IOT프로그래밍

10조 최종발표

김남교 | 김보건 | 유진웅 | 한강빈

목차

1. 시연영상


러시안룰렛&폭탄돌리기
CLCD로 게임진행 표시 ▶
RUSSIAN ROULETTE
PRESS DIP SWITCH
8.8.8.8.
HyBus
임의의 DIP Switch를 켜서 게임시작 ▲

2. 구현설명 - 게임시작

```
#define DIP "/dev/dipsw"

int dipsw;
int clcds;
```

```
int FIRST_PRINT() {
    unsigned char d;            // Tact Switch 값 변수

    clcds = open(CLCD, O_RDWR);
    if (clcds < 0) { printf("Can't open Character LCD.₩n"); exit(0); }
    char pr_clcd[40] = "RUSSIAN ROULETTEPRESS DIP SWITCH! ";
    write(clcds, pr_clcd, strlen(pr_clcd));
    close(clcds);

    while (1) {
        dipsw = open(DIP, O_RDWR);
        if (DIP < 0) { printf("Can't open dip₩n"); exit(0); }
        read(dipsw, &d, sizeof(d));
        close(dipsw);

        if (d != 0) {
            return 0;
        }
    }
}
```

임의의 Dip Switch를 켜서 게임 시작

2. 구현설명 – 게임진행 표시

```c
// CLCD 출력 함수
int PRINT(char P[]) {
    clcds = open(CLCD, O_RDWR);
    if (clcds < 0) { printf("Can't open Character LCD.₩n"); exit(0); }
    write(clcds, P, strlen(P));
    close(clcds);
}
```

```c
clcds = open(CLCD, O_RDWR);
if (clcds < 0) { printf("Can't open Character LCD.₩n"); exit(0); }
char pr_clcd[40] = "RUSSIAN ROULETTEPRESS DIP SWITCH! ";
write(clcds, pr_clcd, strlen(pr_clcd));
close(clcds);
```

```c
while (1) {
    // 현재 플레이어에 따라 CLCD 설정
    if (player == 1) {
        write(clcd_dev, player1_text, strlen(player1_text)); // 플레이어 1의 차례
    }
    else {
        write(clcd_dev, player2_text, strlen(player2_text)); // 플레이어 2의 차례
    }
```

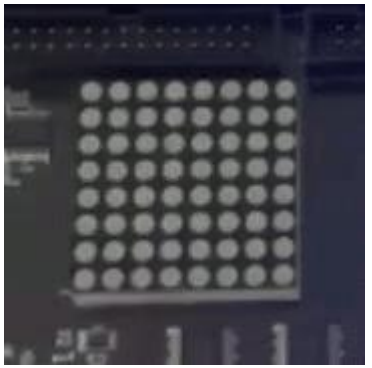
```c
void display_winner(int clcd_dev, int player) {
    char* win_text = (player == 1) ? "Player 2 Win" : "Player 1 Win";
    write(clcd_dev, win_text, strlen(win_text)); // 승리 메시지 출력
    sleep(3); // 승리 메시지를 3초간 표시
    exit(0); // 게임 종료
}
```

CLCD로 게임의 상황들을 표시하였다

2. 구현설명 – 발포

```
tact_dev = open(TACT, O_RDONLY); // Tact switch 장치 열기
if (tact_dev < 0) {
    printf("Can't open Tact switch.₩n");
    close(clcd_dev);
    exit(0); // 장치를 못 불러올 경우 예외 처리 후 종료
}
```

```
unsigned char pattern[8] = {  // 새로운 패턴 배열
    0xFF,  // 11111111
    0xFF,  // 11111111
    0xFF,  // 11111111
    0xFF,  // 11111111
    0xFF,  // 11111111
    0xF0,  // 11110000
    0x00,  // 00000000
    0x00   // 00000000
};

// 패턴 배열과 "sleep" 값을 인자로 받아서 도트 매트릭스를 제어하는 함수
void DOT_control(int time_sleep){
    int dot_d;

    dot_d = open(dot, O_RDWR);
    if (dot_d < 0) {
        printf("dot Error₩n");
        return; // 예외처리
    }

    write(dot_d, pattern, sizeof(pattern)); // 패턴 출력
    sleep(time_sleep); // 몇 초 동안 점등할지

    close(dot_d);
}

int main() {
    while (1) {
        DOT_control(3); // 새로운 패턴을 3초 동안 표시
    }

    return 0;
}
```



