

[



]

- 목차

1. 작품 개요

2. 사용 부품 및 프로그램 소개

3. 구동 영상

4. 코드

5. 개선 할 내용



Part 1

작품 개요



➤ 개요

- 한국의 영어 수준은 선진국에 비해 현저히 낮음.
- 영어를 포기하게 되는 이유 설문 조사 중 가장 큰 비중을 차지 하는 것이 단어를 외우기 방법.
Ex)영단어 낱지 등.
- 현대인에게 영어는 필수적이므로 영단어를 쉽게 받아 들일 수 있는 프로그램을 설계.
- 동의어를 통한 학습으로 같은 뜻의 단어를 연결 지어 학습하는 프로그램 설계.
- 일정량을 외우지 못할 시 반복 학습을 통해 복습 시행.

순위	국가	읽기	듣기	말하기	쓰기	총점
1	아일랜드	24	26	26	25	101
2	오스트리아	24	26	25	24	100
5	싱가포르	24	25	24	24	98
13	캐나다	23	25	24	23	95
	인도	23	24	24	24	95
33	미국	22	23	23	22	90
	말레이시아	22	23	22	23	90
	이탈리아	23	23	22	22	90
45	필리핀	21	22	23	22	88
	프랑스	22	23	22	22	88
57	멕시코	21	23	22	21	86
	인도네시아	21	22	21	22	86
75	대한민국	22	21	20	21	84
	네팔	20	21	21	22	84
83	북한	21	21	20	21	83

