

# 임베디드 설계 및 실험 3주차 결과 보고서

7조 이석우, 손수민, 김춘수, 정지용

## 1. 실험 목적

가. 레퍼런스와 스키매틱 등을 참고하여, register의 주소 설정 방법과 임베디드 설계의 기본 원리 이해

나. Cortex-m3 보드의 사용법을 파악

다. GPIO를 이용하여 조이스틱을 통한 LED점멸 제어

## 2. 함수 설명

### 가. init()

사용할 레지스터들에 대한 초기화를 하는 함수이다. 먼저 RCC를 사용하여 사용하고자 하는 GPIO에 clock을 인가해야 한다. Port A~E 중에서 조이스틱의 up, left, right, down이 Port C, LED는 Port D에 설정되어 있고, 이것은 APB2 peripheral에 소속되어 있다. 따라서 레퍼런스를 참고하여 RCC\_APB2ENR 레지스터에 0b00110000 (=0x30)을 넣어 포트 C, D에 clock을 인가했다.  
( RCC\_APB2ENR = 0x30 ).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART 1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved			IOPB EN	IOPA EN	Res.	Res.	Res.	AFIO EN
	rw		rw	rw	rw	rw				rw	rw	rw	rw	rw	rw

그 다음 LED를 사용하기 위해 GPIOD\_CRL을 reset 해준다. 레퍼런스에 따르면 reset bit는 0x44444444이다. 그리고 LED를 general purpose output push-pull로 사용하기 위해 MODE 비트에 11, CNF 비트에 00을 넣는다. 즉, 각각의 번호 비트에 0x3을 주는데 LED는 2, 3, 4, 7번 핀을 사용하기 때문에 0x30033300을 인가한다.

### 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]				CNF6[1:0]				CNF5[1:0]				CNF4[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]				CNF2[1:0]				CNF1[1:0]				CNF0[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 **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

Bits 29:28, 25:24, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 **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.

그리고 조이스틱을 사용하기 위해 GPIOC\_CRL에 0x44444444를 넣어 reset 해주고, 조이스틱을 input with pull up/pull-down로 사용하기 위해 각 번호 핀에 1000(2)(=0x8) 을 준다. 즉, 조이스틱은 2, 3, 4, 5번핀을 사용하기 때문에 0x00888800을 인가한다. 그리고 조이스틱의 방향을 입력 받아 저장하기 위해 GPIOC\_IDR 레지스터를 reset한다. reset bit는 0x00000000 이다.

### 9.2.3 Port input data register (GPIOx\_IDR) (x=A..G)

Address offset: 0x08h

Reset value: 0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDRy**: Port input data (y= 0 .. 15)

These bits are read only and can be accessed in Word mode only. They contain the input value of the corresponding I/O port.

## 나. main( )

Init()함수를 통해 초기화를 한 후 while 반복문에서 조이스틱의 입력에 따라 if – else if 조건문을 통해 분기하여 LED를 조작한다. 조이스틱의 입력은 GPIOC\_IDR 레지스터를 통해 해당 방향의 비트가 0이 되는 방식으로 동작한다. 조이스틱 방향 입력의 핀 번호는 2, 3, 4, 5번이고 이는 0x4,

0x8, 0x10, 0x20 에 해당한다. 따라서 이 값과 GPIOC\_IDR 레지스터 값에 not연산 한 것을 AND (&) 연산하여 조이스틱의 입력을 확인하였다. 조이스틱에 입력이 들어오지 않을 때는 off() 함수에 의해 모든 LED가 꺼진다.

```
int main(void)
{
    init();

    while(1){
        if(~GPIOC_IDR & 0x4){ //0100
            down_led3on(); //down 하면 PD3 LED 가 켜짐
        }
        else if(~GPIOC_IDR & 0x20){ //0010 0000
            up_led2on(); //up 하면 PD2 LED 가 켜짐
        }
        else if(~GPIOC_IDR & 0x8){ //0000 1000
            left_led4on(); //left 하면 PD4 LED 가 켜짐
        }
        else if(~GPIOC_IDR & 0x10){ //0001 0000
            right_led7on(); //right 하면 PD7 LED 가 켜짐
        }
        else off();
    }
}
```

#### 다. off()

off함수는 점등된 LED를 끄는 기능을 수행하는 함수이다. LED는 Port D에서 다루기 때문에 GPIOD의 레지스터를 활용해야 한다. 비트에 set/reset을 할당하는 GPIOD\_BSRR을 살펴보자. GPIOD\_BSRR은 `#define GPIOD_BSRR *(volatile unsigned int *)0x40011410`으로 초기 설정이 되어 있다. 이는 Port D의 시작 범위가 0x40011400이고 여기에 BSRR의 offset 주소 0x10을 더한 값이다. 16진수를 풀어서 확인하면 0100 0000 0000 0001 0001 0100 0001 0000이다. 레퍼런스를 확인하면 0은 기본적으로 no action이고 상위 16자리는 1일 때 reset이고 하위 16자리는 1일 때 set이다.

우리는 2, 3, 4, 7번 LED를 끄고 싶다. 따라서 상위 16자리 중에서 2, 3, 4, 7번 자리의 비트에 1을 주고 나머지는 모두 no action을 하도록 한다. 그렇게 하면 GPIOD\_BSRR = 0x009c0000; 이 되고 우리가 제어하려는 LED가 꺼지게 된다.

#### 라. up\_led2on()

함수 이름에서 알 수 있듯이 조이스틱을 위로 당기면 PD 2 LED가 켜지도록 하는 함수이다. LED를 출력하기 위해서 GPIOD\_BSRR 레지스터를 사용한다. GPIOD\_BSRR 레지스터의 2번 BS 비트에 1을 주어 set을 시키면 PD 2 LED가 켜진다.

