

2020-2학기 임베디드SW

Term-project 보고서



한국산업기술대학교  
KOREA POLYTECHNIC UNIVERSITY

2014146012 박동훈

# 목차

1. 개요

2. 구성도

3. 순서도

4. 기능설명

5. 실행 화면

## 1. 개요

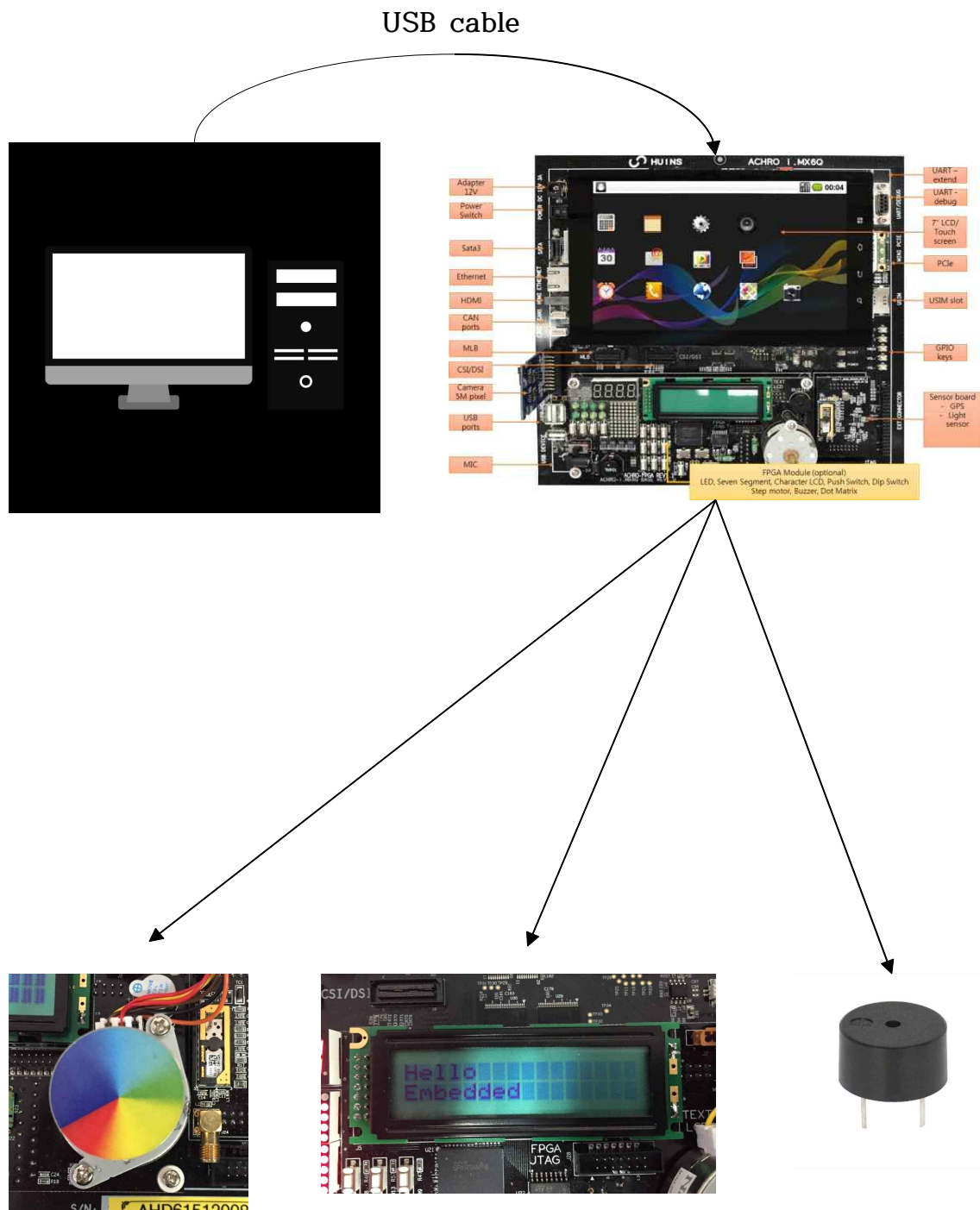
- 간단한 가위,바위,보 게임에 3가지 디바이스를 연결하여 표현해 주기로 했다.

