

09_example_2 도전과제 코드 설명

정보보안암호수학과

20161928 이윤진

```
// Arduino pin assignment
#define PIN_LED 9
#define PIN_TRIG 12
#define PIN_ECHO 13

// configurable parameters
#define SND_VEL 346.0 // sound velocity at 24 celsius degree (unit: m/s)
// #define INTERVAL 100 // sampling interval (unit: ms)
#define INTERVAL 25
#define _DIST_MIN 100 // minimum distance to be measured (unit: mm)
#define _DIST_MAX 300 // maximum distance to be measured (unit: mm)
#define N 10 // number of samples to save
// global variables
float timeout; // unit: us
float dist_min, dist_max, dist_raw; // unit: mm
unsigned long last_sampling_time; // unit: ms
float scale; // used for pulse duration to distance conversion
float queue_array[N] = {0, };
int median, save_num = 0;
int err;

void setup() {
// initialize GPIO pins
pinMode(PIN_LED, OUTPUT);
pinMode(PIN_TRIG, OUTPUT);
digitalWrite(PIN_TRIG, LOW);
pinMode(PIN_ECHO, INPUT);

// initialize USS related variables
dist_min = _DIST_MIN;
dist_max = _DIST_MAX;
timeout = (INTERVAL / 2) * 1000.0; // precalculate pulseIn() timeout value. (unit: us)
dist_raw = 0.0; // raw distance output from USS (unit: mm)
scale = 0.001 * 0.5 * SND_VEL;

// initialize serial port
Serial.begin(57600);

// initialize last sampling time
last_sampling_time = 0;
}

void loop() {
int fadeValue = 0;
float cal;
// wait until next sampling time.
if(millis() < last_sampling_time + INTERVAL) return;

// get a distance reading from the USS
dist_raw = USS_measure(PIN_TRIG, PIN_ECHO);
```