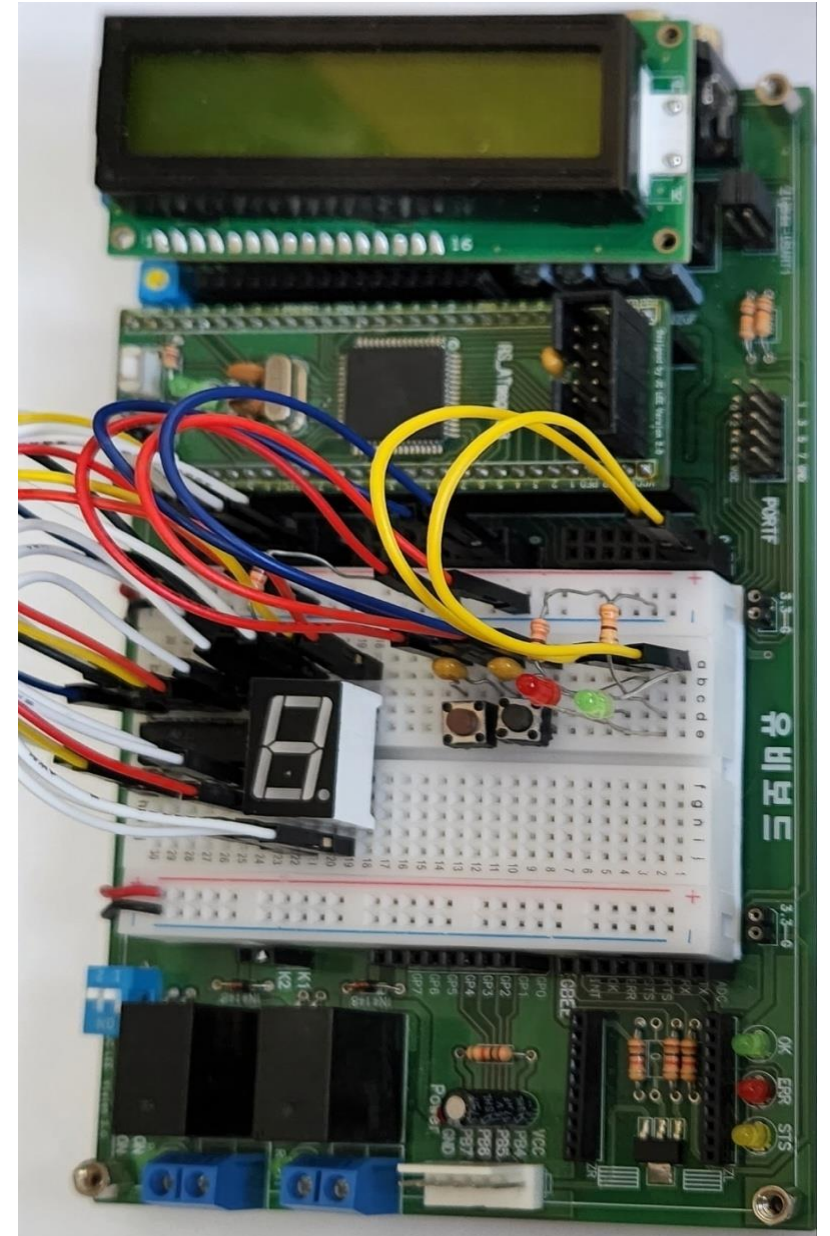
Part 2

사용 부품 및 프로그램 소개



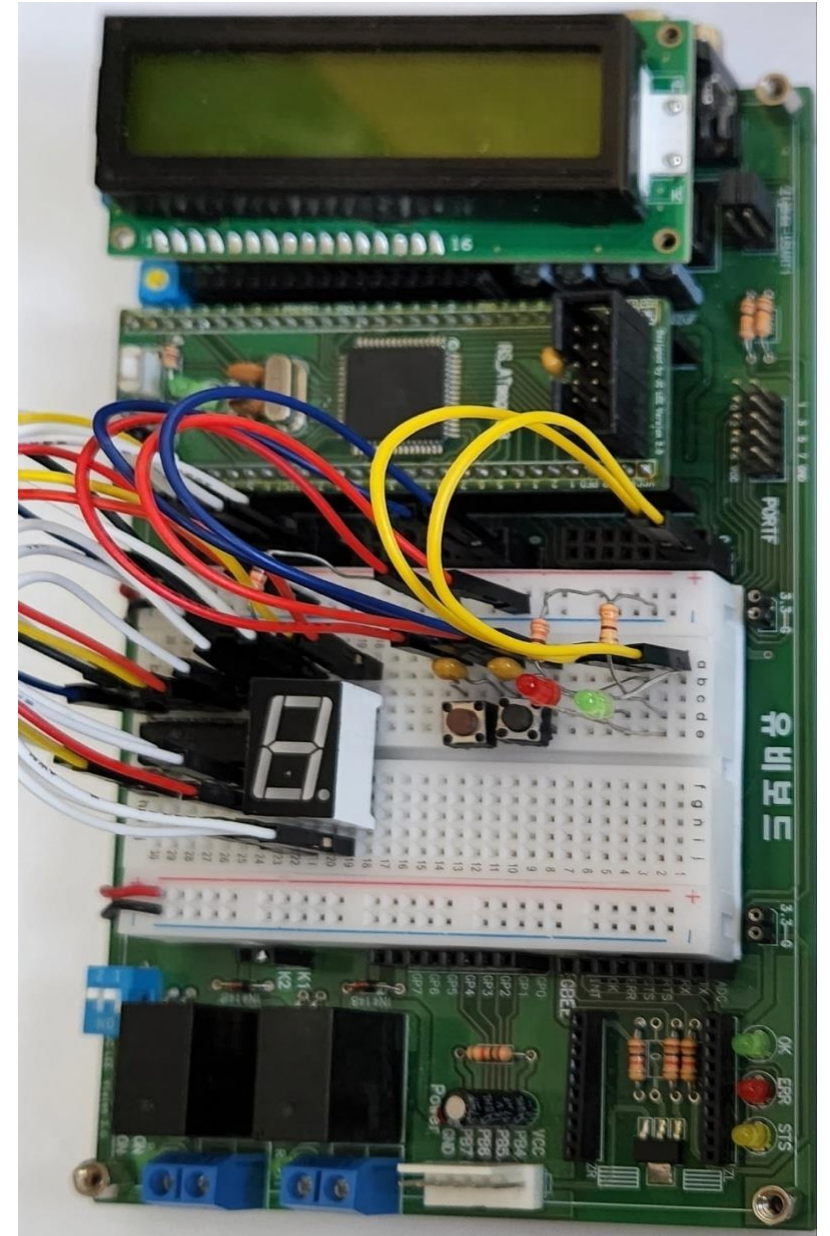
사용부품

- ATmega128(1개)
- 유비보드(1개)
- LCD(1개)
- FND(1개)
- LED(2개)
- LS7447(1개)
- 스위치(2개)
- 커패시터(2개)
- 330Ω 저항(3개)



