

Sensor Programming

센서 프로그래밍

SPI

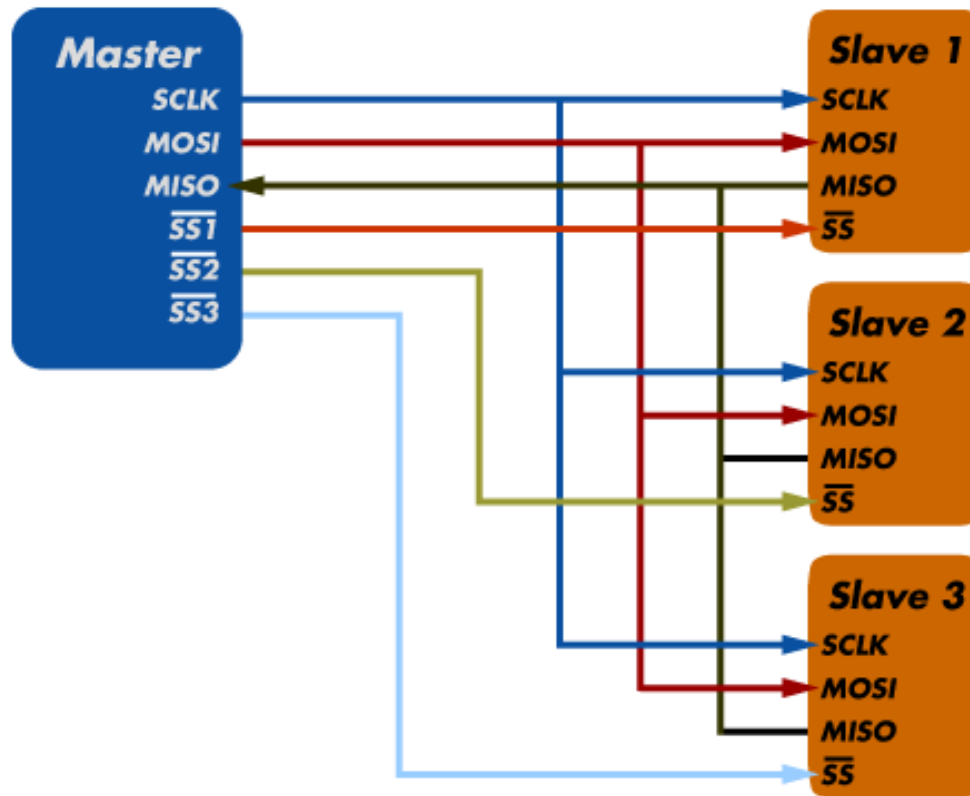


RaspberryPi



RASPBIAN

SPI



SPI

- SPI (Serial Peripheral Interface Bus)
 - 아키텍처 전이중 통신 모드로 동작.
 - 모토로라 아키텍처에 이름을 딴 동기화 직렬 데이터 연결 표준.
 - 장치들은 마스터 슬레이브 모드로 통신하며 마스터 장치는 데이터 프레임을 초기화한다.
 - 여러 슬레이브 장치들은 개별 슬레이브 셀렉트 (칩 셀렉트) 라인과 함께 동작할 수 있다.
- SPI 버스는 4가지 논리 신호를 지정한다
 - SCLK: 직렬 클럭 (마스터로부터의 출력)
 - MOSI; SIMO: 마스터 출력, 슬레이브 입력 (마스터로부터의 출력)
 - MISO; SOMI: 마스터 입력, 슬레이브 출력 (슬레이브로부터의 출력)
 - SS: 슬레이브 셀렉트 (active low, 마스터로부터의 출력).

