

다기능 썬用微信

아름다운 썬微信

리모콘 유니콘

:강영수, 박동현, 홍영은



목차

1. 구현목표
2. 개발환경과 모듈
3. 모듈 소개
 - 바람세기
 - 타이머
 - RGB LED
 - 리모컨
4. 동작결과
5. 결론
6. 질의응답

구현목록

버튼으로 기능 선택, led로 상태 표시

01

02

03

04

바람세기

1단, 2단, 3단, OFF

타이머

1시간, 3시간, 5시간, OFF

LED 조명

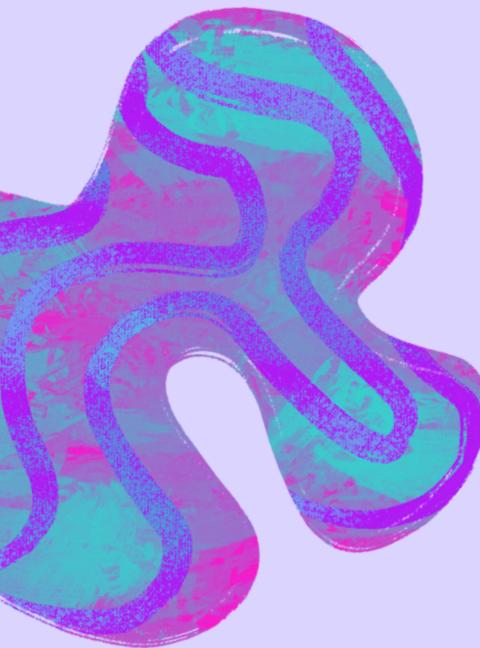
밝기 조절: 1단, 2단, 3단, OFF

리모컨

리모컨으로 제어

색 조절: RED, GREEN, BLUE,
MAGENTA, CYAN,
YELLOW, WHITE

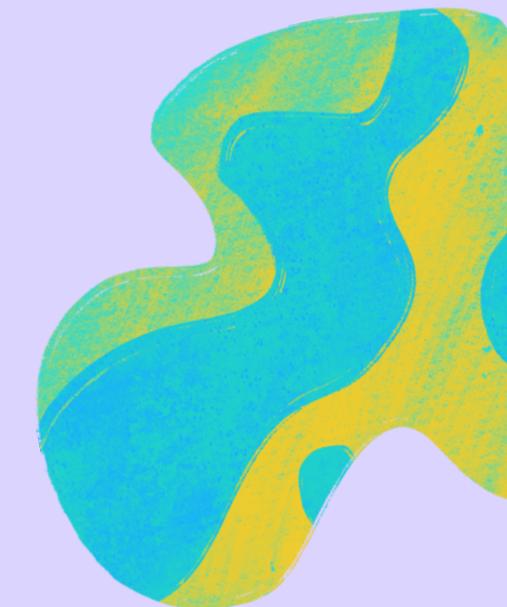
개별화한 멜라 모듈



리모컨



IR 리시브 센서



TOP 9

```
module remocon_unicorn_top(
    input clk, reset_p,
    input [2:0] btn,
    input ir_signal,
    inout dht11_data,
    output reg [5:0] led,
    output motor_pwm,
    output timer_end,
    output [2:0] rgb_led,
    output [3:0] module_led,
    output [3:0] com,
    output [7:0] seg_7
);
```

```
fan_speed_top fan_speed (
    .clk(clk),
    .reset_p(reset_p),
    .btn(btn[0]),
    .ir_signal(ir_signal),
    .timer_end(timer_end),
    .led(fan_speed_led),
    .motor_pwm(motor_pwm)
);
```

