

삼성 청년 SW 아카데미

Embedded Project

day 1

DEMO – day1. AR시계

금일 교육 내용

1. 라즈베리파이 setup
2. Oled display 작동하기
3. Pillow (Python Image Library)
4. Datetime module
5. AR glass 조립
6. GPIO / Button 입력

■ AR시계 기능명세

- Raspberry pi zero w와 0.96" monochrome oled display를 사용하여 시계/ 캘린더를 제작한다.
- tact switch(모드버튼)을 사용해 **시계모드와 달력모드**를 전환한다.
- 모드 버튼은 **인터럽트**를 사용하여 언제라도 즉각적으로 반응하도록 한다.
- 중요한 정보가 잘 드러나면서도 시각적으로 균형잡히도록 **둘 이상의 크기로 문자를 표시**한다.
- 시계모드, 달력 모드 외에 차후 **다른 모드가 추가될 수 있도록 확장성을 고려한 구조**로 프로그램 제작한다.

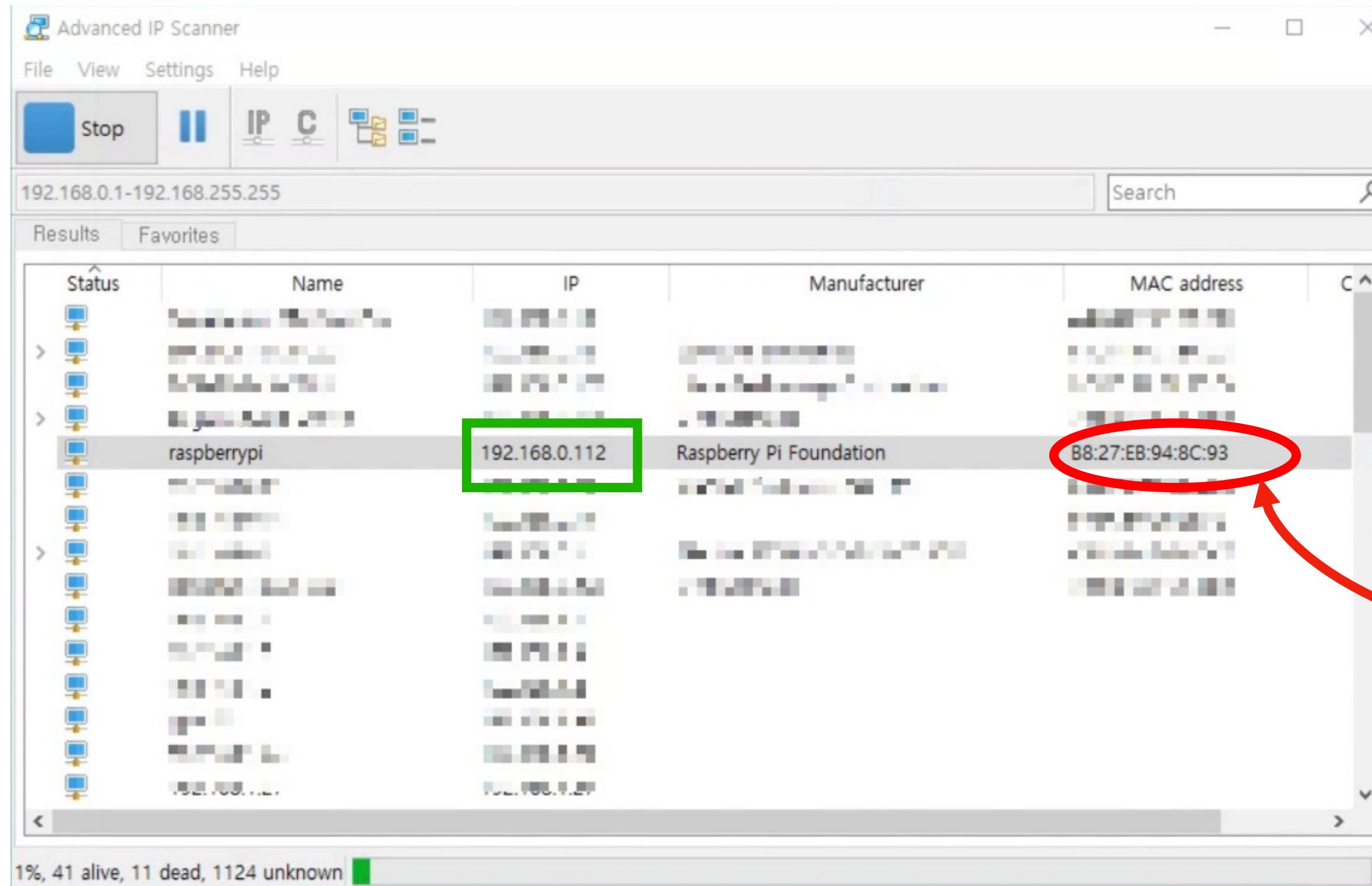
■ AR시계 기능명세

- 제공된 템플릿을 조립하여 **착용가능한 AR 시계/ 캘린더 디바이스**로 발전시킨다.
- 제공된 템플릿을 기본으로 제작하되, 사용자 편의성과 제작 용이성, 심미성을 고려하여 본인의 아이디어를 추가하도록 한다.
- 착용상태에서 시간과 날짜가 시야에 오버레이되어보인다. (5m 이상 먼 곳을 보았을 때 디스플레이 화면에 눈의 촛점이 맞도록 한다.)

라즈베리파이 setting-up

■ IP address 찾기

- Advanced IP Scanner 사용
(download: <https://advanced-ip-scanner.com>)



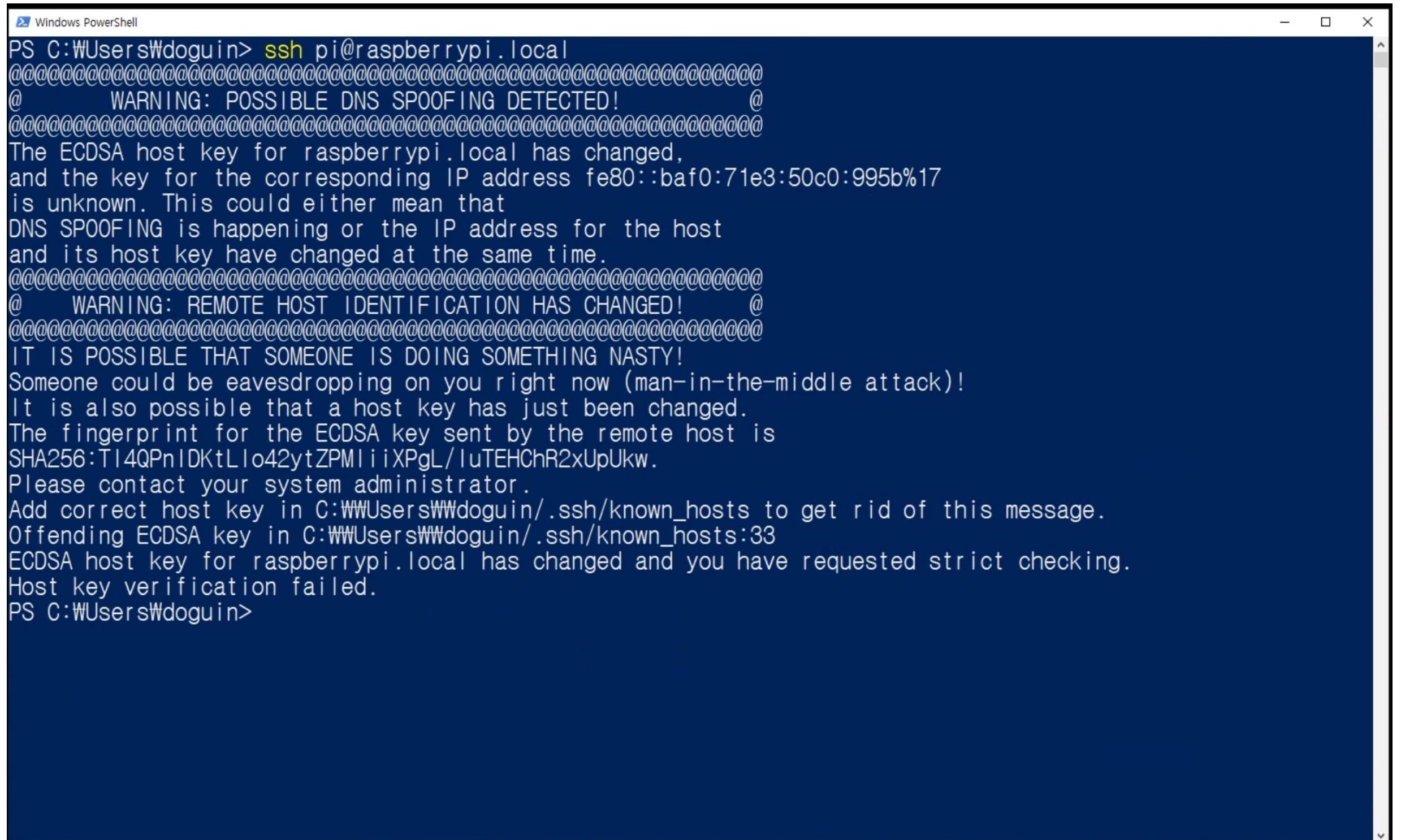
■ SSH 접속

- windows 의 powershell 혹은 다른 터미널 프로그램 사용.

```
C:\> ssh pi@192.168.0.101
```

- User : pi
- Password : raspberry

SSH서버 공개키 변경시 대처



```
PS C:\Users\doguin> ssh pi@raspberrypi.local
@@@@@@@@@@@ WARNING: POSSIBLE DNS SPOOFING DETECTED! @@@
The ECDSA host key for raspberrypi.local has changed,
and the key for the corresponding IP address fe80::baf0:71e3:50c0:995b%17
is unknown. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!)
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:Tl4QPnIDKtLlo42ytZPMliiXPgL/IuTEHChR2xUpUkw.
Please contact your system administrator.
Add correct host key in C:\Users\doguin/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in C:\Users\doguin/.ssh/known_hosts:33
ECDSA host key for raspberrypi.local has changed and you have requested strict checking.
Host key verification failed.
PS C:\Users\doguin>
```

