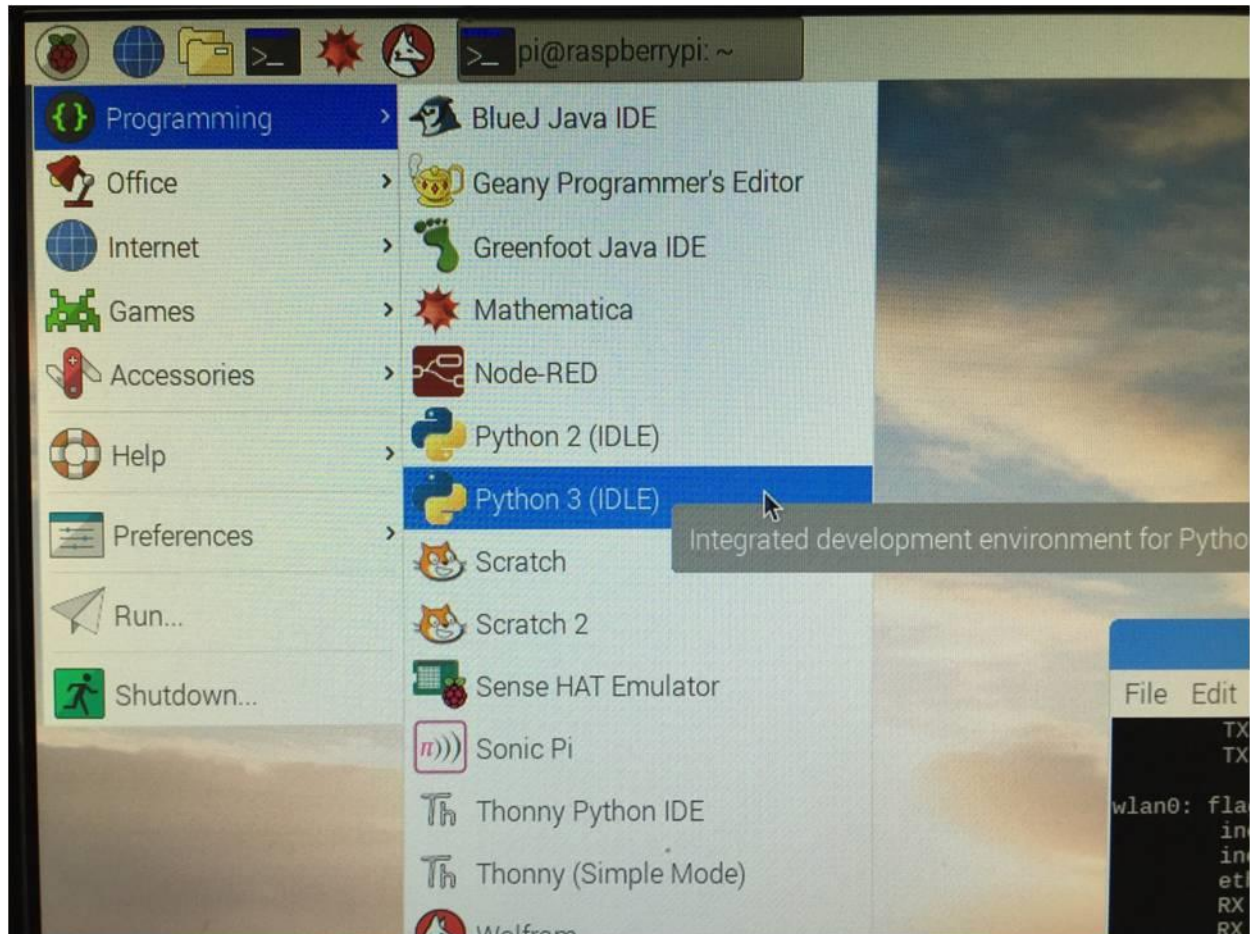


Python

- 1991년, 귀도 반 로섬이란 프로그래머에 의해 발표
- 플랫폼 독립적
- 인터프리터방식
- 객체지향적
- 동적 타이핑 대화형 언어
- 다양한 플랫폼에서 사용 가능
- 라이브러리(모듈)가 풍부
- 여러 연구, 교육 기관 및 산업계에서 이용

