

소리 감지 센서

김 동 훈

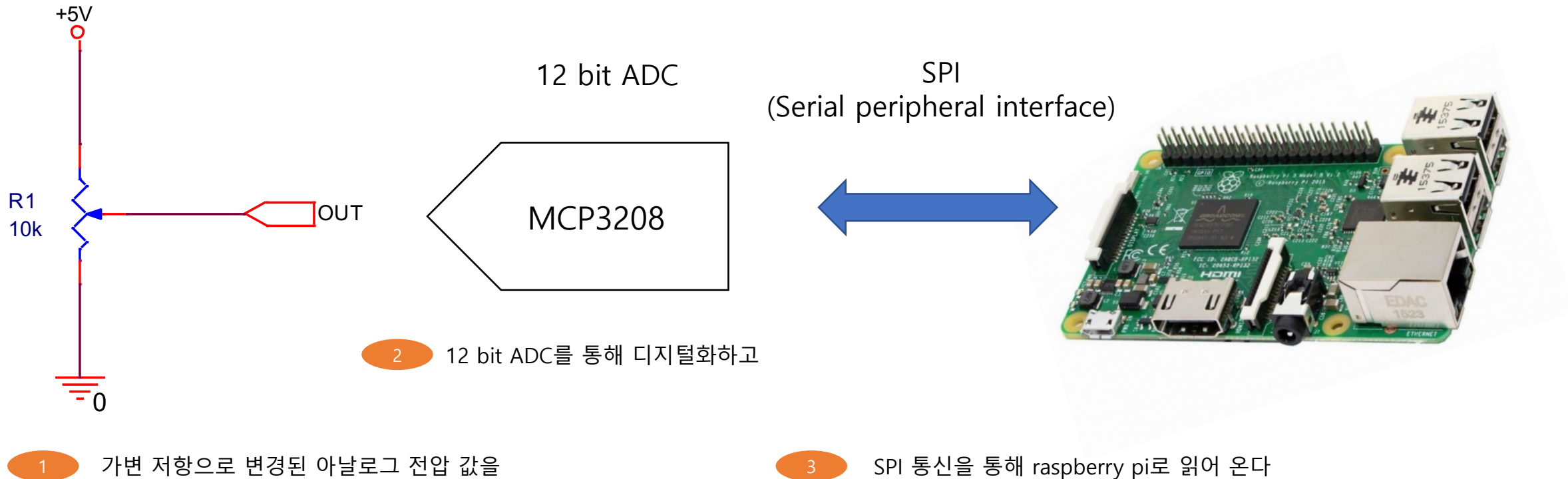
시작

• 강의 소개

- 이번 강의에서는 소리 감지 센서와 함께 ADC를 좀더 확장하여 샘플링 이론에 대한 내용을 다루게 됩니다.
- 소리는 AC(Alternative current) 신호입니다. DC와 달리 소리를 ADC하기 위해서는 샘플링 이론을 바탕으로 샘플링 주파수 및 이의 복조화 과정을 이해하고 있어야 합니다.
- 이번 강의에서는 소리를 감지하기 위한 이론적 내용과 이의 실습을 진행합니다.
- 실습 프로그램은 코드 리딩을 통해 프로그램의 흐름과 동작을 이해 할 수 있습니다.
- 실습 프로그램은 다음과 같이 진행하기 바랍니다.
 - 먼저 실습에서 주어진 문제를 읽고 이해하시기 바랍니다.
 - 실습코드를 공개 했으니 코드 리딩을 통해 프로그램의 흐름을 파악하시기 바랍니다.
 - 실습 코드의 흐름이 파악되면 그 동작을 이해 할 수 있습니다.
 - 이러한 과정은 프로그램 개발과정의 일부분이니 익숙해 지시는 것이 필요합니다.
 - 실습이 가능해지면 실습을 통해서 동작을 확인할 예정이니 큰 부담 갖지 말고 진행하시기 바랍니다.
 - 코드 리딩에 필요한 주석은 프로그램에 달려 있으니 꼼꼼히 확인하시기 바랍니다.

소리 감지 센서

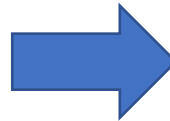
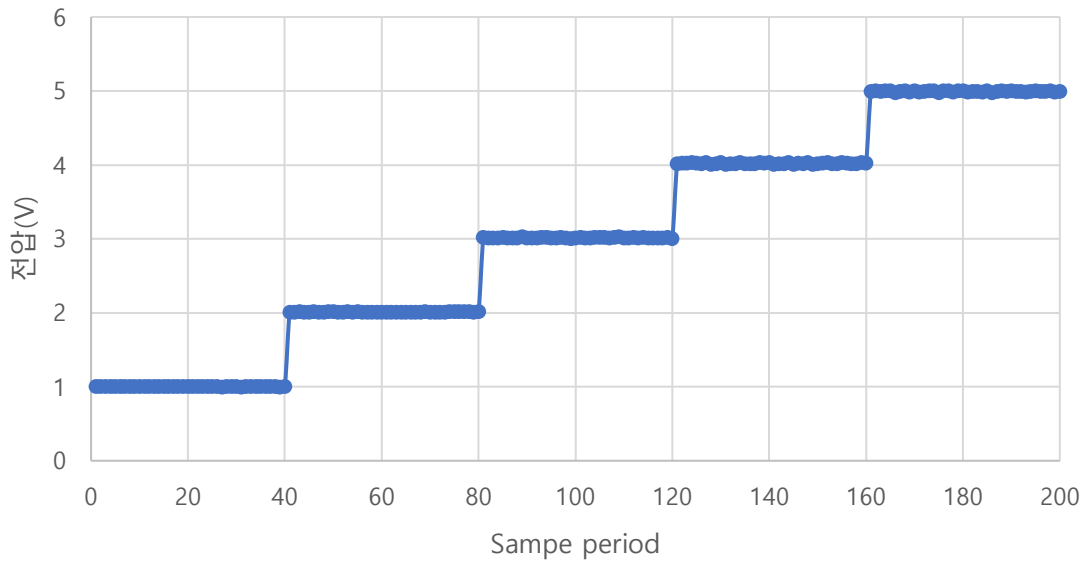
- ADC(Analog-to-digital converting)



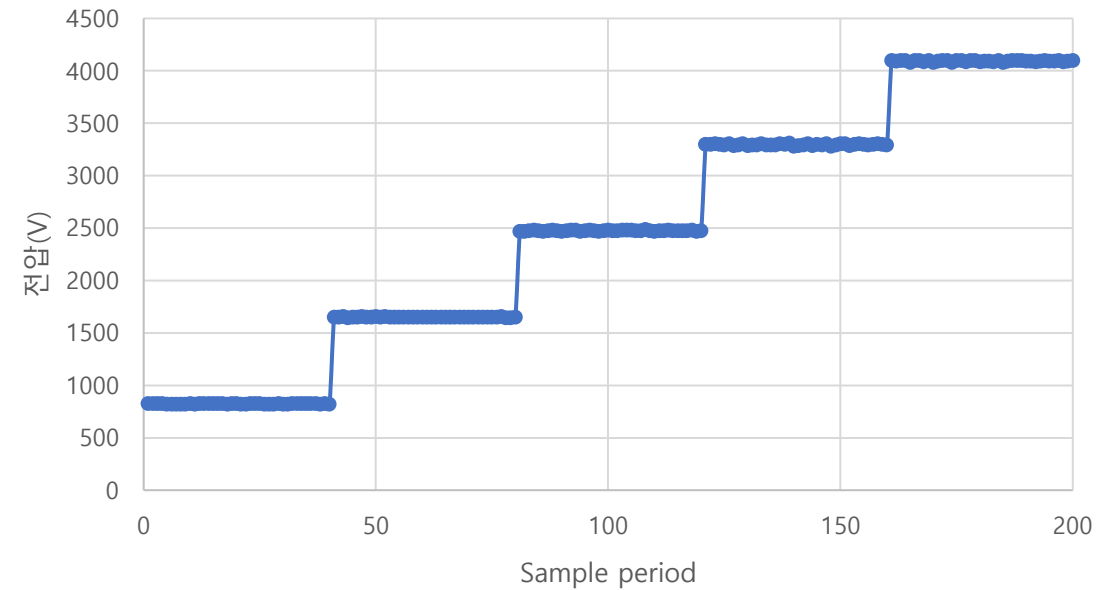
소리 감지 센서

- ADC(Analog-to-digital converting)

Analog

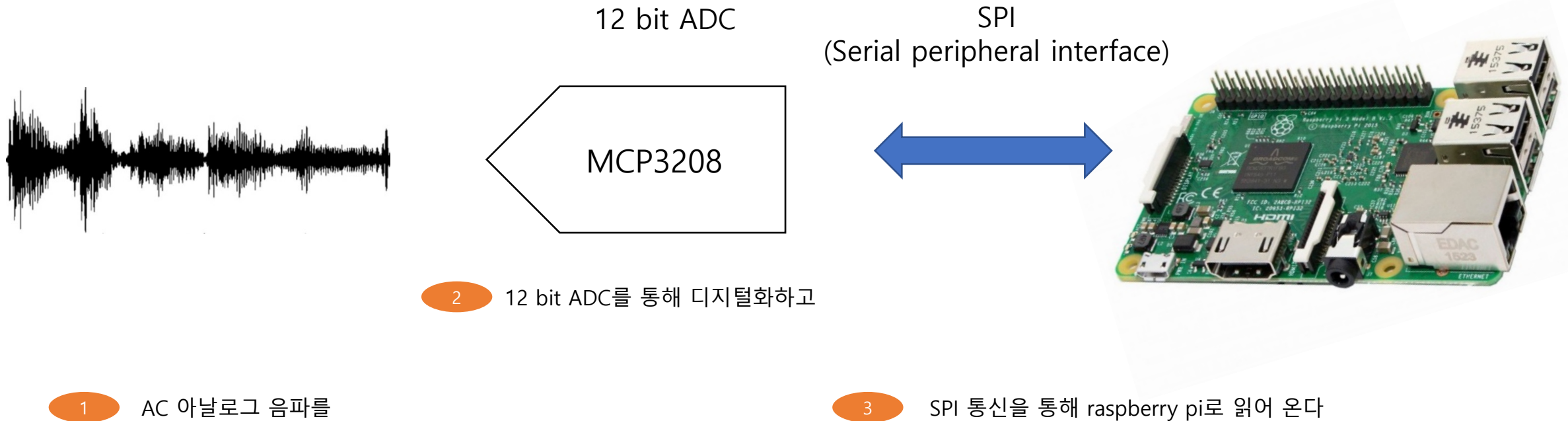


ADC



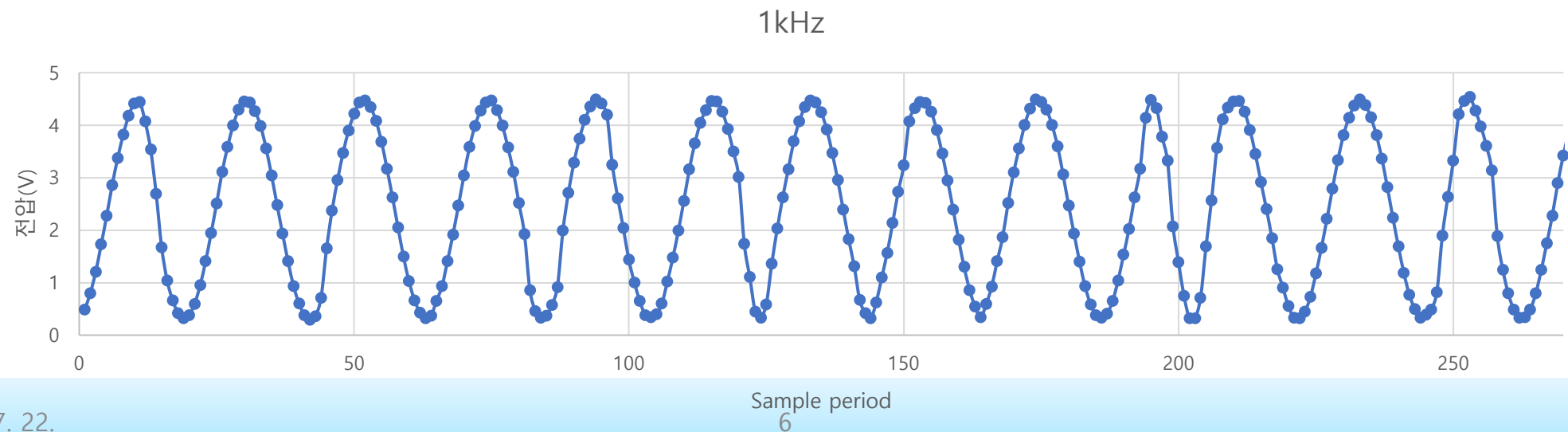
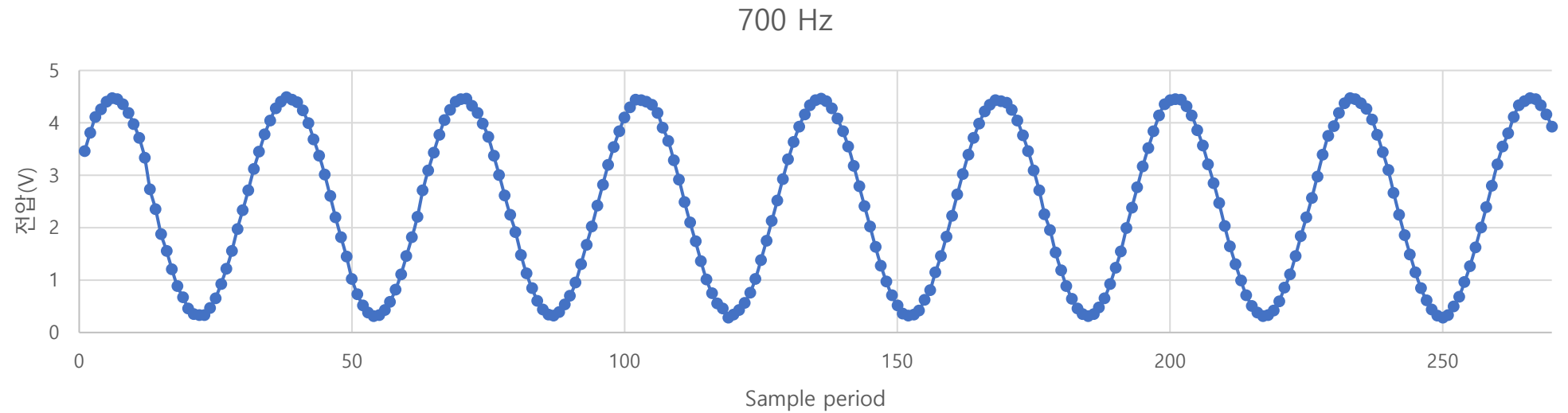
소리 감지 센서