- 동일한 hostname을 사용하는 새로운 디바이스에 연결하고자 하는 경우, 보안 우려가 있기 때문에 ssh 연결이 이루어지지 않는다. 이 경우 pc에 저장되어있는 예전 접속기록을 삭제해 새로운 디바이스에 연결할 수 있도록 한다.

```
C:\> ssh-keygen -R raspberry
```

or

```
C:\> ssh-keygen -R 192.168.0.101
```

■ Hostname 변경

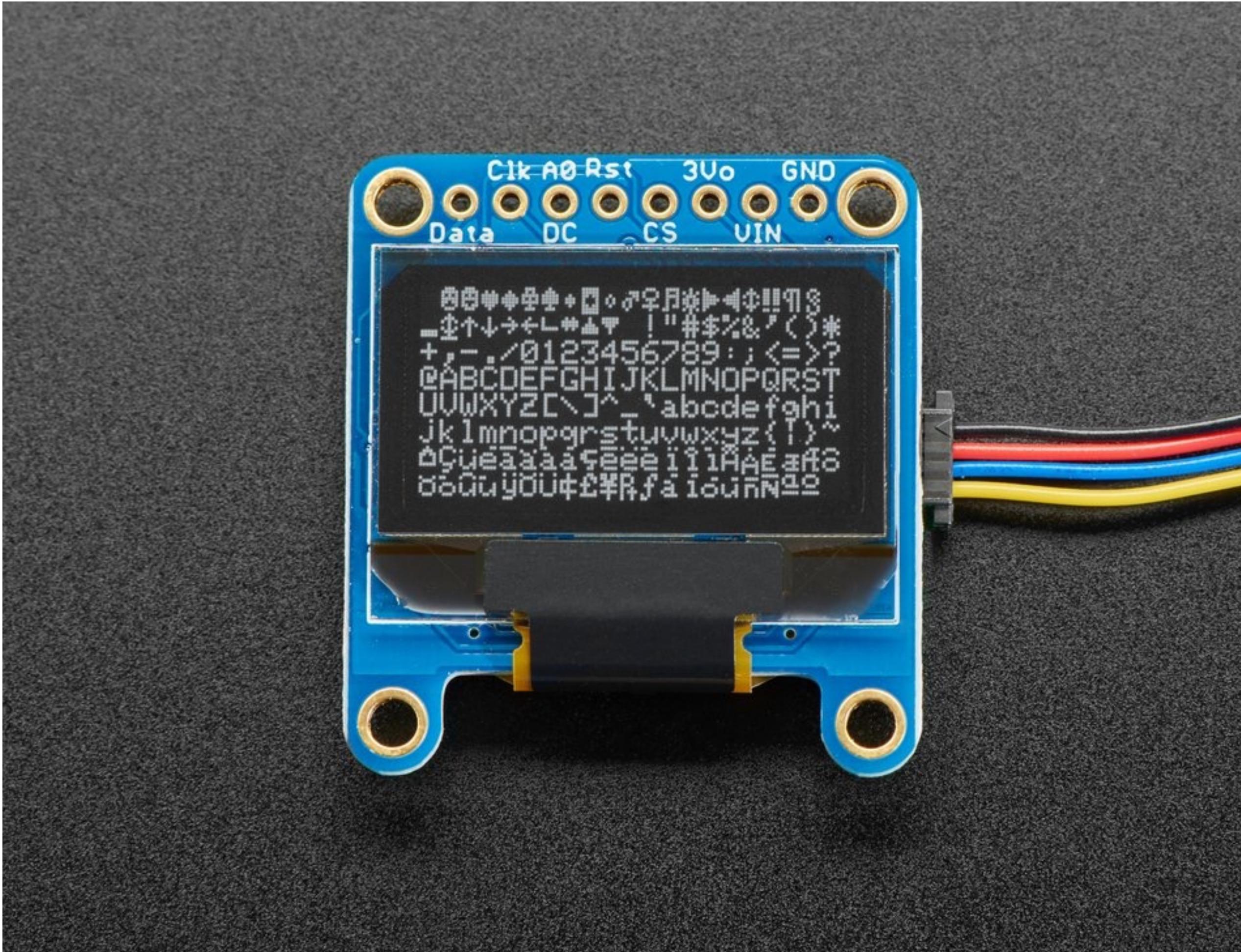
- 서로 혼동되지 않도록 주어진 2개의 raspberry pi를 각각 고유한 hostname으로 변경한다.
- raspi-config 명령을 사용한다.

```
$ sudo raspi-config
...
> 1.System Options > S4 Hostname > 원하는 hostname으로 교체
(예를 들어, raspberrypi -> doguinpi )
```

- 재부팅 후 다시 SSH 접속한다.

OLED display

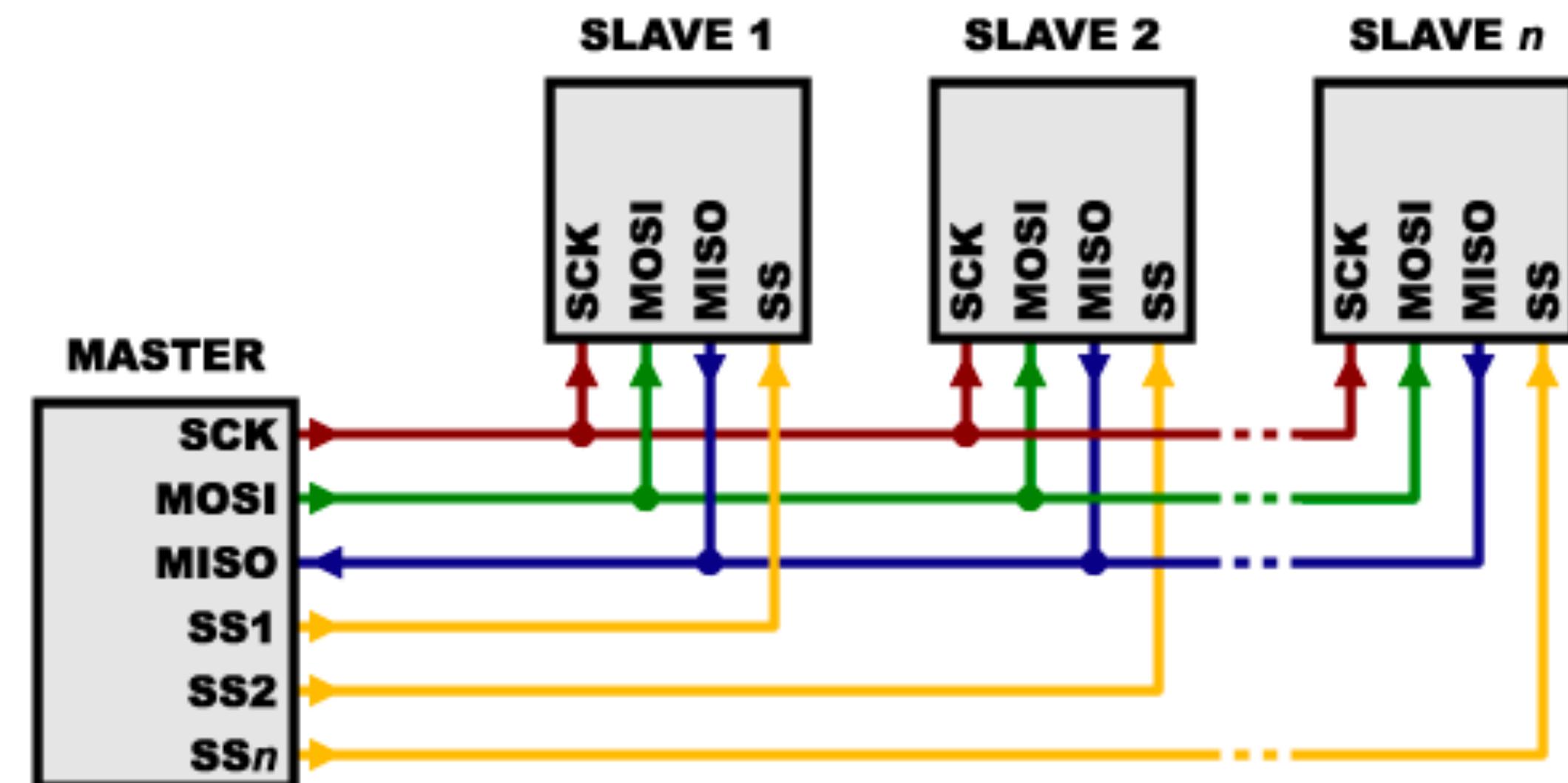
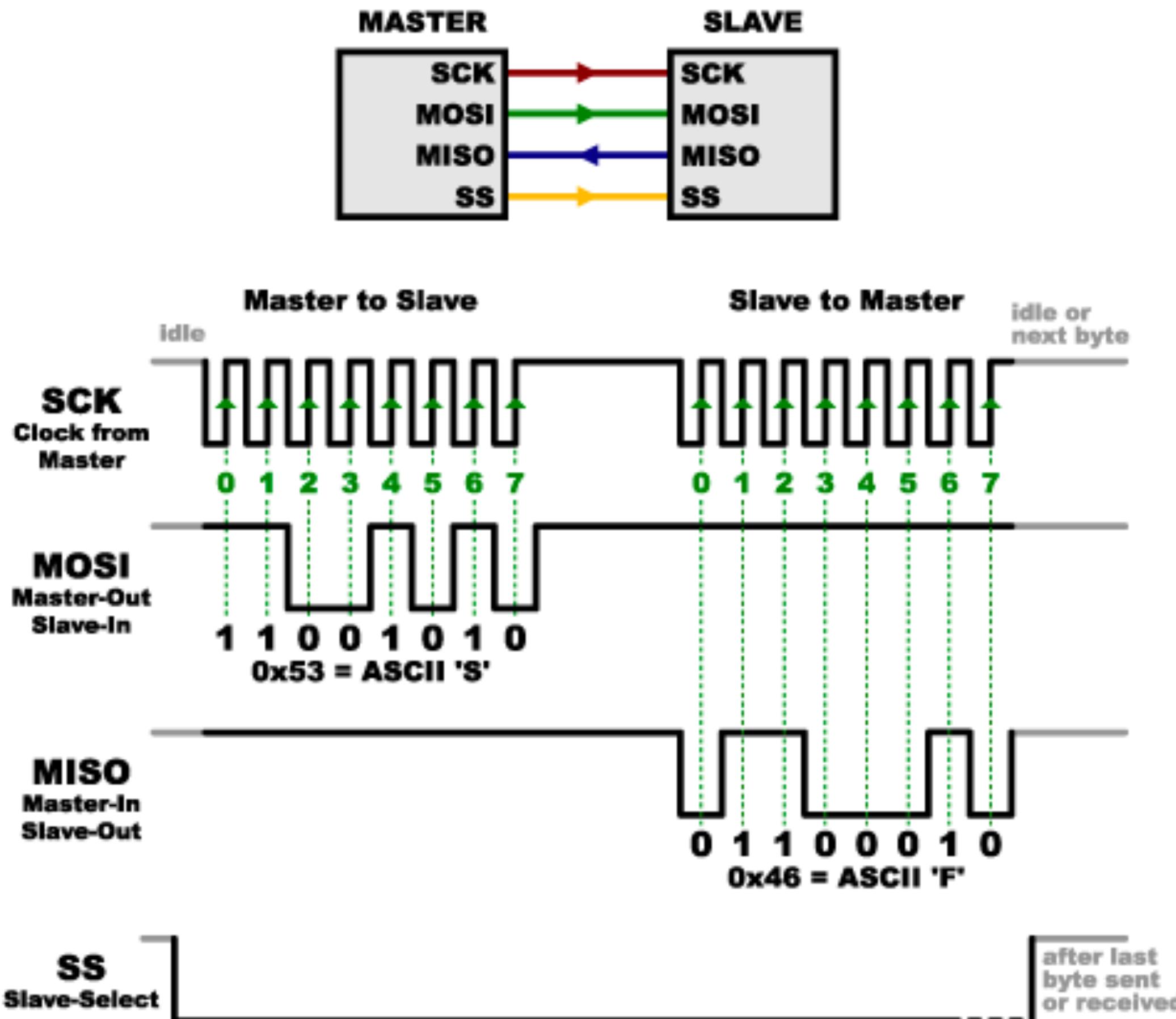
■ SSD1306 0.96" monochrome oled display



