MISSION IMPOSSIBLE

윤희○
곽민○
김진○

1

스위치 ON!

일어나서 불을 켜주세요!

2

물 마시기!

냉장고를 열어서 물을 한 잔!

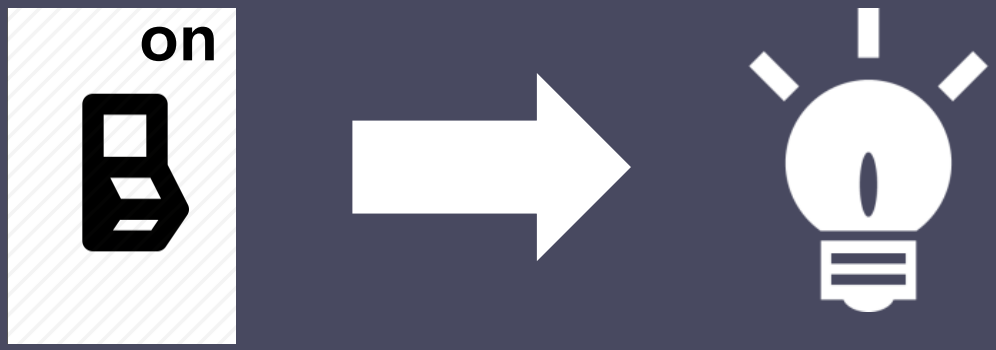
3

화장실 GO!

씻으러 갑시다!

MISSION 1

스위치 ON!!
In bedroom



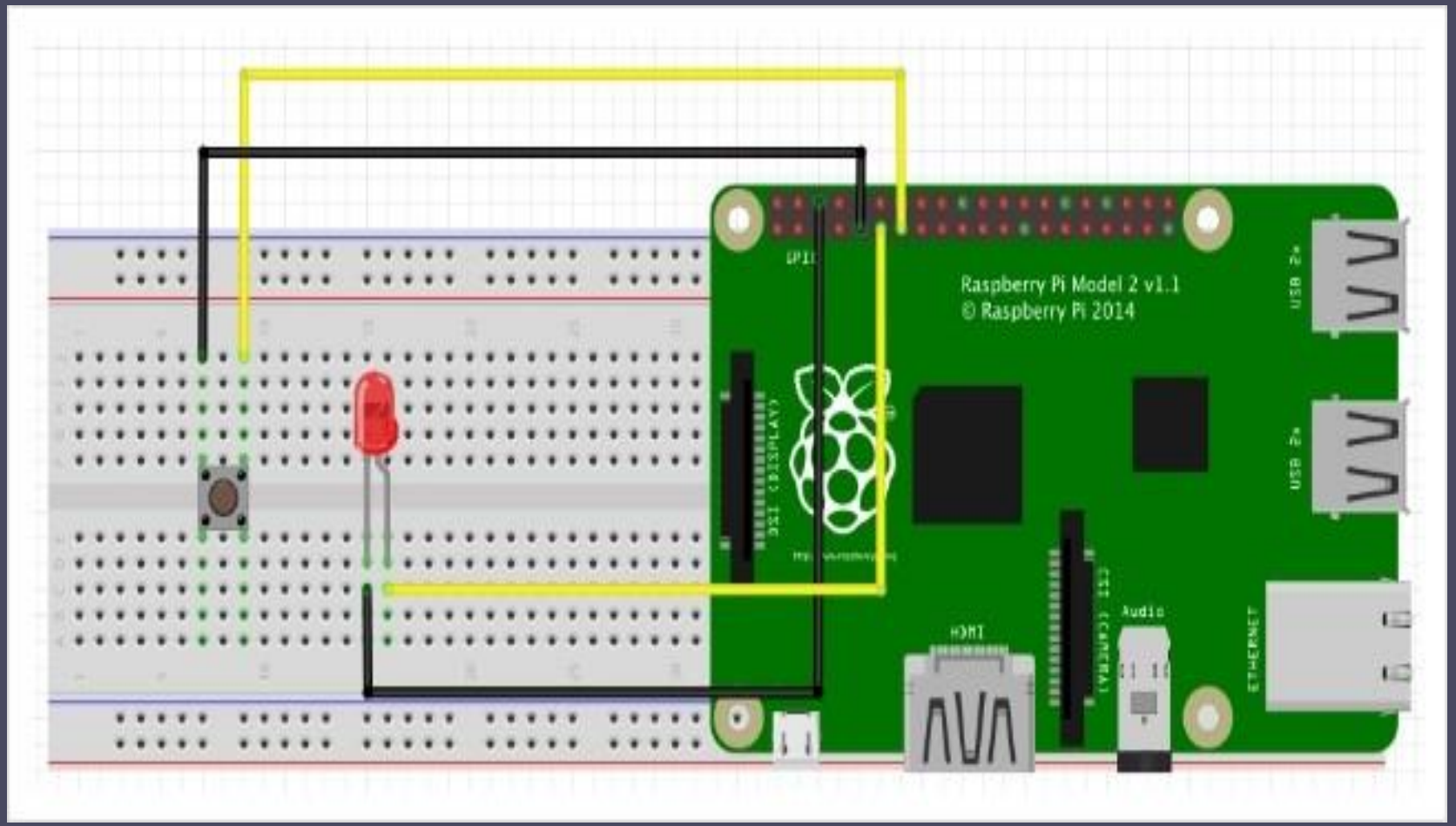
- 스위치 on → Mission Clear → 2단계 이동
- 푸쉬버튼 & LED 사용

