

System Programming and Practice

-Lab13-

이름 : 김태석

학번 : 201721083

학과 : 미디어학과

```
double distance = 0;
int state = 1;
int prev_state = 1;
int serv_sock, clnt_sock = -1;
struct sockaddr_in serv_addr, clnt_addr;
socklen_t clnt_addr_size;
char msg[4];

state = GPIORead(PIN_1);

if(prev_state == 0 && state == 1){
    //light = (light+1)%2;
    //snprintf(msg, 2, "%d", light);
    sprintf(msg, "%d", (int)distance);
    write(clnt_sock, msg, sizeof(msg));
    printf("distance = %dcm\n", (int)distance);
}

prev_state = state;
```

```
int main(int argc, char *argv[]){
    if(-1 == GPIOExport(PIN_1) || -1 == GPIOExport(POUT_1))
        return(1);

    if(-1 == GPIODirection(PIN_1, IN) || -1 == GPIODirection(POUT_1, OUT))
        return(2);

    if(-1 == GPIOWrite(POUT_1, 1))
        return(3);

    if(argc != 2){
        printf("Usage : %s <port>\n", argv[0]);
    }

    serv_sock = socket(PF_INET, SOCK_STREAM, 0);
    if(serv_sock == -1)
        error_handling("socket() error");

    memset(&serv_addr, 0, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(atoi(argv[1]));

    if(bind(serv_sock, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) == -1)
        error_handling("bind() error");

    if(listen(serv_sock, 5) == -1)
        error_handling("listen() error");

    if(clnt_sock < 0){
        clnt_addr_size = sizeof(clnt_addr);
        clnt_sock = accept(serv_sock, (struct sockaddr*)&clnt_addr, &clnt_addr_size);
        if(clnt_sock == -1)
            error_handling("accept() error");
    }
}
```

코드 설명

버튼과 초음파 센서는 기존 실습에서 진행한 예제와 동일하게 GPIO 를 통해 입출력을 제어 해 주었다. LED 의 밝기는 PWM 을 이용하여 입출력을 제어 했고 초음파 센서를 통해 얻은 값을 LED 의 밝기 조절에 쓰기 위해 Pthread 를 사용했다. 또한 쓰레딩을 활용하기 위해 distance 변수를 전역 변수로 선언해 주었던 것처럼 소켓을 제어하거나 소켓을 통해 보내는 메시지 변수를 전역 변수로 선언해 주어 main 함수 이외에 다른 함수에서도 사용할 수 있도록 하였으며, 이는 초음파 센서와 관련된 코드인 ultrawave_thd 함수에서 버튼이 눌렸을 때 그 시점에서의 distance 값을 정수값으로 변환한 후 소켓을 통해 write 함과 동시에 server 를

실행한 터미널에 출력되도록 사용되었다. Double 로 선언된 distance 값을 정수형으로 바꿔준 이유는 자릿수를 일치 시킴과 동시에 msg 의 크기를 줄이기 위함이다.

실행 결과

```
pi@raspberrypi:~ $ ./LAB11_client 127.0.0.1 8888
Receive message from Server
DISTANCE : 190cm
Receive message from Server
DISTANCE : 4cm
```

-클라이언트 실행파일 실행 결과-

버튼을 클릭할 경우 버튼 클릭 시점의 초음파센서와 물체간 거리값을 서버로부터 받고 터미널에 띄운다.

```
pi@raspberrypi:~ $ ./lab13 8888
Hi
distance = 190cm
distance = 4cm
```

-서버 파일 실행 결과-

클라이언트에서 버튼을 통해 초음파센서와 물체간 거리값을 요청할 경우 값을 전송함과 동시에 터미널에 해당 값을 띄운다.

고찰

Pthread 를 사용해야 하기에 ultrawave_thd 함수에 argc, argv 인자를 전달하고 그 안에서 소켓에 관한 코드를 넣은 뒤 구동 해 보았지만, 쓰레딩이 작동하지 않는 문제가 발생했고 플래그를 찍어 확인 해 본 결과 ultrawave_thd 함수가 실행되고 함수 내의 while 문을 빠져나오지 않는 채 코드가 동작하고 있는 것을 확인했다. 때문에 소켓 통신에 필요한 변수들을 전역 변수로 선언 해준 후 main 함수에서 소켓을 열어주고 ultrawave_thd 에선 버튼이 입력 받았을 경우 소켓을 통해 값을 넘겨주는 기능만 넣어주어 해결 할 수 있었다.