

실험 결과 보고서 : 4주차 GPIO 조작

실험날짜: 2023-09-23

실험자:3조(강태진, 김기윤, 김도완, 임성표)

목차

1. 실험 목적.....	3
2. 세부 목표.....	3
3. 실험 과정.....	3
4. 실험 결과.....	9
5. 참고 문헌.....	13

1. 실험 목적

- 임베디드 시스템 설계의 기본 원리 습득
- 디버깅 툴 사용방법 및 레지스터 제어를 통한 임베디드 펌웨어 개발

2. 세부 목표

- IAR Embedded Workbench 개발 환경 구축
- Datasheet 및 Reference Manual을 참고하여 해당 레지스터 및 주소에 대한 설정 이해
- GPIO(general-purpose input/output)를 사용하여 LED제어
- 오실로스코프에 대한 이해와 DebugPin설정

3. 실험 과정

1) Datasheet 및 Reference Manual을 참고해 각 레지스터의 값 설정

- Datasheet의 memory map에서 base address를 가져옴

APB2	RCC	0x4002 1000 - 0x4002 13FF
	Reserved	0x4002 0800 - 0x4002 0FFF
	DMA2	0x4002 0400 - 0x4002 07FF
	DMA1	0x4002 0000 - 0x4002 03FF
	Reserved	0x4001 3C00 - 0x4001 FFFF
	USART1	0x4001 3800 - 0x4001 3BFF
	Reserved	0x4001 3400 - 0x4001 37FF
	SPI1	0x4001 3000 - 0x4001 33FF
	TIM1	0x4001 2C00 - 0x4001 2FFF
	ADC2	0x4001 2800 - 0x4001 2BFF
	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

- RCC_APB2ENR: RCC에 전원 인가(clock enable 설정 register)

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
Reserved										TIM11 EN	TIM10 EN	TIM9 EN	Reserved		
										rw	rw	rw			
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
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Port A,B,C,D에 clock을 enable하기 위해 2,3,4,5번째 비트를 1로 설정

- GPIOx_CRL, GPIOx_CRH(x=A...G): GPIO핀의 input/output 설정 register

9.2 GPIO registers

Refer to [Section 2.1 on page 47](#) for a list of abbreviations used in register descriptions.

The peripheral registers have to be accessed by words (32-bit).

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

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.

9.2.2 Port configuration register high (GPIOx_CRH) (x=A..G)

Address offset: 0x04

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[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
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30, 27:26, **CNFy[1:0]**: Port x configuration bits (y= 8 .. 15)
 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= 8 .. 15)
 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.

CRL은 0~7번 핀, CRH는 8~15번 핀의 input/output 모드를 설정 가능

- GPIOx_IDR(x=A..G): 해당 I/O Port의 input data를 포함하는 register

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.

- GPIOx_ODR (x=A..G): 해당 I/O Port의 output data를 포함하는 register

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).

- GPIOx_BSRR (x=A..G): 해당 GPIOx_ODR register의 비트를 set/reset하는 register

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

2) IAREW에서 프로젝트 생성 후 관련 설정 변경

3) 버튼을 이용한 LED 제어

key1 push -> LED1, 3 on

key2 push -> LED 1, 3 off

key3 push -> LED 2, 4 on

key4 push -> LED 2, 4 off

(1) RCC를 사용하여 사용하고자 하는 GPIO에 clock을 인가

```
// port A, B, C, D에 clock 인가 -> set 1
// 2번, 3번, 4번, 5번
// 0x04 + 0x08 + 0x10 + 0x20
RCC_APB2ENR |= 0x3C;
```

(2) 사용하려는 GPIO Port, Pin의 input/output 설정 : Key -> Input, LED -> Output 설정

```
// A0, key4 (0-7)
// A0 -> 0번째 비트를 1000으로 수정 -> push-pull + input 모드로 변경
GPIOA_CRL &= 0xFFFFFFFF0;
GPIOA_CRL |= 0x00000008;

// B10, key2 (8-15)
// B10 -> 10번째 비트를 1000으로 수정 -> push-pull + input 모드로 변경
GPIOB_CRH &= 0xFFFFF0FF;
GPIOB_CRH |= 0x00000800;

// C4, key1
// C4 -> 4번째 비트를 1000으로 수정 -> push-pull + input 모드로 변경
GPIOC_CRL &= 0xFFF0FFFF;
GPIOC_CRL |= 0x00080000;

// C13, key3
// c13 -> 13번째 비트를 1000으로 수정 -> push-pull + input 모드로 변경
GPIOC_CRH &= 0xFF0FFFFF;
GPIOC_CRH |= 0x00800000;

// D2, D3, D4, D7
// 2, 3, 4, 7비트를 0001로 수정 -> analog + output 모드로 변경
```



```

GPIOD_CRL &= 0xFF0000FF;
GPIOD_CRL |= 0x10011100;
// PD2, PD3, PD4, PD7
// LED1, 2, 3, 4 off -> port set bits
GPIOD_BSRR |= 0x9C; // 0x80 + 0x10 + 0x08 + 0x04