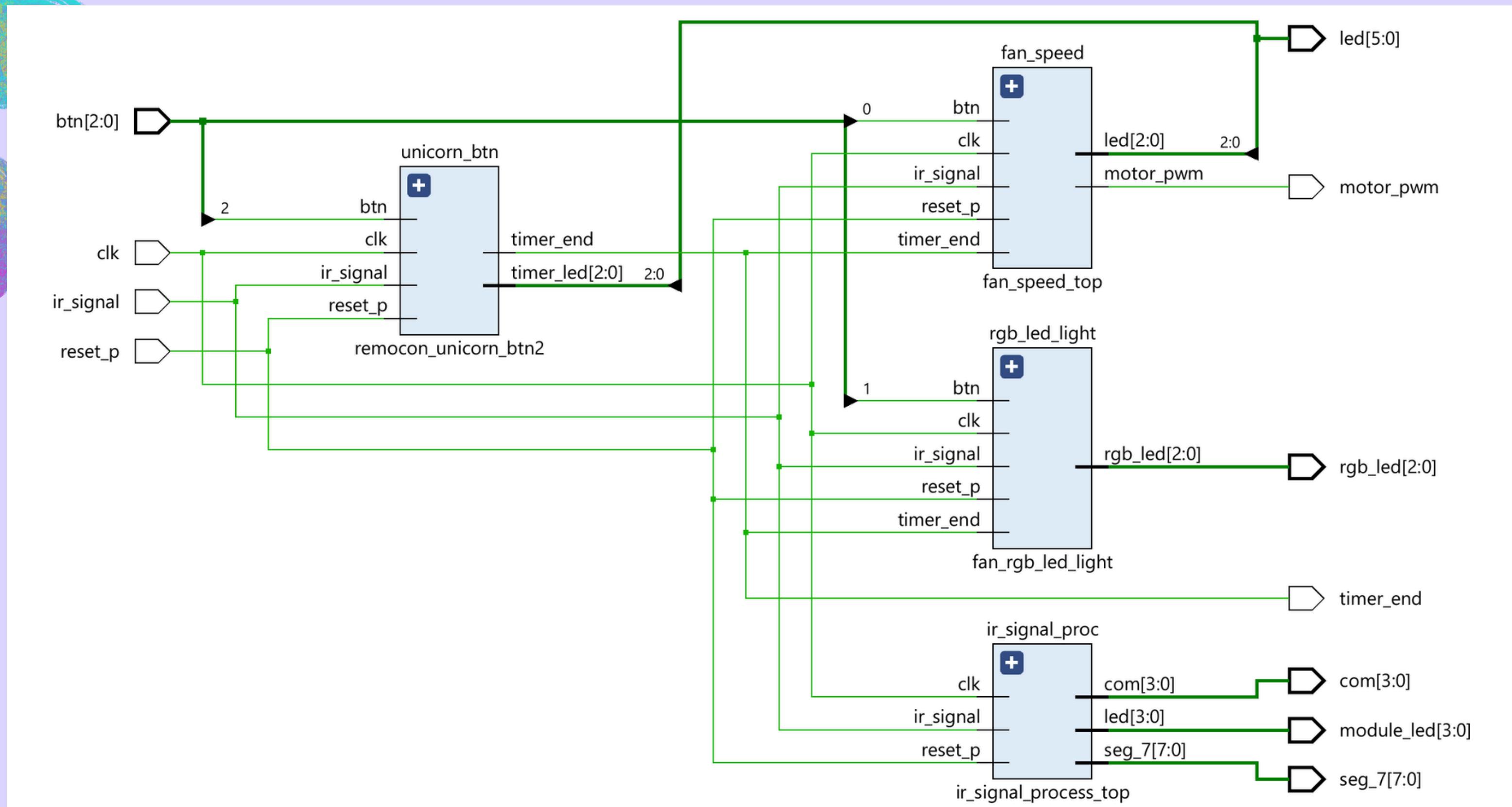
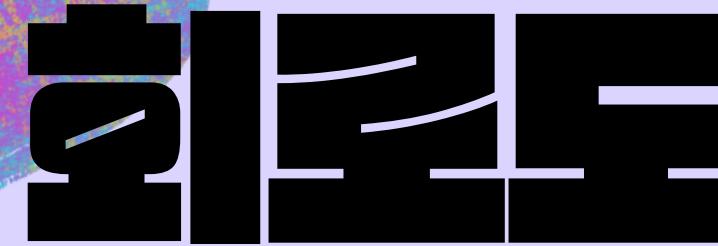
```
fan_rgb_led_light rgb_led_light (
    .clk(clk),
    .reset_p(reset_p),
    .btn(btn[1]),
    .timer_end(timer_end),
    .ir_signal(ir_signal),
    .rgb_led(rgb_led)
);
```

```
remocon_unicorn_btn2 unicorn_btn (
    .clk(clk),
    .reset_p(reset_p),
    .btn(btn[2]),
    .ir_signal(ir_signal),
    .timer_end(timer_end),
    .timer_led(timer_led)
);
```

```
fnd_4digit_cntr fnd_cntr(
    .clk(clk),
    .reset_p(reset_p),
    .value(value),
    .seg_7(seg_7),
    .com(com));
```

