임베디드시스템소프트웨어 프로젝트 발표

20161577 김인호

Topic – Hangman Game

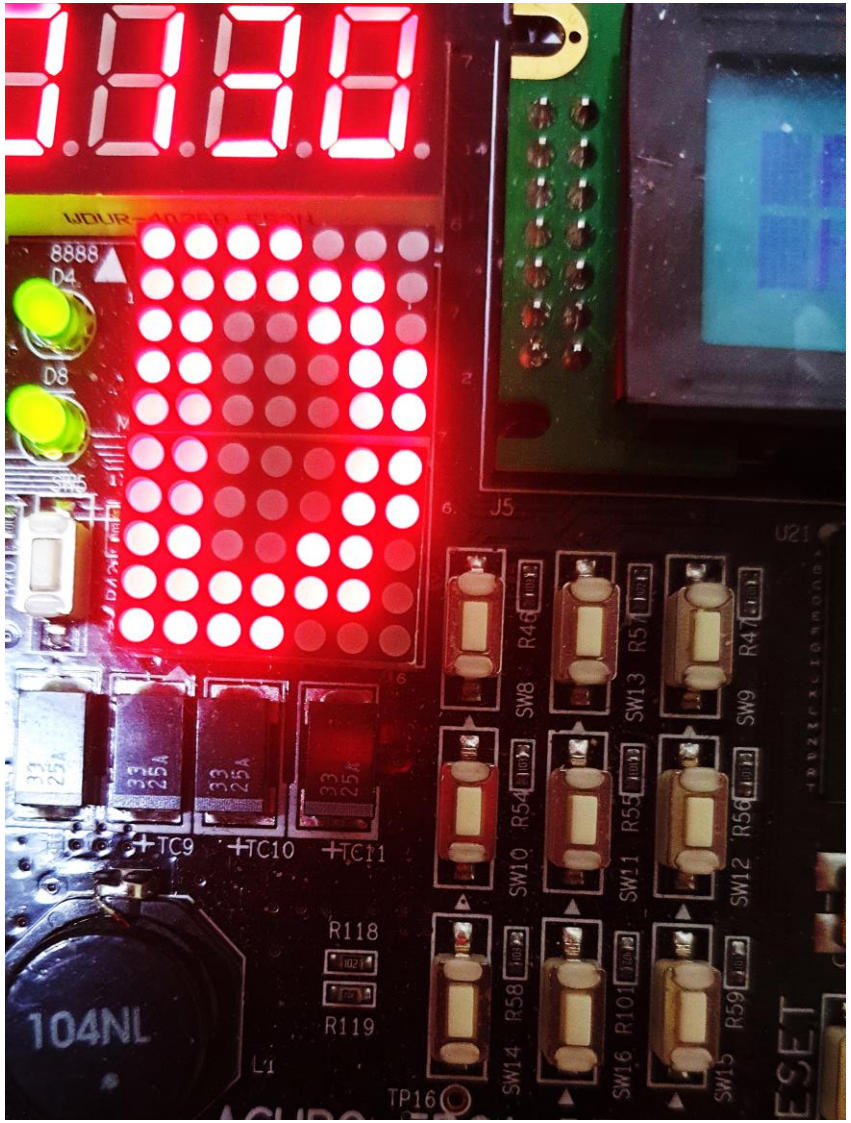
영어 단어를 한 글자 씩 맞추는 게임

단어마다 제한된 도전 횟수

긴 단어일 수록 점수를 더 부여

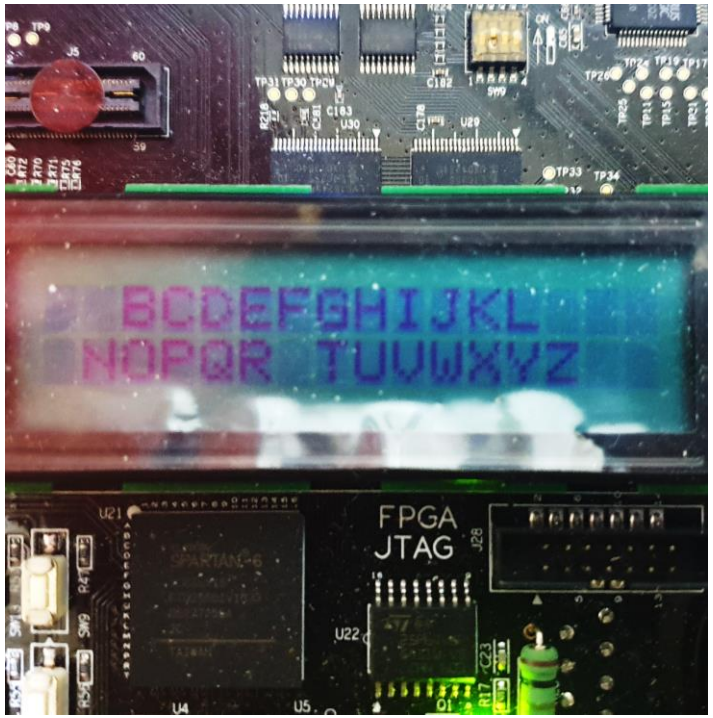
Features – Letter selection

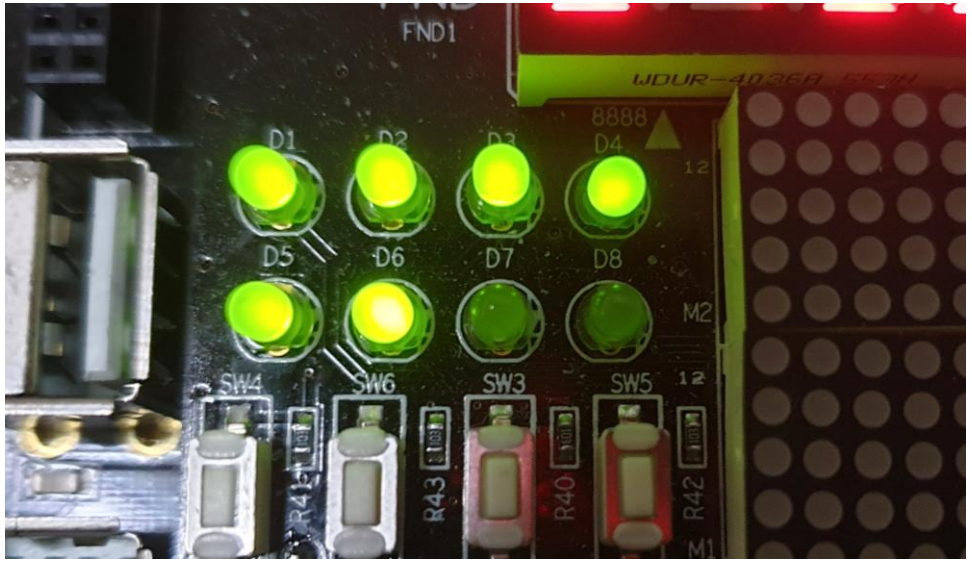
- FPGA의 switch로 글자를 선택
- 선택된 글자는 Dot Matrix에 출력
- 각 switch 마다 알파벳 순서대로 3개씩 번갈아 가며 선택이 되고 마지막 switch는 Y와 Z 이 2가지만 번갈아 가며 선택
- 현재 Dot Matrix에 출력 된 글자로 도전을 하고 싶을 때, Home button을 눌러서 도전
- 주기적으로 switch 입력 확인 (커널 타이머 이용)



Features – Attempts tracker