- ADC(Analog-to-digital converting)



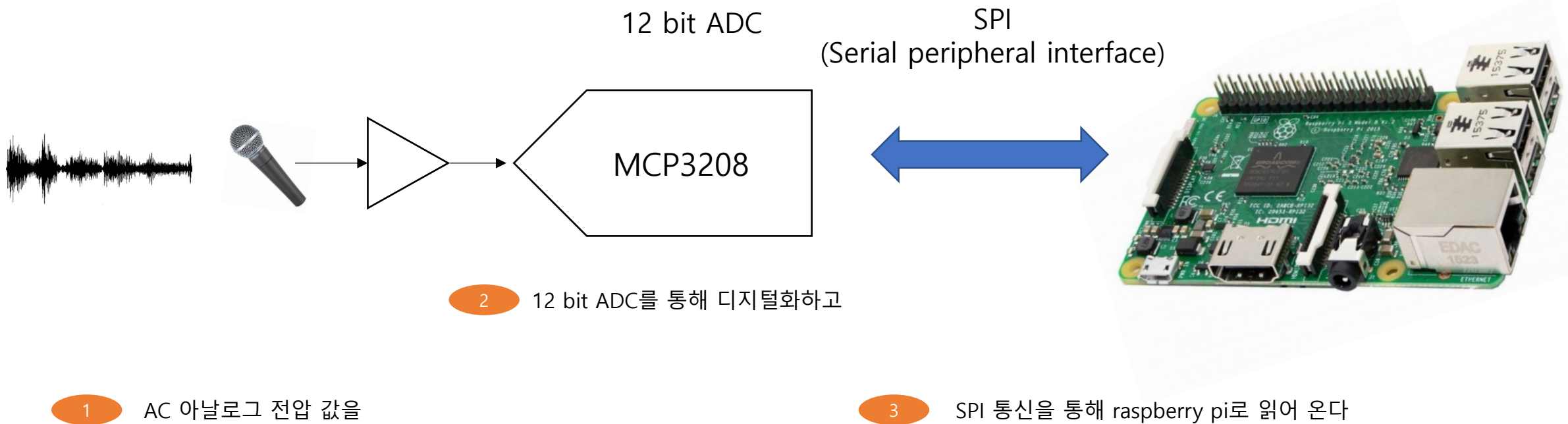
소리 감지 센서

- ADC(Analog-to-digital converting)



소리 감지 센서

- ADC(Analog-to-digital converting)



소리 감지 센서

- 소리 감지 센서

- Microphone

- Mike/mic. 라고도 통칭
 - 음파를 전기적인 에너지로 변환하기 위한 장치를 의미
 - 소리를 같은 파형의 전기 신호로 변환하는 장치
 - Dynamic microphone과 condenser microphone이 대표적

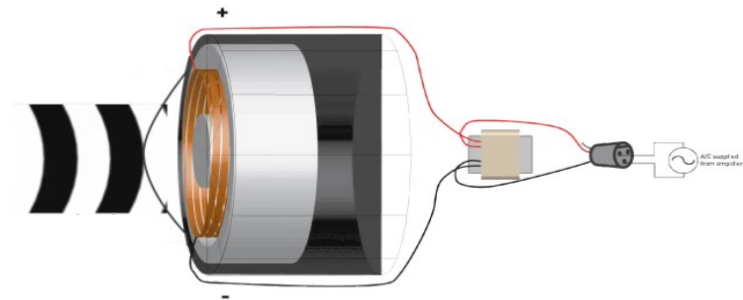
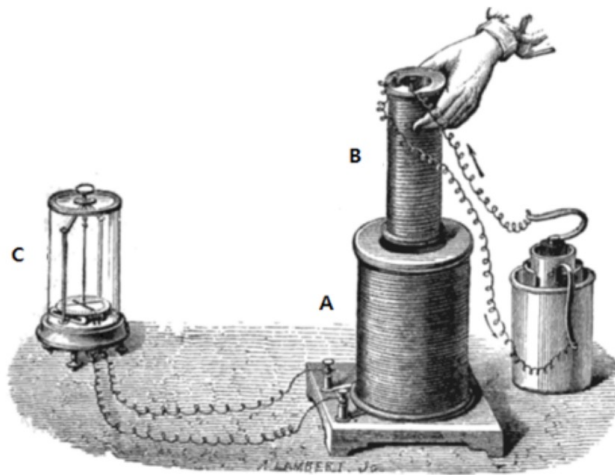


소리 감지 센서

• 소리 감지 센서

- Dynamic microphone

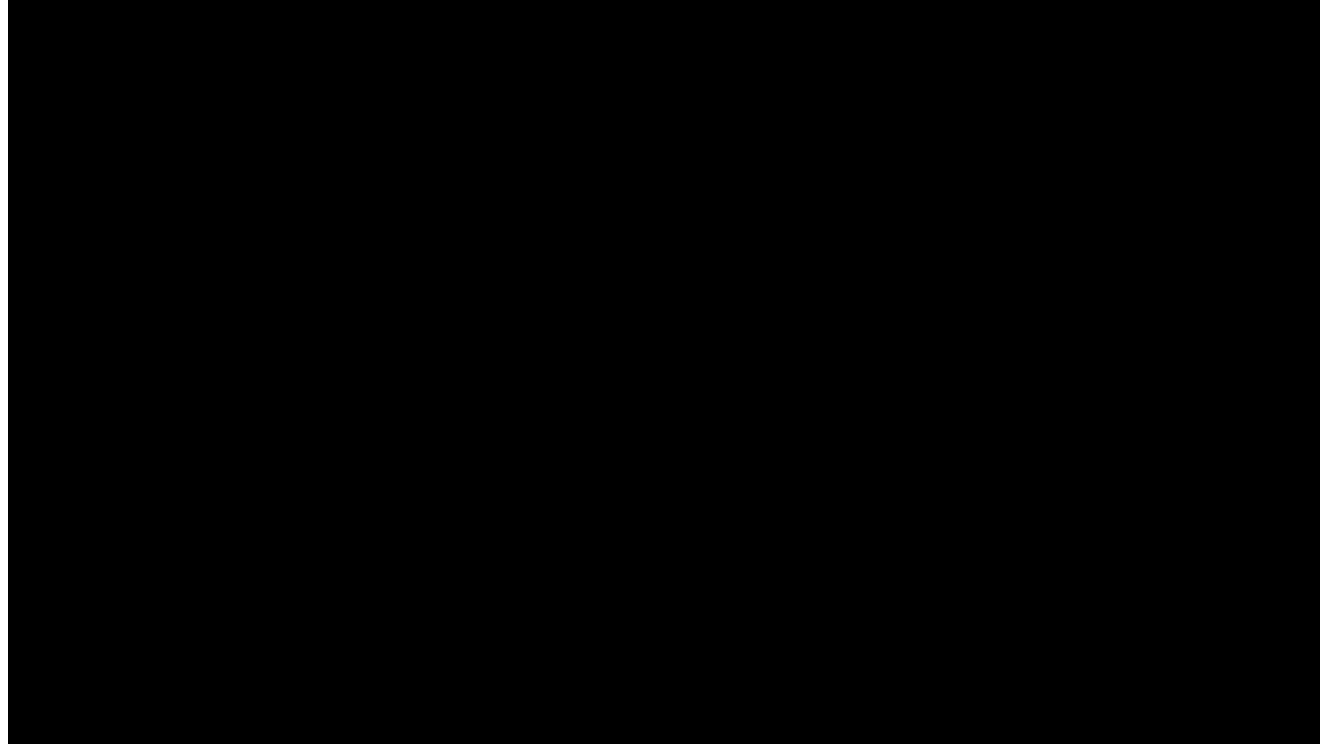
- Coil-magnetic 구조의 가장 일반적인 형태의 마이크
 - Coil의 무게의 의해 10,000 Hz이상의 음파에는 반응성이 좋지 못함
- 온도와 습기의 영향이 적어 안정적
- 외부 전력이 필요 없음



페라데이 전자석 유도(Faraday's electromagnetic induction experiment)

소리 감지 센서

- 전자기 유도

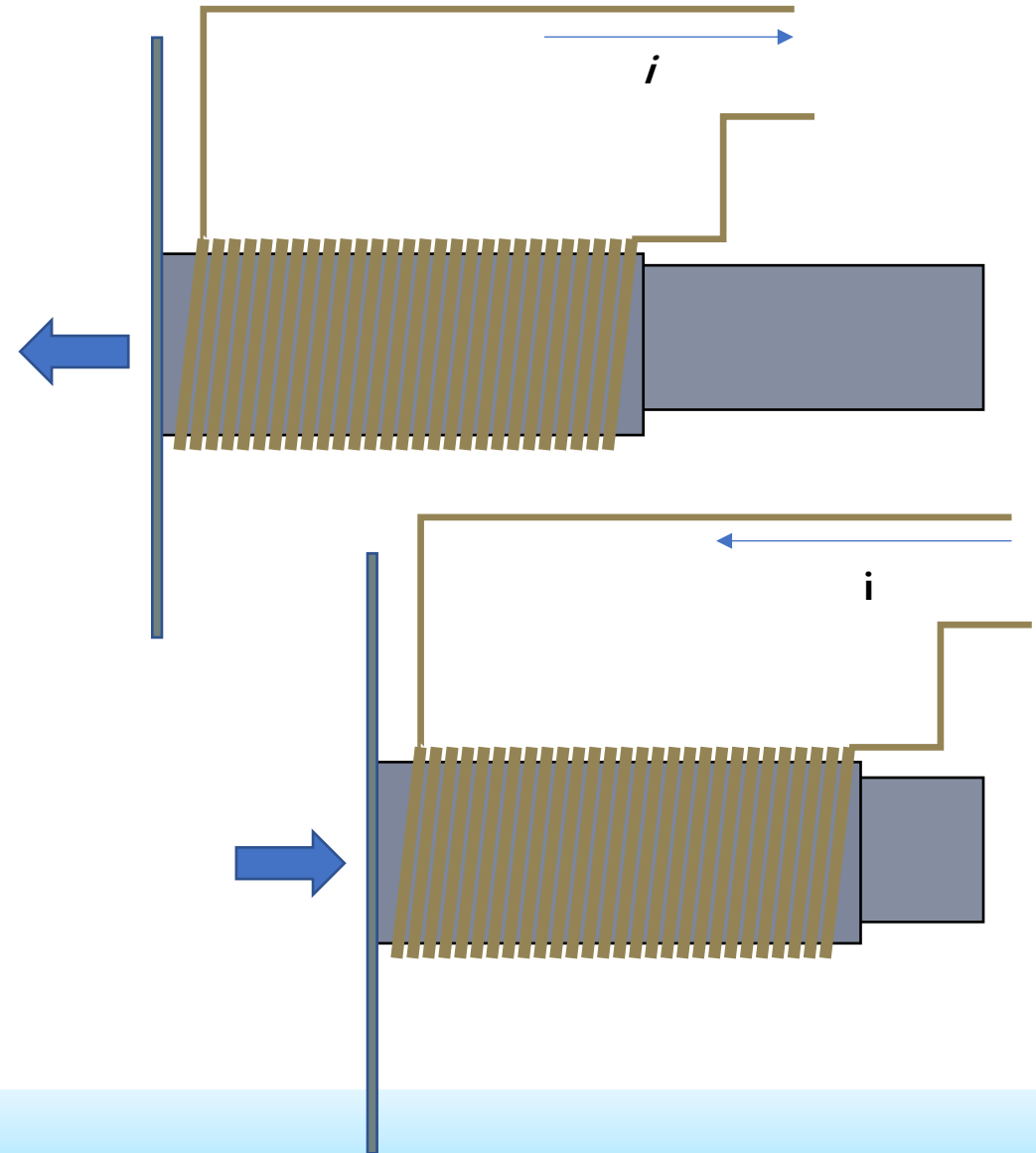
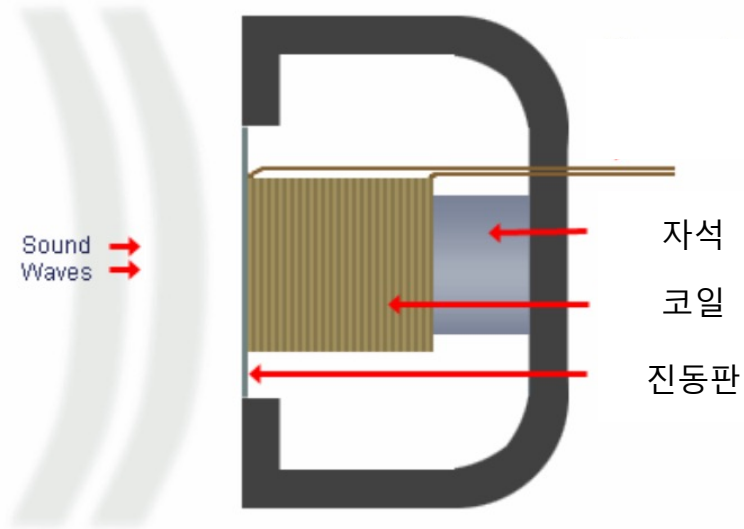


