

# Yacht Dice 최종 발표

1조

2019210157 이주훈

2019250014 김호탁

2024810106 원대현

2024810109 이정인

# 목차

1

주제 선정 이유

2

각 모듈의 기능

3

시연 영상

4

깃허브

5

진행시 어려웠던 점

# 주제 선정 이유

1

기존 프로젝트들이 게임이 많았음

2

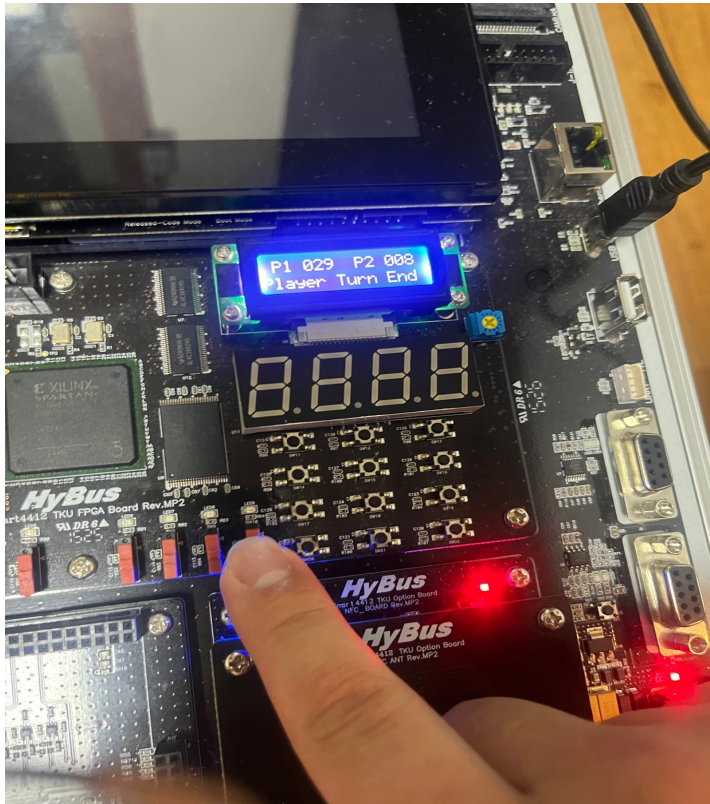
어려가지 인터페이스 사용가능해 보임

3

버퍼를 사용해서 동적 개발이 가능해 보였음

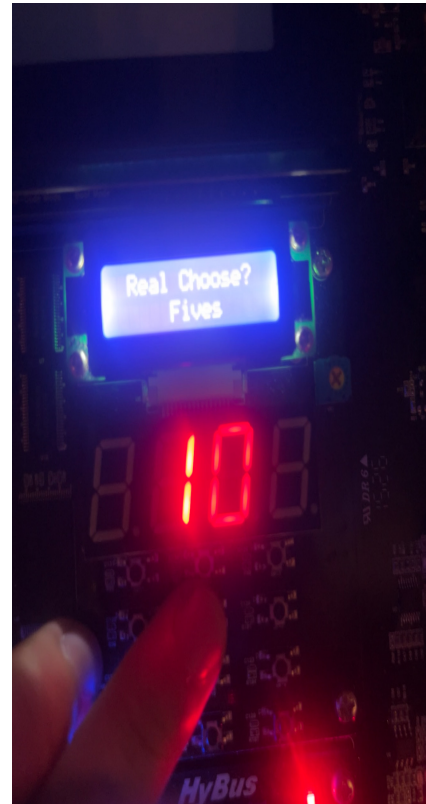
# 각 모듈의 기능 - CLCD, FND

CLCD - 다양한 문장 출력



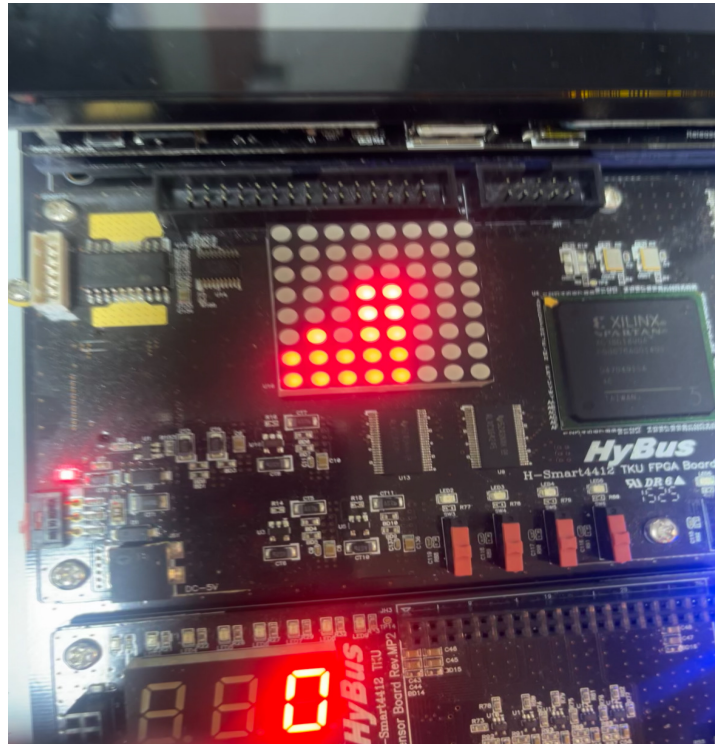
FND

- 해당 족보의 점수 출력
