

초음파 센서: 거리측정

김 동 훈

시작

• 강의 소개

- 이번 강의는 지난주에 학습했던 거리 측정에서 평균과 표준편차에 대한 프로그램 실습내용을 학습하게 됩니다.
- 먼저 측정에서 사용되는 평균과 표준편차를 구하는 수식을 이해하고 이를 프로그램 실습을 통해 어떻게 구현되는지 학습합니다.
- 사전에 프로그래밍 되어진 평균과 표준편차를 사용하여 Raspberry Pi와 초음파 센서를 사용하여 획득된 데이터를 이용해 거리 측정 시 거리 측정의 평균과 표준 편차를 구하는 프로그램 실습을 진행 합니다.
- 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 진행 하시기 바랍니다.
 - 코드 리딩에 필요한 주석은 프로그램에 달려 있으니 꼼꼼히 확인하시기 바랍니다.

거리 측정

• 거리 측정 오차

- 거리 측정에는 다양한 원인에 의해 발생하는 측정 오차를 포함
 - 측정의 불확실성을 포함
- 측정 오차를 제공 할 수 있는 원인들
 - 센서 오차
 - 센서의 정확도
 - 센서의 view angle 등
 - 환경적 오차
 - 온도, 습도, 그리고 날씨에 의한 오차
 - 측정 환경 주변의 재질
 - 목표물 오차
 - 목표물의 모양 및 구성 재질에 의한 오차
 - 측정 방법 오차
 - 측정의 횟수
 - 측정 도구의 설치 오차
 - 측정을 실시하는 사람에 기인한 오차

거리 측정

• 거리 측정

- 거리 측정에서 측정 오차를 특정 한다는 것은 매우 어려운 일
- 거리 측정의 값을 **평균**과 **표준 편차**를 사용하여 표현 가능
 - 다수의 거리 측정 샘플을 획득하여 평균과 표준 편차를 구함

• 평균

- 다수의 측정 샘플을 획득 한 후 산술적 평균을 구하여 하나의 대표 값 산출
- 측정 오차를 구하여 측정의 정확도(accuracy)를 판단하기 위해 사용
- 측정 노이즈를 제거하기 위한 방법으로 사용

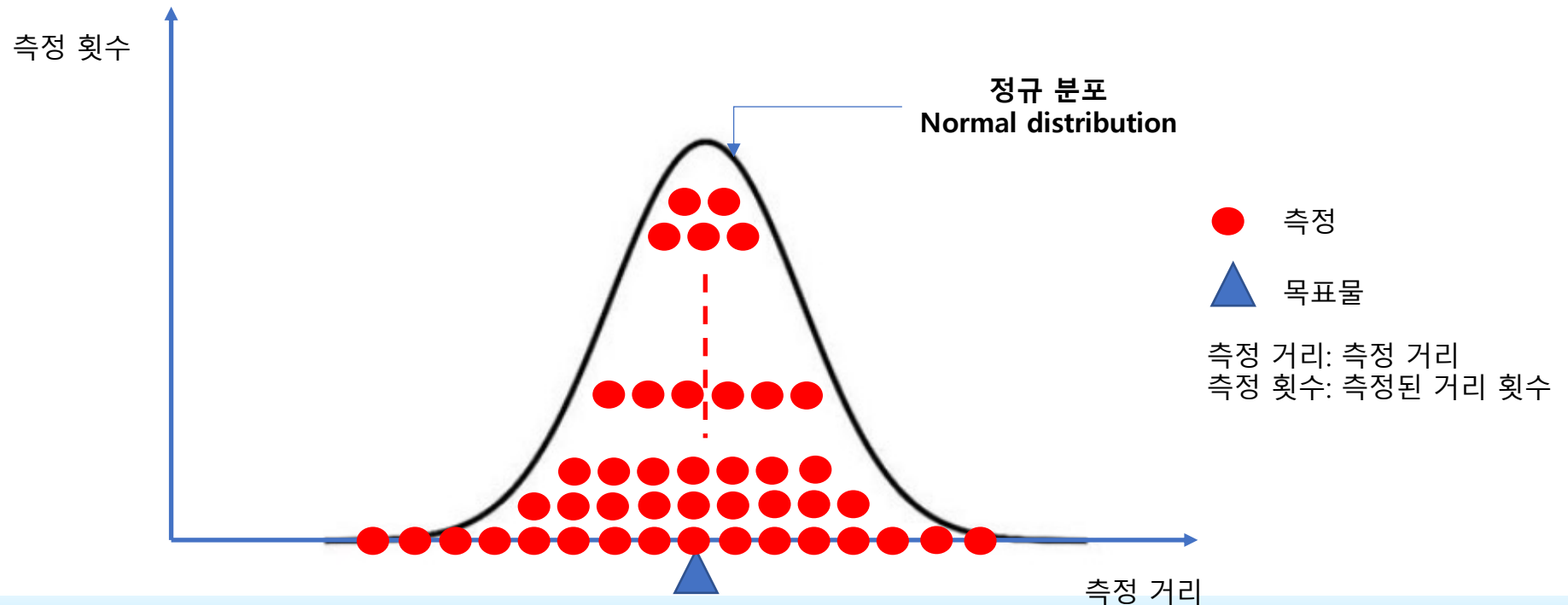
• 표준 편차

- 측정 샘플들이 평균 값에서 얼마나 멀리 떨어져 분포하는 가를 구함
- 측정에 대한 정밀도(precision)를 나타내기 위하여 사용

거리 측정

- 정규 분포(normal distribution)

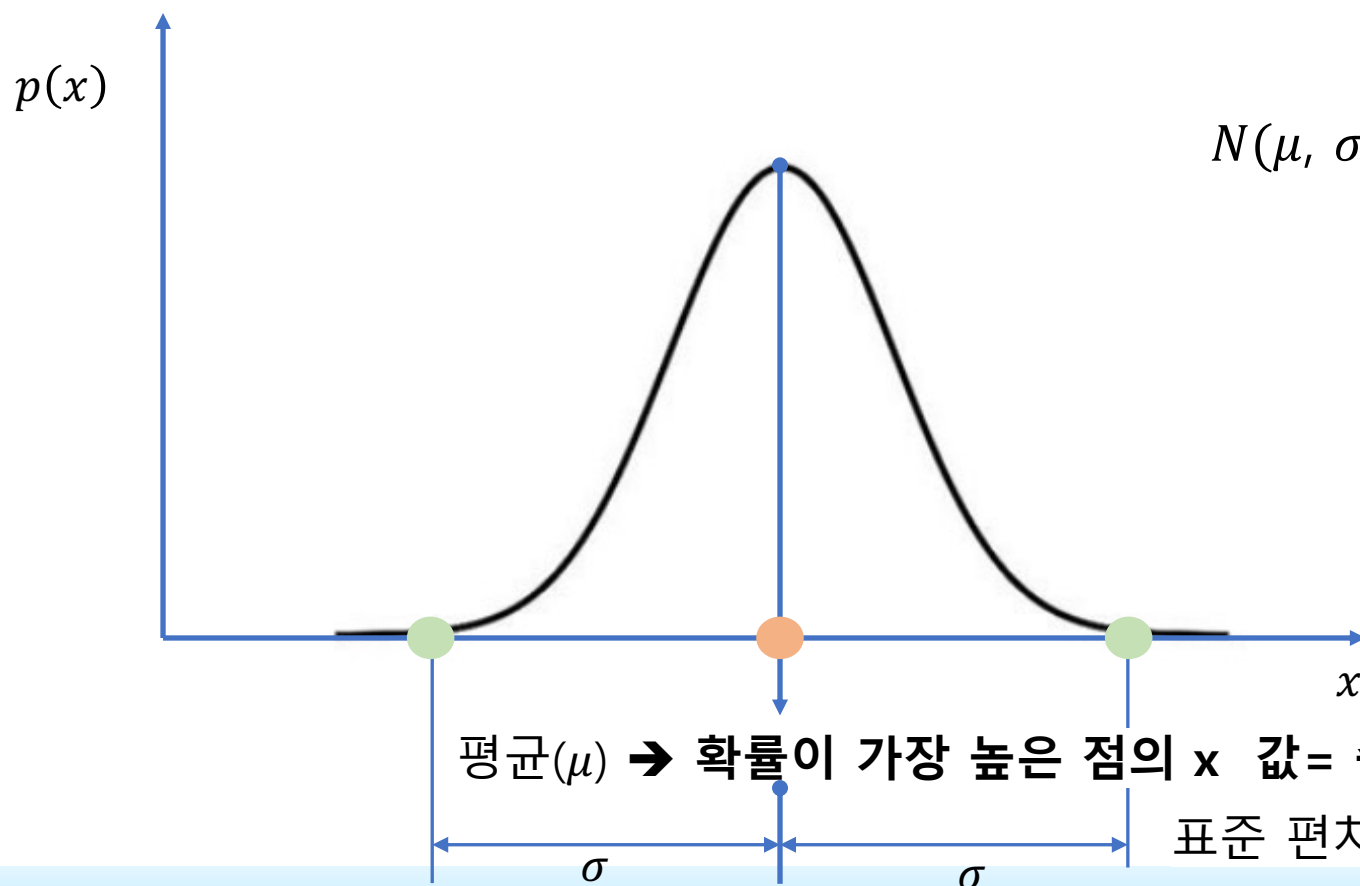
- 다수의 거리 측정을 수행하면 측정 샘플들이 일정한 분포를 형성함
 - 정규 분포 또는 가우시안 분포(Gaussian distribution) 칭함
 - 측정 거리에 대해 측정된 거리의 횟수 분포



거리 측정

- 정규 분포(normal distribution)

- 평균(μ)과 표준 편차(σ)에 의해 표현되는 확률 분포
- 수집된 자료 또는 측정된 샘플을 사용하여 확률 분포를 근사화 하기위해 사용



$$N(\mu, \sigma^2) \rightarrow p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

