SAMSUNG SW ACADEMY FOR YOUTH

삼성 청년 SW 아카데미

Embedded Project

day 2

SAMSUNG SW ACADEMY FOR YOUTH

DEMO – day2. RC Vehicle

금일 교육 내용

1. 차량 조립
2. Motor / Servo 구동
3. MEMS microphone

RC Vehicle 기능명세

- 주어진 킷트를 조립하여 rc-car를 제작한다.
- Raspberry pi 4 model B 를 컨트롤러로 사용하여 ssh로 연결된 호스트컴퓨터(pc)에서 콘솔에 명령어를 입력함으로써 차량을 제어할 수 있다.
- 구현해야할 기본 명령은 '앞으로/ 뒤로/ 정지/ 빠르게/ 느리게/ 오른쪽/ 왼쪽/ 중앙' 이다.
- wifi통해 네트워크 연결되도록 한다.

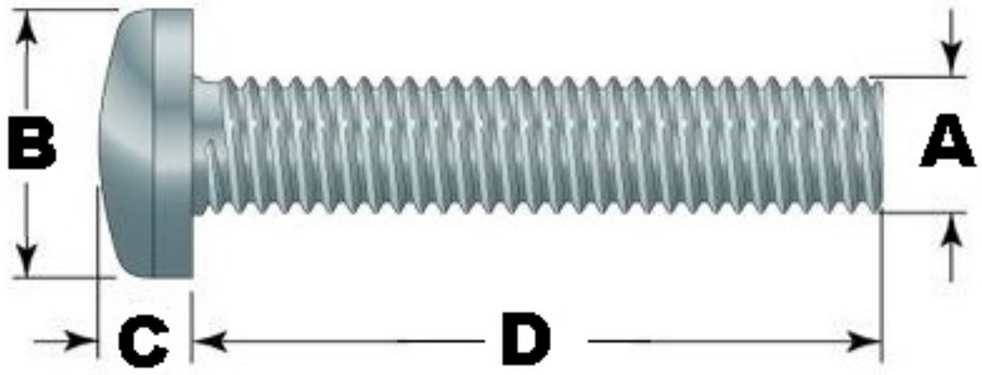
AR glass + mic 기능명세

- 제공되는 mems microphone을 raspberry pi에 설치하고 작동을 확인한다.