- 해당 족보 사용시 USED 출력

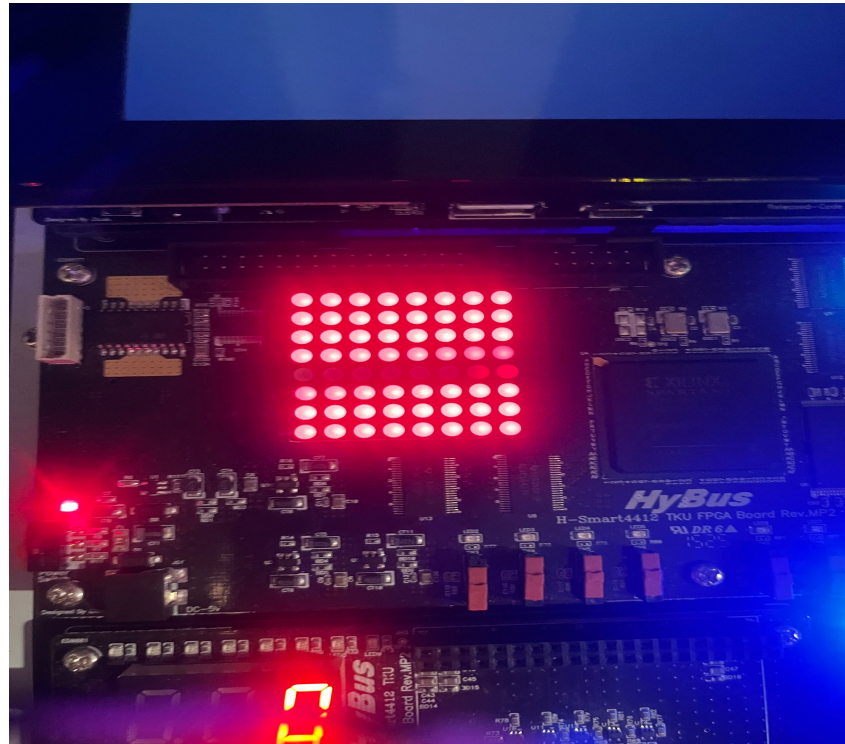


# 각 모듈의 기능 - DOT MATRIX

주사위 출력



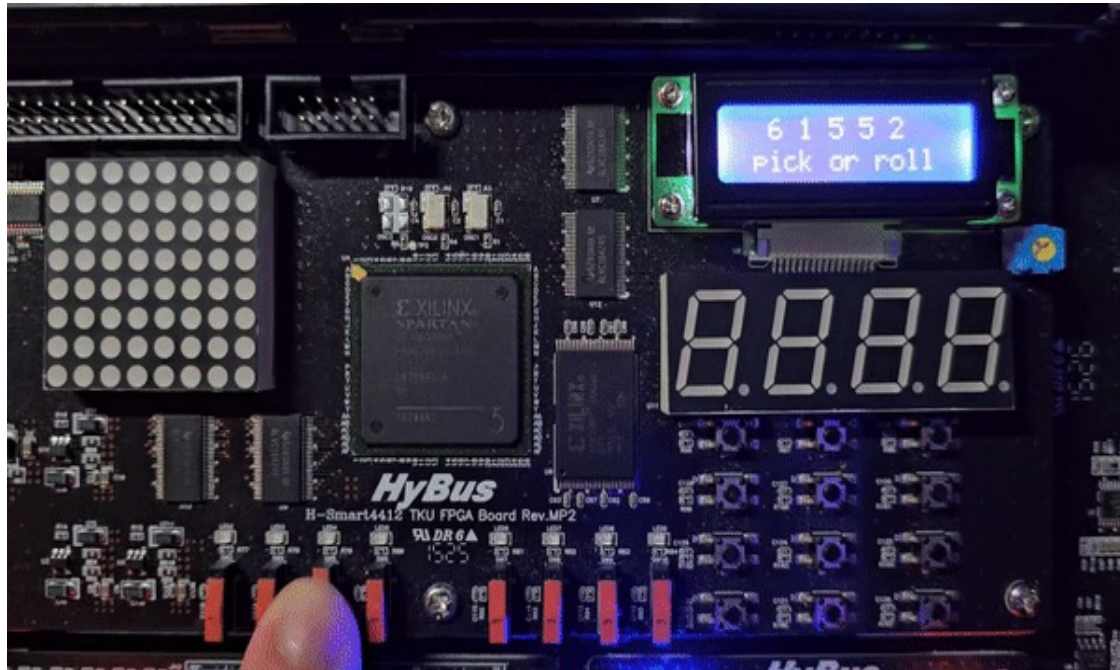
첫 시작 Dip스위치 홀드시 경고 표시



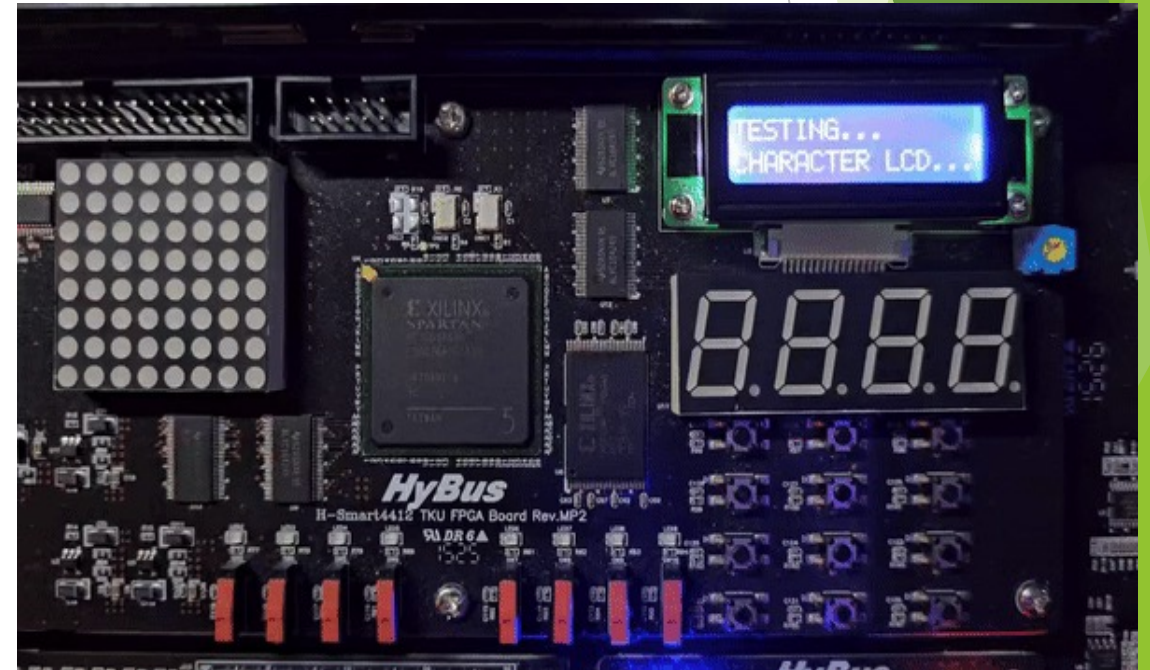


# 각 모듈의 기능 - DIP

주사위 홀드 : 1 ~ 5번



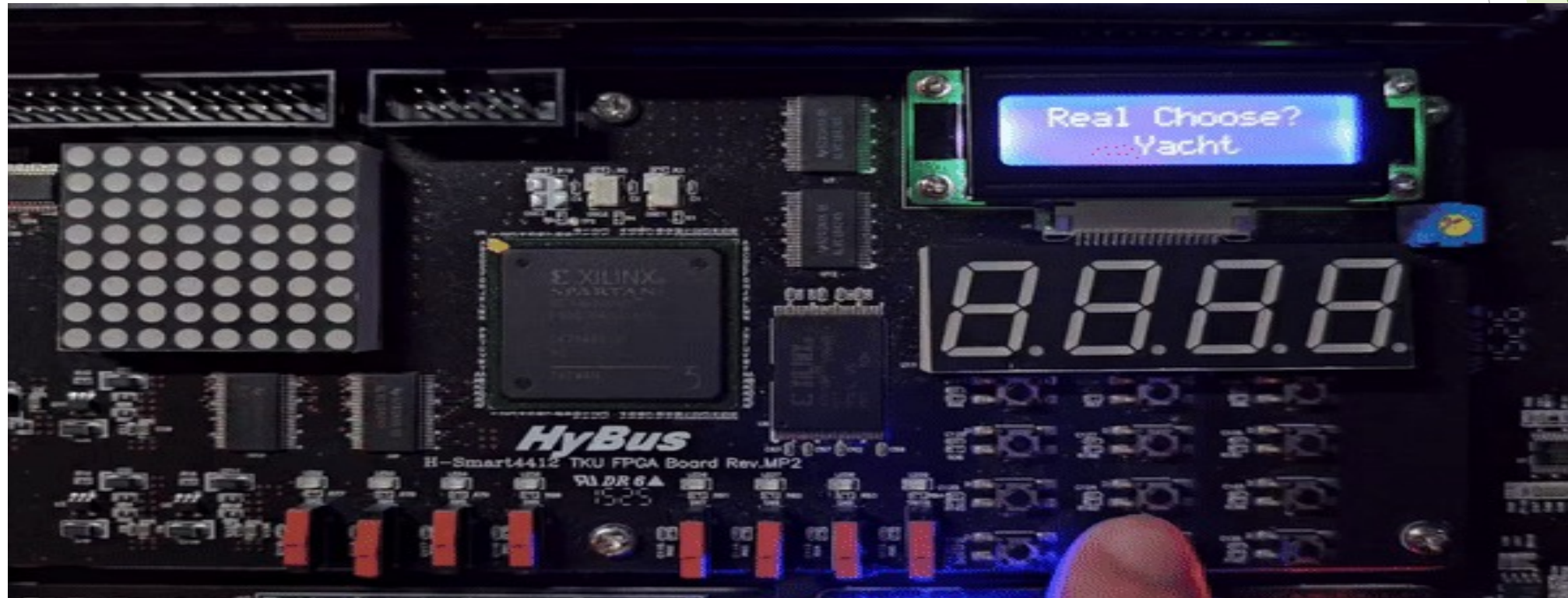
주사위 던지기 : 8번



# 각 모듈의 기능 - TACT

족보 확인 : 처음 클릭시

점수 확정 : 두번 클릭시



시연영상(영상 길이상 축소하여 게임 진행)

# IoT 프로그래밍 1조 시연 영상

<https://youtu.be/WEZicodjqPA>



