

예제 1)

FND 모듈을 이용하여 0~9의 숫자를 모듈의 원하는 위치에 출력하는 예제입니다. FND 모듈을 RaspberryPi Adapter에 연결해 주었고 FND_SEL 포트의 S0~S5까지 위치에 맞는 GPIO핀과 숫자를 표현하는 8가지 FND_DB에 맞는 GPIO핀을 배열로 선언하였습니다. Init 함수를 만들어서 각 핀의 출력을 셋팅 하였습니다. 재밌게도 Fnd_SEL을 출력할 때는 LED를 켤 때와는 다르게 LOW상태에서 해당 위치가 출력되는 것을 확인할 수 있었습니다. FND_DB 포트의 8가지 핀은 HIGH상태에서 출력됩니다.

FndDisplay함수는 position과 number를 parameter로 받습니다. 즉 S0~S5의 원하는 포지션에 0~9까지 원하는 숫자를 출력하는 함수입니다. 변수 flag의 상태에 따라 FND_DB 핀이 각각 출력이 될지 안 될지가 결정됩니다. shift 변수는 0b00000001부터 0b10000000 까지 비트를 옮겨가며 8개의 비트를 탐색합니다. 가령 0을 0b00111111로 표현할 수 있다면 shift가 1번째부터 8번째 비트까지 AND연산을 통해서 1인 bit에 해당하는 핀을 키고 0인 bit에 해당하는 핀을 끄는 식으로 총 6개의 핀을 키게 합니다.

FND_DB에 어느 핀을 출력할지 셋팅을 한 후에는 원하는 포지션에 다시금 출력을 할 수 있게 합니다.

main문에서 argc는 main함수에 전달된 인자의 개수, argv는 가변적인 개수의 문자열입니다. argv는 문자열이므로 atoi를 써서 문자열을 정수로 변환하였습니다.

컴파일 후 인수를 받아서 실행한 결과 원하는 위치에 숫자를 출력하는 것을 확인할 수 있었습니다.

lab2-4_1.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <wiringPi.h>
4
5  const int FndSelectPin[6]={4,17,18,27,22,23};
6  const int FndPin[8]={6,12,13,16,19,20,26,21};
7  const int FndFont[10]={0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07,0x7F, 0x67};
8
9  void Init() {
10     int i;
11     if(wiringPiSetupGpio() == -1) {
12         printf("wiringPiSetupGpio()error\n");
13         exit(-1);
14     }
15
16     for(i=0; i<6; i++) {
17         pinMode(FndSelectPin[i], OUTPUT);
18         digitalWrite(FndSelectPin[i], HIGH);
19     }
20
21     for(i=0; i<8; i++) {
22         pinMode(FndPin[i], OUTPUT);
23         digitalWrite(FndPin[i], LOW);
24     }
25 }
26
27 void FndDisplay(int position, int num) {
28     int i,j;
29     int flag = 0;
30     int shift = 0x01;
31
32     for(i=0; i<8; i++) {
33         flag=(FndFont[num] & shift);
34         digitalWrite(FndPin[i], flag);
35         shift<<=1;
36     }
37     digitalWrite(FndSelectPin[position], LOW);
38 }
39
40 int main(int argc, char **argv) {
41     if(argc!=3) {
42         printf("Usage: %s [position][number]", argv[0]);
43         exit(-1);
44     }
45
46     Init();
47
48     FndDisplay(atoi(argv[1]), atoi(argv[2]));
49     return 0;
50 }
51
52
```

<lab2-4_1.c 코드>

예제2)

0부터 5 까지의 숫자를 화면에 동시에 출력하는 예제입니다. FND_SEL 포트의 FND_DB는 1번 예제와 동일하게 설정해주었고 0부터9까지 8비트로 출력할 수 있는 font를 설정하였습니다.

FndSelect함수는 선택된 FND Select핀만 on시키고 나머지 핀은 모두 off시키는 함수입니다. LED를 켤 때와는 반대로 LOW byte일 때 켜지고 HIGH 일 때 꺼집니다. 그 후 위와 동일한 FndDisplay함수를 사용하여 원하는 포지션과 숫자를 출력할 수 있게 하였습니다.

FndSelect함수에서 선택한 핀만을 켜고 나머지는 꺼지므로 6개의 숫자를 동시에 출력할 수 없다고 생각할 수 있지만 while문에서 delay를 1m초씩 주었습니다. 즉, 0핀에 0, 1핀에 1 2핀에 2...5핀에 5를 1m씩 돌아가면서 출력하게 되는데 이렇게 된다면 사람의 눈에는 거의 동시에 불이 켜지는 것으로 보입니다.

컴파일 후 실행결과 0부터 5까지 각 핀에 숫자가 알맞게 출력되는 것을 확인할 수 있었습니다.