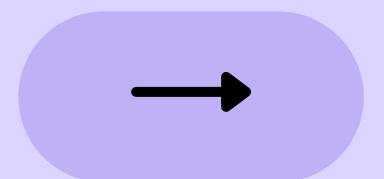
인증학적 설계

모듈 설계



바람쐬기 프로젝트

모듈

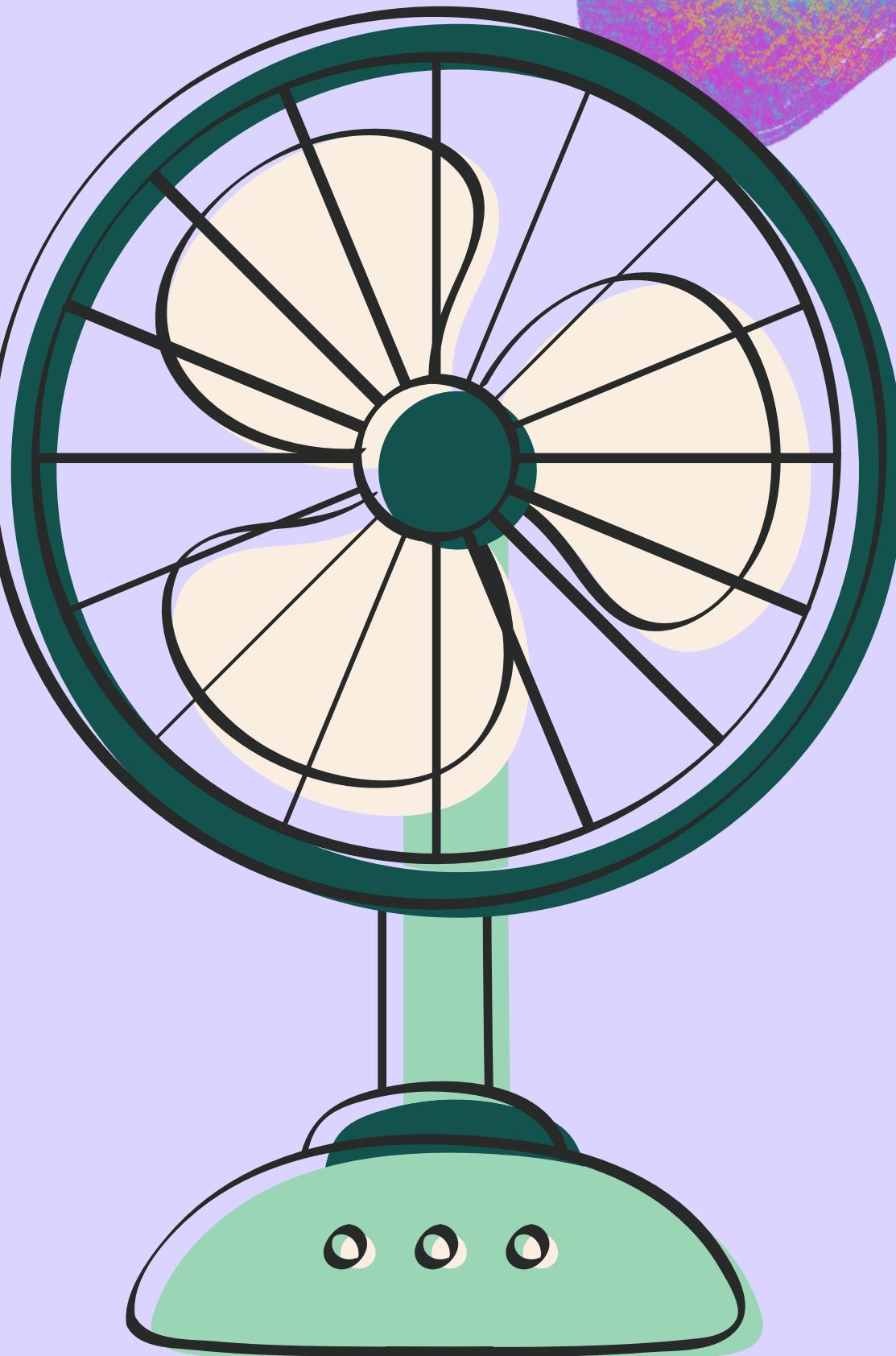


구현목표 : 1단, 2단, 3단, OFF

버튼 하나로 모드 변경

방법: state 머신

도면: 선풍기 연결하는 방법



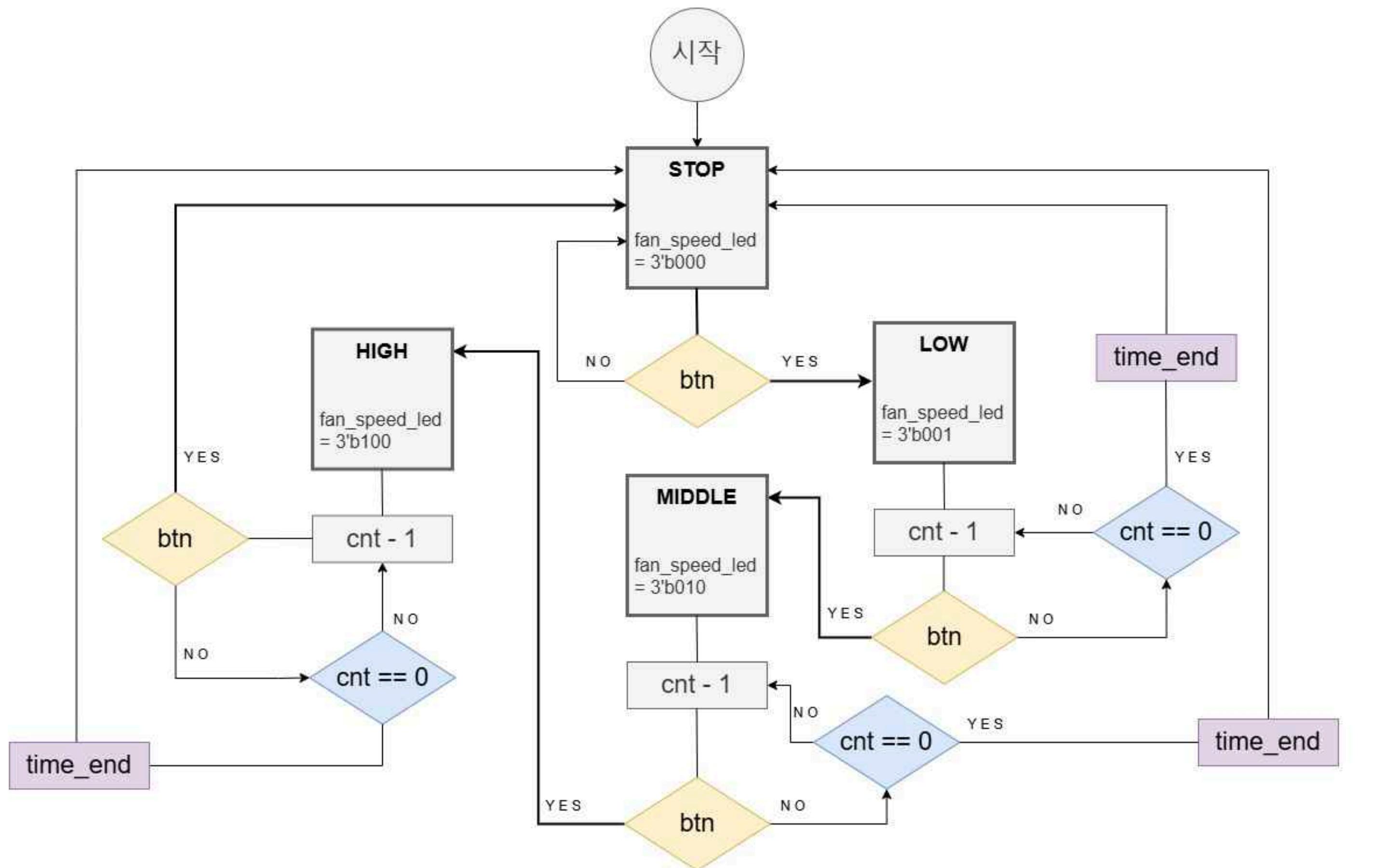
OFF->1단-> 2단->3단->OFF

```
parameter STOP = 4'b0001; // OFF 상태  
parameter LOW = 4'b0010; // 팬속도 1단  
parameter MIDDLE = 4'b0100; // 팬속도 2단  
parameter HIGH = 4'b1000; // 팬속도 3단
```



```
STOP: begin  
    duty = 7'd0;  
    led = 3'b000;  
end  
LOW: begin  
    duty = 7'd32;  
    led = 3'b001;  
end  
MIDDLE: begin  
    duty = 7'd64;  
    led = 3'b010;  
end  
HIGH: begin  
    duty = 7'd96;  
    led = 3'b100;  
end  
default: begin  
    duty = 7'd0;  
    led = 3'b000;  
end
```

