# 2023-2 임베디드 시스템 설계 및 실험
# 3주차 보고서

수요일(003) 분반 4조

## 개요

데이터시트와 레퍼런스 문서를 직접 리딩하여 임베디드 보드를 제어하는 능력을 향상하고 레지스터와 주소 제어를 통해 GPIO를 제어하는 방법을 알아본다.

## 목표

1. 임베디드 시스템의 기본 원리 습득
2. 레지스터와 주소 제어를 통한 임베디드 펌웨어 개발 이해

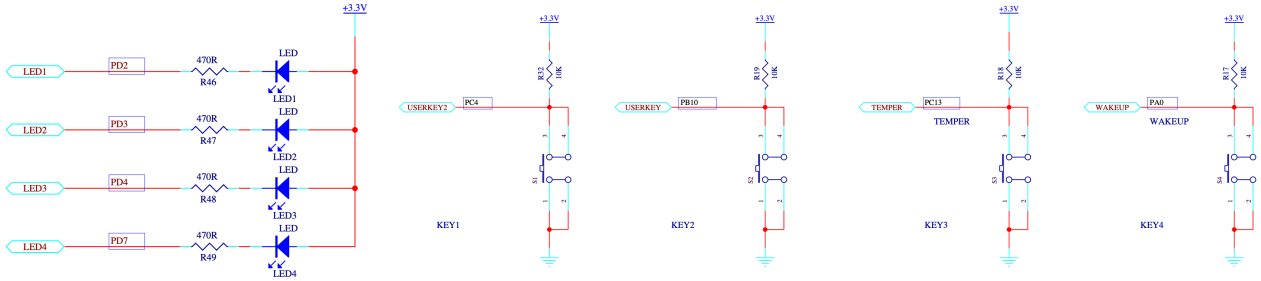
## 세부 실험 내용

1. Datasheet 및 Reference Manual을 참고하여 해당 레지스터 및 주소에 대한 설정 이해
2. IAR EW에서 프로젝트 생성 후 관련 설정 변경
3. 버튼을 이용한 LED 제어
   - KEY1: PD2, PD3 LED On
   - KEY2: PD2, PD3 LED Off
   - KEY3: PD4, PD7 LED On
   - KEY4: PD4, PD7 LED Off
4. 정상적인 동작 유무 확인
5. 오실로스코프를 이용한 디버깅

## 실험 과정

### 1. RCC를 사용하여 사용하고자 하는 GPIO에 clock 인가

Schematic 문서에 따르면 LED는 GPIO 포트 D, 버튼은 GPIO 포트 A, B, C를 사용한다.



데이터시트 문서의 메모리 맵에 따르면 GPIO 포트는 APB2 버스에 연결되어 있으므로 APB2 버스를 활성화하기 위해 RCC 레지스터 중 APB2 버스에 관련된 것을 찾는다.

### 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 |

사용할 각 포트에 대응하는 IOPxEN 비트를 1로 설정하면 해당 포트에 clock이 부여된다.

데이터시트 문서의 메모리 맵에 따르면 RCC 레지스터에 할당된 주소는 0x40021000이다. 여기에 RCC_APB2ENR 레지스터의 오프셋을 더해서 RCC_APB2ENR 레지스터의 주소를 얻고, GPIO 포트의 clock을 활성화한다.

```
#define RCC_APB2ENR  (*(volatile uint32_t*) 0x40021018)
```

```
// GPIO A, B, C, D 포트 clock 활성화
RCC_APB2ENR |= (1 << 2) | (1 << 3) | (1 << 4) | (1 << 5);
```

## 2. 사용하려는 GPIO Port, Pin의 input/output 설정

포트에서 0~7번째 핀은 CRL, 8~15번째 핀은 CRH을 사용한다. 즉 앞서 첨부한 LED와 버튼의 Schematic에 따르면 포트 A와 D는 CRL, 포트 B는 CRH, 포트 C는 CRL와 CRH 모두를 사용해야 한다.