SAMSUNG SW ACADEMY FOR YOUTH

RC Vehicle 조립

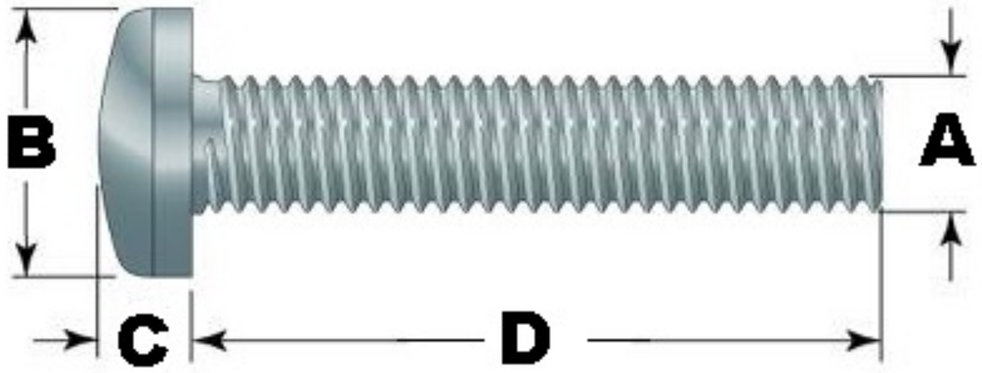
SCREW



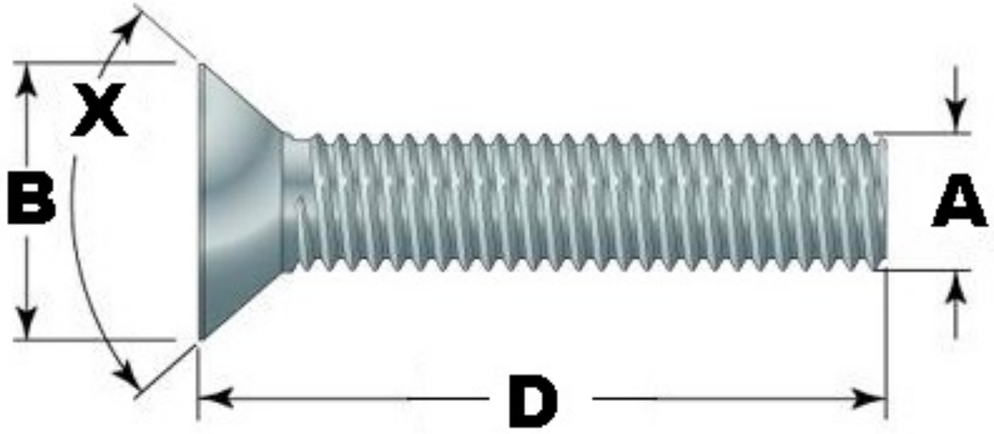
M3*8

A	나사 (유효)직경
B	머리 직경
C	머리 두께
D	길이

SCREW

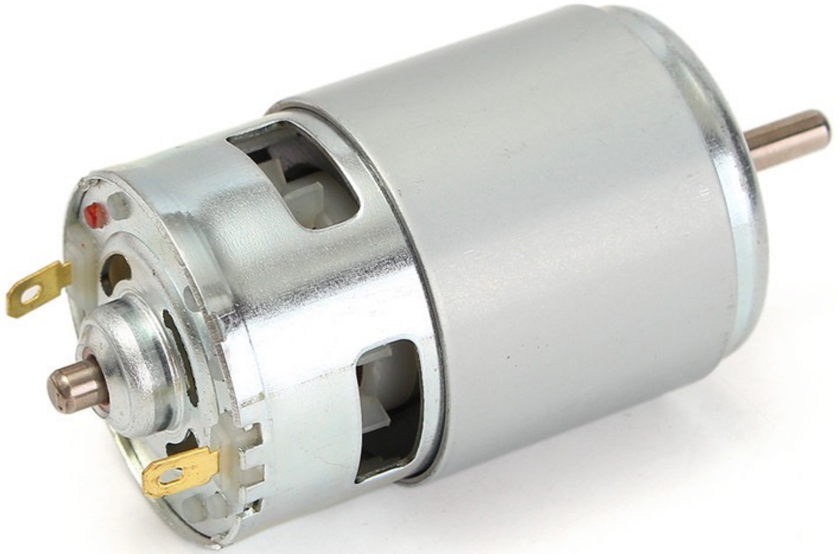


냄비머리 (PAN) 나사



접시머리 (Flat Head) 나사

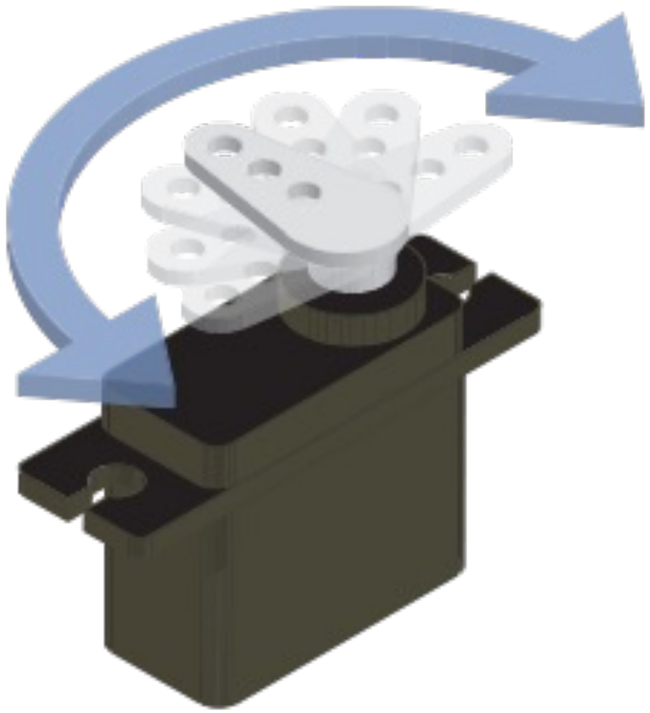
DC-motor

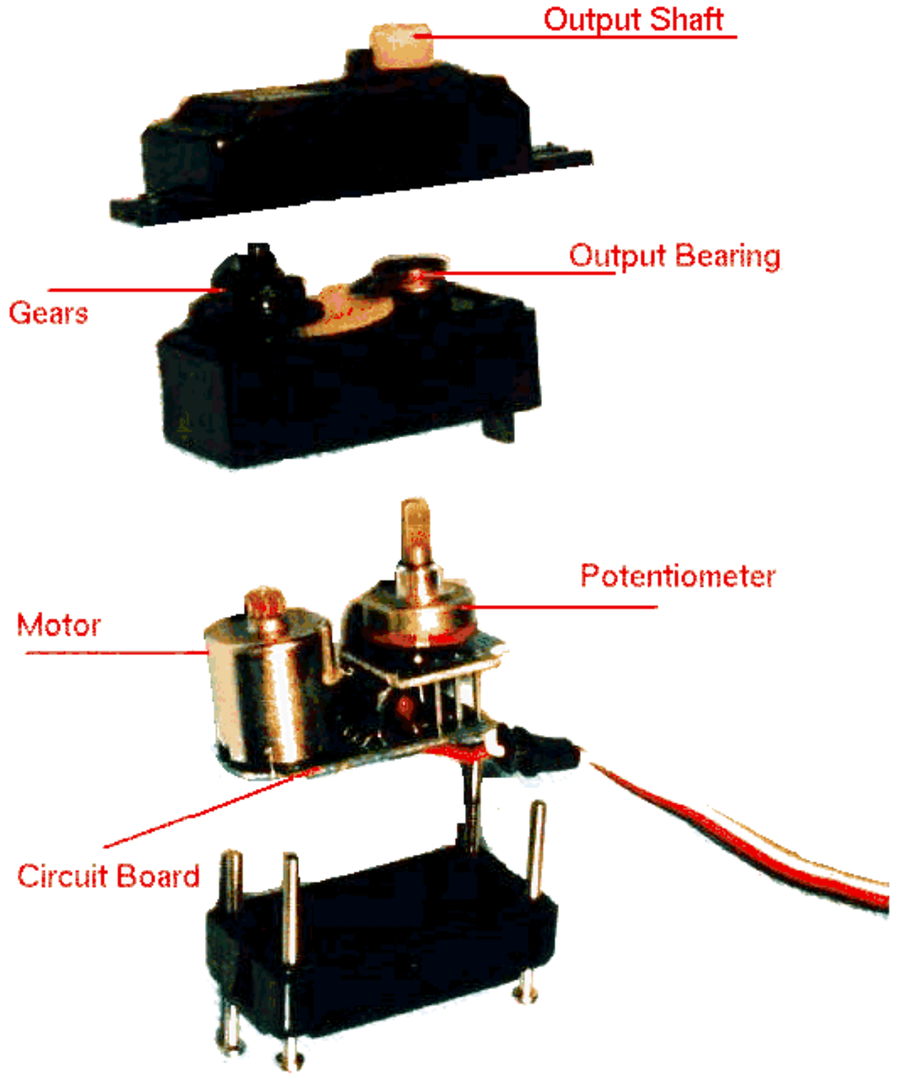


Gear

Bearing

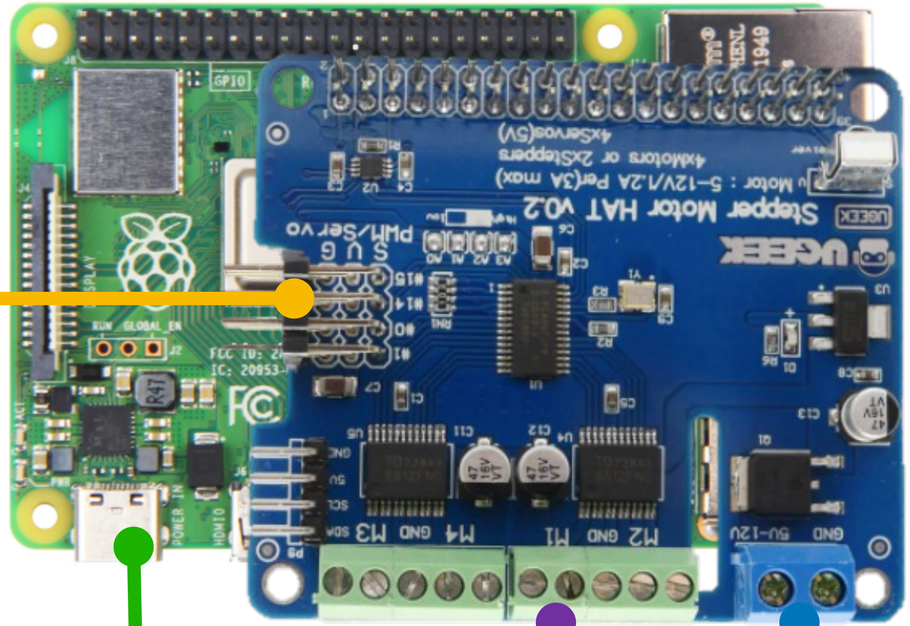
Servo




Output Shaft
Output Bearing
Gears
Potentiometer
Motor
Circuit Board

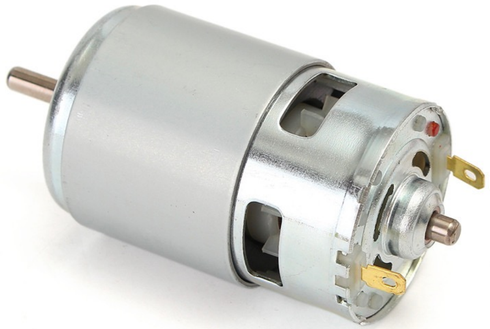
18650 Li-ion battery

RC Vehicle 구성



12V 전원 어댑터

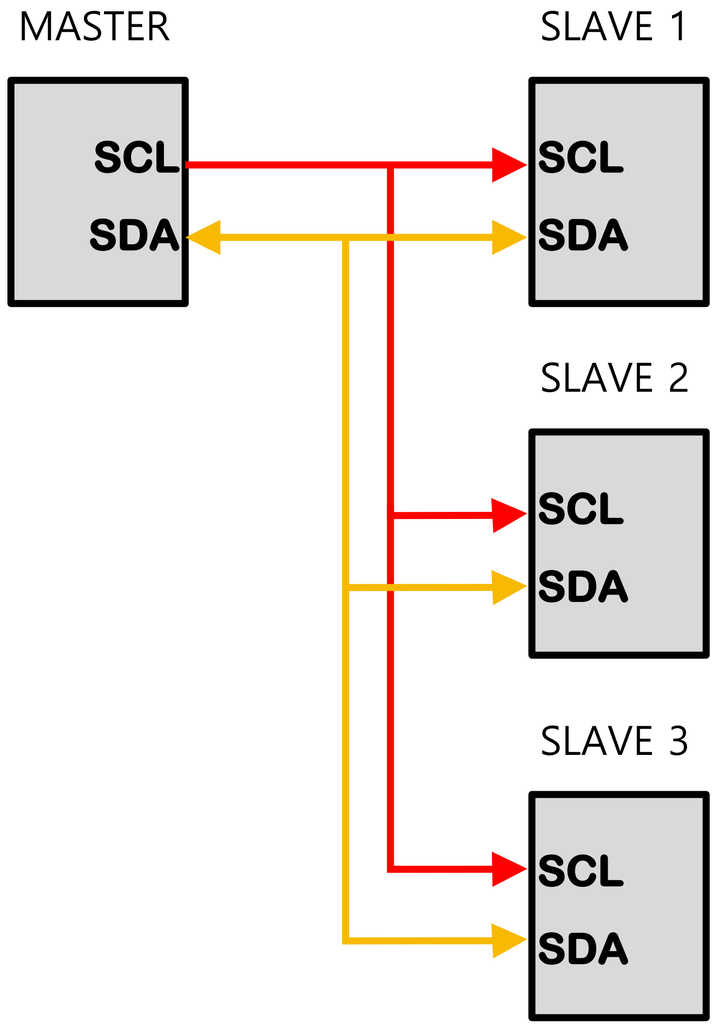
배터리

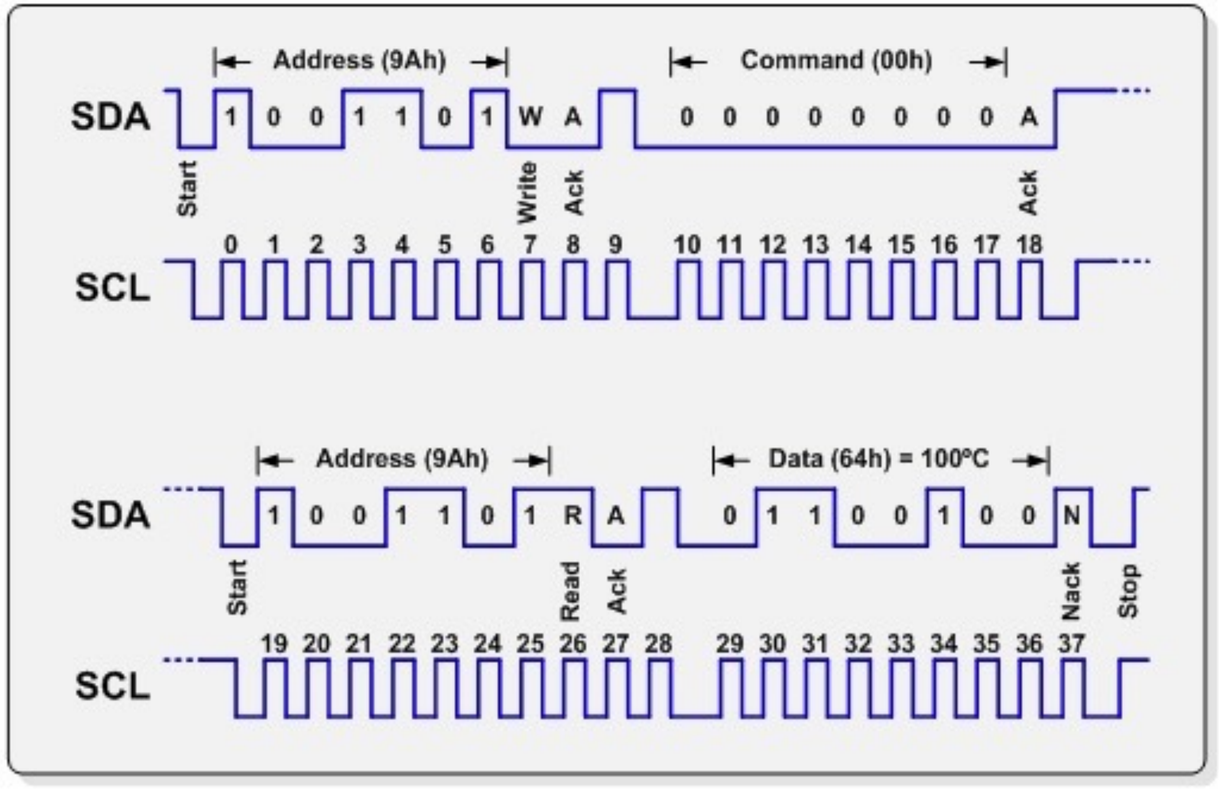


SAMSUNG SW ACADEMY FOR YOUTH

Motor 구동

I²C interface





I²C interface 활성화

- raspi-config 명령을 사용해 raspberry pi에서 i2c를 활성화한다.

```
$ sudo raspi-config
…
> 3 Interface Options > I5 I2C > YES