프로그램 소개

1. LCD에 영어 단어를 출력.
2. 출력 된 영단어의 동의어를 찾아서 1번 스위치, 2번 스위치 중 선택하여 누름.
3. 동의어를 맞추면 푸른색 LED를, 틀리면 붉은색 LED를 불이 들어오게 하고 FND에 맞은 개수를 출력.
4. LCD에 5개 다 맞으면 Genius를, 4개 맞으면 Excellent를, 3개 맞으면 Good을 출력.
5. 영단어는 총 5개를 출력하고 5개중 3개 이상 맞추어야 종료.
6. 3개 이상 맞지 않을 경우 처음으로 돌아 간 후 반복.

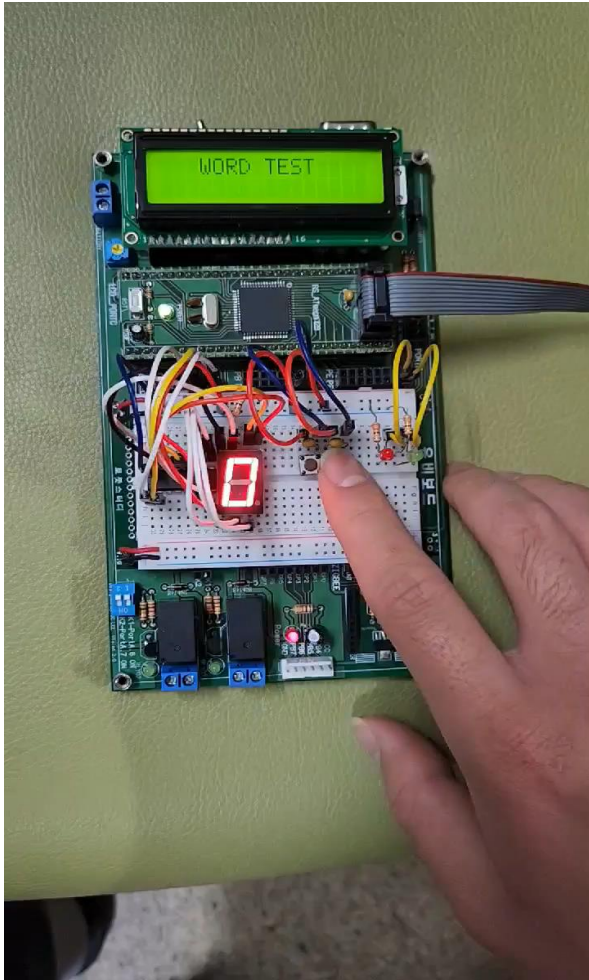


Part 3

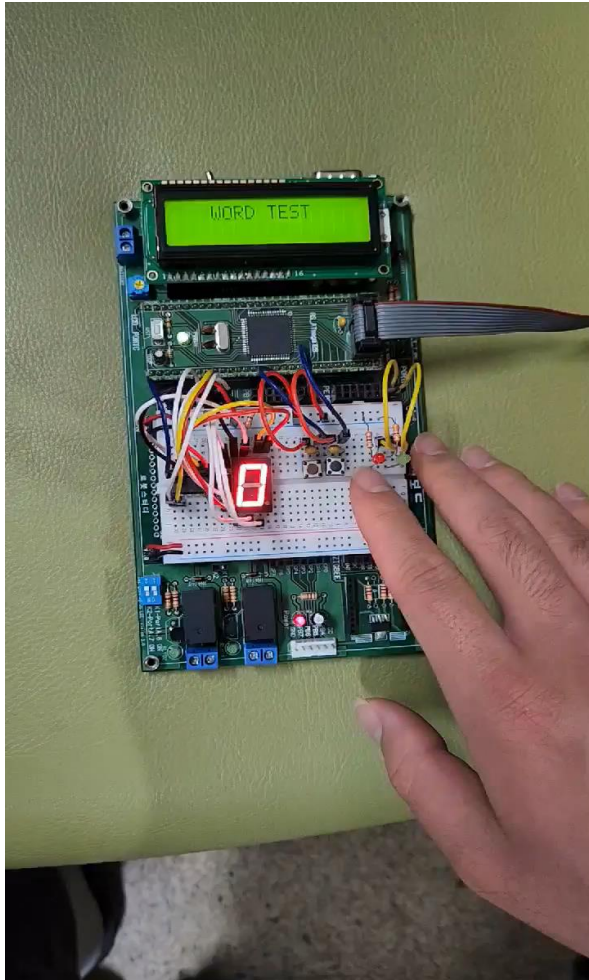
구동 영상



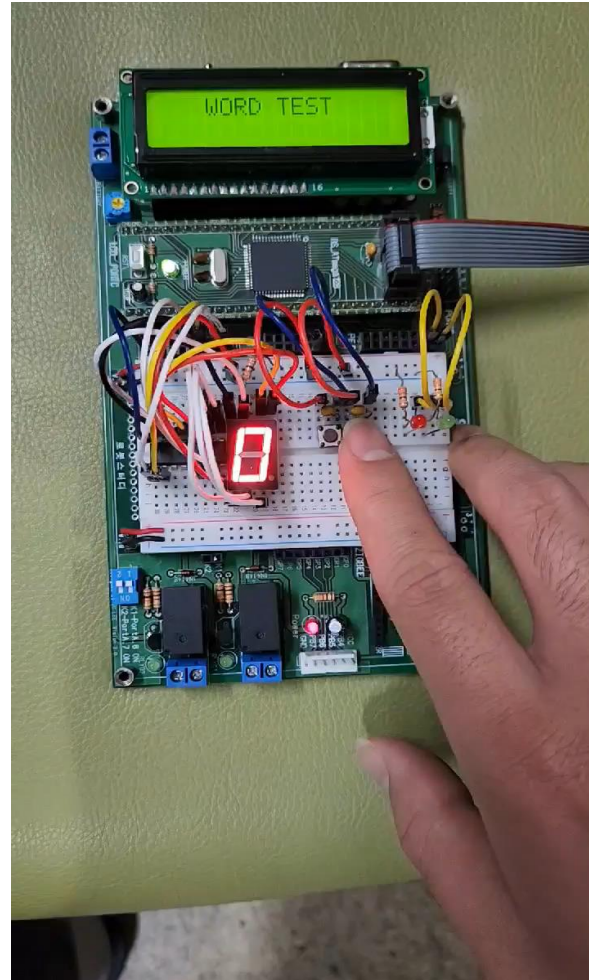
구동 영상



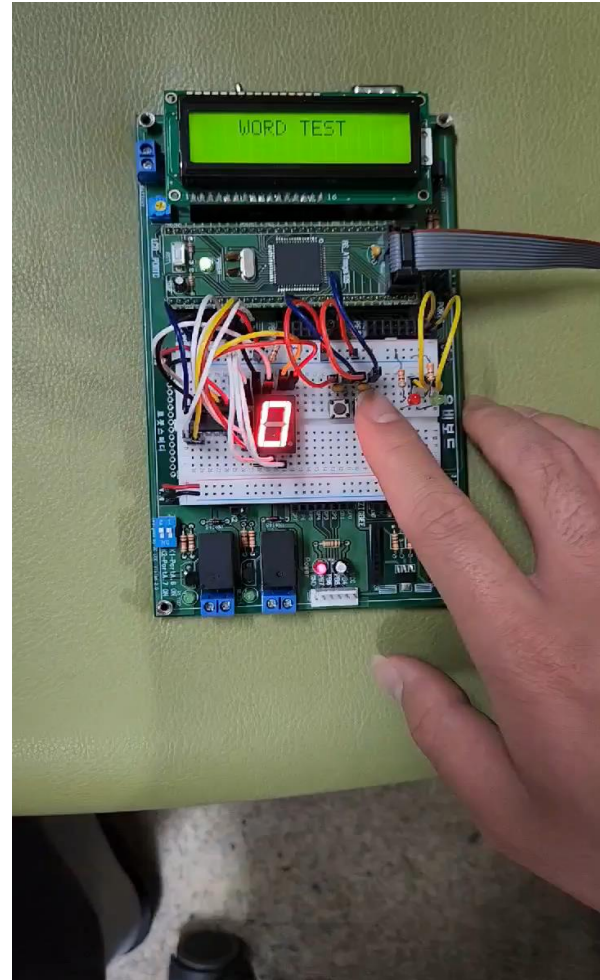
5개 다 맞았을 때



4개 맞았을 때



3개 맞았을 때



3개 미만 맞았을 때

Part 4

코드



코드

```
1  #include <megal28.h>
2  #include <delay.h>
3  #include <alcd.h>
4
5  unsigned char i=0, sw, k;
6  //1번 정답 : 1,   2번 정답 : 2,   3번 정답 : 1,   4번 정답 : 2,   5번 정답 : 1
7  //문제 1번을 Array에 설정
8  char num1[]="Talk";
9  char num1_1[]="1.Conversation";
10 char num1_2[]="2.Conservation";
11 //문제 2번을 Array에 설정
12 char num2[]="Influence";
13 char num2_1[]="1.Effect";
14 char num2_2[]="2.Affect";
15 //문제 3번을 Array에 설정
16 char num3[]="Counsel";
17 char num3_1[]="1.Advise";