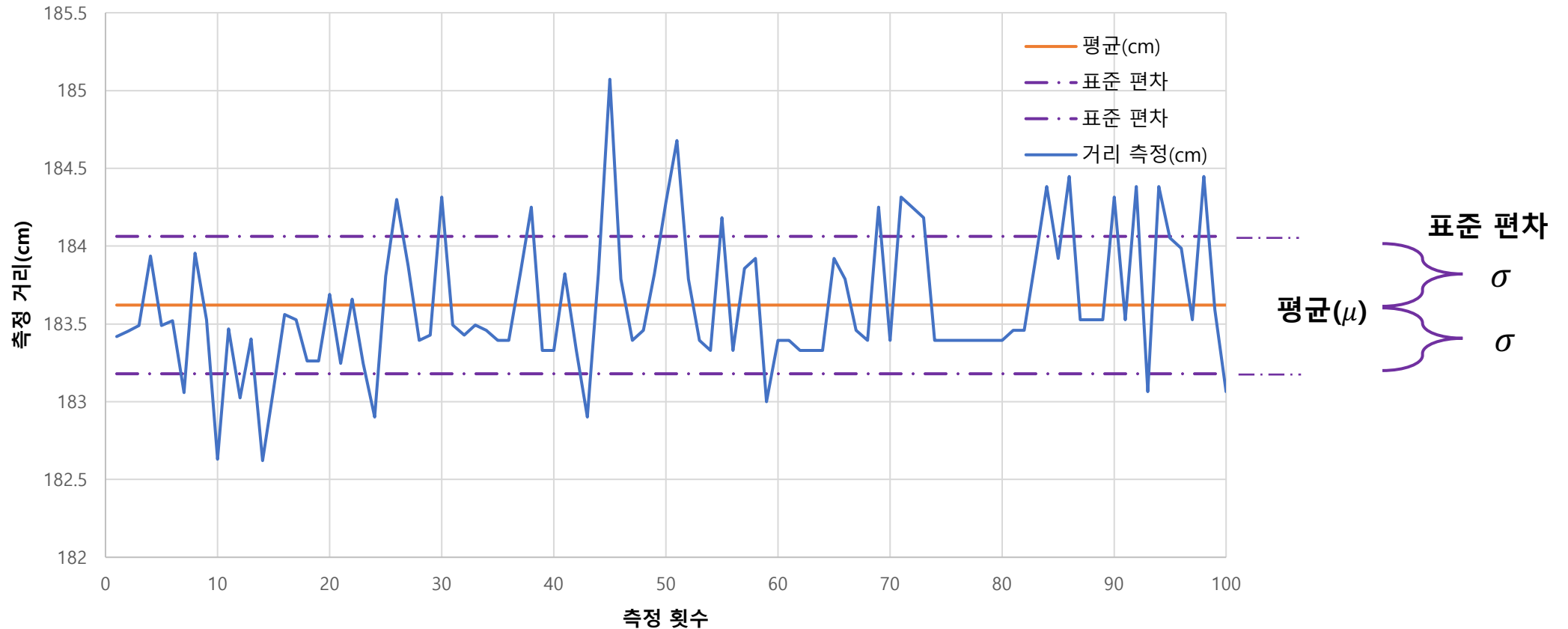
$$P(x) = \int_{-\infty}^{\infty} p(x)dx = 1$$

$p(x)$: 확률 밀도 함수

거리 측정

- 거리 측정

- 측정된 데이터로부터 평균 (μ)과 표준 편차(σ)를 구할 수 있음



거리 측정

• 평균과 표준 편차 정의

• 평균(μ)

- 평균(mean) 또는 기대 값(expectation value)이라고 함
- 모든 측정 값을 합계(sum)한 다음 시도 횟수(n)로 나누어 구한 값

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$

x_i : 거리 측정 샘플

i : 측정 샘플 *index*

$i = 1, 2, \dots, n$

n : 거리 측정 시도 횟수

• 표준 편차(σ)

- 측정 샘플들이 평균 값에서 얼마나 멀리 떨어져 있는지를 나타내는 값

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n - 1}}$$

거리 측정

• 실습 1

- 평균과 표준 편차를 구하는 프로그램 작성
 - 임의로 측정한 데이터를 입력을 받아 평균과 표준 편차를 구하는 함수 작성
 - 임의로 측정한 데이터는 후반부에 제공

평균 수식

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$

x_i : 거리 측정 샘플
 i : 측정 샘플 *index*
 $i = 1, 2, \dots, n$
 n : 거리 측정 샘플 총 개수

표준 편차 수식

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n - 1}}$$

거리 측정

- 실습 1

- 평균 함수

```
double mean(double* x, int size);
```

입력:

- x: 측정 값을 저장한 배열의 포인터
- size: 측정 값들을 저장한 배열의 크기

리턴:

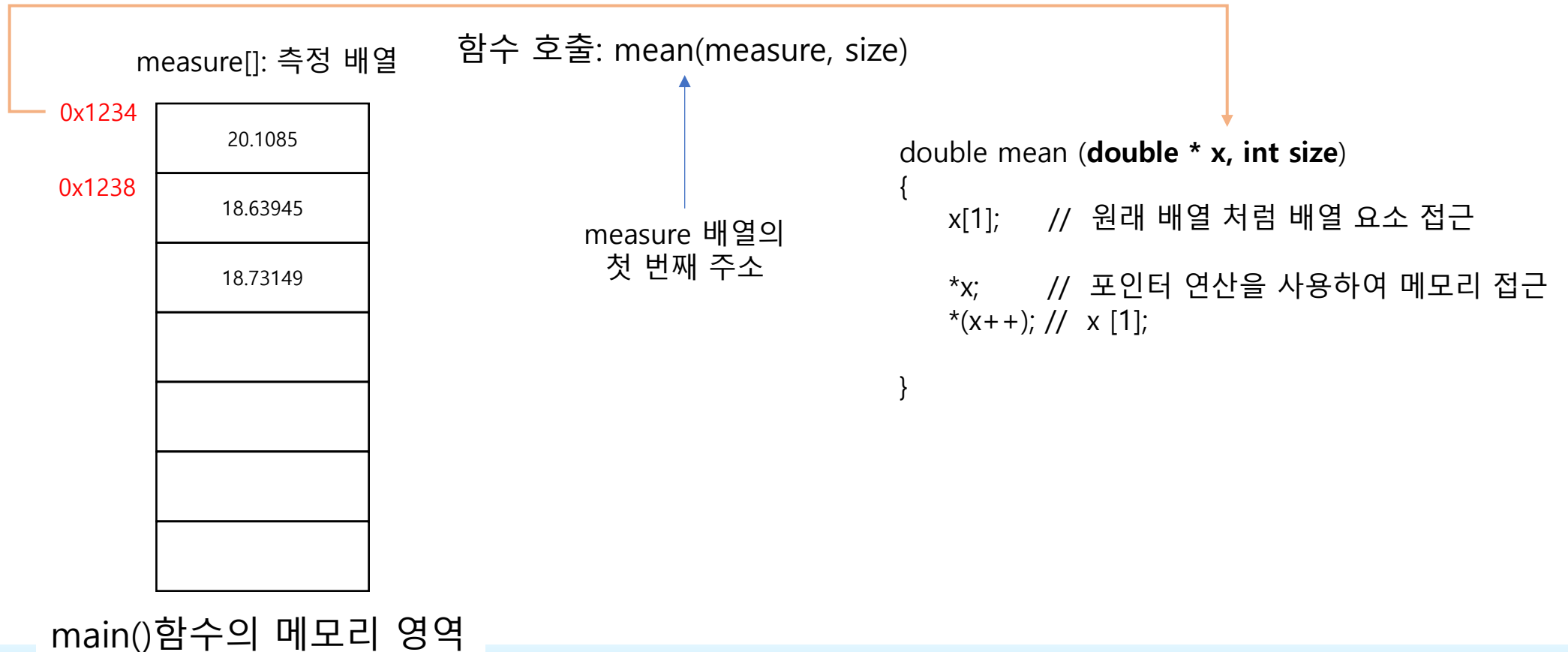
- 측정치의 평균 값

거리 측정

• 실습 1

• 평균 함수

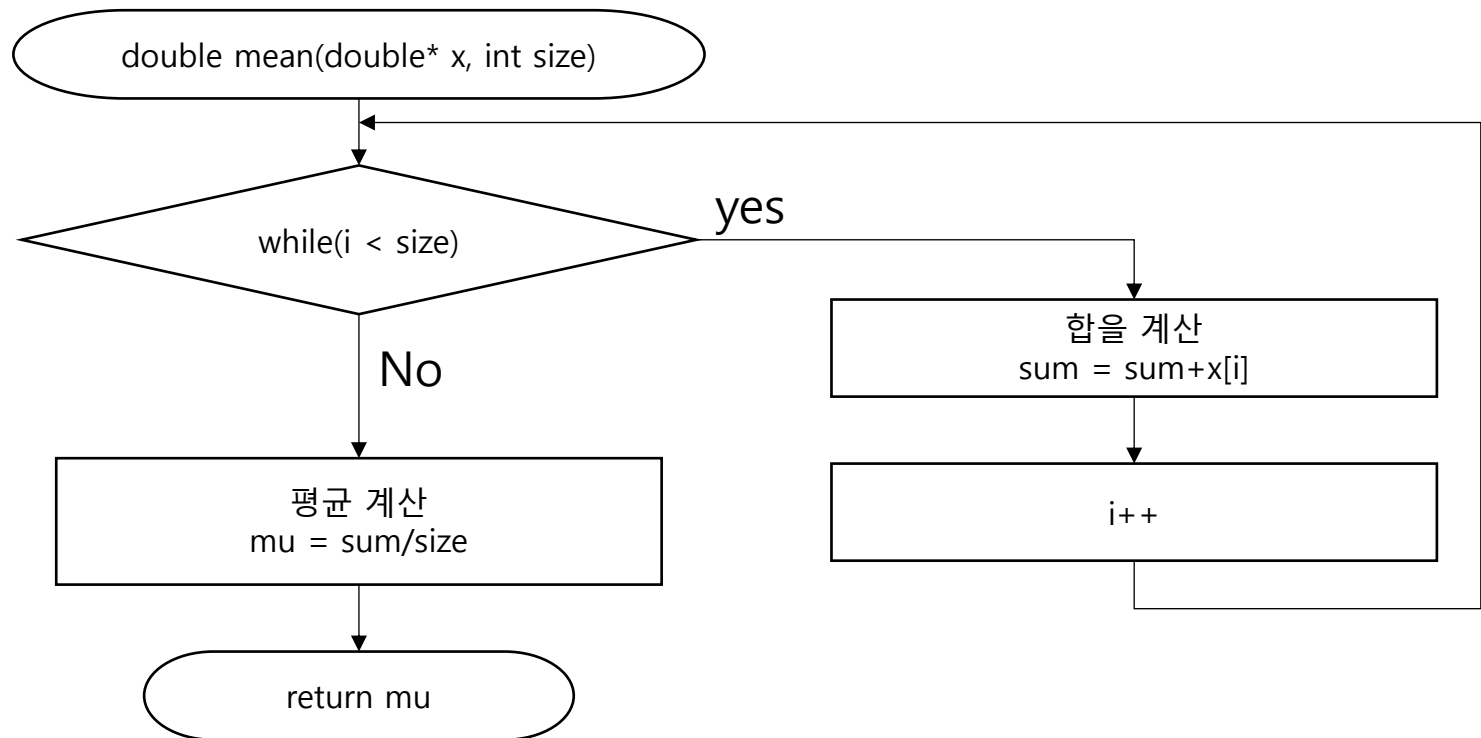
- 함수의 인자로 배열을 사용 불가 ➔ 배열의 첫 번째 주소를 함수에 전달