### - 프로그램 동작조건

1. 가위바위보게임이라는 글귀는 300ms 간격으로 깜빡거리도록 한다.
2. ImaveView는 300ms 간격으로 가위, 바위, 보 이미지가 순서적(혹은 랜덤하게) 바뀌도록 한다.
3. (2)에 사용할 이미지는 1벌(3개)을 사용하되 사용자와 컴퓨터의 이미지는 서로 마주보게 출력한다.(y축 대칭)
4. 사용자가 버튼을 누르면 프로그램은 난수를 발생하여 승부를 결정한다.
5. 부저가 울려서 승부가 결정됐음을 알려준다.
6. 승패에 따라 Step Motor에 방향, TextLcd에 나타내준다.
7. 승부를 판단한 후에는 계속 진행할지, 종료할지를 결정하는 버튼을 표시한다.
8. 화면 맨 위에는 사용자와 프로그램과의 승패, 전제 게임 횟수가 표시되도록 한다.
9. 종료 또는 재경기를 실행하면 TextLcd의 화면이 초기화 된다.

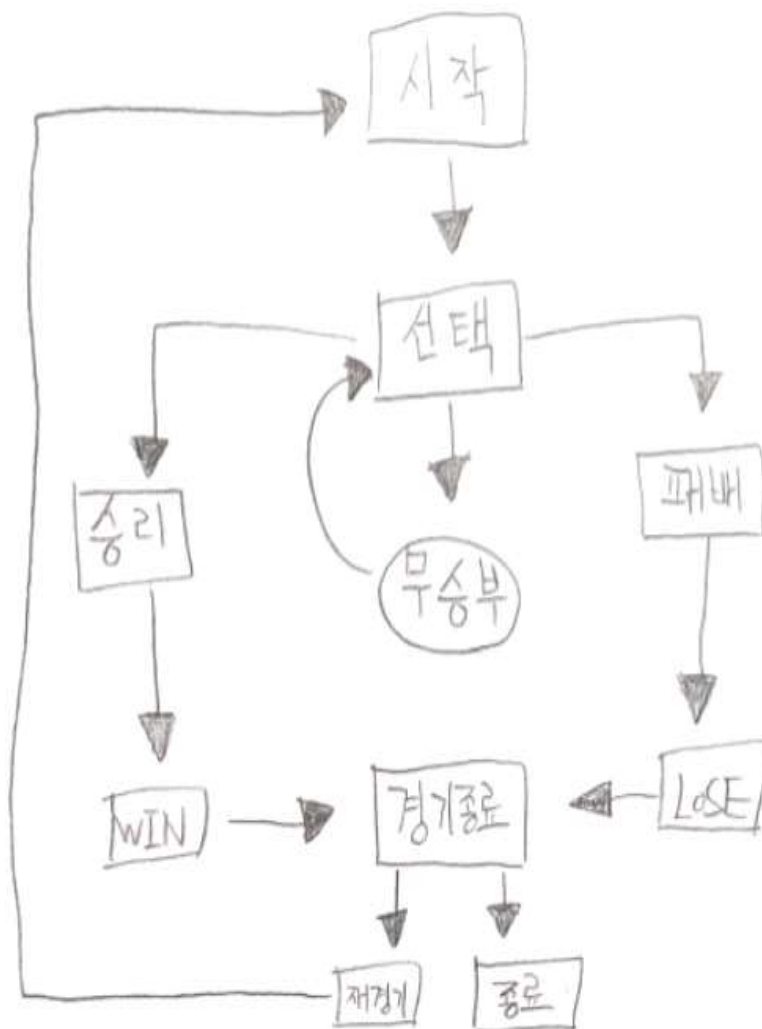
이러한 조건들을 만족시키는 프로그램을 만들어준다.

