



[임베디드 설계 및 실험] 2 조 3 주차 결과 보고서

- GPIO 제어, 조이스틱 활용 LED 켜기



■ 과 목 명	임 베 디 드 설 계 및 실 험
■ 담 당 교 수	김 원 석
■ 제 출 일	2 0 2 2 . 0 9 . 2 2
■ 학 과	정 보 컴 퓨 터 공 학 부
■ 조 원	202055600 정홍빈 201924617 끼얏띠풍유엔 201824534 윤상호 201824636 이강우

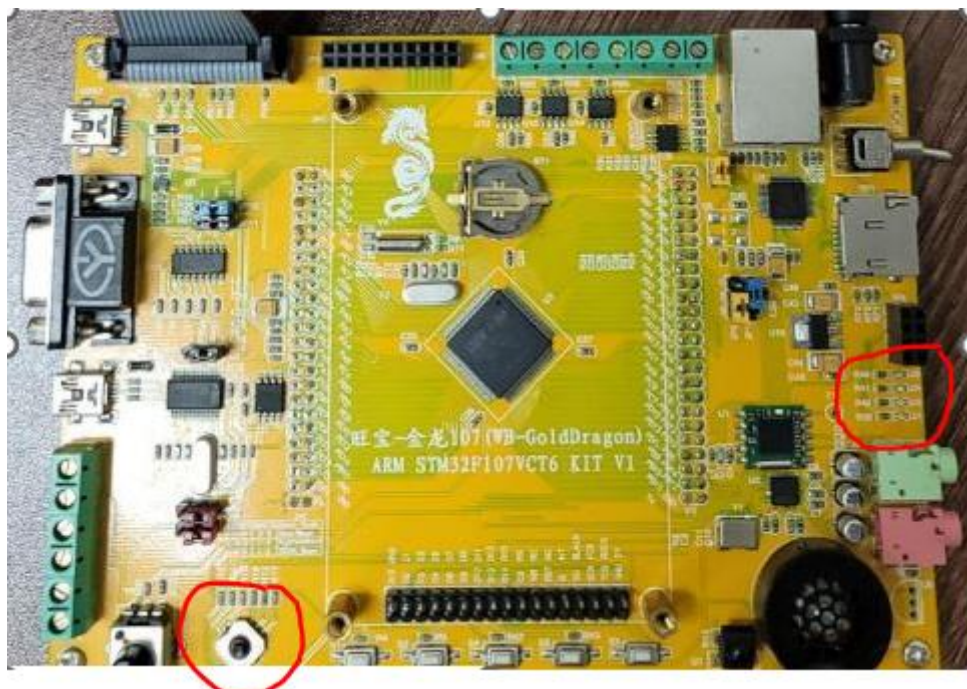
1. 실험 목적

임베디드 시스템 설계의 기본 원리를 습득하여 디버깅 툴 사용방법과 레지스터 제어를 통한 임베디드 펌웨어를 개발한다.

- 개발 환경 구축
- IAR Embedded Workbench 프로젝트 생성 및 설정
- 데이터시트와 레퍼런스 매뉴얼을 참고하여 레지스터 및 주소에 대한 이해
- GPIO(General-Purpose Input/Output) 사용을 통한 LED 제어.

2. 실험 원리 및 이론.