```

- 라즈베리파이에 연결된 i2c장치를 검색해본다.

```
$ i2cdetect -y 1
```

- 모터햇의 주소 0x6F를 확인할 수 있다.

motorHAT python library

- 모터햇 제조사에서 제공하는 라이브러리를 설치한다.

```
$ git clone https://github.com/ssafy-embedded-project/Raspi-
MotorHAT-python3.git
```

- 라이브러리 3개파일 (Raspi_MotorHAT.py, Raspi_PWM_Servo_Driver.py, Raspi_I2C.py)을 현재 작업디렉토리에 넣어둔다.

```
$ cd Raspi-MotorHAT-python3
$ cp Raspi_* ~/
```

Motor HAT – DC motor

```
from Raspi_MotorHAT import Raspi_MotorHAT, Raspi_DCMotor
mh = Raspi_MotorHAT(addr=0x6f)
myMotor = mh.getMotor(2)    # M2 터미널이 모터 연결
myMotor.setSpeed(150)   # 모터 스피드 (0~255)

myMotor.run(Raspi_MotorHAT.FORWARD)
import time
time.sleep(10)
myMotor.run(Raspi_MotorHAT.RELEASE)# 종료시 반드시 모터 정지
```

```
# Raspi_MotorHAT.run(mode)
```

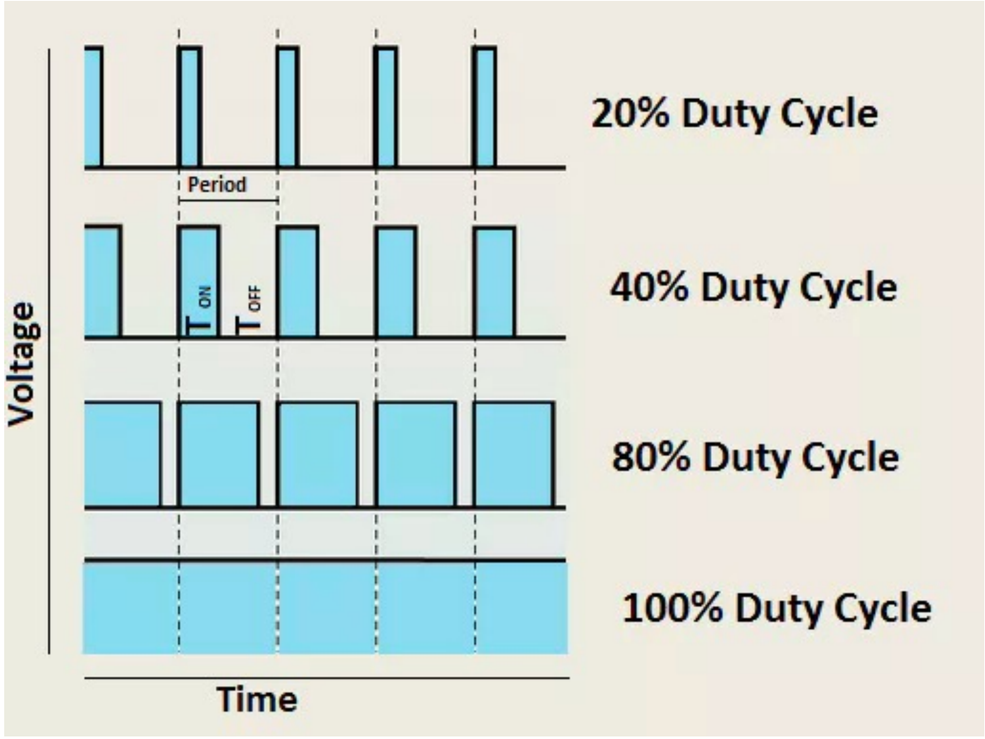
mode	
Raspi_MotorHAT.FORWARD	전진
Raspi_MotorHAT.BACKWARD	후진
Raspi_MotorHAT.RELEASE	정지

