



임베디드 시스템 설계 및 실험 3주차 보고서

002분반 3조

201724601 최성렬

201729163 이희근

202055522 김은지

202055574 이다은

202055590 장서윤

실험 목표

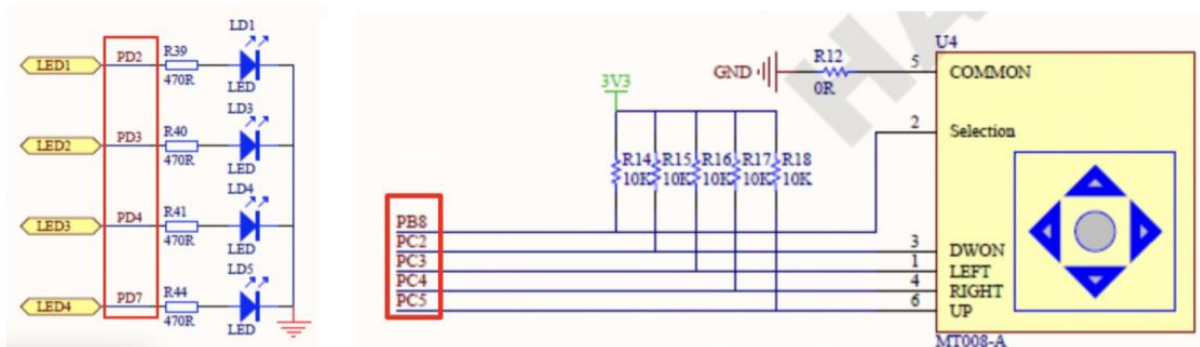
이번 실험의 목표는 조이스틱 up, down, right, left 제어를 통한 LED 점등 제어로, 홀수조이기 때

문에 다음 그림처럼 led를 제어해야 한다

- DOWN -> LED1, LED2 off
- LEFT -> LED3, LED4 off
- RIGHT -> LED3, LED4 on
- UP -> LED1, LED2 on

schematic을 보았을 때, led와 조이스틱은 GPIO의 Port C, D에 각각 연결되어 있다.

LED 1,2,3,4는 각각 Port D의 2,3,4,7 PIN에 연결되어 있고, 조이스틱의 down, left, right, up은 각각 Port C의 2,3,4,5 PIN에 연결되어 있다.



실험 과정

1. RCC를 이용해 사용할 조이스틱(Port C), LED(Port D)에 Clock 인가

RCC의 레퍼런스 페이지를 보면, 32bit중 4,5번째 bit에 1을 줘야만 port C,D에 clock이 인가된다.

데이터시트에서 RCC주소는 0x40021000으로, offset 0x18을 더해준 0x40021018에 OR 연산자로 0b110000(= 0x30)을 넣어준다.

7.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x18

Reset value: 0x0000 0000

Bit 4 **IOPCEN**: IO port C clock enable
Set and cleared by software.
0: IO port C clock disabled
1: IO port C clock enabled

Bit 5 **IOPDEN**: IO port D clock enable
Set and cleared by software.
0: IO port D clock disabled
1: IO port D clock enabled

```
*((volatile unsigned int *)0x40021018) |= 0x30; // clock enable port C, D
```

2. 사용할 Port PIN 초기화

PIN을 사용하기 위해서 초기화를 먼저 해주어야 한다. 레퍼런스를 보면 Port configuration register low(CRL), high(CRH) 2종류가 있는데, Port에서 0~7번째 pin은 CRL, 8~15번째 pin은 CRH를 써줘야 한다.

우리는 조이스틱의 2,3,4,5 PIN, LED의 2,3,4,7 PIN을 설정하기 때문에 CRL을 써준다. 여기서 offset은 0x00으로, 조이스틱(Port C)의 주소는 0x40011000이며 2,3,4,5번째 PIN만 0으로 초기화하기 위해 $\&= 0xFF0000FF$ 연산을 해준다.

LED(Port D)의 주소는 0x40011400이며 2,3,4,7번째 PIN만 0으로 초기화하기 위해 $\&= 0x0FF000FF$ 연산을 해준다

3. 사용할 Port PIN input/output 설정

조이스틱은 입력을 받아오므로 input mode, LED는 출력하므로 output mode로 설정해야 한다.

9.2.1 Port configuration register low (GPIOx_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

CRL 레퍼런스를 보면, 4bit가 하나의 PIN설정을 나타내는데 input with pull-up/pull down mode는 $MODE[1:0] = '00'$, $CFN[1:0] = '10'$ 이다. 즉 '0b1000'을 16진수로 변환하면 8이다. 우리 조이스틱의 PIN 2,3,4,5를 input mode로 설정해야 하므로 해당자리에만 0b1000(=0x8)을 넣어준다.