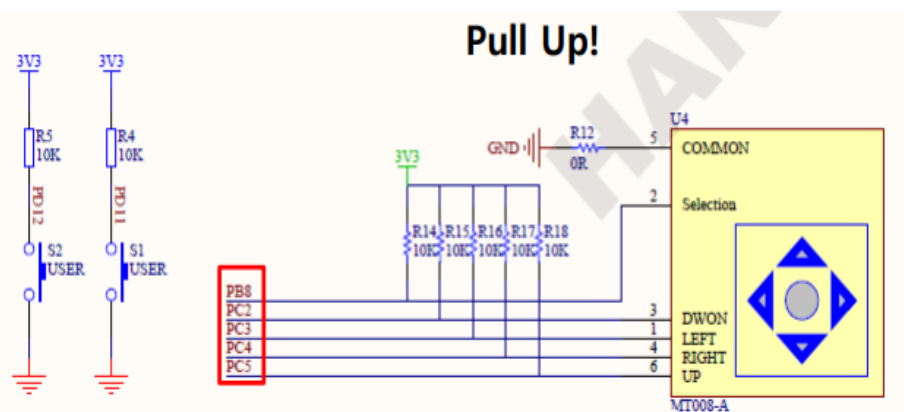
- 실험 보드



위의 실험보드 사진에서 표시된 왼쪽 아래가 조이스틱, 오른쪽이 LED 이다.

이번 실험 내용은 제공된 보드의 조이스틱을 활용한 LED 제어이다.

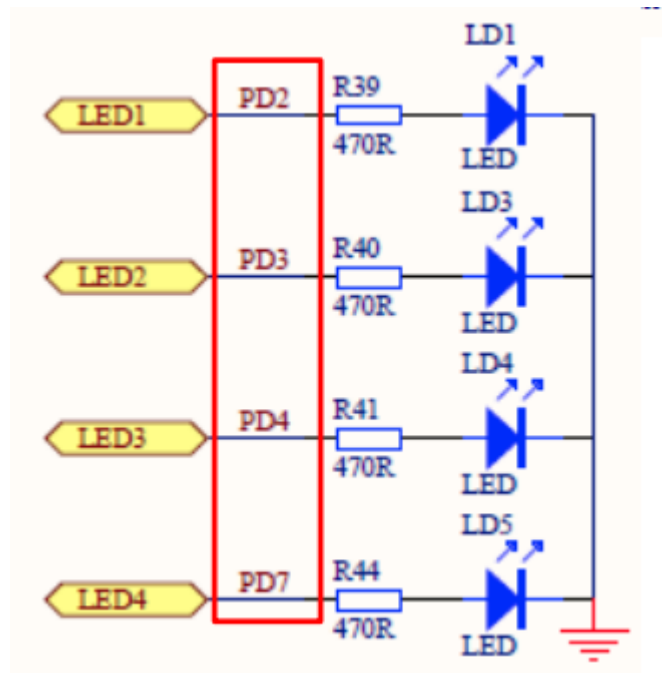
- 조이스틱 회로도



위의 회로도에서 보드의 조이스틱의 상하좌우 움직임이 입력신호로 들어간다.

회로도를 보면 조이스틱이 아래(PC2), 왼쪽(PC3), 오른쪽(PC4), 위(PC5)에 연결되어 있다.

■ LED 회로도



위의 회로도에서 연결방식은 LED1(PD2), LED2(PD3), LED3(PD4), LED4(PD7) 이다.

■ 실험 제공 자료

실험 제공 자료에 포함된 Datasheet 을 통해 레지스터의 Base 주소 값을 얻고, Reference Manual 을 참고해서 우리가 설정할 offset 값과 대입 값을 찾을 수 있다.

(Base 주소 + offset 값)을 통해 우리가 원하는 레지스터에 접근, 제어를 할 수 있다.

[Datasheet] P.33 Figure 5. Memory map

➔ (RCC Base: 0x4002 1000)

AHB	Reserved	0x4002 2400 - 0x4002 2FFF
	Flash interface	0x4002 2000 - 0x4002 23FF
	Reserved	0x4002 1400 - 0x4002 1FFF
	RCC	0x4002 1000 - 0x4002 13FF
	Reserved	0x4002 0800 - 0x4002 0FFF

➔ (APB2 Port C Base: 0x4001 1000 and Port D Base : 0x4001 1400)

APB2	ADC1	0x4001 2400 - 0x4001 27FF
	Reserved	0x4001 1C00 - 0x4001 23FF
	Port E	0x4001 1800 - 0x4001 1BFF
	Port D	0x4001 1400 - 0x4001 17FF
	Port C	0x4001 1000 - 0x4001 13FF
	Port B	0x4001 0C00 - 0x4001 0FFF
	Port A	0x4001 0800 - 0x4001 0BFF
	EXTI	0x4001 0400 - 0x4001 07FF

[Reference Manual]

9.2.4 Port output data register (GPIOx_ODR) (x=A..G)

Address offset: 0x0C

Reset value: 0x0000 0000

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
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

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

Bits 15:0 **ODRy**: Port output data (y= 0 .. 15)

These bits can be read and written by software and can be accessed in Word mode only.

