

# Embedded System Software 프로젝트

## (과제 수행 결과 보고서)

과목명: [CSE4116] 임베디드시스템소프트웨어  
담당교수: 서강대학교 컴퓨터공학과 박 성 용

학번 및 이름: 20151550, 박태준

개발기간: 2020. 05. 22. -2020. 05. 28.

# 최 종 보 고 서

## I. 개발 목표

Android, JNI, Device Driver, Timer, Interrupt Handler를 이용해 퍼즐 맞추기 게임을 만든다. Android에서 스크린에 짝을 맞추기 게임을 구현한다. Device Driver에서 FND와 Timer를 이용해 점수를 표기한다. 1초마다 점수는 1점씩 떨어지며, 한 쌍의 페어를 맞추 때마다 30점의 점수를 얻는다. 모든 짝을 찾으면 Timer를 멈춰 점수를 표기한다. 만약 게임의 점수가 최고 기록을 달성할 경우 스크린에서는 최고 기록을 달성했다고 알린다. 이후, Home 버튼을 이용해 Interrupt를 걸 경우, TextLCD에서 최고 기록을 출력한다.

## II. 개발 범위 및 내용

### 가. 개발 범위

#### 0. Android :

Android를 이용해 짝 맞추기 게임 화면을 구현한다. Button을 동적으로 할당하며 원하는 개수의 Button을 할당하도록 한다. 이후 게임이 시작되면 버튼의 Text에 숫자가 표기되어야 한다. OnClickListener를 이용해 게임이 진행될 수 있어야 한다. 화면에 표기되는 텍스트 또한 게임의 결과에 따라 출력 결과가 달라야 한다. JNI로 작성한 Library를 Load하여 사용해야 한다.

#### 1. JNI :

JNI는 C로 작성된 코드가 Java 가상머신과 상호작용하기 위해서 필요하다. Android에서는 JNI에 선언된 함수를 Load하여 사용한다. JNI에서는 디바이스를 열고, IOCTL를 호출하여 필요한 작업을 수행한다. 작업 내용으로는 점수를 FND에 표기하기, 게임이 끝났을 때 최고 점수인지 확인하기, 게임이 끝났음을 알리기, 게임 시작 시 값 초기화하기가 있다. 즉, 이전 과제에서의 유저 프로그램 역할을 수행한다고 할 수 있다.

#### 2. Device Driver :

JNI에서 사용하는 Device Driver를 작성해야 한다. Driver를 열고, IOCTL 기능을 수행하며, Java 프로그램이 닫힐 때 device driver도 release 작업을 수행하기 때문에 이를 모두 구현해야 한다. C로 구현하며 구현 후 Kernel로 ko 파일을 보내 device driver로 등록한다.