- SSD1306 controller
- 128 x 64 pixel
- b/w color
(greyscale 표현불가)
- 0.96"
- 3v-5v
- I2c (qwiic) / SPI interface

SSAFY Embedded Project

SPI interface

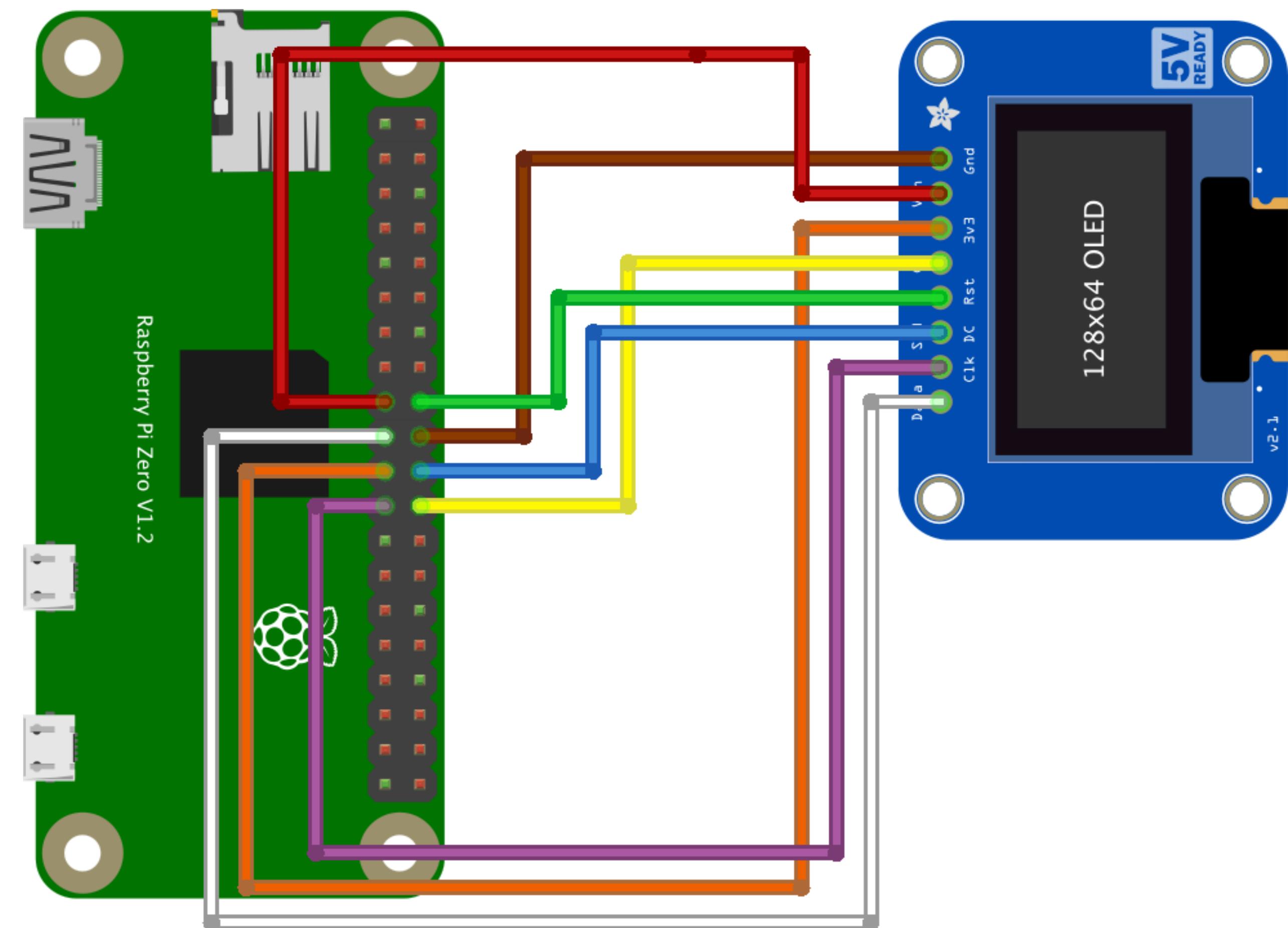
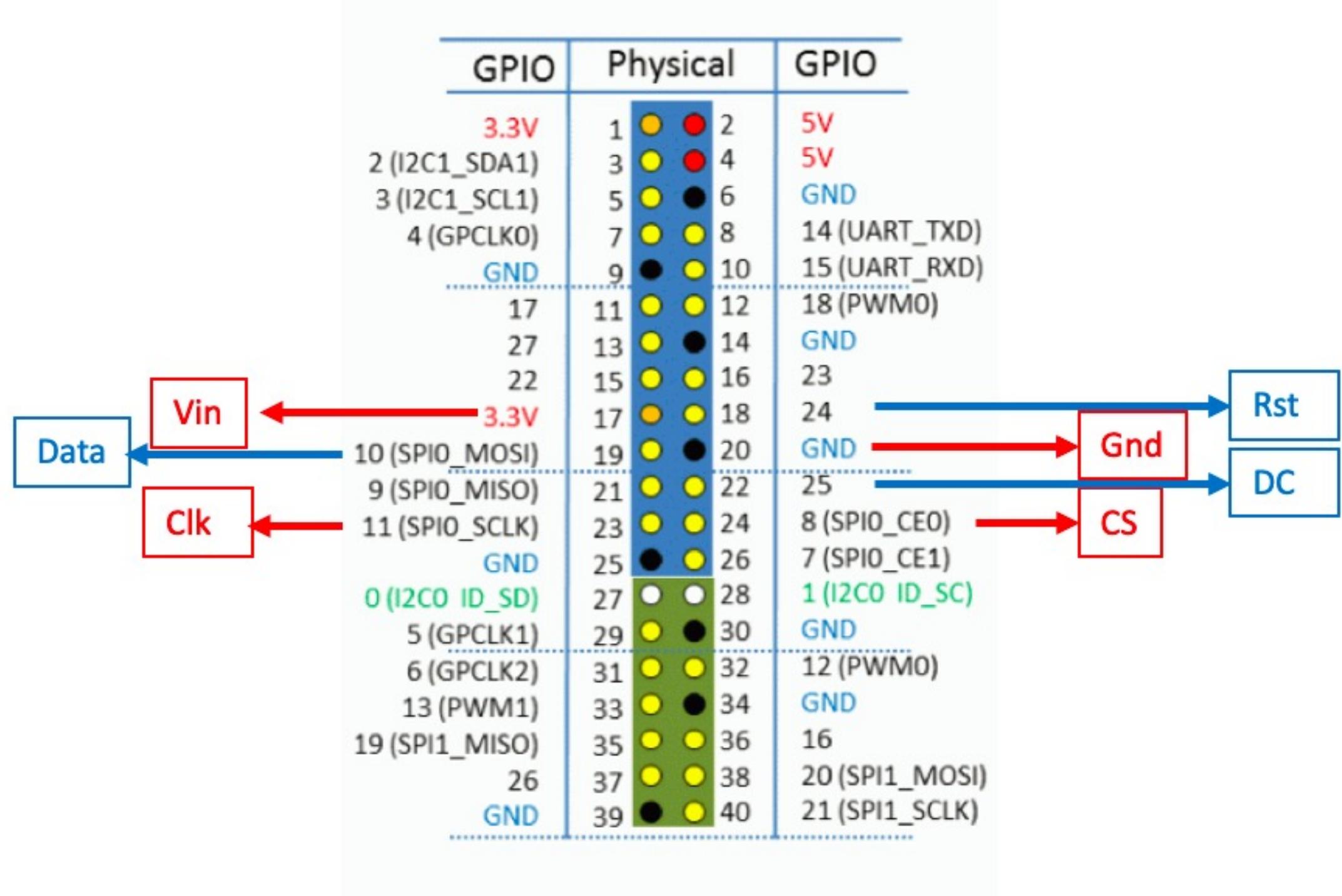