- 각 단어를 맞추기 시작할 때마다 FPGA의 Text LCD 디바이스에 26개 알파벳이 모두 출력
- 각 글자로 도전을 할 때 마다 Text LCD에서 해당 글자가 있던 자리가 빈칸으로 바뀜
- 단어가 긴 경우 어떤 글자를 이미 사용했는지 기억하고 있기가 불편하기 때문에 이를 쉽게 알기 위해 Text LCD로 오른쪽 그림과 같이 기록

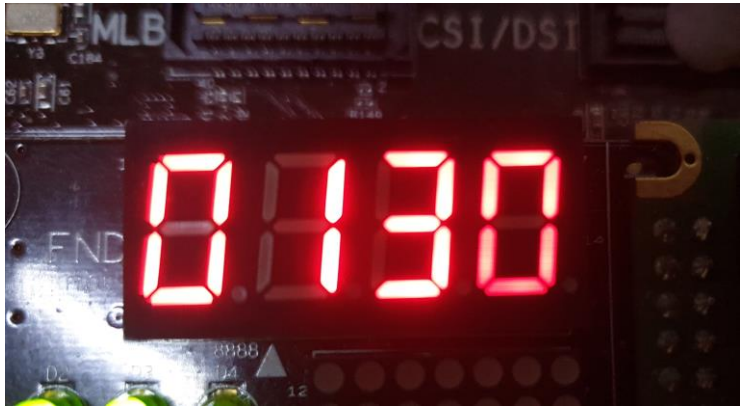




Features – Lives

- 각 단어마다 '목숨'이 8개 부여
- 단어에 없는 글자로 도전하거나 이미 도전한 글자로 도전하면 1씩 감소
- 남은 '목숨' 만큼 LED가 켜지며 모두 꺼지면 게임이 종료

Features – Scoring system



- 누적 점수는 FND 디바이스에 출력
- 한 단어를 맞추면 단어의 길이 x 10점 만큼의 기본 점수 부여
- 단어의 길이 x 5초 내에 맞추면 보너스 점수 50점 추가 부여 (커널 타이머 이용)