# 깃허브 - 리드미

Chrome파일 수정 보기 방문 기록 북마크 프로필 탭 창 도움말

whdjh/IoT\_Proj

github.com/whdjh/IoT\_Proj

모든 북마크

README

## IoT

한경대학교 컴퓨터공학과 소프트웨어융합 학부생

개발기간:2024.05 - 2024.06

### 개발팀 소개



이주훈	김호탁	원대현	이정인
@whdjh	@kimhotac		
한경대 컴퓨터공학과 sc	한경대 컴퓨터공학과 sc	한경대 컴퓨터공학과 sc	한경대 컴퓨터공학과 sc

### 프로젝트 소개


4학년 IoT시스템 프로그래밍 프로젝트로 진행한 Hybus의 H-Smart4412 타겟시스템을 이용한 Yacht Dice 프로젝트입니다.

- CLCD모듈에서 다양한 문장을 출력해줍니다. (Ex, Game Start, Player0 Turn, Roll a Dice)
- FND모듈에서 족보의 점수를 출력해줍니다.
- Dip 스위치로 주사위 던지기과 홀드 기능을 사용할 수 있습니다.
- Tact 스위치를 이용해 Yacht Dice의 처음 클릭시 족보를 확인할 수 있고 한번 더 누르면 점수를 확정할 수 있습니다. (단, 사용한 족보는 다시 쓰지못하게 했습니다.)
- Dot Matrix를 이용해 주사위를 출력하고 경고표시합니다



### 기술스택 & 개발환경



### 다이어그램



Contributors 2



Languages

C 100.0%

Suggested workflows

Based on your tech stack

C/C++ with Make

Configure

Build and test a C/C++ project using Make.