SPI interface 활성화

- raspi-config 명령을 사용해 raspberry pi에서 SPI를 활성화한다.

```
$ sudo raspi-config
...
> 3 Interface Options > I4 SPI > YES
```

raspberry pi – ssd1306 oled 연결



■ **ssd1306 python library**

- OLED 구동을 위한 파이썬 라이브러리를 설치한다.

```
$ pip install adafruit-circuitpython-ssd1306
```

- 제조사에서 제공하는 예제를 다운로드하여 기본 사용법을 알아본다.
(예제 참고: <https://learn.adafruit.com/monochrome-oled-breakouts/python-usage-2>)

```
$ wget -O oled_sample.py  
http://learn.adafruit.com/pages/17419/elements/3041225/downlo  
ad?type=py  
...  
$nano oled_sample.py
```

ssd1306 예제 코드

- Rst,CS,DC 핀의 연결 위치가 예제코드와 다를 수 있으므로 확인한다.
- 또한 연결방식(SPI/I2C)에 따라서도 코드가 달라지므로 확인한다.

```
GNU nano 5.4          oled_sample.py
=====

import board
import digitalio
from PIL import Image, ImageDraw, ImageFont
import adafruit_ssd1306

# Define the Reset Pin
oled_reset = digitalio.DigitalInOut(board.D24) reset pin 확인

# Change these
# to the right size for your display!
WIDTH = 128
HEIGHT = 64 # Change to 64 if needed 화면 해상도 확인
BORDER = 5

# Use for I2C.                                i2c 설정관련 주석처리
#i2c = board.I2C()
#oled = adafruit_ssd1306.SSD1306_I2C(WIDTH, HEIGHT, i2c, addr=0x3C, reset=oled_>

# Use for SPI                                 spi 설정관련 un-comment
spi = board.SPI()
oled_cs = digitalio.DigitalInOut(board.D8) CS pin 확인
oled_dc = digitalio.DigitalInOut(board.D25) DC pin 확인
oled = adafruit_ssd1306.SSD1306_SPI(WIDTH, HEIGHT, spi, oled_dc, oled_reset, oled_>

# Clear display.
oled.fill(0)
oled.show()

# Create blank image for drawing.
# Make sure to create image with mode '1' for 1-bit color.
image = Image.new("1", (oled.width, oled.height))
```

```
# Get drawing object to draw on image.
draw = ImageDraw.Draw(image)

# Draw a white background
draw.rectangle((0, 0, oled.width, oled.height), outline=255, fill=255)

# Draw a smaller inner rectangle
draw.rectangle(
    (BORDER, BORDER, oled.width - BORDER - 1, oled.height - BORDER - 1),
    outline=0,
    fill=0,
)

# Load default font.
font = ImageFont.load_default()

# Draw Some Text
text = "Hello World!"
(font_width, font_height) = font.getsize(text)
draw.text(
    (oled.width // 2 - font_width // 2, oled.height // 2 - font_height // 2),
    text,
    font=font,
    fill=255,
)

# Display image
oled.image(image)
oled.show()
```

Pillow (Python Image Library)

OLED boilerplate

```
import board
import digitalio
import adafruit_ssd1306

# Define the Reset Pin
oled_reset = digitalio.DigitalInOut(board.D24)

# Change these to the right size for your display!
WIDTH = 128
HEIGHT = 64

# Use for SPI
spi = board.SPI()
oled_cs = digitalio.DigitalInOut(board.D8)
oled_dc = digitalio.DigitalInOut(board.D25)
oled = adafruit_ssd1306.SSD1306_SPI(WIDTH, HEIGHT,
spi, oled_dc, oled_reset, oled_cs)
```

```
# Clear display.
oled.fill(0)
oled.show()

# Create PIL image
image =

# Display image
oled.image(image)
oled.show()
```

Pillow

- 비트맵 이미지 표시
 - (테스트용 이미지 다운로드 : \$ wget -O catImage.ppm <https://bit.ly/3d03XI5>)

```
from PIL import Image

image = Image.open('catImage.ppm').convert('1')
```

Pillow

- 이미지 자르기 , 회전하기 , 붙이기

```
from PIL import Image

image = Image.open('catImage.ppm').convert('1')

box = (0,0,50,50)
region = image.crop(box)

region = region.transpose(Image.ROTATE_180)
image.paste(region,box)
```

※ Image.FLIP_LEFT_RIGHT / Image.FLIP_TOP_BOTTOM /
Image.ROTATE_90 / Image.ROTATE_180 / Image.ROTATE_270 /
Image.TRANSPOSE / Image.TRANSVERSE

Pillow

- 파일로 저장하기

```
image.save('myImage.jpg')
```

- 빈 이미지 만들기

```
from PIL import Image

image = Image.new( '1', (128,64),0) # mode, size, background_color
```

Pillow

- 도형 그리기

```
from PIL import Image, ImageDraw

image = Image.open('catImage.ppm').convert('1')

draw = ImageDraw.Draw(image)

draw.line((0,0,128,64), fill=1)
draw.line((0, image.size[1], image.size[0], 0), fill=1)
```

※ draw.rectangle((x1, y1, x2, y2), fill = 1)
draw.ellipse((x1, y1, x2, y2), fill = 1)

Pillow

- text 표시하기
(맑은고딕 폰트 다운로드: \$ wget -O malgun.ttf <https://bit.ly/3BzjZfq>)

```
from PIL import Image, ImageDraw, ImageFont

image = Image.new('1', (128, 64), 0)
draw = ImageDraw.Draw(image)
font = ImageFont.truetype('malgun.ttf', 15) # 폰트파일, 폰트크기

draw.text((10,10), '안녕하세요^^ 반가워요!', font=font, fill=1)
```

- Pillow 공식 문서
<https://pillow.readthedocs.io/>

Date & Time

Raspberry pi Time setting

- raspi-config 명령 사용

```
$ sudo raspi-config  
...  
raspi-config 프로그램 실행 > 5. Localisation Options >  
L2. Timezone > Asia, Seoul을 차례로 선택. > Finish
```

- 현재시간 설정

```
$ sudo date -s '2022-10-21 13:04:00'
```

```
$ sudo date -s '13:04:00'
```

Datetime module

Datetime module

- 특정 날짜 표시

```
from datetime import date

my_birthday = date(1978, 8, 6)
today = date.today()

print(f'나는 {my_birthday.year}년에 태어났습니다.')
print(f'나는 {my_birthday.month}월 {my_birthday.day}일에 태났어요.')

day = '월화수목금토일'[today.weekday()]
print(f'오늘은 {day}요일이에요.')
```

Datetime module

- 특정 시간 표시

```
from datetime import time  
  
at = time(10, 30, 0)  
  
print(at)  
print(f'기차는 {at.hour}시 {at.minute}분에 떠난다.')
```

- 날짜+시간

```
from datetime import datetime  
  
time = datetime(2022, 10, 21, 13, 30)  
print(time)
```

Datetime module

- 현재 시간 표시

```
from datetime import datetime

now = datetime.now()
print(now)
print(f'지금은 {now.month}월 {now.day}일 {now.hour}시 {now.minute}분
입니다.')
```

- 포맷 지정 시간 표시 (strftime)

```
print(now.strftime('%Y %h.%d %p%I:%M'))
```

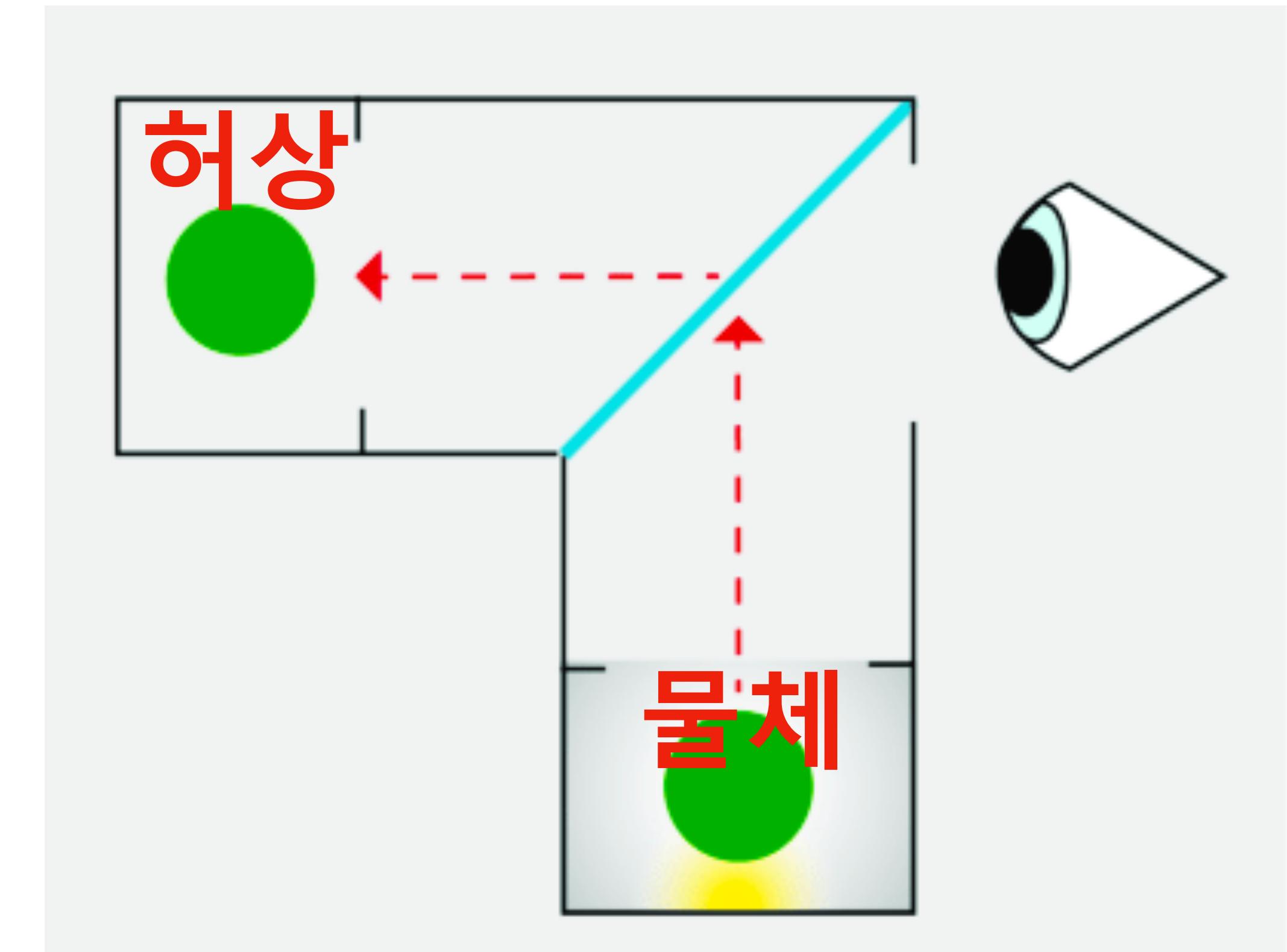
※ %Y : 4자리 년도 %d : 날 %I : 시
 %h : 달 %p : am/pm %M : 분