Note: For atomic bit set/reset, the ODR bits can be individually set and cleared by writing to the GPIOx_BSRR register (x = A .. G).

제목에는 어떤 설정을 담당하는 부분인지 나와 있고, 바로 밑에 이 설정을 이용하기 위해 base 에 더해 줘야할 offset 주소값이 있다.

Offset 주소 바로 밑에는 이 주소로 접근 시 레지스터가 가지는 초기값이 나와 있다.

그림에는 세부 설정 항목으로 몇 번 레지스터를 어떤 모드로 할지, 활성화시킬지 말지 와 같은 세부사항을 비트 값으로 설정 가능하다.

그림 아래에는 어떤 비트가 어떤 설정을 가지는지에 대한 설명이 나와있다.

■ Volatile Keyword

이번 실험에서 계속 사용하게 되는 volatile 키워드는 컴파일러가 해당 volatile 변수를 최적화에서 제외하여 항상 메모리에 접근할 수 있도록 하는 키워드이다.

즉, volatile 변수를 참조하면 레지스터의 로드 된 값을 사용하지 않고 메모리를 참조한다.

[예시] `*(volatile unsigned int *) (R_CC + 0x18) |= 0x30;`

3. 실습 진행 과정

■ 프로젝트 생성 및 IAR 프로그램 설정.

IAR Embedded Workbench 프로그램을 실행 후 작업할 프로젝트 폴더와 같은 구조로 IAR 작업공간을 설정한다.

IAR 작업공간 설정이 완료되면 실험보드와 컴퓨터를 연결한다.

(※ 전압에 맞지 않는 전원연결시 보드가 탈 수 있으니 주의해야한다.)

연결된 보드가 정상 작동하는지 확인하기위해 디버깅 모드를 실행하였다.

- 조이스틱과 LED 제어를 위한 Datasheet/Reference Manual 참고.

Step1) 앞에 이론에서 우리는 Datasheet 분석을 통해, 우리가 제어하고자 하는 레지스터가 APB2 에 속해 있다는 것을 알고 있다. APB2 에 속해 있는 레지스터를 사용하기 위해서는 먼저 Clock 을 인가해주어야 한다. 이 내용은 Reference Manual 7.3.7 을 참고하였다.

7.3.7 APB2 peripheral clock enable register (RCC_APB2ENR)

Address: 0x18

Reset value: 0x0000 0000

Access: word, half-word and byte access

No wait states, except if the access occurs while an access to a peripheral in the APB2 domain is on going. In this case, wait states are inserted until the access to APB2 peripheral is finished.

Note: When the peripheral clock is not active, the peripheral register values may not be readable by software and the returned value is always 0x0.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Low-, medium-, high- and XL-density reset and clock control (RCC)

RM0008

Reserved										TIM11 EN	TIM10 EN	TIM9 EN	Reserved		
										r/w	r/w	r/w			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 EN	USART 1EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w

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

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

[Reference 7.3.7 의 일부 내용]

Clock enable register 의 Offset 값 0x18 을 우리가 제어하고자 하는 레지스터의 Base 주소에 더해주어 활성화시킬 수 있다. 우리가 제어할 레지스터는 RCC 에 있으므로, Datasheet 에 Base 주소값을 보면, RCC Base: 0x4002 1000 이다.

즉, RCC_APB2 레지스터에 접근하기 위해서는 $0x4002\ 1000 + 0x18 = 0x4002\ 1018$ 로 접근하면 된다.

