

---

# 프로젝트 최종 발표

---

8팀

안병한 이상민 이정민 장민석

# 목차

**1** 프로젝트 구현 과정

**2** 프로젝트 시연

## 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

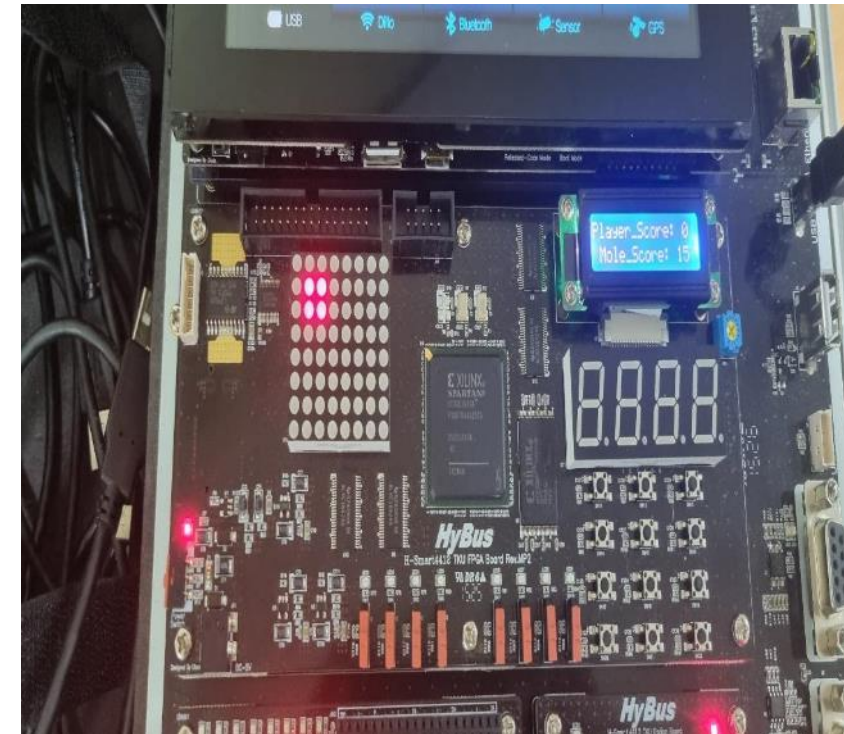
추가적으로 DIP 스위치를 통해  
서 난이도를 조정할 수 있게 함

## 타겟 시스템 구현

타겟 시스템  
출력 화면을  
확인함

```
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)

```
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: ";
```

## void clcd\_input2

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

## startmole

1일 시, 현재 스테이지의 두더지 위치를 도트 매트릭스에 표시함.

```
//스테이지 시작시 매트릭스에 두더지 표시
if (startmole == 1)
{
    startmole = 0;
    for (q = 0; q < stage; q++)
```

## isNext

변수가 1이면 새로운 스테이지의 두더지 위치와 시간을 설정함.

## nextmole

1일 시, 새 두더지 위치를 추가함.

```
if (nextmole == 1)
{
    nextmole = 0;
    for (p2 = 0; p2 < 8; p2++)
```

두더지가 생성된 DotMatrix의 위치의 tact  
switch를 누르면  
두더지가 잡히는 방식 구현

## Part 1, 폭탄 추가 결정

```
// 폭탄을 맞춘 경우
if (random[z] > 10) {
    if (num == random[z] - 10)
    {
        count++;
        score_player -= 2;
        printf("점수 : %d\n", score_player);
        clcd_input2(score_player, score_mole);

        for (x1 = 0; x1 < 8; x1++)
        {
            printmatrix[x1] = printmatrix[x1] - bomb[num-1][x1];
        }