#### 3. Interrupt Handler :

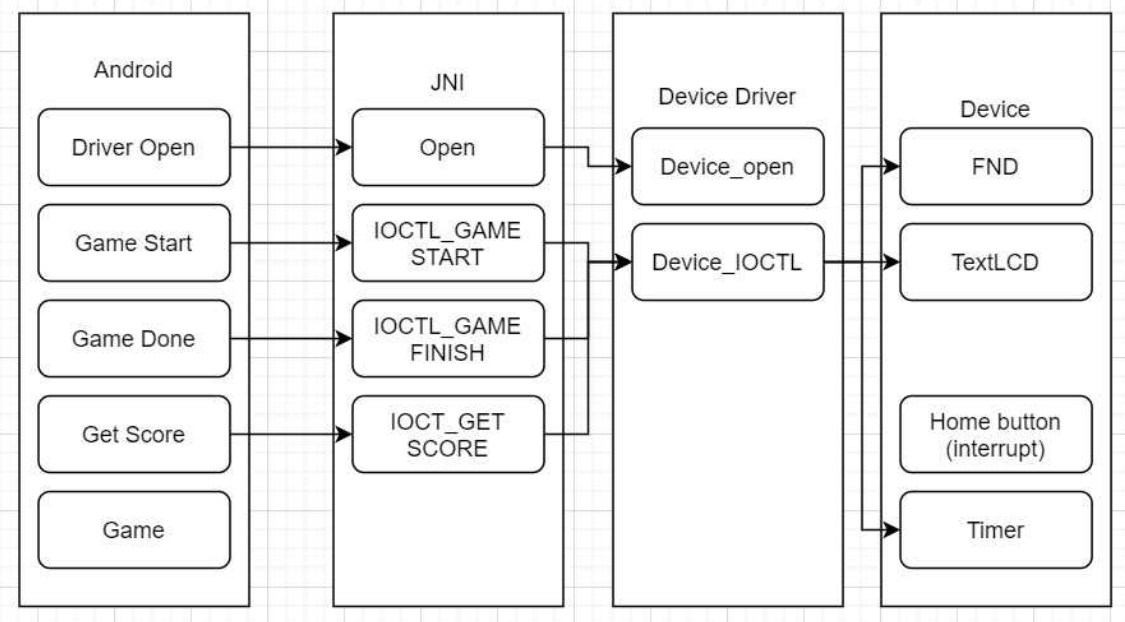
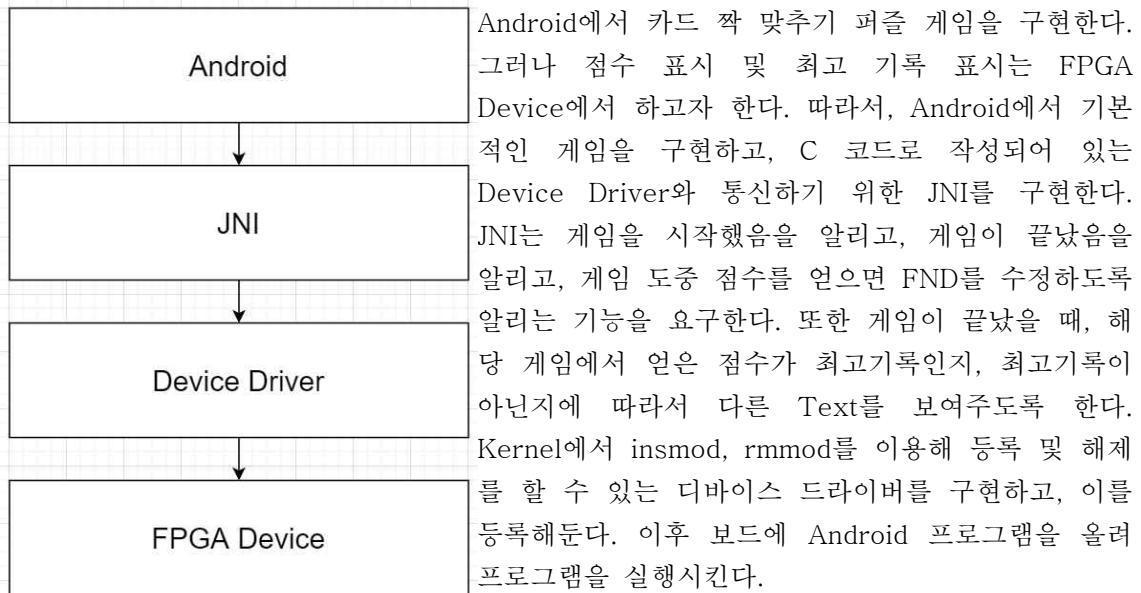
인터럽트 핸들러를 만들어 Home 버튼에 대해 기능을 구현해야 한다. Home 버튼은 저장된 최고 기록을 TextLCD에 출력시킨다.

이를 위해서 종료된 게임 중 가장 고득점의 점수를 저장하고 있어야 한다.

