
프로젝트 최종 발표

8팀

안병한 이상민 이정민 장민석

목차

- 1 프로젝트 구현 과정
- 2 프로젝트 시연
- 3 과정 평가

타겟 시스템 구현

테라텀을 이용하여
타겟 시스템 구현

Dot Matrix <-> tact sw

Dot Matrix와 tact
switch 간의
상호작용을 구현

폭탄 추가 결정

아이템 대신 두더지와
헛갈리게 하는 요소로
폭탄을 추가하기로 결정

DIP LED 구현

DIP LED가 켜있는 상태를
기본적으로 하여 스위치를
올리면 꺼지는 방식 구현

기본 두더지 잡기
코드 구현

두더지가 생성된 DotMatrix
의 위치를 누르면 두더지가
잡히는 방식을 구상

아이템 삭제

기존에 만들었었던
아이템의 오류로 인해
삭제 결정

DIP 스위치를 통한 난이도 조정

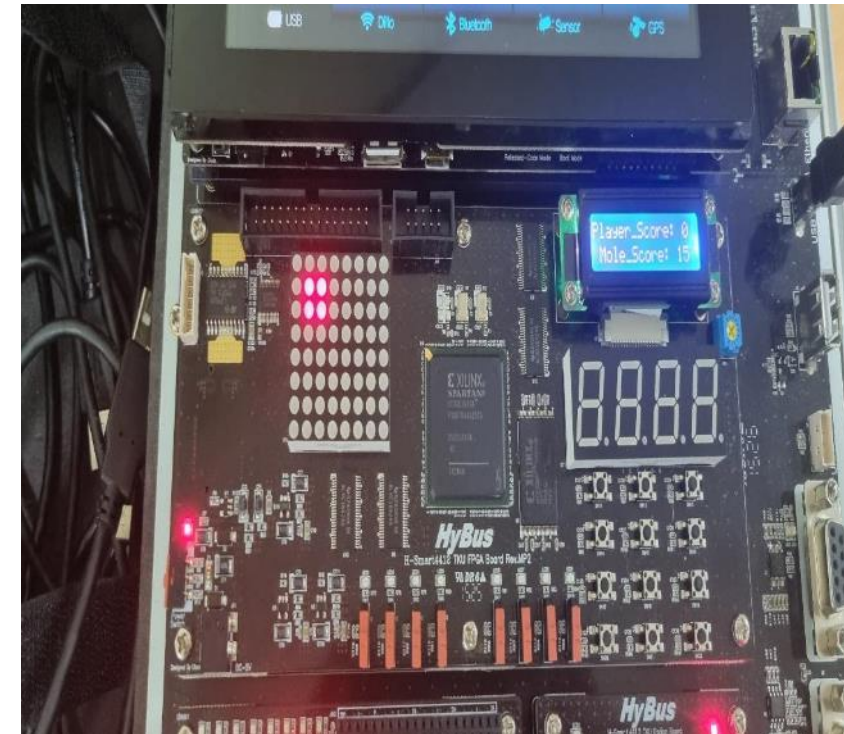
추가적으로 DIP 스위치를 통해
서 난이도를 조정할 수 있게 함

타겟 시스템 구현

테라텀을 이용하여
타겟 시스템을 구현함

```
CPU2: Booted secondary processor
Switched to NOHz mode on CPU #2
main 53
[root@linux /]# [root@linux /]# ls
Mole_CLCD_Item  home/      mnt/      sbin/
Mole_Game_main lib/        opt/      share/
bin/            linuxrc@   proc/     sys/
dev/            lost+found/ root/     tmp/
etc/            media/     run@      usr/
[root@linux /]# [root@linux /]#
```

```
Mole_CLCD_Item  home/      mnt/      sbin/
Mole_Game_main lib/        opt/      share/
bin/            linuxrc@   proc/     sys/
dev/            lost+found/ root/     tmp/
etc/            media/     run@      usr/
[root@linux /]# [root@linux /]# ./Mole_CLCD_Item
sh: ./Mole CLCD Item: Permission denied
[root@linux /]# chmod 777 Mole_Game_main
[root@linux /]# ./Mole_Game_main
두더지 점수 : 1
두더지 점수 : 2
두더지 점수 : 3
두더지 점수 : 4
두더지 점수 : 5
□
```