```

(3) Port bit set/reset을 이용해 제어

KEY1, 2, 3, 4는 input with pull-up이어서 버튼을 누르지 않은 상태일 때는 전류가 input pin으로 흘러서 input data 해당 pin의 bit 값이 1이 되고 버튼을 누른 상태일 때는 전류가 그라운드로 흘러서 값이 0이 된다. 그래서 (GPIOC_IDR & 0x10)같은 &식 앞에 !을 써야 버튼을 눌렀을 때 if 조건문이 참이 된다.

실험에 사용한 보드의 LED1, 2, 3, 4는 output data의 해당 pin의 bit가 reset될 때 켜지고 set될 때 꺼진다.

```

if (!(GPIOC_IDR & 0x10)) {
//BR2, BR4 -> 1로 변경
//LED1(PD2), LED3(PD4)을 ON
GPIOD_BSRR |= 0x140000;
//18번, 20번 비트 1 -> 2번, 4번 BR값 1로 변경 -> 위에서 set(꺼짐)으로 변경한
ODRx bit를 reset
// 0x100000 + 0x040000 == 0000 0000 0001 0100 0000 0000 0000 0000
}
// B10, key2(PB10) push -> LED 1, 3 off
if (!(GPIOB_IDR & 0x0400)) {
//BS2, BS4 -> 1로 변경
//LED1(PD2), LED3(PD4)을 OFF
GPIOD_BSRR |= 0x14;
//2번, 4번 비트 1 -> BS값 1로 변경 -> set(꺼짐)으로 설정
// 0x10 + 0x04 = ... 0001 0100
}
// C13, key3(PC13) push -> LED 2, 4 on
if (!(GPIOC_IDR & 0x2000)) {
GPIOD_BSRR |= 0x880000; // 0x800000 + 0x080000
}
// A0, key4(PA0) push -> LED 2, 4 off
if (!(GPIOA_IDR & 0x01)) {
GPIOD_BSRR |= 0x88; // 0x80 + 0x08
}

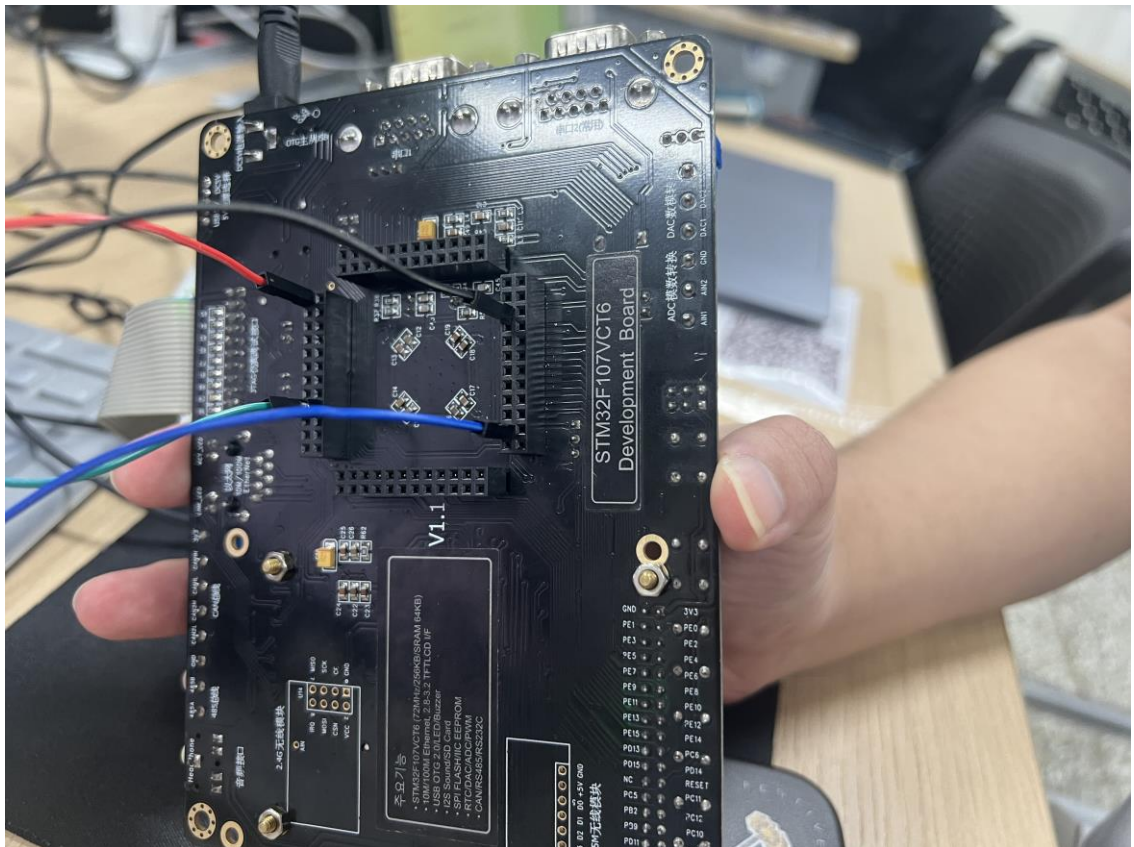
```

