

예제 1)

키패드를 이용하여 키패드 버튼과 일치하는 Led 번호를 점등하는 예제입니다. Led모듈과 키패드 모듈을 각각 RaspberryPi Adapter에 연결해 주었습니다. 코드에서의 LedRed 및 Keypad 배열에는 각 버튼마다 대응되는 핀 번호를 넣어주었습니다.

KeypadRead 함수에서는 눌린 키패드의 번호를 받습니다. 눌린 상태에서 digitalRead의 반환 값은 0이므로 계속 눌리고 있는 상태라면 keypadnum은 계속 그 상태를 유지합니다.

LedControl 함수에서는 keypad번호와 일치하는 번호의 Led를 ON시키고 그 외의 번호는 모두 OFF됩니다. 따라서 하나의 Led만이 켜지게 됩니다.

이후 main문에서는 Led모듈과 연결된 핀번호를 모두 출력으로 하고 키패드모듈과 연결된 핀번호는 모두 입력으로 설정하였습니다. 이후 while문에서 수시로 keypadread 함수를 통해 입력값을 받고 그 입력값에 따라 Led를 점등하였습니다.

컴파일 후 실행 결과 키패드 번호가 눌릴 때 그에 대응되는 LED가 켜지는 것을 확인할 수 있었습니다.

```

lab2-3_1.c
1  #include <wiringPi.h>
2  const int LedRed[8]={4,17,18,27,22,23,24,25};
3  const int Keypad[8]={16,13,12,6,21,26,20,19};
4
5  int KeypadRead() {
6      int i, keypadnum = -1;
7      for(i=0; i<8; i++) {
8          if(!digitalRead(Keypad[i])) {
9              keypadnum = i;
10             break;
11         }
12     }
13     return keypadnum;
14 }
15
16 void LedControl(int keypadnum) {
17     int i;
18     for(i=0; i<8; i++) {
19         if(i==keypadnum)
20             digitalWrite(LedRed[i], HIGH);
21         else
22             digitalWrite(LedRed[i], LOW);
23     }
24 }
25
26 int main(void) {
27     int i, keypadnum = -1;
28     if(wiringPiSetupGpio() == -1) return 1;
29
30     for(i=0; i<8; i++) {
31         pinMode(LedRed[i], OUTPUT);
32         digitalWrite(LedRed[i], LOW);
33         pinMode(Keypad[i], INPUT);
34     }
35
36     while(1) {
37         keypadnum=KeypadRead();
38         LedControl(keypadnum);
39     }
40
41     return 0;
42 }

```

<lab2-3_1.c 코드>

예제 2)

1번 키패드를 누르면 1~8번 까지의 LED를 오름차순으로 8번 키패드를 누르면 내림차순으로 점등하는 예제입니다. KeypadRead와 LedControl 함수의 기능은 예제 1번과 같습니다.

이후 main문에서 출력핀과 입력핀을 각각 셋팅하였습니다. while문 안에서는 수시로 KeypadRead함수를 통해 keypadnum값을 입력받고 입력값이 1번 키패드의 GPIO면 for문을 돌면서 오름차순 점등을 하고 입력값이 8번 키패드의 GPIO면 for문을 돌면서 내림차순 점등을 합니다. 컴파일 후 실행 결과 눌리는 키패드 번호에 따라 점등이 다르게 되는 것을 확인할 수 있었습니다.

```
int main(void) {
    int i, keypadnum = -1;
    if(wiringPiSetupGpio()==-1) return 1;

    for(i=0; i<8; i++) {
        pinMode(LedRed[i], OUTPUT);
        digitalWrite(LedRed[i], LOW);
    }

    for(i=0; i<2; i++)
        pinMode(Keypad[i], INPUT);

    while(1){
        keypadnum=KeypadRead();
        if(keypadnum==0) {
            for(i=0; i<8; i++) {
                LedControl(i);
                delay(500);
            }

            digitalWrite(LedRed[i-1], LOW);
        }

        else if(keypadnum==1) {
            for(i=7; i>=0; i--) {
                LedControl(i);
                delay(500);
            }
            digitalWrite(LedRed[i+1], LOW);
        }
    }
    return 0;
}
```

<lab2-3_2.c 의 main문 코드>

과제 1)

keypad번호에 따라 해당 Led번호를 켜다 키는 예제입니다. ON -> OFF, OFF -> ON 이 되어야 합니다. 인터럽트를 이용하여 과제를 구현해보고 싶어서 wiringPiISR 함수를 사용하였습니다. wiringPiISR 함수는 3가지 parameter가 필요합니다. 첫 번째는 Input Gpio, 두 번째는 Falling, Rising 등 의 Edge 조건, 세 번째는 인터럽트 발생 시 구현되는 함수입니다. -1을 return 할 시 함수가 제대로 작동한 것이 아니게 됩니다.

ISR을 사용하기 위해 각 핀번호에 대응하는 8가지 함수를 만들었습니다. Button1~8의 함수는 전역변수로 Output을 선언한 후 해당 output을 출력합니다. 출력 후에는 %2 연산을 통해 0은 1, 1은 0 값으로 계속 바뀌게 됩니다.

main문에서 출력핀, 입력핀 셋팅을 한 후 8가지 wiringPiISR 함수를 선언하였습니다.

또한 While문은 비워두었습니다.