## 2. 구성도



- 어플에 따라 디바이스들이 동작하도록 한다.

### 3. 순서도



## 4. 기능설명

### 1) 가위,바위,보 제목과 이미지 랜덤으로 나타나도록 하기

```
235 // -----
236 //   제목 깜박거림, 이미지 랜덤 표시
237 // -----
238 public void DrawBlink() {
239     Random rnd = new Random();
240     int n1 = _counter % 3;
241     int n2 = rnd.nextInt( bound: 3);
242
243     _counter++;
244     if (_counter % 2 == 1) {      // 글자는 이미지 절만의 주기로 깜박거림
245         TitleColor = 0xFF0000FF - TitleColor;
246         ((TextView) findViewById(R.id.txtTitle)).setTextColor(TitleColor);
247     }
248     MakeReverse(arrImage[n1]); // 이미지 뒤집기
249     ((ImageView) findViewById(R.id.imageView02)).setImageResource(arrImage[n2]);
250 } // DrawBlink
251
```

- 타이머 핸들러를 이용해 글씨가 깜빡거리는 시간과 가위,바위,보 이미지가 랜덤으로 나타나는 시간을 정해준다.

### 2) 이미지 뒤집기

```
220 // -----
221 //   이미지 뒤집기
222 // -----
223 void MakeReverse(int id) {
224     Bitmap bitmap, reverse;
225     Matrix matrix = new Matrix();
226     matrix.postScale( sx: -1, sy: 1); // y축 대칭
227
228     bitmap = BitmapFactory.decodeResource(getResources(), id);
229     int w = bitmap.getWidth();
230     int h = bitmap.getHeight();
231     reverse = Bitmap.createBitmap(bitmap, x: 0, y: 0, w, h, matrix, filter: false);
232     ((ImageView) findViewById(R.id.imageView01)).setImageBitmap(reverse);
233 } // Make...
234
```

- Bitmap 이미지를 Matrix 함수로 뒤집어준다.

### 3) 승부 판정

```
107 // -----
108 //      승부 판정
109 // -----
110 void PanJung(int you, int phone) {
111     int result;
112     int speed = 5;
113
114     if (you == phone) result = 0; // 비김
115     else if (you - phone == 1 || you - phone == -2) result = 1; // 이김
116     else result = 2; // 짐
117
118     if (result != 0) arrScore[result]++; // 승패 기록
119     arrScore[0]++; // 전체 게임 수
120
121     Display(you, phone, result);
122 }
```

- 가위(0) 바위(1) 보(2)에 각각 숫자를 할당해 주고 간단한 계산을 통해 나오는 값으로 승패를 판정하도록 한다.
- 둘의 숫자가 같으면 비긴다.
- 보(2) > 바위(1) > 가위(0) > 보(2) .... 앞의 숫자에서 다음의 숫자를 빼면 1, 1, -2, 1... 1과 -2가 나오면 “내가” 이긴다는 결과가 나오게 된다.
- 이를 제외한 경우는 “내가” 지는 경우가 된다.

### 4) 경기가 끝나면 ‘재경기’ 와 ‘게임 종료’ 버튼 표시

```
60 // -----
61 //      Button.OnClickListener
62 // -----
63 Button.OnClickListener MyButtonListener = new Button.OnClickListener() {
64     public void onClick(View v) {
65         int you = Integer.parseInt(v.getTag().toString()); // 버튼의 Tag 읽기
66         if (_isRun == true) { // 게임이 진행중이면
67             int phone = new Random().nextInt( bound: 3); // 난수 발생하고
68             _isRun = false; // 타이머 중지
69             SetButtons( flag: false); // [시작]/[게임끝] 보이도록
70             PanJung(you, phone);
71         } else { // [시작]/[게임끝] 중하나를 눌렀을 때
72             if (you == 1) { // [시작] 버튼이면
73                 SetButtons( flag: true); // 버튼을 [가위/바위/보] 상태로 바꾸고
74                 ReceiveTextLcdValue("", ""); //TextLcd 초기화
75                 _isRun = true; // 타이머를 기동하도록 하고
76                 mHandler.sendEmptyMessageDelayed( what: 0, delayMillis: 0); // 타이머 호출
77             } else { // [게임끝] 버튼이면
78                 finish(); // 프로그램 종료
79                 ReceiveTextLcdValue("", ""); //TextLcd 초기화
80                 return;
81             }
82         } // if
83     }
84 }; // OnClickListener
85
```