[참고]: <https://www.youtube.com/watch?v=SI1rz1rjq8>

소리 감지 센서

- **Dynamic microphone**

- 전자기 유도
 - 음파의 진동에 의해 coil 움직임
 - Coil-magnetic에 의해 유도 전류 발생



소리 감지 센서

- **Condenser microphone**

- Capacitor microphone 또는 electrostatic microphone
- 1916년에 벨 연구소에 E. C. Wente가 발명
- 축전기의 정전기 용량이 변하면 두 전극 사이에 축적된 전하가 변하는 원리 이용
 - 콘덴서 마이크의 바이어스를 위한 전원과 증폭 회로를 포함
- 장점/단점
 - 소리에 대한 감도가 높고 수음 범위가 넓은 것이 장점
 - **다진동 및 노이즈에 민감**하며 가격이 비싸고 복잡한 회로 구조를 가진 것이 단점
 - Shock mount나 pop screen등의 보조 장치가 필요



Shock mount

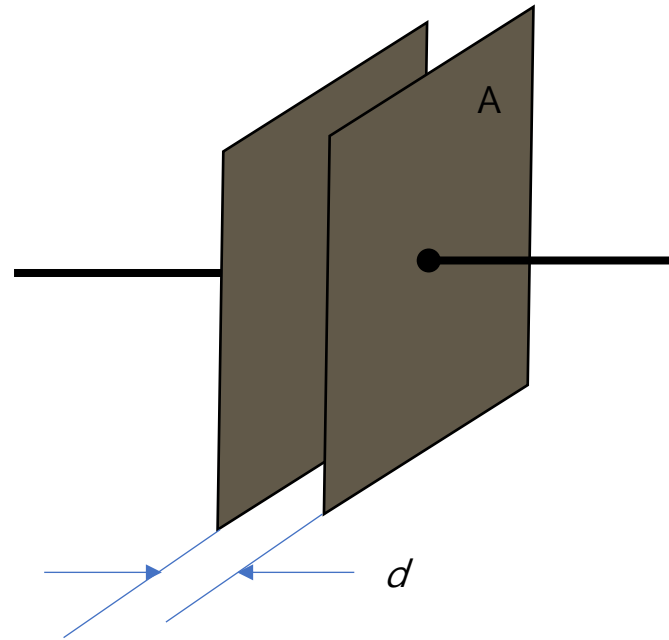
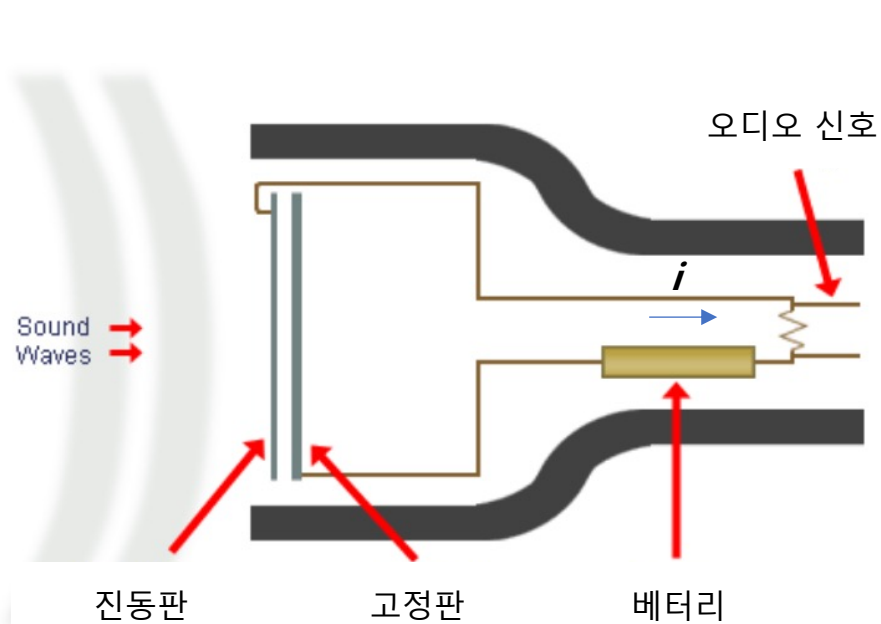


Pop screen

소리 감지 센서

• Condenser microphone

- 음파에 의해 진동판의 거리가 변화하면 콘덴서 용량이 변화
- 콘덴서 용량변화에 의해 전류가 변화
- 변화하는 전류를 높은 저항을 사용 전압 신호로 변환



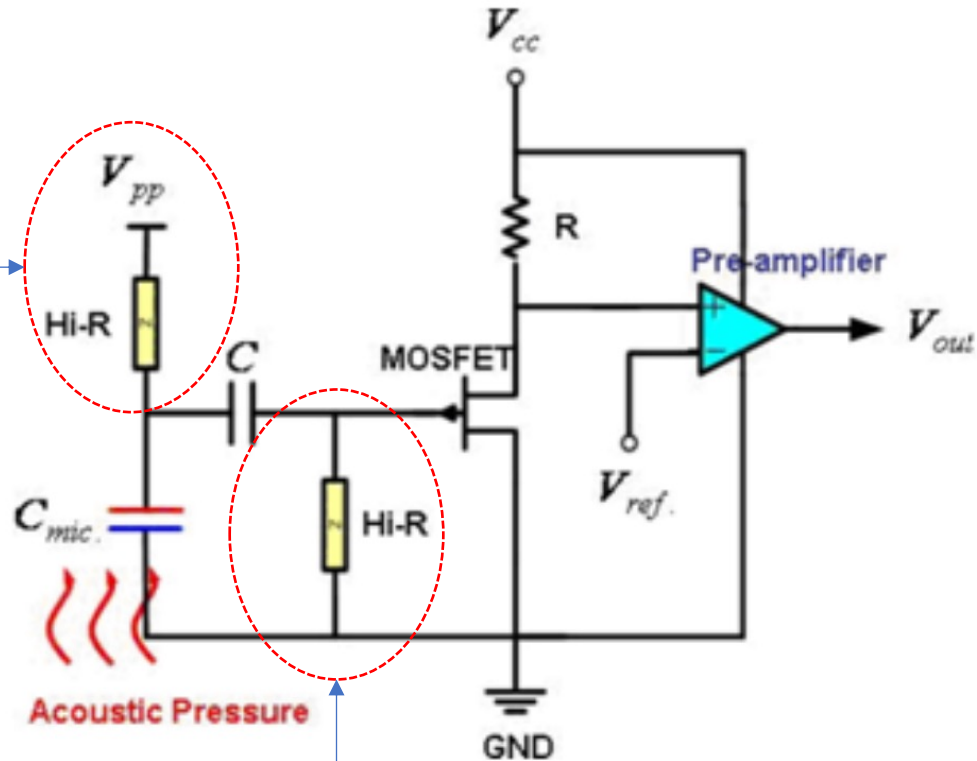
$$C = \frac{\epsilon_0 A}{d} = \frac{Q}{V}$$

C : Capacitance(F)
 Q : Charge(C)
 ϵ_0 : Permittivity
 d : Distance(m)
 A : Area(m²)

소리 감지 센서

- Condenser microphone

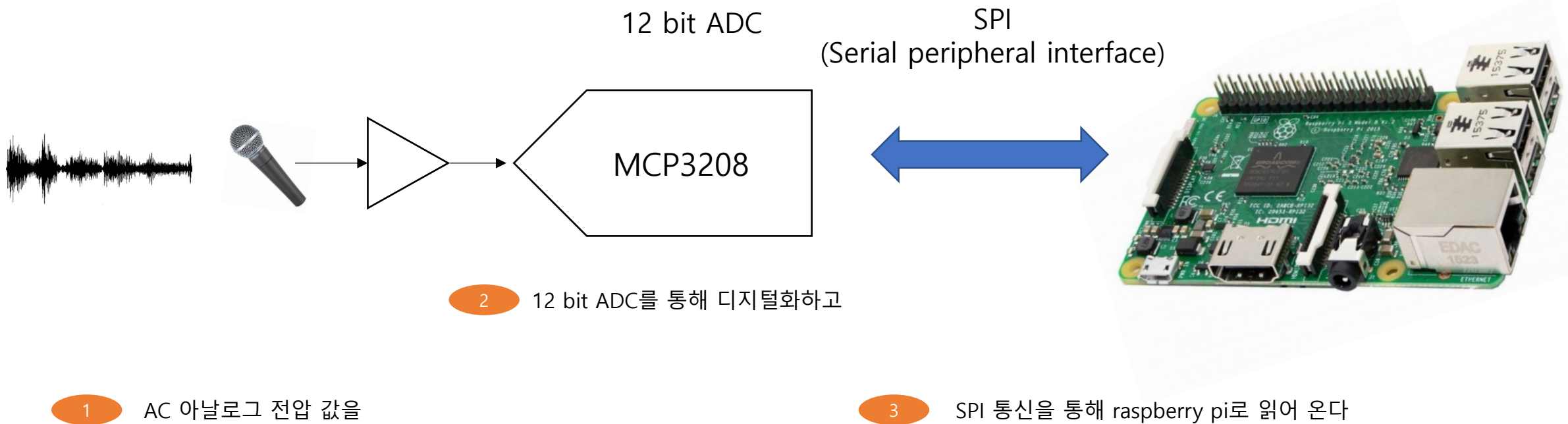
Condenser의 bias를 위한 전원 회로 필요



출력 검출을 위한 회로 필요

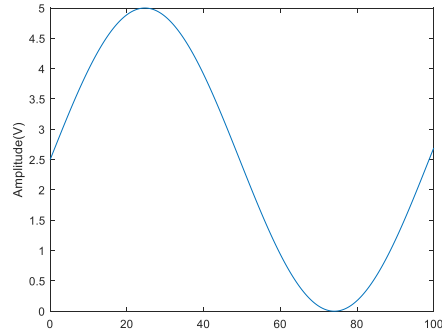
소리 감지 센서

- ADC(Analog-to-digital converting)