- Tact Switch로 발포
- 발포시 랜덤으로 Dot Matrix 가로줄 제거

2. 구현설명 – 타이머

```c
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;
}
```

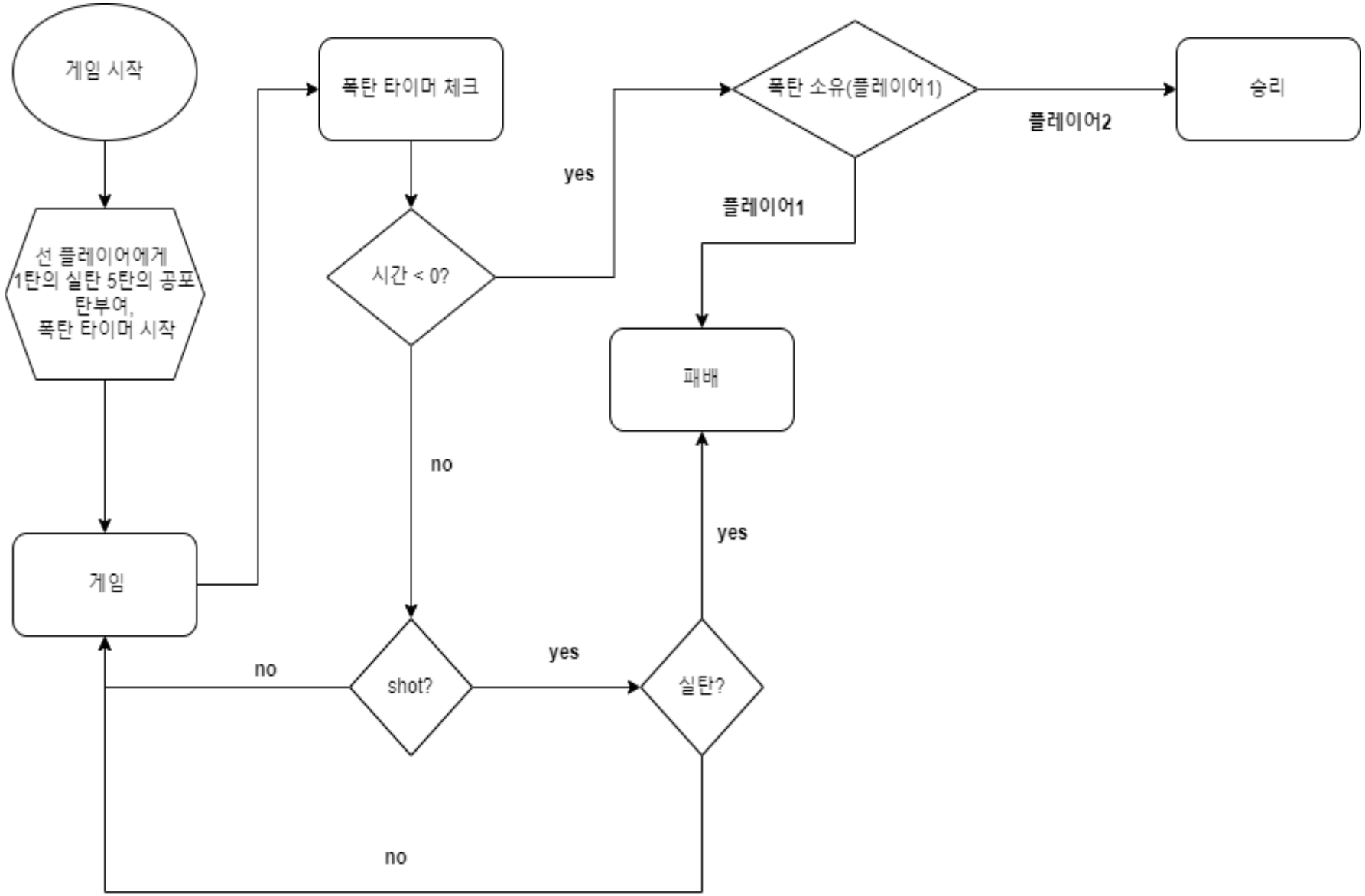
타이머가 0이 되면 폭탄이 터져 게임종료

2. 구현설명 – 턴

```
while (1) {
    // 현재 플레이어에 따라 CLCD 설정
    if (player == 1) {
        write(clcd_dev, player1_text, strlen(player1_text)); // 플레이어 1의 차례
    }
    else {
        write(clcd_dev, player2_text, strlen(player2_text)); // 플레이어 2의 차례
    }

    // 타이머 시작
    gettimeofday(&start, NULL);

    while (1) {
        // 현재 시간 체크
        gettimeofday(&end, NULL);
        elapsed_time = (end.tv_sec - start.tv_sec) * 1000 + (end.tv_usec - start.tv_usec) / 1000;

        if (elapsed_time >= 5000) { // 5초가 경과하면
            player = (player == 1) ? 2 : 1; // 플레이어 차례 변경
            break;
        }

        // Tact switch 상태 읽기
        read(tact_dev, &tact_data, sizeof(tact_data));
        if (tact_data[0] & 0x01) { // Tact switch 1이 눌렸을 때
            player = (player == 1) ? 2 : 1; // 플레이어 차례 변경
            break;
        }

        usleep(100000); // 0.1초 대기 (폴링 주기)
    }

