
폭탄 해체 게임

- IoT프로그래밍 -

8조

2017250039 임형택

2017250042 정범진

작성 날짜 : **2022. 06. 05.**

목차

목차	2
1. 선 끊기 게임	3
① 매뉴얼	3
② 작동 원리	4
③ 실행 결과	5
④ 코딩 및 구현	5
2. 타이머 구현	6
① 작동 원리	6
② 실행 결과	6
③ 코딩 및 구현	6
3. Character-LCD 문자 출력 구현	7
① 작동 원리	7
② clcd.c 코드 및 코드 링크	7
4. 다음 발표까지 목표 및 과제	8
참고문헌 및 출처	9

1. 선 끊기 게임

① 매뉴얼

3개의 선:

초록 선이 없다면 두 번째 선을 끊습니다.

위 설명에 해당되지 않고 마지막 선이 파란색이라면, 마지막 선을 끊습니다.

위 설명에 해당되지 않고 빨간 선이 둘 이상이라면, 마지막 빨간 선을 끊습니다.

위 설명들 중 해당사항이 없다면 마지막 선을 끊습니다.

4개의 선:

붉은 선이 둘 이상이면 마지막 붉은 선을 끊습니다.

위 설명에 해당되지 않고 마지막 선이 초록색이며 붉은 선이 없다면 첫 번째 선을 끊습니다.

위 설명에 해당되지 않고 초록 선이 딱 하나만 있으면, 첫 번째 선을 끊습니다.

위 설명에 해당되지 않고 파란 선이 두 개 이상이면, 마지막 선을 끊습니다.

위 설명들 중 해당사항이 없으면 첫 번째 선을 끊습니다.

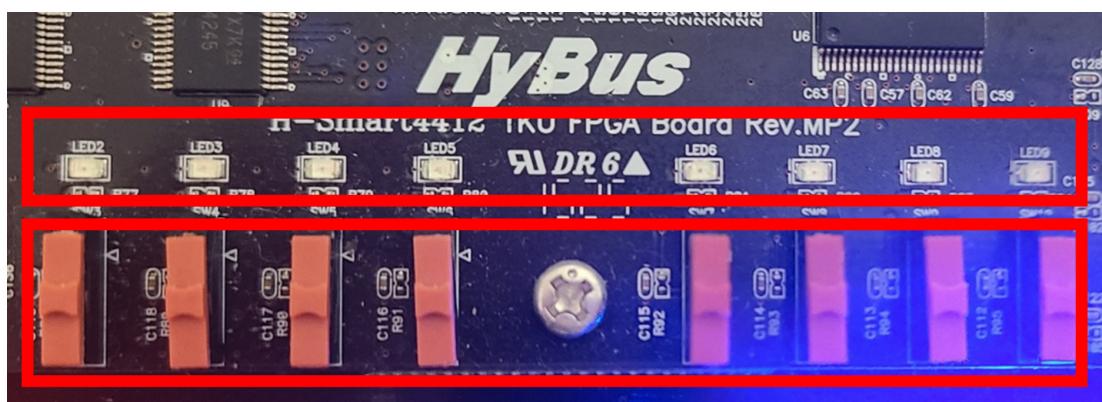
5개의 선:

마지막 선이 노란색이면 네 번째 선을 끊습니다.

위 설명에 해당되지 않고 붉은 선이 딱 하나만 있으며 초록 선이 두 개 이상이면, 첫 번째 선을 끊습니다.

위 설명에 해당되지 않고 파란선이 없으면, 두 번째 선을 끊습니다.

위 설명들 중 해당사항이 없다면 첫 번째 와이어를 자릅니다.