State 쌍으로



모터 드라이버 주요 기능

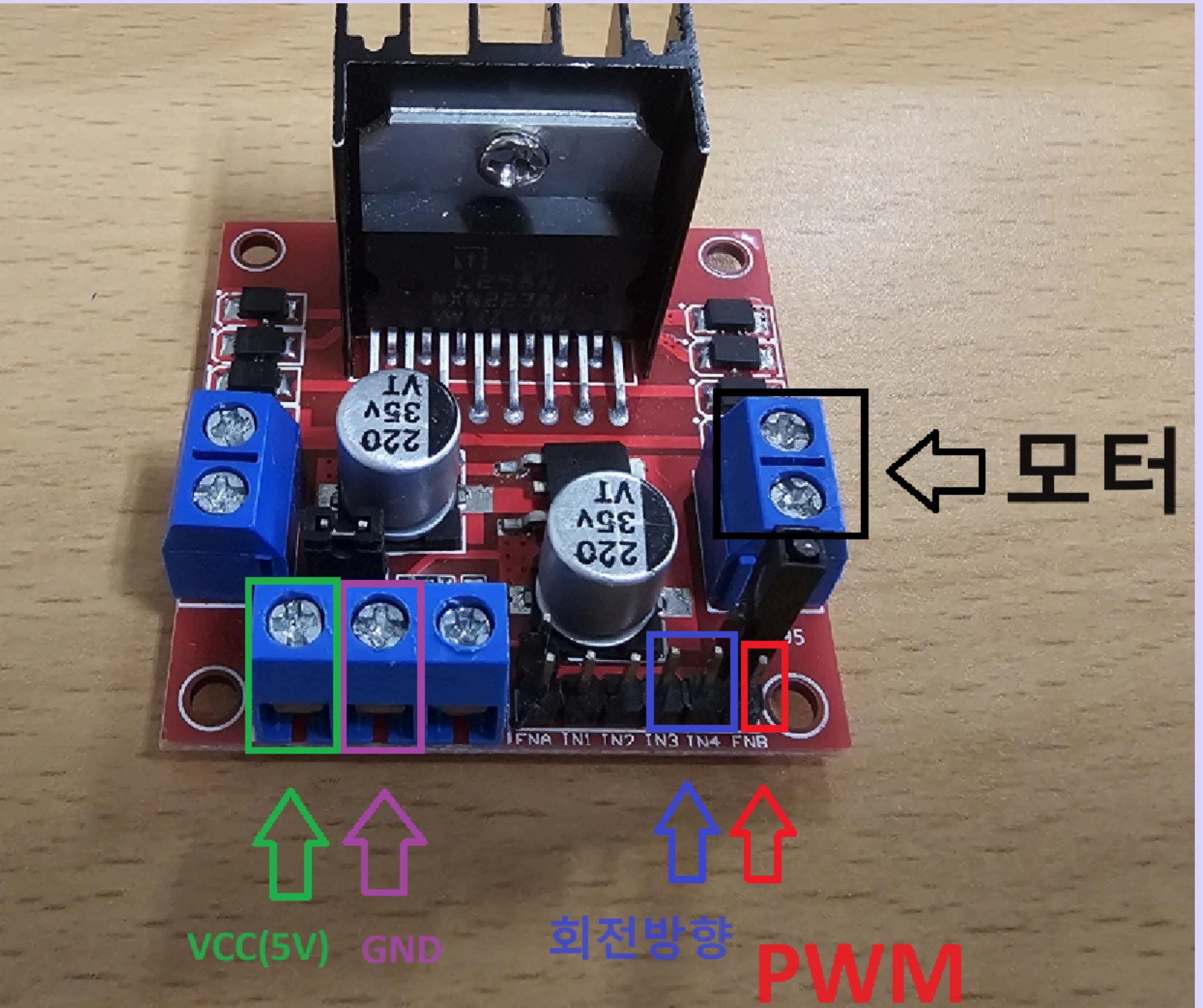
속도 제어: PWM(펄스 폭 변조) 신호를 사용하여
듀티 사이클을 변경하여 모터의 속도를 제어

방향 제어: H-브리지 회로를 사용하여
모터의 양방향 회전을 제어

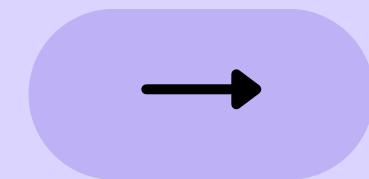
전류 보호: 과전류 보호 기능이 포함되어 있어
모터나 드라이버가 손상되지 않게 함