```
void FndSelect(int position) {
    int i;
    for(i=0; i<6; i++) {
        if(i==position)
            digitalWrite(FndSelectPin[i], LOW);
        else
            digitalWrite(FndSelectPin[i], HIGH);
    }
}

void FndDisplay(int position, int num) {
    int i,j;
    int flag = 0;
    int shift = 0x01;

    for(i=0; i<8; i++) {
        flag=(FndFont[num] & shift);
        digitalWrite(FndPin[i], flag);
        shift<<=1;
    }

    FndSelect(position);
}

int main() {
    int pos;
    int data[6]={0,1,2,3,4,5};

    Init();

    while(1) {
        for(pos=0; pos<6; pos++) {
            FndDisplay(pos, data[pos]);
            delay(1);
        }
    }

    return 0;
}
```

<lab2-4_2.c의 FndSelect와 main문>

과제 1)

Hello 단어를 왼쪽으로 0.5초간 Select 핀을 옮기면서 출력하는 실습입니다. 먼저 숫자가 아니라 각 알파벳을 알맞게 8비트로 출력하기 위해 H, E, L, O 가 어떻게 표현될지 생각하였습니다.

H는 01110110, E는 01111001, L은 00111000, O는 00111111로 표현할 수 있었고 이를 16진수로 표현하면 각각 0x76, 0x79, 0x38, 0x3F입니다.

0번 핀에 O, 1번 핀에 L, 2번 핀에 L, 3번 핀에 E, 4번 핀에 H가 들어가게 되므로 각 배열의 위치에 맞게 FndFont를 수정하였습니다.

그 후 FndSelect, FndDisplay함수는 위와 동일하게 작성하였습니다.

Hello를 0.5초 즉, 500msec마다 왼쪽으로 옮겨줘야 했습니다. 이 말은 for문을 돌면서 FndDisplay 함수에 앞의 숫자는 포지션, 뒤의 숫자는 폰트입니다.

00 11 22 33 44

10 21 32 43 54

20 31 42 53 04

30 41 52 03 14처럼 숫자가 반복되면서 들어가야 한다는 뜻입니다. 즉 포지션을 6번

바뀌어야 하기 때문에 먼저 0부터 6까지 돌게 되는 반복문을 만들었습니다. 또한 뒤의

number는 항상 갖고 앞의 position만 달라지기 때문에 pos 변수를 하나 만들었습니다.

pos는 0부터 4까지 반복문을 돌게 되는데, FndDisplay의 첫 번째 parameter인 position에는 pos+i를 6으로 나눈 나머지 값이 들어가게 됩니다. 이렇게 함수를 구현함으로써 select핀을

옮겨가며 Fnd를 제어 할 수 있었습니다. 또한 위 예제처럼 모든 핀이 출력한 것처럼

보이려면 1ms씩 빠르게 옮겨가며 출력을 해줘야 합니다. 5개의 핀이 1m씩 커지면 총

5m만큼 커집니다. 따라서 delaysum이라는 변수에다 5씩 더해줌으로써 delaysum이 500이

되면 0.5초가 지난것이므로 이때마다 i를 1씩 더해주어 0.5초의 시간을 계산하였습니다.

컴파일 후 실행결과 0.5초마다 핀을 옮겨가며 HELLO 단어를 출력하는 것을 확인할 수 있었습니다.