18 char num3_2[]="2.Advice";
19 //문제 4번을 Array에 설정
20 char num4[]="Develop";
21 char num4_1[]="1.Revolution";
22 char num4_2[]="2.Evolution";
23 //문제 5번을 Array에 설정
24 char num5[]="Overseas";
25 char num5_1[]="1.Aboard";
26 char num5_2[]="2.Aboard";
27 //등급 산정을 위한 Array
28 char grade1[]="Genius!";
29 char grade2[]="Excellent!";
30 char grade3[]="Good!";
31
32 void main(void)
33 {
34     PORTA=0xFF;    //PORTF=LED, PORTA=switch, PORTD=FND
35     DDRA=0x00;    //사용 할 포트 초기화 및 입출력 포트 지정
36     PORTF=0x00;    //LCD는 유비보드 특성상 c로 통일이므로 생략
37     DDRF=0xFF;
38     PORTD=0x00;
39     DDRD=0xFF;
40
41     lcd_init(16);
```

• 지역변수 선언

• 동의어 테스트에 사용될 문자열 어레이 선언

• PORTA 초기화(switch)

• PORTF 초기화(LED)

• PORTD 초기화(FND)

코드

```
43 while (i<3) //문제 5개중 맞은 개수가 3개 미만일때 반복
44 {
45     i=0; //맞은 개수가 3개 미만 일 때를 위한 i값 초기화
46     PORTD=0x00; // 맞은 개수가 3개 미만 일 때를 위한 FND값 초기화
47     lcd_gotoxy(3,0);
48     lcd_putsf("WORD TEST"); //프로그램 제목 출력
49     delay_ms(3000);
50     lcd_clear(); //제목을 지우기 위한 클리어
51
52     //no.1 : 1
