**REPORT**

Term Project 레포트

과목명 : 임베디드시스템

제출일 : 2018-12-08

2014154001 컴퓨터공학과 강준혁

2014154007 컴퓨터공학과 김병갑

2014152037 컴퓨터공학과 차해운



# 목 차

# 개요 --------------------------- 3p

# 팀 활동 내역 ----------------- 4p

# 설계 -------------------------- 5p

# 소스 코드의 구성 ------------ 7p

# 실행 결과 및 분석 ----------- 11p

# 결론 ------------------------- 14p

# 참고문헌 --------------------- 15p

# 개요

## 개발목표 및 요구사항 분석

- 개발목표 : ACHRO 5250의 FPGA 보드와 Android 태블릿에서 실행하는 Application을 연동한 Microwave System을 개발하고자 한다.

* Android의 태블릿 화면을 통해서 FPGA 디바이스를 조작한다.
* FND Device를 통해서 전자레인지 작동 시 남은 조리시간을 표시한다.
* Moter Device를 사용하여 전자레인지가 작동하는 모습을 시연한다.
* Buzzer가 조리의 종료 및 일시정지를 알린다.
* Text LCD Device는 태블릿 화면에서 작동버튼을 선택했을 때, 어떤 내용을 작동하는지 출력해주며 또한 조리가 완료되었을 때 종료내용을 출력해준다.

- 요구사항 분석 : 시간을 숫자로 표기, 상태에 대한 텍스트 표기, 멈춤 시 부저 알림, 전자레인지 내부 돌림판을 돌리는 모터 작동. 안드로이드로 사용자에게 시각적 편의 제공.

## 해결방안

- ACHRO-5250의 FPGA보드의 각 장치들을 사용하여 FND, Motor 등의 기능을 수행하며, Android Application에서 JNI코드의 공유라이브러리 파일을 Load해서 사용자의 입력이 App을 통해 보드를 제어할 수 있도록 한다.

## 팀 구성 역할분담

김병갑 : Android Application 개발 (.xml, .java)

강준혁 : 프로젝트 발표, 시스템 구성 및 분석, PPT 제작

차해운: Device Driver 분석, JNI 소스코드 작성과 연동

2. 팀 활동 내역

|  |  |  |
| --- | --- | --- |
| 주차 | 날짜 | 활동 내역 |
| 1 | 11-13 | 프로젝트 주제 선정 |
| 1 | 11-15 | 필요한 디바이스 선정 및 시나리오 작성 |
| 2 | 11-20 | 제안서 작성 |
| 2 | 11-22 | 안드로이드 화면 구성 |
| 3 | 11-27 | JNI 소스코드 작성 |
| 3 | 11-29 | 안드로이드 와 FPGA디바이스 연동 |
| 4 | 12-04 | 작성된 프로그램 시연 및 수정 |
| 4 | 12-06 | 프로그램 오류 수정 및 보완 |

3. 설계

* 설계 개요

장비 시작 전에 설계해 둔 쉘 파일을 사용하여 안드로이드로 모듈을 적재한다. 드라이버 설치 이후, FPGA와 안드로이드를 사용하여 ACHRO-5250의 장치인 FPGA로 장치 모듈을 활성화하며, 안드로이드의 어플 중 xml을 통해 시각화하여 쉽게 접근할 수 있는 Microwave 시스템을 설계한다. 사용자가 원하는 조리모드와 시간을 버튼으로 입력해서 시간을 설정하고, Start, Pause, Finish, Exit 버튼 등으로 사용자가 어플을 이용하여 장치를 조작할 수 있도록 설계한다.

각 요구사항에 맞게 사용하는 장치들에는 몇 가지 제약사항이 있다.

1. FND가 숫자 4자리 이외는 지원을 하지 않는다.
2. TEXT\_LCD는 32자의 char 배열로 되어있으며, 그 이상을 지원하지 않는다.
3. Buzzer의 소리는 하나의 소리패턴 이외는 지원을 하지 않는다.
4. Motor의 속도는 일정 구간의 속도는 소음을 발생한다.

* 설계 내용
* ACHRO-5250의 FPGA 보드 장치 : FND, Text LCD, Step Motor, Buzzer
* Android Application 소스코드에서 JNI파일의 공유라이브러리 로드
* 호스트 PC에서 타겟보드로 장치사용에 필요한 Device Driver 들을

/storage/sdcard0 디렉토리에 옮긴 후 insmod로 모듈을 적재한다.

이후 mknod명령으로 /dev 위치에 장치파일을 생성한다.