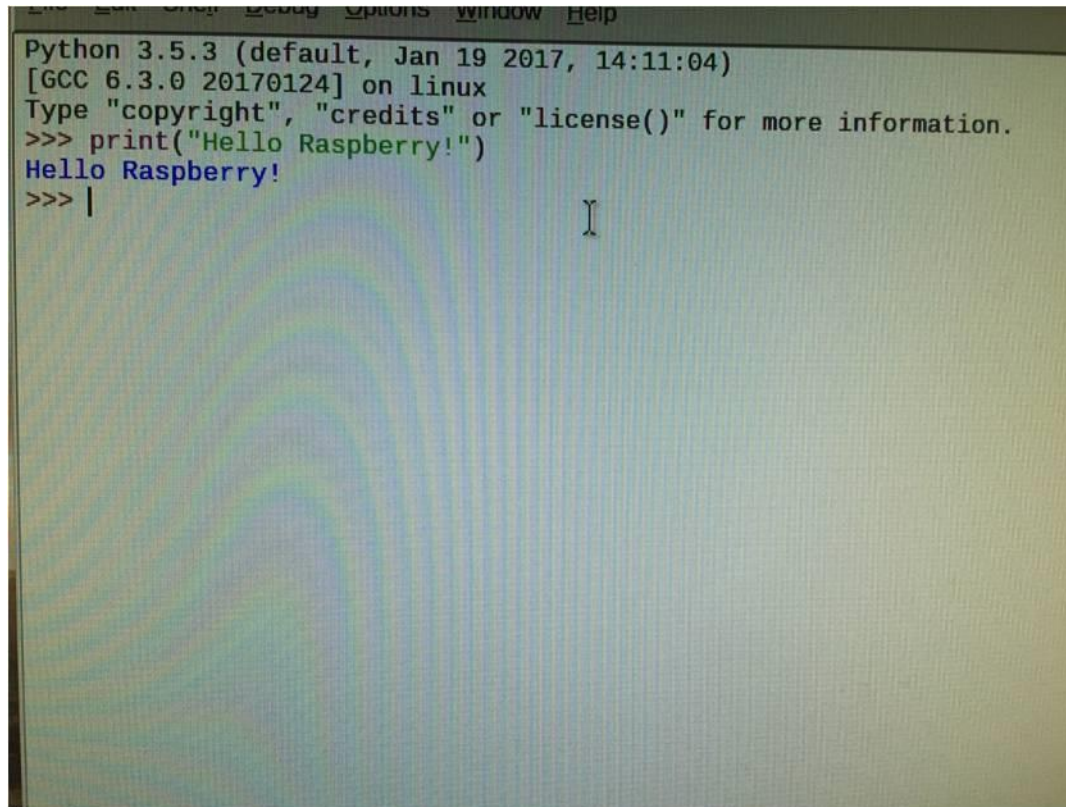


Python



Python

- `>>> print("Hello Raspberry!")`

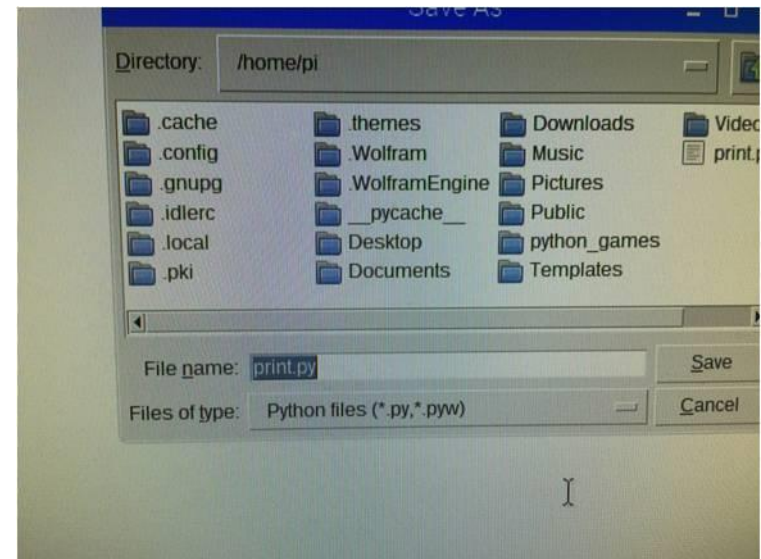
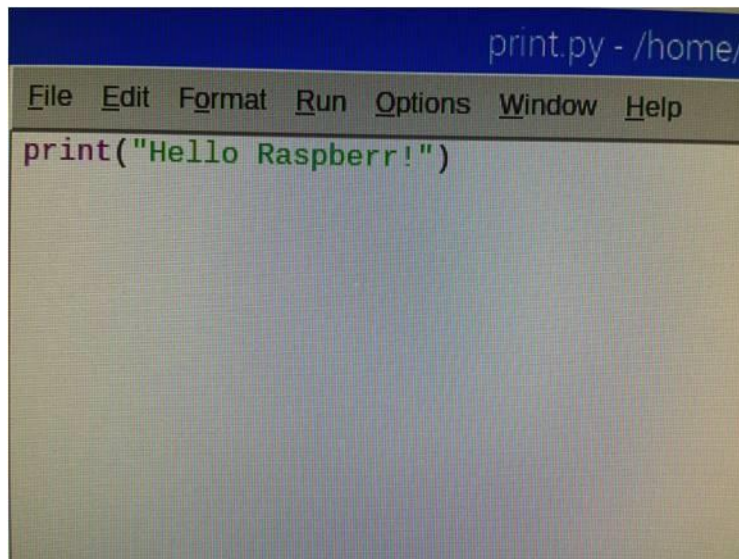


```
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "copyright", "credits" or "license()" for more information.
>>> print("Hello Raspberry!")
Hello Raspberry!
>>> |
```



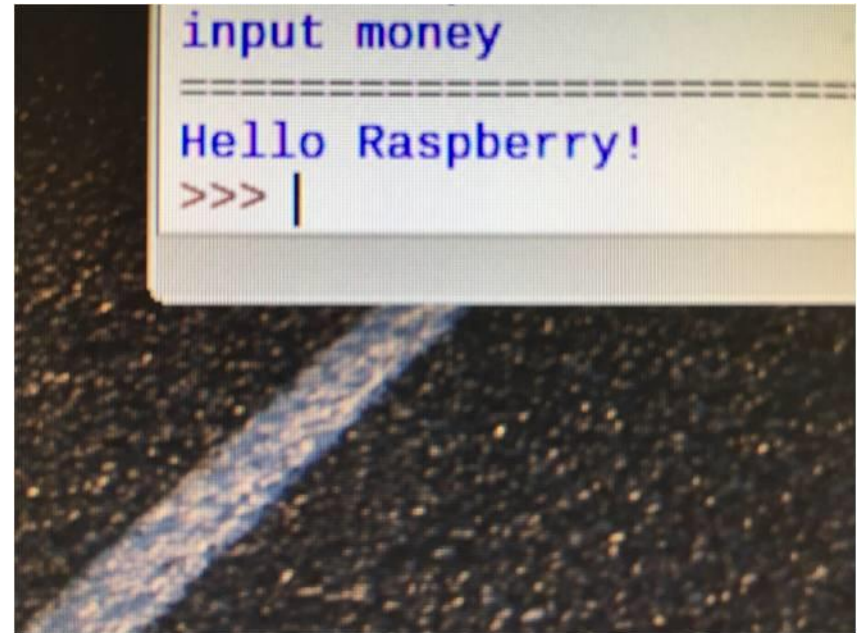
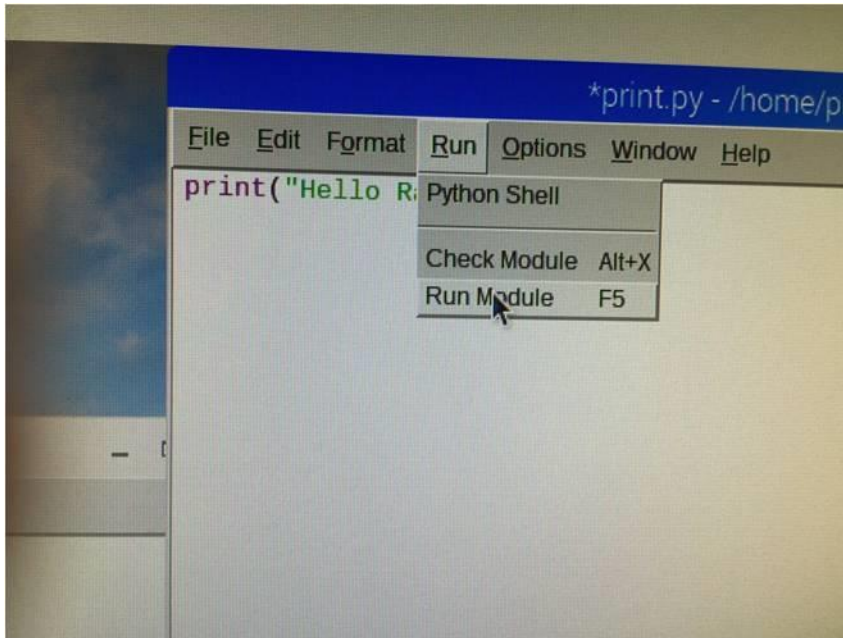
Python