        timeleft[z] = rand() % 10;
        num = 0;
        renumber = (rand() % 11) + 1;
        if (renumber > 9) {
            isBomb = 1;
            renumber = (rand() % 9) + 1;
        }
    }
}
```

### isBomb

해당 변수는 폭탄의 여부를 나타냄  
1인 경우 폭탄을 클릭한 것이고,  
0인 경우 폭탄 아닌 두더지를 클릭.

폭탄 클릭 시, 해당 두더지의 번호를 표시하기 위해  
10을 더한 값으로 표현됨.

전체적으로 폭탄 클릭의 여부를 추적하는데 사용됨.

폭탄을 맞춘 경우, 점수를 감소시키며  
다음 폭탄을 재설정

# Part 1, DIP 스위치를 통한 난이도 조절

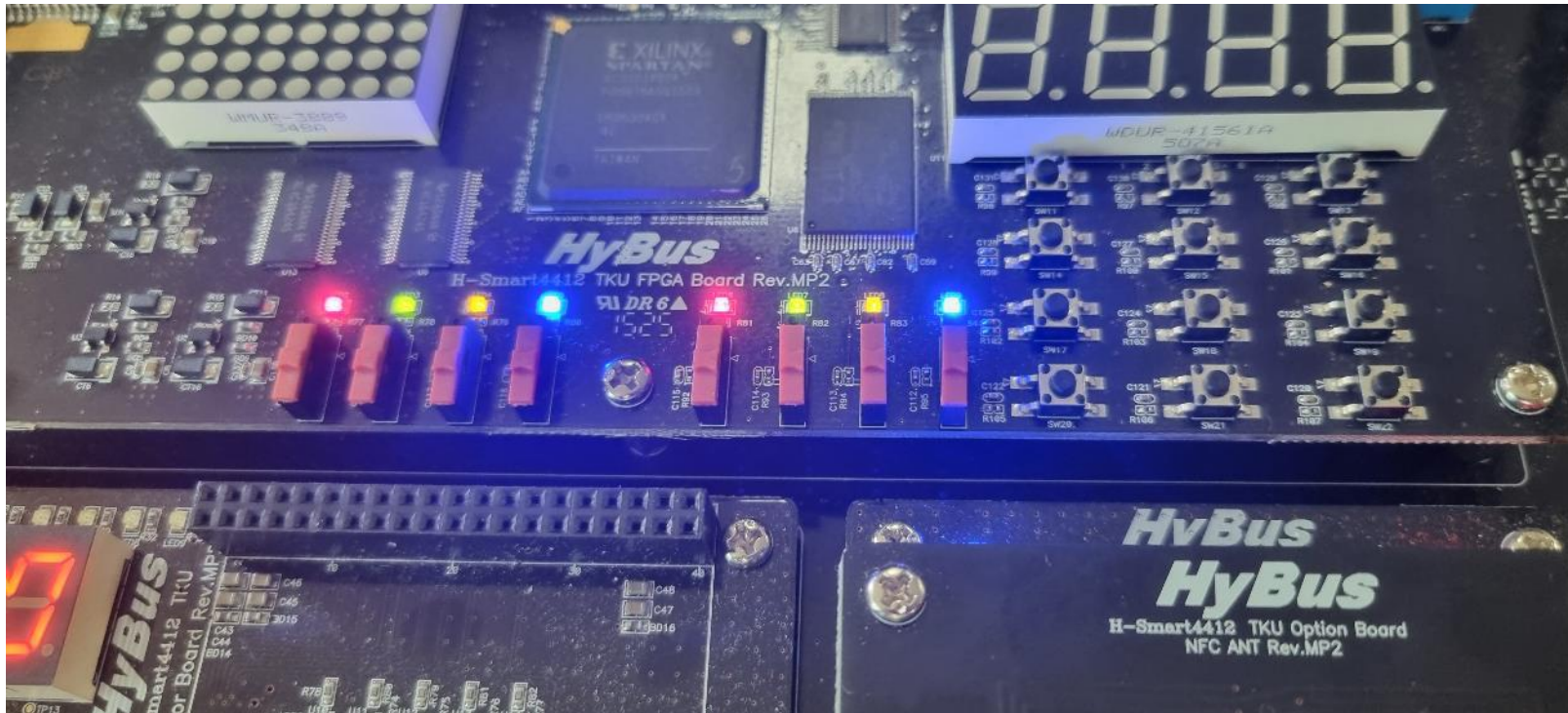
```
if (dipval != 0) {  
    // DIP 스위치 값에 따라 난이도 설정  
    if (dipval & 0x01) {  
        stagetime = 40; // DIP 스위치 1번 ON: 난이도 1 (가장 쉬움)  
    } else if (dipval & 0x02) {  
        stagetime = 35; // DIP 스위치 2번 ON: 난이도 2  
    } else if (dipval & 0x04) {  
        stagetime = 30; // DIP 스위치 3번 ON: 난이도 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



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



---

Part 2, 프로젝트 시연

---



# 참고 자료

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

타겟 시스템 시뮬레이션

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

H-Smart4412TKU FPGA Board 사진

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

VHDL로 구현한 두더지게임

“

감사합니다

”