스위치로 LED를 동작시키는 코드

```
import Rpi.GPIO as GPIO
from time import sleep

def main():
    led = 22
    switch = 27
    state = 1

    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnigns(False)
    GPIO.setup(led, GPIO.out)
    GPIO.setup(switch, GPIO.IN, GPIO.PUD_UP)
```

```
try:
    while True:
        if GPIO.input(switch) == 0:
            while True:
                if GPIO.input(switch) == 1:
                    state = not state
                    GPIO.output(led, state)
                    sleep(0.2)
                    break

finally:
    print 'end'
```

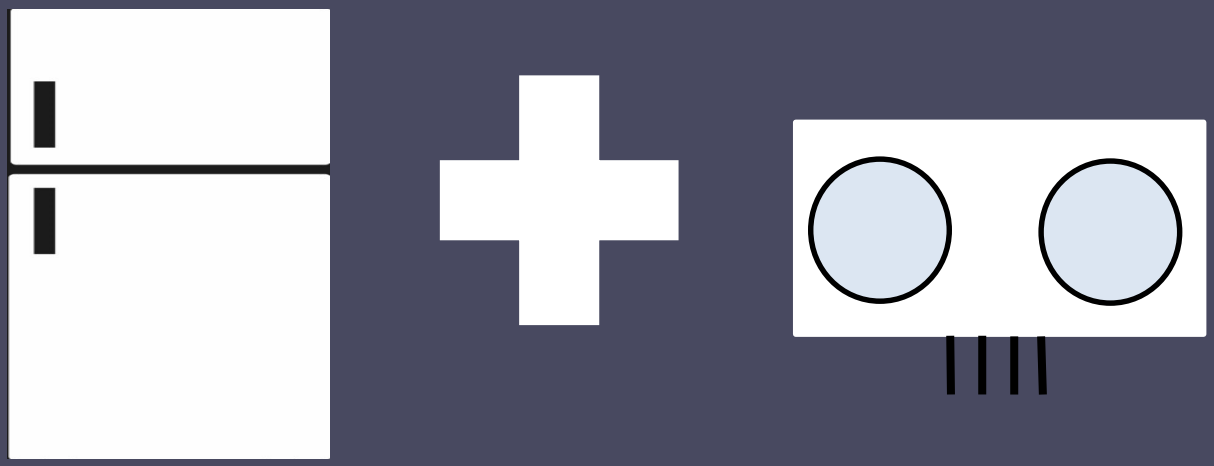
스케치


Raspberry Pi Model 2 v1.1
© Raspberry Pi 2014
USB 2x
USB 2x
Audio
ETHERNET

MISSION 2

냉장고를 열어라!

