
프로젝트 중간 발표 2

8팀

목차

- 1** 코드 구현 구상
- 2** 컴파일 및 타겟 시스템 전송-출력
- 3** 임베디드 실행

Part 1, 코드 구현 구상

두더지 생성 1

DotMatrix에
두더지를 랜덤하게
생성하도록 함

두더지 생성 2

2초를 간격으로
한 마리 씩
나오도록 함

Dot Matrix <-> tact sw 1

tact sw와 Dot Matrix가
연동되도록 코드를 작성

Dot Matrix <-> tact sw 3

12번 버튼을 누르면 종료

두더지 생성 3

1~9번 버튼을 누르면 Dot
Matrix의 위치에 맞는
두더지가 잡히고
다음으로 넘어가게 됨

Dot Matrix <-> tact sw 2

1~9번 버튼을 누르면 Dot
Matrix의 위치에
맞는 두더지가 잡히고
다음으로 넘어가게 함

Dot Matrix <-> tact sw 4

10번 버튼을 통해
아이템을 사용

A.

기본적인 기반 코드

```
#include <asm/ioctls.h>
#include <fcntl.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ioctl.h>
#include <sys/signal.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <termios.h>
#include <time.h>
#include <unistd.h>

#define dot "/dev/dot"

unsigned char mole[9][8] = {
    // 도트 매트릭스 화면
    {0x00, 0x60, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00}, //7번두더지
    {0x00, 0x18, 0x18, 0x00, 0x00, 0x00, 0x00, 0x00}, //8번두더지
    {0x00, 0x06, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00}, //8번두더지
    {0x00, 0x00, 0x00, 0x60, 0x60, 0x00, 0x00, 0x00}, //4번두더지
    {0x00, 0x00, 0x00, 0x18, 0x18, 0x00, 0x00, 0x00}, //5번두더지
    {0x00, 0x00, 0x00, 0x06, 0x06, 0x00, 0x00, 0x00}, //6번두더지
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x60, 0x60, 0x00}, //1번두더지
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18, 0x00}, //2번두더지
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x06, 0x00} //3번두더지
};

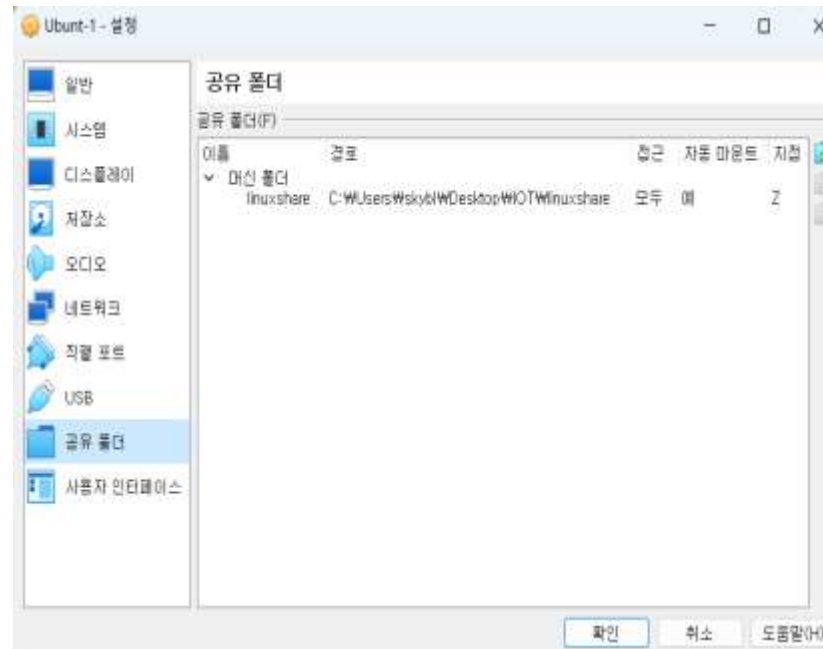
int main() {
    int dot_d, i, random;
    dot_d = open(dot, O_RDWR);
    if(dot_d < 0)
    {
        printf("오류 발생");
        exit(0);
    }

    for(i=0; i<9; i++)
    {
        random = rand()%9; //난수 생성
        write(dot_d, &mole[random], sizeof(mole));
        sleep(50000);
    }
    close(dot_d);
}
```

공유폴더 설정

STEP 1

공유 폴더를 설정하여
'/opt/gnueabi/opt/ext-
toolchain/bin/arm-linux-
gnueabihf-gcc
[파일명].c-o
[실행파일명]'을 통해
컴파일함