```
int main() {
    Init();

    int i, pos, delaysum;
    while(1) {
        for(i=0; i<6; i++) {
            delaysum=0;
            while(delaysum<500) {
                for(pos=0; pos<5; pos++) {
                    FndDisplay((pos+i)%6,pos);
                    delay(1);
                }
                delaysum+=5;
            }
        }

        return 0;
    }
}
```

<ass2-4_1.c의 의 main문>

과제 2)

1/100 초 타이머를 만드는 실습입니다. 0~9의 숫자를 시간에 맞춰서 출력하면 되기 때문에 예제 1~2번과 동일하게 FndFont를 설정해 주었습니다. 추가로 0, 1번은 각각 1/100초 1/10초이고 2번 select pin부터 1초입니다. 예를 들어 35.78초 같은 경우는 2번 핀은 5.처럼 point가 있기 때문에 A~H의 FndPin중에서 H값 즉, 8번째 비트 값은 항상 1로 해줘야합니다. 따라서 원래의 FndFont배열에 추가로 0b10000000 (0x80)을 ADD 하여 FndFontpoint를 따로 만들어주었습니다.

FndSelect, FndDisplay 함수는 예제와 같이 설정해주었습니다. 또한 다수의 select 핀을 연속적으로 출력하기 때문에 무조건 delay(1)을 해줘야합니다. 따라서 main문에서 여러 번 선언하지 않게끔 FndDisplay함수에 delay(1)을 더해주었습니다.

아두이노처럼 millis()를 통해 프로그램이 시작되고의 시간을 전달받을 수 있습니다. 전달받을 시간, 그리고 0~5번 select pin까지 출력될 숫자를 int로 선언해 주었습니다. 또한 1/100초 타이머를 만들 것이기 때문에 1/1000초까지는 계산을 할 필요가 없습니다. 따라서 입력받는 시간을 10만큼 나누어서 1/100초부터 제어할 수 있게 time = millis()/10 으로 설정하였습니다. 그 후 d0~d5의 1/100의 자리부터 1000의자리 까지 몫과 나머지 연산을 통해 설정하였습니다.

그 다음에는 times를 입력받는 시간에 따라 select pin의 출력될 때를 정해주었습니다. 예를들어 millis가 10보다 커지면 0.01초가 되므로 0번 pin은 항상 켜지게 되고 times가 10보다 커지면 (millis가 100보다 커지면) 0.1초가 넘어가므로 항상 켜지게 되는 식입니다. 이런식으로 1000의 자리까지 설정을 해주었습니다. 다만 2번 핀은 소수점을 표현하는 점이 들어가기 때문에 이때는 FndDisplay 함수에 if문을 설정해주었습니다. position이 2일 때는 FndFontpoint배열의 값을 접근할 수 있게 고쳐주었습니다.

컴파일 후 실행결과 정확한 속도로 1/100 타이머가 작동하는 것을 확인할 수 있었습니다.

```

int main() {
    Init();
    int times, d0,d1,d2,d3,d4,d5, i;
    for(i=0; i<10; i++)
        FndFontpoint[i]=FndFont[i]+0x80;
    while(1) {
        times=millis()/10;

        d0 = times % 10;
        d1 = (times / 10) % 10;
        d2 = (times / 100) % 10;
        d3 = (times / 1000) % 10;
        d4 = (times / 10000) % 10;
        d5 = (times / 100000) % 10;

        FndDisplay(0,d0);
        if(times >=10) FndDisplay(1,d1);
        if(times >=100) FndDisplay(2,d2);
        if(times >=1000) FndDisplay(3,d3);
        if(times >=10000) FndDisplay(4,d4);
        if(times >=100000) FndDisplay(5,d5);

    }

    return 0;
}

```

<ass2-4_2.c 의 main문>

각 예제 및 과제의 코드와 실행결과 영상은 폴더에 첨부하였습니다.