In kitchen



- 냉장고에 가까이 → 거리 변화 측정 → Mission Clear → 3단계 이동
- 초음파 거리 측정 센서 모듈을 냉장고에 부착

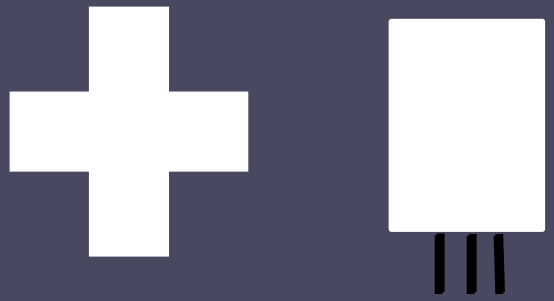
거리변화를 인식하는 코드

```
import RPi.GPIO as GPIO
import time

#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BOARD)

#set GPIO Pins
GPIO_TRIGGER = 18
GPIO_ECHO = 16

#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER,GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    start_time = time.time()
    end_time = time.time()

    # save start_time
    while GPIO.input(GPIO_ECHO) == 0:
        start_time = time.time()

    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        end_time = time.time()

    # time difference between start and arrival
    pulse_duration = end_time – start_time

    # multiply with the sonic speed (34000 cm/s)
    # and divide by 2, because there and back
    # distance = pulse_duration * 340 * 100 / 2 distance =
    pulse_duration * 17000

    return distance

if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Measured Distance = %.1f cm" % dist)
            time.sleep(1)

        # Reset by pressing CTRL + C

    except KeyboardInterrupt:
        print("Measurement stopped by User")
        GPIO.cleanup()
```

MISSION 3

화장실에 체온을!

In toilet



- 센서 접촉 → 온도 증가 → Mission Complete!!
- 온도 감지 센서를 욕실에 부착

RPi 라이브러리를 이용한 코드

```
import RPi.GPIO as GPIO
import time

def bin2dec(string_num):
    return str(int(string_num, 2))

data = []

pin_num = 18

GPIO.setmode(GPIO.BCM)

GPIO.setup(pin_num,GPIO.OUT)
GPIO.output(pin_num,GPIO.HIGH)
time.sleep(0.025)
GPIO.output(pin_num,GPIO.LOW)
time.sleep(0.02)

GPIO.setup(pin_num, GPIO.IN,
pull_up_down=GPIO.PUD_UP)

for i in range(0,500):
    input_d = GPIO.input(pin_num)
    data.append(input_d)
```

```
bit_count = 0
tmp = 0
count = 0
HumidityBit =
TemperatureBit =
crc =

try:
    while data[count] == 1
        tmp = 1
        count = count + 1

    for i in range(0, 32)
        bit_count = 0

        while data[count] == 0
            tmp = 1
            count = count + 1

        while data[count] == 1
            bit_count = bit_count + 1
            count = count + 1
```