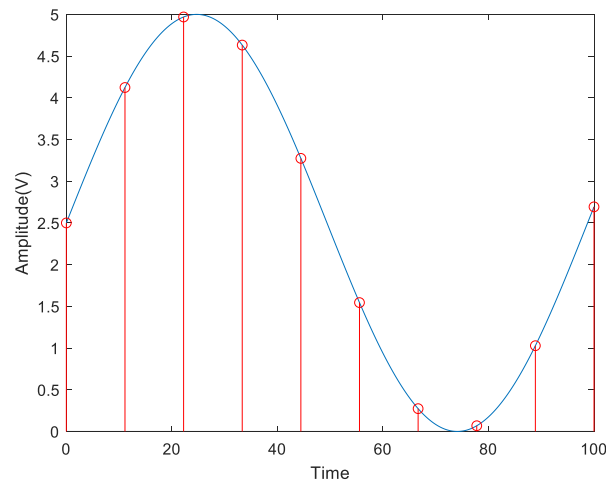


소리 감지 센서

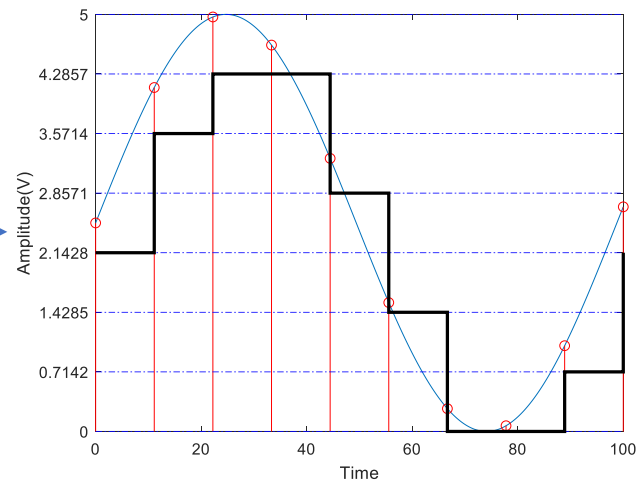
• ADC(Analog-to-digital converting) 단계



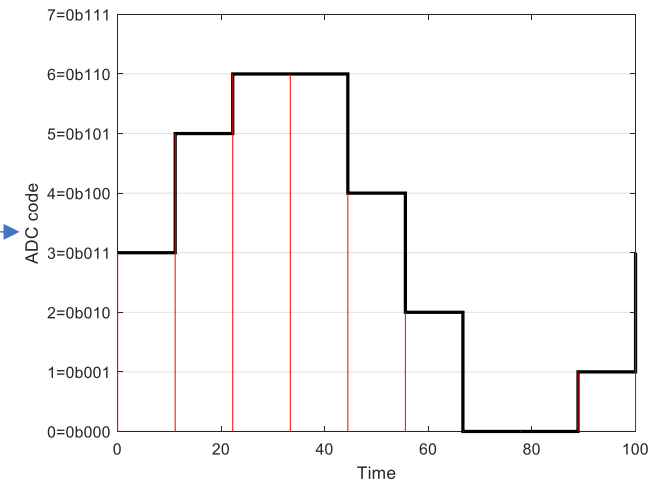
1 Sampling



2 Quantization



3 Encoding



소리 감지 센서

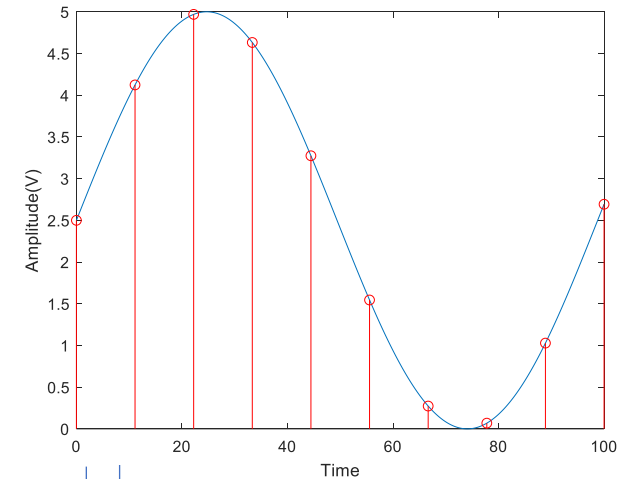
- Sampling

- Sampling frequency
 - Sampling을 수행하는 주파수

$$f_{sample} = \frac{1}{\Delta t} \text{ (Hz)}$$

- Sampling rate

- 1 초 동안 sample period 간격으로 획득한 sample의 개수
 - ADC를 위한 sampling 속도를 의미
 - n sps → n sample/sec (n: 샘플 개수)
 - Ex) 50 ksp/s = 1초당 50,000개의 sample을 획득
50 kHz의 sampling frequency로 1초 동안 50,000개의 sample을 획득

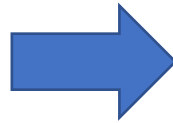
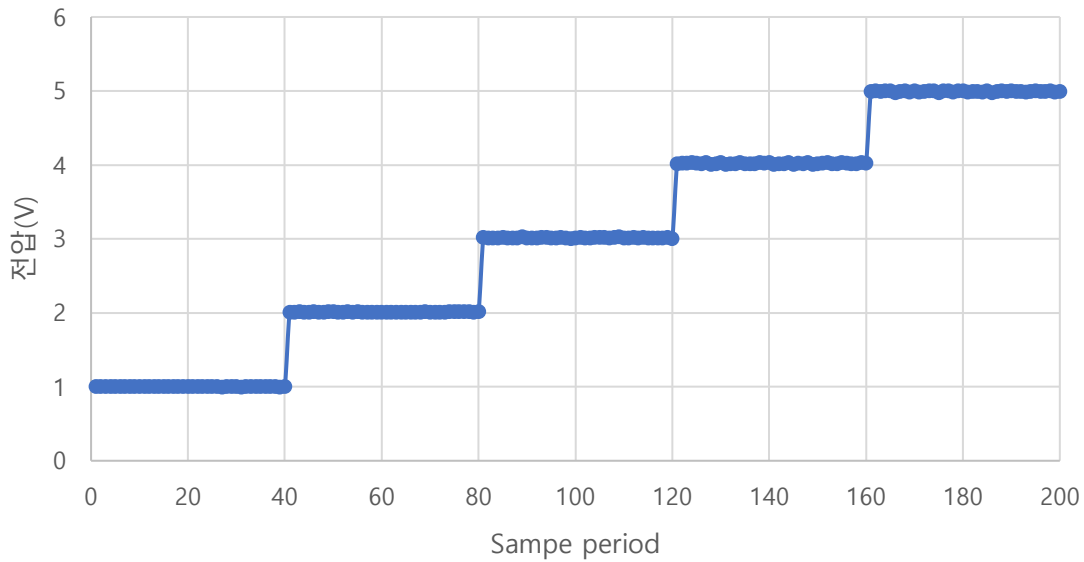