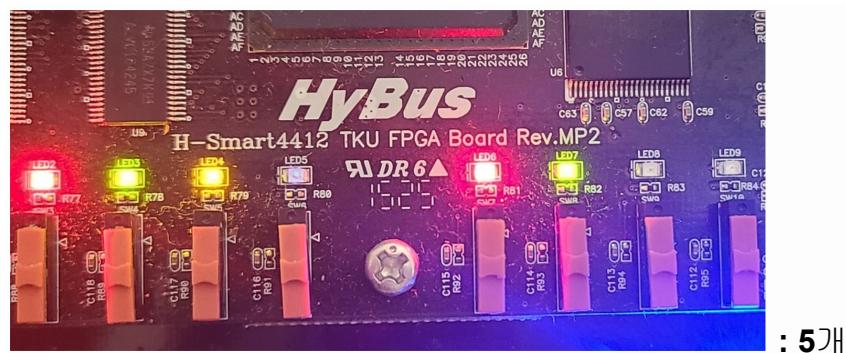
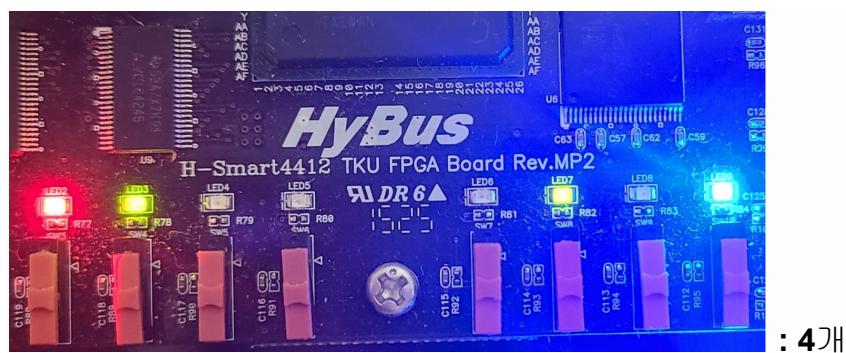
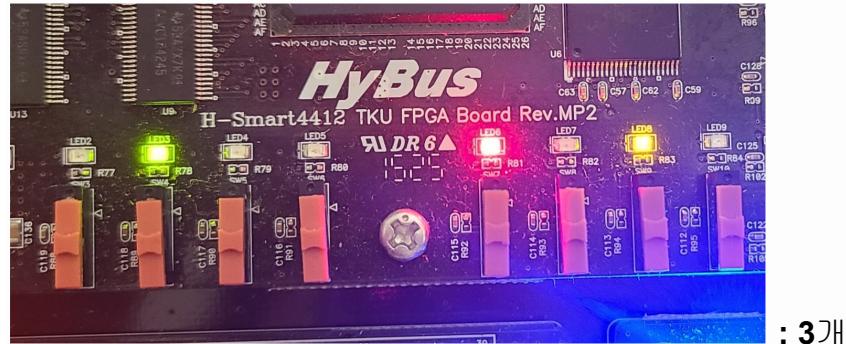