거리 측정

- 실습 1
 - 평균 함수

$$\mu = \frac{\sum_{i=1}^n x_i}{n}$$



거리 측정

- 실습 1
 - 평균 함수

```
double mean(double* x, int size)
{
    double sum = 0.0;    // sum 변수
    double mu = 0.0;     // 평균 결과
    int i = 0;           // 샘플 인덱스

    while(i < size)
    {
        sum += x[i]; // sum 누적
        i++;
    }
    mu = sum / size; // 평균 계산
    return mu;
}
```

sum 누적

거리 측정

- 실습 1

- 표준 편차 함수
 - 측정치가 하나 이상일 때만 구한다.

double STD (double* x, int size);

입력:

- x: 측정 값을 저장한 배열의 포인터
- size: 측정 값들을 저장한 배열의 크기

리턴:

- 측정치의 표준 편차

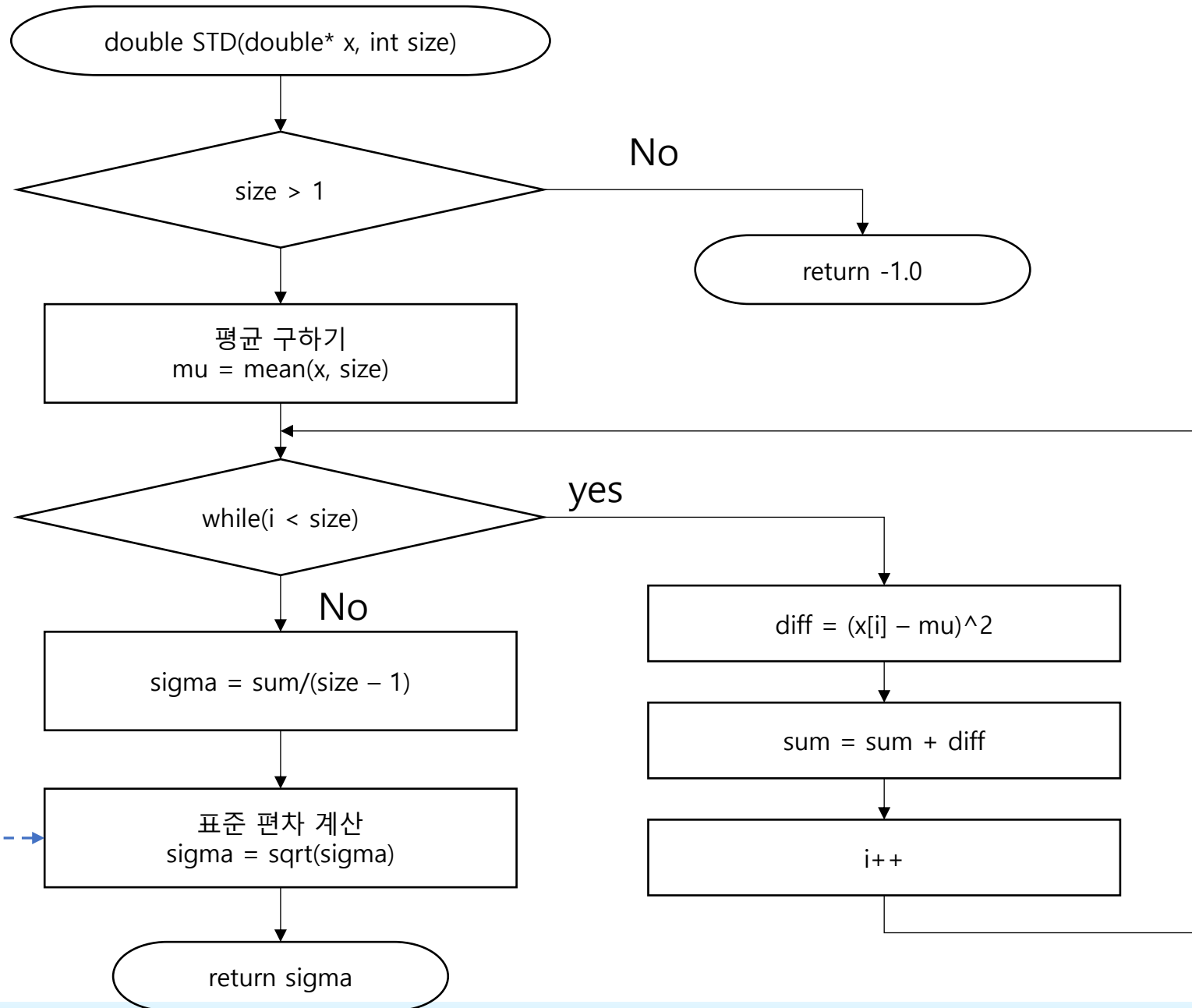
거리 측정

• 실습 1

- 표준 편차 함수

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n - 1}}$$

#include <math.h>
sqrt() 함수



거리 측정

- 실습 1
 - 표준 편차 함수

```
double STD(double* x, int size)
{
    double sum = 0.0;    // sum 변수
    double sigma = 0.0;  // 표준 편차
    double diff;
    double mu = 0;
    int i = 0;

    // 배열 요소가 1개 이하 일 때는
    // -1로 리턴
    if (size < 2) return sqrt(-1.0);

    // 평균 계산
    mu = mean(x, size);

    while(i < size)
    {
        diff = x[i] - mu;
        sum += diff * diff;
        i++;
    }

    sigma = sqrt(sum / (size - 1));

    return sigma;
}
```

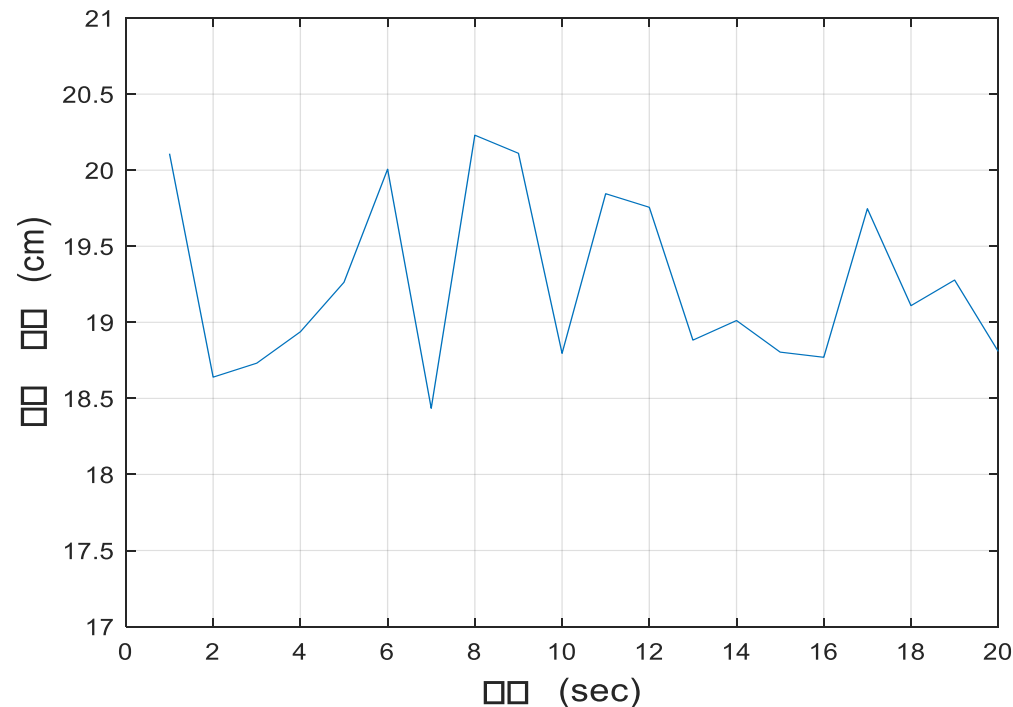

거리 측정

- 실습 1

- 거리 측정 데이터

- 1초 간격으로 20번 거리 측정한 데이터

```
double measure[] = {20.1085009279365, 18.6394598016467, 18.7314943876017, 18.9377690340056, 19.2632879107021,  
20.0065099280603, 18.4349393359215, 20.2300541812547, 20.1110822295729, 18.7961369161676,  
19.8453965649874, 19.7562004539821, 18.8828657454277, 19.0116376956973, 18.8041085481254,  
18.7700861304889, 19.7469885207948, 19.1094240806558, 19.2778747368023, 18.8087337714446};
```



거리 측정

- 실습 1
 - main

<파일명>
mean_std.c

<Compile 명령>
gcc mean_std.c -o mean_std.c -lm

<실행>
./ mean_std

```
#include <stdio.h>
#include <math.h> // sqrt()

double mean(double* x, int size);    // 평균 함수 선언
double STD(double* x, int size);     // 표준 편차 함수 선언

double measure[] = {20.1085009279365, 18.6394598016467, 18.7314943876017, 18.9377690340056,
                    19.2632879107021, 20.0065099280603, 18.4349393359215, 20.2300541812547,
                    20.1110822295729, 19.8453965649874, 19.7562004539821, 18.8828657454277,
                    19.0116376956973, 18.8041085481254, 18.7700861304889, 19.7469885207948,
                    19.1094240806558, 19.2778747368023, 18.8087337714446};

int main(void)
{
    // 평균 구하기
    printf("평균: %f cmWn", mean(measure, sizeof(measure) / sizeof(measure[0])));

    // 표준 편차 구하기
    printf("표준 편차: %f cmWn", STD(measure, sizeof(measure) / sizeof(measure[0])));

    return 0;
}
```