>>

Part 2, 컴파일 및 타겟 시스템 전송-출력

타겟시스템전송

STEP 2

이후, 'TeraTerm'을
이용하여 컴파일
한

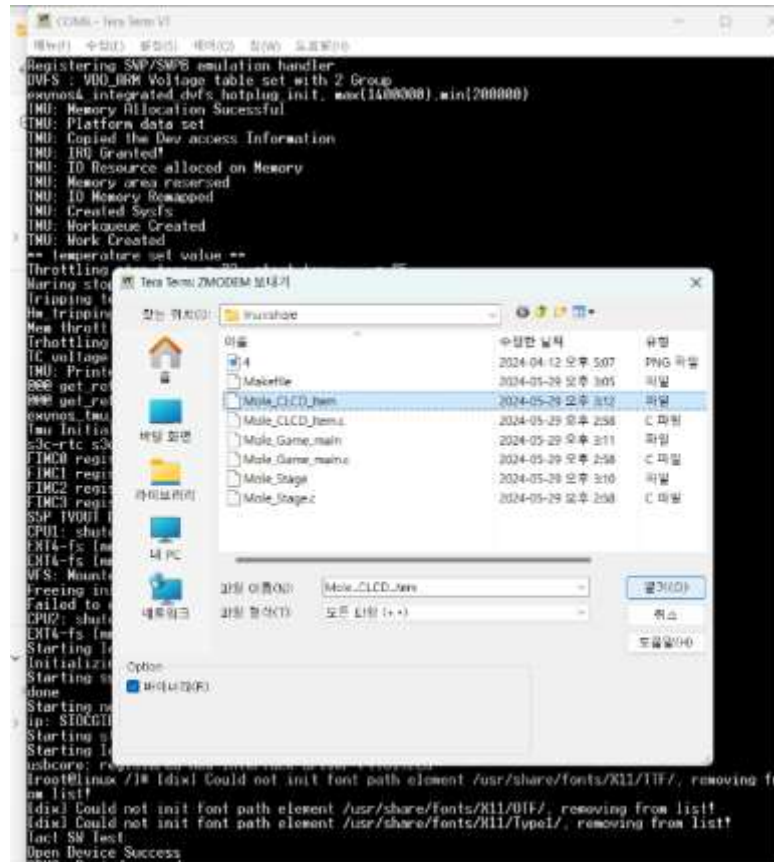
'Mole_Game_main'
파일을 타겟시스

템에 전송

`chmod 777`

`Mole_Game_main`

`./Mole_Game_main`



>>

타겟시스템전송 확인

STEP 3

출력화면을
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 /]#

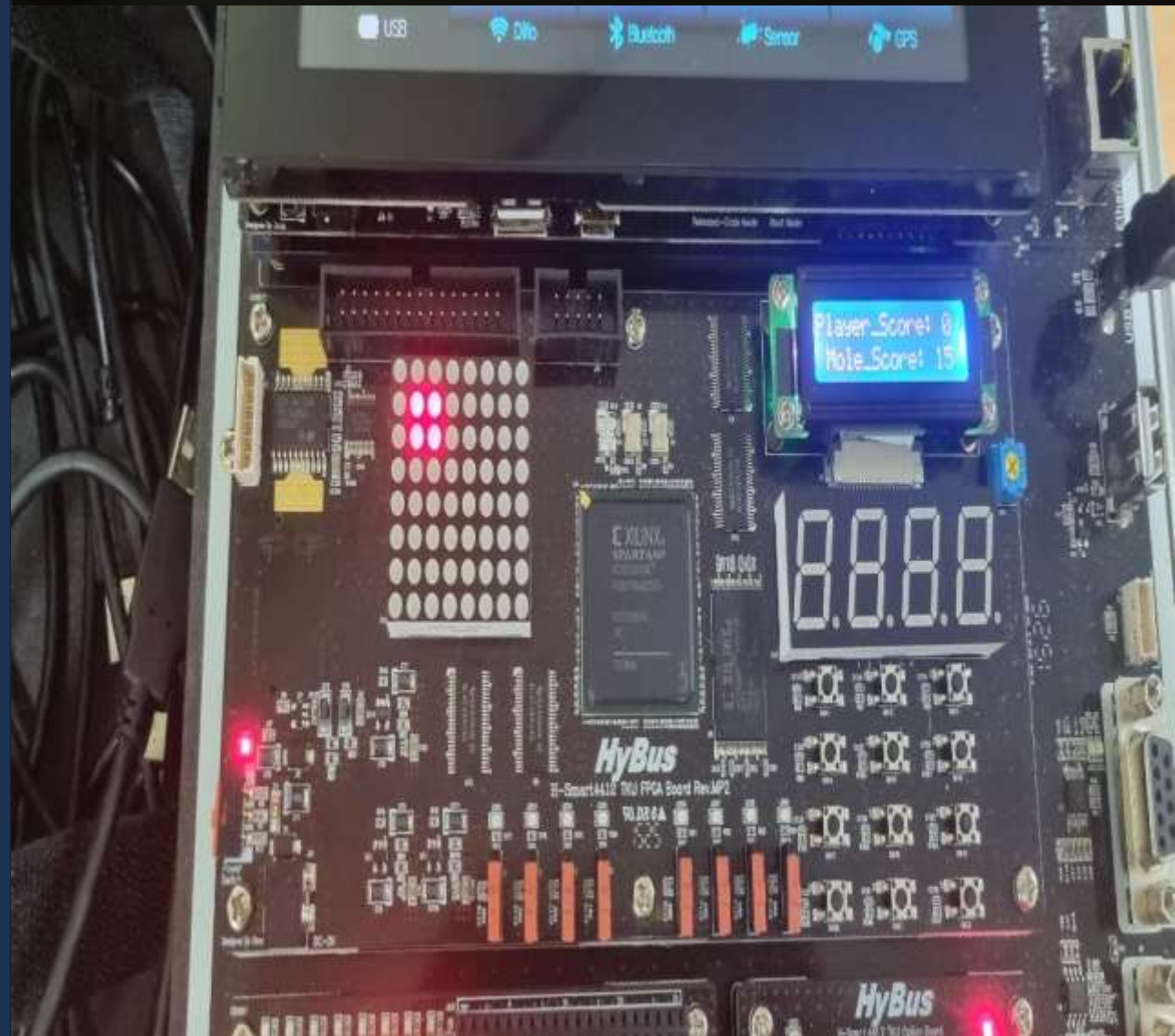
```

```

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
두더지점수 : 1
두더지점수 : 2
두더지점수 : 3
두더지점수 : 4
두더지점수 : 5

```

Part 3, 임베디드 실행 확인



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

”