우리는 위 그림에서 빨간색으로 표시한 Port C 와 Port D 가 필요하다. 따라서 그림과 설명에 따라 4,5 번째 비트에 1 을 인가해주면 PC, PD 가 활성화된다.

즉, $0x4002\ 1018 = 0x30(2\ \text{진수로 } 110000)$

Step2) APB2_RCC 에 클럭을 활성화시켰으므로, 각 포트가 low(풀 업)로 동작하는지 high(풀 다운)으로 동작하는지 설정해주어야 한다. 이 내용은 Reference Manual 9.2.1 을 참고하였다.

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]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
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]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:30, 27:26, **CNFy[1:0]**: Port x configuration bits (y= 0 .. 7)
 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 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, **MODEy[1:0]**: Port x mode bits (y= 0 .. 7)
 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 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.

[Reference 9.2.1 일부 내용]

우리가 제어할 조이스틱과 LED 는 각각 Port C, Port D 에 속해 있으므로

Datasheet 을 찾아보면, Port C Base: 0x4001 1000, Port D Base: 0x4001 1400 이다.

위 그림설명에서 Base 에 offset 값인 0x00 을 더하면 각 포트는 GPIO_CRL 레지스터 주소가 된다. 초기화 값은 0x4444 4444 로 리셋 된다. (cf. 4 == 0100)

비트구조가 CNF2 비트 + MODE2 비트 이런 식으로 구성되어 있기 때문에 설정에 따라 비트 값을 설정해주어야 한다.

우리는 조이스틱 Port C 를 Input, LED Port D 를 Output 으로 사용해야한다.

PC 는 2,3,4,5 번째 비트를 Input, PD 는 2,3,4,7 번째 비트를 Output 으로 설정한다.

즉, PD: 0x4001 1400 + 0x00 = 0x3003 3300(0011 0000 0000 0011 0011 0011 0000 0000)

PC : 0x4001 1400 + 0x00 = 0x00888800(0000 0000 1000 1000 1000 1000 0000 0000)

Step 3) RCC 와 PORT C, D 의 설정이 끝났으므로, LED 제어를 위한 코드작성을 한다.

```
typedef volatile unsigned int* VOL;

int main(void)
{
    *(VOL) (R_CC + 0x18) |= 0x30;          // clock enable port C, port D
    *(VOL) (PORT_D + 0x00) = 0x30033300;    // configuration low
    0011 0000 0000 0011 0011 0011 0000 0000
    *(VOL) (PORT_C + 0x00) = 0x08888800;    // configuration low
```

위의 코드는 step1, 2 에서 진행한 내용을 기술한 것이다.

down2	100(2)	= 4(10)	= 0x04	PD3(LED2)	0000 1000(2)	= 8(10)	= 0x08
left3	1000(2)	= 8(10)	= 0x08	PD4(LED3)	0001 0000(2)	= 16(10)	= 0x10
right4	10000	= 16(10)	= 0x10	PD2(LED1)	0000 0100(2)	= 4(10)	= 0x04
up5	100000(2)	= 32(10)	= 0x20	PD7(LED4)	1000 0000(2)	= 128(10)	= 0x80

위에 정리된 것처럼 down(PC2), left(PC3), right(PC4), up(PC5)이고,
LED1(PD2), LED2(PD3), LED3(PD4), LED4(PD7) 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.

PORT_C 는 조이스틱 제어 포트로 Port C IDR(Input Data Register)이다.

```
if(!(*VOL)(PORT_C + 0x08) & 0x20)){          //up 5
    *(VOL) (PORT_D + 0x10) = 0x80;            //set
    *(VOL) (PORT_D + 0x14) = 0x1C;            //reset
}
```

위의 코드처럼 PORT_C + 0x08 로 Port C IDR 에 접근하여 up(PC5)가
만약 0(ON)이고, 5 번째비트이면 조이스틱이 위로 움직인 것으로 간주한다.