Δt → ← Δt : sample period

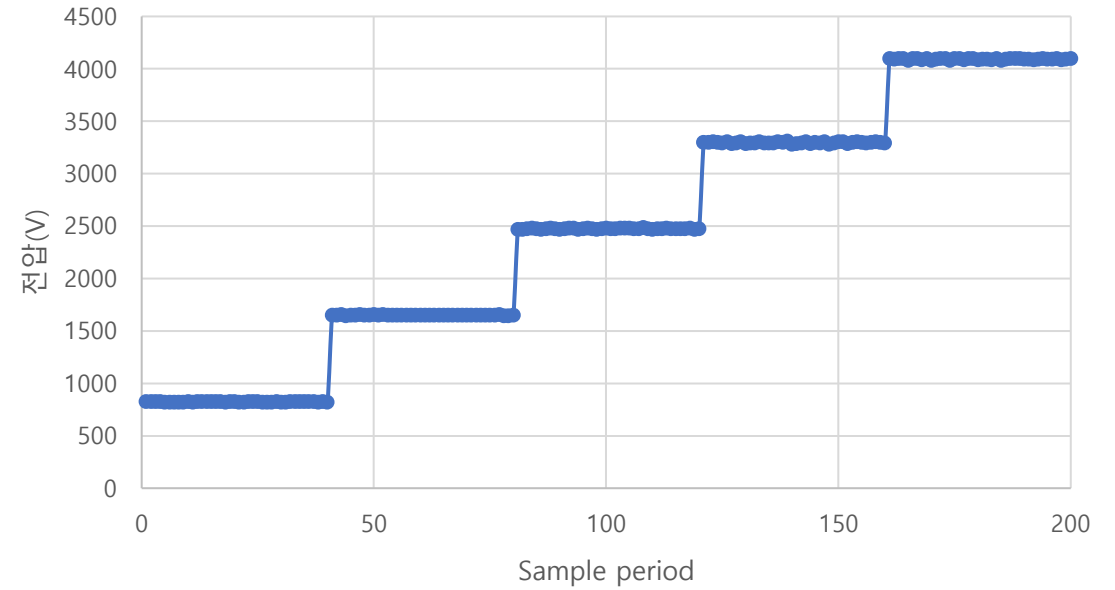
소리 감지 센서

- Sampling

Analog



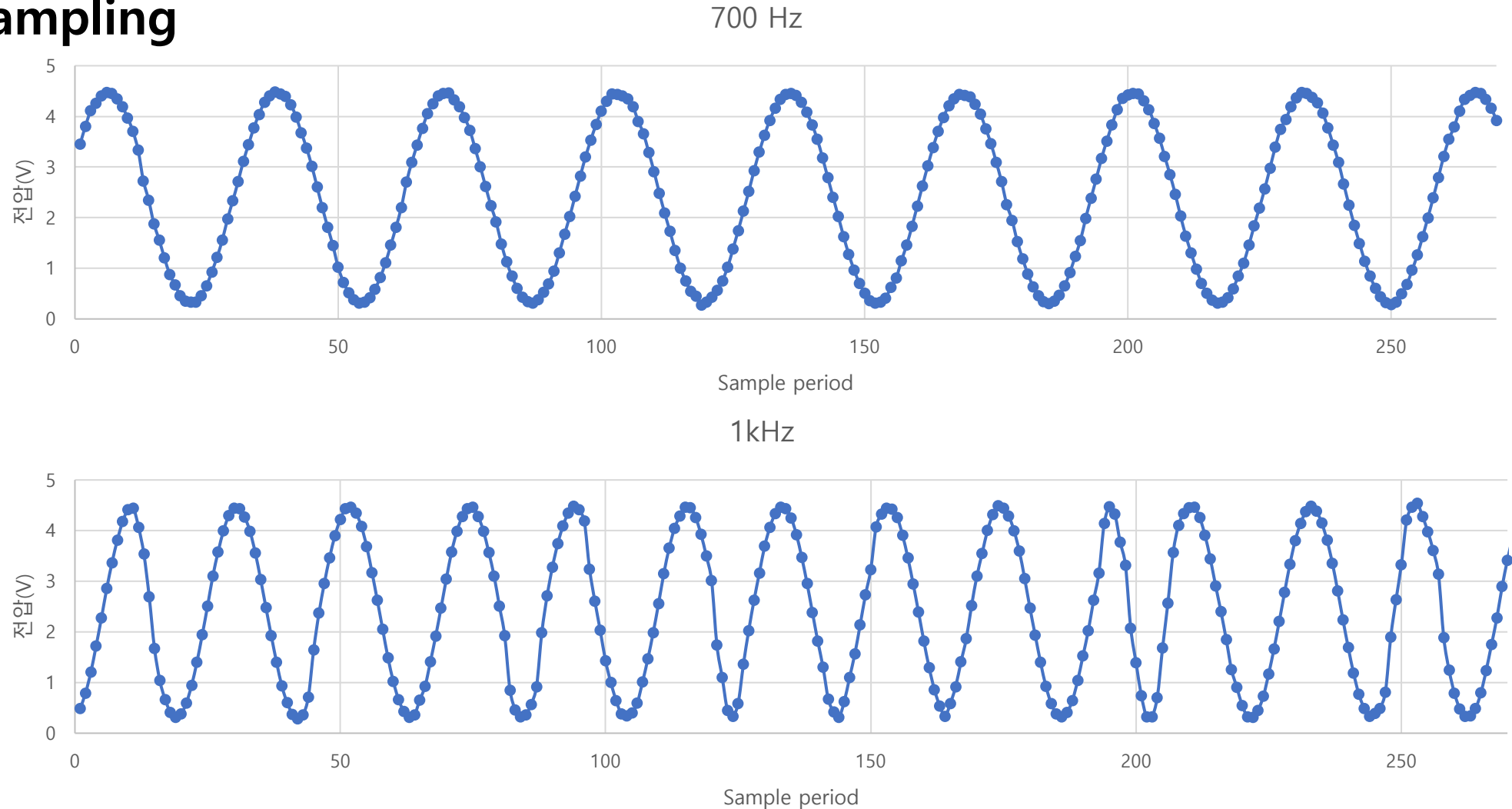
ADC



DC 또는 매우 느리게 변화하는 신호의 경우 ADC를 위한 sampling 주기에 영향을 받지 않음

소리 감지 센서

- Sampling



AC의 경우 ADC를 위한 sampling 주기에 따라 ADC 결과 영향을 받음

소리 감지 센서

- Sampling

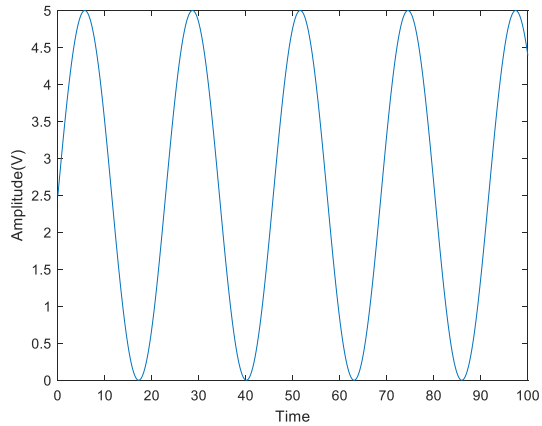
$$frequency = \frac{1}{Period}$$

주기(period)가 짧은 것은 주파수(frequency)가 높은 것

주기(period)가 긴 것은 주파수(frequency)가 낮은 것

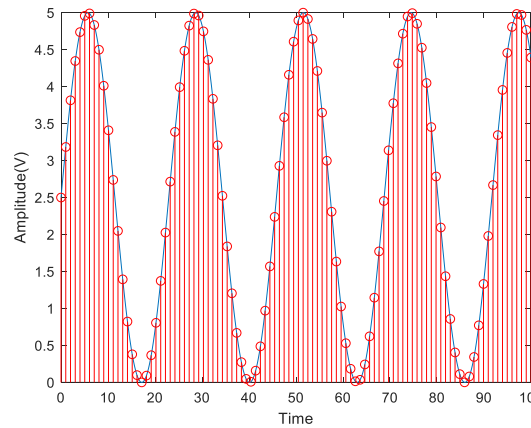
소리 감지 센서

• Sampling

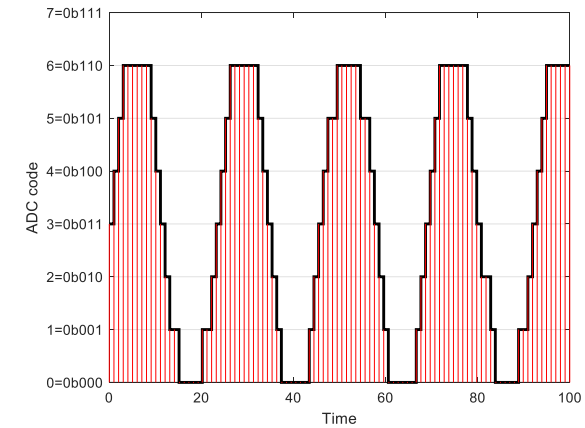


$$\Delta t \ll t_s$$

Sampling



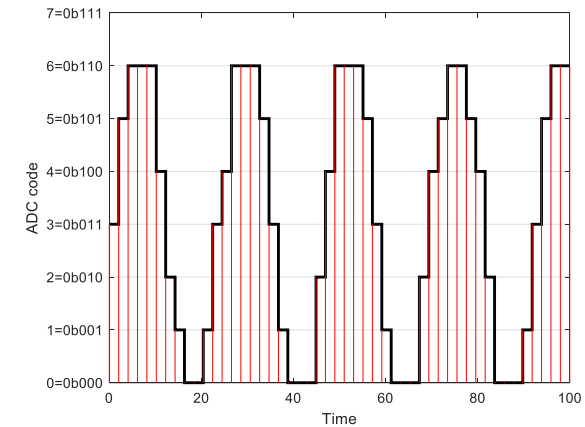
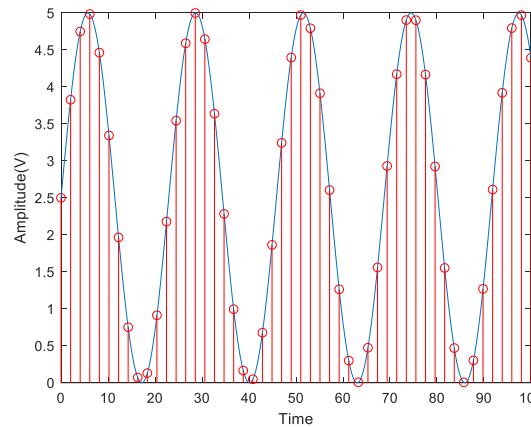
ADC



Sampling 주기(Δt) 가 신호의 주기(t_s)보다 매우 작은 경우

Δt : Sampling 주기
 t_s : 신호 주기
 $t_s = 1/f$

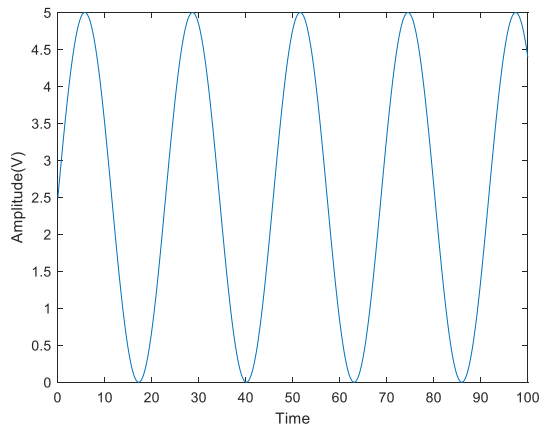
$$\Delta t < t_s$$



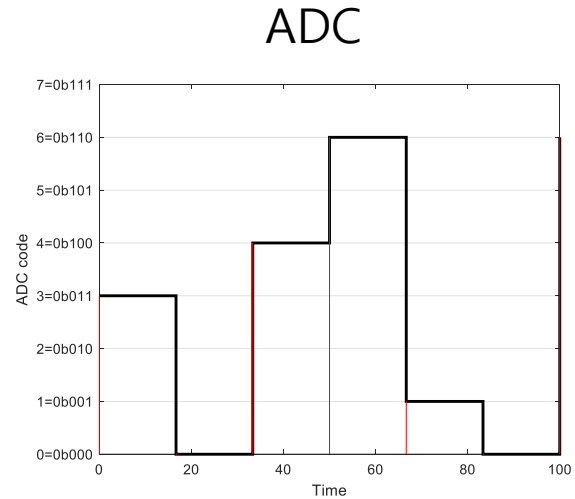
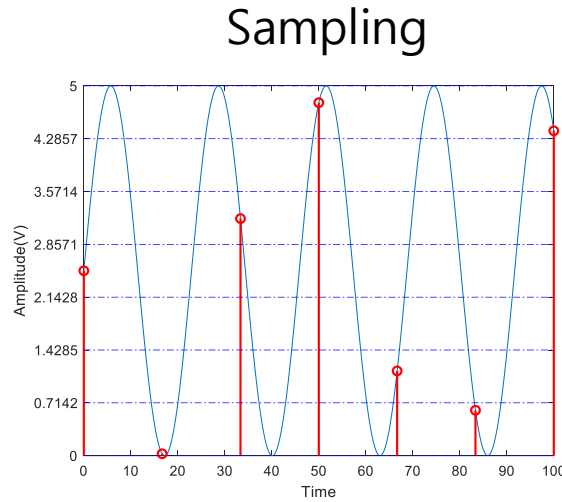
Sampling 주기(Δt) 가 신호의 주기(t_s)보다 작은 경우

소리 감지 센서

• Sampling



$$\Delta t \geq t_s$$



Sampling 주기(Δt) 가 신호의 주기(t_s)보다 큰 경우

Δt : Sampling 주기

t_s : 신호 주기

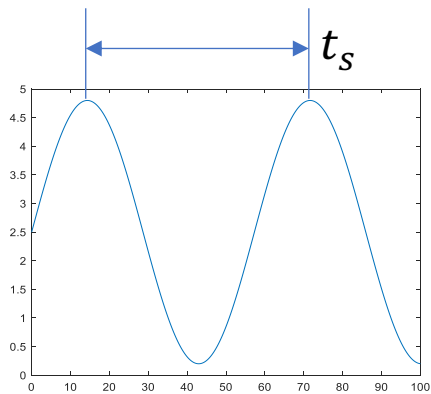
$$t_s = 1/f$$

ADC된 결과로 부터 Sampling 주기(Δt) 는 신호의 주기(t_s)보다 매우 작아야 됨

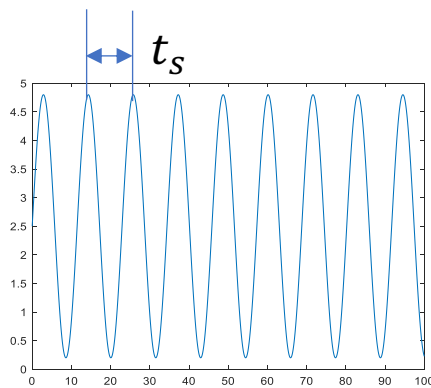
ADC된 결과로 부터 Sampling 주파수(f_{sample})는 신호의 주파수(f)보다 높아야 됨

소리 감지 센서

- Sampling

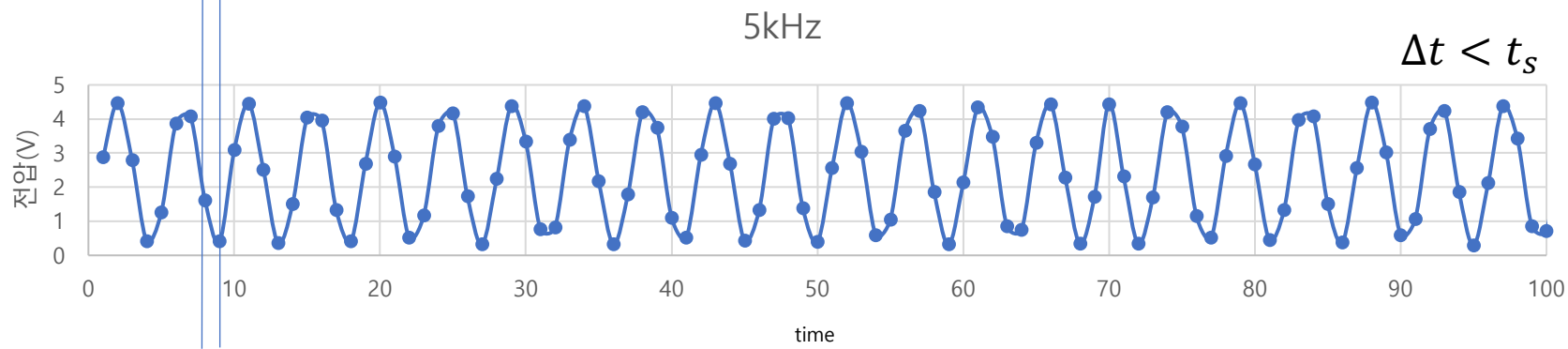
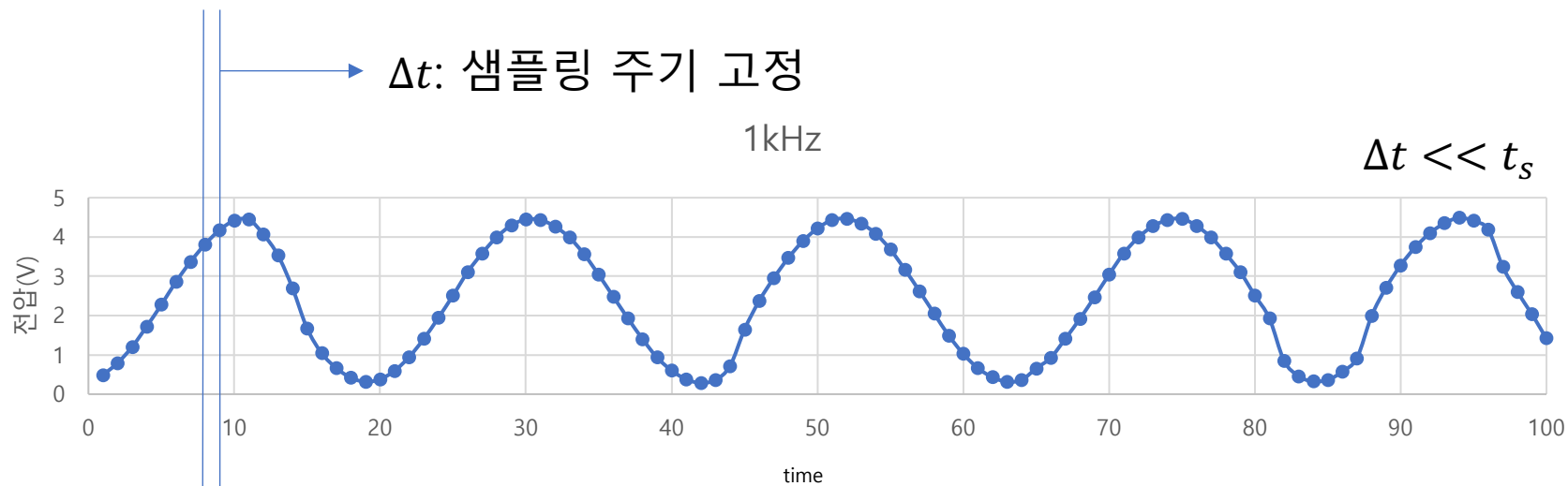