저장할 샘플의 개수를 N으로 선언

N개의 샘플을 queue_array에 저장
save_num은 현재까지 저장한 샘플의 개수
이전 측정값과 비교해서 차이가 크면 측정 오류라고 판단할 수 있으므로,
이를 판단하기 위해 err 변수 선언

```

// output the read value to the serial port
Serial.print("Min:0,");
Serial.print("raw:");
Serial.print(dist_raw);
Serial.print(",");
Serial.print("median:");
Serial.print(map(median,0,400,100,500));
Serial.print(",");
Serial.println("Max:500");

// turn on the LED if the distance is between dist_min and dist_max
if(dist_raw < dist_min || dist_raw > dist_max) {
    analogWrite(PIN_LED, 255);
}
else {
    cal = (dist_min + dist_max)/2;
    fadeValue = (abs(cal - dist_raw) * 256)/100;
    analogWrite(PIN_LED, fadeValue);
}

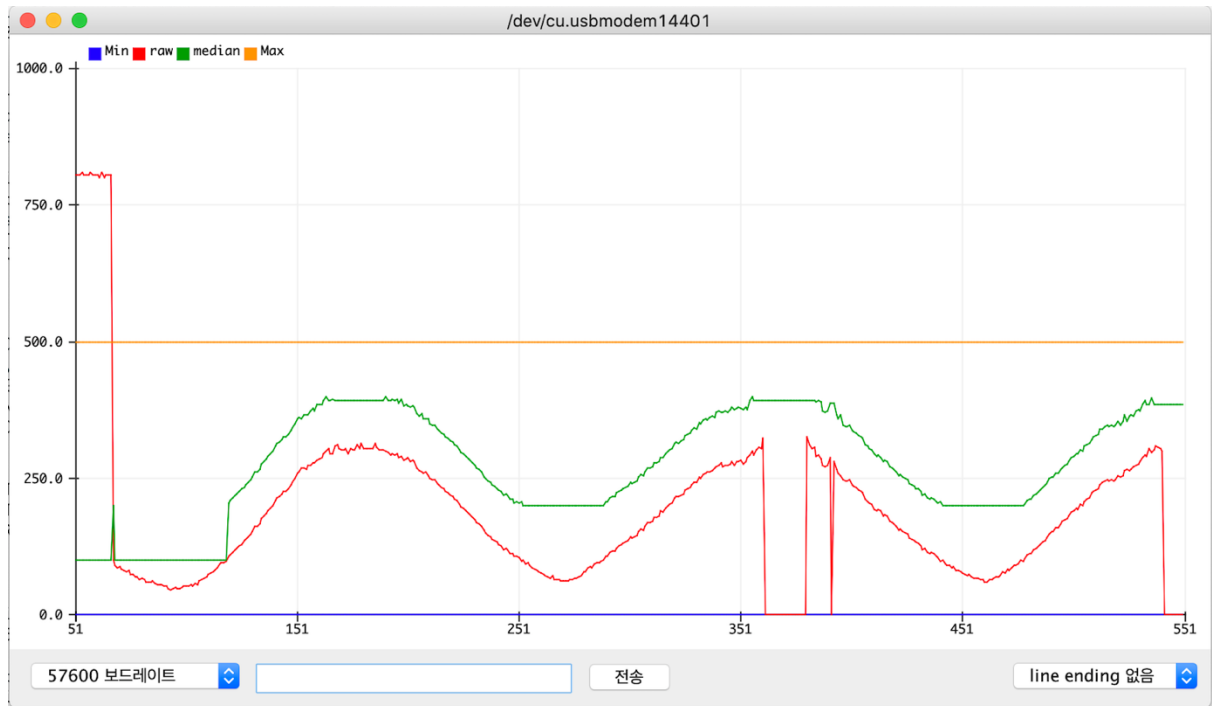
// update last sampling time
last_sampling_time += INTERVAL;
}

// get a distance reading from USS. return value is in millimeter.
float USS_measure(int TRIG, int ECHO)
{
    float reading;
    //float err;
    //int num = N/2; //to calculate medium
    digitalWrite(TRIG, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG, LOW);
    1 reading = pulseIn(ECHO, HIGH, timeout) * scale;
    median = reading; // unit: mm
    2 err = abs(reading - queue_array[save_num%N - 1]);
    3 if(reading < dist_min || reading > dist_max || err>250) {
        median = queue_array[int(N/2)];
    }
    queue_array[save_num%N]=median;
    4 save_num++;
    return reading;
}

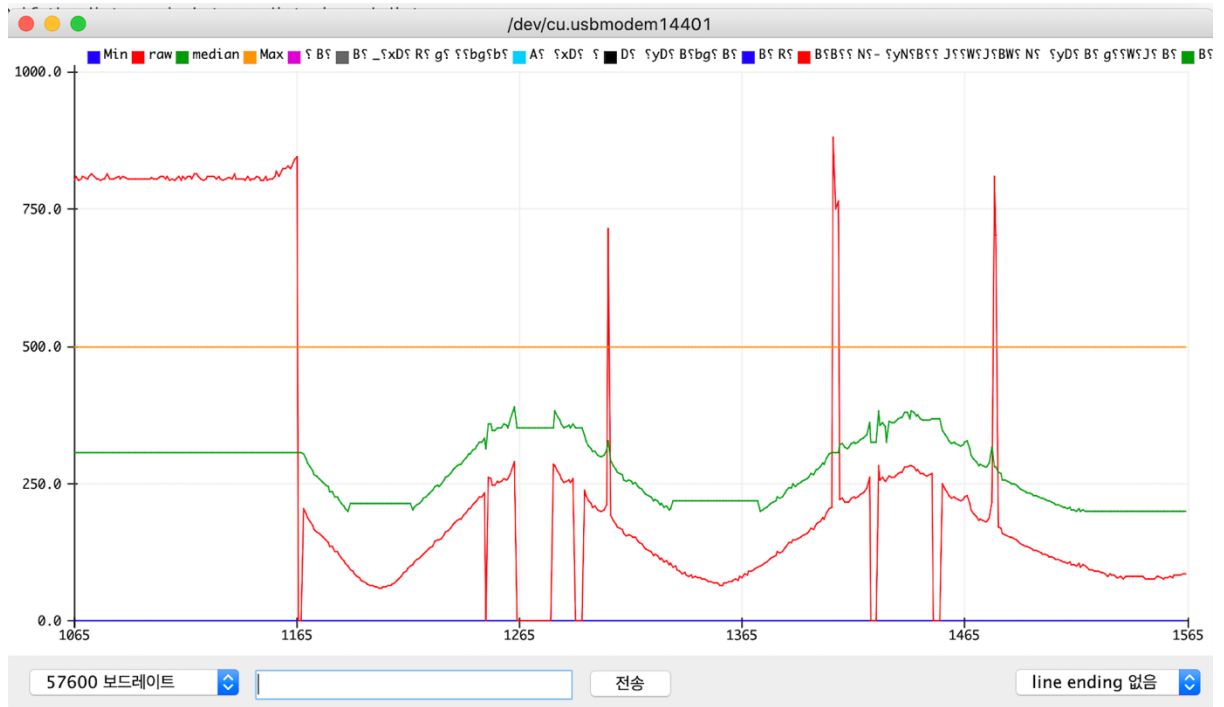
```

1. N개의 배열인 queue_array에 초음파 센서를 통해 측정한 reading 값을 저장한다. 이때, save_num%N을 사용해 최근 N개의 값만 저장되도록 한다.
측정 오류가 없다면 median 값은 reading 값과 같도록 설정한다.
2. 직전에 측정한 값인 queue_array[save_num%N - 1]과 현재 측정값인 reading을 비교해 그 차이를 err에 저장한다.
3. 측정 오류라고 판단되는 상황은 reading이 측정 범위 내에 속하지 않거나, 이전 측정값과 비교했을 때 차이가 많이 나는 경우라고 생각했기 때문에, 해당 조건들을 조건문에 추가했다. 이 때 오차는 250mm 이상인 경우로 설정하였다.
측정 오류가 발생하면 중앙값을 median에 저장한다.
4. 위 과정이 모두 끝나면 save_num++을 통해 저장한 샘플의 개수를 1 증가시킨다.
측정 오류가 발생하면 중앙값을 median에 저장한다.

N=3일 때



N=10일 때



N=30일 때