- File -> New File
- `print("Hello Raspberry!")`
- Save
 - save "print.py"



Python

- F5 or Run-> Run Module



GPIO



Raspberry Pi GPIO Header

BCM	WiringPi	Name	Physical	Name	WiringPi	BCM
		3.3v	1	2	5v	
2	8	SDA.1	3	4	5V	
3	9	SCL.1	5	6	0v	
4	7	1-Wire	7	8	TxD	15 14
		0v	9	10	RxD	16 15
17	0	GPIO. 0	11	12	GPIO. 1	1 18
27	2	GPIO. 2	13	14	0v	
22	3	GPIO. 3	15	16	GPIO. 4	4 23
		3.3v	17	18	GPIO. 5	5 24
10	12	MOSI	19	20	0v	
9	13	MISO	21	22	GPIO. 6	6 25
11	14	SCLK	23	24	CE0	10 8
		0v	25	26	CE1	11 7
0	30	SDA.0	27	28	SCL.0	31 1
5	21	GPIO.21	29	30	0v	
6	22	GPIO.22	31	32	GPIO.26	26 12
13	23	GPIO.23	33	34	0v	
19	24	GPIO.24	35	36	GPIO.27	27 16
26	25	GPIO.25	37	38	GPIO.28	28 20
		0v	39	40	GPIO.29	29 21
BCM	WiringPi	Name	Physical	Name	WiringPi	BCM

BCM: Broadcom chip-specific pin numbers mode



wiringPi

- **\$ sudo apt-get install wiringpi**
- **\$ gpio -v gpio readall**



wiringPi

- C 언어의 gpio 핀 제어를 위한 라이브러리
- github를 통해 다운로드 가능
- python에서 사용되었던 BCM과 핀의 위치가 다르므로 주의



wiringPi

- `gpio -v gpio readall` 명령어로 설치 확인

```
pi@raspberrypi00:~/wiringPi $ gpio -v gpio readall
gpio version: 2.44
Copyright (c) 2012-2017 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
Type: Pi 3, Revision: 02, Memory: 1024MB, Maker: Unknown
* Device tree is enabled.
*--> Raspberry Pi 3 Model B Rev 1.2
* This Raspberry Pi supports user-level GPIO access.
```



wiringPi

- BCM 규격과 wPi 규격에서 스위치 확인
 - \$ gpio readall

```
pi@raspberrypi00:~/wiringPi $ gpio readall
```

-----Pi 3-----											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	IN	1	3	4		5v			
3	9	SCL.1	IN	1	5	6		0v			
4	7	GPIO. 7	OUT	1	7	8	0	IN	TxD	15	14
		0v			9	10	1	IN	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	0	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v			
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	1	IN	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21

GPIO pins

Raspberry Pi GPIO Header						
BCM	WiringPi	Name	Physical	Name	WiringPi	BCM
		3.3v	1	2	5v	
2	8	SDA.1	3	4	5V	
3	9	SCL.1	5	6	0v	
4	7	1-Wire	7	8	TxD	15 14
		0v	9	10	RxD	16 15
17	0	GPIO. 0	11	12	GPIO. 1	1 18
27	2	GPIO. 2	13	14	0v	
22	3	GPIO. 3	15	16	GPIO. 4	4 23
		3.3v	17	18	GPIO. 5	5 24
10	12	MOSI	19	20	0v	
9	13	MISO	21	22	GPIO. 6	6 25
11	14	SCLK	23	24	CE0	10 8
		0v	25	26	CE1	11 7
0	30	SDA.0	27	28	SCL.0	31 1
5	21	GPIO.21	29	30	0v	
6	22	GPIO.22	31	32	GPIO.26	26 12
13	23	GPIO.23	33	34	0v	
19	24	GPIO.24	35	36	GPIO.27	27 16
26	25	GPIO.25	37	38	GPIO.28	28 20
		0v	39	40	GPIO.29	29 21
BCM	WiringPi	Name	Physical	Name	WiringPi	BCM

BCM: Broadcom chip-specific pin numbers mode



LED in python

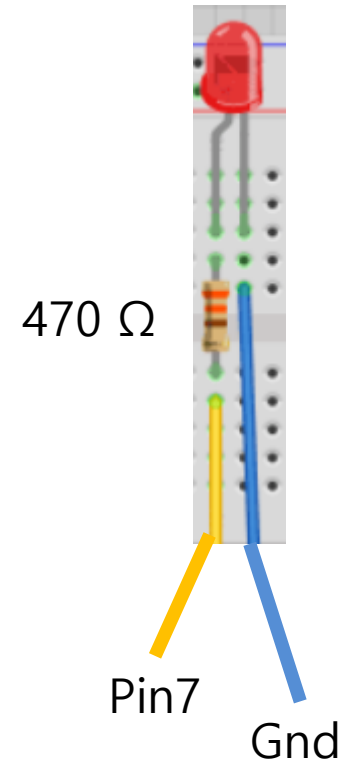
```
import RPi.GPIO as gpio  
import time
```

#Use the GPIO module of RPi
#Use time module

```
PIN = 7          # BCM: pin 4  
gpio.setmode(gpio.BOARD)  
gpio.setup(PIN, gpio.OUT)  
print("Press (CTRL-c) to exit")
```

```
try:  
    while True:  
        gpio.output(PIN, True)  
        time.sleep(0.1)  
        gpio.output(PIN, False)  
        time.sleep(0.1)
```

```
except KeyboardInterrupt:  
    gpio.cleanup()
```



Thonny - /home/pi/...