| | |
|---|---|
| Port D | 0x4001 1400 - 0x4001 17FF |
| Port C | 0x4001 1000 - 0x4001 13FF |
| Port B | 0x4001 0C00 - 0x4001 0FFF |
| Port A | 0x4001 0800 - 0x4001 0BFF |

각 레지스터의 주소를 얻는다. CRL의 오프셋은 0x00, CRH의 오프셋은 0x04이다.

```
#define GPIOA_CRL    (*(volatile uint32_t*) 0x40010800)
#define GPIOB_CRH    (*(volatile uint32_t*) 0x40010C04)
#define GPIOC_CRL    (*(volatile uint32_t*) 0x40011000)
#define GPIOC_CRH    (*(volatile uint32_t*) 0x40011004)
#define GPIOD_CRL    (*(volatile uint32_t*) 0x40011400)
```

핀 모드를 설정하기 전에, 레지스터의 사용하려는 부분을 우선 0으로 초기화해야 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 |

예를 들어 포트 D는 2, 3, 4, 7번째 핀을 사용하므로 이 부분을 0으로 초기화하기 위해 포트 D의 주소와 ~0xF00FFF00을 AND 연산한다.

```
GPIOA_CRL &= ~0xF;          // 포트 A의 CRL 레지스터 초기화 (0번 핀)
GPIOB_CRH &= ~0xF00;        // 포트 B의 CRH 레지스터 초기화 (10번 핀)
GPIOC_CRL &= ~0xF0000;      // 포트 C의 CRL 레지스터 초기화 (4번 핀)
GPIOC_CRH &= ~0xF00000;     // 포트 C의 CRH 레지스터 초기화 (13번 핀)
GPIOD_CRL &= ~0xF00FFF00;   // 포트 D의 CRL 레지스터 초기화 (2, 3, 4, 7번 핀)
```

이제 핀 모드를 설정한다. 레퍼런스 문서에서 Bit Setting 값을 확인할 수 있다.

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.

Input mode: CNF[1:0] = ’10’, MODE[1:0] = ’00’ => 0b1000 = 0x8

Output mode: CNF[1:0] = ’00’, MODE[1:0] = ’11’ => 0b0011 = 0x3

input mode로 설정할 부분에는 0x8, output mode로 설정할 부분에는 0x3을 넣는다.

```
GPIOA_CRL &= ~0x8;          // KEY4: input mode
GPIOB_CRH &= ~0x800;        // KEY2: input mode
GPIOC_CRL &= ~0x80000;      // KEY1: input mode
GPIOC_CRH &= ~0x800000;     // KEY3: input mode
GPIOD_CRL &= ~0x30033300;   // LED: output mode
```

## 3. 버튼을 사용한 LED ON/OFF

GPIOx_IDR 레지스터로 버튼 입력 값을 읽어들이고 GPIOx_BRR, GPIOx_BSRR 레지스터로 LED 출력 값을 제어한다.

### 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.

버튼이 연결된 각 포트에 IDR의 오프셋 0x08을 더해 GPIOx_IDR 레지스터의 주소를 얻는다. 버튼을 누르면 해당하는 핀 자리 비트가 0으로 세팅된다. 따라서, 예를 들어 PC4에 연결된 KEY1에서의 입력 여부를 확인하려면, GPIOC_IDR과 0x10000를 AND 연산한 결과가 0이 되는지 확인하면 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