열 보호: 과열 보호 기능이 포함되어 있어
드라이버가 과열되면 동작을 멈춤

센서(PIR) 연결 방법



개념 타이머 모듈

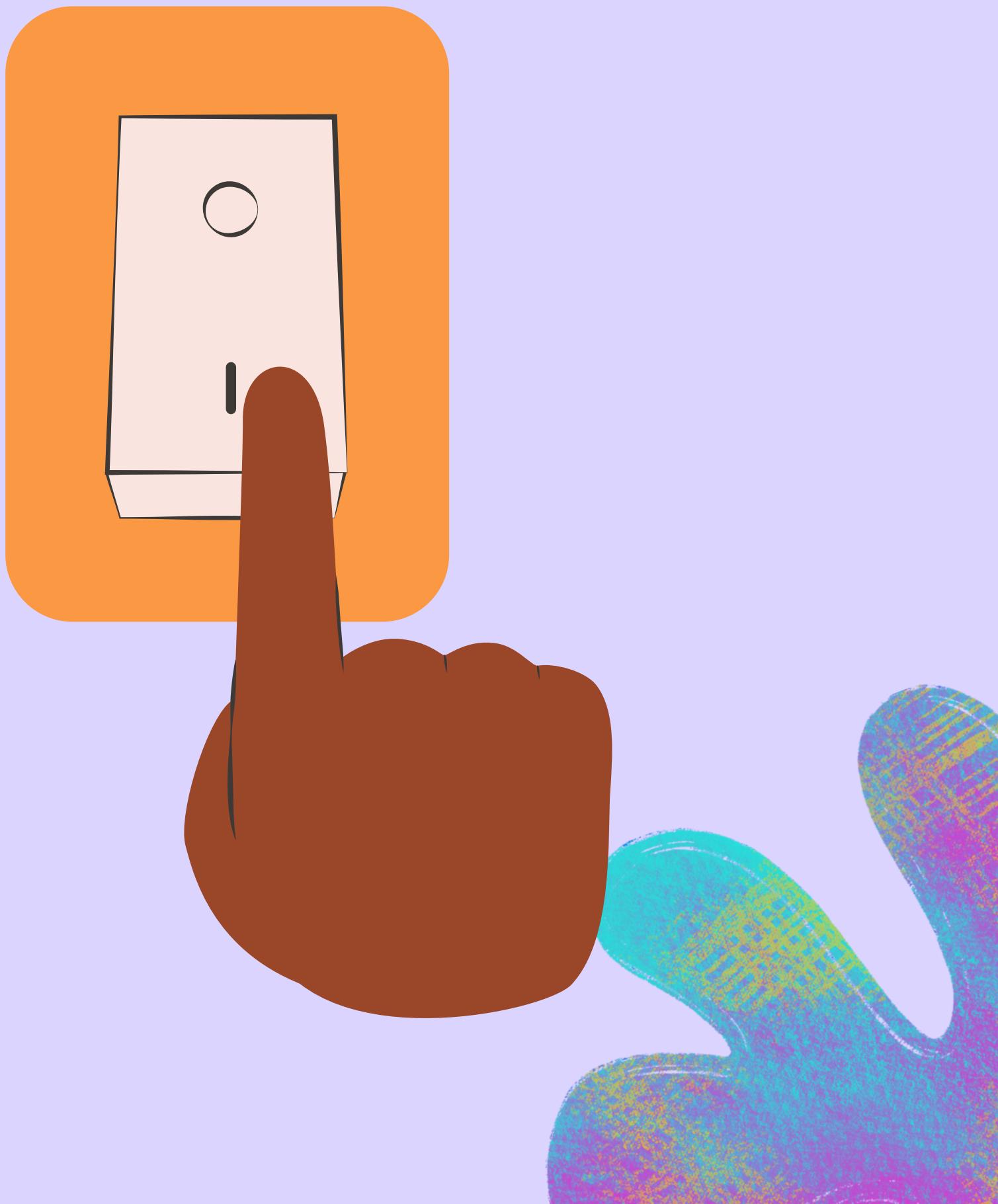


구현목표

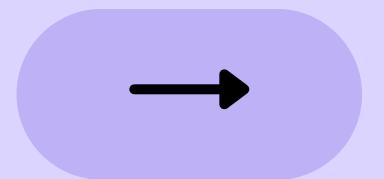
: 버튼 하나로 타이머 조절

- 1시간, 3시간, 5시간 3단 조절.