컴파일 후 실행 결과 놀리는 키패드 번호에 따라 인터럽트가 걸려서 해당 인터럽트의 함수를 수행하는 것을 확인할 수 있었습니다.

```
ass2-3_1.c
1  #include <wiringPi.h>
2  const int LedRed[8]={4,17,18,27,22,23,24,25};
3  const int Keypad[8]={16,13,12,6,21,26,20,19};
4
5  void Button1() {
6      static int output1= HIGH;
7      digitalWrite(LedRed[0], output1);
8      output1 = (output1+1)%2;
9  }
10
11 void Button2() {
12     static int output2= HIGH;
13     digitalWrite(LedRed[1], output2);
14     output2 = (output2+1)%2;
15 }
16
17 void Button3() {
18     static int output3= HIGH;
19     digitalWrite(LedRed[2], output3);
20     output3 = (output3+1)%2;
21 }
22
23 void Button4() {
24     static int output4= HIGH;
25     digitalWrite(LedRed[3], output4);
26     output4 = (output4+1)%2;
27 }
28 void Button5() {
29     static int output5= HIGH;
30     digitalWrite(LedRed[4], output5);
31     output5 = (output5+1)%2;
32 }
33 void Button6() {
```

<Button : 키패드에 대응하는 led 점등>

```

int main(void) {
    int i, keypadnum = -1;
    if(wiringPiSetupGpio()==-1) return 1;

    for(i=0; i<8; i++) {
        pinMode(LedRed[i], OUTPUT);
        digitalWrite(LedRed[i], LOW);
        pinMode(Keypad[i], INPUT);
    }

    if(wiringPiISR(Keypad[0], INT_EDGE_RISING, Button1)<0) return 1;
    if(wiringPiISR(Keypad[1], INT_EDGE_RISING, Button2)<0) return 1;
    if(wiringPiISR(Keypad[2], INT_EDGE_RISING, Button3)<0) return 1;
    if(wiringPiISR(Keypad[3], INT_EDGE_RISING, Button4)<0) return 1;
    if(wiringPiISR(Keypad[4], INT_EDGE_RISING, Button5)<0) return 1;
    if(wiringPiISR(Keypad[5], INT_EDGE_RISING, Button6)<0) return 1;
    if(wiringPiISR(Keypad[6], INT_EDGE_RISING, Button7)<0) return 1;
    if(wiringPiISR(Keypad[7], INT_EDGE_RISING, Button8)<0) return 1;

    while(1) {

    }

    return 0;
}

```

< main문에서 wiringPiISR함수를 사용하여 인터럽트 >

과제 2)

예제 2번에서 각 동작을 반복문으로 수정하고 추가로 키패드 하나를 더해 일시정지 하는 기능을 추가하는 과제입니다. 입력받은 키패드 번호를 의미하는 keypadnum과 켜지는 led 번호를 의미하는 lednum을 전역 변수로 선언하였습니다. KeypadRead 함수와 LedControl 함수는 예제 1, 2번과 동일합니다. 후에 main문에서 입력핀 및 출력핀을 셋팅하였습니다.

while문 안에서는 예제 2번을 수정하여 1~8, 8~1의 led를 도는 for문 밖에 while문을 추가하려 하였으나 이렇게 하면 각 반복문마다 keypad의 값을 읽고 읽은 값이 달라질 때마다 break를 해줘야하기 때문에 매우 번거로웠습니다. 따라서 while문 안에서 별도의 반복문 없이 오름차순, 내림차순으로 led를 점등시킬 방법을 고민하였습니다. 방법은 바로 불이 켜질 때 마다 전역변수로 선언된 lednum의 값을 1씩 더하거나 빼주는 것입니다. 그리고 3번 째 입력을 받을 땐 켜진 불은 켜있고 꺼진 불은 꺼있을 수 있도록 일시정지를 해야 하므로 LedControl 함수를 delay없이 선언하였습니다.

이렇게 코드를 짜자 while문에서 수시로 keypadnum의 값을 읽어 Led제어를 할 수 있었습니다.

컴파일 후 실행결과 각 3가지 keypad에 오름차순 점등, 내림차순 점등, 일시정지가 원활히 되는 것을 확인할 수 있었습니다.

ass2-3_2.c

```
1  #include <wiringPi.h>
2  const int LedRed[8] = {4,17,18,27,22,23,24,25};
3  const int Keypad[3] = {16,13,12};
4  int keypadnum=3;
5  int lednum = 0;
6
7  void KeypadRead() {
8      int i;
9      for (i=0; i<3; i++) {
10         if(!digitalRead(Keypad[i])) {
11             keypadnum=i;
12             break;
13         }
14     }
15 }
16
17 void LedControl(int num) {
18     int i;
19     for(i=0; i<8; i++) {
20         if(i==num)
21             digitalWrite(LedRed[i], HIGH);
22         else
23             digitalWrite(LedRed[i], LOW);
24     }
25 }
26
27
28
29
```

<전역변수로 선언된 keypadnum 과 lednum>

```
int main(void) {
    int i;
    if(wiringPiSetupGpio() == -1) return 1;
    for (i=0; i<8; i++) {
        pinMode(LedRed[i], OUTPUT);
        digitalWrite(LedRed[i], LOW);
    };

    for(i=0; i<3; i++)
        pinMode(Keypad[i], INPUT);

    while(1) {
        KeypadRead();
        if(keypadnum==0) {
            if(lednum==8) lednum=0;
            LedControl(lednum);
            delay(500);
            lednum++;
        }

        else if(keypadnum==1) {
            if(lednum==7) lednum=0;
            LedControl(lednum);
            delay(500);
            lednum--;
        }

        else if(keypadnum==2) {
            LedControl(lednum);
        }

    }

    return 0;
}
```

<while문 안에서 Led제어>

각 예제 및 과제의 실행결과 영상은 폴더에 첨부하였습니다.