- 승패가 판정되고 나면 '바위' 와 '보자기' 버튼이 '재경기'와 '경기 종료' 버튼으로 바뀌게 된다.
- ReceiveTextLcdValue에 공백을 넣어줘서 초기화가 되도록 한다.

## 5) 무승부 일 경우에 디바이스 동작

```

123 if (result == 0) {
124     ReceiveTextLcdValue("      Draw", "      GGa b");
125     SetMotorState(action,direction,speed);
126     ReceiveBuzzerValue(x: 1);
127
128     Handler mHandler = new Handler();
129     mHandler.postDelayed(new Runnable() {
130         @Override
131         public void run() {
132
133             SetMotorState(action = 0, direction = 0, speed);
134             ReceiveBuzzerValue(x: 0);
135
136         }
137     }, delayMillis: 2000);
138
139 }

```

- TextLcd에 비겼다는 것을 알려주는 문구가 나타난다.
- 무승부 일 경우 Step Motor는 동작 하지않는다.
- Buzzer는 경기가 끝나자마자 울리고 2초뒤에 종료된다.



## 6) 승리 일 경우에 디바이스 동작

```
140         else if (result == 1) {
141             ReceiveTextLcdValue("    !!Win!!", "!!!feel good!!!");
142             SetMotorState(action = 1 , direction = 0 ,speed);
143             ReceiveBuzzerValue( x: 1);
144
145             Handler mHandler = new Handler();
146             mHandler.postDelayed(new Runnable() {
147                 @Override
148                 public void run() {
149                     SetMotorState(action = 0, direction = 0, speed);
150                 }
151             }, delayMillis: 150);
152
153             mHandler.postDelayed(new Runnable() {
154                 @Override
155                 public void run() {
156                     SetMotorState(action = 1, direction = 1, speed);
157                     ReceiveBuzzerValue( x: 0);
158                 }
159             }, delayMillis: 2000);
160
161             mHandler.postDelayed(new Runnable() {
162                 @Override
163                 public void run() {
164                     SetMotorState(action = 0, direction = 0, speed);
165                 }
166             }, delayMillis: 2150);
167         }
```

- TextLcd에 이겼다는 것을 알려주는 문구가 나타난다.
- Buzzer는 경기가 끝나자마자 울리고 2초뒤에 종료된다.
- Step Motor가 왼쪽(“내가” 위치한 방향)으로 돌아가 승리 했다는 것을 알려준다.
- 일정시간뒤에 Step Motor가 원점으로 다시 돌아간다.