거리 측정

- 실습 1

<출력>

평균: 19.263628 cm

표준 편차: 0.576327 cm

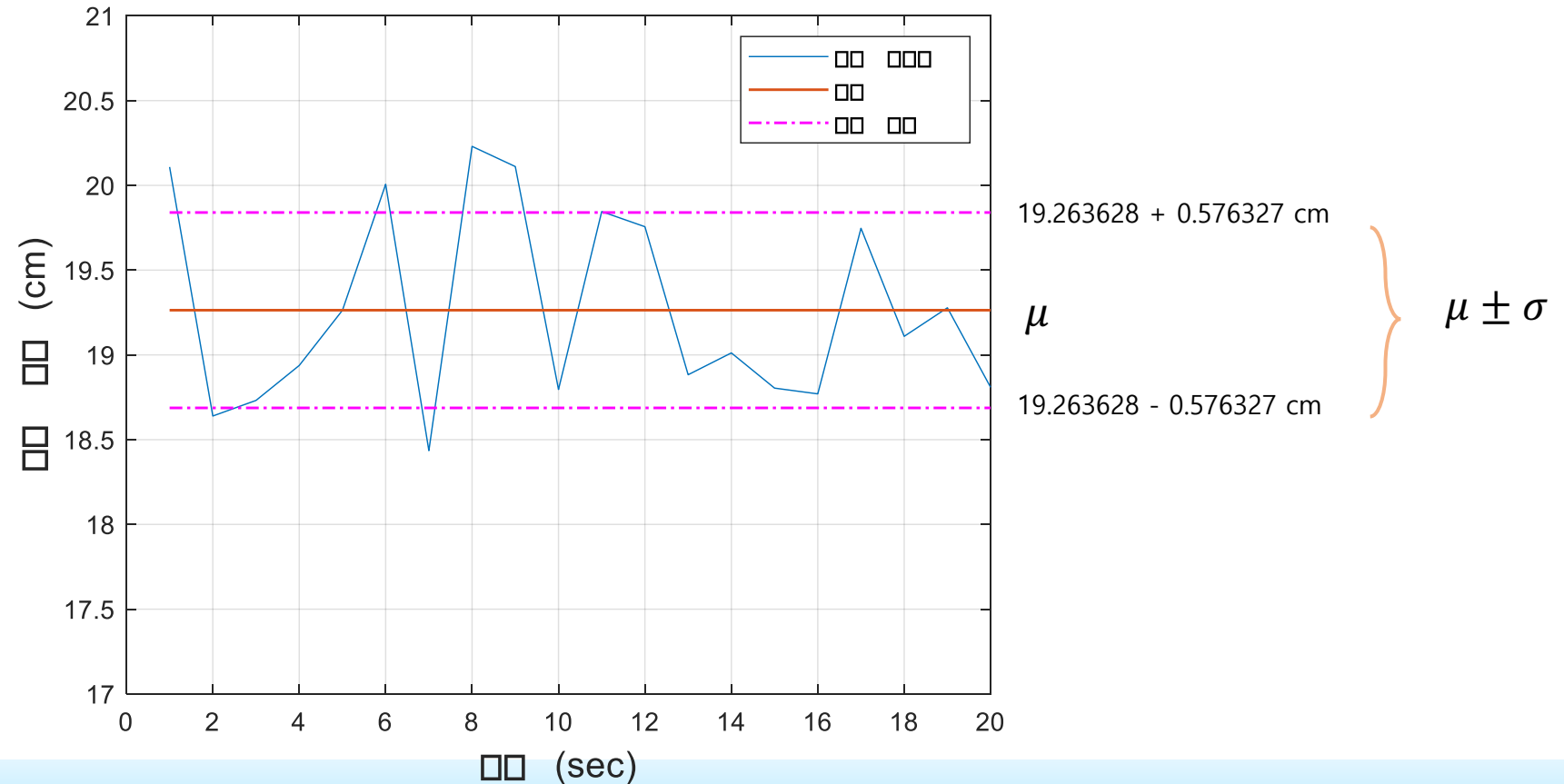
거리 측정

• 실습 1

```
double measure[] = {20.1085009279365, 18.6394598016467, 18.7314943876017, 18.9377690340056, 19.2632879107021,
20.0065099280603, 18.4349393359215, 20.2300541812547, 20.1110822295729, 18.7961369161676,
19.8453965649874, 19.7562004539821, 18.8828657454277, 19.0116376956973, 18.8041085481254,
18.7700861304889, 19.7469885207948, 19.1094240806558, 19.2778747368023, 18.8087337714446};
```

<출력>

평균: 19.263628 cm
표준 편차: 0.576327 cm

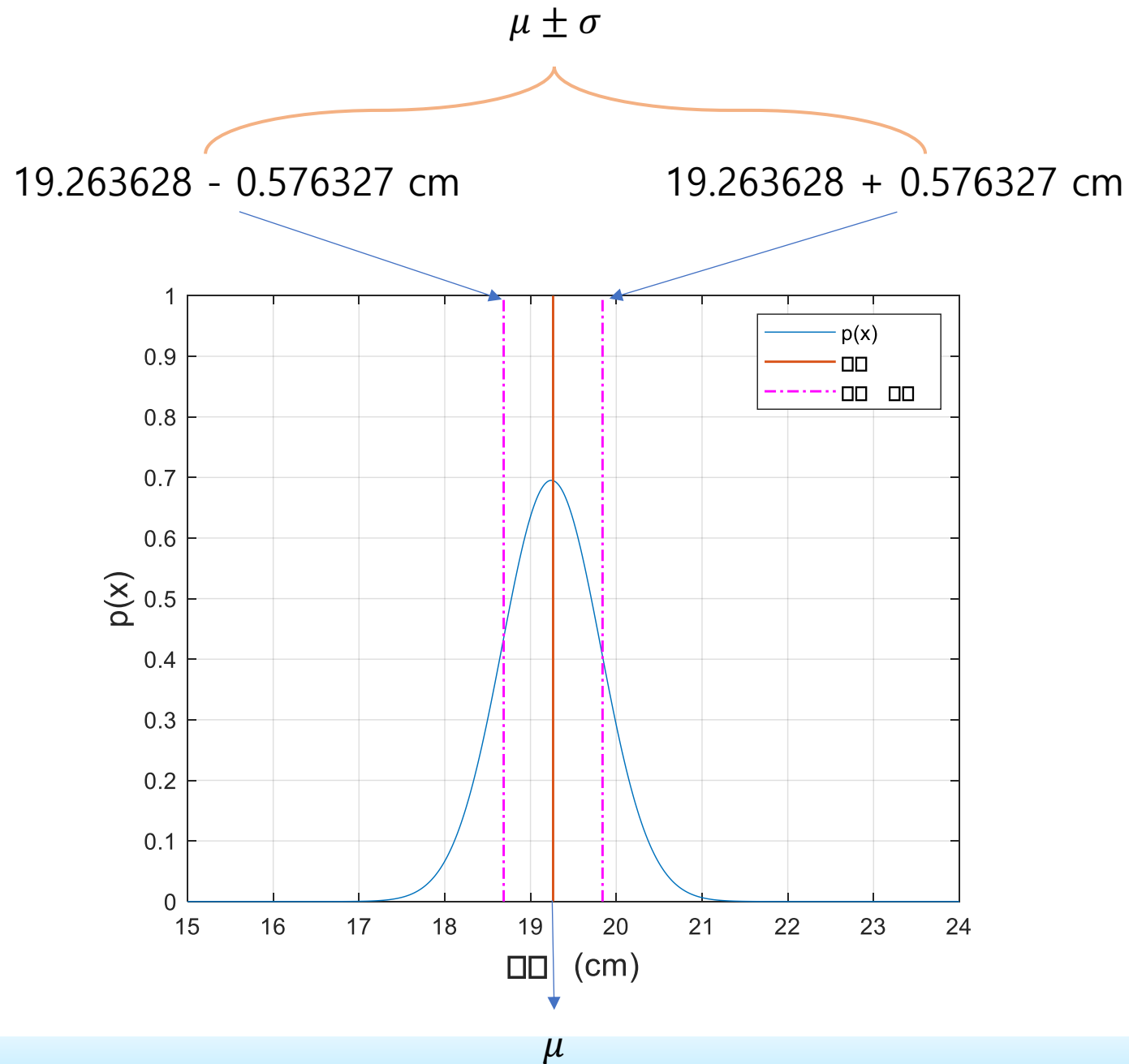


거리 측정

- 실습 1
 - 정규 분포

<출력>

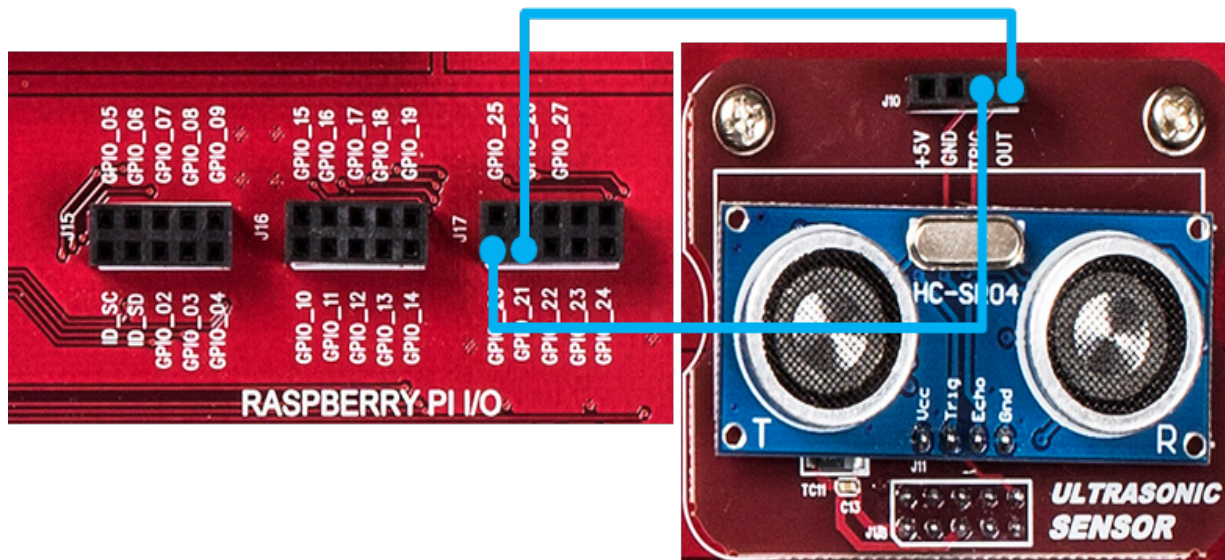
평균: 19.263628 cm
표준 편차: 0.576327 cm



초음파 센서

- Raspberry Pi GPIO 연결

BCM (Raspberry Pi)	wPi (Wiring Pi)	Ultrasonic Sensor	설명
GPIO_20	28	TRIG	Trigger 신호
GPIO_21	29	OUT	Echo 신호



거리 측정

• 실습 2

- 초음파 센서를 사용하여 거리를 측정 후 이것의 평균과 표준 편차 산출
 - 초음파 센서를 사용하여 거리를 100회 측정
 - 거리 측정에 대한 프로그램은 " 초음파 센서:거리 측정 강의" 실습 1 코드를 사용
 - 100회 측정을 위해서 측정 결과를 저장 할 수 있는 배열을 선언

```
double measure[MAX_MEASURE];
```