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

문제조건이 조이스틱이 up 하면 LED4 번이 켜져야 하므로

PORT_D + 0x10 으로 Port D ODR(Output Data Register)로 접근하여 LED4(PD7)에 1 값을 set 하여준다. 9.2.5 의 설명에 따라 0x80(PD7, 7 번째 비트에 1 인가)값을 인가해주면 LED4 가 ON 이 된다.

문제 조건에 의해 나머지 LED 들은 모두 꺼져 있어야 한다.

9.2.6 Port bit reset register (GPIOx_BRR) (x=A..G)

Address offset: 0x14

Reset value: 0x0000 0000

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
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

Bits 31:16 Reserved

Bits 15:0 **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

9.2.6 에 설명에 따라 나머지 LED 를 OFF 하기위해서 LED1~3 는 각각 PD

2,3,4 번째 비트에 1 을 가하면 리셋이 되므로 10000 + 1000 + 100 = 11100(2) =

28(10) = 0x1C 가 된다. 따라서 PORT_D + 0x14 는 BRR(bit reset register)에

접근하여 LED1,2,3 의 값을 0 으로 리셋 하여 LED 를 OFF 한다.


```
else if(!(*VOL)(PORT_C + 0x08) & 0x8)){           //left 3
    *(VOL) (PORT_D + 0x10) = 0x10;
    *(VOL) (PORT_D + 0x14) = 0x8C;
}
else if(!(*VOL)(PORT_C + 0x08) & 0x04)){           //down 2
    *(VOL) (PORT_D + 0x10) = 0x8;
    *(VOL) (PORT_D + 0x14) = 0x94;
}
else if(!(*VOL)(PORT_C + 0x08) & 0x10)){           //right 4
    *(VOL) (PORT_D + 0x10) = 0x4;
    *(VOL) (PORT_D + 0x14) = 0x98;
}
```

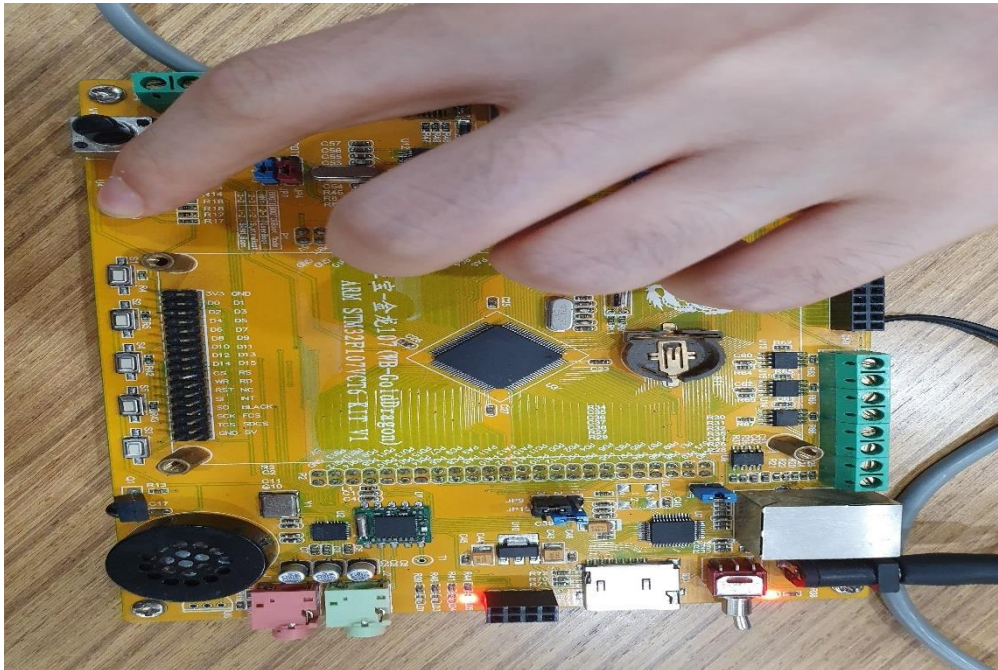
위의 코드들 또한 step3 의 과정과 똑같이 진행하면 된다.

```
else{
    *(VOL) (PORT_D + 0x14) = 0x9C;
}
```

중립을 구현하기 위해서 위의 코드처럼 모든 LED(1~4)를 더한 값이 0x9C 이므로
BRR 에 인가해주면 모든 LED 가 OFF 가 된다.

