

IoT프로그래밍

10조 팀프로젝트 6/9중간 발표

김남교 | 김보건 | 유진웅 | 한강빈

타이머 구현

```
unsigned char fnd_num[4] = {0,};
int fnds;
unsigned char Time_Table[] = {~0x3f, ~0x06, ~0x5b, ~0x4f, ~0x66, ~0x6d, ~0x7d, ~0x07, ~0x7f, ~0x67, ~0x00};

void update_display(int m, int s);

int main(){
    fnds = open(fnd, O_RDWR);
    if(fnds < 0){
        printf("Can't open FND.\n");
        exit(0);
    }

    int m = 3; // 시작 분 설정 (3분)
    int s = 0; // 시작 초 설정 (0초)

    while(m >= 0 && s >= 0){
        update_display(m, s);
        sleep(1);

        if(s == 0){
            if(m == 0){
                break;
            }
            m--;
            s = 59;
        } else {
            s--;
        }
    }

    // 시간 종료 후 00:00 표시
    update_display(0, 0);
    close(fnds);
    return 0;
}
```

```
void update_display(int m, int s){
    fnd_num[0] = Time_Table[m / 10]; // 분의 10의 자리
    fnd_num[1] = Time_Table[m % 10]; // 분의 1의 자리
    fnd_num[2] = Time_Table[s / 10]; // 초의 10의 자리
    fnd_num[3] = Time_Table[s % 10]; // 초의 1의 자리

    write(fnds, &fnd_num, sizeof(fnd_num));
    ...
}
```



초록색 = 분, 주황색 = 초

기능 설명

1. 타이머 기능으로, 7세그먼트 디스플레이를 제어하여 분과 초를 표시
2. Sleep() 함수를 이용해 1초마다 업데이트하여 남은 시간을 보여줌
3. 타이머가 끝나면 00:00을 표시하고 종료

장전 구현

```
#include <stdio.h>

void printMatrix(int matrix[8][8]) {
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            printf("%d", matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    // 8x8 DOT matrix 초기화
    int dotMatrix[8][8] = {0};

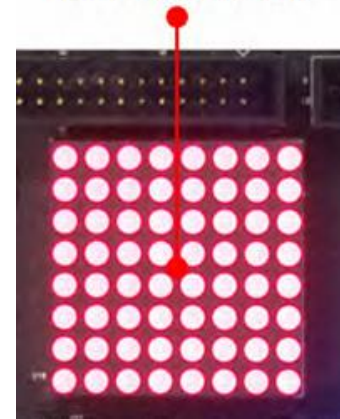
    // 5발의 가짜탄을 설정
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 8; j++) {
            dotMatrix[j][i] = 1;
        }
    }

    // 1발의 실탄을 설정 (6번째 열에 실탄을 놓음)
    for (int j = 4; j < 8; j++) {
        dotMatrix[j][5] = 1;
    }

    // DOT matrix 출력
    printMatrix(dotMatrix);

    return 0;
}
```