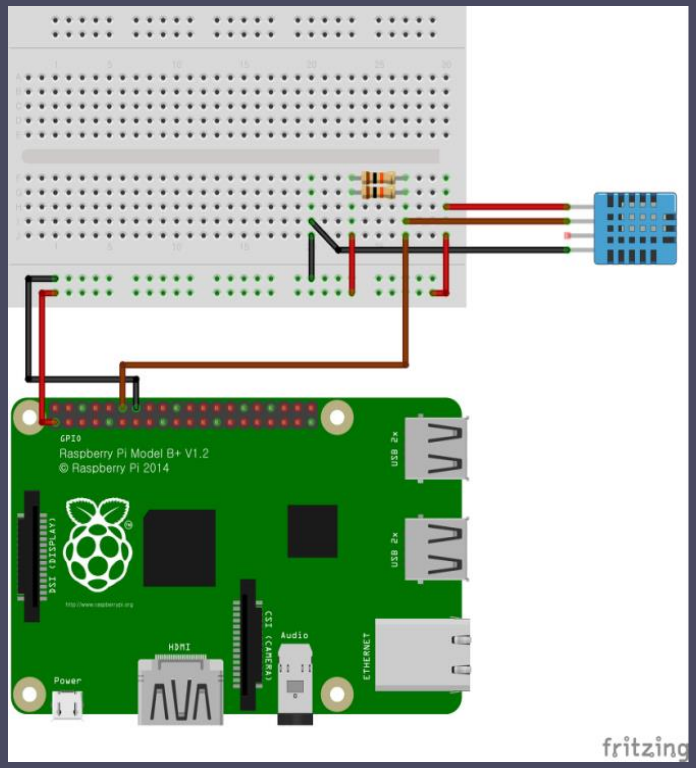
```
   if bit_count  3:
      if i=0 and i8:
         HumidityBit = HumidityBit + 1
      if i=16 and i24:
         TemperatureBit = TemperatureBit + 1
      else:
         if i=0 and i8:
            HumidityBit = HumidityBit + 0
         if i=16 and i24:
            TemperatureBit = TemperatureBit + 0

except:
   print ERR_RANGE1
   exit(0)

try:
    for i in range(0, 8):
        bit_count = 0

        while data[count] == 0
            tmp = 1
            count = count + 1

        while data[count] == 1
        bit_count = bit_count + 1
        count = count + 1

       if bit_count  3:
          crc = crc + 1
       else:
          crc = crc + 0
```

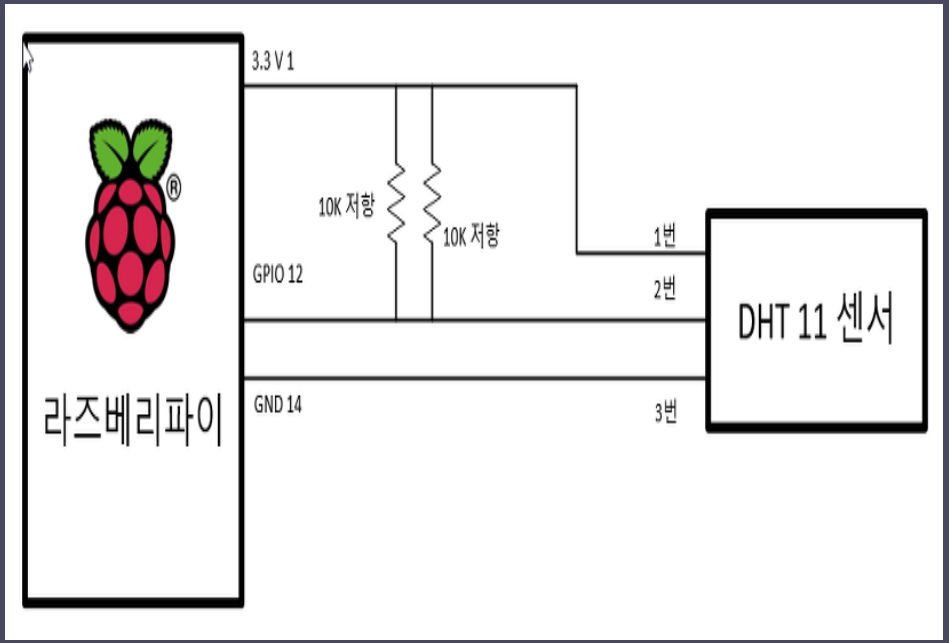
```
except
    print ERR_RANGE2
    exit(0)

Humidity = bin2dec(HumidityBit)
Temperature = bin2dec(TemperatureBit)

if int(Humidity) + int(Temperature) - int(bin2dec(crc))
== 0
    print Humidity
    print Temperature
else:
    print ERR_CRC
```