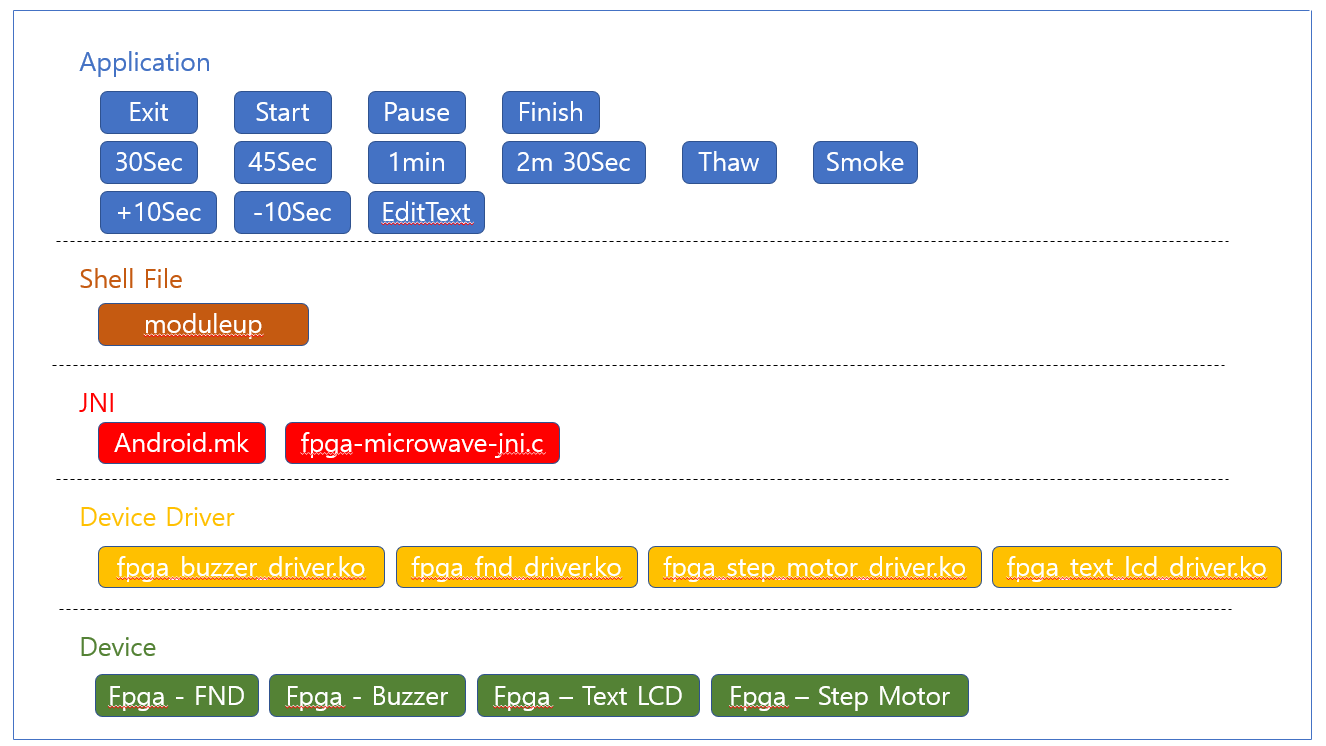
* 각 모듈의 적재를 손쉽게 하도록 쉘 스크립트 파일을 실행.
* Android Application에서 설계한 기능
  + Microwave의 기능에 맞게 시간설정을 한 후 동작시킬 수 있다.
  + 시간 설정관련 버튼은 30sec, 45sec, 1min, 2min 30sec, Thaw(해동), Smoke(훈제) 버튼이 있으며, 각 버튼을 클릭 시 초 단위 버튼은 Edittext 부분에 시간이 입력되고, -10sec / +10sec 버튼으로 10초단위로 시간을 추가하거나 줄일 수 있다.
  + 이때 해동 / 훈제 버튼을 누르면 모터속도가 해동일땐 빨라지고, 훈제일 땐 느려지도록 설계하였다.
  + Start / Pause / Finish / Exit 버튼으로 각 내용에 맞는 기능을 수행한다.
* 시간을 설정하고 Start 버튼을 클릭하면 Text LCD에 각 기능 별 알맞는 텍스트를 출력한다. (30sec으로 Start 시에 Start!! Set : 30 출력)

이와 함께 데모를 위한 모터가 설정한 속도(10)으로 돌아가며, 설정한 시간이 종료되거나, 일시정지를 누를 경우 Buzzer에서 경고음이 출력된다.

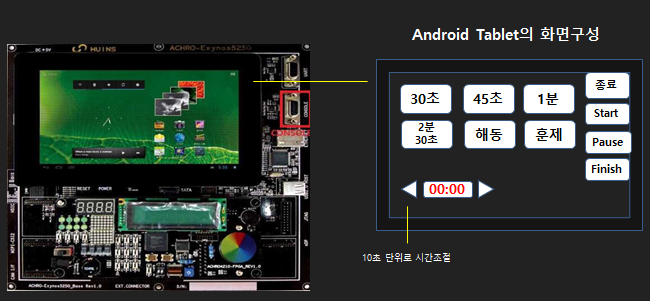
또한 중간에 시간을 추가할 수 있으며, 줄일 수도 있도록 설계하였다.