    // 루프 처음으로 돌아가서 다시 실행
    continue;
}
```

5초가 경과하면 턴 변경
턴 변경시 0.1초의 대기시간을 부여하였다.(폴링)

3. 플로우차트 체크


게임 시작
선 플레이어에게
1탄의 실탄 5탄의 공포탄부여,
폭탄 타이머 시작
게임
폭탄 타이머 체크
시간 < 0?
yes
no
폭탄 소유(플레이어1)
플레이어2
승리
플레이어1
패배
shot?
no
yes
실탄?
yes
no

4. 참고자료 – 폭탄 해체 게임

택택 (tistory.com) – 폭탄 해체 게임

- 7-segment를 사용하여 타이머 활용
- 타이머가 0이 되면 폭탄이 터져 게임이 종료되는 요소를 채택하였다.

4. 참고자료 – 인디언 포커

H-Smart4412를 이용한 인디언 포커 게임 (velog.io)

- 플레이어 vs 플레이어 턴제 방식을 참고하였다.

4. 참고자료 – Chat GPT

```c
void random_change_pattern() {
    int index = rand() % 6; // 0부터 5까지의 랜덤 인덱스 생성
    pattern[index] = 0x00;  // 랜덤 인덱스의 값을 0x00으로 변경
}

int main() {
    int tact_dev;
    struct input_event ev; // Tact switch 이벤트를 위한 구조체
    fd_set readfds;
    struct timeval tv;
    int retval;

    srand(time(NULL)); // 랜덤 시드 설정

    tact_dev = open(TACT_SWITCH, O_RDWR); // Tact switch 장치 열기
    if (tact_dev < 0) {
        printf("Can't open Tact switch.\n");
        exit(0); // 장치를 못 불러올 경우 예외 처리 후 종료
    }
```

Chat GPT(https://chatgpt.com)

- 발포 구현 - 랜덤함수를 적용하여 Dot Matrix제어
- 그밖에 코드작성의 방향이나 자잘한 오류들을 해결하기 위해 참고하였다.

감사합니다.