4) 정상적인 동작 유무 확인: 보드와 PC를 J-Link로 연결하고 코드를 보드에 다운

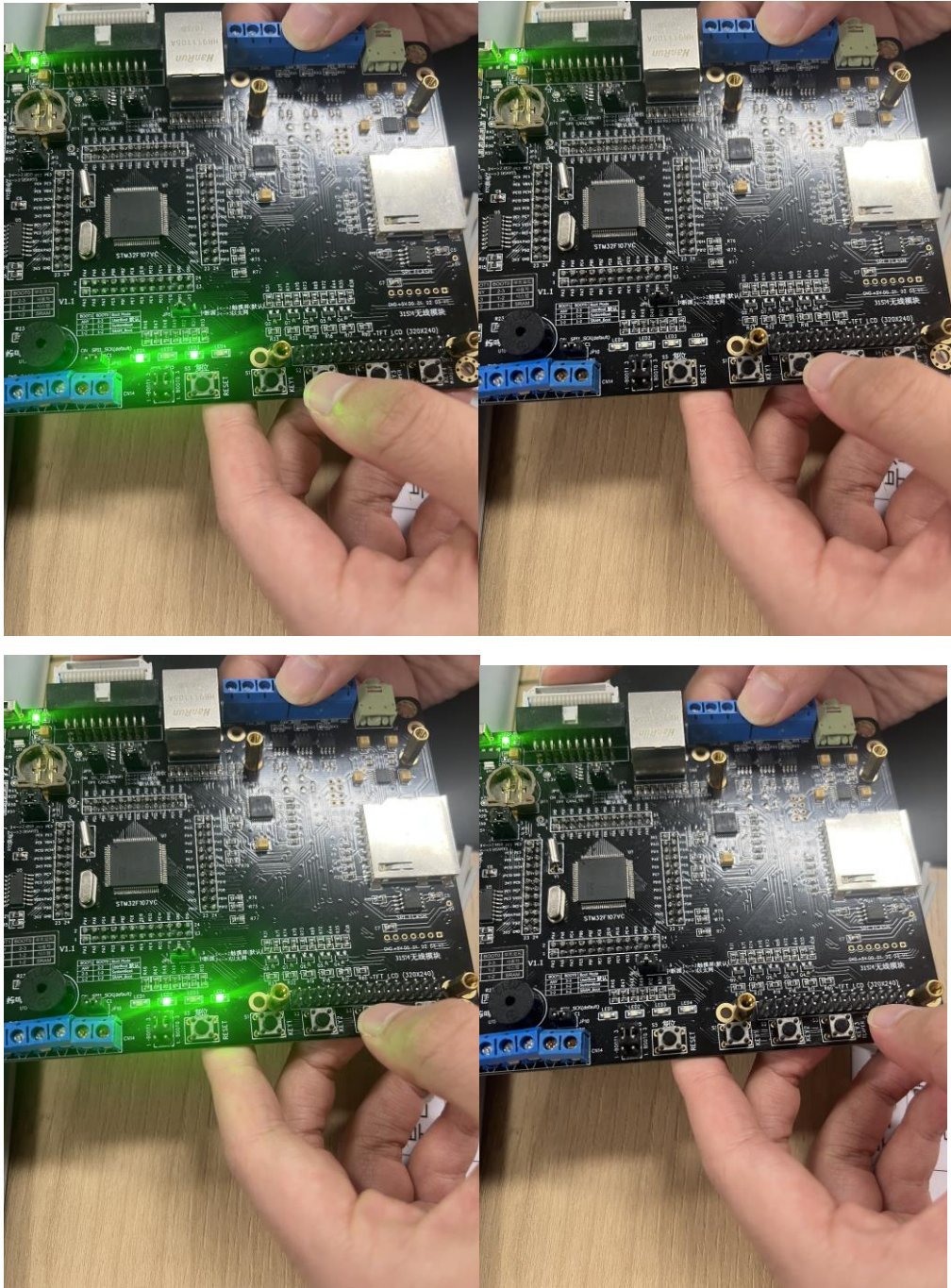
로드해서 작동 확인

5) 오실로스코프를 이용한 디버깅(버튼의 값 변화에 따른 LED 점등까지의 시간 측정, Digital Pin 의 trigger를 이용하여 캡처)

(1)보드의 그라운드와 Pin에 점프선을 꽂는다