Dot Matrix 8x8

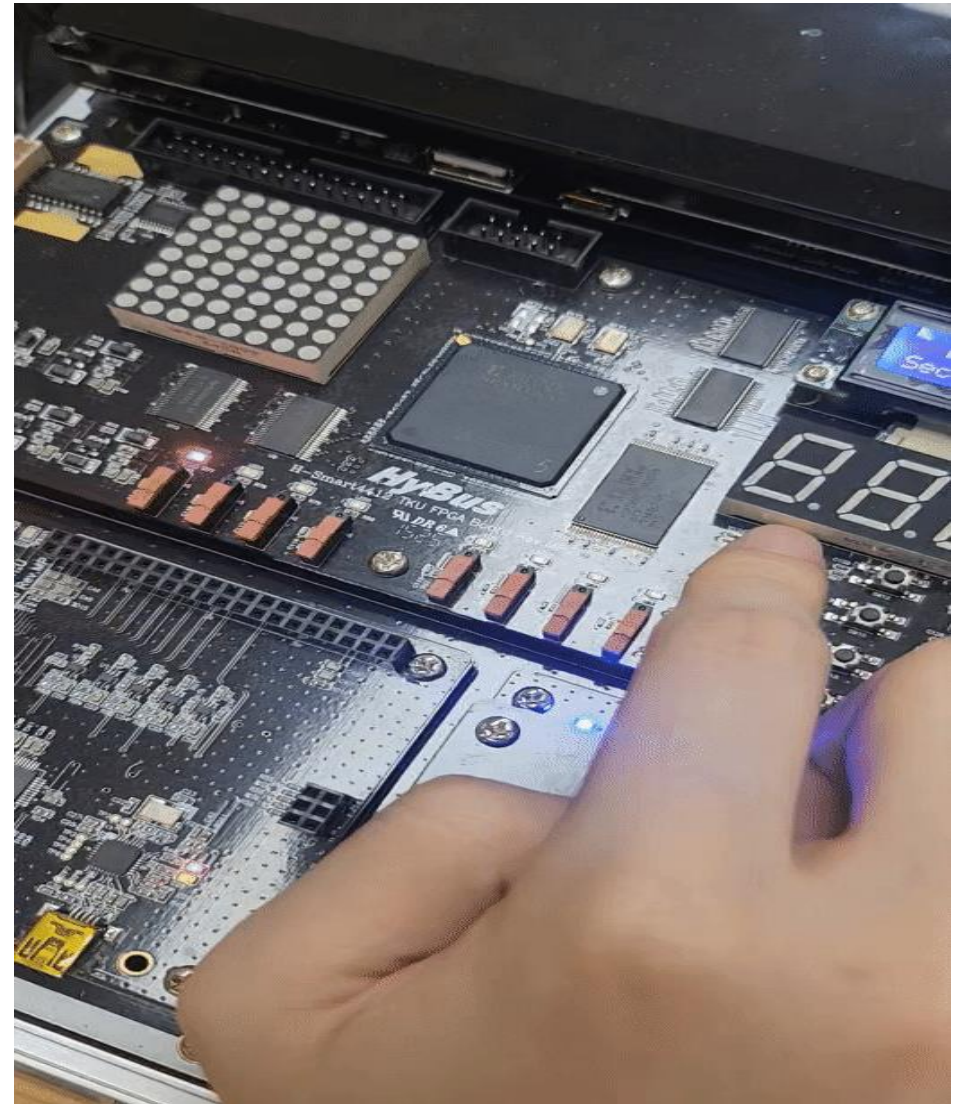


기능 설명

1. 8x8에 왼쪽부터 6탄의 탄이 표시
2. 불빛이 들어오는 각 열은 탄을 표현
3. 4칸만 불이 들어온 열 열은 실탄, 8칸에 모두 불이 들어온 탄은 가짜탄
4. 코드는 8x8 DOT matrix를 초기화하고, 5발의 실탄과 1발의 가짜탄을 설정한 후, 출력

플레이어 턴 변경(LED)

```
12 int main() {
13     int led_dev, tact_dev;
14     unsigned char led_data;
15     char tact_data[16]; // Tact switch 상태를 읽기 위한 버퍼
16     int player = 1; // 처음 시작한 플레이어 1의 차례
17
18     led_dev = open(LED, O_RDWR); // LED 장치 열기
19     if (led_dev < 0) {
20         printf("Can't open LED.\n");
21         exit(0); // 장치를 못 불러올 경우 예외 처리 후 종료
22     }
23
24     tact_dev = open(TACT_SWITCH, O_RDWR); // Tact switch 장치 열기
25     if (tact_dev < 0) {
26         printf("Can't open Tact switch.\n");
27         close(led_dev);
28         exit(0); // 장치를 못 불러올 경우 예외 처리 후 종료
29     }
30
31     while (1) {
32         if (player == 1) {
33             led_data = 0xFE; // 플레이어 1의 차례: 1번 LED만 OFF
34         } else {
35             led_data = 0xEF; // 플레이어 2의 차례: 5번 LED만 OFF
36         }
37
38         write(led_dev, &led_data, sizeof(unsigned char));
39         usleep(300000); // 0.3초 대기
40
41         // Tact switch 상태 읽기
42         read(tact_dev, &tact_data, sizeof(tact_data));
43         if (tact_data[0] & 0x01) { // Tact switch 1이 눌렸을 때
44             player = (player == 1) ? 2 : 1; // 플레이어 차례 변경
45             usleep(300000); // Debounce를 위해 잠시 대기
46         }
47     }
48 }
```