Part 1, 기본 두더지 잡기 코드 구현(Dot Matrix <-> tact switch)

코드 구현

```
void clcd_input2(int score_player, int score_mole)
{
    int clcd;
    char player_score[5];
    char mole_score[5];
    char clcd_text1[30] = "Player_Score: ";
    char clcd_text2[30] = "Mole_Score: ";
```

clcd_input2

CLCD 장치를 연 후, 점수를 문자열로 변환 및 조합함. 이후 CLCD에 출력함.

```
// 두더지 맞혔는지 검사
if (num == random_mole) {
    // 폭탄 두더지를 클릭한 경우
    if (bomb_mole_index == random_mole - 1) {
        is_bomb = 1;
        score_player -= 2; // 점수 2점 차감
        printf("폭탄을 클릭했습니다! 점수: %d\n", score_player);
```

startmole

1일 시, 현재 스테이지의 두더지 위치를 도트 매트릭스에 표시함.

```
// 두더지 못잡았을 경우
timeleft++;
if (timeleft > stagetime) {
    // 폭탄이 아닌 경우에만 두더지 점수 증가
    if (bomb_mole_index != random_mole - 1) {
        score_mole += 1;
        clcd_input2(score_player, score_mole);
        printf("두더지점수: %d \n", score_mole);
    }
}
```

```
while (1) {
    // 두더지 생성
    if (startmole == 1) {
        startmole = 0;
        random_mole = (rand() % 9) + 1;
        timeleft = (rand() % 10);
```

두더지가 생성된 DotMatrix의 위치의
tact switch를 누르면
두더지가 잡히는 방식 구현

폭탄 구현

```
int main() {  
    int bomb_mole_index;  
    int is_bomb = 0;  
    int dip;  
    unsigned char dipval = 0;  
  
    int dot = 0;  
    int tact = 0;  
    int num, isStop = 0;  
    int random_mole = 0;  
    int timeleft = 0;  
    int startmole = 1;  
    int stagetime = 20;  
    int bomb_probability = 3;
```

bomb_mole_index

bomb_probability 변수에 따라 폭탄 생성
확률이 결정됨

생성된 폭탄의 인덱스를 bomb_mole_index에 저장하
고, 폭탄을 클릭한 경우 2점을 차감함

폭탄을 맞춘 경우, 점수를 감소시키며
다음 폭탄을 재설정함

DIP 스위치 코드 구현

```
if (dipval != 0) {  
    // DIP 스위치 값에 따라 난이도 설정  
    if (dipval & 0x01) {  
        stagetime = 40; // DIP 스위치 1번 ON: 난이도 1 (가장 쉬움)  
    } else if (dipval & 0x02) {  
        stagetime = 35; // DIP 스위치 2번 ON: 난이도 2  
    } else if (dipval & 0x04) {  
        stagetime = 30; // DIP 스위치 3번 ON: 난이도 3  
    } else if (dipval & 0x08) {  
        stagetime = 25; // DIP 스위치 4번 ON: 난이도 4  
    } else if (dipval & 0x10) {  
        stagetime = 20; // DIP 스위치 5번 ON: 난이도 5  
    } else if (dipval & 0x20) {  
        stagetime = 15; // DIP 스위치 6번 ON: 난이도 6  
    } else if (dipval & 0x40) {  
        stagetime = 10; // DIP 스위치 7번 ON: 난이도 7  
    } else if (dipval & 0x80) {  
        stagetime = 5; // DIP 스위치 8번 ON: 난이도 8 (가장 어려움)  
    }  
}
```