Thonny - /home/pi/Desktop/led.pi @ 6:26

New Load Save Run Debug Over Into Out Stop Zoom Quit [Switch to regular mode](#)

led.pi ✕

```
1 import RPi.GPIO as gpio #Use the GPIO module of RPi
2 import time             #Use time module
3
4 PIN = 7
5 gpio.setmode(gpio.BOARD)
6 gpio.setup(PIN, gpio.OUT)
7 print("Press (CTRL-c) to exit")
8
9 try:
10     while True:
11         gpio.output(PIN, True)
12         time.sleep(0.1)
13         gpio.output(PIN, False)
14         time.sleep(0.1)
15
16 except KeyboardInterrupt:
17     gpio.cleanup()
18
```

Shell

```
led.pi:6: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings
(False) to disable warnings.
  gpio.setup(PIN, gpio.OUT)
Press (CTRL-c) to exit

Python 3.7.3 (/usr/bin/python3)
>>>
```

파일(F)
pi@raspb
Python 3
[GCC 8.3
Type "he
>>> quit
pi@raspb



New



Load



Save



Run



Debug



Over



Into



Out



Stop



Zoom



Quit

Switch to
regular
mode

led.pi X

```
1 import RPi.GPIO as gpio
2 import time
3                                     #Use the GPIO module of RPi
4                                     #Use time module
5 PIN = 7
6 gpio.setmode(gpio.BOARD)
7 gpio.setup(PIN, gpio.OUT)
8 print("Press (CTRL-c) to exit)")
9
10 try:
11     while True:
12         gpio.output(PIN, True)
13         time.sleep(0.1)
14         gpio.output(PIN, False)
15         time.sleep(0.1)
16 except KeyboardInterrupt:
17     gpio.cleanup()
18
```

Shell

```
File "/home/pi/Desktop/led.pi", line 7
print("Press (CTRL-c) to exit)")
    ^
```

SyntaxError: invalid character in identifier

>>> %Run led.pi

Press (CTRL-c) to exit)

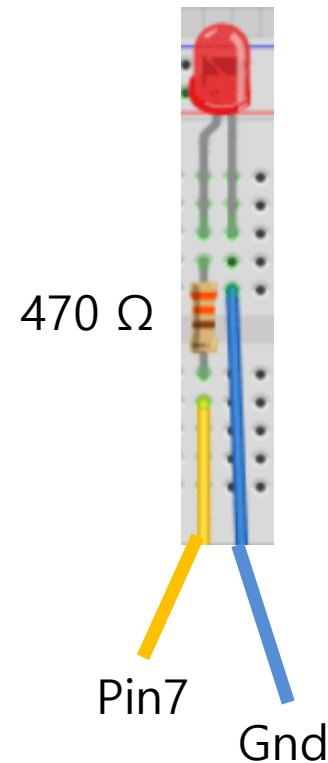
31개 항목 1개 항목 선택함 3.79MB

LED in C

```
#include <stdio.h>
#include <wiringPi.h> // include wiringPi library

#define LED 7 // BCM: pin 4

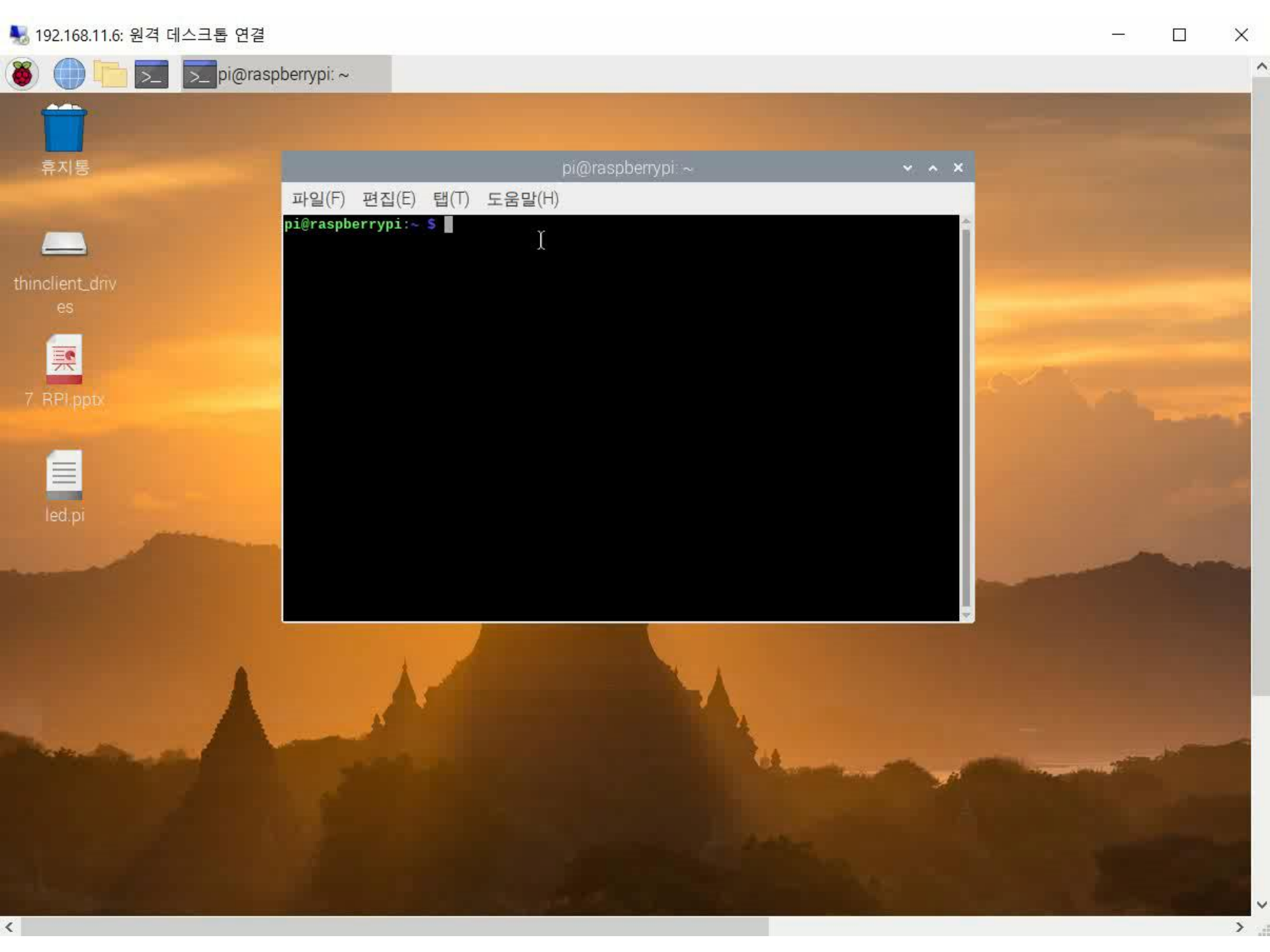
int main(void) {
    if (wiringPiSetup() == -1 ) { //init pin set
        printf("Error occurred!\n");
        return -1;
    }
    pinMode(LED, OUTPUT); // set LED to OUT
    while (1) {
        digitalWrite(LED, 1); //set LED HIGH
        delay(1000);
        digitalWrite(LED, 0); //set LED LOW
        delay(1000);
    }
    return 0;
}
```