CNFy[1:0]: Port x configuration bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port. Refer to [Table 20: Port bit configuration table on page 161](#).

In input mode (MODE[1:0]=00):

00: Analog mode
01: Floating input (reset state)
10: Input with pull-up / pull-down
11: Reserved

In output mode (MODE[1:0] > 00):

00: General purpose output push-pull
01: General purpose output Open-drain
10: Alternate function output Push-pull
11: Alternate function output Open-drain

MODEy[1:0]: Port x mode bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port. Refer to [Table 20: Port bit configuration table on page 161](#).

00: Input mode (reset state)
01: Output mode, max speed 10 MHz.
10: Output mode, max speed 2 MHz.
11: Output mode, max speed 50 MHz.

마찬가지로 LED의 PIN 2,3,4,7을 output mode로 설정해야 하므로, 해당자리에만 0b0011(=0x3)을 넣어주면 된다.

```
*((volatile unsigned int *)0x40011000) &= 0xFF0000FF; // init port C
*((volatile unsigned int *)0x40011000) |= 0x00888800; // port C input mode

*((volatile unsigned int *)0x40011400) &= 0xFF0000FF; // init port D
*((volatile unsigned int *)0x40011400) |= 0x30033300; // port D output mode
```

4. GPIO의 input, output data를 통한 LED 제어

레퍼런스의 port input data register(IDR), port bit set/reset register(BSRR)을 통해 조이스틱의 input data를 가져오고, LED를 점등할 것이다.

Input의 경우, 조이스틱 IDR을 사용하기 위해, Port C의 주소(0x40011000) + offset (0x08) => 0x40011008 값을 가져온다.

Output의 경우, LED의 BSRR을 사용하기 위해, Port D의 주소(0x40011400) + offset (0x10) => 0x40011410에 원하는 값을 넣어준다. BSRR의 경우 하위 16bit가 값을 set하므로, 만약 2,3번째 PIN LED를 키고자 한다면, 0b1100으로, 0x0000000C를 레지스터에 넣어주면 된다. 또한 상위 16bit가 값을 reset하므로, 2,3번째 PIN LED를 끄고자 한다면 0x000C0000을 레지스터에 넣어주면 된다.

9.2.5 Port bit set/reset register (GPIOx_BSRR) (x=A..G)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 **BRy**: Port x Reset bit y (y= 0 .. 15)

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Reset the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

Bits 15:0 **BSy**: Port x Set bit y (y= 0 .. 15)

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Set the corresponding ODRx bit

4-1. DOWN의 경우 -> LED 1,2 OFF

PD 2,3을 0으로 만들어야 한다. DOWN인지 확인하기 위해서는 0x40011008(조이스틱 IDR)의 2번째 bit에 '0'이 들어왔는가를 체크해야 한다. 2번째 bit값의 NOT연산 값과, 0x1의 AND연산결과가 1이 된다면, 해당 if문을 만족하므로 0x000C0000을 레지스터에 넣어줘, PD 2,3을 reset시켜 LED를 끈다.

4-2. LEFT의 경우 -> LED 3,4 OFF

PD 4,7를 0으로 만들어야 한다. LEFT인지 확인하기 위해서는 0x40011008(조이스틱 IDR)의 3번째 bit에 '0'이 들어왔는가를 체크해야 한다. 3번째 bit값의 NOT연산 값과, 0x1의 AND연산결과가 1이 된다면, 해당 if문을 만족하므로 0x00900000을 레지스터에 넣어줘, PD 4,7을 reset시켜 LED를 끈다.

4-3. RIGHT의 경우 -> LED 3,4 ON

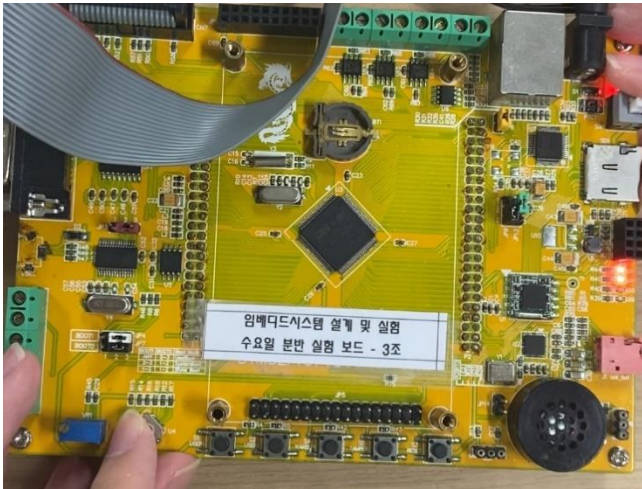
PD 4,7을 1으로 만들어야 한다. RIGHT인지 확인하기 위해서는 0x40011008(조이스틱 IDR)의 4번째 bit에 '0'이 들어왔는가를 체크해야 한다. 4번째 bit값의 NOT연산 값과, 0x1의 AND연산결과가 1이 된다면, 해당 if문을 만족하므로 0x00000090을 레지스터에 넣어줘, PD 4,7을 set시켜 LED를 켤다.