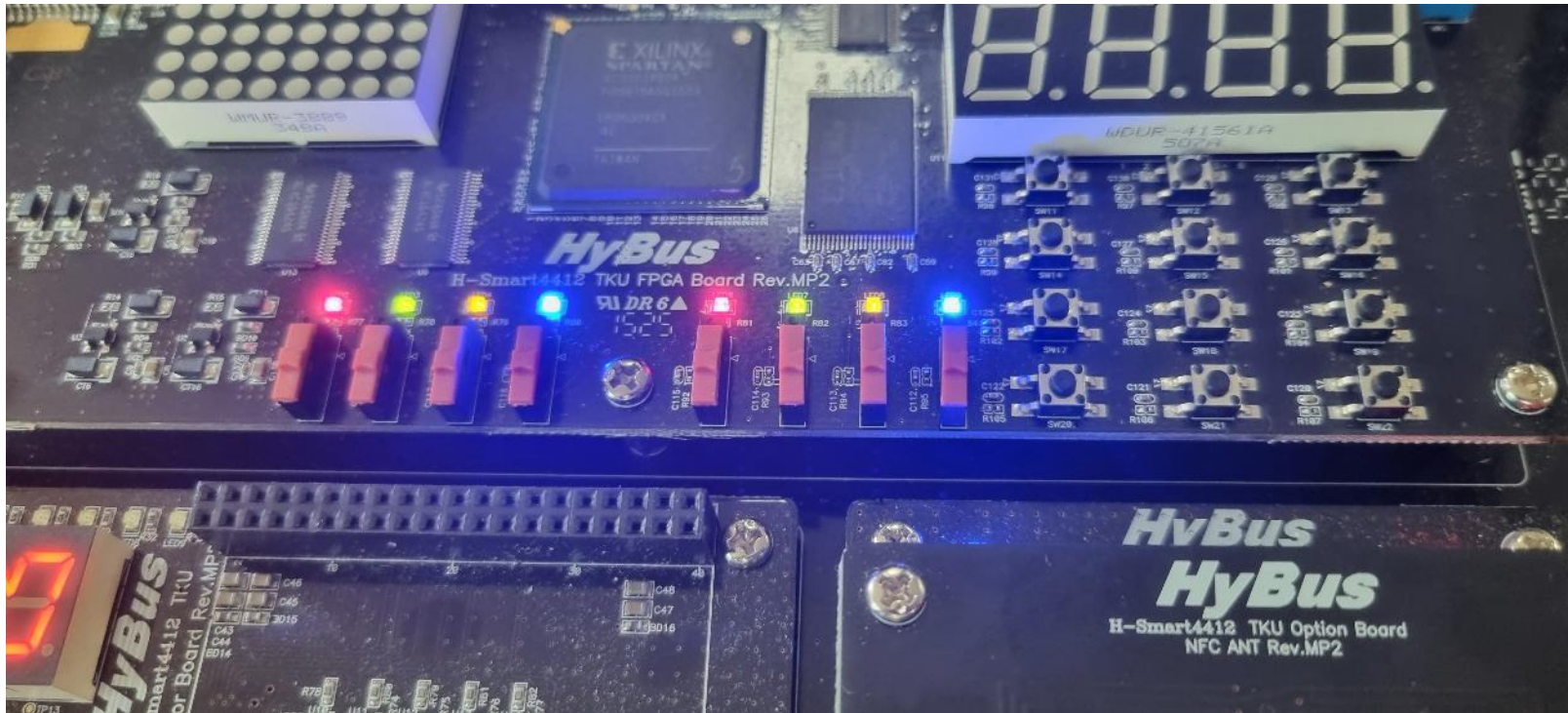
dipval

DIP 스위치의 값을 저장하는 변수임.
각각의 스위치가 ON 또는 OFF 상태를
나타내면, 난이도가 조정됨.