4. 소스코드의 구성

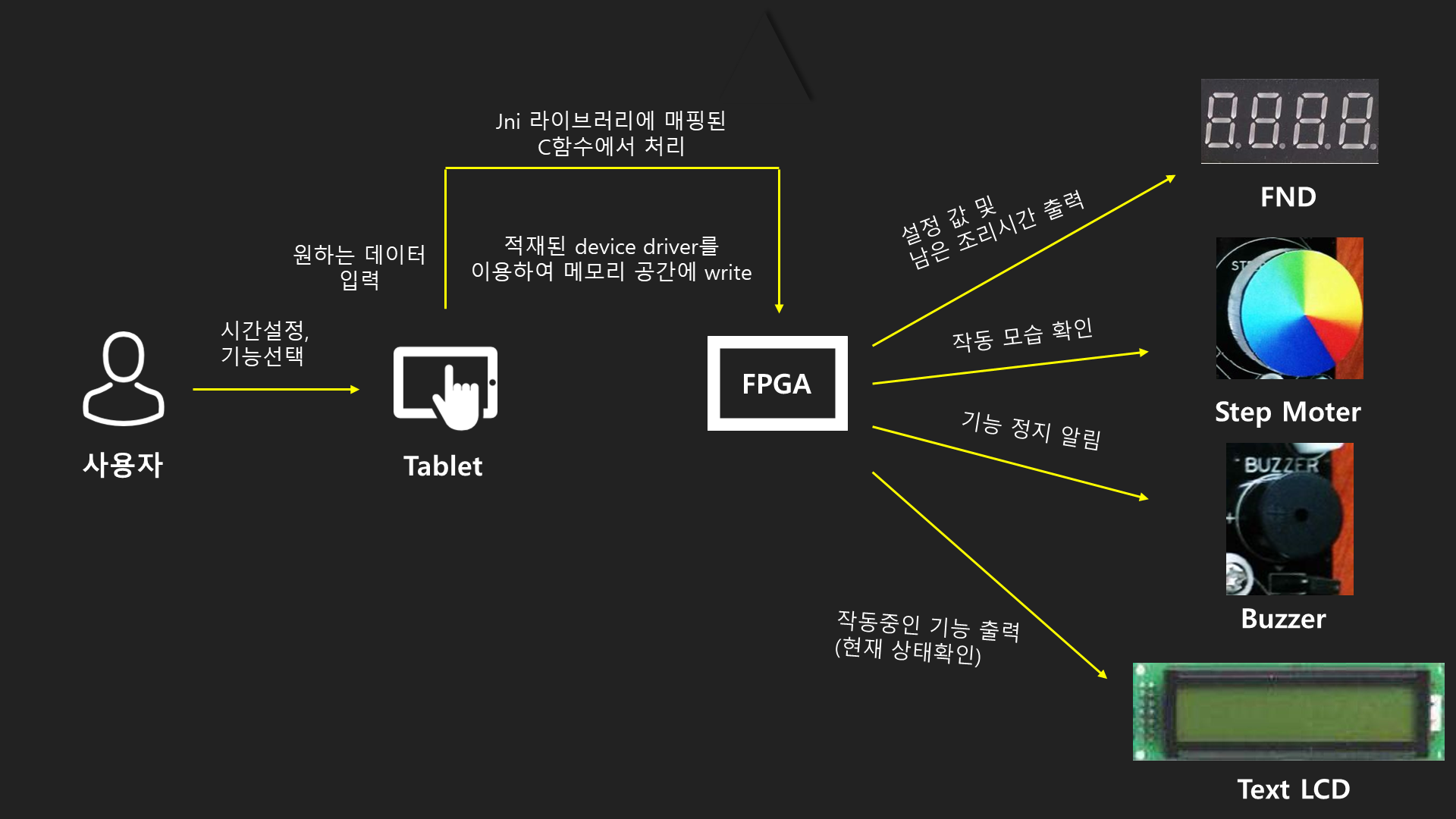
[SW 구성도]

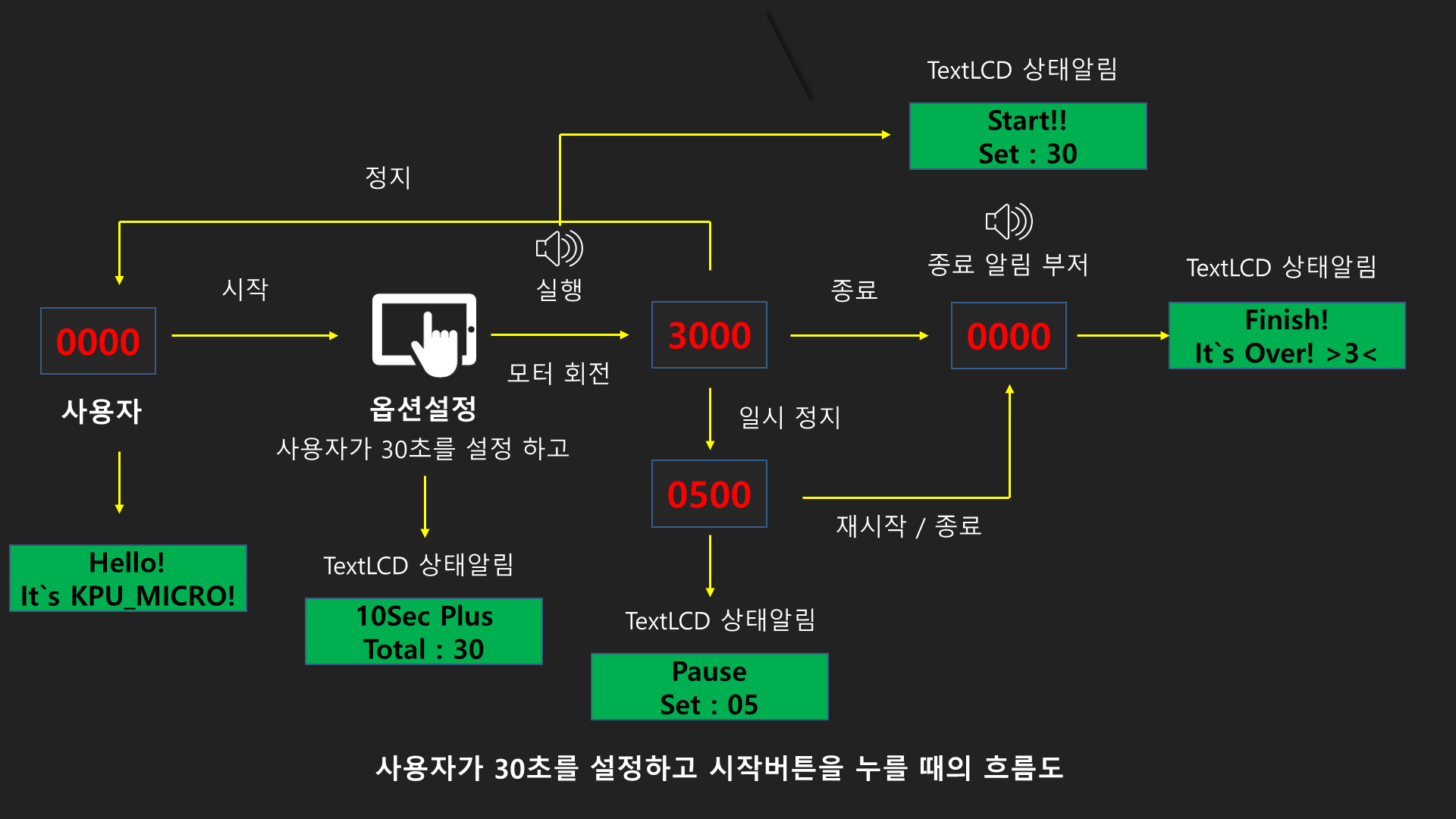






[시나리오(흐름도)]