AR glass 조립

Hologram



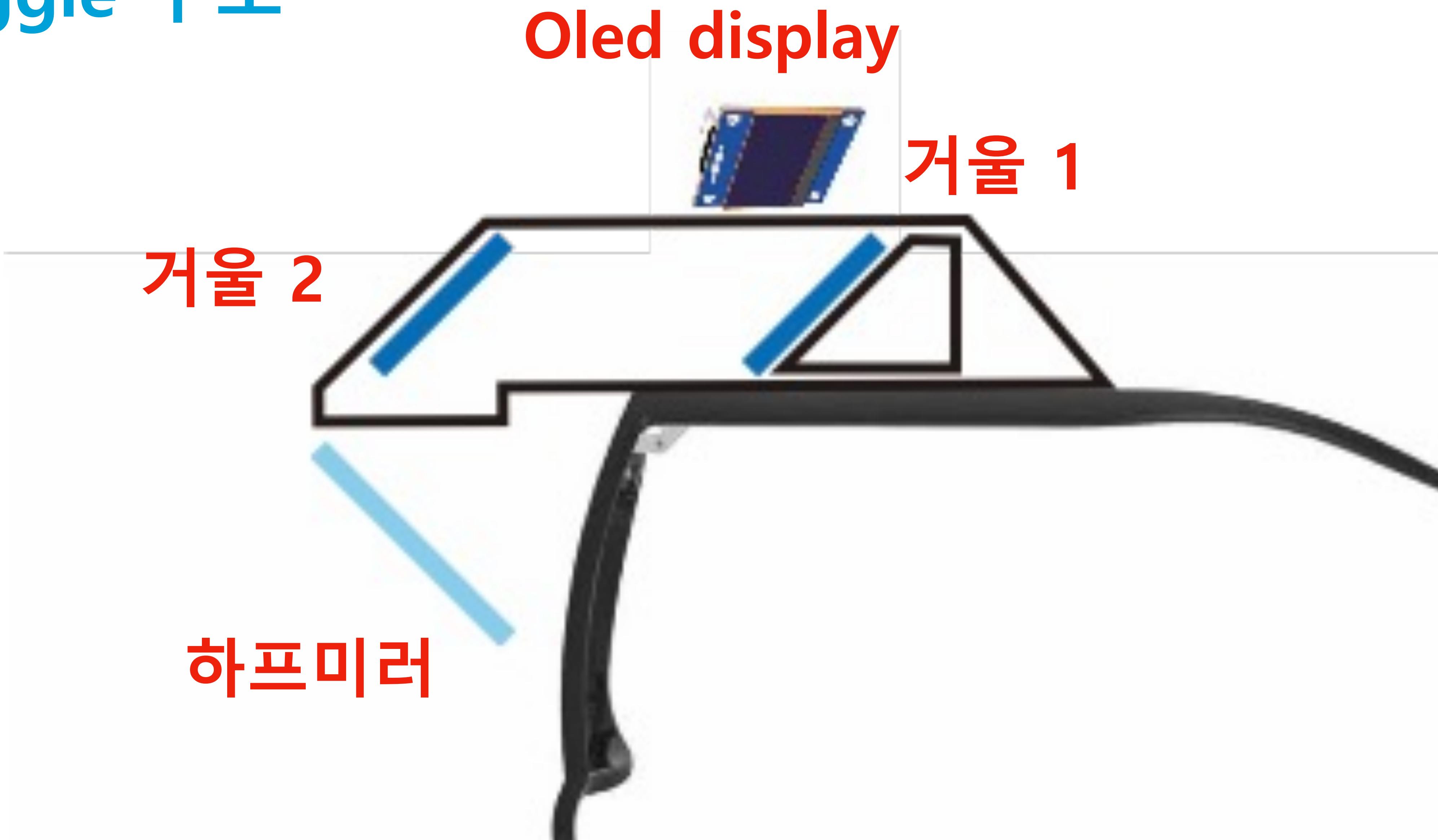
■ Pepper's ghost



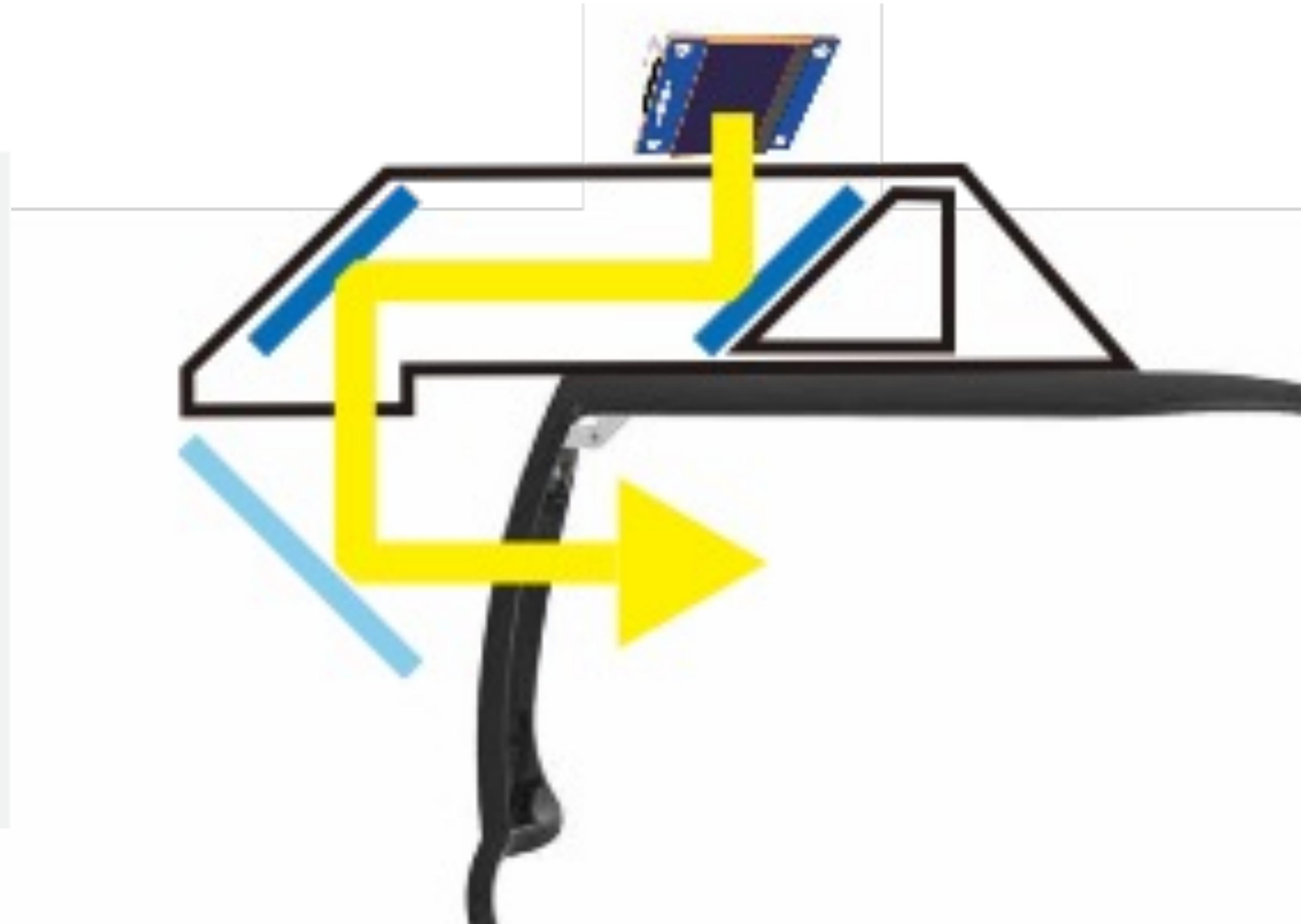
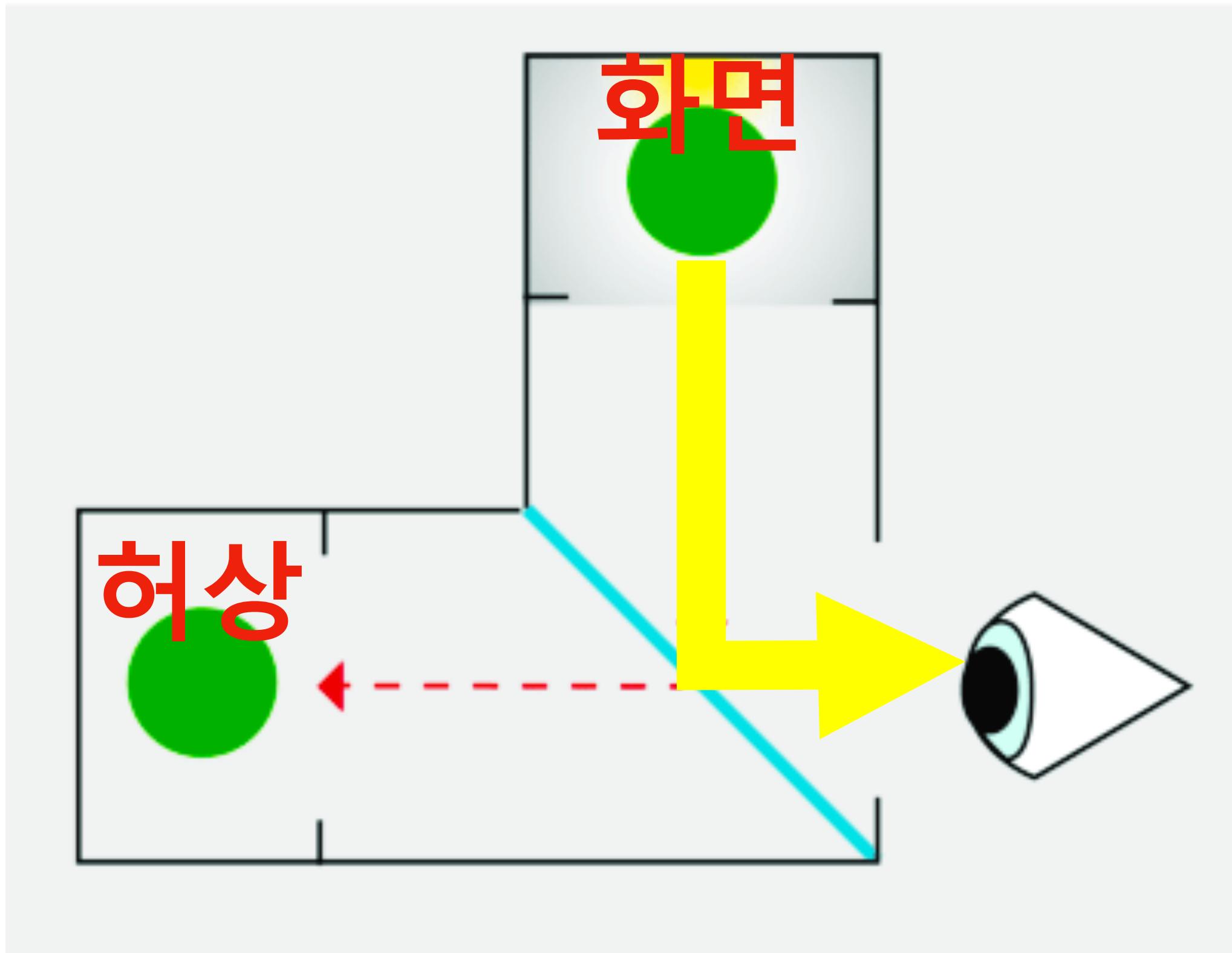
■ AR Goggle 구조