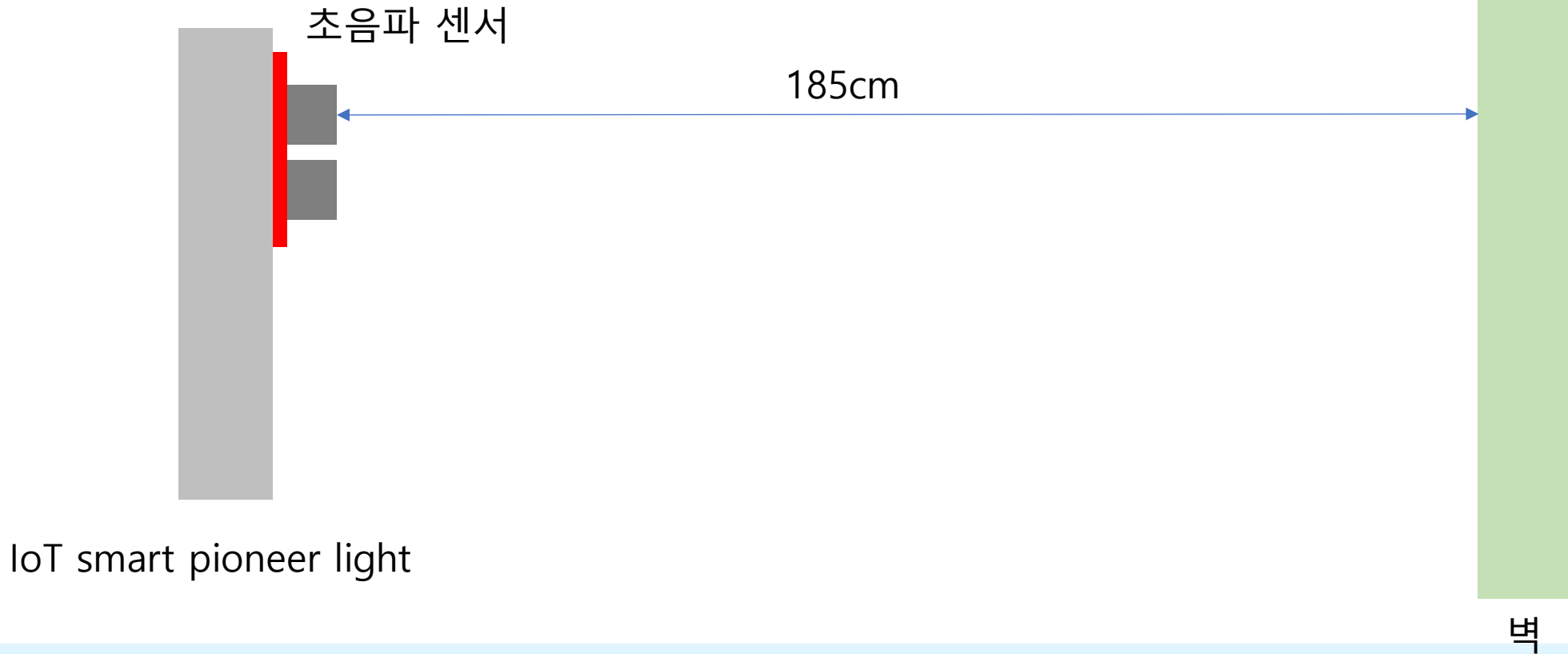
- 1회 측정 후 600msec delay
- 100 회 측정이 끝나면 측정에 대한 평균과 표준 편차를 구한다.
 - 평균과 표준 편차를 위한 함수는 실습 1의 코드를 재 사용

```
평균 = mean(measure, sizeof(measure)/sizeof(measure[0]));
```

```
표준 편차 = STD (measure, sizeof(measure)/sizeof(measure[0]));
```

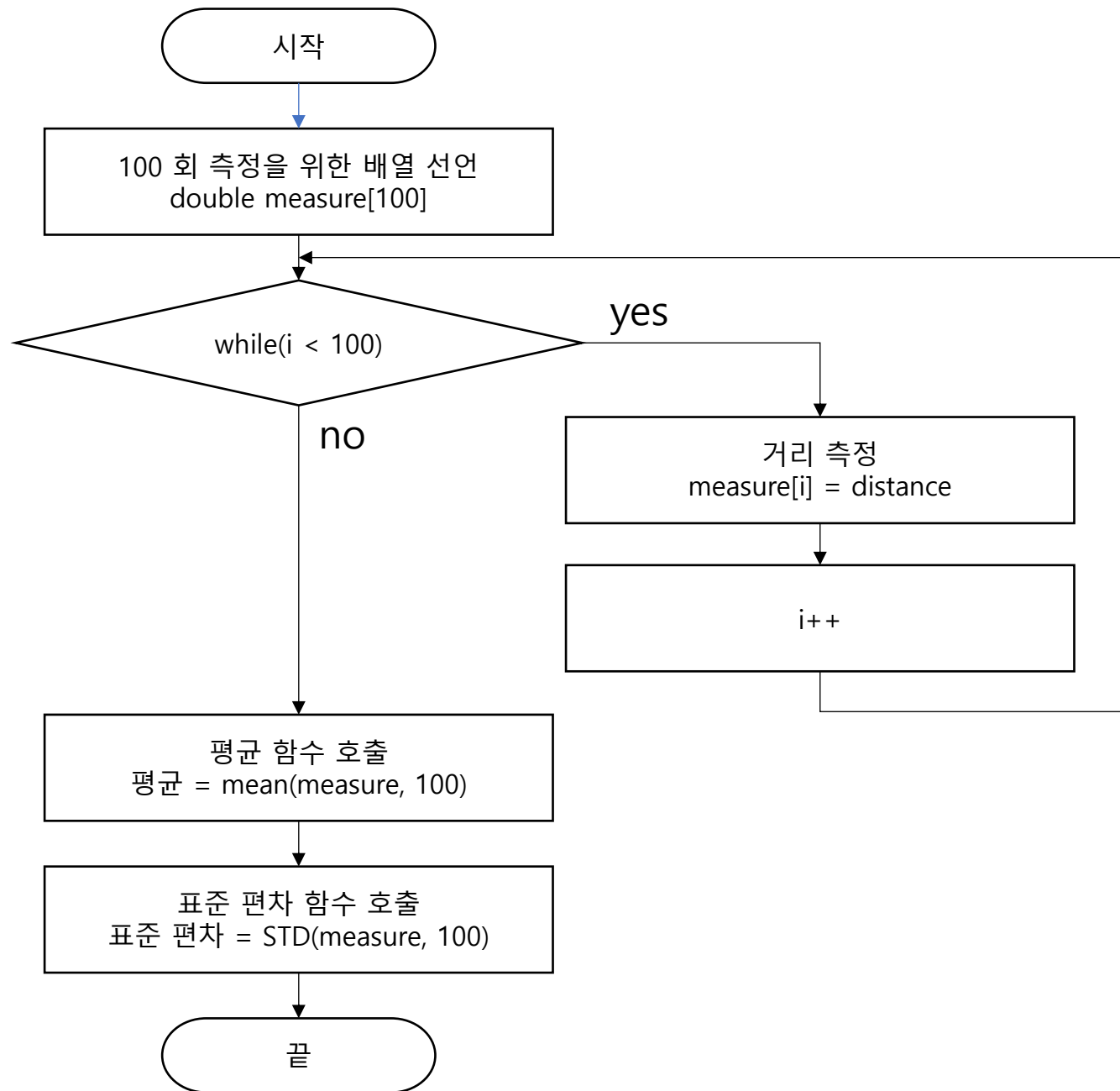
거리 측정

- 실습 2
 - 측정 셋팅



거리 측정

• 실습 2



거리 측정

• 실습 2

주의: float형을 double로 변경

```
#include <stdio.h>
#include <wiringPi.h>
#include <unistd.h>
#include <time.h>
#include <math.h> // sqrt()

#define TRIG 28 // Trig 핀
#define ECHO 29 // Echo 핀
#define MICRO_SEC 1000000 // Microsecond
#define CM 100 // 센치미터
#define VS 345.0 // 음속 m/sec

#define MAX_TRY 10 //
#define NUM_MEASURE 100 // 측정 횟수

double mean(double* x, int size); // 평균함수 선언
double STD(double* x, int size); // 표준 편차 함수 선언

int main(void)
{
    int i = 0;
    double distance = 0.0;
    double startTime, endTime;

    double measure[NUM_MEASURE];

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

    pinMode (TRIG, OUTPUT); // Trig 핀 OUTPUT
    pinMode (ECHO, INPUT); // Echo 핀 INPUT
```

거리 측정

• 실습 2

초음파 센서 거리 측정

```
while(i < NUM_MEASURE)
{
    // Trigger 신호 생성
    digitalWrite (TRIG, LOW);
    usleep(2);
    digitalWrite (TRIG, HIGH);
    usleep(10);           // 10 usec
    digitalWrite (TRIG, LOW);

    // Echo 신호 수신 ==> Start time
    while(digitalRead(ECHO) == LOW);
    startTime = ((float)micros())/MICRO_SEC; // sec 단위 환산
    // Echo 신호 수신 ==> End time
    while(digitalRead(ECHO) == HIGH);
    endTime = ((float)micros())/MICRO_SEC; // sec 단위 환산

    // 거리 계산
    distance = (((endTime - startTime)*VS)*(1.0/2.0))*CM;

    measure[i] = distance;
    printf("measure: %f cm\n", measure[i]);

    i++;
    delay(600);
}

// 평균 구하기
printf("평균: %f cm\n", mean(measure, sizeof(measure) / sizeof(measure[0])));

// 표준 편차 구하기
printf("표준 편차: %f cm\n", STD(measure, sizeof(measure) / sizeof(measure[0])));

return 0;
}
```