[핵심함수 설명]

**<Android의 FpgaFndActivity.java>**

각 버튼들(시간 관련버튼, Start, Pause, Finish, Exit)은 setOnClickListener으로 클릭 이벤트 처리를 실행한다. 시간 관련 버튼을 누르면 각 시간에 맞는 초가 timesec 변수에 변화를 주며, edittext로 timesec의 값을 [분:초] 형태로 보여준다.

startButton의 Onclick() : start버튼을 클릭하면 CounterTask 클래스를 시작하며 모드에 따라 모터를 동작시킨다. mode가 0일땐 일반모드로 모터의 속도를 10으로 설정한 후 동작시키고, 1일땐 해동모드로 모터의 속도를 5로 빠르게 동작시키며, 2일땐 훈제모드로 모터의 속도를 15로 느리게 동작시킨다. 이후 Text Lcd에

“Start!! Set : ”+ settingTime을 출력한다.

pauseButton의 Onclick() : 일시정지 버튼이며, 클릭하면 모터가 멈추고 Text LCD에 “Pause Set : ” +settingTime을 출력한다.

finishButton의 Onclick() : 정지 버튼이며, 클릭하면 timesec를 0으로 설정 후 모터가 멈추고 Text LCD에 “Finish! It’s Over!” 을 출력하며 FND에 0을 전달한다.

class CounterTask extends AsyncTask<> { } :

Start 버튼을 누르게 되면 AsyncTask를 상속받은 CounterTask클래스가 실행된다. 이는 여러 기능에 대한 변화를 비동기 처리하기 위해 사용되었다.

doInBackground메소드에서 1초가 지났을때 timesec 변수의 값을 -1 해주게되면,

onProgressUpdate메소드에서 timesec의 변수값에 따라 다시 FND로 데이터를 넘겨주고 UI상의 edittext의 값에도 변화를 주게 된다.

doInBackground에서 while문이 종료되어서 timesec를 return하면, onPostExcute 메소드에 진입한다. 이 메소드에서 모터를 정지한 후 2초간 부저를 울리도록 하며, TextLcd 값을 “Finish! It’s Over!” 으로 변경한다.

public native int ReceiveFndValue() : 네이티브 라이브러리 함수로써, 입력받은 데이터를 Load한 JNI의 라이브러리파일(.so)를 통해 FND 장치에 값을 전달한다.

이때 반환값으로 디바이스에 접근이 제대로 되었는지를 확인할 수 있다.

public native int ReceiveTextLcdValue() - Text LCD 데이터 전달함수

public native int ReceiveBuzzerValue() - Buzzer 데이터 전달함수

public native String SetMotorState() - Step Motor 데이터 전달함수

: 나머지 3가지 함수도 FND와 같은 방식으로 데이터가 전달된다.

System.loadLibrary(“fpga-microwave-jni”) : 라이브러리 파일을 로딩하는 함수.

개발한 프로젝트에 사용된 jni의 소스코드는 fpga-microwave-jni.c 코드이며, 이를 컴파일 한 결과로 libfpga-microwave-jni.so 파일이 생성되어 사용된다.

**<fpga-microwave-jni.c>**

JNIEXPORT int JNICALL Java\_achro4\_fpga\_ex2\_FpgaFndActivity\_ReceiveFndValue()

: java코드에서 FND데이터를 전달하면 이 함수에서 데이터를 전달받는다. 매개변수 중 jstring val 을 통해서 받으면 이를 변환하여 fpga\_fnd 함수로 데이터를 넘긴다.

int fpga\_fnd() : 함수가 호출되어 데이터를 받으면 해당 디바이스를 open()으로 연 후에 write() 함수를 통해 디바이스로 입력값을 전달하도록 되어있다.

나머지 3가지 함수도 위와 비슷한 기능으로 구성되어있다.