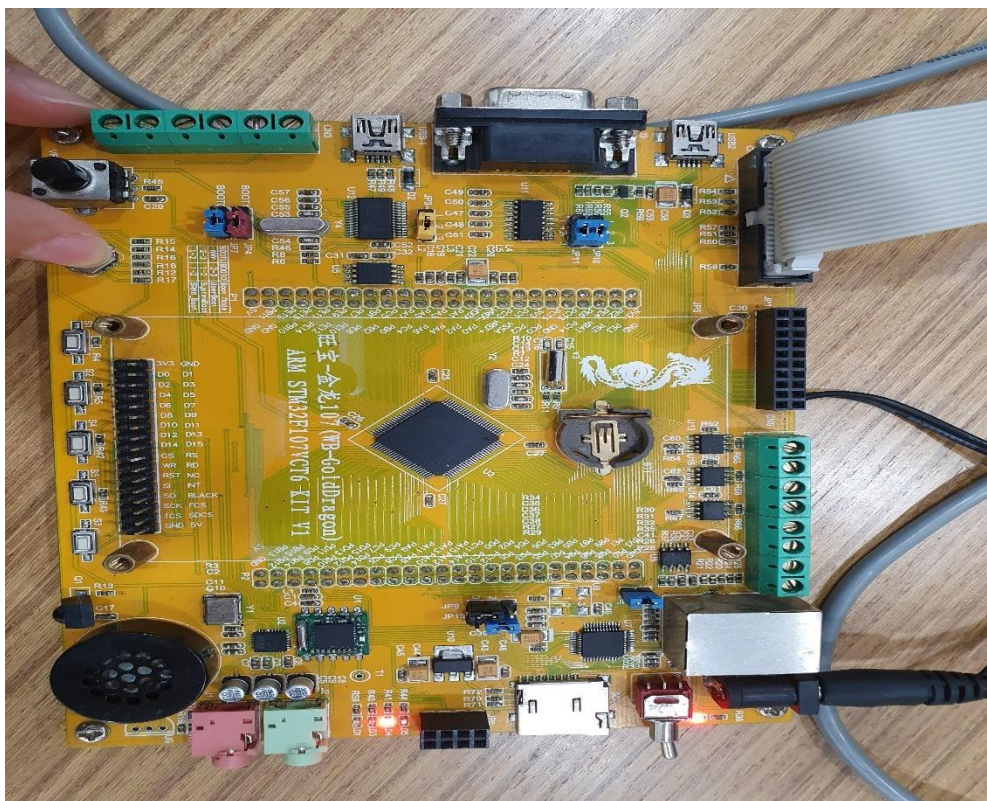
4. 실험 결과

■ Up



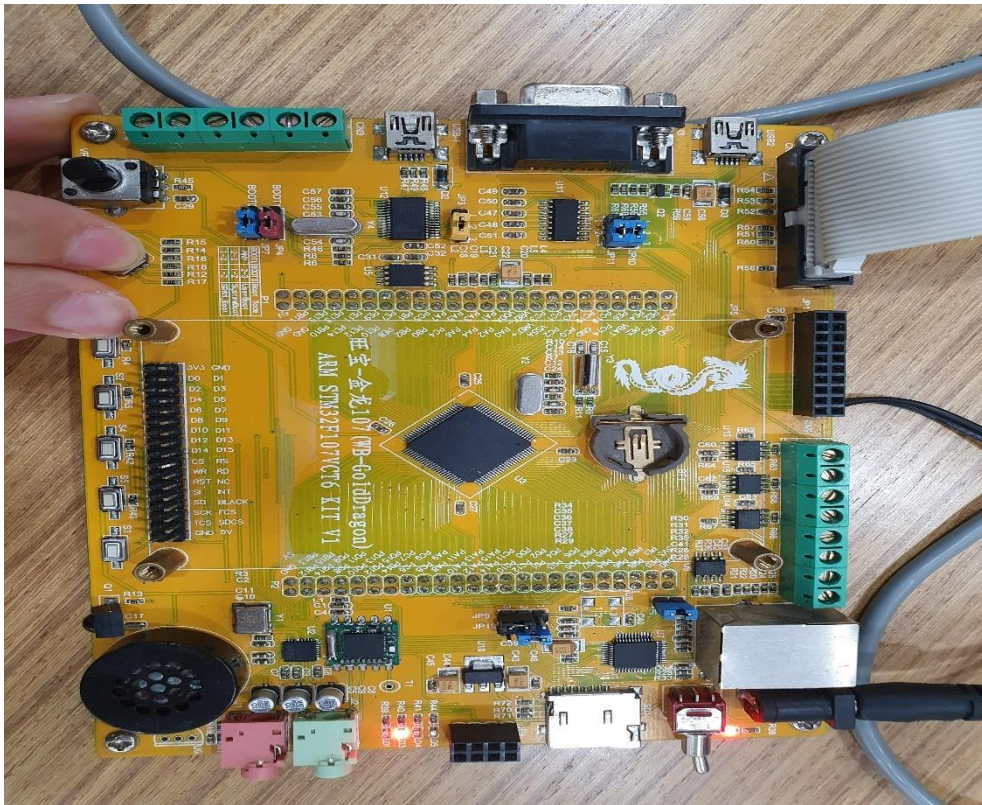