- 장점
 - 완전한 전이중 통신
 - 전송되는 비트에 대한 완전한 프로토콜 유연성
 - 전송기가 필요하지 않음
 - 매우 단순한 하드웨어 인터페이스 처리
 - IC 패키지에 4개의 핀만 사용하며 이는 병렬 인터페이스에 비해 수가 적은 것이다.
- 단점
 - 하드웨어 슬레이브 인식이 없음
 - 슬레이브에 의한 하드웨어 흐름 제어가 없음
 - 오류 검사 프로토콜이 정의되어 있지 않음
 - 일반적으로 노이즈 스파이크에 영향을 받는 경향이 있음
 - RS-232, RS-485, CAN 버스보다 비교적 더 짧은 거리에서 동작
 - 하나의 마스터 장치만 지원

SPI

- SPI blacklist 제거

```
# sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

```
# blacklist spi and i2c by default (many users don't need them)

#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

저장 : control-X 누른 후 Y 누름

- 재부팅

```
# sudo reboot
```

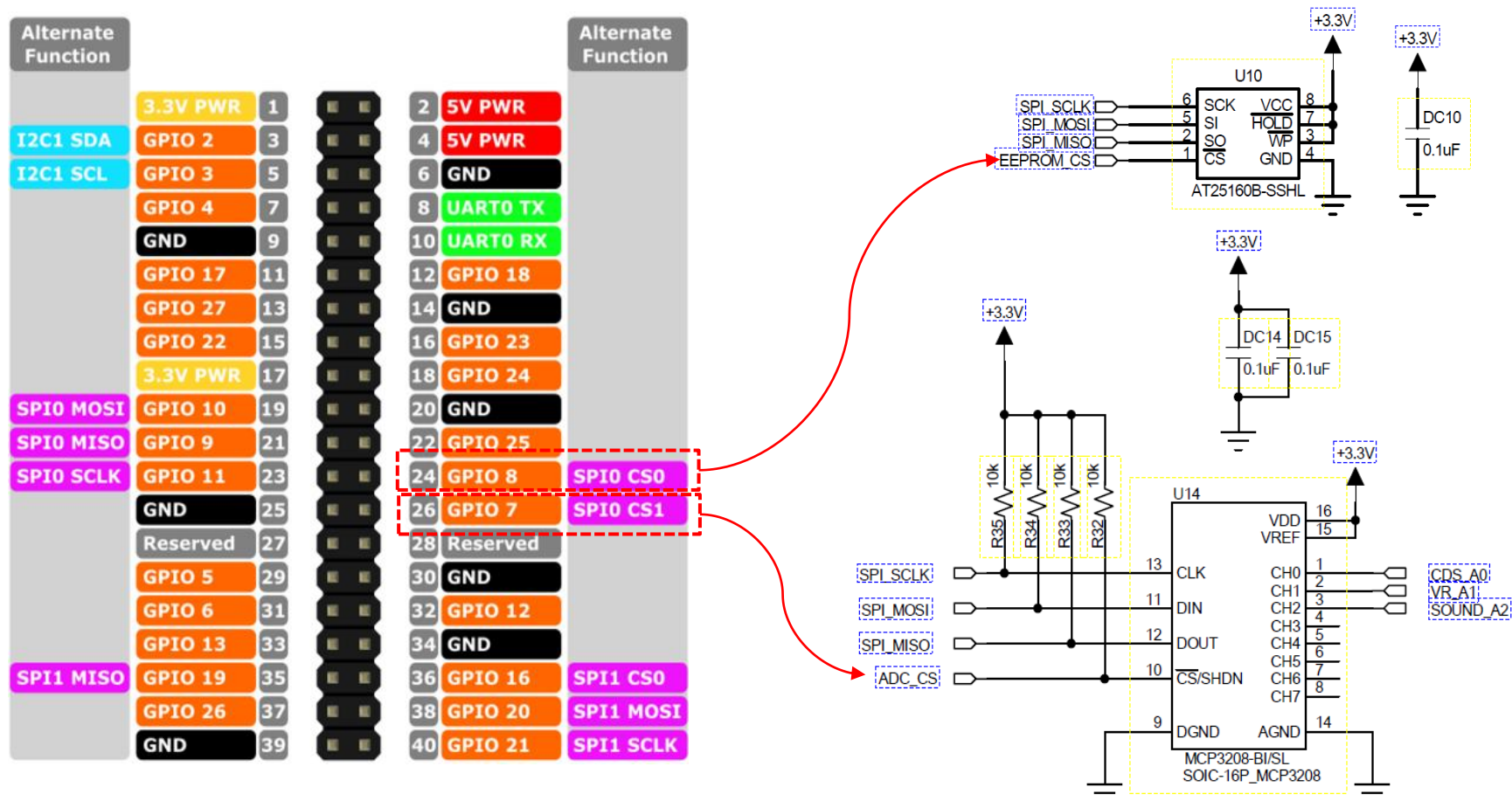
- SPI 활성화 확인

```
pi@pi2 ~ $ ls -l /dev/spidev*
crw-rw---T 1 root spi 153, 0 Jan  1  1970 /dev/spidev0.0
crw-rw---T 1 root spi 153, 1 Jan  1  1970 /dev/spidev0.1
```