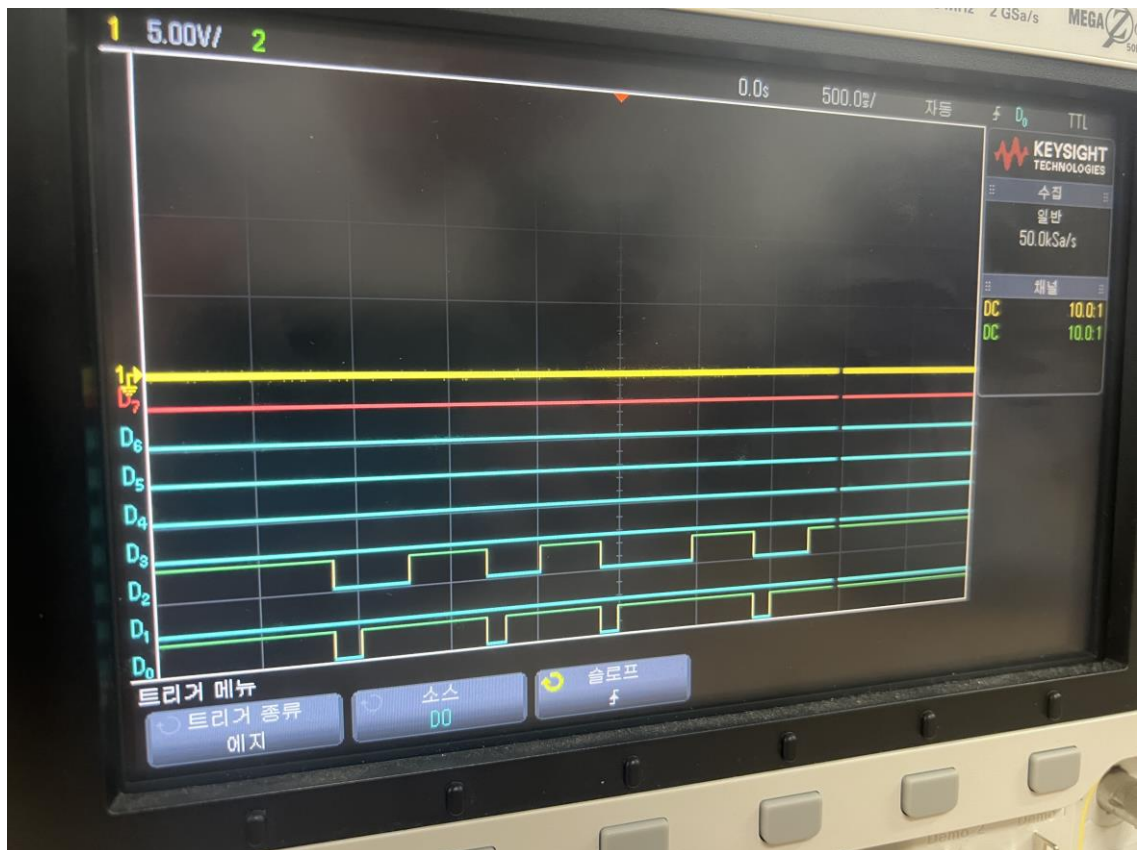


(2) 오실로스코프 디지털 핀의 D0, D2에 보드에 꽂은 점프선을 연결한다



모두 정상적으로 작동됨을 알 수 있다.

아래는 Key를 누를시 오실로스코프의작동 유무이다.



5. 참고 문헌

stm32_Datasheet.pdf

stm32_ReferenceManual.pdf