4-4. UP의 경우 -> LED 1,2 ON

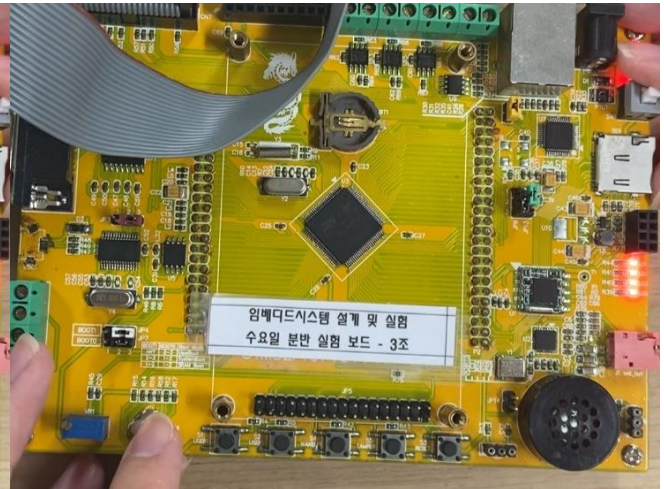
PD 2,3을 1으로 만들어야 한다. UP인지 확인하기 위해서는 0x40011008(조이스틱 IDR)의 5번째 bit에 '0'이 들어왔는가를 체크해야 한다. 5번째 bit값의 NOT연산 값과, 0x1의 AND연산결과가 1이 된다면, 해당 if문을 만족하므로 0x0000000C을 레지스터에 넣어줘, PD 2,3을 set시켜 LED를 켤다.

```
while(1){  
    if( 0x1 & ~(*((volatile unsigned int *)0x40011008) >> 2) ){ // Down 2 bit  
        printf("down");  
        *((volatile unsigned int *)0x40011410) |= 0x000c0000; // reset LED1, LED2 off  
    } else if( 0x1 & ~(*((volatile unsigned int *)0x40011008) >> 3) ) { // Left 3 bit  
        printf("left");  
        *((volatile unsigned int *)0x40011410) |= 0x00900000; // reset LED3, LED4 off  
    } else if( 0x1 & ~(*((volatile unsigned int *)0x40011008) >> 4) ) { // Right 4 bit  
        printf("right");  
        *((volatile unsigned int *)0x40011410) |= 0x00000090; // set LED3, LED4 on  
    } else if( 0x1 & ~(*((volatile unsigned int *)0x40011008) >> 5) ) { // Up 5 bit  
        printf("up");  
        *((volatile unsigned int *)0x40011410) |= 0x0000000c; // set LED1, LED2 on  
    } else {  
        printf("?");  
    }  
}
```

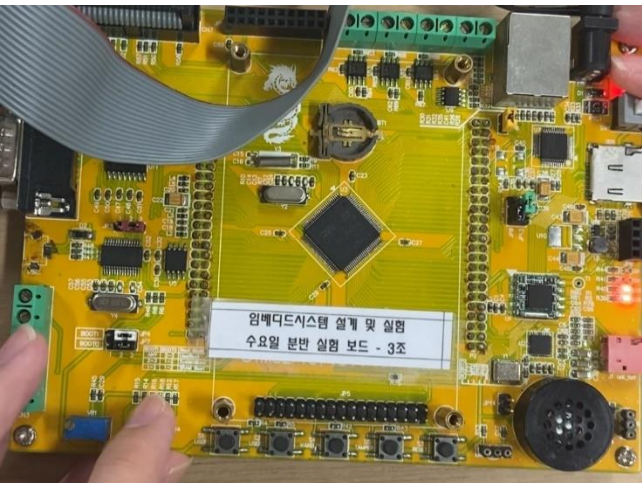

실험 결과



4. RIGHT -> LED 3,4 켜진 모습



3. UP -> LED 1,2 켜진 모습



2. LEFT -> LED 3,4 꺼진 모습



1. UP -> DOWN 1,2 꺼진 모습