SAMSUNG SW ACADEMY FOR YOUTH

Servo 구동

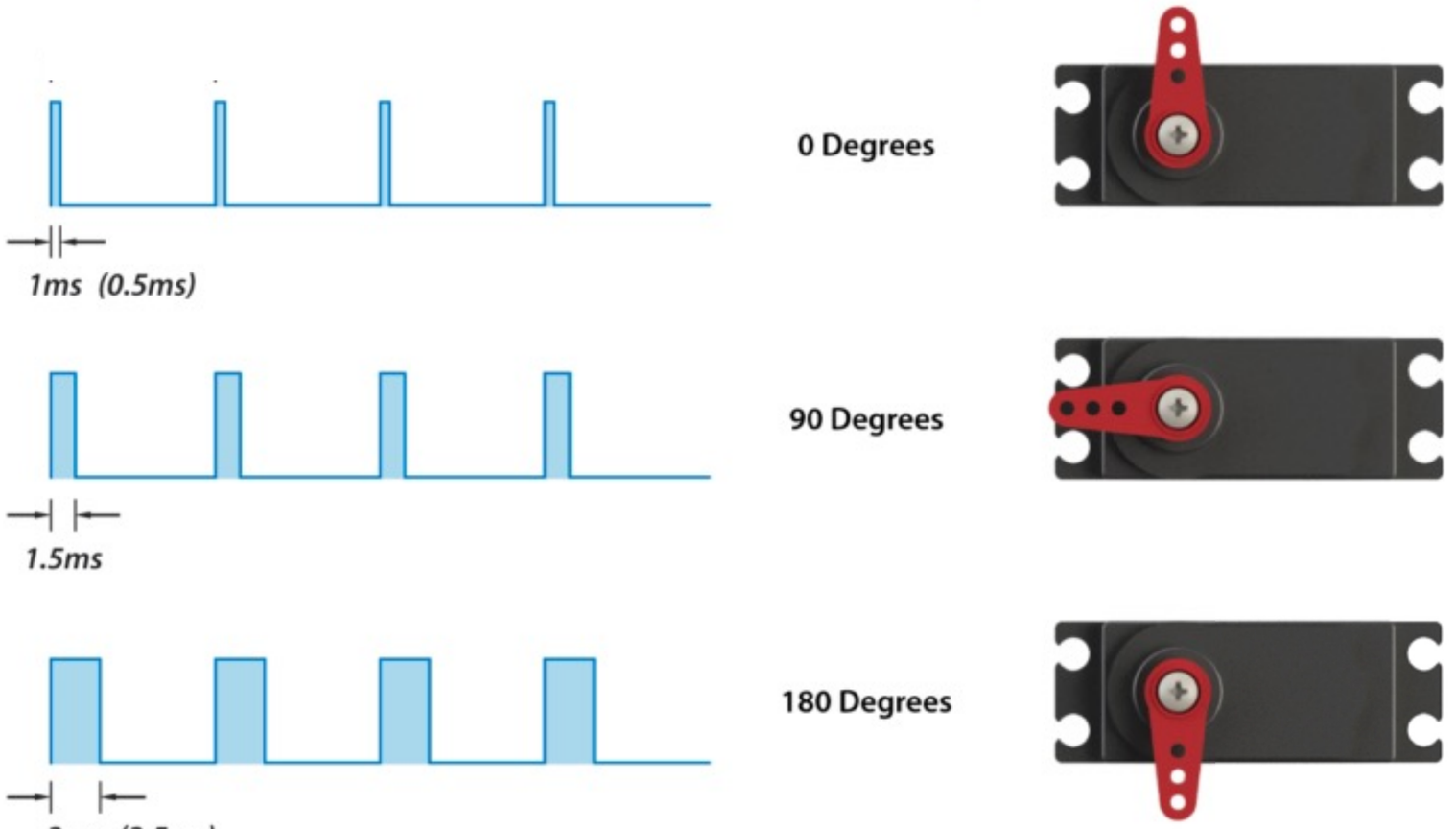
PWM (pulse width modulation) 제어

디지털 신호로 강약을 나타내는 방법



Servo 제어

Pulse의 폭에 따라 특정 각도로 움직인다



Motor HAT - Servo

```
from Raspi_MotorHAT import      \
  Raspi_MotorHAT, Raspi_DCMotor
mh = Raspi_MotorHAT(addr=0x6f)
servo_control = mh._pwm
servo_control.setPWMFreq(50)


import time
while 1:
  servo_control.setPWM(0, 0, 200)
  time.sleep(2)
  servo_control.setPWM(0, 0, 400)
  time.sleep(2)
```

- setPWM(channel, on, off)
  channel: 서보가 연결된 채널
  on: pulse 시작시점
  off: pulse 종료시점

channel	on	off	서보각도(대략)
0, 1, 14, 15	0	200	0°
	0	300	90°
	0	400	180°

- 서보의 가동 범위와 여러가지 특성은 모델마다 다르므로 직접 테스트 해본다.

SAMSUNG SW ACADEMY FOR YOUTH

input()

input() – 파이썬 내장함수

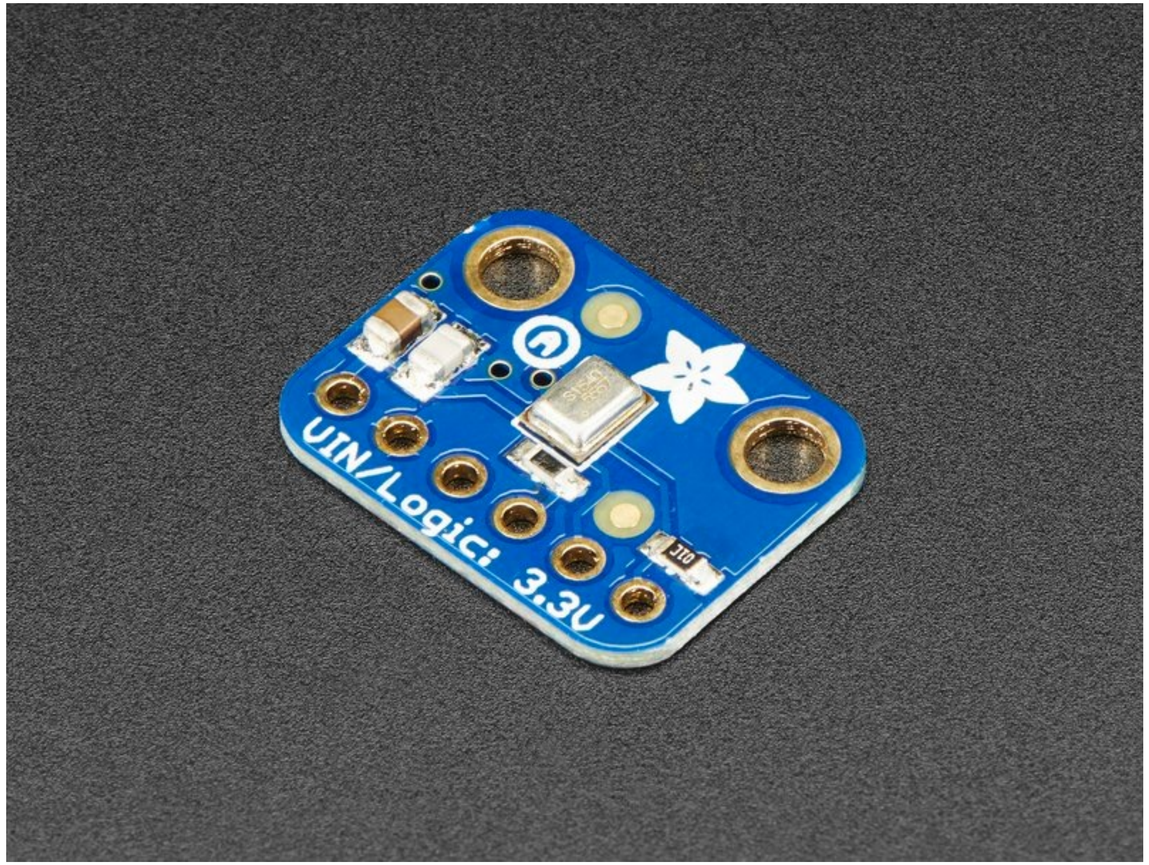
```
>>> birthday = input("생일을 입력하세요:")
생일을 입력하세요:8월6일
>>> birthday
'8월6일'
```