53     lcd_gotoxy(6,0);
54     lcd_putsf("No.1");
55     lcd_gotoxy(6,1);
56     lcd_puts(num1);
57     delay_ms(3000);
58     lcd_clear();
59     while(1)
60     {
61         sw=~PINA&0x0F;
62         lcd_gotoxy(0,0);
63         lcd_puts(num1_1);
64         lcd_gotoxy(0,1);
65         lcd_puts(num1_2);
66         delay_ms(500);
67         lcd_clear();
68
69         if(sw==0x01){
70             PORTF=0x01;
71             delay_ms(500);
72             i++;
73             PORTD=i;
74             delay_ms(500);
75             lcd_clear();
76             PORTF=0x00;
77             break;
78         }
79         else if(sw==0x02){
80             PORTF=0x02;
81             delay_ms(500);
82             lcd_clear();
83             PORTF=0x00;
84             break;
85         }
86     }
```

- 맞은 개수가 3개 미만일 때 반복 하도록 조건 부여
- FND값과 맞은 수를 나타내는 변수 i값 초기화
- LCD에 WORD TEST출력
- 문제 1번 문자열 출력
- 입력 데이터 반전 후 하위 4비트만 유효하게 하여 전달
- 정답인 스위치를 눌렀을 때 푸른 LED 점등
- FND에 맞은 개수 표시
- 프로그램을 다시 돌리기 위한 변수 i 증가
- 오답 스위치를 눌렀을 때 붉은 LED 점등