**<moduleup - 쉘 스크립트>**

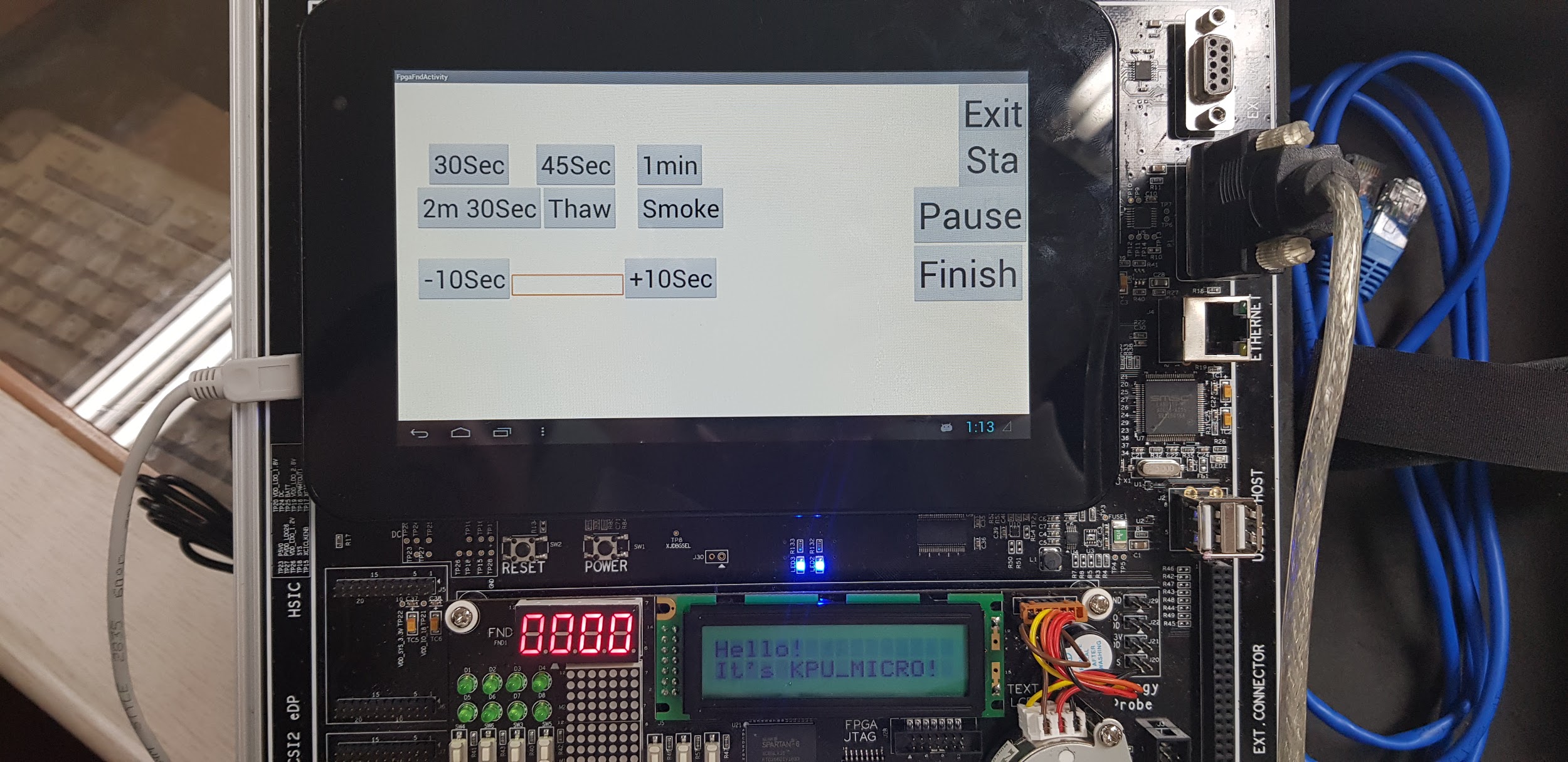
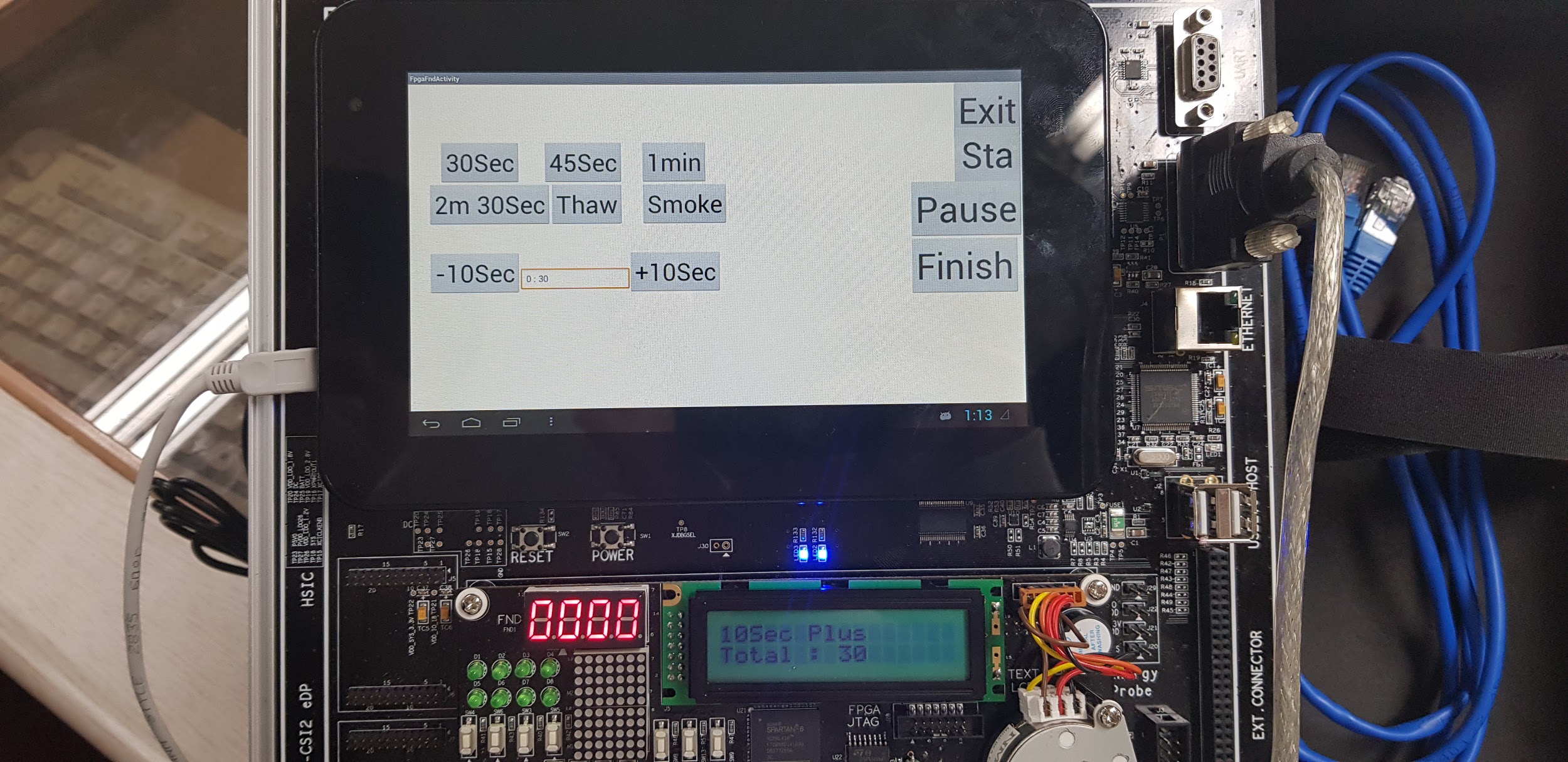