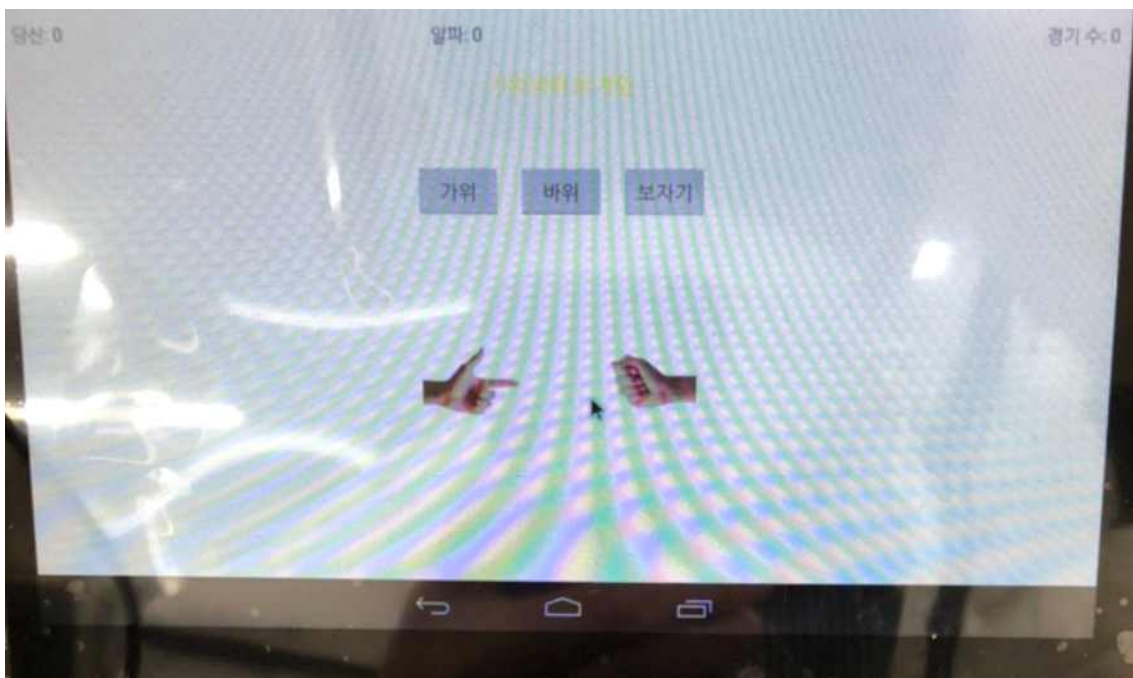
## 7) 패배 일 경우에 디바이스 동작

```
169 if (result == 2) {
170     ReceiveTextLcdValue("    Lose...", "    next time...");
171     SetMotorState(action = 1 , direction = 1 ,speed);
172     ReceiveBuzzerValue( x 1);
173
174     Handler mHandler = new Handler();
175     mHandler.postDelayed(new Runnable() {
176         @Override
177         public void run() {
178             SetMotorState(action = 0, direction = 0, speed);
179         }
180     }, delayMillis 150);
181
182     mHandler.postDelayed(new Runnable() {
183         @Override
184         public void run() {
185             SetMotorState(action = 1, direction = 0, speed);
186             ReceiveBuzzerValue( x 0);
187         }
188     }, delayMillis 2000);
189
190     mHandler.postDelayed(new Runnable() {
191         @Override
192         public void run() {
193             SetMotorState(action = 0, direction = 0, speed);
194         }
195     }, delayMillis 2150);
196
197 }
```

- TextLcd에 졌다는 것을 알려주는 문구가 나타난다.
- Buzzer는 경기가 끝나자마자 울리고 2초 뒤에 종료된다.
- Step Motor가 오른쪽(“알파” 위치한 방향)으로 돌아가 졌다는 것을 알려준다.
- 일정 시간 뒤에 Step Motor가 원점으로 다시 돌아간다.