ADC → DAC



Analog 신호

ADC → DAC

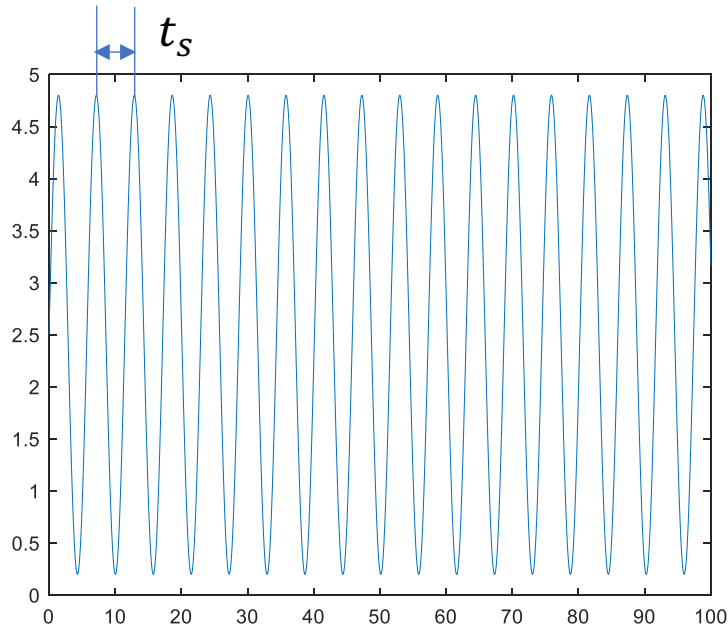


AD 변환된 신호

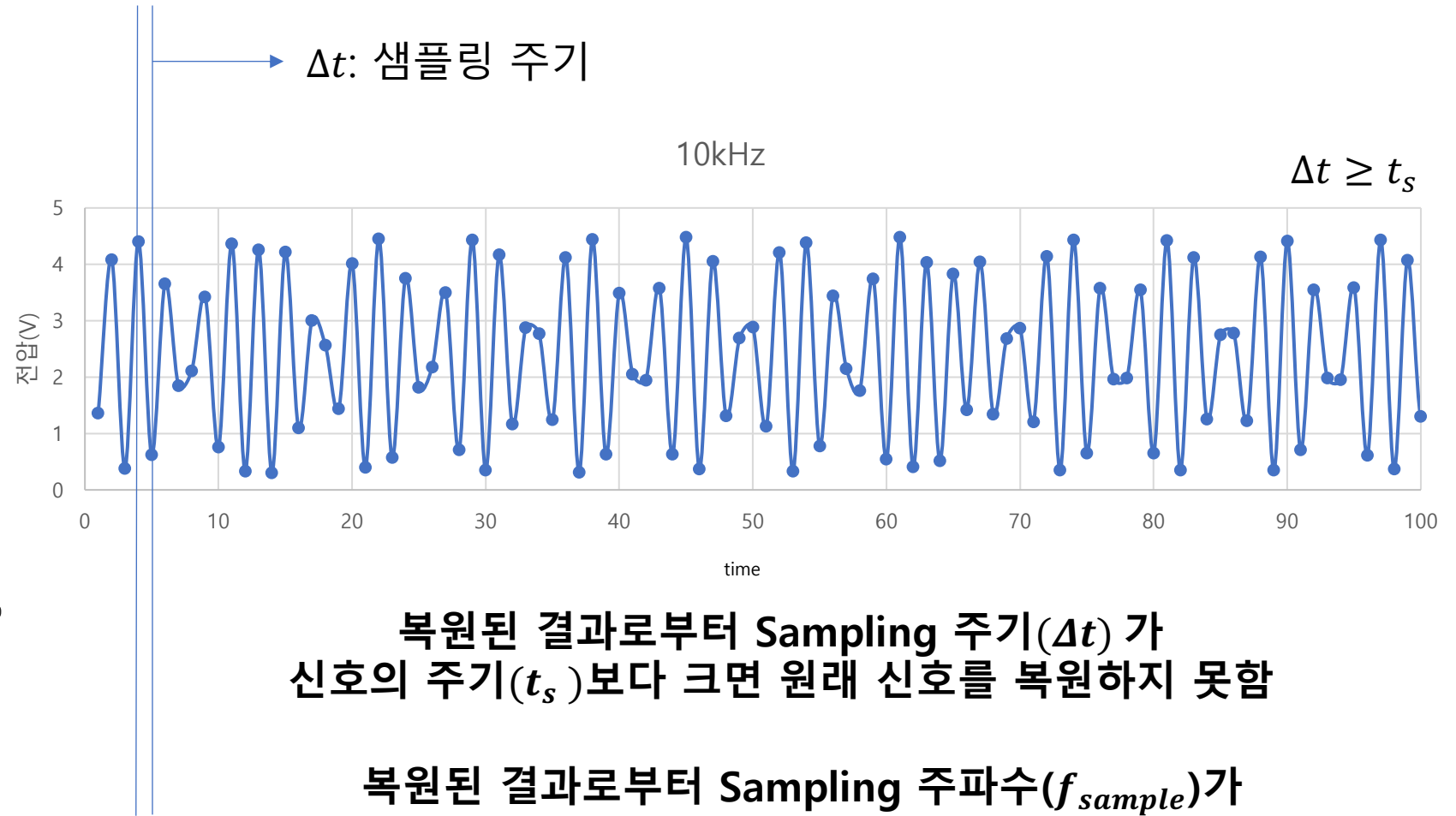
소리 감지 센서

- Sampling

ADC → DAC



Analog 신호



복원된 결과로부터 Sampling 주기(Δt) 가
신호의 주기(t_s)보다 크면 원래 신호를 복원하지 못함

복원된 결과로부터 Sampling 주파수(f_{sample})가
신호 주파수(f)보다 낮으면 원래 신호를 복원하지 못함

소리 감지 센서

- **Sampling 이론**

- ADC를 DAC를 통해 복원한 결과로부터 Sampling 주기가
 - 신호의 주기보다 크면 원래 신호를 복원하지 못함

Nyquist-Shannon sampling theorem

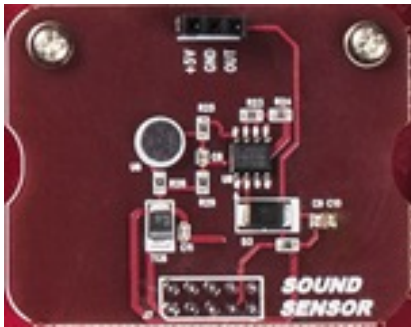
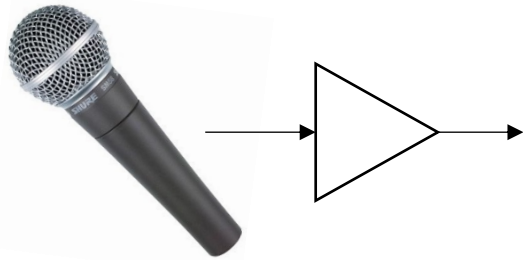
Sampling 주파수는 신호의 주파수 보다 두배 이상이 되어야 한다.

ADC하려는 신호의 가장 높은 주파수보다
2배 이상의 sampling 주파수로 sampling해야만 원래의 신호를 복원 가능

- Ex) 15,000Hz의 목소리를 ADC 하기위해 필요한 sampling 주파수는?
 $15,000 \times 2 = 30,000 \text{ Hz}$

소리 감지 센서

- ADC(Analog-to-digital converting)



1 AC 아날로그 전압 값을

12 bit ADC

MCP3208

2 12 bit ADC를 통해 디지털화하고

SPI
(Serial peripheral interface)

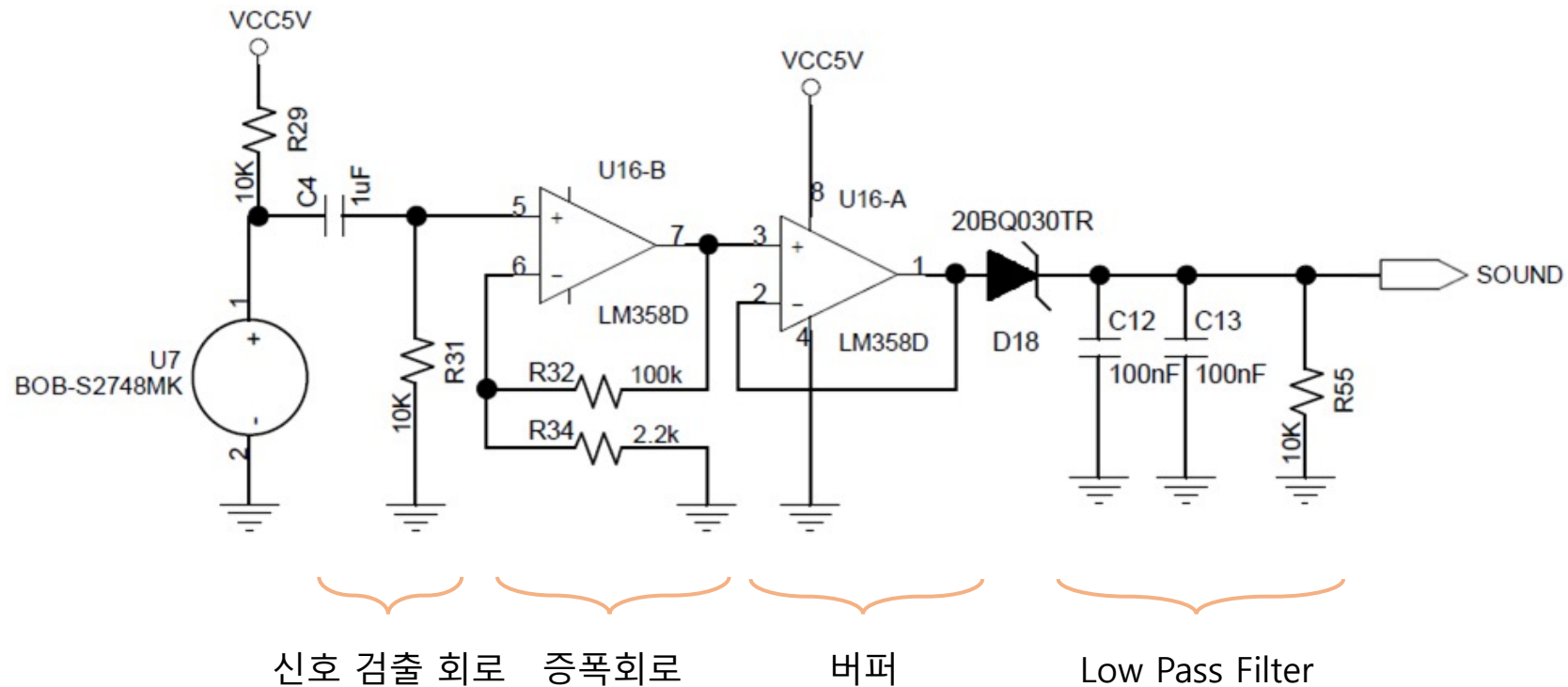


3 SPI 통신을 통해 raspberry pi로 읽어 온다

소리 감지 센서

• 회로도

사운드 센서 외형	모듈 항목	모듈 항목의 내용
	사운드 센서	Microphone
	동작 전압	5V
	I/O Interface	1 Analog OUTPUT
소리를 감지하는 센서		



소리 감지 센서

• Raspberry Pi ADC SPI 설정

- Raspberry Pi는 2개의 SPI chip selection 제공
 - 실습에서는 ADC를 SPI chip selection 0에 연결
 - SPI chip selection 0은 GPIO_8에 할당

BCM (Raspberry Pi)	wPi (Wiring Pi)	ADC 연결	설명
GPIO_8	10	ADC_CS	ADC chip selection
GPIO_9	13	ADC_MISO	ADC MISO
GPIO_10	12	ADC_MOSI	ADC MOSI
GPIO_11	14	ADC_SCK	ADC clock

ADC 연결	Sound Sensor
ADC_CH0	OUT

소리 감지 센서

• 실습 1

- MCP3208 ADC를 사용하여 소리 감지 센서의 출력 측정
 - SPI channel:0
 - SPI interface clock speed: 500 kHz
 - ADC channel: 0 channel
 - Reference voltage: 5V
 - Bit Resolution: 12 bit
 - **소리 감지 센서의 출력을 ADC 후 200msec 대기 후 ADC 수행**
- MCP3208 사용하여 소리 감지 센서 출력을 ADC를 하기 위한 제어 함수 작성
 - SPI를 통해서 ADC 변환을 제어하고 ADC된 값을 반환하는 함수
 - **가변 저항 강의 실습 1에서 구현된 코드 사용**

int32_t ADCRead(int SPI_CH, uint32_t ADC_CH)

소리 감지 센서

- 실습 1

- SPI를 통해서 ADC 변환을 제어하고 ADC된 값을 반환하는 함수

int32_t ADCRead(int SPI_CH, uint32_t ADC_CH)

입력:

- SPI_CH: Raspberry Pi SPI channel [0:1]
- ADC_CH: MCP3208 ADC channel 선택 [0:7]