① **Chip LED** : 선 표현

② **Dip Switch** : 스위치를 올렸을 때, 선을 끊는 것으로 표현

① 작동 원리

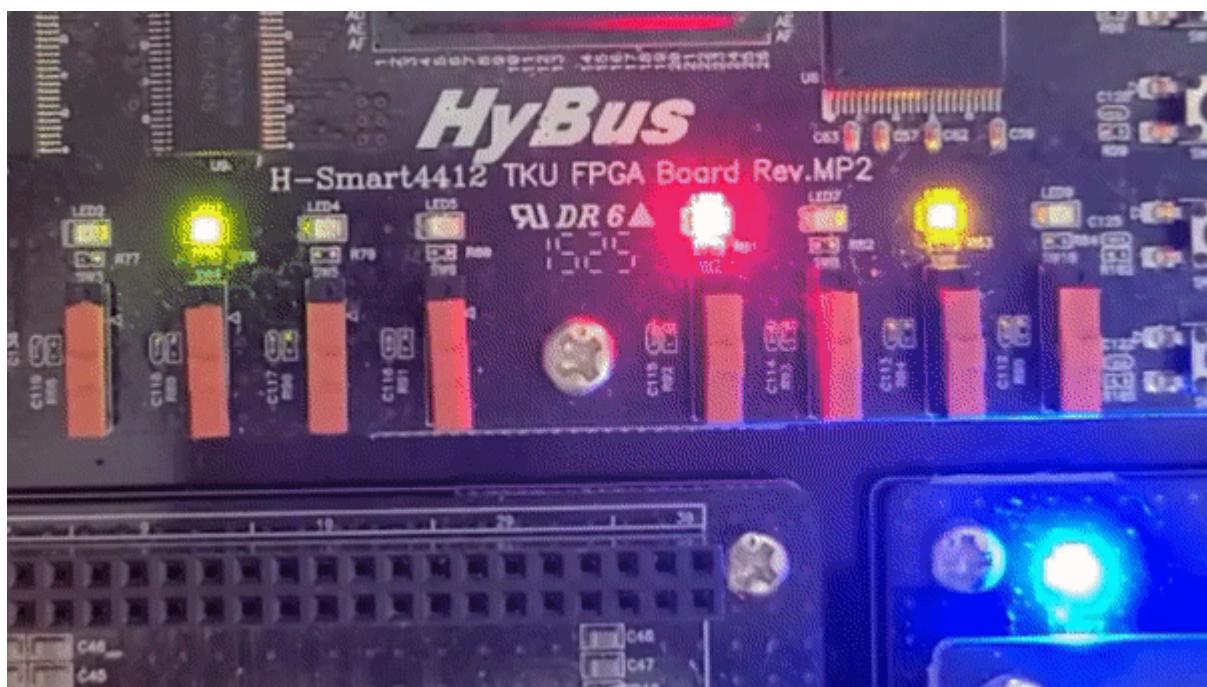
1. LED(선)가 랜덤으로 3~5개로 출력



2. 선의 개수에 해당하는 매뉴얼의 지시를 따라 Dip Switch를 올려 값 입력

3. True 값이면 Game Clear, False 값이면 Game Over

④ 실행 결과



⑤ 코딩 및 구현

https://github.com/hyoungteak/IoT_HSmart4412/blob/main/Codes/Game3.c

2. 타이머 구현

① 작동 원리



- (1) 타이머 기능 제작 후 **7 - Segment**를 이용하여 숫자 출력
- (2) ①과 ②로 나누어서 ①은 분 단위를, ②는 초 단위를 표현
- (3) **Time.c - sleep()** 함수를 이용한 단순 타이머 제작
- (4) 타이머의 시간이 0이 되면 실행 종료

② 실행 결과

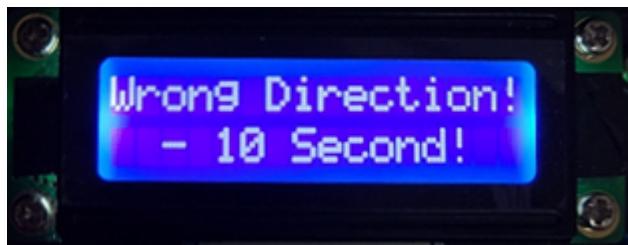


③ 코딩 및 구현

https://github.com/hyoungteak/IoT_HSmart4412/blob/main/Codes/time.c

3. Character-LCD 문자 출력 구현

① 작동 원리



(1) Char 형식의 문자열 타이핑 후 Character LCD에 문자열 출력

㉡ clcd.c 코드 및 코드 링크

```
11 #define clcd "/dev/clcd"
12 #define tact "/dev/tactsw"
13 int clcds;
14 int tactsw;
15 unsigned char t;
16
17 //int FIRST_PRINT(){
18 int main(){
19     clcds = open(clcd, O_RDWR);
20     if(clcds < 0){printf("Can't open Character LCD.\n"); exit(0);}
21     char pr_clcd[40] = "Wrong Direction! - 10 Second! ";
22     write(clcds, pr_clcd, strlen(pr_clcd));
23     close(clcds);
24
25     while(1){
26         tactsw = open(tact, O_RDWR);
27         if (tact < 0) {printf("Can't open tact\n"); exit(0);}
28         read(tactsw, &t, sizeof(t));
29         close(tactsw);
30
31         if (t != 0){
32             return 0;
33         }
34     }
35 }
```

1. 타겟 시스템의 **Tact Switch**와 **Character LCD** 장치 불러오기
(불러오지 못할 경우 예외처리)
2. 40 크기의 **char** 형 **pr_clcd** 선언
3. **wirte()** 함수로 출력
4. **Tact Switch**의 처음 값 0을 제외한 다른 값이 입력이 될 경우 코드 종료
(Link:https://github.com/hyoungteak/IoT_HSmart4412/blob/main/Codes/time.c)

4. 다음 발표까지 목표 및 과제

1. **main.c** 구현 : 구현해 놓은 세 가지 게임과 타이머를 **main.c**에 통합
2. **Character LCD** 응용 : **main.c**에서 **Game Clear, Over, Guide, Penalty** 표현

참고문헌 및 출처

1. Keep Talking and Nobody Explodes 매뉴얼

<https://www.bombmanual.com/ko/print/KeepTalkingAndNobodyExplodes-BombDefusalManual-v1-ko.pdf>

2. array-equal 함수

<https://patiencelee.tistory.com/667>

3. Dot Matrix 점멸하기(어떻게 사용되는지 원리 참고)

<https://hongci.tistory.com/96>

4. Snake 게임 Tact switch 부분(어떻게 사용되는지 원리 참고)

<https://github.com/jinwoo1225/SnakeGameWithSmart4412/blob/main/Snake/TactSW.cpp>

5. time 모듈 참고

<https://makerejoicegames.tistory.com/99>