LD5 에 불이 들어온다.

■ Left



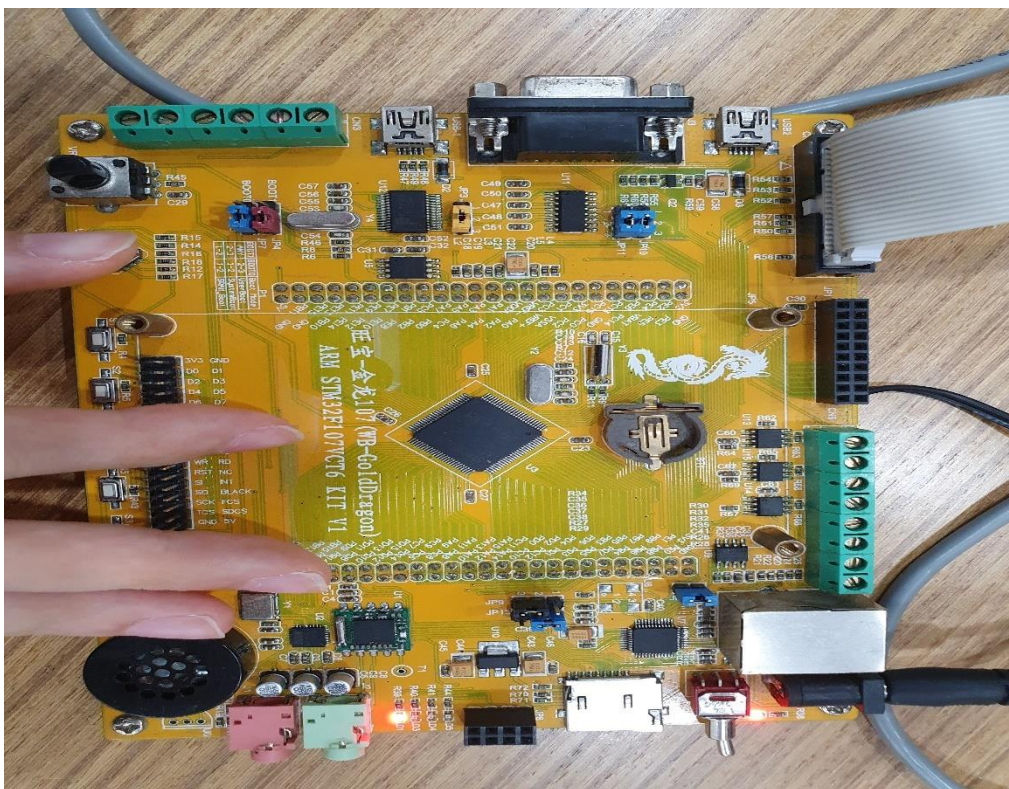
LD4 에 불이 들어온다.