SPI

- `spi = spidev.SpiDev()`
 - SPI 객체 생성
- `spi.open(bus, device)`
 - `bus` ; spi port,
 - `device` : chip select number
- `spi.close`
 - 종료 전 port close
- `spi.max_speed_hz`
 - HZ로 bus 최대 버스 속도 설정
- `spi.readbytes(len)`
 - Len 만큼의 바이트를 디바이스에서 읽어 list로 리턴함
- `spi.writebytes([value,value1,...])`
 - List에 저장된 값들을 디바이스에게 전송함
- `spi.xfer2 ([value, value1, ...])`
 - 리스트의 개수만큼 디바이스에 전송하고 **전송한 개수**만큼 list로 리턴함

SPI 마스터 연결

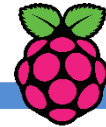


- SPI0, SPI1 : 2개의 채널 지원, CS0 : 칩 셀렉터 0번, CS1 : 칩 셀렉터 1번
- 현재 스펙에서 지원 가능한 SPI 슬레이브 디바이스 개수는? 3개

Sensor Programming

센서 프로그래밍

SPI EEPROM



RaspberryPi



RASPBIAN

SPI EEPROM

```
import spidev
import time

spi = spidev.SpiDev()
spi.open(0, 0) # SPI 채널 0번 , cs0 (칩 선택터 0번)
spi.max_speed_hz = 1000000 # 1Mhz

WREN = 0x06
WRITE = 0x02
READ = 0x03
WRDI = 0x04
RDSR = 0x05
WRSR = 0x01

dummy = 0
max_size = 15
```

SPI EEPROM

```
try:
    spi.writebytes( [WREN] )
    time.sleep(0.001)

    buff = [WRITE, 0x00, 0x11]
    for i in range(max_size):
        buff.append(i)
    spi.writebytes(buff)
    time.sleep(0.001)

    spi.writebytes ( [WRDI] )
    time.sleep(0.001)
```

SPI EEPROM

```
#try: (continue)
    buff = [READ, 0x00, 0x11]
    for i in range(max_size):
        buff.append(dummy)

    read = spi.xfer2( buff )
    time.sleep(0.001)
    print read

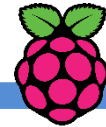
except KeyboardInterrupt:
    pass

finally:
    spi.close()
```