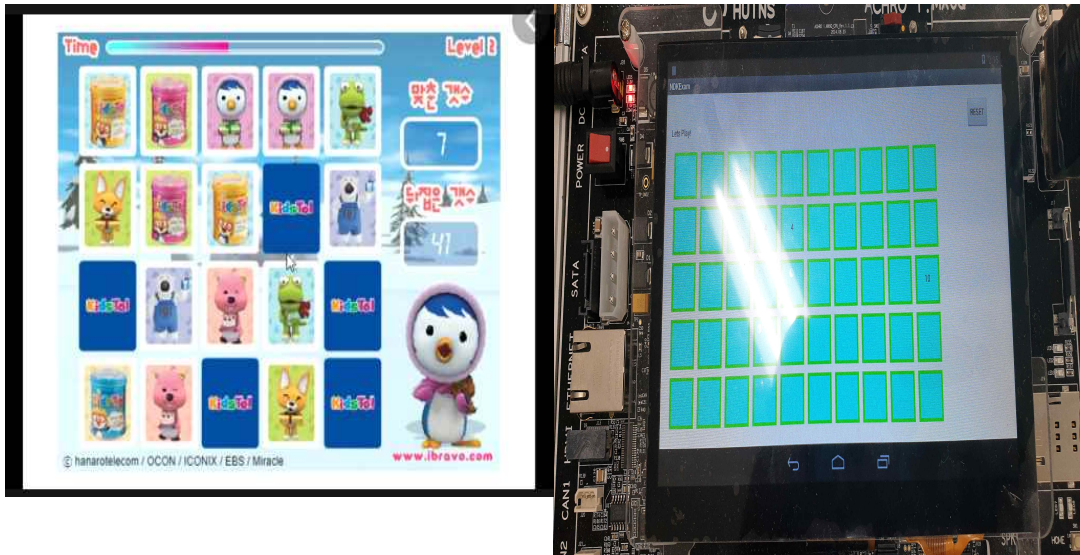
#### 4. Timer :

게임 진행을 위해 Timer를 필요로 한다. 퍼즐 맞추기 게임은 빠르게 진행할수록 잘한다는 뜻이기 때문에 점수가 0점 이상일 경우 1초마다 1점씩 점수를 잃는다. 1초마다 점수를 잃게 만들게 하기 위해서, 게임이 시작했음을 IOCTL를 통해 알게 되는 순간, 타이머가 작동한다. 이후, 게임이 종료되거나 Reset 명령이 들어올 경우 Timer가 초기화된다.