LED in C

- `$ gcc -o led led.c -lwiringPi`
- `$./led`





휴지통



thinclient_drives



7_RPI.pptx



led.pi

```
pi@raspberrypi: ~
파일(F) 편집(E) 탭(T) 도움말(H)
pi@raspberrypi:~ $
```


led.c - /home/pi - Gea...

pi@raspberrypi: ~

파일(F) 편집(E) 탭(T) 도움말(H)

```
pinMode(LED, OUTPUT); // set LED to OUT
while (1) {
    digitalWrite(LED, 1); //set LED HIGH
    delay(1000);
    digitalWrite(LED, 0); //set LED LOW
    delay(1000);
}
return 0;
```

pi@raspberrypi:~ \$

pi@raspberrypi:~ \$

pi@raspberrypi:~ \$

pi@raspberrypi:~ \$

pi@raspberrypi:~ \$

pi@raspberrypi:~ \$

pi@raspberrypi:~ \$ gcc -o led led.c -lwiringPi

pi@raspberrypi:~ \$ ls led

led

pi@raspberrypi:~ \$ ls

Bookshelf Music

Desktop Pictures

Documents Public

Downloads RasplineStalker-master

pi@raspberrypi:~ \$./led

Templates led.c
Videos test_image.py
get-pip.py test_video.py
led thinclient drives

파일(F)

심볼 문서

함수

ma

매크로

LED [4]

```
4 #define LED // BCM: pin 4
5
6 int main(void) {
7     if (wiringPiSetup() == -1 ) { //init pin set
8         printf("Error occurred!\n");
9         return -1;
10    }
11    pinMode(LED, OUTPUT); // set LED to OUT
12    while (1) {
```

Switch in C

```
#include <stdio.h>
#include <wiringPi.h> // include wiringPi library

#define SW 7 // BCM: pin 4

int main(void) {
    if (wiringPiSetup() == -1 ) { //init pin set
        printf("Error occurred!\n");
        return -1;
    }
    pinMode(SW, INPUT); // set SW to IN
    //Continued on the back
    while (1) {
        if (digitalRead(SW) == 0) {
            printf("Button pressed\n");
            delay(1000);
            printf("Press (CTRL-c) to exit\n");
        }
    }
    return 0;
}
```



1k Ω

Gnd

3.3V

Pin7

5V와 연결하면, 저항
2개를 사용하여 Pin7
입력 전압을
3.3V로 낮춰야 함
(다음 실험 참고)