### 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 |

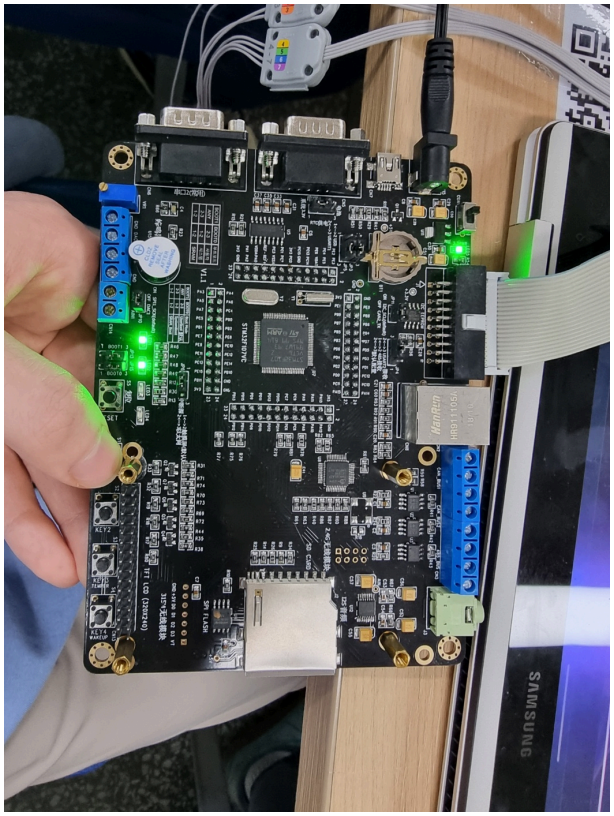
Bits 31:16 Reserved

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

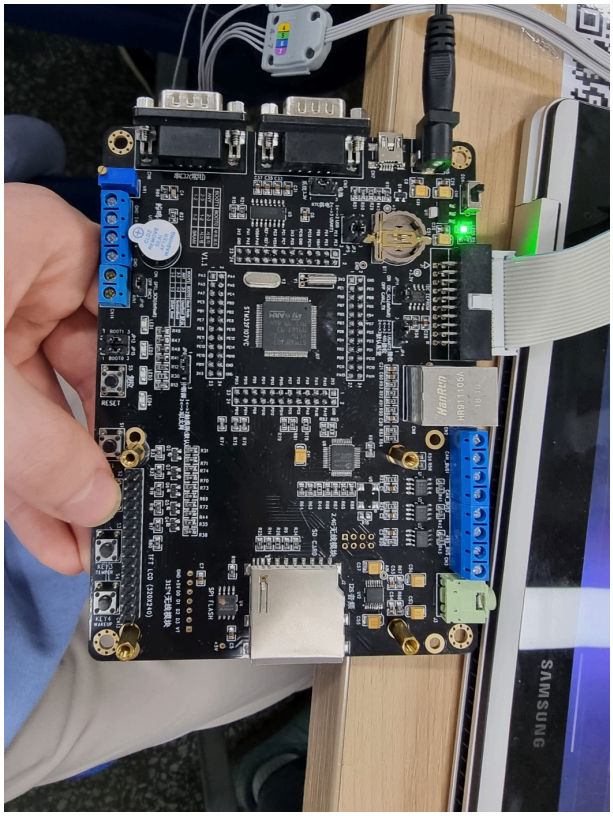
LED가 연결된 포트 D에 BSRR과 BRR의 오프셋 0x10, 0x14를 각각 더해 GPIOx_BSRR, GPIOx_BRR 레지스터의 주소를 얻는다. BSRR 레지스터로 set하여 LED를 켜고, BRR 레지스터로 reset하여 LED를 끈다. 예를 들어 PD2와 연결된 LED를 켜려면 0b100, 즉 0x4를 GPIOD_BSRR 레지스터에 넣어 주면 되고, 끄려면 동일한 값을 GPIO_BRR 레지스터에 넣어 주면 된다.

```
while (1)
{
    if (!(GPIOC_IDR & (1 << 4)))  // KEY1(PC4)을 눌렀을 때
    {
        GPIOD_BSRR |= 0x04; // LED1(PD2) ON
        GPIOD_BSRR |= 0x08; // LED2(PD3) ON
    }

    if (!(GPIOB_IDR & (1 << 10)))  // KEY2(PB10)을 눌렀을 때
    {
        GPIOD_BRR |= 0x04; // LED1(PD2) OFF
        GPIOD_BRR |= 0x08; // LED2(PD3) OFF
    }

    if (!(GPIOC_IDR & (1 << 13)))  // KEY3(PC13)을 눌렀을 때
    {
        GPIOD_BSRR |= 0x10; // LED3(PD4) ON
        GPIOD_BSRR |= 0x80; // LED4(PD7) ON
    }

    if (!(GPIOA_IDR & (1 << 0)))  // KEY4(PA0)을 눌렀을 때
    {
        GPIOD_BRR |= 0x10; // LED3(PD4) OFF
        GPIOD_BRR |= 0x80; // LED4(PD7) OFF
    }
}
```