코드

```
88 //no.2 : 2
89 lcd_gotoxy(6,0);
90 lcd_putsf("No.2");
91 lcd_gotoxy(3,1);
92 lcd_puts(num2);
93 delay_ms(3000);
94 lcd_clear();
95 while(1)
96 {
97     sw=~PINA&0x0F;
98     lcd_gotoxy(0,0);
99     lcd_puts(num2_1);
100     lcd_gotoxy(0,1);
101     lcd_puts(num2_2);
102     delay_ms(500);
103     lcd_clear();
104
105     if(sw==0x02){
106         PORTF=0x01;
107         delay_ms(500);
108         i++;
109         PORTD=i;
110         delay_ms(500);
111         lcd_clear();
112         PORTF=0x00;
113         break;
114     }
115     else if(sw==0x01){
116         PORTF=0x02;
117         delay_ms(500);
118         lcd_clear();
119         PORTF=0x00;
120         break;
121     }
122 }
```

- 문제 2번 문자열 출력
- 입력 데이터 반전 후 하위 4비트만 유효하게 하여 전달
- 정답인 스위치를 눌렀을 때 푸른 LED 점등
- FND에 맞은 개수 표시
- 프로그램을 다시 돌리기 위한 변수 i 증가
- 오답 스위치를 눌렀을 때 붉은 LED 점등

코드

```
124 //no.3 : 1
125 lcd_gotoxy(6,0);
126 lcd_putsf("No.3");
127 lcd_gotoxy(5,1);
128 lcd_puts(num3);
129 delay_ms(3000);
130 lcd_clear();
131 while(1)
132 {
133     sw=~PINA&0x0F;
134     lcd_gotoxy(0,0);
135     lcd_puts(num3_1);
136     lcd_gotoxy(0,1);
137     lcd_puts(num3_2);
138     delay_ms(500);
139     lcd_clear();
140
141     if(sw==0x01){
142         PORTF=0x01;
143         delay_ms(500);
144         i++;
145         PORTD=i;
146         delay_ms(500);
147         lcd_clear();
148         PORTF=0x00;
149         break;
150     }
151     else if(sw==0x02){
152         PORTF=0x02;
153         delay_ms(500);
154         lcd_clear();
155         PORTF=0x00;
156         break;
157     }
158 }
```

- 문제 3번 문자열 출력
- 입력 데이터 반전 후 하위 4비트만 유효하게 하여 전달
- 정답인 스위치를 눌렀을 때 푸른 LED 점등
- FND에 맞은 개수 표시
- 프로그램을 다시 돌리기 위한 변수 i 증가
- 오답 스위치를 눌렀을 때 붉은 LED 점등

코드

```
160 //no.4 : 2
161 lcd_gotoxy(6,0);
162 lcd_putsf("No.4");
163 lcd_gotoxy(5,1);
164 lcd_puts(num4);
165 delay_ms(3000);
166 lcd_clear();
167 while(1)
168 {
169 sw=~PINA&0x0F;
170 lcd_gotoxy(0,0);
171 lcd_puts(num4_1);
172 lcd_gotoxy(0,1);
173 lcd_puts(num4_2);
174 delay_ms(500);
175 lcd_clear();
176
177 if(sw==0x02){
178     PORTF=0x01;
179     delay_ms(500);
180     i++;
181     PORTD=i;
182     delay_ms(500);
183     lcd_clear();
184     PORTF=0x00;
185     break;
186 }
187 else if(sw==0x01){
188     PORTF=0x02;
189     delay_ms(500);
190     lcd_clear();
191     PORTF=0x00;
192     break;
193 }
194 }
```

- 문제 4번 문자열 출력
- 입력 데이터 반전 후 하위 4비트만 유효하게 하여 전달
- 정답인 스위치를 눌렀을 때 푸른 LED 점등
- FND에 맞은 개수 표시
- 프로그램을 다시 돌리기 위한 변수 i 증가
- 오답 스위치를 눌렀을 때 붉은 LED 점등

코드

```
196 //no.5 : 1
197 lcd_gotoxy(6,0);
198 lcd_putsf("No.5");
199 lcd_gotoxy(4,1);
200 lcd_puts(num5);
201 delay_ms(3000);
202 lcd_clear();
203 while(1)
204 {
205     sw=~PINA&0x0F;
206     lcd_gotoxy(0,0);
207     lcd_puts(num5_1);
208     lcd_gotoxy(0,1);
209     lcd_puts(num5_2);
210     delay_ms(500);
211     lcd_clear();
212
213     if(sw==0x01){
214         PORTF=0x01;
215         delay_ms(500);
216         i++;
217         PORTD=i;
218         delay_ms(500);
219         lcd_clear();
220         PORTF=0x00;
221         break;
222     }
223     else if(sw==0x02){
224         PORTF=0x02;
225         delay_ms(500);
226         lcd_clear();
227         PORTF=0x00;
228         break;
229     }
230 }
```