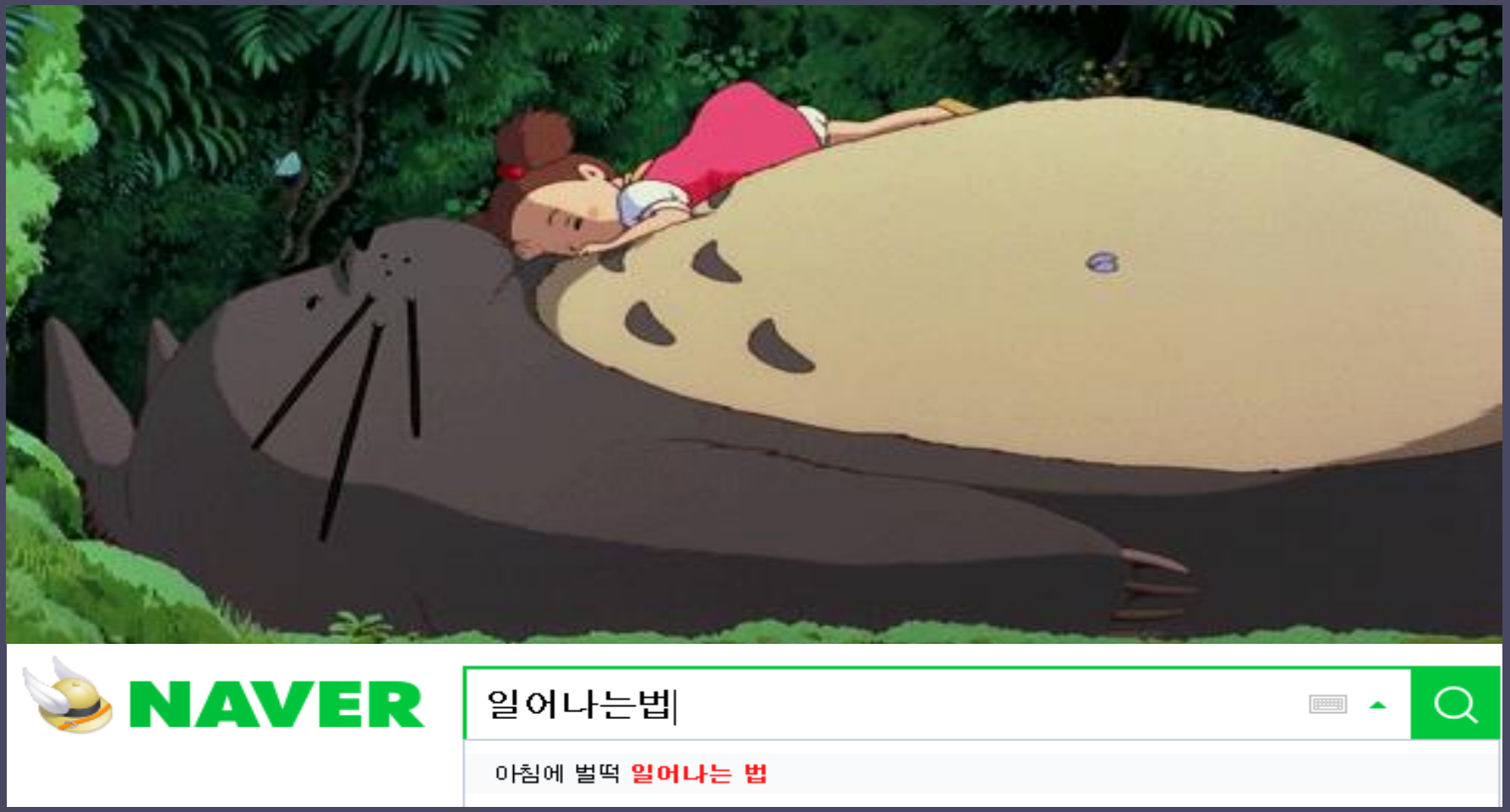
adafruit의 라이브러리를 이용한 코드

```
import Adafruit_DHT as dht
h,t = dht.read_retry(dht.DHT11,18)

print h
print t
```

스케치


GPIO
Raspberry Pi Model B+ V1.2
© Raspberry Pi 2014
USB 2x
USB 2x
DSI (DISPLAY)
CSI (CAMERA)
HDMI
Audio
ETHERNET
Power
fritzing