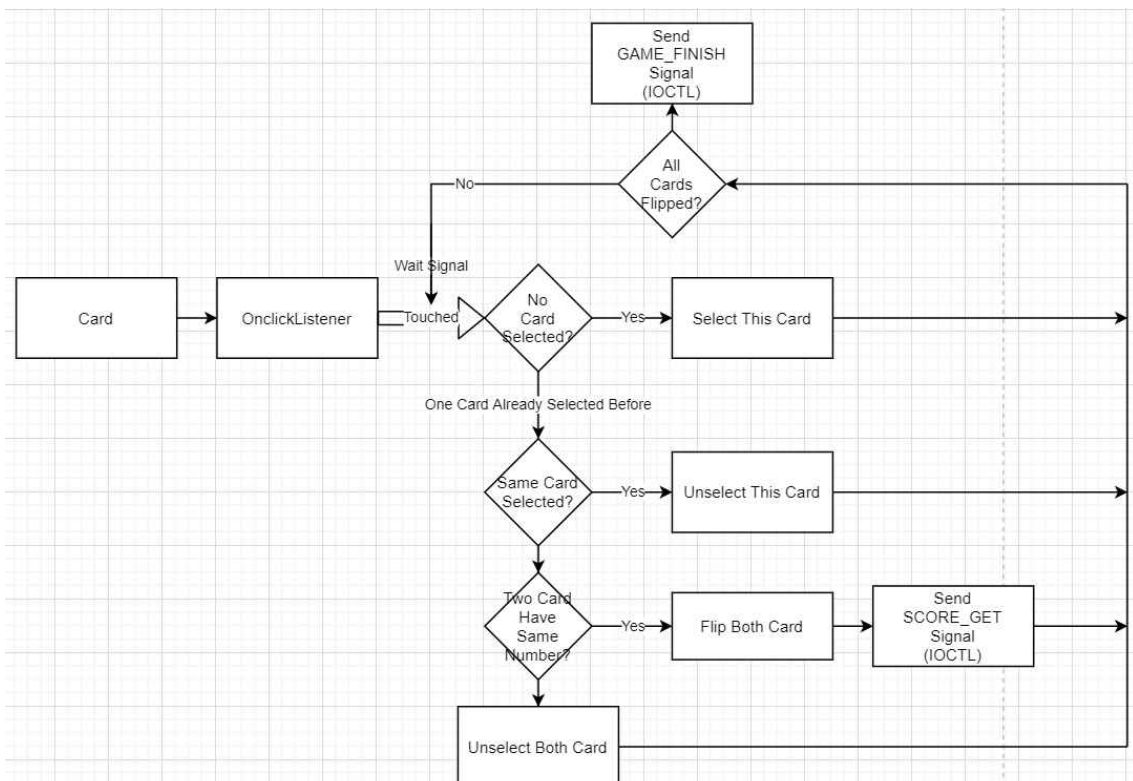
##### 나. 개발 내용



개략적인 그림은 위와 같다. 이때, 구현하려는 게임은



왼쪽과 같으며, 실제 구현한 화면은 오른쪽과 같다. 카드 맞추기 게임 로직은 Java로 구현한다.



카드 한 장마다 OnClickListener를 적용시켰는데, OnClickListener를 이용한 기본적인 구조는 위와 같다.

### III. 추진 일정 및 개발 방법

#### 가. 추진 일정

- 6.22-23 : 프로젝트 기획
- 6.24-25 : 필요한 기능 설계 및 학습 ,
- 6.24-26 : Java 프로그램 작성 및 JNI 구현
- 6.27-28 : Device Driver 구현
- 6.28 : 전체적인 디버깅 및 정리

#### 나. 개발 방법

##### 1. Android Program : NDKExam.java

실습예제 프로젝트 중 NDKExam 프로젝트를 뼈대로 작성하였다.

##### NDKExam Class:

퍼즐 맞추기를 위해 생성한 Class.

OnCreate에서 대부분의 작업을 수행한다.

AssignNum() : 동적으로 생성된 Button에게 임의의 번호를 부여하는 함수. 각 번호마다 2개의 같은 번호를 같은 버튼이 존재한다.

onCreate() : puzzle-game이라는 JNI 라이브러리를 로드한다.

시작 버튼을 만든다.

지정한 개수의 Button을 만들어 카드로 사용한다.

시작 버튼의 onClickListener를 선언한다. onClickListener가 호출되면 게임을 시작하고, JNI함수 중 startGame을 호출한다. 게임이 시작되면 3초간 어떤 번호가 카드에 부여되어 있는지 보여주고, 카드를 뒤집는다.

각 카드의 onClickListener를 선언한다. 각 카드의 onClickListener는 게임이 시작되어야만 동작한다. 한 번 누르면 가지고 있는 숫자를 화면에서 보여준다. 2개의 카드를 뒤집었을 때 같은 번호를 가지고 있다면 30점의 점수를 받고 완전히 뒤집는다. 2개의 카드를 뒤집었을 때 다른 번호를 가지고 있다면 다시 보이지 않게 뒤집는다. 만약 모든 카드를 뒤집는 데 성공한다면, 게임이 완료된다.

동적으로 버튼을 만든다. 원하는 개수를 정하고 가로, 세로의 수를 정하고 Button을 활성화한다. 모든 버튼을 한 onClickListener로 연결하여 사용한다.

##### 2. Android Program : JNI

NDKExam class에서 getScore, startGame, finishGame, driveOpen 이 네가지 native 함수를 사용하기 위해서는 이를 위한 c 파일이 필요하다.

```
jint JNICALL Java_org_example_ndk_NDKExam_driveOpen(JNIEnv *env, jobject obj)
```

drive를 프로그램에 연결하는 함수.

drive를 열고, major number를 리턴한다.

```
jint JNICALL Java_org_example_ndk_NDKExam_startGame(JNIEnv *env, jobject obj,
jint dev)
```

FND에 점수를 얻었음을 전달하기 위해 사용하는 함수. ioctl의 IOCTL\_GET\_SCORE를 불러서 사용한다.

```
jint JNICALL Java_org_example_ndk_NDKExam_startGame(JNIEnv *env, jobject obj,
jint dev)
```

게임이 시작했을 때 호출되는 함수. Device 내부에 게임 시작시 초기화되어야 하는 값들을 모두 초기화하는 역할을 수행하며, ioctl의 IOCTL\_START\_GAME을 불러서 사용한다.

```
jint JNICALL Java_org_example_ndk_NDKExam_finishGame(JNIEnv *env, jobject obj,
jint dev)
```