Sensor Programming

센서 프로그래밍

SPI MPU3208

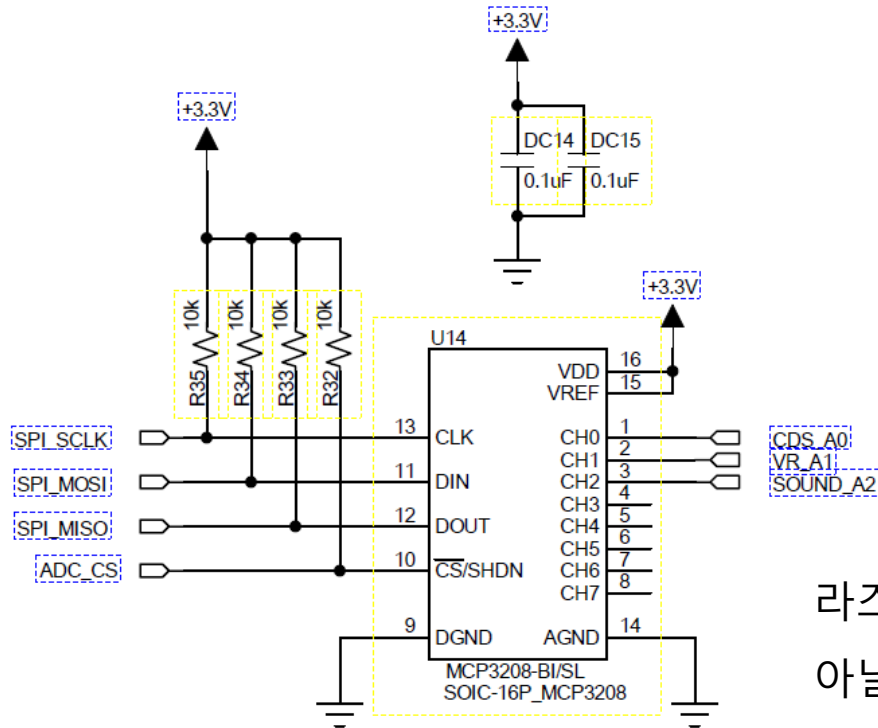


RaspberryPi



RASPBIAN

SPI MCP3208

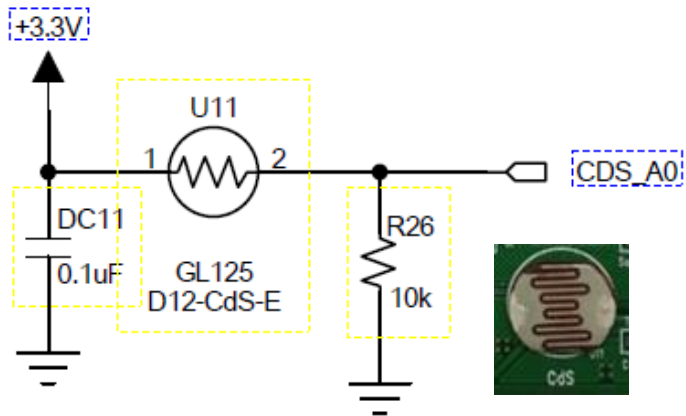


라즈베리파이는 칩 내부에 ADC 기능이 없으므로
아날로그 센서 디바이스를 직접 연결 불가

MCP3208

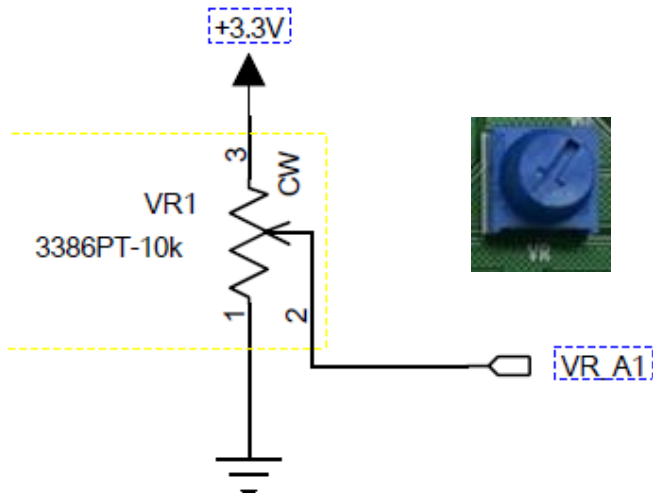
- 0v ~ 3.3v 의 센서 출력 값을 0 ~ 4095 (12bit) 의 디지털값으로 변환해주는 IC
- SPI 통신을 통해 각 채널에 연결된 디바이스의 센싱 값과 명령을 READ/WRITE 가능
- 최대 8-Channel 사용 가능, 현재 회로에서는 3개의 디바이스만 연결됨