CMake based, multi-platform projects

Configure

Build and test a CMake based project on multiple platforms.

MSBuild based projects

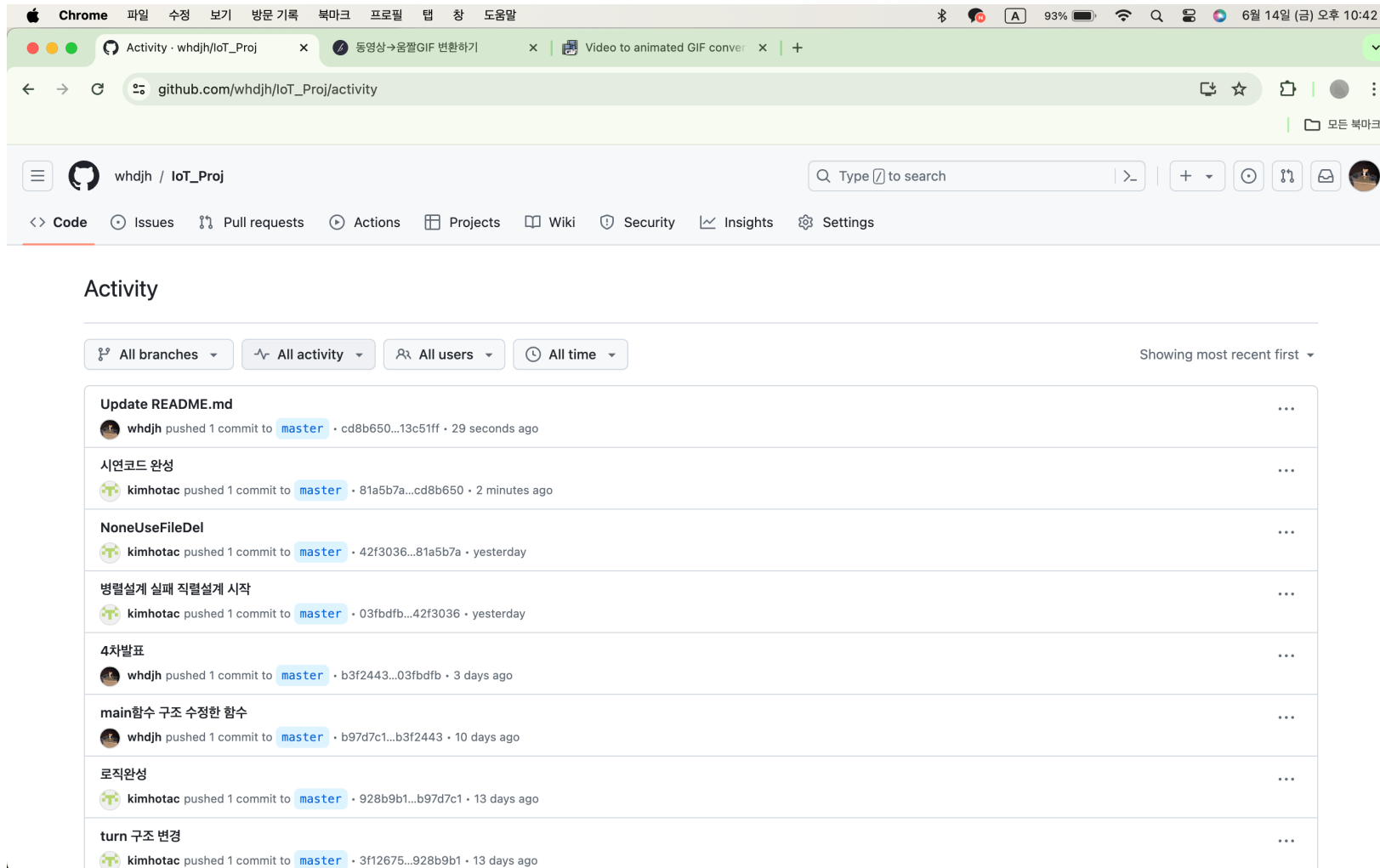
Configure

Build a MSBuild based project.

[More workflows](#)

Dismiss suggestions

# 깃허브 - 커밋 히스토리 관리



The screenshot shows the GitHub Activity page for the repository `whdjh / IoT_Proj`. The page displays a list of recent commits, filtered by 'All branches', 'All activity', 'All users', and 'All time'. The commits are sorted by 'Showing most recent first'.