Switch in C

- `$ gcc -o switch switch.c -lwiringPi`
- `$./switch`





휴지통



thinclient_drives

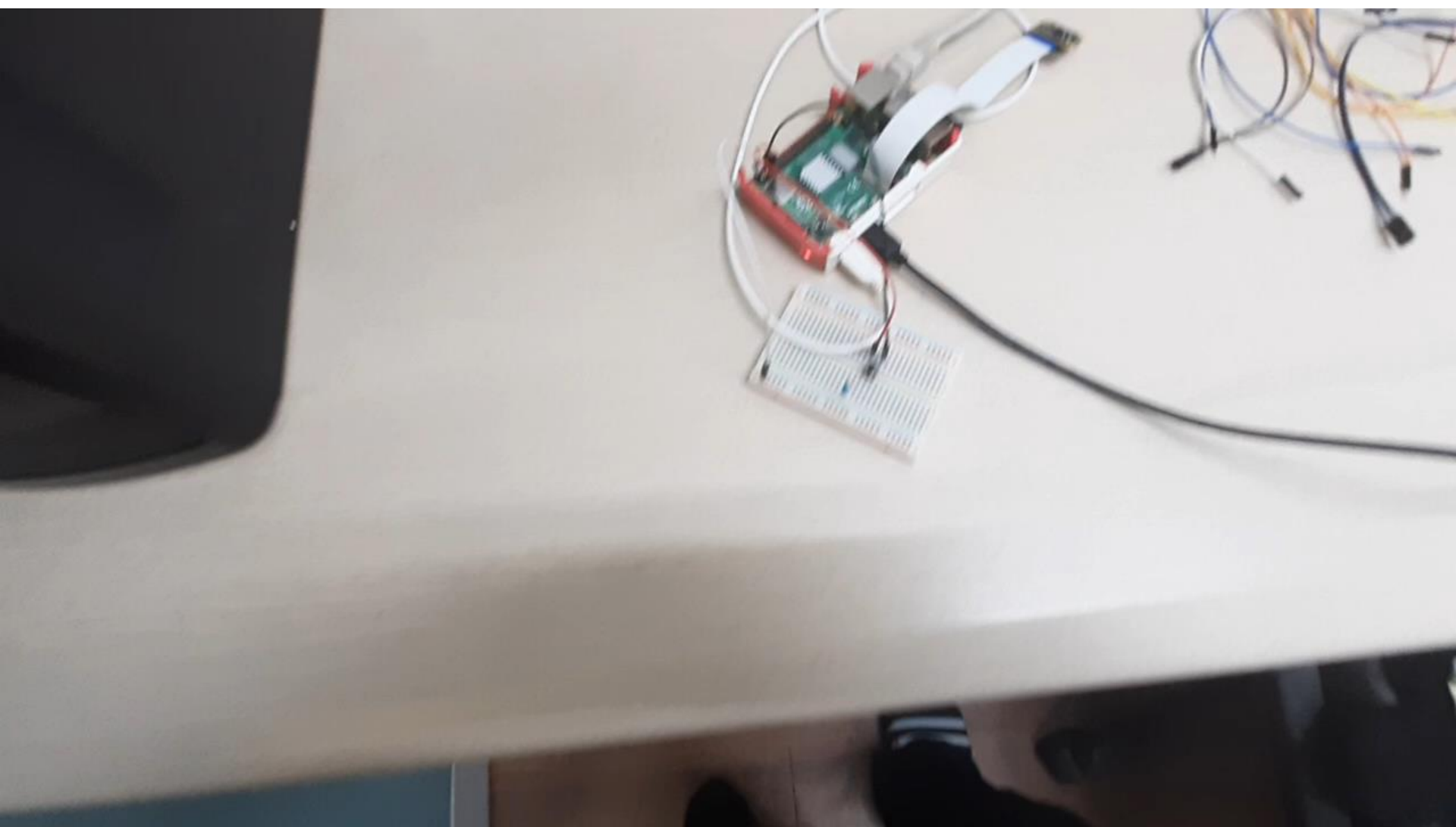


7.RPI.pptx



led.pi

```
pi@raspberrypi: ~
파일(F) 편집(E) 탭(T) 도움말(H)
[2]+ Done geany switch.c
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
```



Switch in Python

```
import RPi.GPIO as gpio #Use the GPIO module of RPi  
import time #Use time module
```

```
gpio.setmode(gpio.BCM)  
gpio.setup(4 , gpio.IN)  
print("Press the button")
```

Board pin 7 = BCM 4

```
try:  
    while True :  
        if gpio.input(4)==0:  
            print("Button pressed!")  
            time.sleep(1)  
            print("Press the button (CTRL-C to exit)")  
  
except KeyboardInterrupt:  
    gpio.cleanup()
```

pi@raspberrypi: ~

pi@raspberrypi: ~

파일(F) 편집(E) 탭(T) 도움말(H)

```
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
Button pressed
Press (CTRL-c) to exit
^C
pi@raspberrypi:~$
```

휴지통

thinclient_driv
es

7.RPI.pptx



led.pi



switch.pi

라즈베리파이 실습

- 초음파센서



GPIO pins



Raspberry Pi GPIO Header

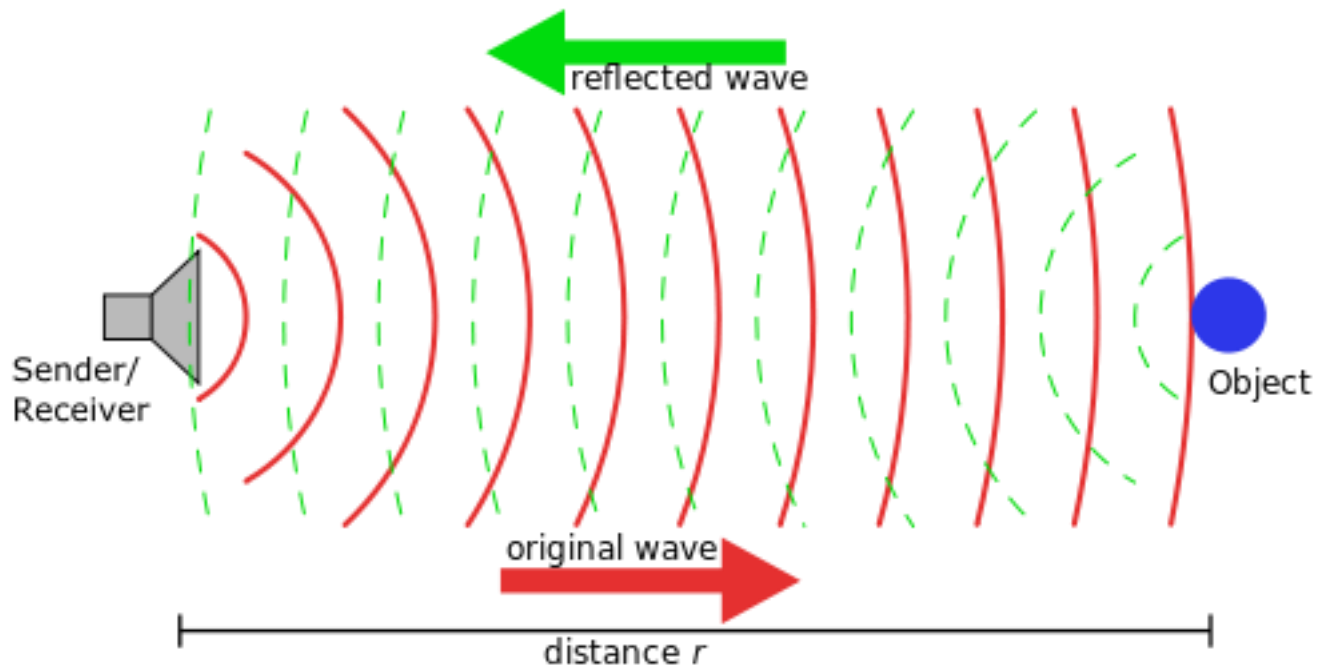
BCM	WiringPi	Name	Physical	Name	WiringPi	BCM
		3.3v	1	2	5v	
2	8	SDA.1	3	4	5V	
3	9	SCL.1	5	6	0v	
4	7	1-Wire	7	8	TxD	15
		0v	9	10	RxD	16
17	0	GPIO. 0	11	12	GPIO. 1	1
27	2	GPIO. 2	13	14	0v	18
22	3	GPIO. 3	15	16	GPIO. 4	4
		3.3v	17	18	GPIO. 5	5
10	12	MOSI	19	20	0v	24
9	13	MISO	21	22	GPIO. 6	6
11	14	SCLK	23	24	CE0	10
		0v	25	26	CE1	11
0	30	SDA.0	27	28	SCL.0	31
5	21	GPIO.21	29	30	0v	1
6	22	GPIO.22	31	32	GPIO.26	26
13	23	GPIO.23	33	34	0v	12
19	24	GPIO.24	35	36	GPIO.27	27
26	25	GPIO.25	37	38	GPIO.28	28
		0v	39	40	GPIO.29	29
BCM	WiringPi	Name	Physical	Name	WiringPi	BCM