플레이어 턴 변경(LCD)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <fcntl.h>
5  #include <sys/types.h>
6  #include <sys/stat.h>
7  #include <sys/time.h>
8  #include <string.h>
9
10 #define CLCD "/dev/clcd"
11 #define TACT "/dev/tactsw"
12 |
13 int main() {
14     int clcd_dev, tact_dev;
15     char* player1_text = "Player 1";
16     char* player2_text = "Player 2";
17     unsigned char tact_data[2];
18     int player = 1; // 처음 시작한 플레이어 1의 차례
19     struct timeval start, end;
20     long elapsed_time;
21
22     clcd_dev = open(CLCD, O_RDWR); // CLCD 장치 열기
23     if (clcd_dev < 0) {
24         printf("Can't open CLCD.\n");
25         exit(0); // 장치를 못 불러올 경우 예외 처리 후 종료
26     }
27
28     tact_dev = open(TACT, O_RDONLY); // Tact switch 장치 열기
29     if (tact_dev < 0) {
30         printf("Can't open Tact switch.\n");
31         close(clcd_dev);
32         exit(0); // 장치를 못 불러올 경우 예외 처리 후 종료
33     }
34 }
```



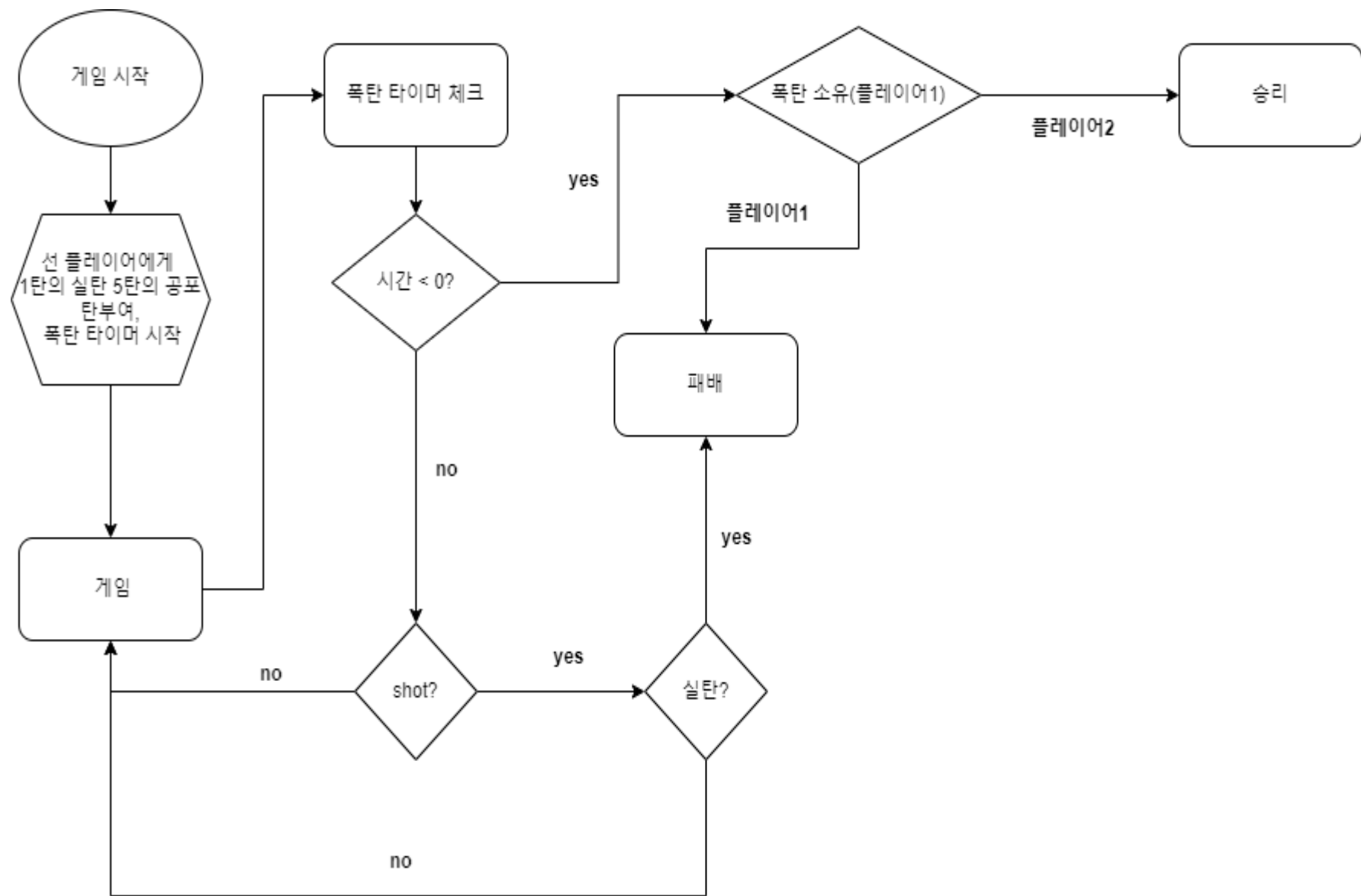
기능 설명

1. CLCD와 Tact switch 장치를 엽니다
2. 현재 플레이어에 따라 CLCD에 "Player 1 또는 Player 2를 표시합니다.
3. 타이머를 시작하고 내부 루프를 실행한다.
4. 내부 루프에서는 경과 시간을 체크하고, 5초가 경과하거나 또는 Tact switch 1이 눌리면 플레이어를 변경시킵니다.
5. 플레이어가 변경되면 메인 루프로 돌아가서 반복합니다.