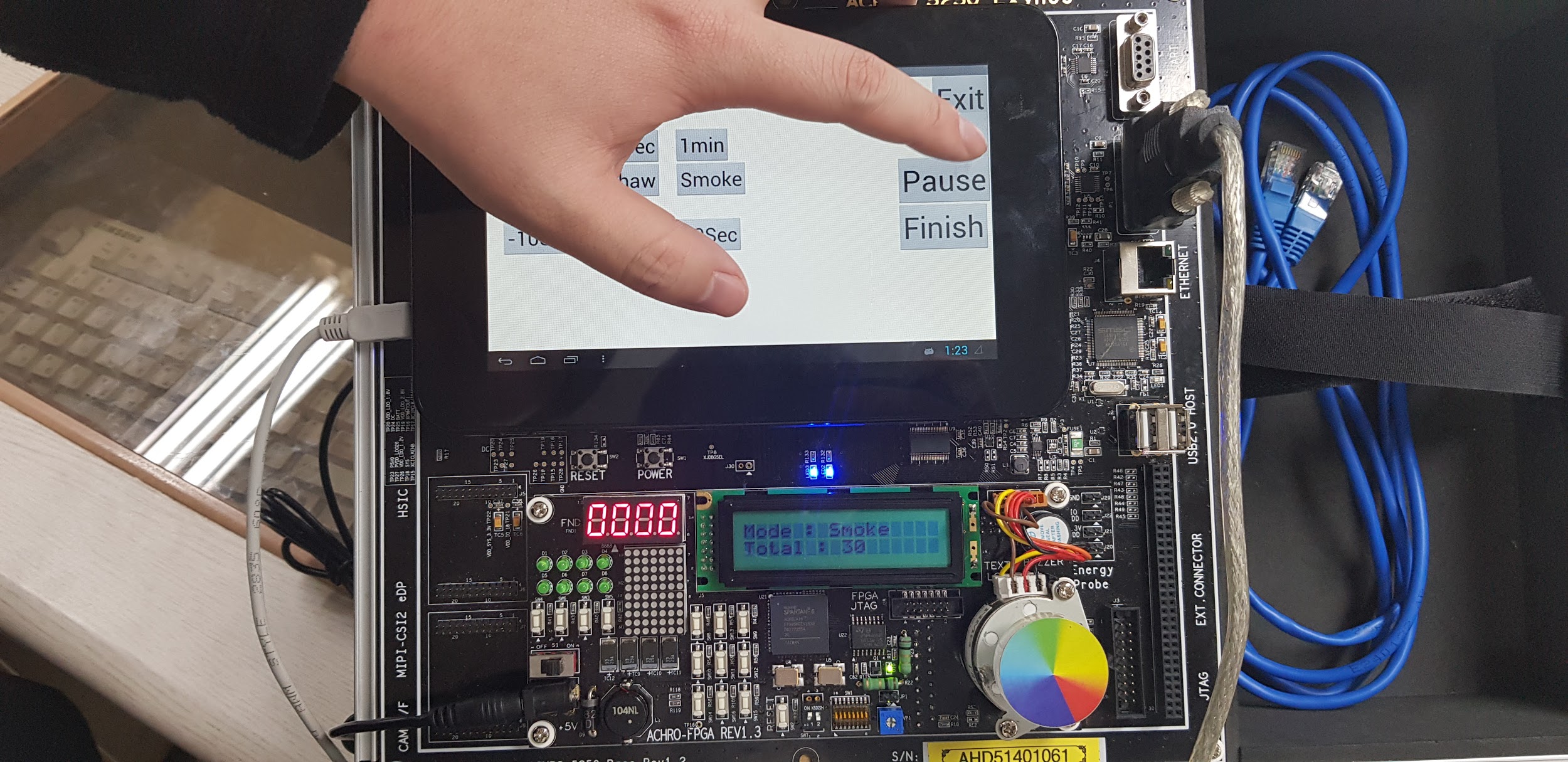
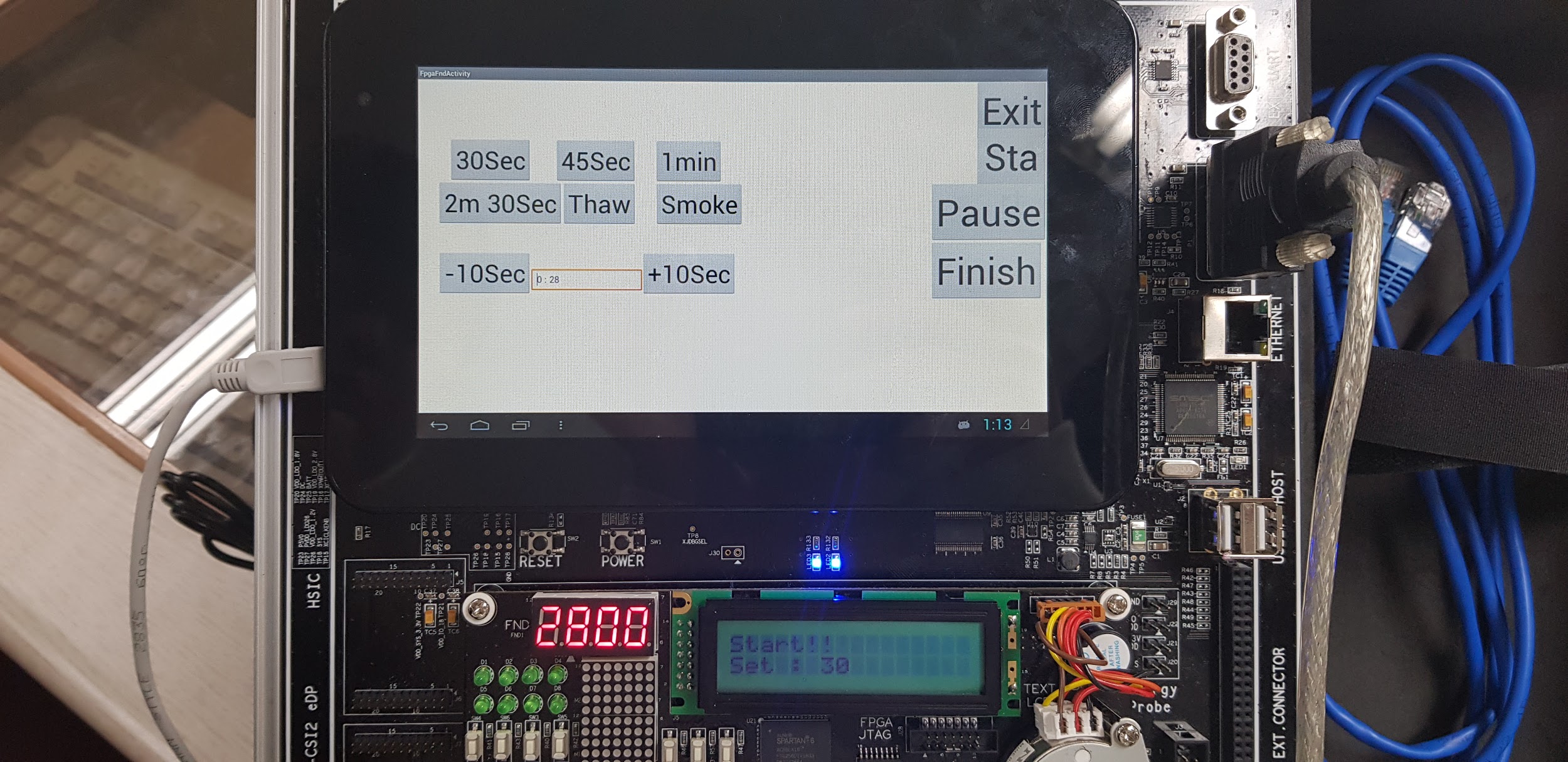
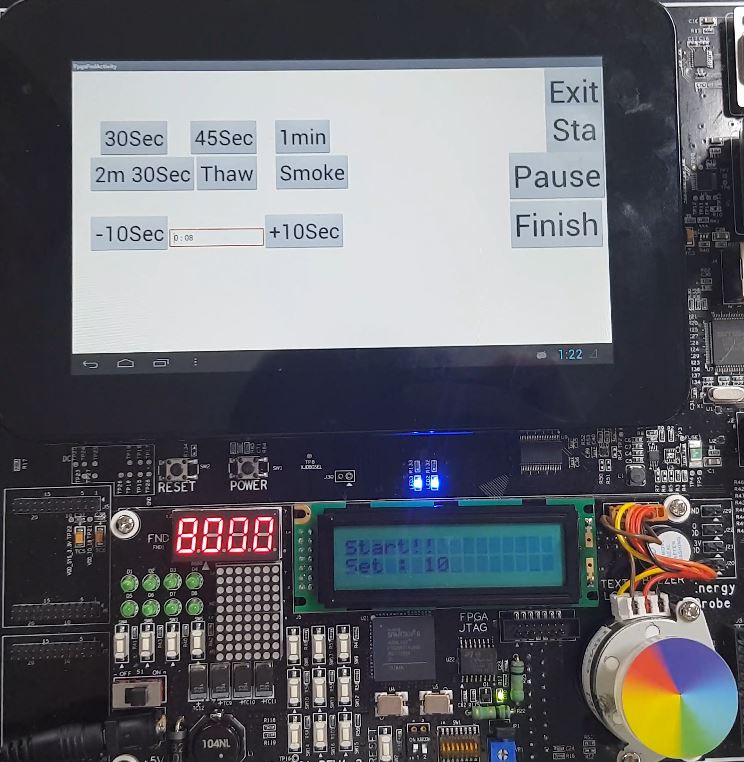