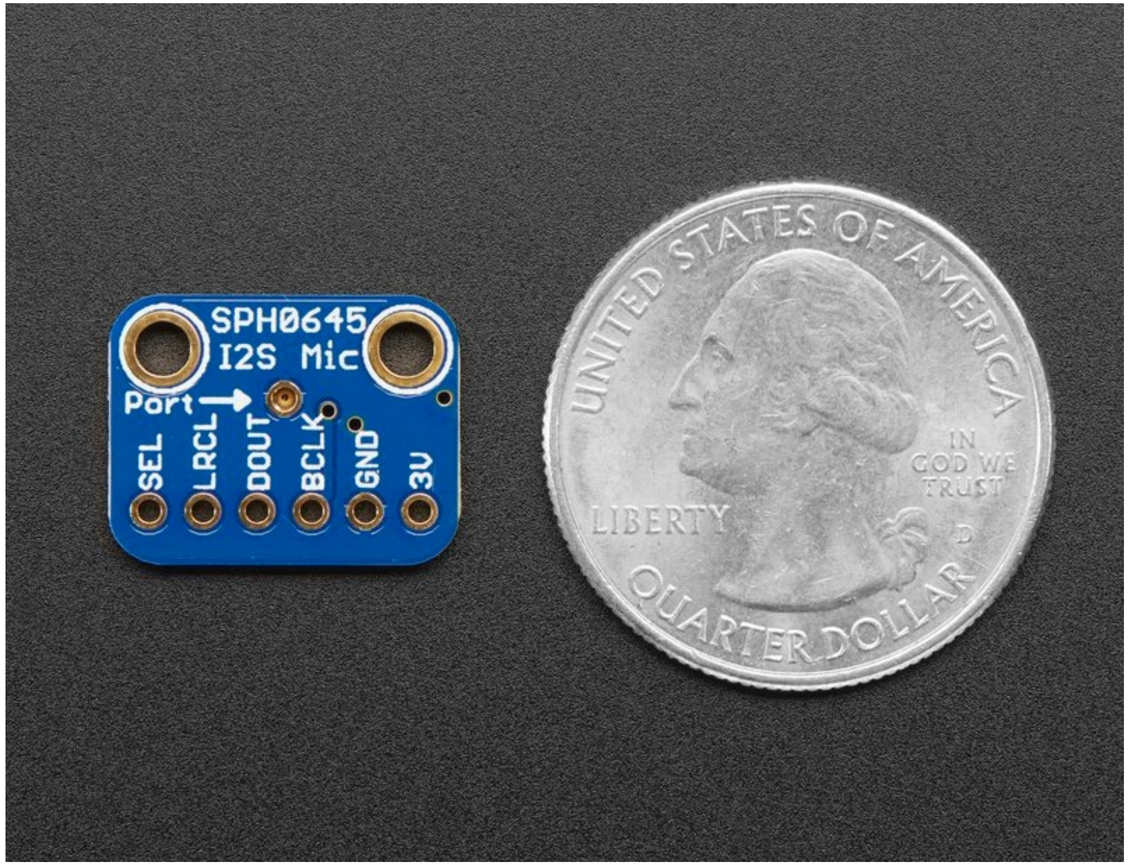
- input(prompt)
  표준입력장치로부터 문자열을 입력받는다.