거리 측정

• 실습 2

- 100번 측정 결과

<파일명>

ultra_measureX100.c

<Compile 명령>

gcc ultra_measureX100.c -o ultra_measureX100 -lwiringPi -lm

<실행>

./ultra_measureX100

<출력>

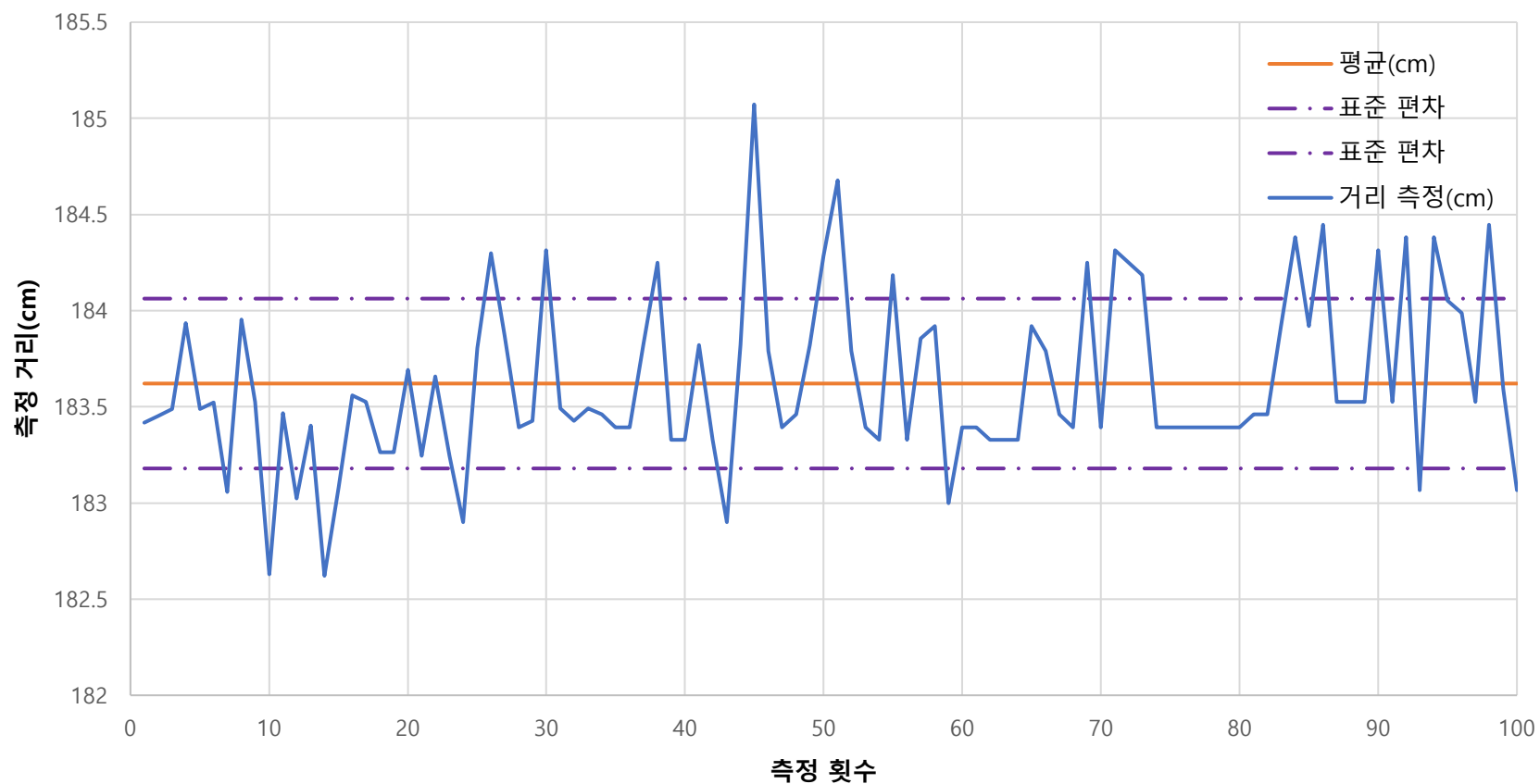
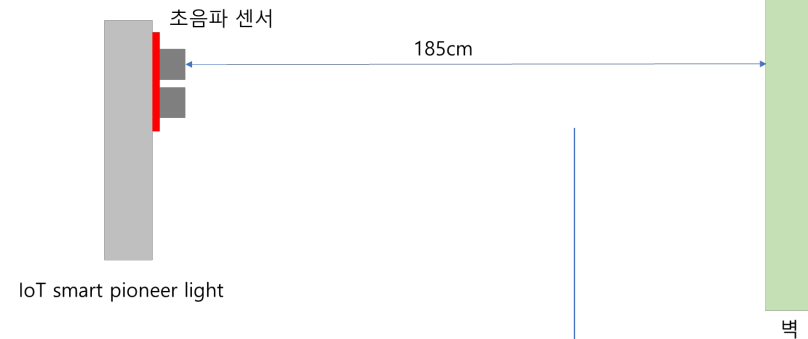
```
measure: 183.460236
measure: 183.920860
measure: 184.381485
measure: 183.920860
measure: 184.447289
measure: 183.526039
measure: 183.526039
measure: 183.526039
measure: 184.315681
measure: 183.526039
measure: 184.381485
measure: 183.065414
measure: 184.381485
measure: 184.052467
measure: 183.986664
measure: 183.526039
measure: 184.447289
measure: 183.591843
measure: 183.065414
```

평균: 183.621075 cm

표준 편차: 0.441565 cm

거리 측정

• 실습 2



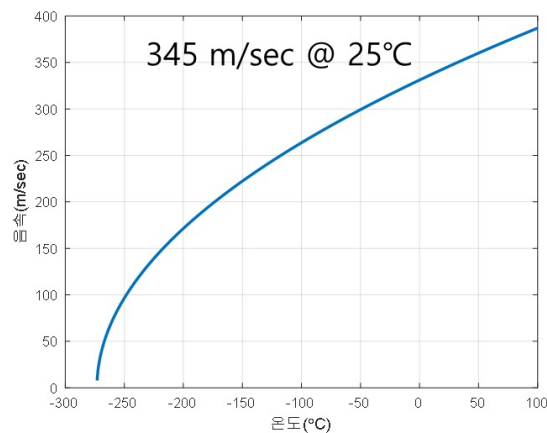
$$\text{오차(error)} = |185 - 183.621|$$

평균: 183.621075 cm
표준 편차: 0.441565 cm

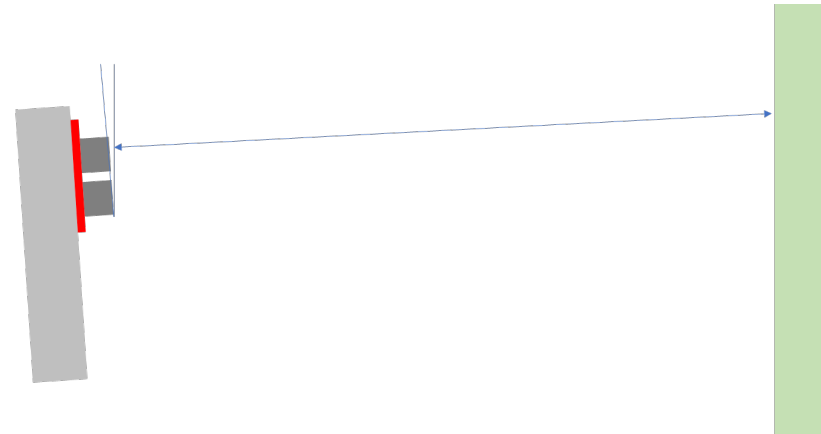
거리 측정

• 실습 2

- 어디서 약 1.4cm의 오차를 발생 시켰나?
 - 오차를 발생 시킨 예상 원인
 - 온도
 - 온도에 따라 음속 변화
 - 각도 오차
 - 설치 각도에 의한 길이 측정 오차 발생
 - 벽의 재질
 - 벽면이 고르지 못 하면 측정 오차 발생



온도에 따른 음속의 변화

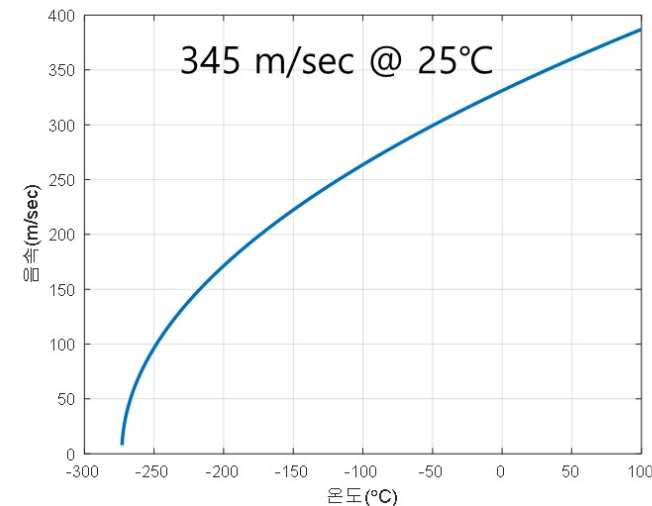


설치 각도에 따른 측정 오차

거리 측정

• 실습 2

- 오차를 발생 시킨 원인
 - 온도에 따른 음속 변화
 - 345m/sec → 357m/sec
 - 345m/sec 방안의 온도 25°C 가정 → 약 27°C 측정



음속: 345m/sec@25 °C

음속: 347m/sec@27 °C

<출력>

평균: 183.621075 cm
표준 편차: 0.441565 cm



<출력>

평균: 185.176197 cm
표준 편차: 0.442754 cm

정밀한 측정을 위해서는 오차 요인 고려!

거리 측정

- 실습 2
- 그래프

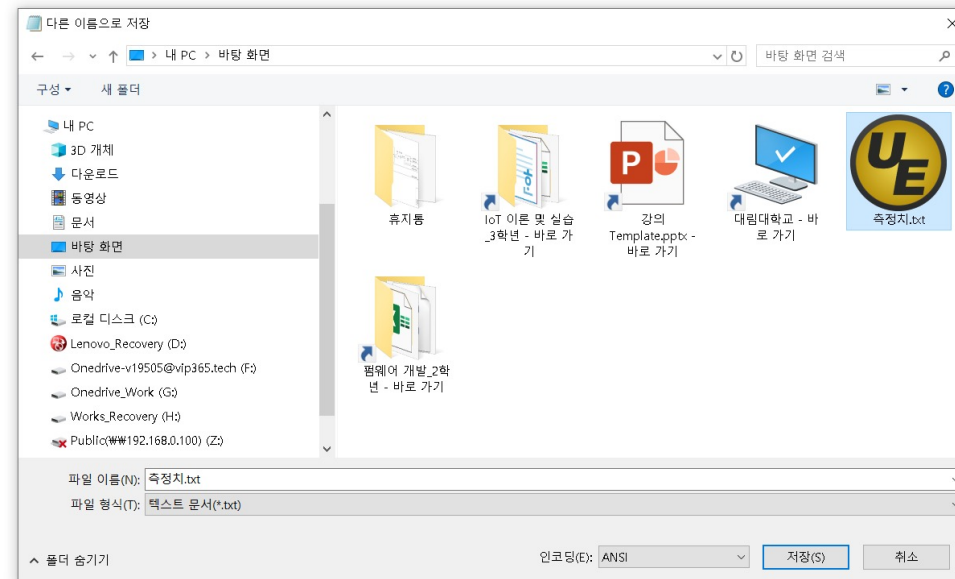
1 Putty 화면을 마우스로 긁어 복사

```
measure: 183.460236
measure: 183.822155
measure: 184.282780
measure: 184.677601
measure: 183.789253
measure: 183.394432
measure: 183.328629
measure: 184.184074
measure: 183.328629
measure: 183.855057
measure: 183.920860
measure: 182.999611
measure: 183.394432
measure: 183.394432
measure: 183.328629
measure: 183.328629
measure: 183.328629
measure: 183.920860
measure: 183.789253
measure: 183.460236
measure: 183.394432
measure: 184.249878
measure: 183.394432
measure: 184.315681
measure: 184.249878
measure: 184.184074
measure: 183.394432
measure: 183.394432
measure: 183.394432
measure: 183.394432
measure: 183.394432
measure: 183.460236
measure: 183.460236
measure: 183.920860
measure: 184.381485
measure: 183.920860
measure: 184.447289
measure: 183.526039
measure: 183.526039
measure: 183.526039
measure: 184.315681
measure: 183.526039
measure: 184.381485
measure: 183.065414
measure: 184.381485
measure: 184.052467
measure: 183.986664
measure: 183.526039
measure: 184.447289
measure: 183.591843
measure: 183.065414
```

