

System Programming and Practice

-Lab10-

이름 : 김태석

학번 : 201721083

학과 : 미디어학과

```
int main(int argc, char *argv[]){
    int pressed = 0;
    int state = 0;
    if(-1==LED_GPIOExport(POUT))
        return(1);
    if(-1==LED_GPIODirection(POUT,OUT))
        return(2);
    if(-1==BUTTON_GPIOExport(POUT2) || -1==BUTTON_GPIOExport(PIN))
        return(1);
    if(-1==BUTTON_GPIODirection(POUT2,OUT) || -1==BUTTON_GPIODirection(PIN,IN))
        return(2);
    while(1){
        if(BUTTON_GPIORead(PIN) == 1 && state == 1 && pressed == 1){
            pressed = 0;
            state = 0;
            LED_GPIOWrite(POUT,0);
            printf("POUT : %d          |          LED OFF\n",POUT);
        }
        if(BUTTON_GPIORead(PIN) == 1 && state == 0 && pressed == 1){
            pressed = 0;
            state = 1;
            LED_GPIOWrite(POUT,1);
            printf("POUT : %d          |          LED ON\n",POUT);
        }
        if(BUTTON_GPIORead(PIN) == 0){
            pressed = 1;
            printf("POUT : %d          |          Button Clicked\n",POUT2);
        }
        usleep(100000);
    }

    if(-1==LED_GPIOUnexport(POUT))
        return(4);
    if(-1==BUTTON_GPIOUnexport(POUT2) || -1==BUTTON_GPIOUnexport(PIN))
        return(4);
    return(0);
}
```

코드 설명

GPIO 를 통한 LED 및 Button Export, Direction 과 Button Read, Write, LED Write 는 실습 예제의 코드를 사용하였다. 또한 라즈베리 파이의 회로를 실습 예제와 같이 그대로 사용하였기에 LED 의 POUT 값은 그대로 17 을 사용하였고 Button 의 PIN 값은 20, POUT 값은 21 을 그대로 사용하였다. Main 함수에서는 LED, Button 을 사용하기 위해 Export, Direction 함수를 사용 해 주었고, 연습 예제에서는 repeat 값을 따로 설정 해 두어 반복문이 일정 시간이 지나면 종료되었지만, 이번 Lab10 과제에서는 버튼 토글을 테스트하기 위해 반복을 무한대로 설정 해 주었다. 종료는 control + c 로 가능한데, 이를 통해 종료할 경우 제어권 반납이 원활히 일어나지 않는다는 문제가 있지만, 이는 signal() 함수를 통해 event catch 로 처리가 가능하다. (하지만 해당 코드에는 작성하지 않았다.)

버튼 토글을 이용한 LED ON/OFF 는 버튼을 누르게 되면 button read 값이 0, 떴을 때는 1 이라는 점을 이용하여 버튼을 눌렀다 떴을 때 pressed 값과 state 값을 변화시키는 논리를 사용하였다.

(state : LED 가 켜져 있으면 1, 꺼져 있으면 0 | pressed : 버튼이 눌리는 상황 1, 떴고 난 후 LED ON/OFF 시 0 으로 초기화)

실행 결과

```
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED OFF
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED ON
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED OFF
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED ON
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED OFF
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED ON
POUT : 20      |      Button Clicked
POUT : 20      |      Button Clicked
POUT : 17      |      LED OFF
```

결과 설명

Pressed,state 값은 0 으로 초기화 되어 있기에 버튼을 눌렀다 떼 려면 pressed=1 로 바뀌며 read =1,state=0,pressed=1 조건에 걸려 불이 켜지고 pressed 값은 초기화 되며 이후 state 값은 LED ON/OFF 에 따라 0,1로 변화한다.