SAMSUNG SW ACADEMY FOR YOUTH

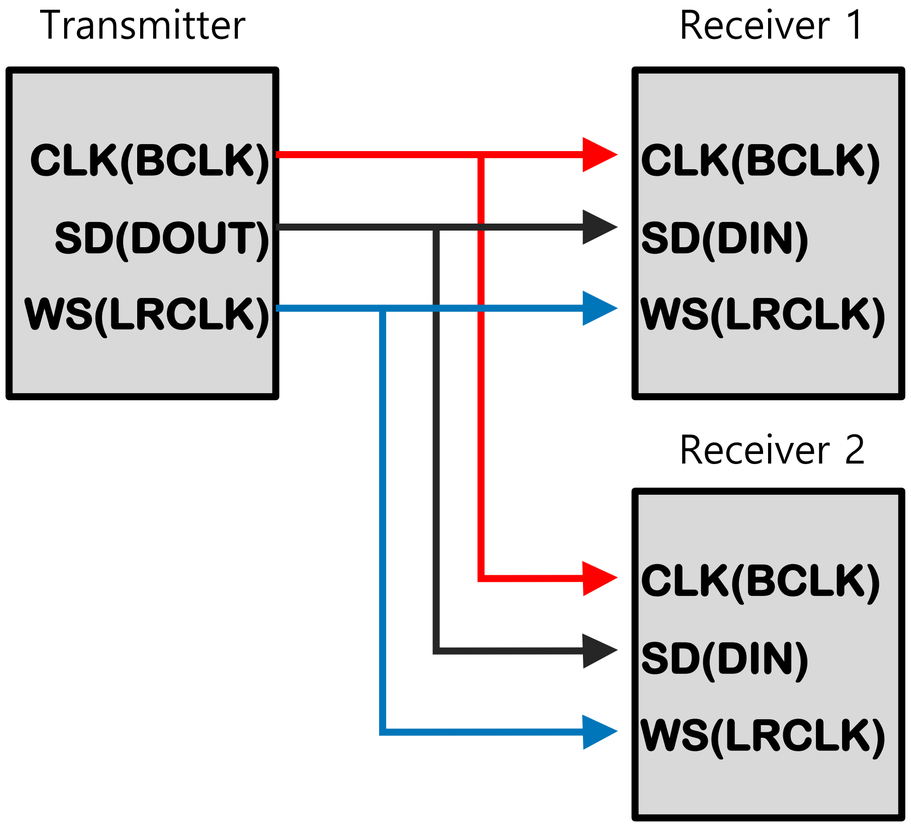
MEMS microphone

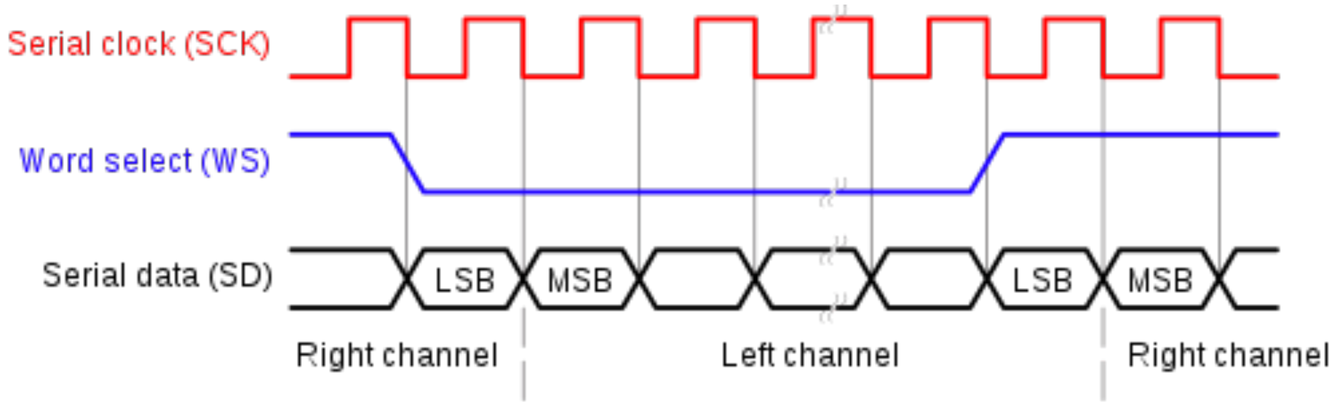
MEMS microphone



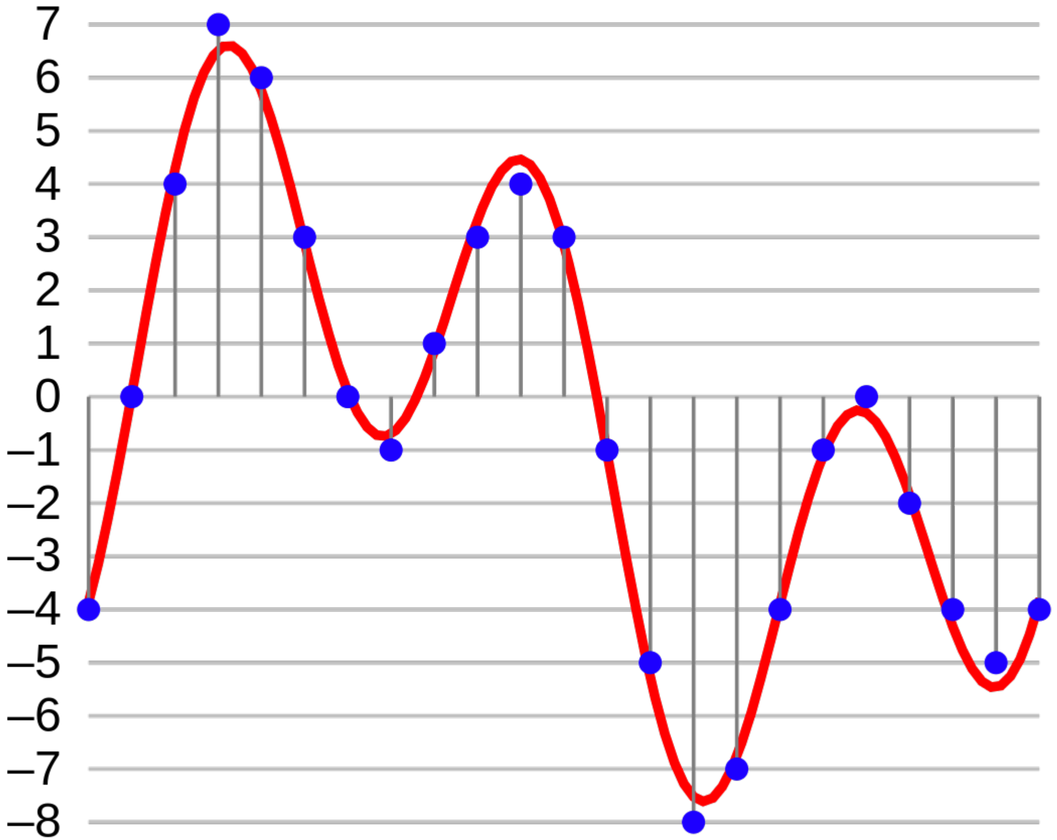


I²S interface (IIS)





PCM - digital sound



Sampling rate : 얼마나 자주, 빠른 주기로 데이터를 획득할 것인가

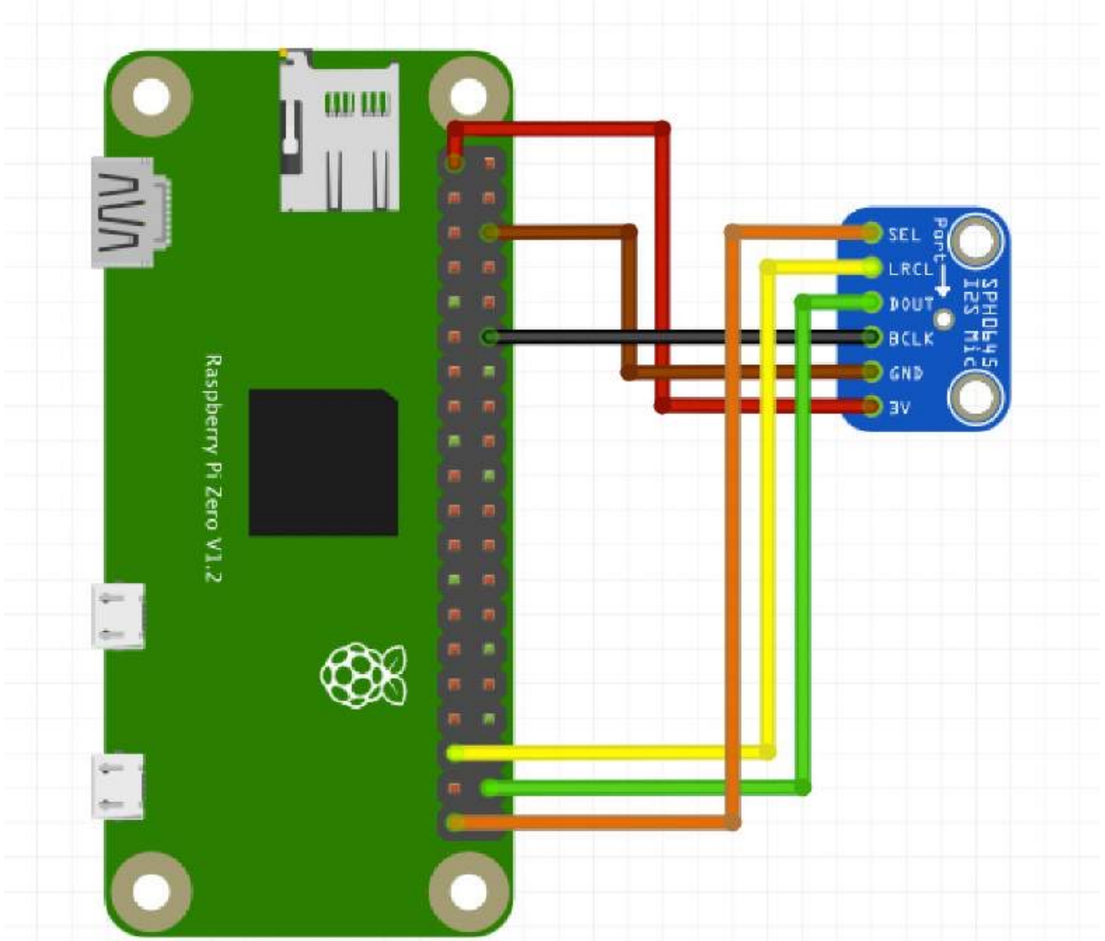
Bit depth : 신호의 크기를 몇 단계로 나타낼 것인가

Channel : mono / stereo

Bitrate : sampling rate와 Bit depth가 결정되면 필요한 데이터량이 결정되지만, 불필요한 데이터를 탈락시켜 '압축'하는 경우가 있다.