SPI MPU3208



GL125

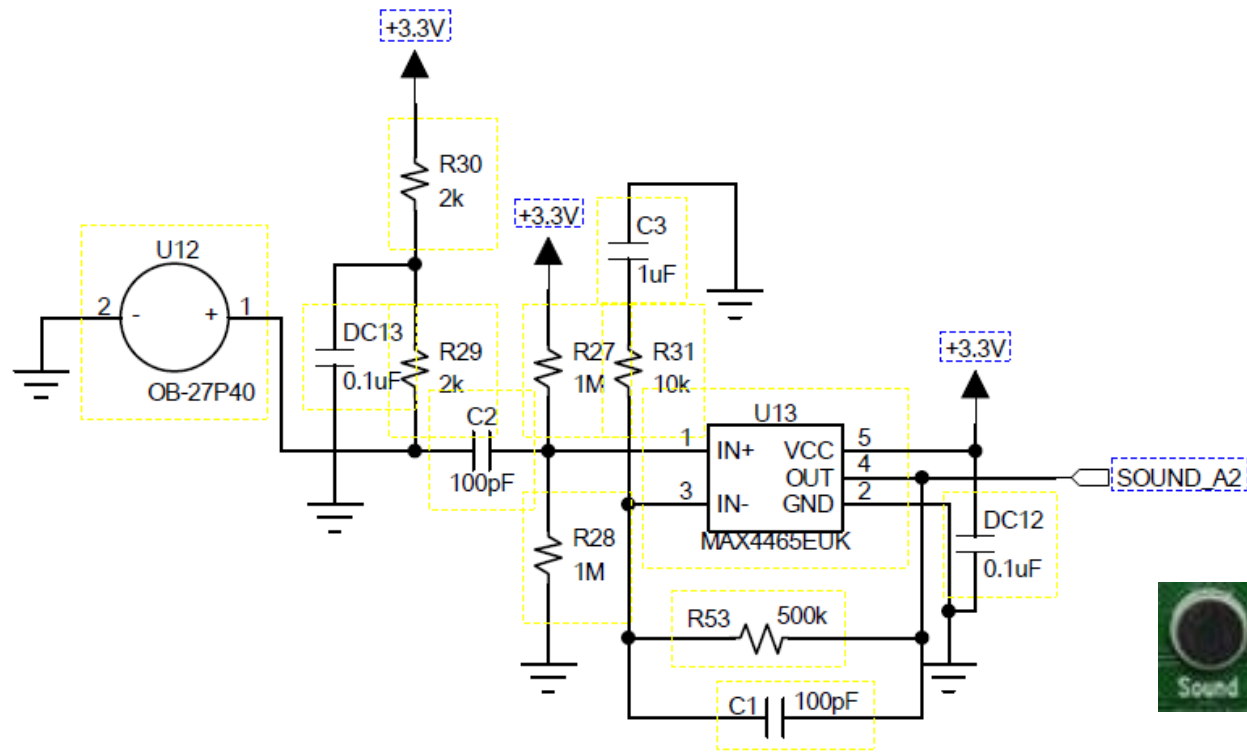
- Cadmium Sulfide (CdS)
- 빛에 따라 저항 값 변경
- 저항에 대한 전압을 CDS_A0로 (MCP3208 채널 0번) 출력



3386PT (가변저항)

- 조절 단자의 회전 각도에 따라 저항값 변화
- 저항에 대한 전압을 VR_A1로 (MCP3208 채널 1번) 출력

SPI MPU3208



OB-27P40

- 소리의 크기를 전압 신호로 변환해 주는 마이크

MAX4465EUK

- 마이크의 출력 신호를 adc에 입력 전압에 맞도록 증폭시키는 역할을 함

SPI MPU3208

```
import spidev
import time

spi = spidev.SpiDev()
spi.open(0,1) # SPI 채널 0번, CS1 : 칩 셀렉터 1번
adc_read =[0,0,0]

def analog_read(ch):
    # 12 bit
    r = spi.xfer2([0x6 | (ch & 0x7) >> 2, ((ch & 0x7) << 6),0 ])
    adcout= ((r[1] & 0xf) << 8) + r[2]

    # 10 bit
    #r= spi.xfer2([1, (8+channel)<<4, 0])
    #ret = ((r[1]&3) << 8) + r[2]
    return adcout
```


SPI MPU3208

```
try:
    while 1:
        adc_read[0] = analog_read(0)
        adc_read[1] = analog_read(1)
        adc_read[2] = analog_read(2)
        print("[csd vr sound]")
        print(adc_read)
        time.sleep(1)

finally:
    spi.close()
```