첫 번째 스위치가 켜져 있으면
난이도가 1, 두 번째 스위치가
켜져 있으면 2와 같은 방식으로
DIP 스위치를 통한 난이도 조절

Part 1, DIP LED 구현

DIP LED 구현



일반적으로 DIP LED가 모두 켜져
있는 상태에서 DIP 스위치를
올리면 LED가 꺼지도록 변경

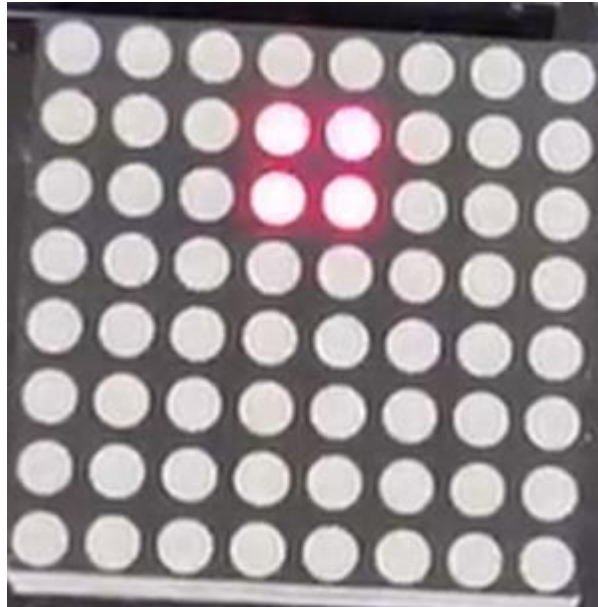
시연 영상

<https://youtu.be/MJYvU7mAKJ8>

두더지, 폭탄의 여부

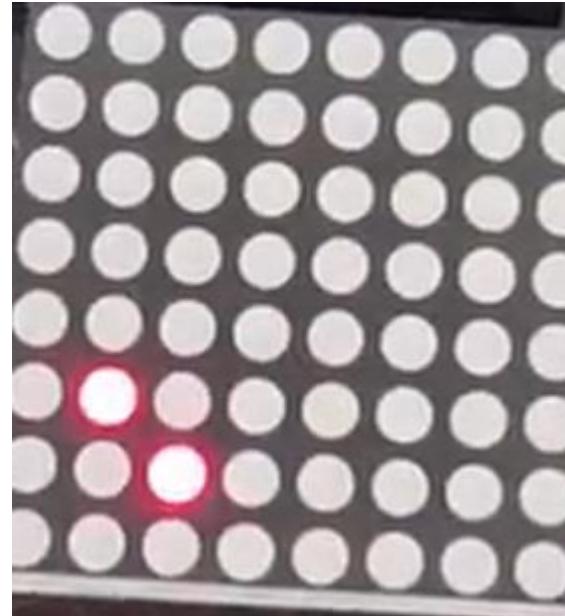
두더지

Dot Matrix가
정사각형 모양으로
4개가 모여 있으면
두더지



두더지, 폭탄

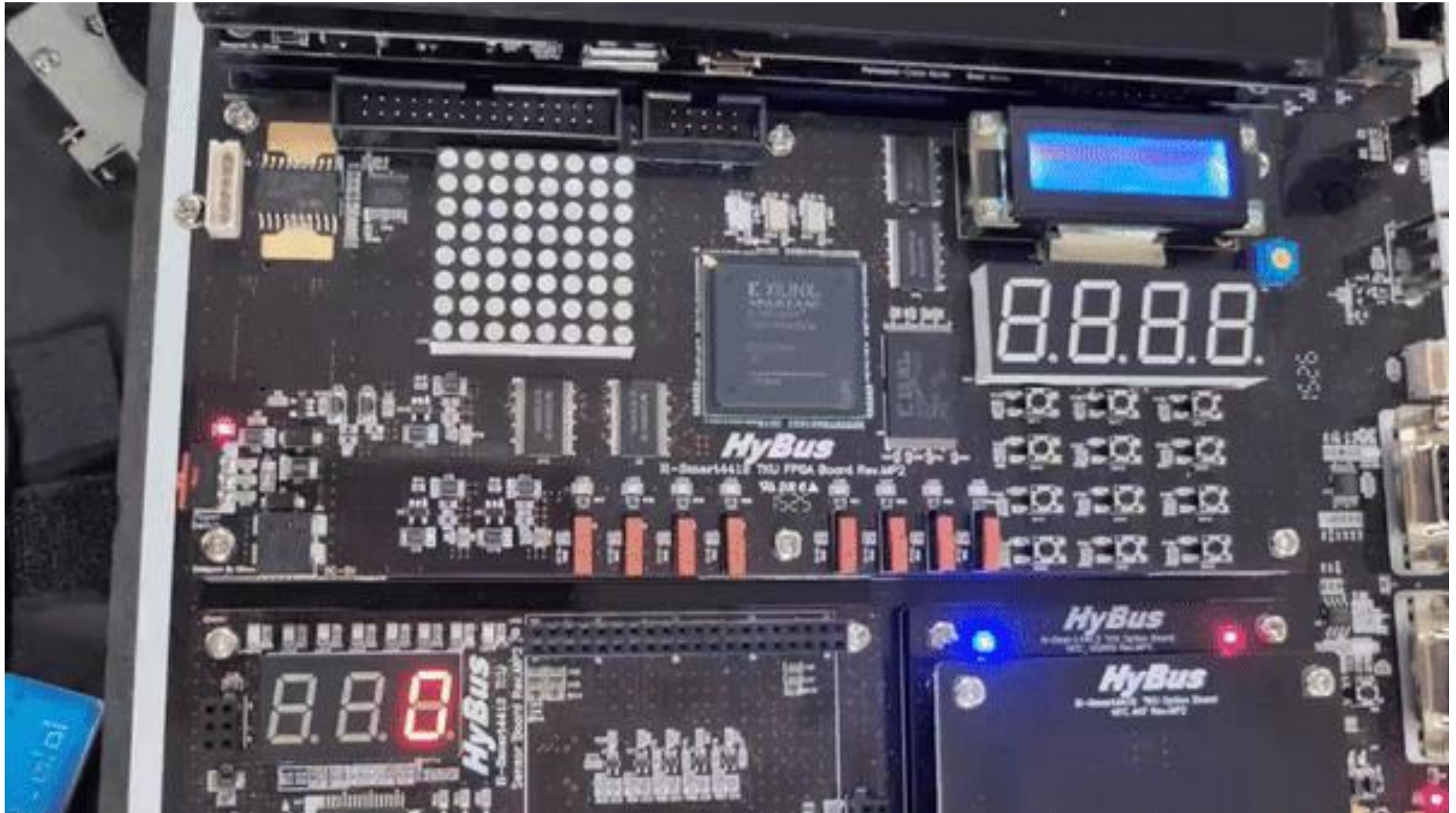
Dot Matrix가
대각선으로
2개가 모여 있으면
폭탄



난이도 조정