MEMS microphone 연결

3v3 Power	1		2	5v Power
GPIO 2 (I2C1 SDA)	3		4	5v Power
GPIO 3 (I2C1 SCL)	5		6	Ground
GPIO 4 (GPCLK0)	7		8	GPIO 14 (UART TX)
Ground	9		10	GPIO 15 (UART RX)
GPIO 17	11		12	GPIO 18 (CLK)
GPIO 27	13		14	Ground
GPIO 22	15		16	GPIO 23
3v3 Power	17		18	GPIO 24
GPIO 10 (SPI0 MOSI)	19		20	Ground
GPIO 9 (SPI0 MISO)	21		22	GPIO 25
GPIO 11 (SPI0 SCLK)	23		24	GPIO 8 (SPI0 CE0)
Ground	25		26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27		28	GPIO 1 (EEPROM SCL)
GPIO 5	29		30	Ground
GPIO 6	31		32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33		34	Ground
GPIO 19 (FS)	35		36	GPIO 16
GPIO 26	37		38	GPIO 20 (DIN)
Ground	39		40	GPIO 21 (DOUT)



ics43432 mic Device module 설치 1

- 리눅스는 기본적으로 새로운 디바이스를 설치하면 디바이스 드라이버를 포함하여 새로이 커널을 컴파일해야 함.
- 이 과정이 번거로우므로 커널의 일부기능만 모듈로 설치할 수 있다. 이를 커널 모듈이라함.
- 미리 컴파일 되어있는 마이크 디바이스의 커널모듈을 설치한다.

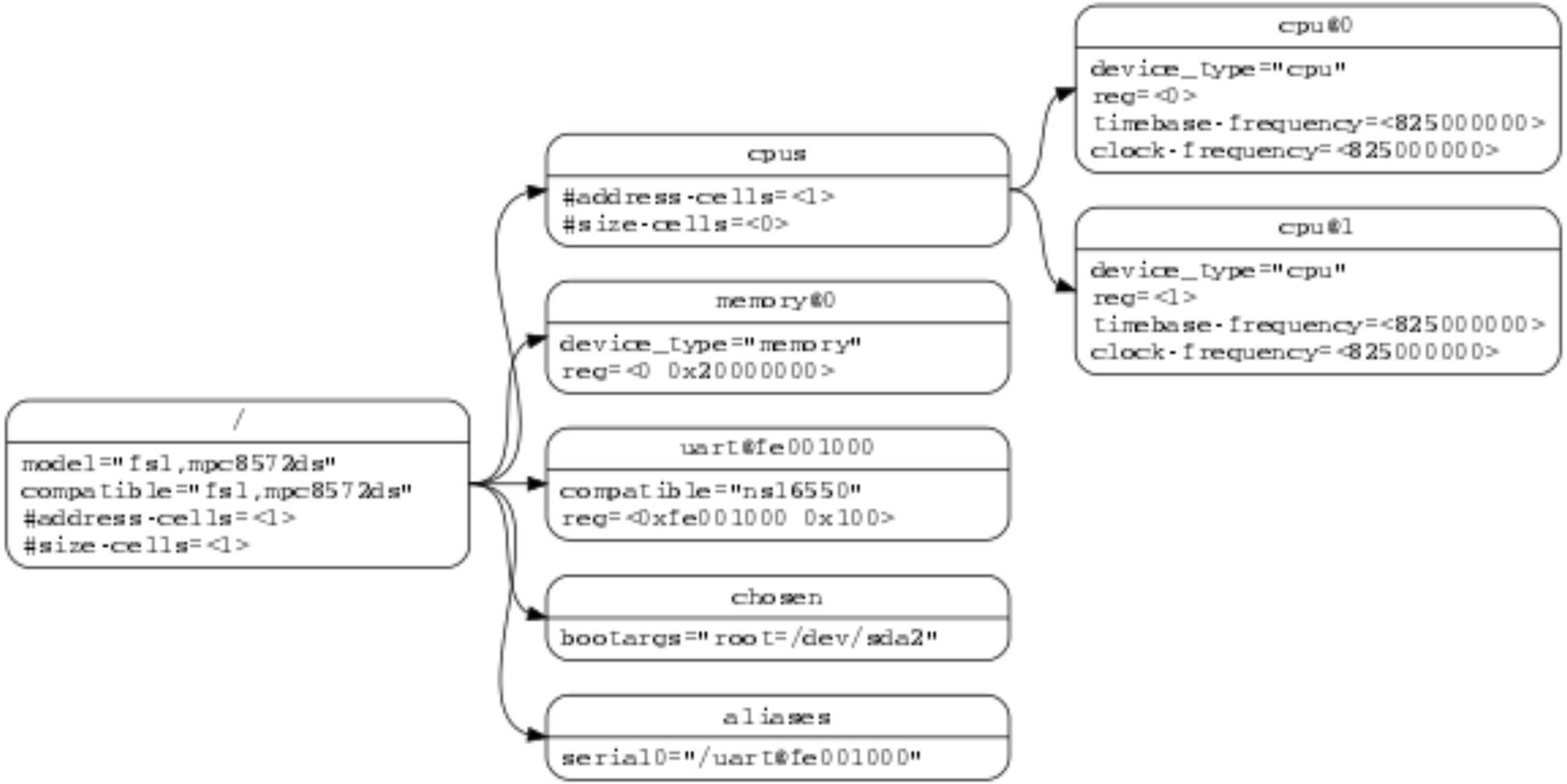
```
$ git clone https://github.com/ssafy-embedded-project/mems-
mic-deviceModule.git
```

ics43432 mic Device module 설치 2

- 해당 디바이스 모듈을 '/lib/modules/$(uname –r)' 위치에 복사하고 dependency 정리한다.

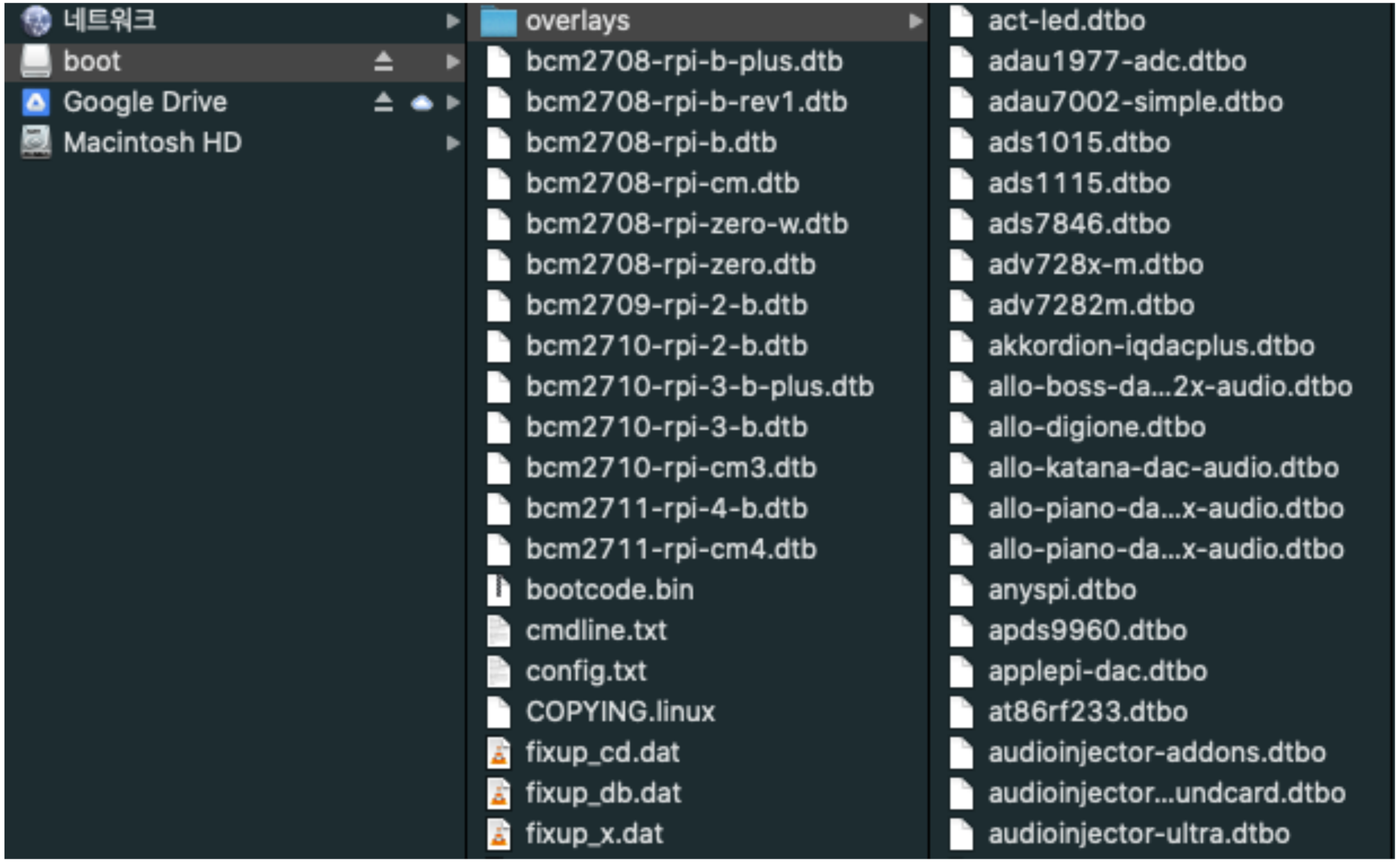
```
$ cd mems-mic-deviceModule/ics43432-pi0
$ sudo cp ics43432.ko /lib/modules/$(uname -r)
$ sudo depmod -a
```