리턴:

- SPI로 부터 읽어 들인 ADC 결과 값

ADC(Analog-to-digital converting)

- 실습 1

소리 감지 센서 전압 출력 범위: $0 \leq V_{sound} \leq 5V$

$$ADC\ Value = \frac{V_{sound}}{Resoultion}$$

$$\begin{aligned} Resoultion &= \frac{V_{REF}}{(2^n - 1)} \\ &= \frac{5V}{(2^{12} - 1)} \\ &\cong 0.001221\ V/bit \end{aligned}$$

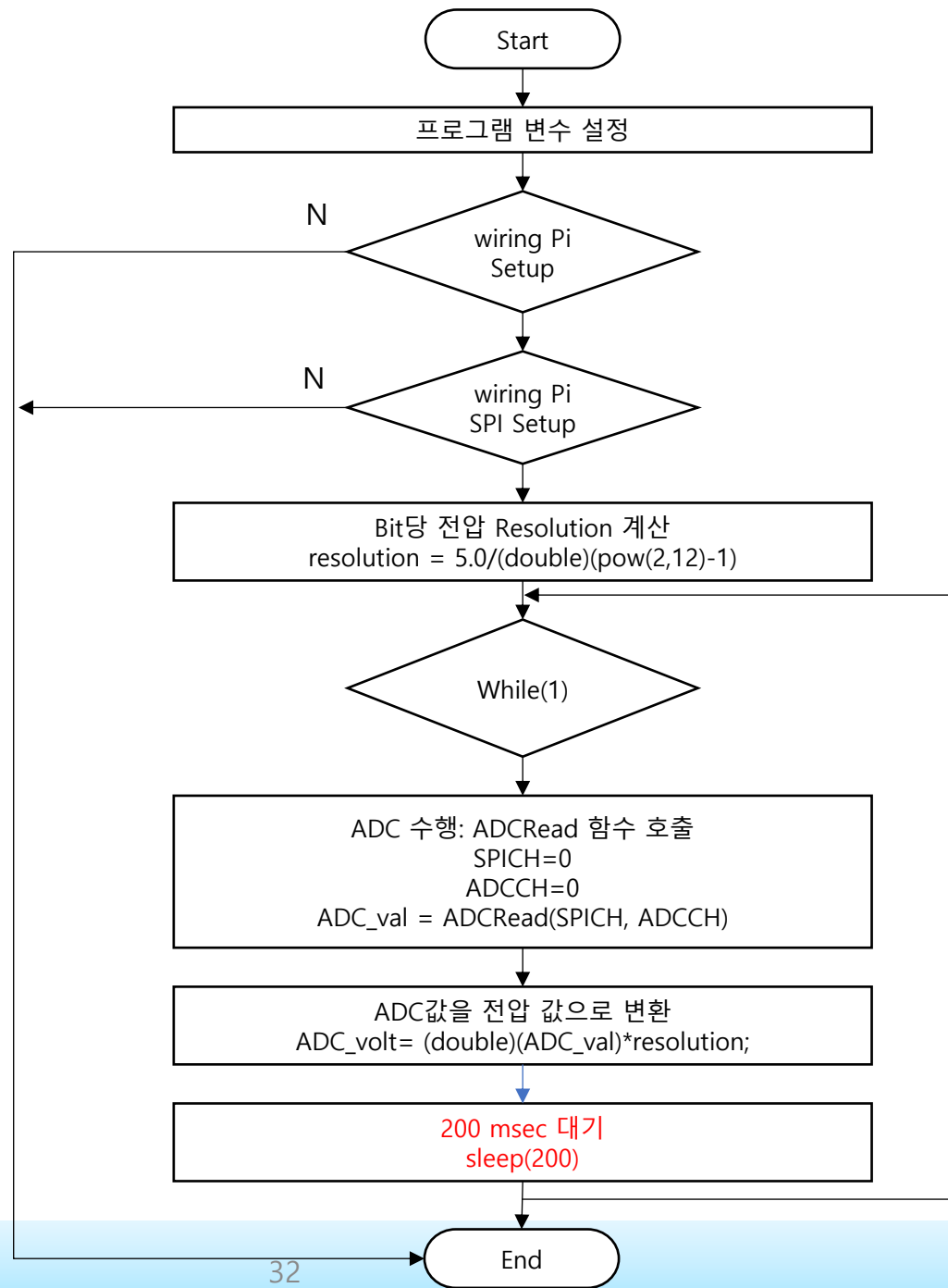
$$V_{sound}(V) = ADC\ Value \times Resoultion$$



V_{REF} : ADC reference voltage
 n : ADC bit resoultion

소리 감지 센서

• 실습 1



소리 감지 센서

• 실습 1

```
#include <stdint.h>
#include <stdio.h>
#include <wiringPi.h>    // wiringPi
#include <wiringPiSPI.h> // SPI
#include <math.h>

#define SPICH          0 //SPI Channel 0
#define ADCCH          0 //ADC Channel 0
#define SPI_SCK        500000 //SPI Clock 500kHz

#define START_BIT      1 // Start bit 1
#define SLG_DIFF_BIT   1 // SLG/DIFF bit

int32_t ADCRead(int SPI_CH, uint32_t ADC_CH);    // SPI를 사용한 ADC 함수 선언
```

소리 감지 센서

• 실습 1

```
int main(void)
{
    uint32_t ADC_val;           // ADC 값
    double ADC_volt;           // ADC 값을 전압으로 변환한 값
    double resolution;         // Bit 당 해상도

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

    if(wiringPiSPISetup(SPICH, SPI_SCK)==-1)
    {
        printf("SPI set-up 실패\n");
        return 1;
    }

    // Bit당 resolution 구하기
    resolution = 5.0/(double)(pow(2,12)-1);

    while(1)
    {
        ADC_val = ADCRead(SPICH,ADCCH); // SPI를 통해서 ADC 값 읽어 오기
        ADC_volt= (double)(ADC_val)*resolution;
        printf("ADC value:%d\n", ADC_val);
        printf("소리 전압(V):%f\n", ADC_volt);
        delay(200);
    }
}
```

소리 감지 센서

- 실습 1

```
// ADC값을 SPI를 통해 읽어오기
int32_t ADCRead(int SPI_CH, uint32_t ADC_CH)
{
    uint8_t buf[3];
    int32_t adcValue = 0;

    buf[0] = (START_BIT<<2)|(SLG_DIFF_BIT<<1)|(ADCCH>>2);
    buf[1] = (ADC_CH&3)<<6;
    buf[2] = 0;

    wiringPiSPIDataRW(SPI_CH, buf, 3);

    adcValue = ((buf[1]&0xF)<<8)|buf[2];

    return adcValue;
}
```

소리 감지 센서

<파일명>

sound_sensor.c

<Compile 명령>

gcc sound_sensor.c -o sound_sensor -lwiringPi -lm

<실행>

sudo ./sound_sensor

ADC value:62	소리 전압(V):0.075702
ADC value:83	소리 전압(V):0.101343
ADC value:313	소리 전압(V):0.382173
ADC value:147	소리 전압(V):0.179487
ADC value:31	소리 전압(V):0.037851
ADC value:1309	소리 전압(V):1.598291
ADC value:255	소리 전압(V):0.311355
ADC value:69	소리 전압(V):0.084249
ADC value:1196	소리 전압(V):1.460317
ADC value:184	소리 전압(V):0.224664
ADC value:57	소리 전압(V):0.069597
ADC value:71	소리 전압(V):0.086691
ADC value:58	소리 전압(V):0.070818
ADC value:74	소리 전압(V):0.090354
ADC value:47	소리 전압(V):0.057387
ADC value:66	소리 전압(V):0.080586
ADC value:63	소리 전압(V):0.076923

소리 감지



주변 노이즈

소리 감지 센서

• 실습 2

- 소리 검출 센서의 출력을 ADC하여 소리 크기에 따라 시각적으로 레벨 표시
 - 시각적 레벨 표시는 아래의 ADC 전압 LEVEL에 따라 LED를 사용하여 표시
 - 시각적 레벨 표시를 위한 각각의 LED는 turn on시 10 msec 동안 유지

소리 검출 시각적 레벨	LED 표시	소리 검출 센서 전압 값 범위
Level 1	<ul style="list-style-type: none">• LED RED on	$0.2\text{ V} \leq V_{\text{sound}} < 0.5\text{ V}$
Level 2	<ul style="list-style-type: none">• LED RED on• LED GREEN on	$0.5\text{ V} \leq V_{\text{sound}} < 0.8\text{ V}$
Level 3	<ul style="list-style-type: none">• LED RED on• LED GREEN on• LED BLUE on	$0.8\text{ V} \leq V_{\text{sound}}$

소리 감지 센서

- 실습 2

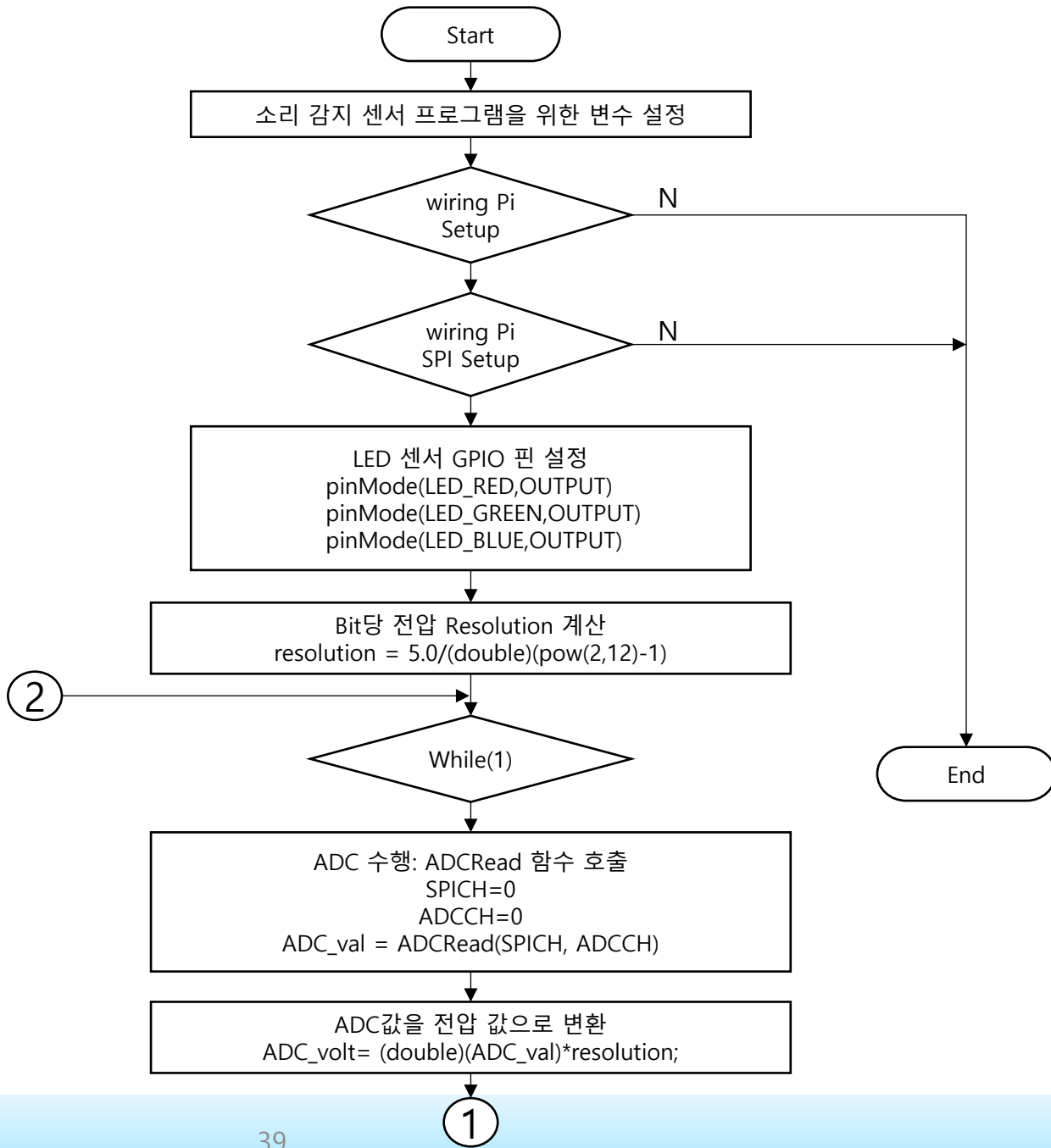
- Raspberry Pi LED 설정

- 소리 크기에 따라 시각적으로 LED on/off 하기 위한 LED 연결 설정

BCM (Raspberry Pi)	wPi (Wiring Pi)	LED 모듈 핀 정보
GPIO_4	7	LED_RED
GPIO_5	21	LED_GREEN
GPIO_6	22	LED_BLUE