DIP 스위치

DIP 스위치를 통해
난이도를 조정함

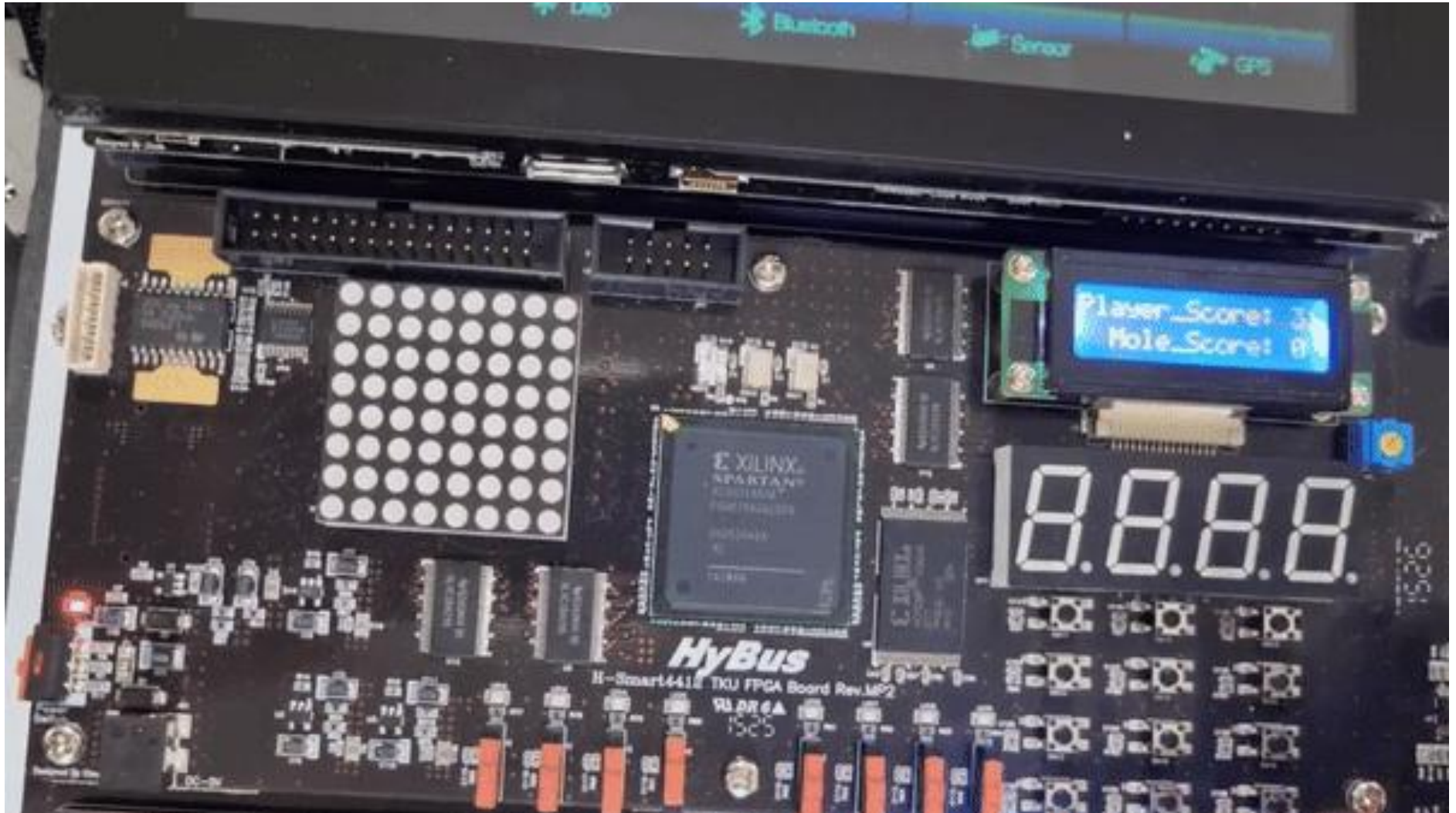


Part 2, 프로젝트 시연

게임 종료

최종 점수

Player_Score가 31점이
되면 게임이 종료됨



SWOT 기법을 통한 과정 평가

- 조원과의 원활한 의사소통
- 빠른 대처를 통해 부족한 부분 보완 및 개선

S, 강점

W, 약점

- 많지 않았던 인터넷 내의 자료 수
- 기술적인 코딩 능력의 난이도 증가

- Chat GPT 등과 같은 AI. 기술의 사용

O, 기회

T, 위협

- 아이템의 오류로 인한 삭제 결정
- 마이크 드라이버 적용 어려움으로 인한 삭제

참고 자료

<https://takethat.tistory.com/21>

타겟 시스템 시뮬레이션

<https://slidesplayer.org/slide/14109337/>

H-Smart4412TKU FPGA Board 사진

<https://github.com/2022HKNUiottprogrammingTeam1/project>

두더지 잡기 프로젝트

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=bieemiho92&logNo=220744722833>

VHDL로 구현한 두더지게임

수행 역할

2019250027 안병한

두더지 게임 순서도 작성,
폭탄 기능 코드 작성

2019250036 이상민

하드웨어 소프트웨어 통합 및 테스트,
시스템 통합 및 연동,
코드 최적화 및 디버깅,
임베디드 시스템 관련 코드 설계 구현

2019250041 이정민

자료 조사,
두더지 게임 기본 룰 구상

2020740067 장민석

보고서 작성, 발표,
ppt 작성, 자료 조사

“

감사합니다

”