Device Tree


/
model="fsl,mpc8572ds"
compatible="fsl,mpc8572ds"
#address-cells=<1>
#size-cells=<1>
cpus
#address-cells=<1>
#size-cells=<0>
cpu@0
device_type="cpu"
reg=<0>
timebase-frequency=<825000000>
clock-frequency=<825000000>
cpu@1
device_type="cpu"
reg=<1>
timebase-frequency=<825000000>
clock-frequency=<825000000>
memory@0
device_type="memory"
reg=<0 0x20000000>
uart@fe001000
compatible="ns16550"
reg=<0xfe001000 0x100>
chosen
bootargs="root=/dev/sda2"
aliases
serial0="/uart@fe001000"

Device Tree, Device Tree Overlay

Raspberry pi의
sd card 내용



mic Device Tree Overlay 설치

- .dtbo 파일을 /boot/overlays에 복사한다.

```
$ sudo cp i2s-soundcard.dtbo /boot/overlays
```

- config.txt에 부팅시 디바이스 사용함을 명시한다

```
$ sudo nano /boot/config.txt
… 아래 내용 uncomment(활성화)
dtparam=i2s=on
dtparam=audio=on
… 아래 내용 추가
dtoverlay=i2s-soundcard,alsaname=i2sPiSound
```

녹음 테스트

- 녹음기 프로그램 ‘arecord’로 마이크 설치 확인

```
$ arecord -l
```

- 10초간 녹음테스트. 스피커가 연결되지 않으므로 pc로 옮겨 들어본다.

```
$ arecord -c1 -r48000 -fS16_LE -d10 -vv test.wav
```

SAMSUNG SW ACADEMY FOR YOUTH

python microphone test

pyaudio

- pyaudio는 오디오 I/O를 위한 closs-platform library인 PortAudio의 파이썬 구현으로, 오디오데이터 획득, 기록 , 교환등에 비교적 간편하게 사용할 수 있다.
- 설치

```
$ sudo apt-get install portaudio19-dev
$ pip3 install pyaudio
```

마이크 입력 샘플 코드

- 마이크 입력으로부터 바이너리 스트림 raw data를 획득한다.

```
$ git clone https://github.com/ssafy-embedded-
project/pyaudio-mic-stream
```

Python context manager

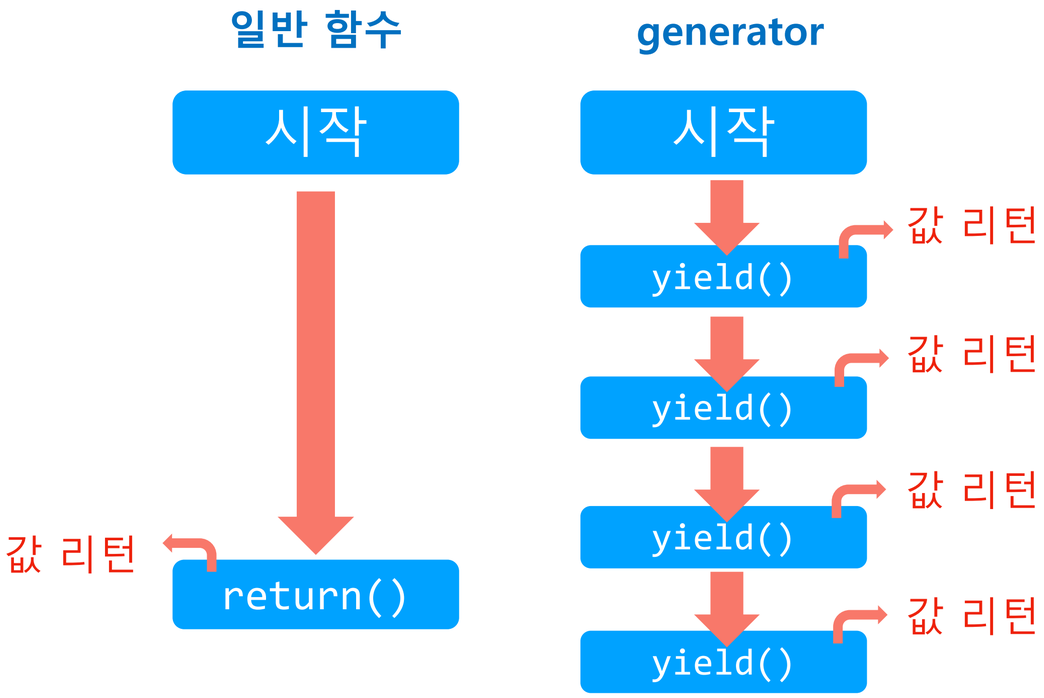
Python context manager 프로토콜을 따라 '__enter__(self)', '__exit__(self, type, value, traceback)' 가 구현된 객체는 'with' 구문에 의해 그 시작과 끝에 __enter__와 __exit__ 메소드의 실행이 보장된다.
일반적으로, 특정한 리소스의 사용을 with구문 내부로 한정하고자 할 때 사용

```
class ContextManager():
    def __init__(self):
        print('init method called')
    def __enter__(self):
        print('enter method called')
        return self
    def __exit__(self, exc_type, exc_value, exc_traceback):
        print('exit method called')


with ContextManager() as manager:
    print('with statement block')
```

Python generator

끝나지 않는 함수. return() 대신 yield()로 값을 반환한 후, 다음 번 함수 호출이 있으면 그 다음 라인부터 계속 수행한다.




```python
# A simple generator function
def my_gen():
    n = 1
    print('This is printed first')
    # Generator function contains yield statements
    yield n

    n += 1
    print('This is printed second')
    yield n

    n += 1
    print('This is printed at last')
    yield n
```

SAMSUNG SW ACADEMY FOR YOUTH

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.

SAMSUNG SW ACADEMY FOR YOUTH

감사합니다

Thank You

Python multi-threading

Threading class 사용.

```
# threading_example.py
import time
import threading #1. threading 모듈 임포트

def sub_task(seconds):  #2. 서브스레드로 샐행할 함수 정의
    val_sub = 0
    while True:
        print(f' subthread count:{val_sub}\n')
        val_sub +=1
        time.sleep(seconds)

# main thread 시작
val_main = 0

t = threading.Thread(target=sub_task, args=(2,))  #3. threading.Thread()로 스레드 오브젝트 생성
t.start() #4. 서브스레드 시작

while True:
    print(f'mainThread count:{val_main}\n')
    val_main +=1
    time.sleep(5)
```