제목 없음 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말

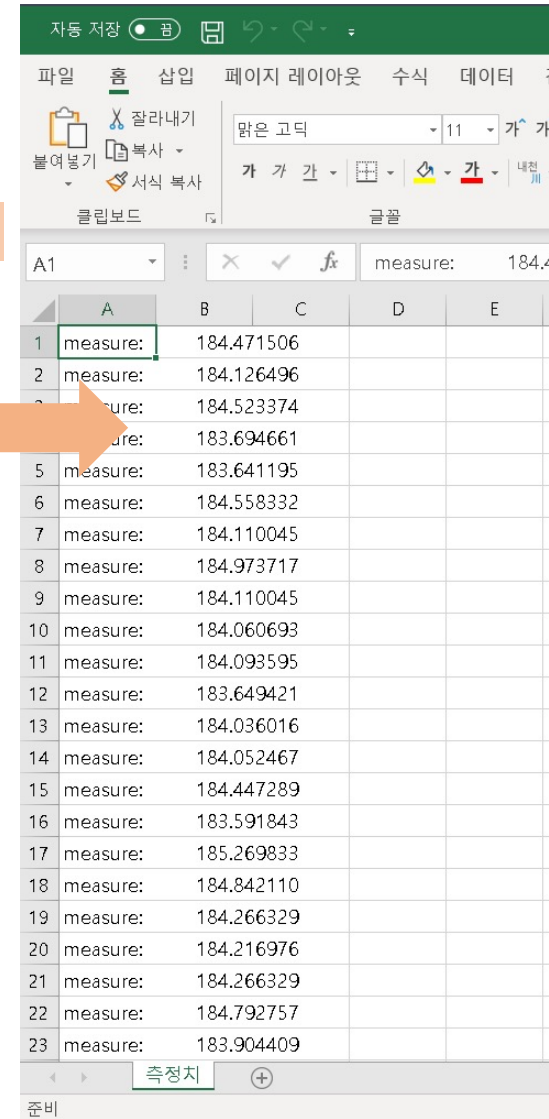
```
measure: 184.471506
measure: 184.126496
measure: 184.523374
measure: 183.694661
measure: 183.641195
measure: 184.558332
measure: 184.110045
measure: 184.973717
measure: 184.110045
measure: 184.060693
measure: 184.093595
measure: 183.649421
measure: 184.036016
measure: 184.052467
measure: 184.447289
measure: 183.591843
measure: 185.269833
measure: 184.842110
measure: 184.266329
measure: 184.216976
measure: 184.266329
measure: 184.792757
measure: 183.904409
measure: 183.739901
measure: 184.628248
measure: 184.216976
measure: 183.756351
measure: 184.644699
measure: 184.151173
measure: 184.677601
measure: 183.822155
measure: 184.611797
measure: 184.216976
measure: 183.756351
measure: 184.644699
measure: 183.822155
measure: 183.789253
measure: 183.328629
measure: 184.513092
measure: 183.591843
measure: 184.019566
measure: 183.624744
measure: 183.723450
```

2 메모장에 붙여 넣고 저장

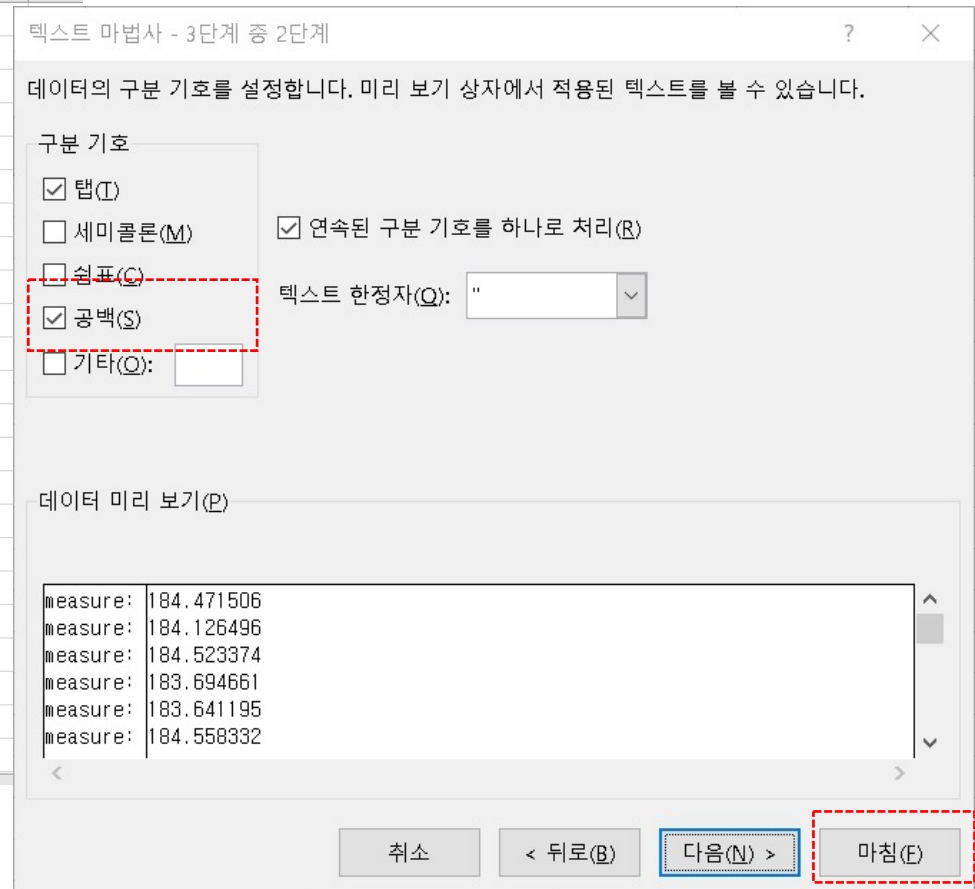
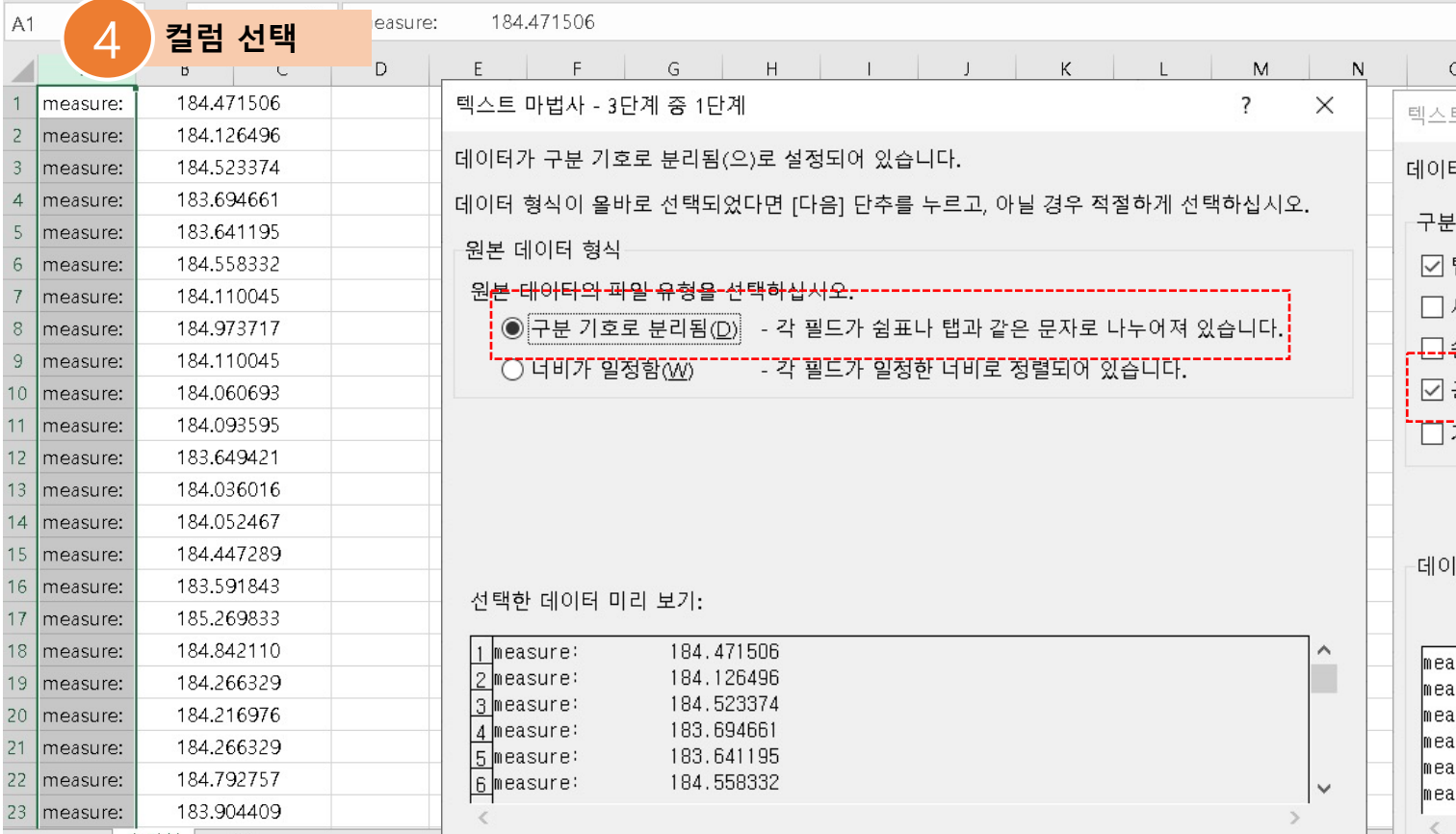
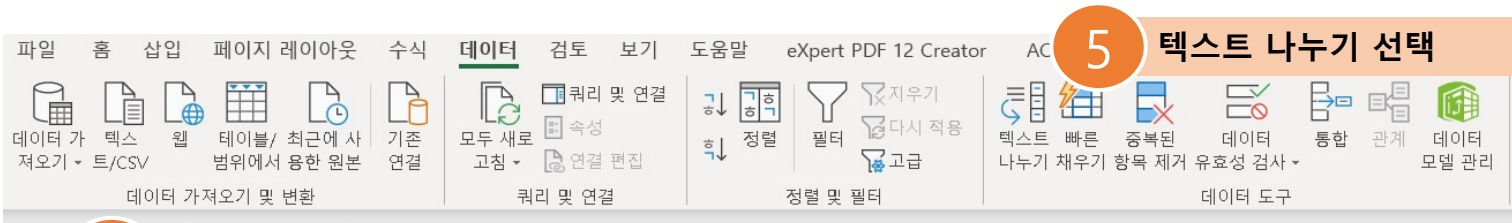