Ultrasound sensor

- 초음파센서

- 초음파를 전방으로 출력, 반사돼 돌아오기까지 시간을 측정

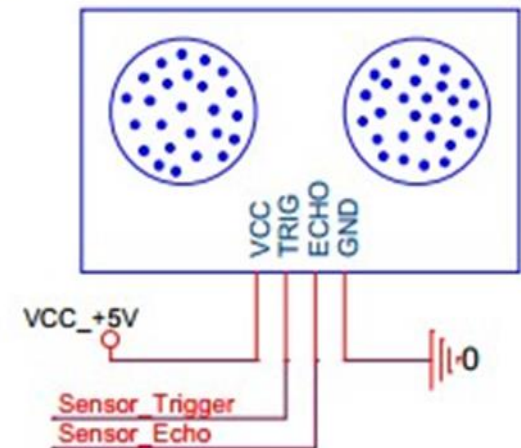


Ultrasound sensor

- Trigger: 초음파 센서의 송신부로 신호를 send
- Echo: 초음파 센서의 수신부에서 신호를 receive

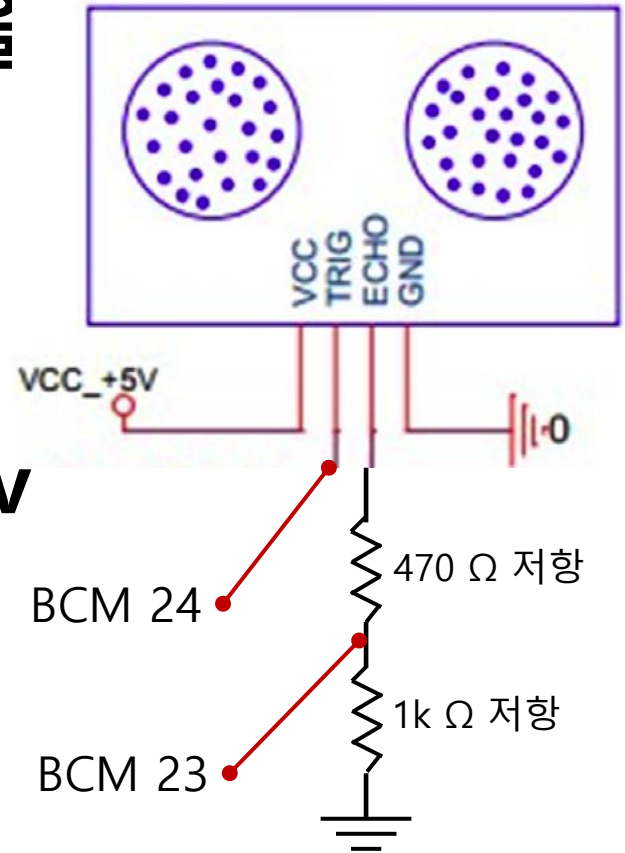


초음파센서



Ultrasound sensor

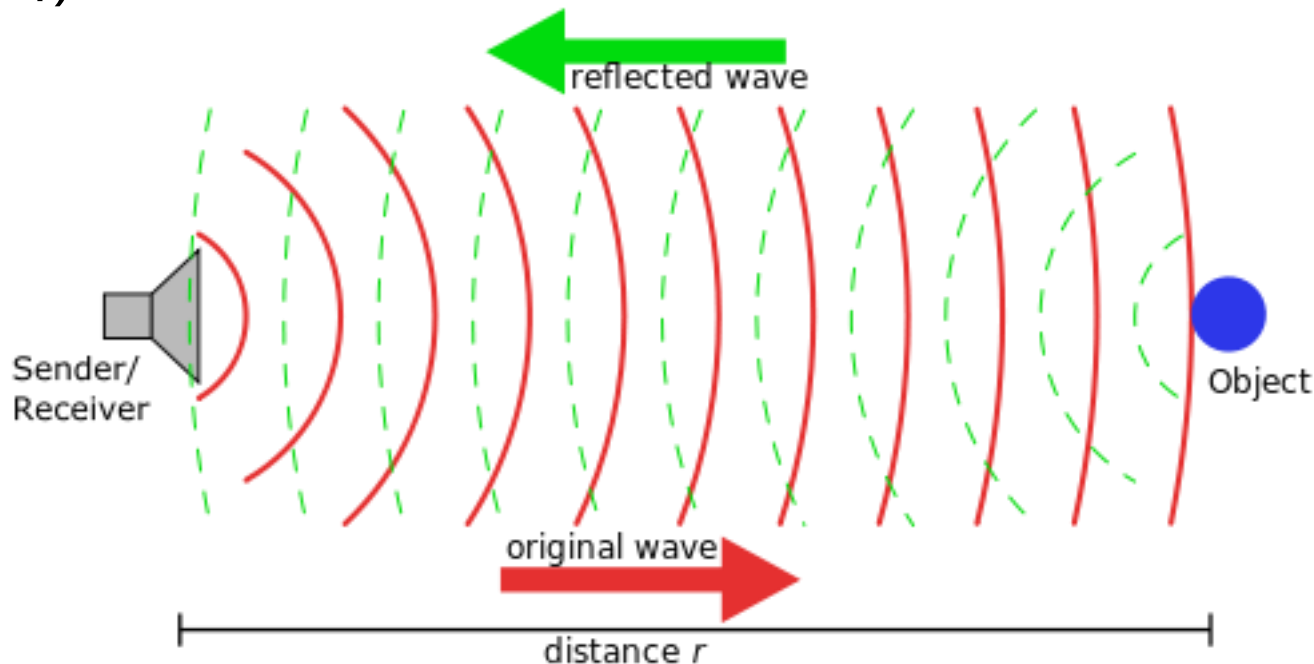
- 라즈베리파이는 3.3v 입력을 요구
- 초음파 센서 Echo 핀은 5v 신호 제공
- $1k\ \Omega$ 과 $470\ \Omega$ 저항으로 5v 를 3.3v 로 변환



