Busan Software Meister High School

MICROPROCESSOR

2309 양유빈

20230316

마이크로프로세서

손정웅선생님

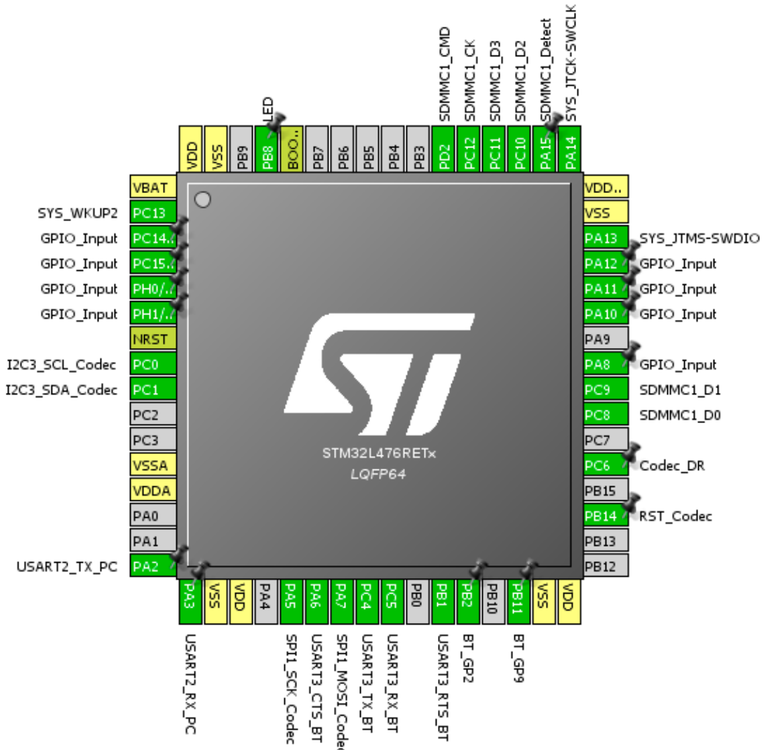
OVERVIEW

레지스터를 활용한 제어

- GPIO
- 참고) 슈미트 트리거
- 레지스터 직접 제어
- STM32 LED

GPIO

GPIO //general-purpose input/output

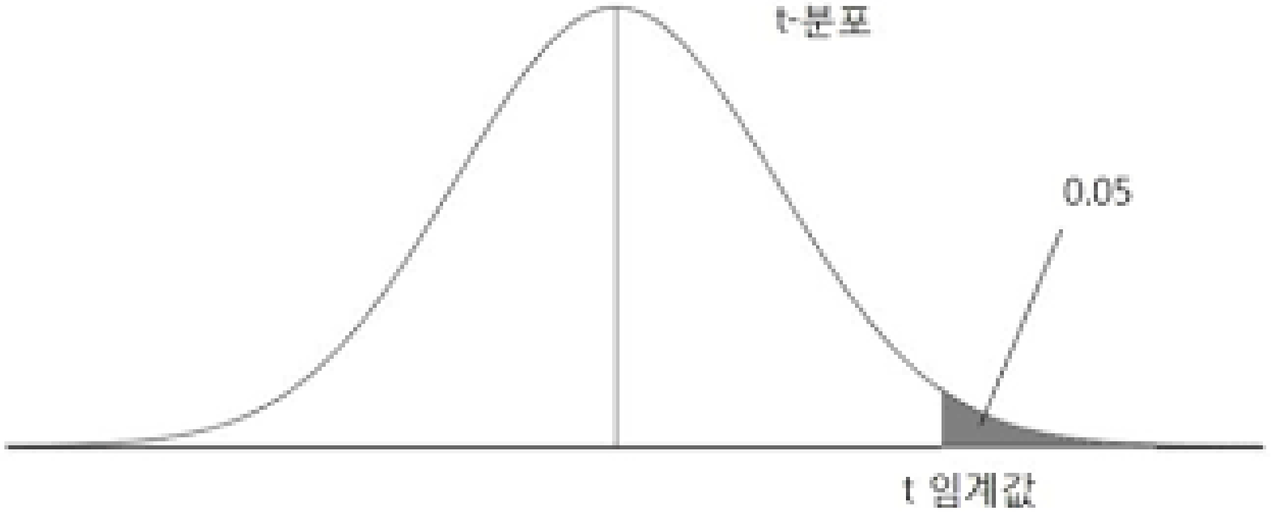


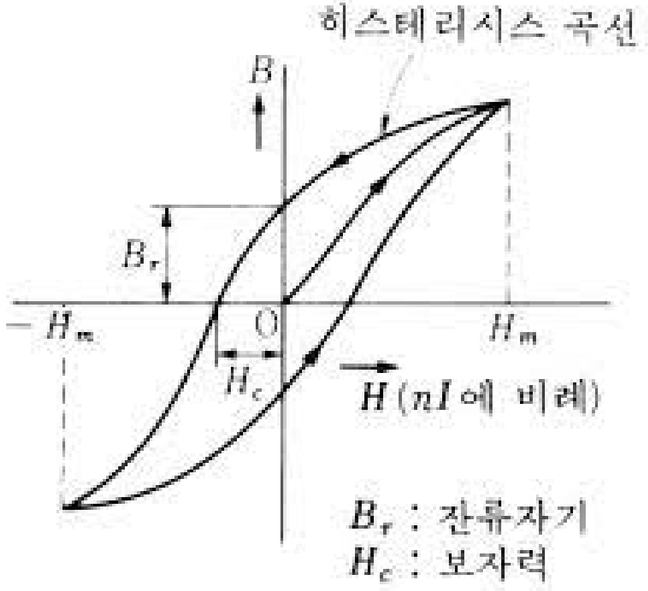
- GPIO는 특정 목적이 미리 정의되지 않은 입출력(다용도).
- 입력이나 출력을 포함한 동작이 런타임시 사용자에 의해 제어 될 수 있는 디지털 신호핀.
- STM32F103RB의 GPIO는 16비트 입출력 포트 A, B, C, D로 구성.
- 4가지 모드 지원.
- 입력모드는 플로팅, 풀업, 풀다운으로 설정 가능.
- 출력 모드는 푸쉬풀, 오픈드레인으로 설정 가능.
- 출력모드 신호의 최대속도는 저속, 중속, 고속 중 설정 가능.
- 출력모드에서 비트 단위로 출력을 세트 또는 리셋 가능.
- GPIO의 모든 단자는 외부인터럽트 신호선으로 사용 가능(입력이 플로팅 상태 일 때).

참고) 슈미트 트리거

Reference) Schmitt trigger

- 임계점: 회로내에서 논리 상태의 변화가 1에서 0으로 또는 0으로부터 다른 값 등으로 바뀌는 신호 기준.
- 슈미트 트리거는 히스테리시스 특성을 이용한 것으로 입력진폭이 일정 값을 넘으면 급격히 작동하여 거의 일정한 출력을 얻고, 일정 값 이하가 되면 즉시 복구하는 동작을 하는 회로.
- 히스테리시스 특성을 이용하면 특정상 이상에서 high가 되면 특정값 이하로 떨어지기 전에는 high 유지 가능.
- 즉, 2진수에서 1에서 0으로 또는 0에서 1로 신호가 변할 때, 잡음에 의해 1인지 0인지를 판별할 수 없을 때 사용.





레지스터 직접 제어

register direct control

- 레지스터 직접 제어
  - 주변 장치를 제어하기 위해 주변장치 관련 레지스터에 직접값을 입력하는 방식
  - 코드 크기를 줄일 수 있으며 효율적으로 코드 구성 가능
- 소프트웨어 구동 방식
  - 제조사에서 제공하는 코드를 이용하여 프로그램을 작성하는 방식
- 내장 LED 점멸하기
  - 포트 확인: PA5
  - 버스 확인: APB2
  - 입출력 선택: 출력
  - 값 선택: ON(1), OFF(0)
- RCC_APB2ENR
  - 초기 값: 0x00000000
  - 세팅 값: 0x00000004

6.3.14 RCC APB2 peripheral clock enable register (RCC_APB2ENR)

Address offset: 0x44

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								SAI2 EN	SAI1 EN				TIM11 EN	TIM10 EN	TIM9 EN
								rw	rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SYSCFG EN	SPI4 EN	SPI1 EN	SDIO EN	ADC3 EN	ADC2 EN	ADC1 EN			USART6 EN	USART1 EN			TIM8 EN	TIM1 EN