게임이 끝났을 때 호출되는 함수. Device에서 호출되고 있던 Timer를 중지시키는 역할을 한다. 또한, device driver가 저장하고 있던 최고 기록을 비교하여 이번 게임에서 얻은 스코어가 최고 스코어인지 아닌지 알려주기 위해 사용한다.

```
JNIEXPORT jint JNICALL Java_org_example_ndk_NDKExam_getScore
(JNIEnv *, jobject, jint);
```

```
/*
 * Class:      org_example_ndk_NDKExam
 * Method:     startGame
 * Signature:  (I)I
 */
```

```
JNIEXPORT jint JNICALL Java_org_example_ndk_NDKExam_startGame
(JNIEnv *, jobject, jint);
```

```
/*
 * Class:      org_example_ndk_NDKExam
 * Method:     finishGame
 * Signature:  (I)I
 */
```

```
JNIEXPORT jint JNICALL Java_org_example_ndk_NDKExam_finishGame
(JNIEnv *, jobject, jint);
```

```
/*
 * Class:      org_example_ndk_NDKExam
 * Method:     driveOpen
 * Signature:  ()I
 */
```

```
JNIEXPORT jint JNICALL Java_org_example_ndk_NDKExam_driveOpen
(JNIEnv *, jobject);
```

### 3. Device Driver : timer.c

명세서에 서술된 요구 사항을 만족시키는 Device Driver 프로그램을 작성하였다.

```
static int __init timer_init(void)
static void __exit timer_exit(void)
module_init(timer_init);
module_exit(timer_exit);
```

디바이스를 등록, 해제하는 함수이다. 디바이스를 등록하고 해제하는 과정에서 register\_chrdev 함수와 unregister\_chrdev 함수, 각 디바이스의 physical memory를 매핑하는 ioremap 함수와 이를 해제하는 iounmap 함수를 이용해 메모리 할당과 해제 과정을 거쳐야 한다. 또한 timer를 위해 init\_timer, del\_timer\_sync 함수도 포함되어야 한다.

```
static int timer_open(struct inode *, struct file *);
static int timer_release(struct inode *, struct file *);
long timer_ioctl( struct file *mfile, unsigned int ioctl_num, unsigned long param);
```

```
static struct file_operations timer_fops =
{
    .open = timer_open,
    .unlocked_ioctl = timer_ioctl,
    .release = timer_release,};
```

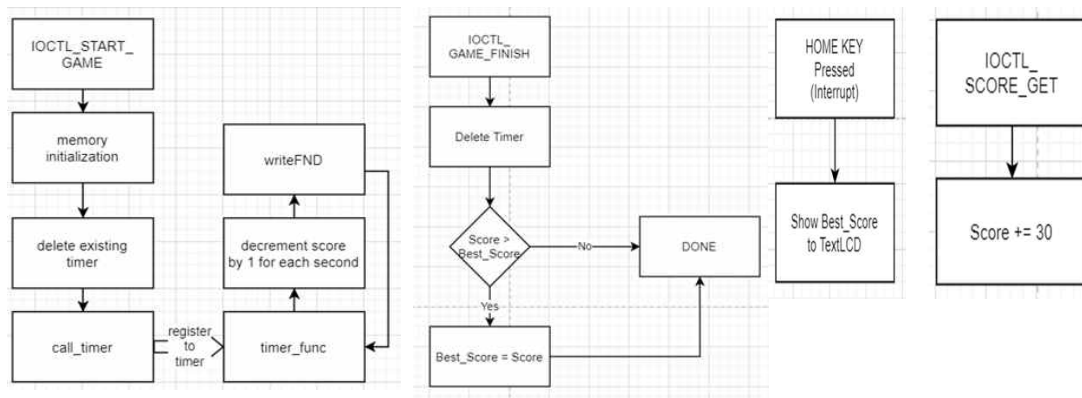
timer\_open 함수는 device driver를 open 함수로 부를 때 동작한다. device open request\_irq 함수를 이용해 Home 버튼에 대한 인터럽트 핸들러를 등록한다.

timer\_ioctl 함수는 android와 소통하는데 필요한 변수를 초기화하고, device에 값을 쓰며, timer 함수를 작동시킨다.

timer\_release 함수는 device driver를 close 함수로 부를 때 동작한다. android 프로그램이 닫히면 자동으로 동작하기에, 변수를 모두 초기화하도록 만들었다.

```
void writeFND(void)
void writeLCDTEXT(void)
void display(void)
```

위의 함수는 FND 값과 TEXTLCD를 수정하기 위한 함수이다. display에선 writeFND에 반영되는 값인 display 배열의 값을 수정하고, writeFND 함수는 직접 이를 FND의 메모리에 반영시키는 역할을 한다. writeLCDText는 text라는 unsigned character 배열에 저장되어 있는 값을 TextLCD에 반영시키는 역할을 한다.