거리 측정

3 엑셀을 열고 저장된 파일을 drag-drop



	A	B	C	D	E
1	measure:	184.471506			
2	measure:	184.126496			
3	measure:	184.523374			
4	measure:	183.694661			
5	measure:	183.641195			
6	measure:	184.558332			
7	measure:	184.110045			
8	measure:	184.973717			
9	measure:	184.110045			
10	measure:	184.060693			
11	measure:	184.093595			
12	measure:	183.649421			
13	measure:	184.036016			
14	measure:	184.052467			
15	measure:	184.447289			
16	measure:	183.591843			
17	measure:	185.269833			
18	measure:	184.842110			
19	measure:	184.266329			
20	measure:	184.216976			
21	measure:	184.266329			
22	measure:	184.792757			
23	measure:	183.904409			

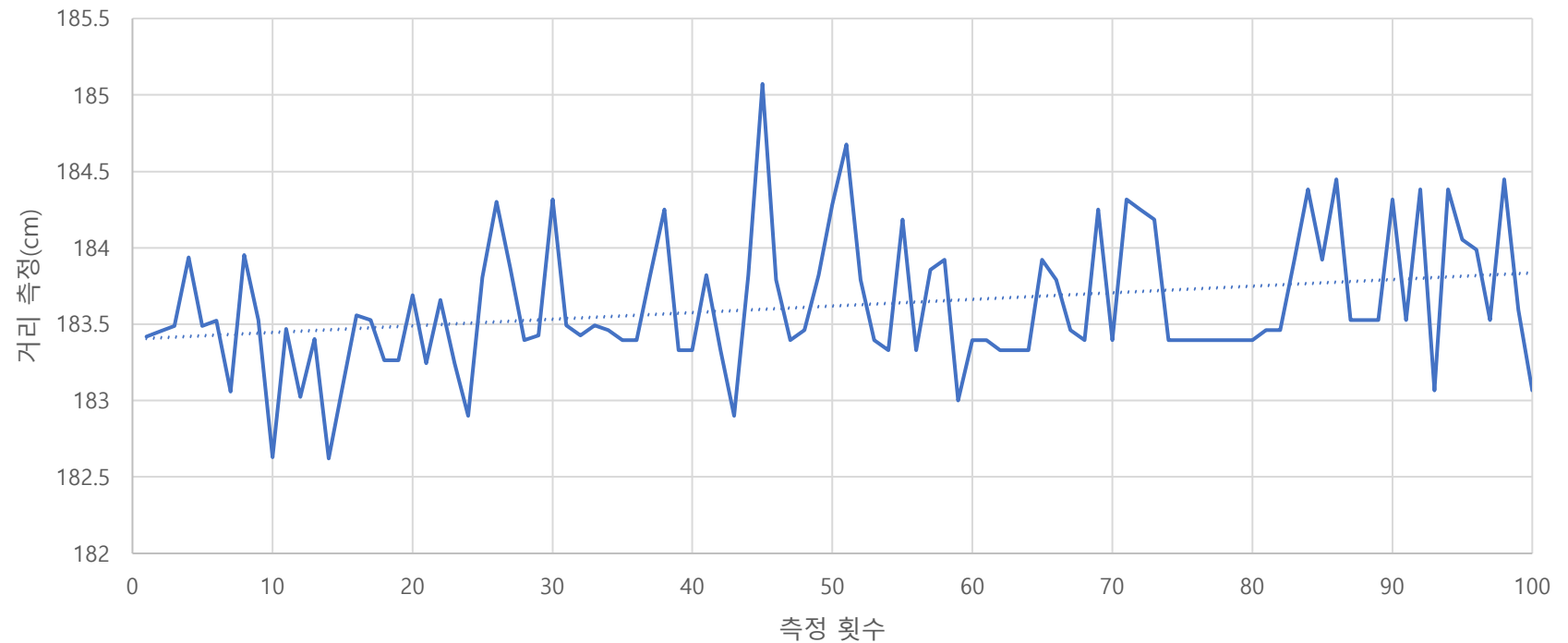


거리 측정

	A	B
1	measure:	184.4715
2	measure:	184.1265
3	measure:	184.5234
4	measure:	183.6947
5	measure:	183.6412
6	measure:	184.5583
7	measure:	184.11
8	measure:	184.9737
9	measure:	184.11
10	measure:	184.0607
11	measure:	184.0936
12	measure:	183.6494
13	measure:	184.036
14	measure:	184.0529
15	measure:	184.4473
16	measure:	183.5918
17	measure:	185.2698
18	measure:	184.8421
19	measure:	184.2663
20	measure:	184.217
21	measure:	184.2663
22	measure:	184.7928
23	measure:	183.9044

6 B 컬럼 전체 선택 후 삽입 -> 차트 -> 분산형

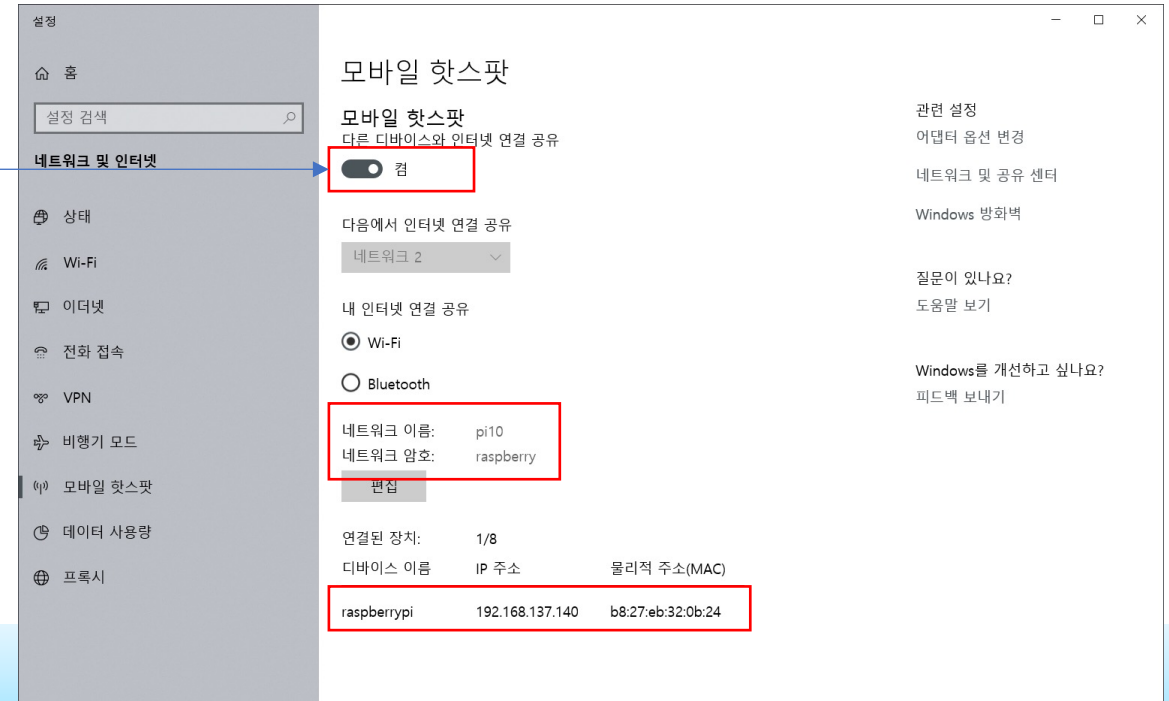
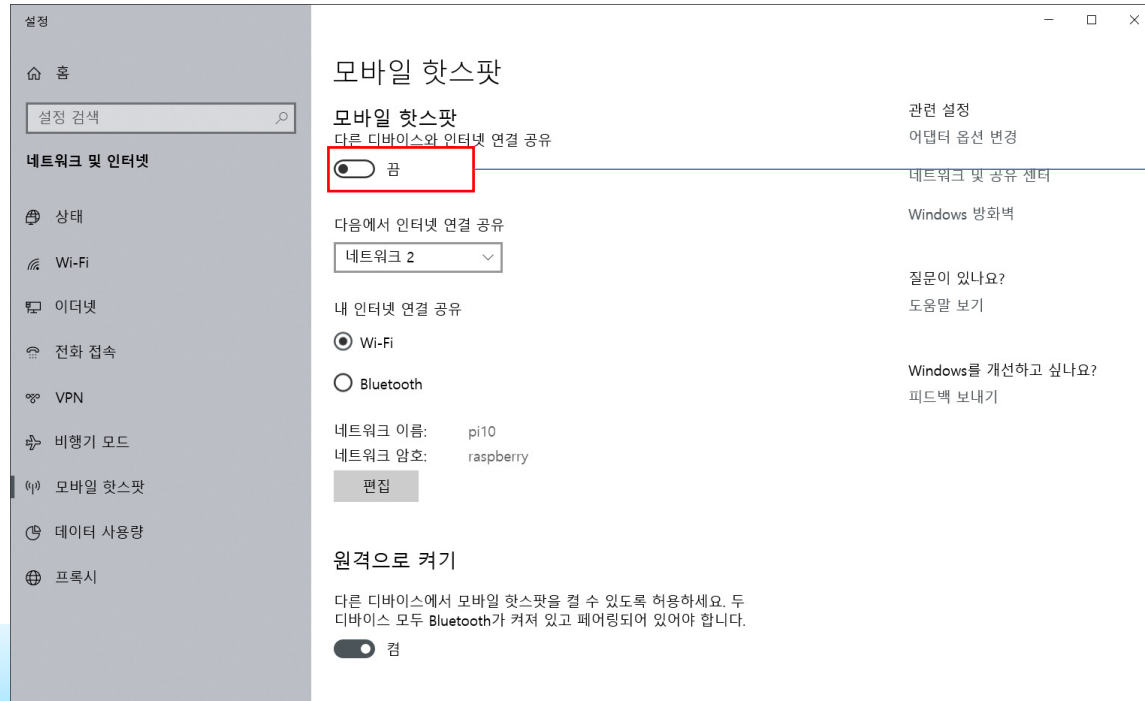
거리 측정 데이터



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!