

도전과제 중위수 필터

1. 선언된 전역변수

```
#define MEDIAN_NUM 30

// global variables
unsigned long last_sampling_time; // unit: msec
float dist_prev = _DIST_MAX;     // Distance last-measured
float dist_ema;                   // EMA distance
float dist_meds[MEDIAN_NUM];      // median distances
int index_meds = 0;               // 현재 dist_meds에 몇 개의 측정값이 들어가는가
```

MEDIAN_NUM : 저장할 샘플 개수

float dist_meds[] : MEDIAN_NUM개의 샘플 저장

int index_meds : 현재 몇 개의 샘플이 저장되어있는지 저장 (최댓값 MEDIAN_NUM)

2. 중위수 필터 구현

```
int Compare(const void* temp1, const void* temp2){
    float a = *((float *)temp1);
    float b = *((float *)temp2);

    return (int)(a - b);
}

float FilterOfMed(float raw){
    if(index_meds == MEDIAN_NUM) {
        for(int i = 0; i < MEDIAN_NUM - 1; i++){
            dist_meds[i] = dist_meds[i + 1];
        }
        dist_meds[MEDIAN_NUM - 1] = raw;
    }
    else dist_meds[index_meds++] = raw;

    float *sorted_meds = (float *)malloc(sizeof(float) * index_meds);
    for(int i = 0; i < index_meds; i++){
        sorted_meds[i] = dist_meds[i];
    }

    qsort(sorted_meds, index_meds, sizeof(sorted_meds[0]), Compare);
    float median = sorted_meds[index_meds / 2];
    free(sorted_meds);
    return median;
}
```

FilterOfMed(float)

1. 현재 들어있는 샘플 값이 최댓값인지 아닌지 판단

만약 최댓값이라면, 가장 처음에 들어온 값을 제외하고 현재 들어온 값을 저장한다.

선입선출구조

만약 아니라면 index_meds번째에 값을 저장 후 index_meds++

2. 새로운 배열 동적 생성 후 정렬하기

C에서 제공하는 qsort() 함수를 사용

```
void qsort(  
    void *base,  
    size_t number,  
    size_t width,  
    int (__cdecl *compare)(const void *, const void *)  
);
```

4번째 매개변수엔 return 값으로 다음과 같은 값이 반환되어야함

비교 함수 반환 값	설명
< 0	elem1 이 elem2 보다 작음
0	elem1 이 elem2 와 같음
> 0	elem1 > elem2

이 조건에 맞는 Compare(const void*, const void*) 함수 구현

3. 중위수 값 반환