Device Driver의 각 기능에 대한 간단한 순서도이다.

#### 4. INTERRUPT HANDLER

```
irqreturn_t home_handler(int irq, void* dev_id, struct pt_regs* reg);
```

인터럽트 핸들러는 Home 버튼을 누를 때 동작한다. 기능은 Home버튼을 누르면, 지금까지 끝난 게임 중 최고 득점을 TextLCD에 출력하는 것이다.

#### 5. IOCTL

```
long timer_ioctl( struct file *mfile, unsigned int ioctl_num, unsigned long param)
{
    switch(ioctl_num)
    {
        case IOCTL_START_GAME:
            memset(tm.display,0,4);
            memcpy(text,"",32);
            del_timer(&tm.timer);
            score = 0;
            writeFND();
            writeLCDTEXT();
            call_timer();
            break;

        case IOCTL_GAME_FINISH:
            del_timer(&tm.timer);
            if(best_score < score)
            {
                best_score = score;
                return 1;
            }
    }
}
```



```

        else return 0;
    case IOCTL_SCORE_GET:
        score += 30;
        display();
        writeFND();
    }
    return 1;
}
irqreturn_t home_handler(int irq, void* dev_id, struct pt_regs* reg)

```

기본적으로 JNI를 이용해 Device와 소통하는 방법은 Device Driver의 IOCTL 기능을 이용한다.

JNI에서는 IOCTL\_GAME\_FINISH, IOCTL\_START\_GAME, IOCTL\_SCORE\_GET 에 대한 처리 방식이 존재한다.

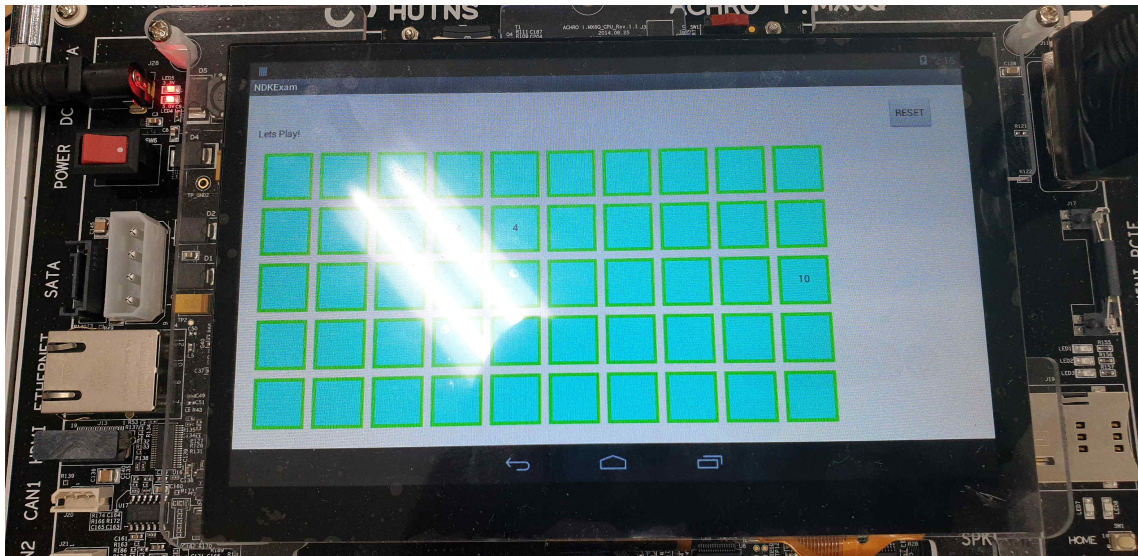
IOCTL\_GAME\_FINISH는 게임이 끝났을 때 호출되며, 타이머를 지우고 현재 점수를 FND에 남겨놓는다. 만약 이 점수가 최고 기록보다 높을 경우 이를 최고 기록으로 저장한다.