■ AR Goggle 구조



■ AR Goggle 구조



Lens Optics

현재시각:
9:35
서두르세요!
약속시간까지 25분 남았습니다.

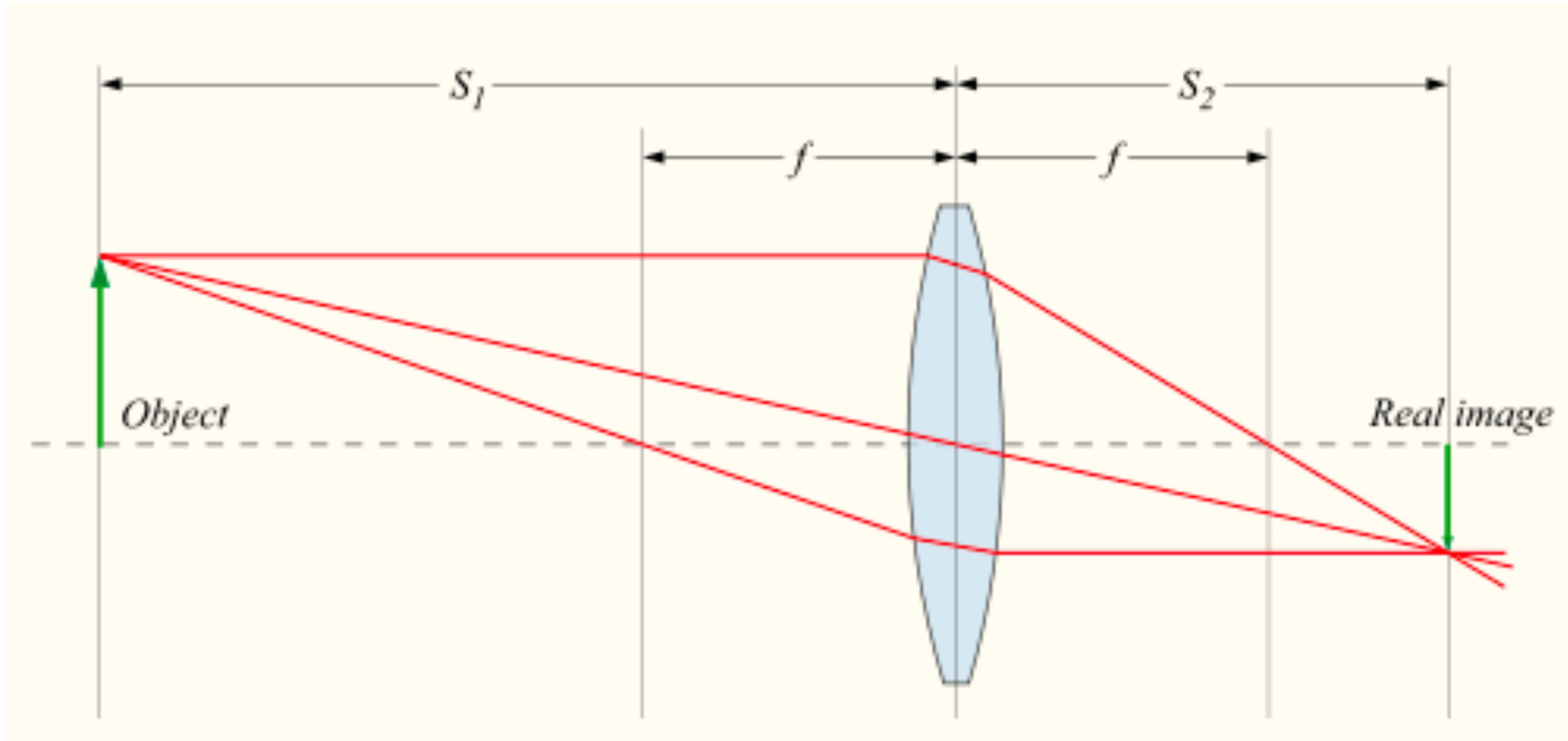


SSAFY Embedded Project

Lens Optics

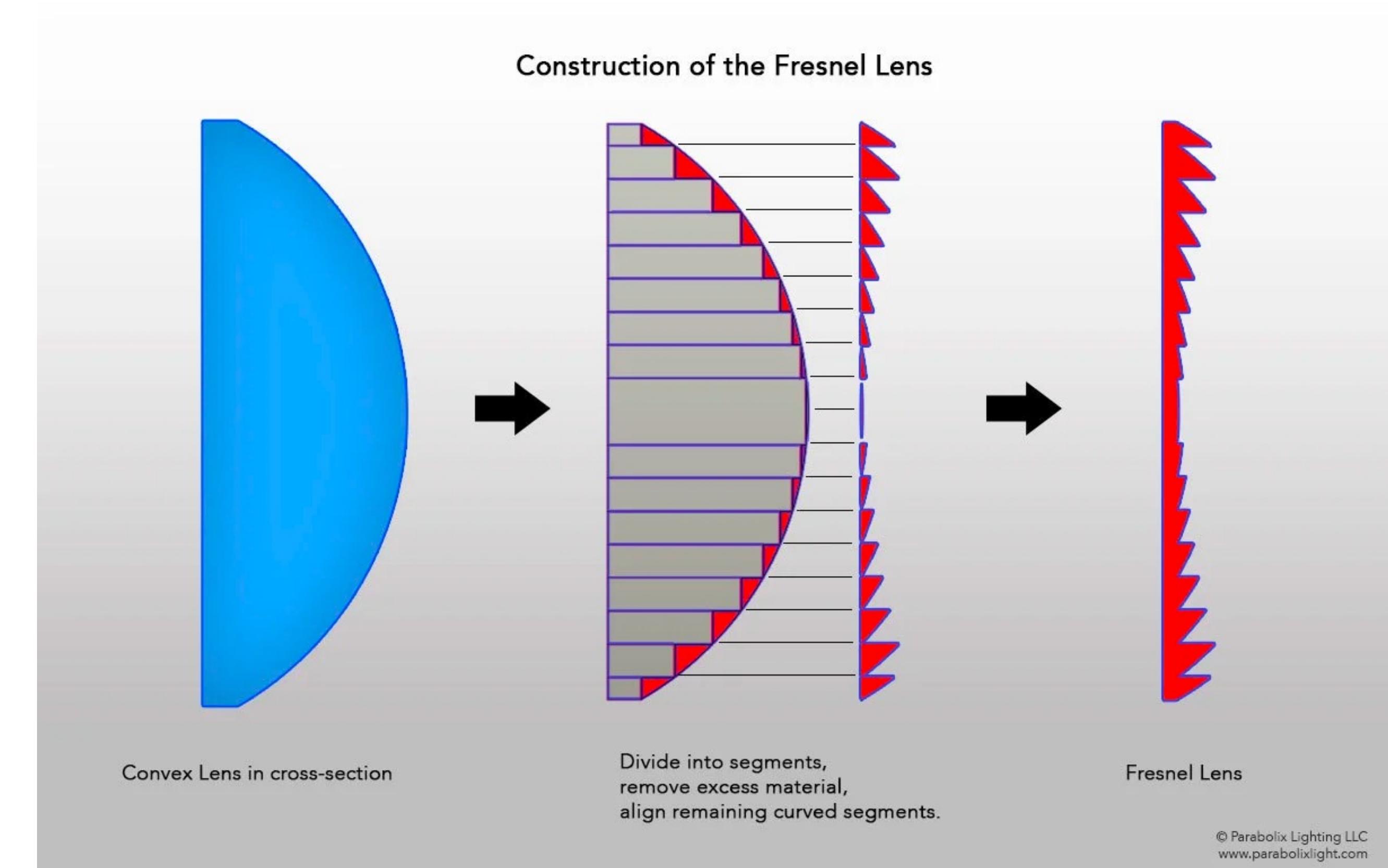


Lens optics

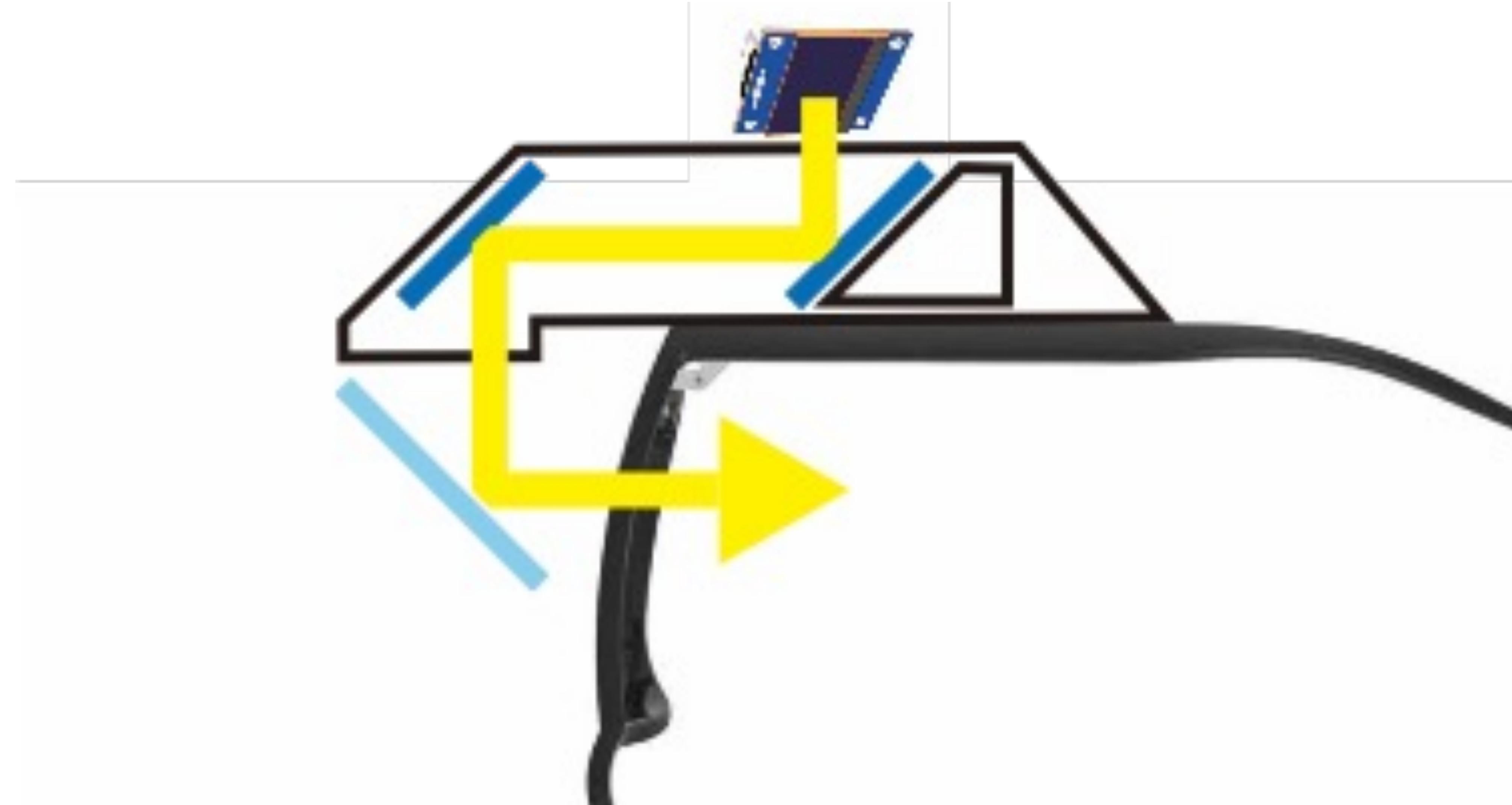


Lens optics

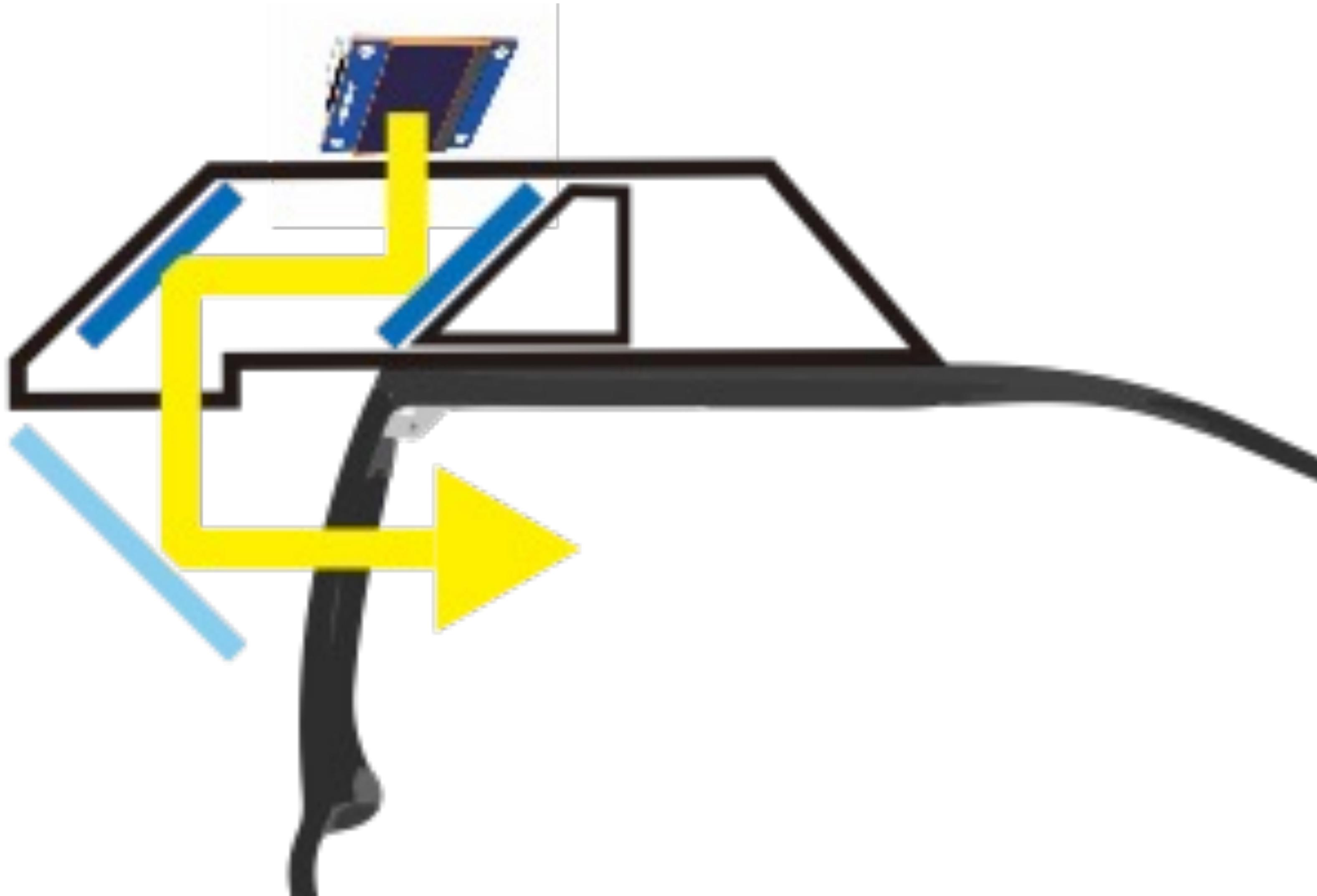
Fresnel lens



■ AR Goggle 구조



■ AR Goggle 구조



GPIO, Button

RPi.GPIO boilerplate

- **GPIO.setmode(mode)**
핀번호를 매기는 방식

mode	
GPIO.BCM	라즈베리파이에 배열된 순서대로
GPIO.BOARD	SOC 제조사가 이름붙인 순서대로

- **GPIO.setup(pin, io)**
특정 핀을 입/출력용으로 설정

io	
GPIO.IN	해당 핀을 입력용으로 설정
GPIO.OUT	해당 핀을 출력용으로 설정

- **GPIO.cleanup()**
GPIO제어권 반납

```
# 라이브러리 임포트
import RPi.GPIO as GPIO
...
# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(12, GPIO.IN)
GPIO.setup(20, GPIO.OUT)

...
# 메인루프
try:
    while 1:
        button = GPIO.input(12)
        ...
        GPIO.output(20, GPIO.HIGH)
        ...
# 반드시 클린업
finally:
    GPIO.cleanup()
```