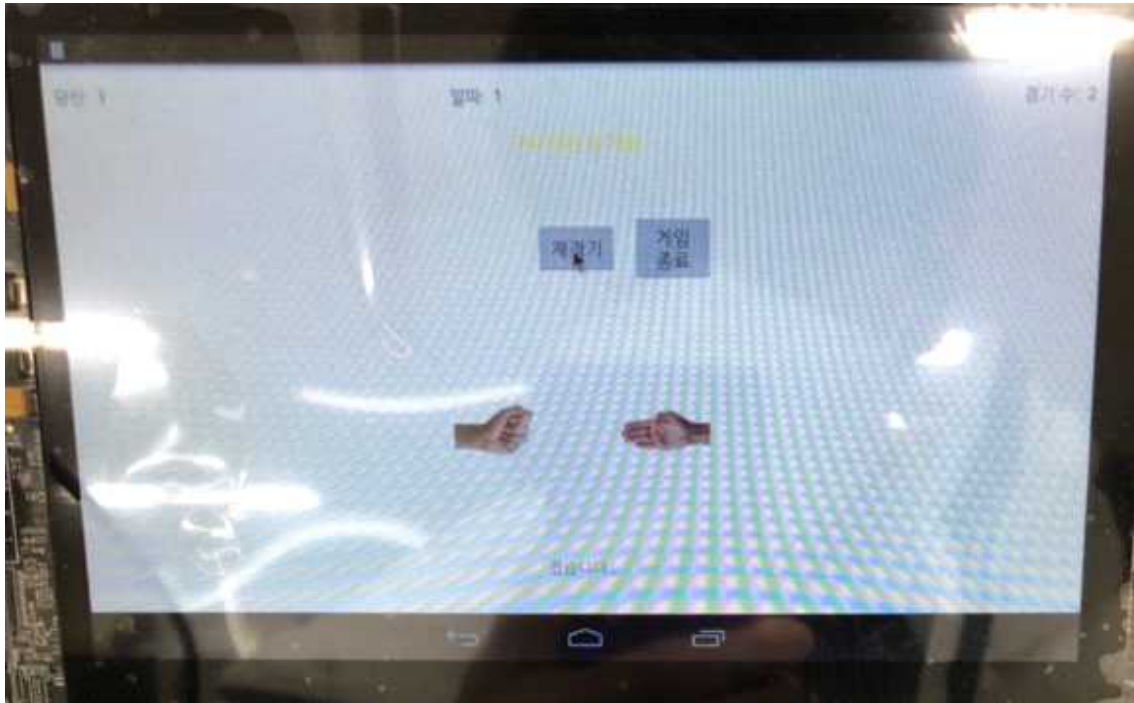
## 5. 실행 화면

### 1) 초기화면



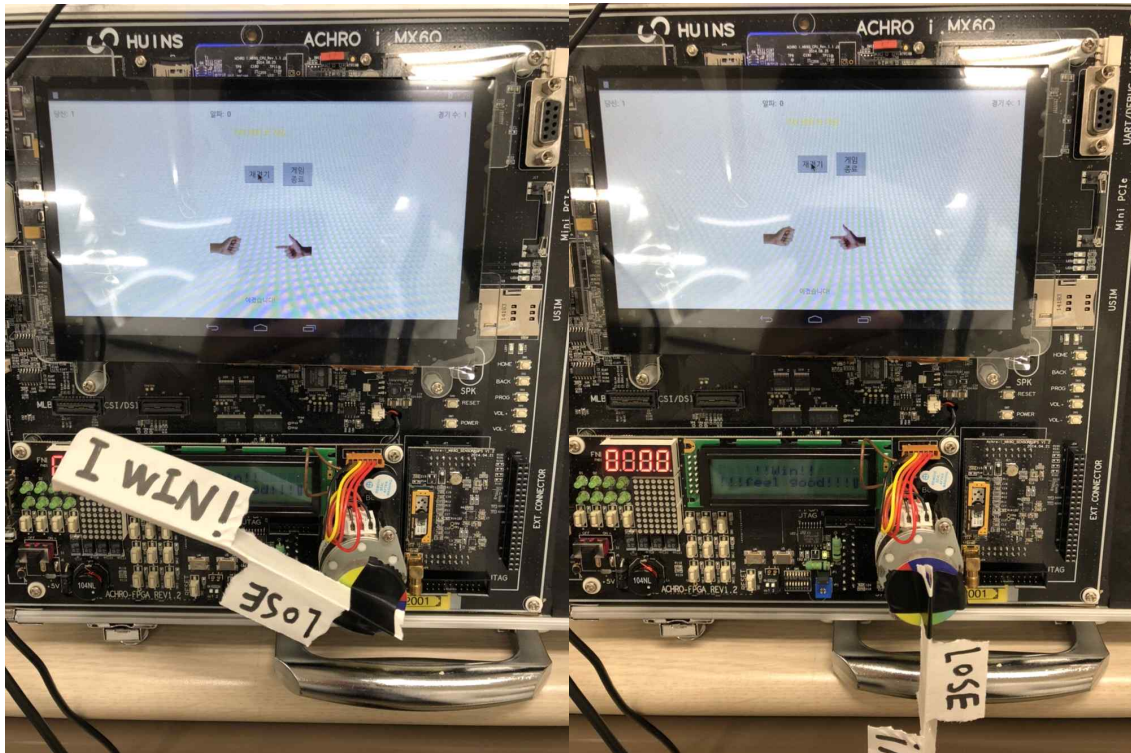
- 화면 위에 '나'와 '알파' 그리고 경기 수를 표시해 준다.
- 사진에는 보이지 않지만 가위,바위,보 게임 이라는 글자가 깜빡거린다.
- 오른쪽 가위 바위 보 이미지는 랜덤으로 나타난다.
- 왼쪽 가위 바위 보 이미지는 원점에 대칭되어 마주 보도록 표시된다.