■ Down



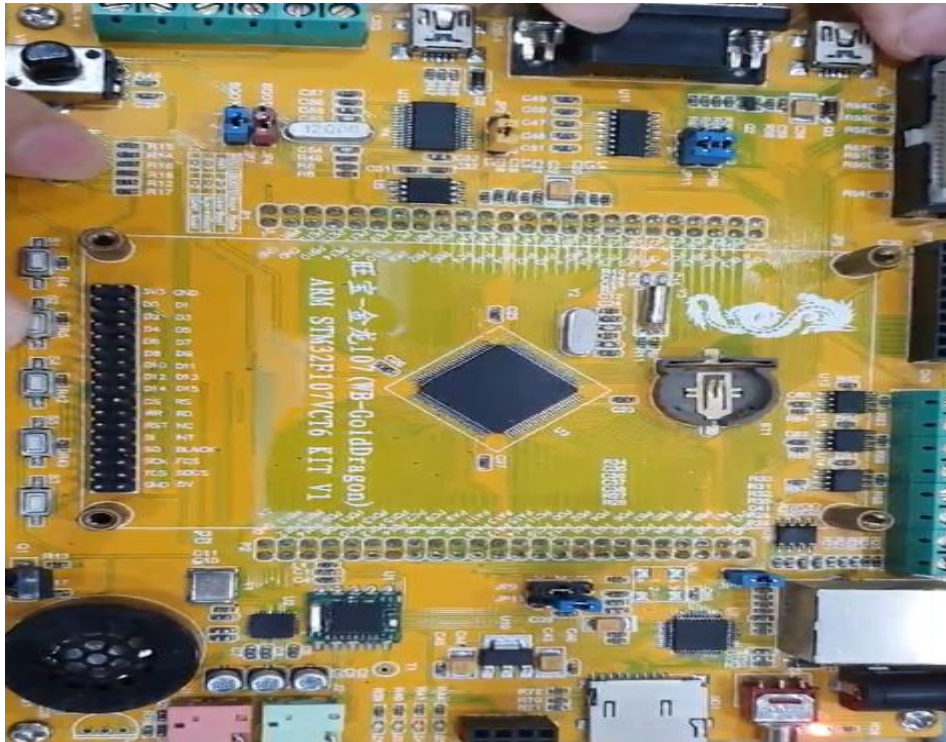
LD3 에 불이 들어온다.

■ Right



LD1 에 불이 들어온다.

■ 중립



중립인 경우에는 어떤 LED 에도 불이 들어오지 않는다.

5. 결론

조이스틱과 LED 를 활용한 GPIO 제어 실험을 통해, 임베디드 시스템에 사용되는 보드와 IAR 프로그램 사용방식을 알게 되었다. 또한 Datasheet 과 Reference Manual 을 참고하여 레지스터의 주소를 찾고 설정하는 제어방법을 이해할 수 있었다. c++과 같은 언어와 달리 레지스터 주소에 직접 접근하여 대입 값을 비트단위로 연산하여 레지스터를 설정하는 방법을 익힐 수 있었다.

마지막으로, 보드의 작동방식과 LED, 조이스틱을 활용한 C 코딩을 함으로써 응용 펌웨어 개발을 한 번 연습해 볼 수 있었다.