IOCTL\_SCORE\_GET은 게임을 진행하면서 호출된다. 두 짝을 맞췄을 때 30점의 점수를 얻으며, 이를 FND에 표시해준다.

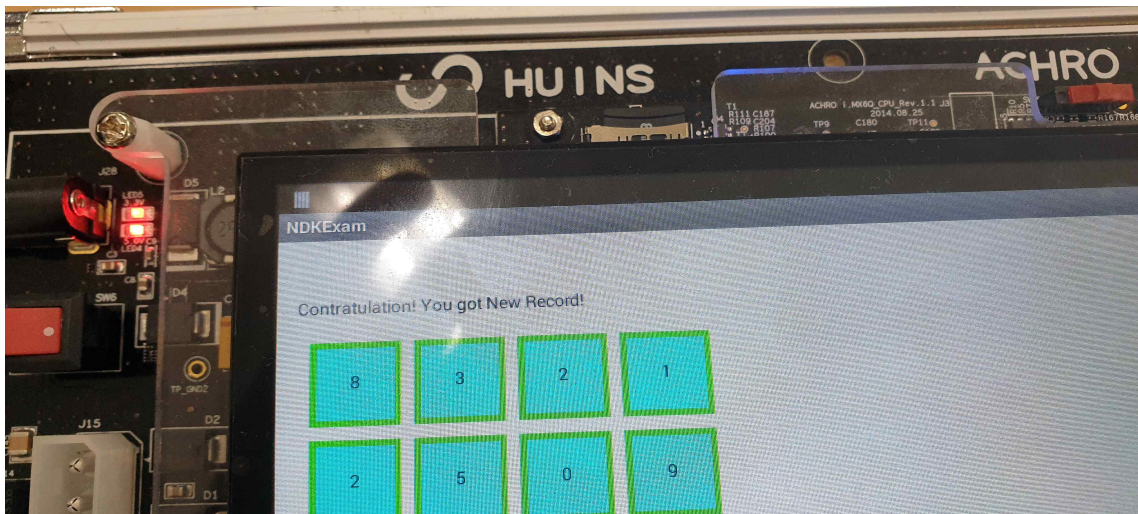
IOCTL\_START\_GAME은 게임이 시작되면서 호출된다. 게임에 영향을 미치는 점수판, 현재 점수, TextLCD를 초기화한다.

#### IV. 연구 결과

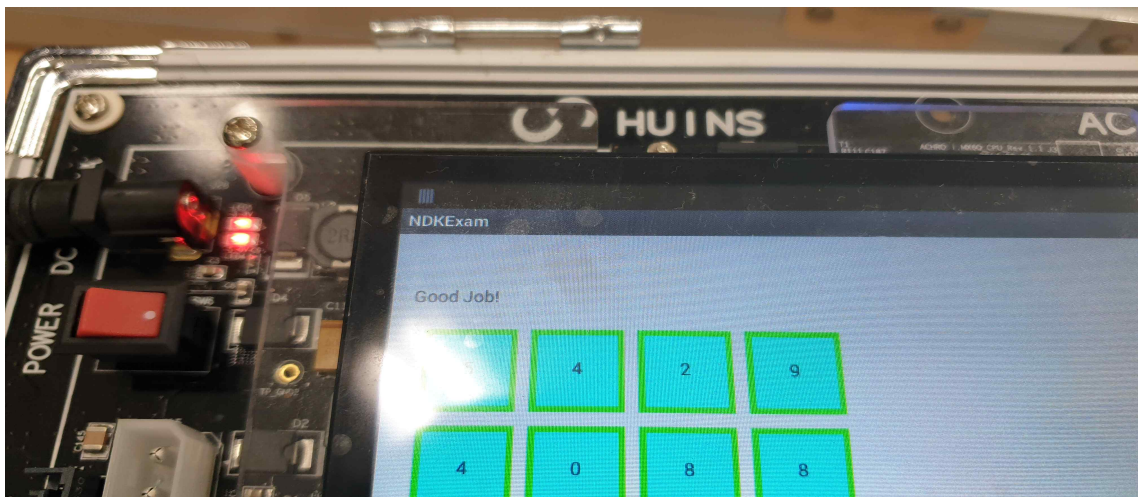
위의 서술한 과정을 바탕으로 Android Program과 Device Driver를 작성하였고, 이를 보드에 전송하였다. 이후, Device Driver를 insmod로 등록 - Android Program 실행 과정을 수행해보았다. Start button으로 퍼즐 게임을 시작해 같은 짝을 찾았을 때 점수를 얻었으며, 점수를 얻으면 FND에 30점이 올라감을 확인할 수 있었다. 점수가 0 이상일 때는 1초마다 점수가 깎이는 것을 볼 수 있었다. 게임이 끝나면 Timer가 멈추며 내 점수를 확인할 수 있었으며, 이번 게임에 최고 점수를 달성했는지 아닌지에 따라 출력 Text가 바뀌는 것도 확인할 수 있었다.