3.3 V 1
10K 저항
10K 저항
GPIO 12
GND 14
1번
2번
3번
DHT 11 센서
라즈베리파이



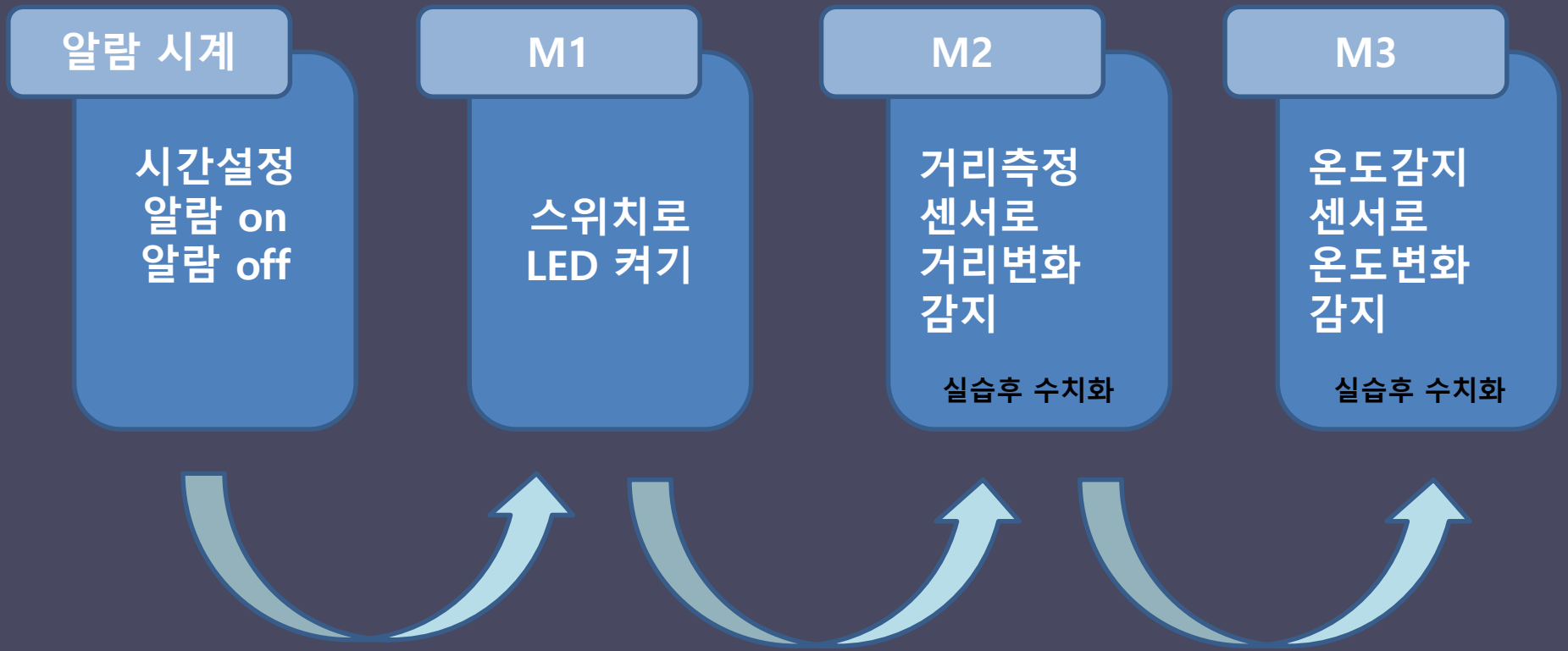
아침에 알람을 끄고 다시 잠들어버리는 상황을 방지하기 위해 미션을 클리어해야만 알람을 끌 수 있도록 설정하여 제 시간에 일어날 수 있도록 도와줍니다

작품 특징

- 소비자 : World People
- 기존 알람에도 쉽게 못 일어 나는 사람들을 위해 늦잠을 방지할 수 있는 제품
- 재미 & 신선함
- Mission을 마음대로 설정가능
- 스마트 홈 시장의 확대에 따른 제품 성행 예상

개발 과정

지금까지 실습해 왔던 각 센서마다의 코딩들을 응용하여 복합적인 코딩을 할 예정입니다.



Thank You