## 2) 결과화면



- 바위 와 보자기 버튼이 재경기와 게임 종료로 바뀌게 된다.

### 3) 이겼을 경우 동작



- TextLcd에 이겼다는 것을 알려주는 문구가 표시된다.
- Step Motor가 왼쪽으로 돌아가 이겼다는 것을 알려준다.
- Buzzer가 2초동안 울렸다가 종료된다.

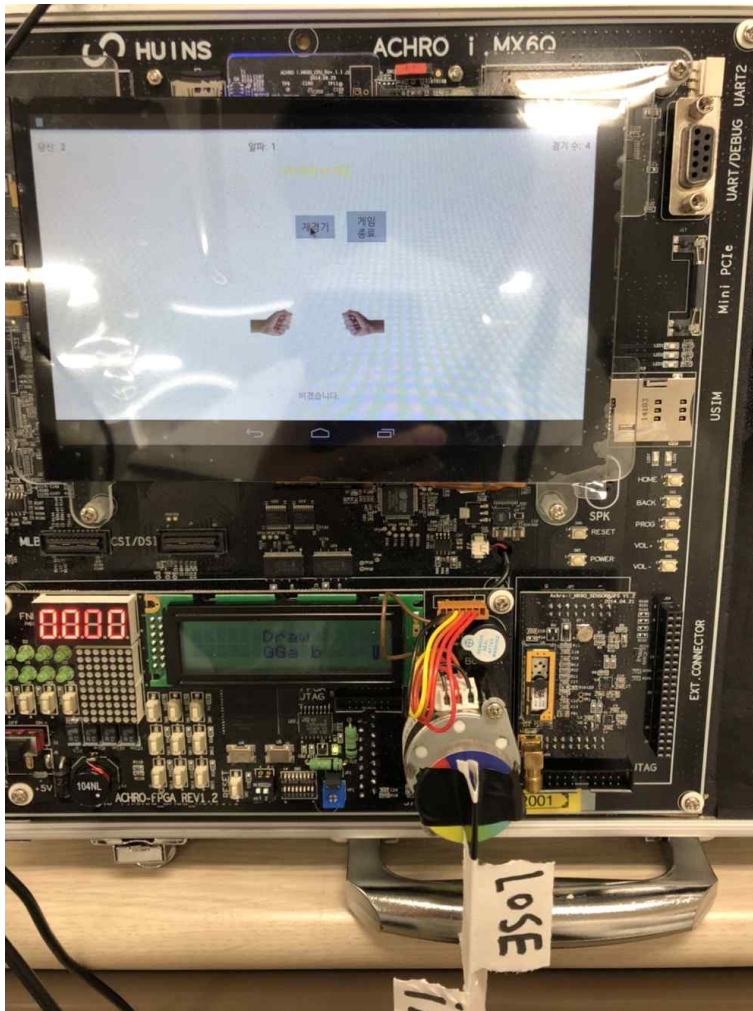


#### 4) 졌을 경우 동작



- TextLcd에 졌다는 것을 알려주는 문구가 표시된다.
- Step Motor가 오른쪽으로 돌아가 졌다는 것을 알려준다.
- Buzzer가 2초동안 울렸다가 종료된다.