5 \* 10 짜리 게임을 실행시킨 모습

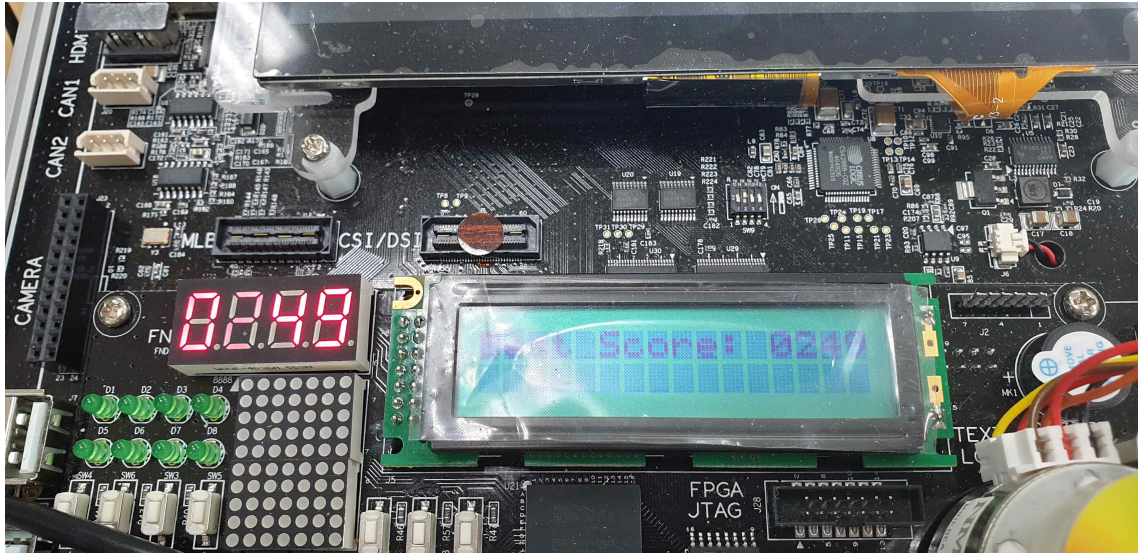


게임 결과 신기록을 달성한 경우



게임 결과 최고 기록에 미치지 못한 경우





보드를 통해 표시되는 결과들. TextLCD는 Home 버튼을 이용해 인터럽트를 주어야 나타난다.

## V. 기타

이번 과제를 수행하면서 동적으로 버튼을 할당하는 과정에서 가장 많은 어려움을 겪었습니다. 이를 해결하기 위해서, 익명의 `onClickListener`를 따로 선언하고, `button`을 동적으로 생성하며 한 `Listener`에서 적용시키는 방법을 선택하였습니다. 이때, `button`을 동적으로 생성할 때, `setID`를 하지 않으면 찾을 수가 없었다는 것을 몰라 난항을 겪었습니다. 이후 이를 검색을 통해 알게 되었고, 이후 배열을 선언해 `Button`을 저장하며 관리할 수 있었습니다. 그러나 `onClickListener`를 하나만 사용하는 것이 가끔씩 터치 오류를 발생시키는 원인이 되기도 하였습니다.

한 학기동안 교수님, 조교님 수고하셨습니다!!