#### 마. down\_led3on()

조이스틱을 아래로 당기면 PD 3 LED가 점등하도록 하는 함수이다. up\_led2on() 함수와 유사한 작동 방식을 가진다. GPIOD\_BSRR = 0x08; (0b0000 1000)을 주어서 3번째 비트값을 set해주면 된다.

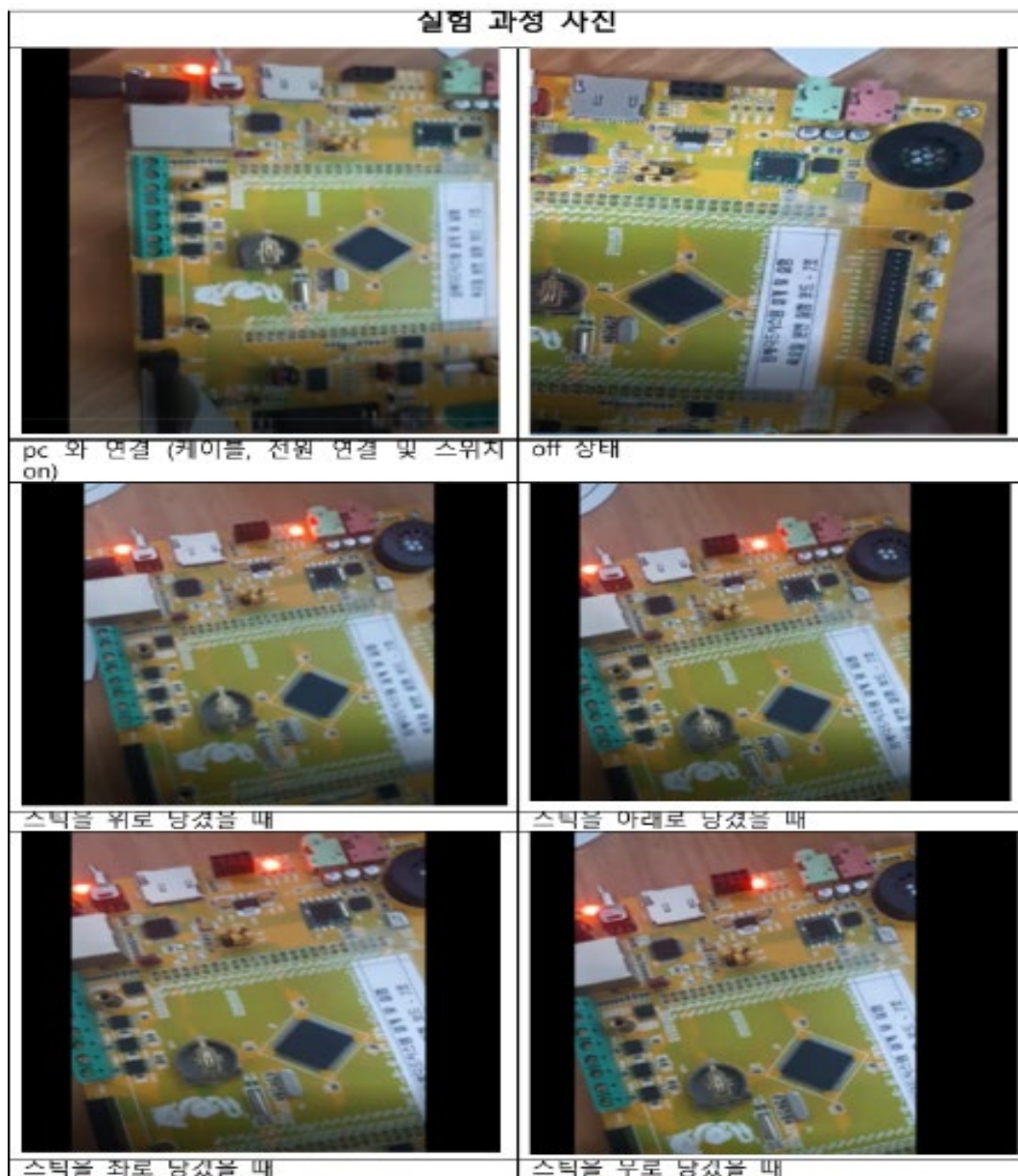
#### 바. left\_led4on( )

조이스틱을 왼쪽으로 당기면 PD 4 LED가 켜지도록 하는 함수이다. GPIOD\_BSRR = 0x10;으로 PD 4 LED 비트를 set한다(0b0001 0000).

#### 사. right\_led7on( )

조이스틱을 오른쪽으로 당기면 PD 7 LED가 점등되도록 하는 함수이다. GPIOD\_BSRR = 0x80;으로 PD 7 LED 비트 자리에 set을 하였다(0b1000 0000).

### 3. 실험 과정



#### 4. 실험 결론 및 향후 계획

Embedded system 설계를 처음 해보기 때문에 register의 주소에 직접 접근하여, 코드를 작성하는 과정에서 많은 어려움이 있었다. 수업에서 제공된 Datasheet, Schematic과 레퍼런스를 이용하여, 나름대로 차근차근 접근했다. 구글에서 자료를 찾아보고, 레퍼런스를 참고하며, register와 bit의 특징을 이해하고 우선 LED를 점멸하는 데 노력을 기울였다. 프로젝트의 원활한 진행을 위해 향후, 이러한 설계 방식에 익숙해지도록 반복을 하여, 숙달할 수 있도록 노력해야겠다.