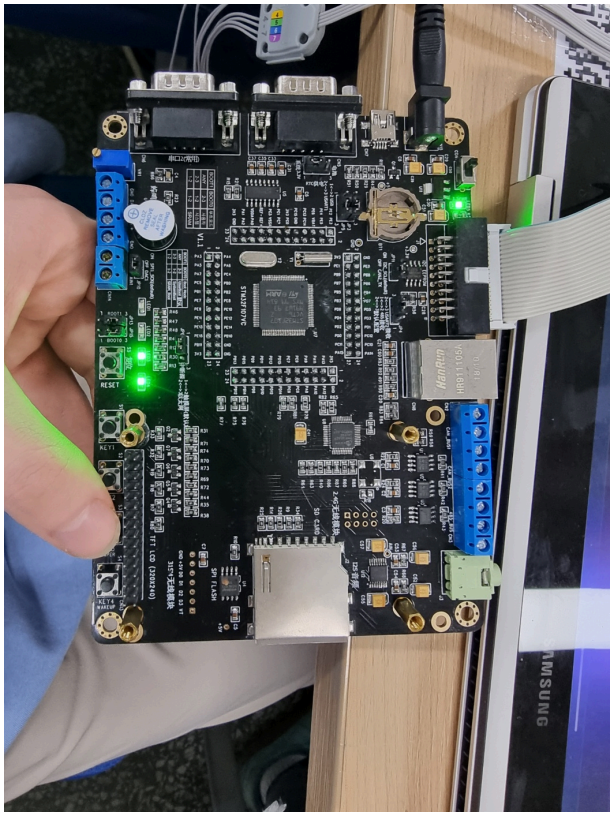
## 실험 결과



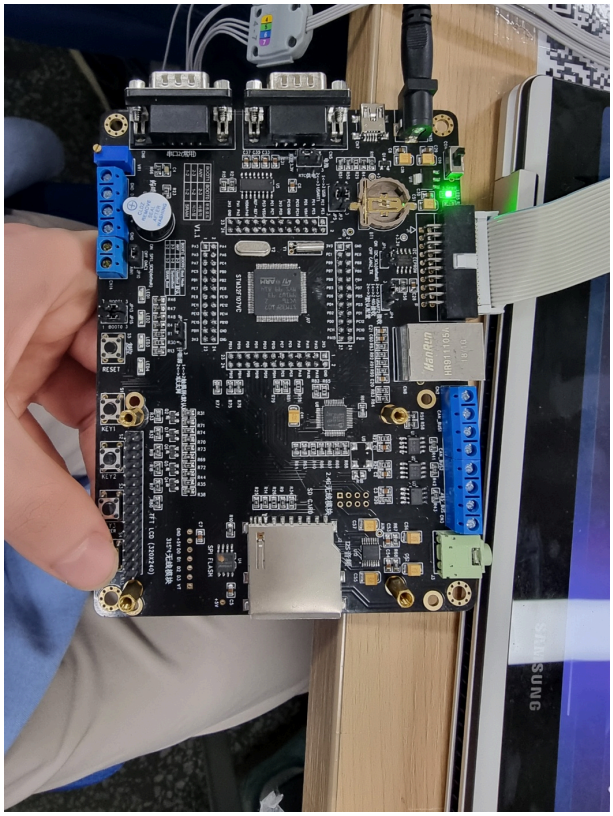
1. KEY1 눌렀을 때 LED1, 2 On



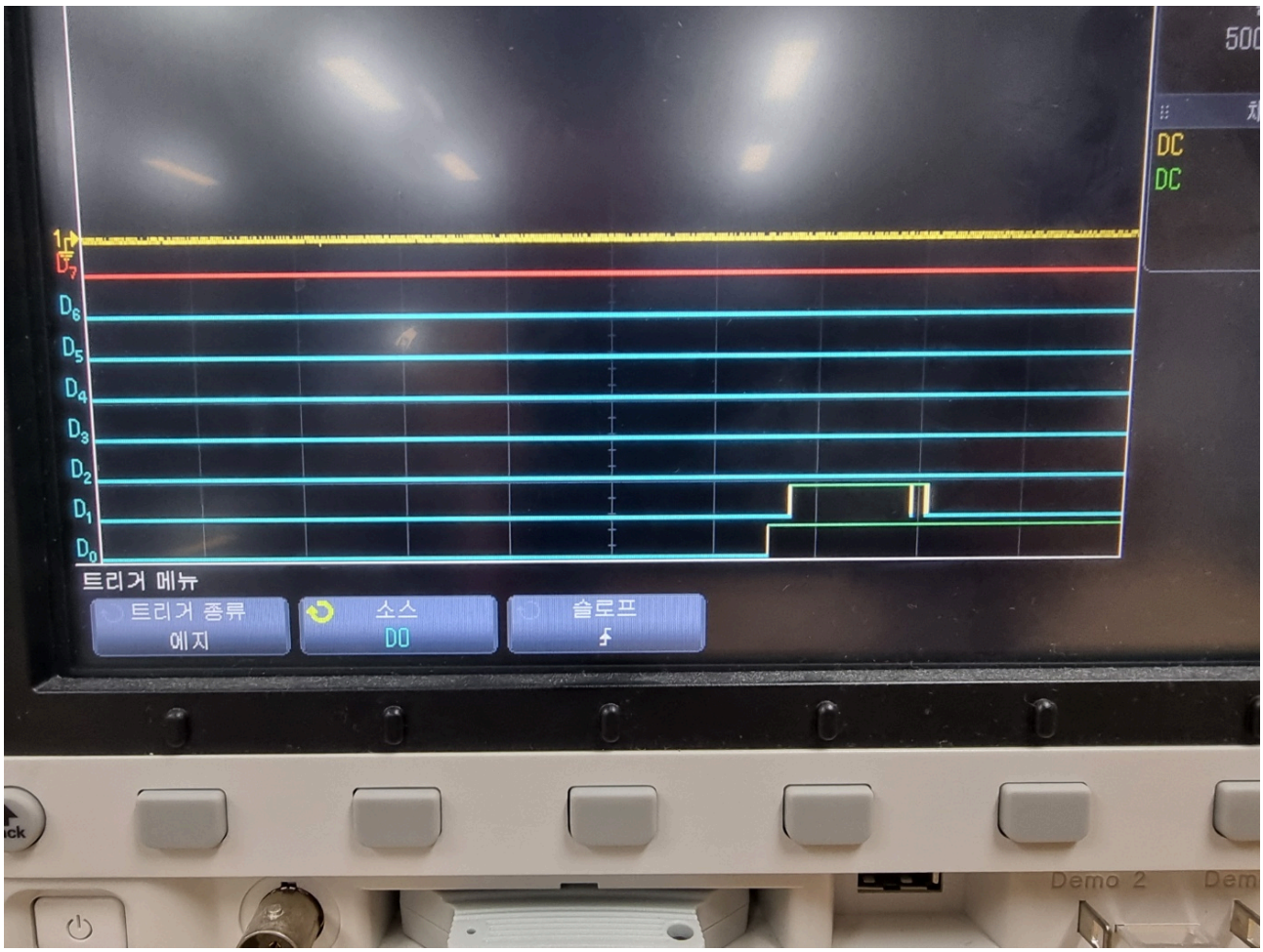
2. KEY2 눌렀을 때 LED1, 2 Off



3. KEY3 눌렀을 때 LED3, 4 On



4. KEY4 눌렀을 때 LED3, 4 Off




KEY1, LED1을 각각 오실로스코프의 D0, D1 핀에 연결하여

KEY1을 눌렀을 때 LED1이 켜지기까지 딜레이가 발생하는 것 확인