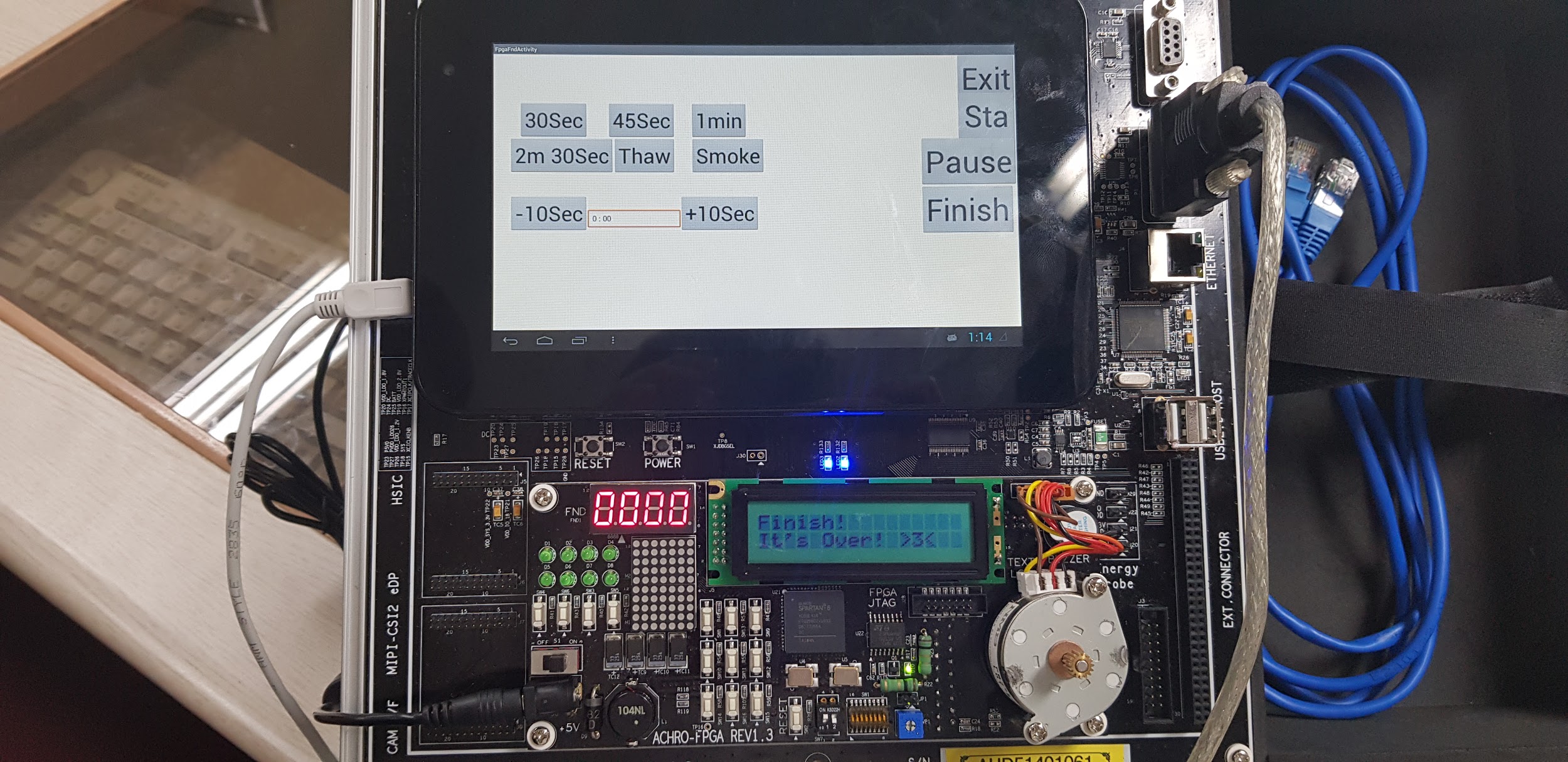
4가지 기능의 모듈을 보드에 쉽게 insmod와 mknod하기 위해서 작성했다.