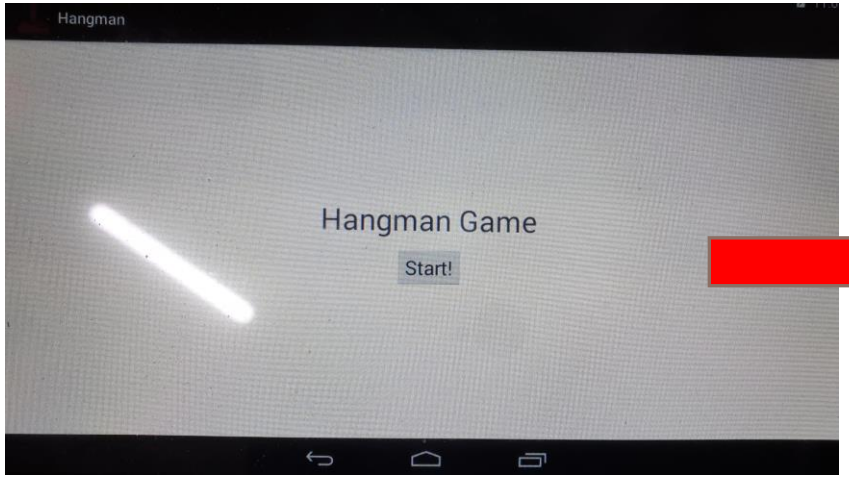
Features – Interrupts

현재 Dot Matrix에
선택되어 있는
글자로 도전

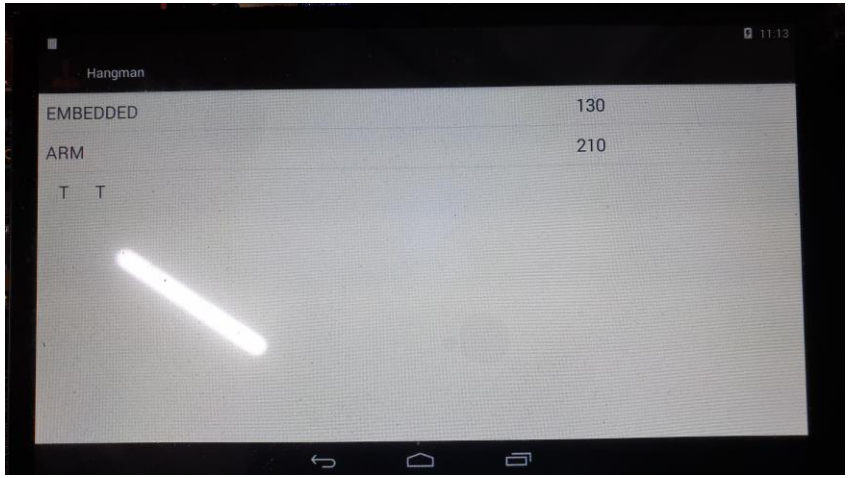
현재 단어 포기하고
다음 단어로 이동

3초 이상 누르고
있을 시 종료
(커널 타이머 이용)

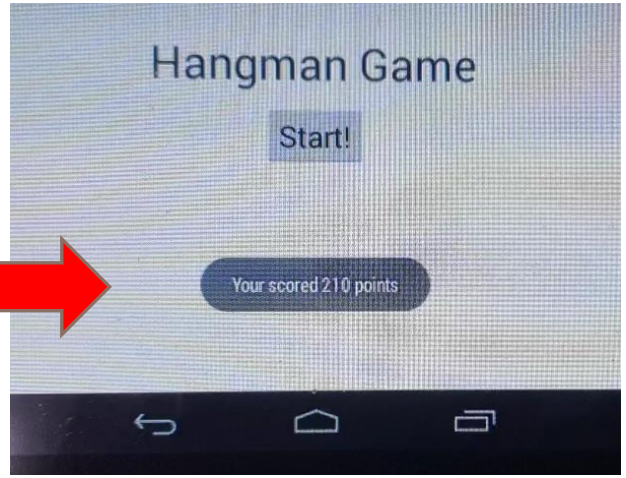
Features – Android application



시작 activity에서
Start 버튼을 누르면 시작



게임을 진행하면서 맞춘
단어와 맞추고 있는 단어가
ListView에 쌓임



종료 조건이 만족되면
첫 activity로 돌아가면서
Toast로 점수 표시

Android NDK (JNI)

```java
public native void startHangman(int fd, Payload pl);
```

```java
Payload pl = new Payload();
startHangman(fd, pl);
```

Payload class object를 parameter로 넘겨서 native 함수에서 해당 object의 field에 값을 지정

```java
/*
 * Payload between Module and App
 */
class Payload {
    String word;
    int status;
    int score;
}
```

```
/*
 * Calls `ioctl()` to wait for Module's input and `read()` to get the data from the Module.
 */
void JNICALL Java_com_example_hangman_MainActivity_startHangman(JNIEnv *env,
        jobject this, jint fd, jobject ret) {

    jclass retClass = (*env)->GetObjectClass(env, ret);
    jfieldID jwordID = (*env)->GetFieldID(env, retClass, "word", "Ljava/lang/String;");
    jfieldID jstatusID = (*env)->GetFieldID(env, retClass, "status", "I");
    jfieldID jscoreID = (*env)->GetFieldID(env, retClass, "score", "I");

    jstring word;

    ioctl(fd, IOCTL_READ_LETTER);
    read(fd, &payload, sizeof(payload));

    word = (*env)->NewStringUTF(env, payload.word);

    ...

    (*env)->SetIntField(env, ret, jstatusID, payload.status);
    (*env)->SetObjectField(env, ret, jwordID, word);
    (*env)->SetIntField(env, ret, jscoreID, payload.score);

    ...
}
```