- 문제 5번 문자열 출력

- 입력 데이터 반전 후 하위 4비트만 유효하게 하여 전달

- 정답인 스위치를 눌렀을 때 푸른 LED 점등

- FND에 맞은 개수 표시

- 프로그램을 다시 돌리기 위한 변수 i 증가

- 오답 스위치를 눌렀을 때 붉은 LED 점등

코드

```
232 if(i==5)
233 {
234     lcd_puts(grade1);
235     for(k=0;k<9;k++){
236         _lcd_write_data(0x1c);
237         delay_ms(500);
238     }
239     for(k=0;k<9;k++){
240         _lcd_write_data(0x18);
241         delay_ms(500);
242     }
243     lcd_clear();
244     lcd_gotoxy(4,0);
245     lcd_puts(grade1);
246     delay_ms(2000);
247     lcd_clear();
248 }
249 else if(i==4)
250 {
251     lcd_puts(grade2);
252     for(k=0;k<6;k++){
253         _lcd_write_data(0x1c);
254         delay_ms(500);
255     }
256     for(k=0;k<6;k++){
257         _lcd_write_data(0x18);
258         delay_ms(500);
259     }
260     lcd_clear();
261     lcd_gotoxy(3,0);
262     lcd_puts(grade2);
263     delay_ms(2000);
264     lcd_clear();
265 }
266 else if(i==3)
267 {
268     lcd_puts(grade3);
269     for(k=0;k<11;k++){
270         _lcd_write_data(0x1c);
271         delay_ms(500);
272     }
273     for(k=0;k<11;k++){
274         _lcd_write_data(0x18);
275         delay_ms(500);
276     }
277     lcd_clear();
278     lcd_gotoxy(6,0);
279     lcd_puts(grade3);
280     delay_ms(2000);
281     lcd_clear();
282 }
```

- 5개 다 맞았을 때 Genius! 문자열 출력
 - 출력된 문자열이 좌우 끝에서 끝으로 각각 1번씩 이동
 - 문자열 가운데로 정렬
-
- 4개 맞았을 때 Genius! 문자열 출력
 - 출력된 문자열이 좌우 끝에서 끝으로 각각 1번씩 이동
 - 문자열 가운데로 정렬
-
- 3개 맞았을 때 Genius! 문자열 출력
 - 출력된 문자열이 끝에서 끝으로 각각 1번씩 이동
 - 문자열 가운데로 정렬


```

283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302

```

```

else if(i<3)
{
    lcd_gotoxy(6,0);
    lcd_putsf("Again");
    delay_ms(1000);
    lcd_clear();
    delay_ms(500);
    lcd_gotoxy(6,0);
    lcd_putsf("Again");
    delay_ms(1000);
    lcd_clear();
    delay_ms(500);
    lcd_gotoxy(6,0);
    lcd_putsf("Again");
    delay_ms(1000);
    lcd_clear();
    delay_ms(500);
}
}

```

- 3개 미만으로 맞았을 때 Again 3번 점멸
- 변수 i가 3미만이기 때문에 while문 처음부터 다시 실행

아쉬웠던 점

- 프로젝트 이름처럼 단어의 뜻을 한글로 출력하고 영단어를 맞추고 싶었지만 한글을 출력 하는 것을 아직 안 배운 상태이기 때문에 아쉬웠음.
- 문제를 랜덤으로 나오게 하고 싶었지만 랜덤 관련하여 아직 모르는 상태여서 구현 못한 것이 아쉬웠음.

추가로 업데이트 하고 싶은 것

- 한글을 출력 하는 것을 배우고 랜덤으로 출력할 수 있게 되면 기존 원하던 방식으로 동의어 문제나 단어의 뜻을 맞추는 것을 랜덤으로 출력.
- 문제를 맞추게 되면 LED 점등과 함께 부저가 울리게 함.

QnA



감사합니다.

hello