Commit Title	Author	Branch	Commit Hash	Time Ago
Update README.md	whdjh	master	cd8b650...13c51ff	29 seconds ago
시연코드 완성	kimhotac	master	81a5b7a...cd8b650	2 minutes ago
NoneUseFileDel	kimhotac	master	42f3036...81a5b7a	yesterday
병렬설계 실패 직렬설계 시작	kimhotac	master	03fbdff...42f3036	yesterday
4차 발표	whdjh	master	b3f2443...03fbdff	3 days ago
main함수 구조 수정한 함수	whdjh	master	b97d7c1...b3f2443	10 days ago
로직완성	kimhotac	master	928b9b1...b97d7c1	13 days ago
turn 구조 변경	kimhotac	master	3f12675...928b9b1	13 days ago

# 진행시 어려웠던 점

1

제한적인 기본 프로젝트 사례

2

타겟시스템 디버깅 프로그램 부족

3

동시 입출력시 제어의 문제

# 하드웨어 키 제어 미숙

```
while(0 < tact_input && tact_input < 13){
    unsigned char temp = 0;
    usleep(1000); // 0.001 초 쉬기
    read(tactsw, &temp, sizeof(temp));
    // tact 뽐
    if(!temp){
        //tact 눌렀을 때 이벤트 처리
        // 점수 출력 부분
        if(tact_input != before_input) {
            set_lcd(23, tact_input - 1);
            printf("[dot] %d %d %d %d %d\n", dice[0], dice[1], dice[2], dice[3], dice[4]);

            if((1 << (tact_input - 1)) & *category) {
                set_turn_score(100);
            }
            else {
                set_turn_score(score[tact_input-1]);
                before_input = tact_input;
            }
        }
        // 점수 확정 부분
        else {
            //TODO category에 사용한 족보 넣기
            *category |= 1 << (tact_input - 1);
            return score[tact_input-1];
        }
        break;
    }
}
```

기존 설계(TACT코드) : 1회 입력 : 출력, 2회 입력 : 확정

1회 입력시에도 확정되는 문제가 발생

tact 스위치가 down 된 동안 모든 코드가 돌고 확정되는 부분 발견

tact 스위치가 down 이후 up 되었을 때 출력, 확정 되는 로직 실행하여 문제 해결



# 동시 입출력 문제 해결 시도

```
int main(){
    if((dipsw = open( dip, O_RDWR )) < 0){printf("dip open error\n"); exit(1);}

    memset(dot_buffer, 0xff, sizeof(dot_buffer)); // 도트 버퍼 초기화
    memset(fnd_buffer, 0x00, sizeof(fnd_buffer)); // fnd 버퍼 초기화
    pthread_t dot_thread;
    pthread_create(&dot_thread, NULL, dot_fnd_thread, NULL);

    unsigned char dip_input = 0;
    while(1) {
        usleep(1000000); // 0.01 초 쉬기
        read(dipsw, &dip_input, sizeof(dip_input));
        if(dip_input)printf(".");
    }
    sleep(30);
}

void* dot_fnd_thread(void* arg) {
    while (1) {
        dot_mtx = open(dot, O_RDWR);
        pthread_mutex_lock(&dot_mutex);
        write(dot_mtx, dot_buffer, sizeof(dot_buffer));
        usleep(500000); // 0.5 초
        pthread_mutex_unlock(&dot_mutex);
        close(dot_mtx);

        fnds = open(fnd, O_RDWR);
        pthread_mutex_lock(&fnd_mutex);
        write(fnds, &fnd_buffer, sizeof(fnd_buffer));
        usleep(500000); // 0.5 초
        pthread_mutex_unlock(&fnd_mutex);
        close(fnds);
    }
    return NULL;
}
```

목적 : 멀티 스레딩을 통해 Dot과 Fnd 동시출력시 발생하는 문제를 해결하고자 함.

방법 : 버퍼를 전역변수로 선언, 한 스레드가 Dot과 Fnd 번갈아 출력 main 흐름에서는 다른 입출력과 로직을 실행, 동시 출력 문제는 상호배제(mutex)로 해결

테스트 결과 : 독립적인 스레드 단위 테스트에서는 문제가 발생하지 않음 기존 코드에 병합 시 동일한 오류 발생 (read와 충돌)

문제 해결 불가 판단 후, 동시에 여러 입출력 설계를 출력 후 sleep을 통해 단일 입출력으로 변경