	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw

STM32 LED

STM32 LED

// main.c에서 코드 수정

LED_0406.ioc | *main.c | main.h

```c
#include "main.h"

void Delay_Timer(uint32_t time) {
    for(;time>0;time--){
    }
}
int main(void) {
    RCC->APB2ENR |= 0x4; //PORTA 버스 탑승권 허용
    GPIOA->CRL &= ~0x400000;
    GPIOA->CRL |= 0x100000; //PORTA PA5 출력 설정

    while(1) {
        GPIOA->ODR |= 0x20; //LED ON
        Delay_Timer(100000);
        GPIOA->ODR &= ~0x20; //LED OFF
        Delay_Timer(100000);
    }
}
```

STM32 LED

STM32 LED

// main 함수의 비트연산을 shift 연산을 사용하여 프로그래밍

```
21 // main 함수의 비트연산을 shift 연산을 사용하여 프로그래밍
22 #include "main.h"
23
24 void Delay_Timer(uint32_t time) {
25     for(;time>0;time--){
26     }
27 }
28 int main(void) {
29     RCC->APB2ENR |= (1<<2); //0x4; PORT A 버스 탑승권 허용
30     GPIOA->CRL &= ~(1<<22); //~0x400000;
31     GPIOA->CRL |= (1<<20); //0x100000; PORT A PA5 출력 설정
32
33     while(1) {
34         GPIOA->ODR |= (1<<5); //0x20; LED ON
35         Delay_Timer(100000);
36         GPIOA->ODR &= ~(1<<5); //~0x20; LED OFF
37         Delay_Timer(100000);
38     }
39 }
```

STM32 LED

STM32 LED

// BSRR

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

```c
#include "main.h"

void Delay_Timer(uint32_t time) {
    for(;time>0;time--){
    }
}
int main(void) {
    RCC->APB2ENR |= (1<<2); //0x4; PORT A 버스 탑승권 허용
    GPIOA->CRL &= ~(1<<22); //~0x400000;
    GPIOA->CRL |= (1<<20); //0x100000; PORT A PA5 출력 설정

    while(1) {
        GPIOA->BSRR |= (1<<5); //LED ON
        Delay_Timer(100000);
        GPIOA->BSRR |= (1<<21); //~0x20; LED OFF
        Delay_Timer(500000);
    }
}
```