Digital Output

- `GPIO.output(pin, onoff)`

onoff	
<code>GPIO.HIGH</code>	5v 출력 (on)
<code>GPIO.LOW</code>	0v 출력 (off)

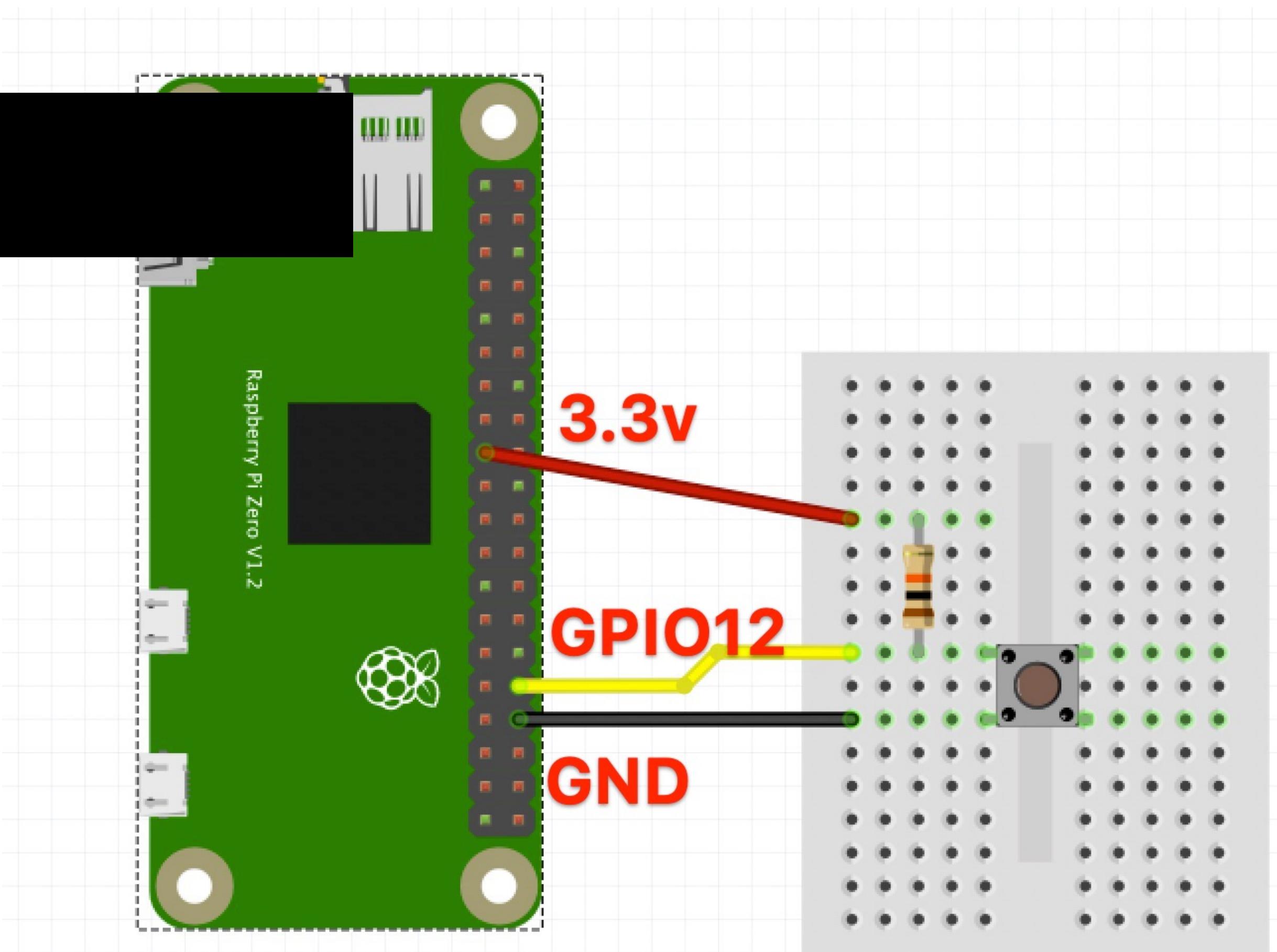
```
import time

GPIO.output(20, GPIO.HIGH)
time.sleep(0.1) # 100 millisecond 딜레이
GPIO.output(20, GPIO.LOW)
```

Digital Input

- `GPIO.input(pin)`

```
pin_read = GPIO.input(12)
```

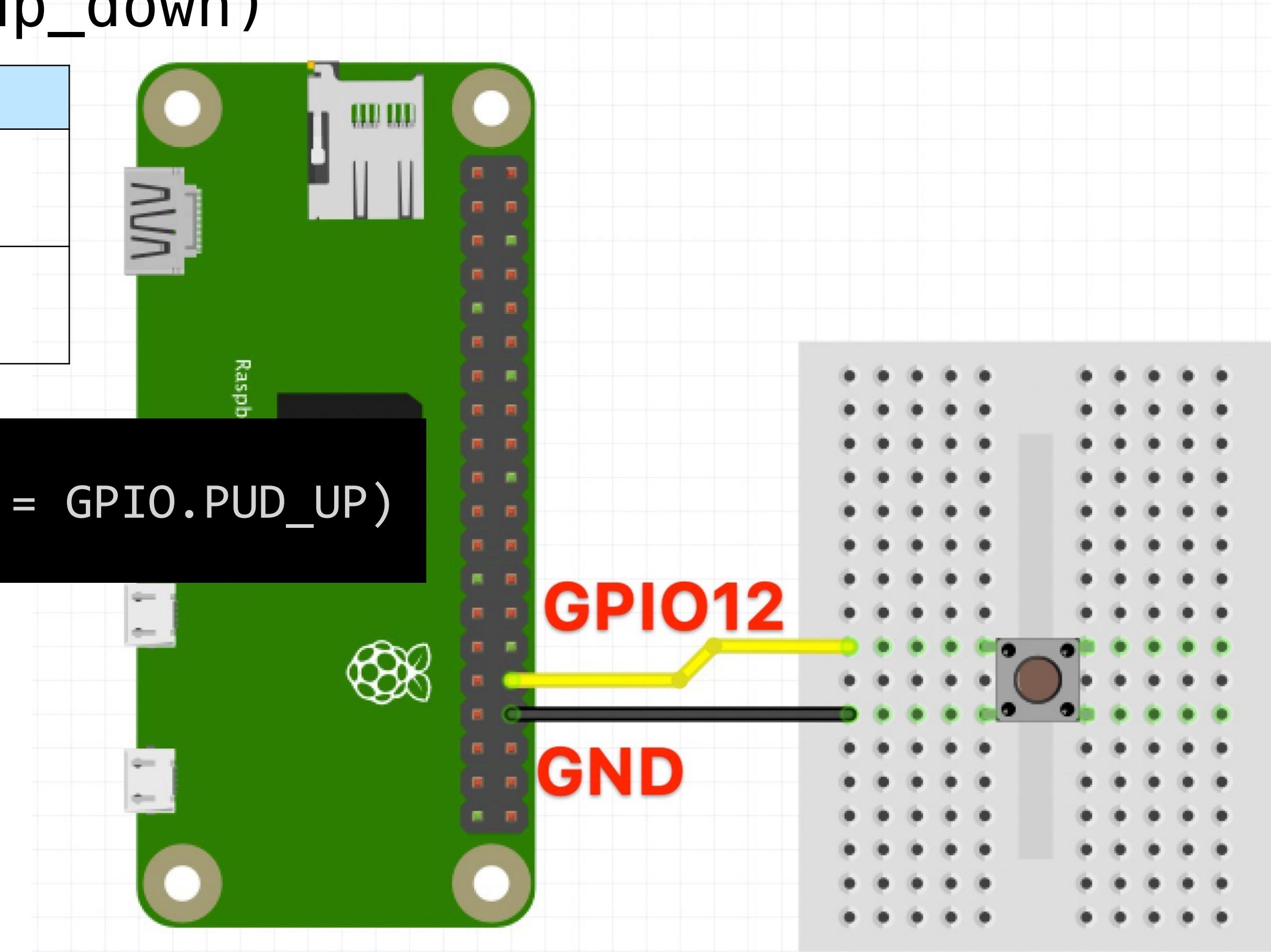


Digital Input (내장 pull-up / pull-down)

- GPIO.setup(pin, mode, pull_up_down)

pull_up_down	
GPIO.PUD_UP	내장 pull-up 활성화 (default HIGH)
GPIO.PUD_DOWN	내장 pul-down 활성화 (default LOW)

```
GPIO.setup(12, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```



GPIO interrupt

- `GPIO.add_event_detect(pin, edge, callback=myfunc)`

특정 핀에 신호가 변경되는 순간 callback을 실행한다.

edge	
<code>GPIO.RISING</code>	<code>LOW -> HIGH</code> 순간
<code>GPIO.FALLING</code>	<code>HIGH -> LOW</code> 순간
<code>GPIO.BOTH</code>	<code>LOW <-> HIGH</code> 순간

```
import RPi.GPIO as GPIO
import time

# 스위치 눌렸을 때 콜백함수
def switchPressed(channel):
    print('channel %s pressed!!'%channel)

# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
# interrupt 선언
GPIO.add_event_detect(12, GPIO.RISING,
callback=switchPressed)
# 메인 루프
try:
    while 1:
        print(".")
        time.sleep(0.1)
finally:
    GPIO.cleanup()
```

추가과제 (Optional)

■ 추가과제

- 고글 디스플레이를 활용하여 나만의 독창적인 화면을 만들어본다.
- 시침과 분침으로 표시되는 아날로그 시계로 표현해본다.
- 필요에 따라 시간 측정을 위한 timer, alarm 기능을 구현해본다.

감사합니다

Thank You