Ultrasound sensor

- 측정 원리

- 초음파는 초당 340m를, 1cm를 29us동안 이동
- (초음파 송신 ~ 수신까지 걸린 시간) / 58(왕복 거리)



Ultrasonic in python

```
import RPi.GPIO as gpio #Use the GPIO module of RPi
import time #Use time module

gpio.setmode(gpio.BCM) #Works with gpio in BCM(Broadcom chip-specific pin numbers mode)

trig = 24
echo = 23

print("start ultrasonic sensor")

gpio.setup(trig, gpio.OUT) #Set trigger pin to OUT
gpio.setup(echo, gpio.IN) #Set echo pin to IN

try: #for Exception
    while True:
```



Ultrasonic in python

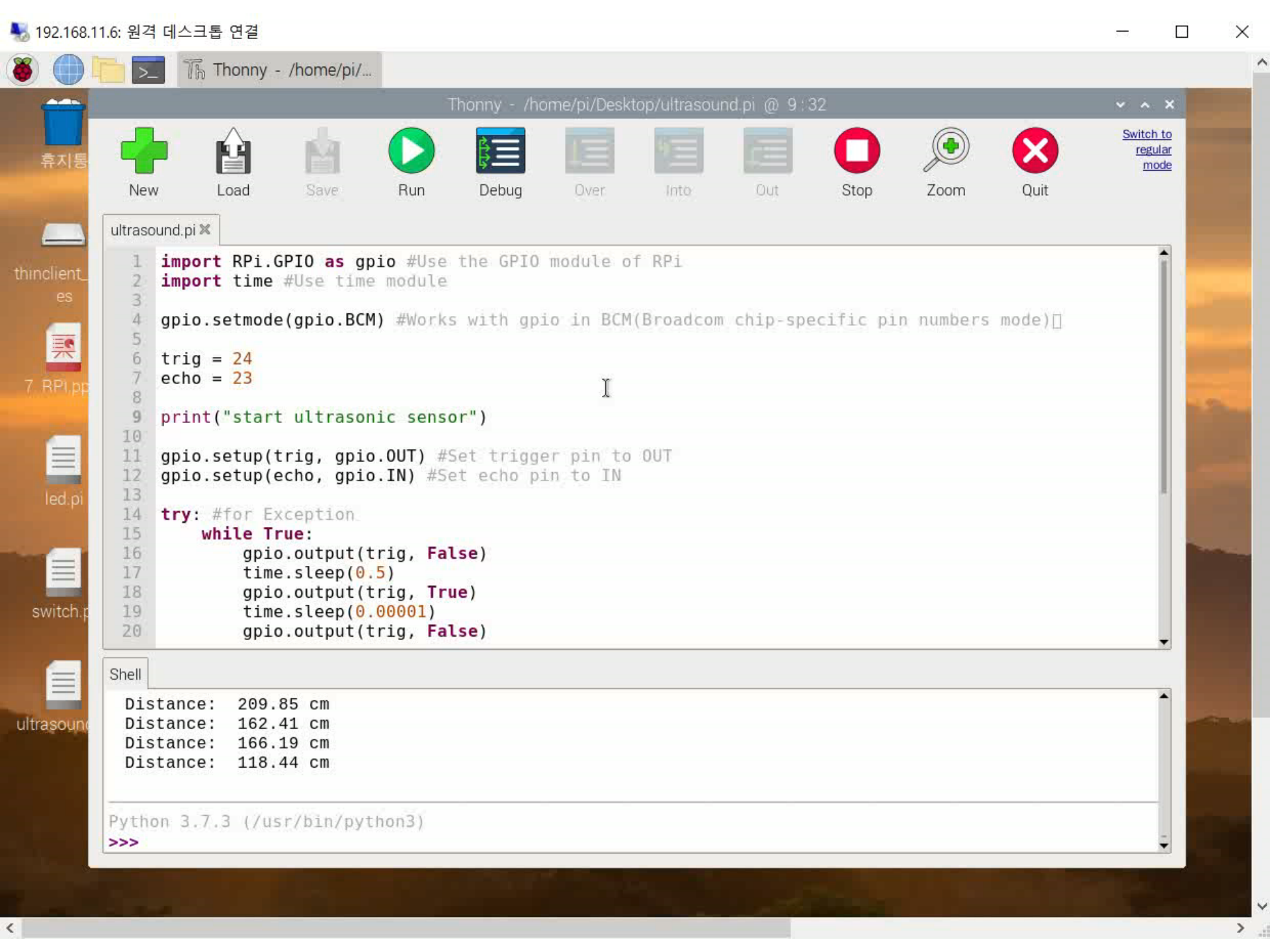
```
gpio.output(trig, False)
time.sleep(0.5)
gpio.output(trig, True)
time.sleep(0.00001)
gpio.output(trig, False)

while gpio.input(echo) == 0:
    pulse_start = time.time() #return the time since January 1, 1970
while gpio.input(echo) == 1:
    pulse_end = time.time() #return the time since January 1, 1970

pulse_duration = pulse_end - pulse_start
distance = pulse_duration * 17000 #time * speed / 2
distance = round(distance, 2) #Cut in second decimal place
print("Distance: ", distance, "cm")

except:
    gpio.cleanup()
```





Ultrasonic in C

```
/* ultrasonic.c */
#include <stdio.h>
#include <wiringPi.h>

#define TRIG 5
#define ECHO 4

int main(void)
{
    long distance = 0, startTime, travelTime;

    if (wiringPiSetup() == -1) {
        return 1;
    }

    pinMode(TRIG, OUTPUT);
    pinMode(ECHO, INPUT);
```



Ultrasonic in C

```
while (1) {  
    digitalWrite(TRIG, LOW);  
    usleep(2);  
    digitalWrite(TRIG, HIGH);  
    usleep(20);  
    digitalWrite(TRIG, LOW);  
  
    while (digitalRead(ECHO) == LOW);  
    startTime = micros();  
    while (digitalRead(ECHO) == HIGH);  
    travelTime = micros() - startTime;  
  
    distance = travelTime / 58;  
    printf("Distance: %ld\n", distance);  
    delay(100);  
}
```