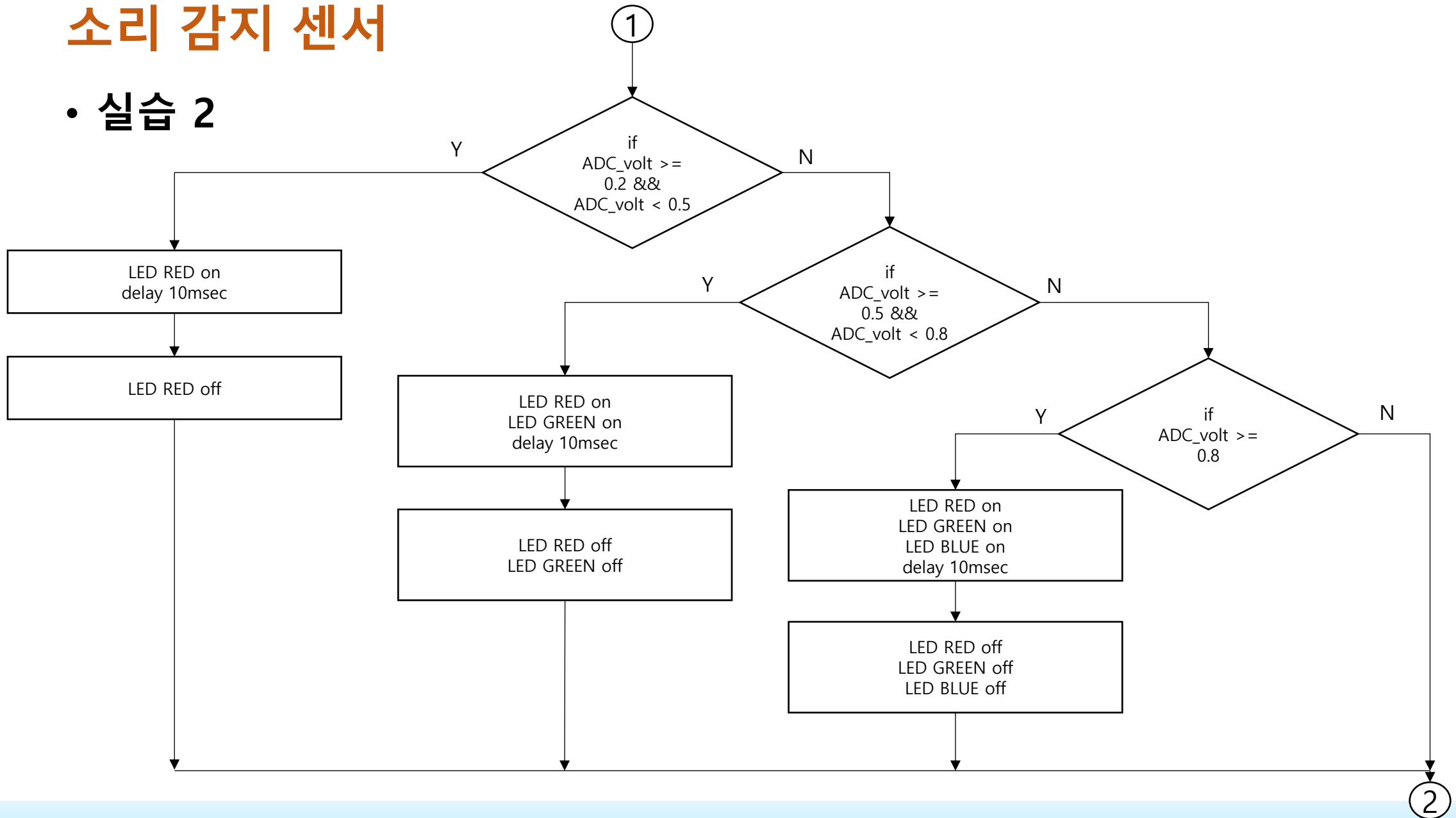
소리 감지 센서

• 실습 2



소리 감지 센서

• 실습 2



소리 감지 센서

• 실습 2

```
#include <stdint.h>
#include <stdio.h>
#include <wiringPi.h>    // wiringPi
#include <wiringPiSPI.h> // SPI
#include <time.h>
#include <math.h>

#define SPICH          0          //SPI Channel 0
#define ADCCH          0          //ADC Channel 0
#define SPI_SCK 500000          //SPI Clock 500kHz

#define START_BIT      1          // Start bit 1
#define SLG_DIFF_BIT 1          // SLG/DIFF bit

#define LED_RED 7          // GPIO_4
#define LED_GREEN 21      // GPIO_5
#define LED_BLUE 22       // GPIO_6

#define LED_ON_DUR      10 // LED turn on duration msec

int32_t ADCRead(int SPI_CH, uint32_t ADC_CH);    // SPI를 사용한 ADC
```

소리 감지 센서

• 실습 2

```
int main(void)
{
    uint32_t ADC_val;          // ADC 값
    double ADC_volt;           // ADC 값을 전압으로 변환한 값
    double resolution; // Bit 당 해상도

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

    if(wiringPiSPISetup(SPICH, 1000000)==-1)
    {
        printf("SPI set-up 실패\n");
        return 1;
    }

    // GPIO pin output 설정
    pinMode(LED_RED,OUTPUT);
    pinMode(LED_GREEN,OUTPUT);
    pinMode(LED_BLUE,OUTPUT);

    // Bit당 resolution 구하기
    resolution = 5.0/(double)(pow(2,12)-1);

    while(1)
    {
        ADC_val = ADCRead(SPICH,ADCCH); // SPI를 통해서 ADC 값 읽어 오기
        ADC_volt= (double)(ADC_val)*resolution;
    }
}
```

소리 감지 센서

• 실습 2

```
if(ADC_volt < 0.5 && ADC_volt >= 0.2)
{
    printf("Level 1Wn");
    digitalWrite(LED_RED, HIGH);
    delay(LED_ON_DUR);
    digitalWrite(LED_RED, LOW);
}
else if(ADC_volt < 0.8 && ADC_volt >= 0.5)
{
    printf("Level 2Wn");
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_GREEN, HIGH);
    delay(LED_ON_DUR);
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_GREEN, LOW);
}
else if(ADC_volt >= 0.8)
{
    printf("Level 3Wn");
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(LED_BLUE, HIGH);
    delay(LED_ON_DUR);
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_BLUE, LOW);
}
}
```

소리 감지 센서

• 실습 2

<파일명>

sound_sensor_LED.c

<Compile 명령>

gcc sound_sensor_LED.c -o sound_sensor_LED -lwiringPi -lm

<실행>

sudo ./sound_sensor_LED

```
// ADC값을 SPI를 통해 읽어오기
int32_t ADCRead(int SPI_CH, uint32_t ADC_CH)
{
    uint8_t buf[3];
    int32_t adcValue = 0;

    buf[0] = (START_BIT<<2)|(SLG_DIFF_BIT<<1)|(ADCCH>>2);
    buf[1] = (ADC_CH&3)<<6;
    buf[2] = 0;

    wiringPiSPIDataRW(SPI_CH, buf, 3);

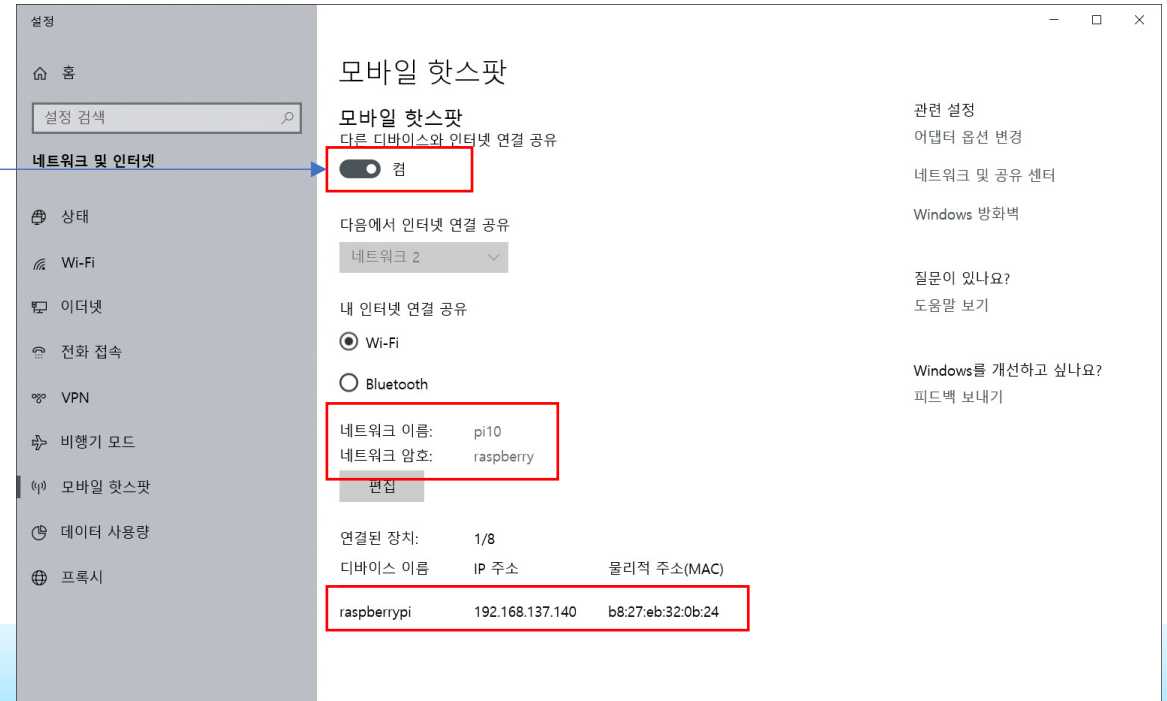
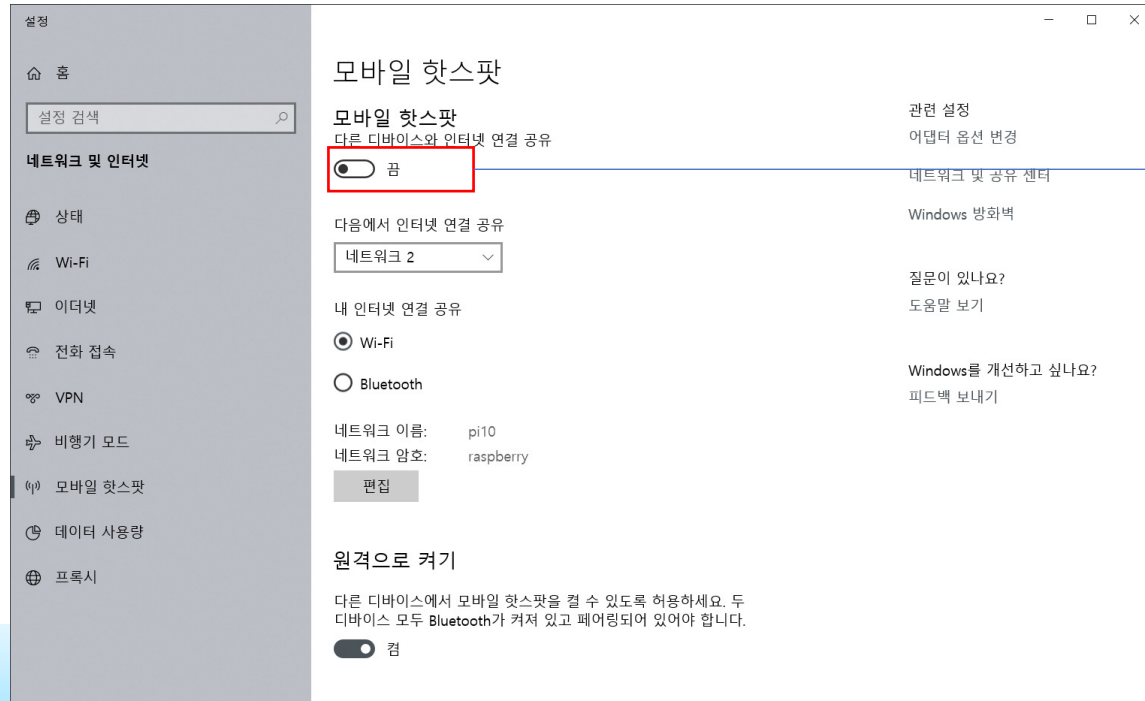
    adcValue = ((buf[1]&0xF)<<8)|buf[2];

    return adcValue;
}
```

참고 사항

• 수업 전 확인사항

- 무선랜 카드를 PC에 설치 및 SD card를 Raspberry Pi에 삽입
- PC의 모바일 핫스팟을 **컴**으로 설정
 - Raspberry Pi의 전원을 켜
- 네트워크 이름 및 네트워크 암호 설정 확인
- 연결된 장치의 IP 주소 확인



참고 사항

- Raspberry Pi 끝 때

sudo shutdown -t now



Superuser 권한으로 명하니 Raspberry Pi를 꺼라 언제? 지금 당장!