각 파일의 insmod와 mknod의 코드로 구성되어있다.

sh moduleup 명령으로 실행하면 된다.

5. 실행 결과 및 분석

* 실행결과와 분석

1. 어플 시작 : 어플을 실행하면 Text LCD에 문구가 출력되고, FND는 0000 으로 초기화 한 상태로 어플이 시작된다.  
   
2. 시간 설정(ex. 30초) : 예를들어 30초를 설정하려고 할 때, 30sec를 눌러서 30초를 설정, 혹은 +10sec을 3번 눌러 30초를 설정한다.  
   
3. 훈제모드 : 30초를 설정 한 후 , 일반모드로 실행하려면 바로 start를 누르지만 훈제모드로 시작할 땐 Smoke 버튼을 눌러 시작한다. 해동모드로 시작하고 싶으면 Thaw버튼을 누른다. 
4. 작동 (30초 -> 28초) : 쓰레드를 비동기로 사용하여, UI에 시간 초 변화가 표시된다. 또한 FND에서 숫자가 변경되고, 모터도 함께 동작되는 모습이다. 
5. 작동 (10초 -> 8초)
6. 종료 : 종료 시에는 2초간 부저가 울리고 모터 정지가 일어나며, TextLCD에서 해당 내용을 전달한다.  
   

6. 결론

* 평가, 반성, 향후 계획

지금까지 ACHRO-5250과 FPGA, JNI코드를 포함한 안드로이드를 기반으로 만든 MicroWave를 제작하였다. Application의 시각적인 수단(GUI)을 이용하고자 mmap 등을 사용하지 않고 안드로이드와 JNI를 이용하였으며, FPGA 장치를 사용하여 모션을 보여주고자 각 장치의 드라이버 모듈을 적재하여 작동시키며 임베디드 텀프로젝트를 진행했다. 프로젝트를 진행하며 임베디드가 어떤 방식으로 돌아가는 지에 대해 알아보는 시간은 정말 큰 도움이 되었고, 그에 대해 견식도 그만큼 넓어졌다. Android와 임베디드 보드 간에 데이터 전송에는 JNI코드의 라이브러리 파일을 이용한다는 점, 디바이스를 제어하기 위해 가상메모리 매핑을 해야 한다는 점 등 많은 것을 배우는 시간이 되었다.

프로젝트를 진행하며 디바이스 드라이버를 개선해야 할 때 코드의 어떤 부분을 고쳐야 할지를 알지 못했던 부분은 반성할 점이다. 이 부분에 대해서는 디바이스 드라이버의 코드에 대해 더욱 공부해야 할 부분이다. 디바이스 드라이버 코드를 개선하는 대신 쉘 스크립트를 작성해서 모듈을 한꺼번에 올리고 드라이버 파일을 한꺼번에 제작하는 것을 대신 제작하였다. 또한 아쉬운 점은 FND에 시간 초를 출력할 때 오른쪽으로 정렬해서 출력했으면 더욱 좋았을 것이다.

그리고 프로젝트 관련 코드를 전부 작성 후 알게 된 것이지만, java파일의 파일명을 mainactivity가 아닌 FpgaFndActivity로 사용해서 바로 고치지 못한 것이 아쉽다.

향후에는 지금까지 배우고 사용해보았던 내용들을 임베디드 보드 뿐만아니라 라즈베리파이, 아두이노 등의 Single Board Computer에 적용시켜서 졸업작품 등 임베디드에 관련된 부분에 이용해 볼 것이다.

7. 참고문헌

* 안드로이드 : 그림으로 쉽게 설명하는 안드로이드 프로그래밍 (2005년 생능출판사)
* Achro-5250을 이용한 안드로이드 설계 및 응용 (휴인스 기술연구소)
* Achro-5250을 이용한 리눅스 설계 및 응용 (휴인스 기술연구소)