## 5) 비졌을 경우 동작



- TextLcd에 비졌다는 것을 알려주는 문구가 표시된다.
- Step Motor는 움직이지 않는다.
- Buzzer가 2초동안 울렸다가 종료된다.

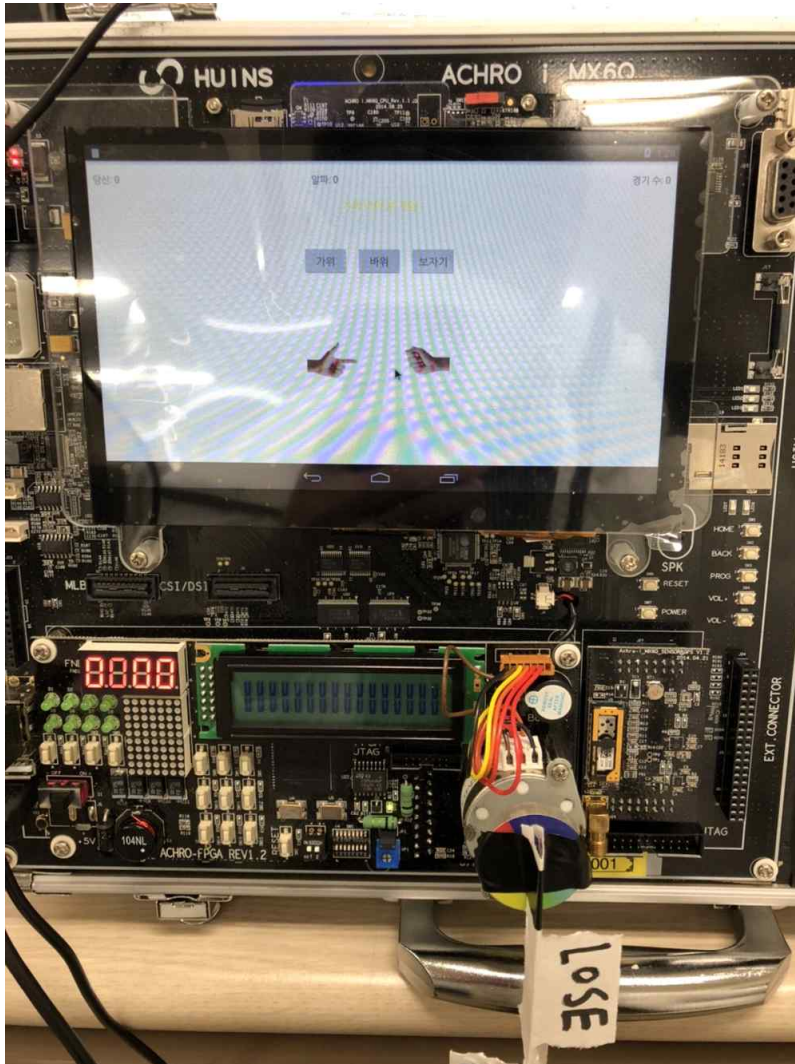
## 6) 게임종료시 동작



- 어플이 종료된다.
- TextLcd 화면이 초기화 된다.



## 7) 재경기시 동작



- 다시 가위 바위 보 버튼이 나타난다.
- TextLcd 화면이 초기화 된다.
- 오른쪽 이미지는 다시 랜덤으로 나타나기 시작한다.