```

34
35 while (1) {
36     // 현재 플레이어에 따라 CLCD 설정
37     if (player == 1) {
38         write(clcd_dev, player1_text, strlen(player1_text)); // 플레이어 1의 차례
39     }
40     else {
41         write(clcd_dev, player2_text, strlen(player2_text)); // 플레이어 2의 차례
42     }
43
44     // 타이머 시작
45     gettimeofday(&start, NULL);
46
47     while (1) {
48         // 현재 시간 체크
49         gettimeofday(&end, NULL);
50         elapsed_time = (end.tv_sec - start.tv_sec) * 1000 + (end.tv_usec - start.tv_usec) / 1000;
51
52         if (elapsed_time >= 5000) { // 5초가 경과하면
53             player = (player == 1) ? 2 : 1; // 플레이어 차례 변경
54             break;
55         }
56
57         // Tact switch 상태 읽기
58         read(tact_dev, &tact_data, sizeof(tact_data));
59         if (tact_data[0] & 0x01) { // Tact switch 1이 눌렸을 때
60             player = (player == 1) ? 2 : 1; // 플레이어 차례 변경
61             break;
62         }
63
64         usleep(100000); // 0.1초 대기 (폴링 주기)
65     }
66
67     // 루프 처음으로 돌아가서 다시 실행
68     continue;
69 }

```



감사합니다.