타이머 OFF시 선풍기와 LED에 신호 전달



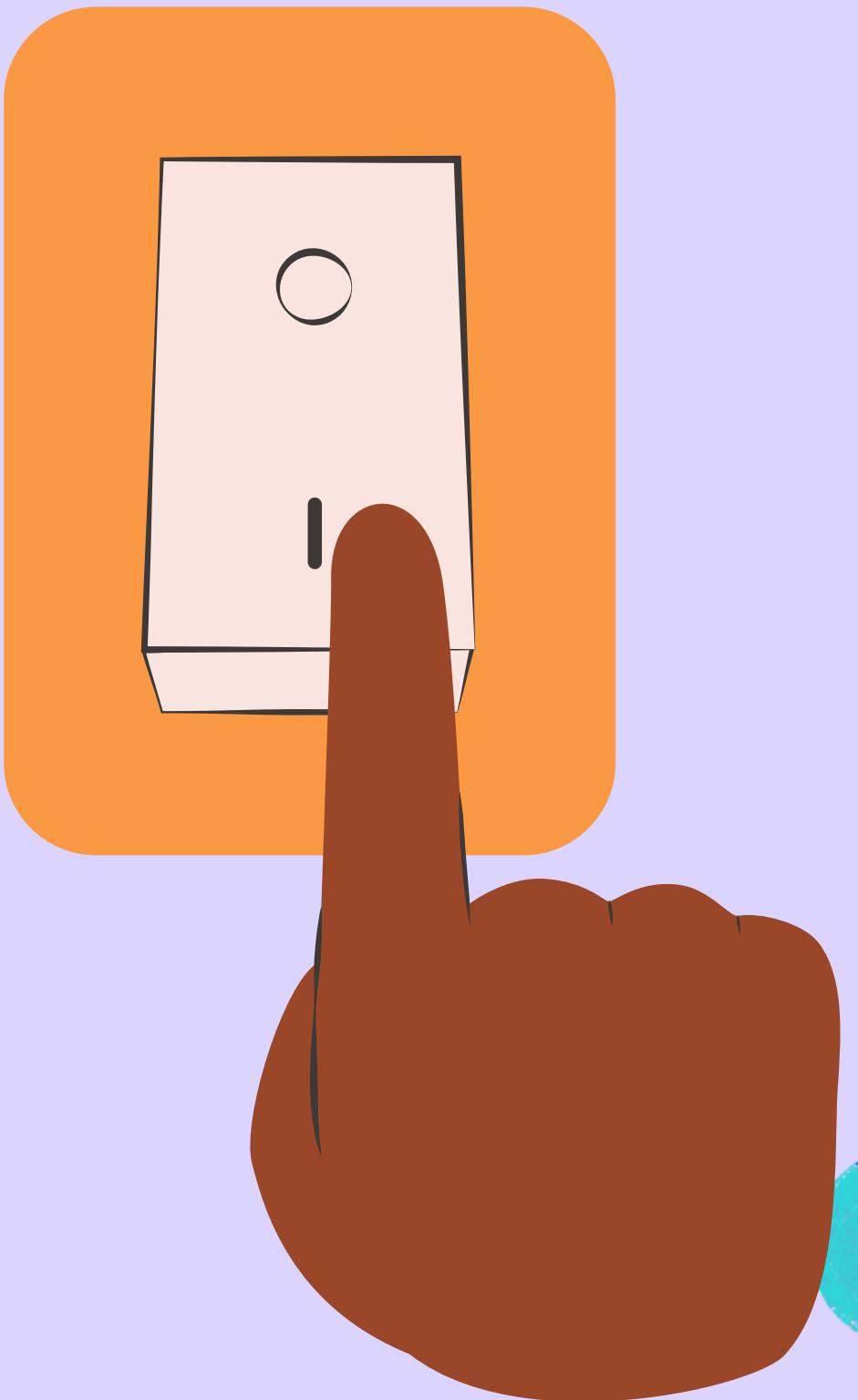
개념 타이머
모듈



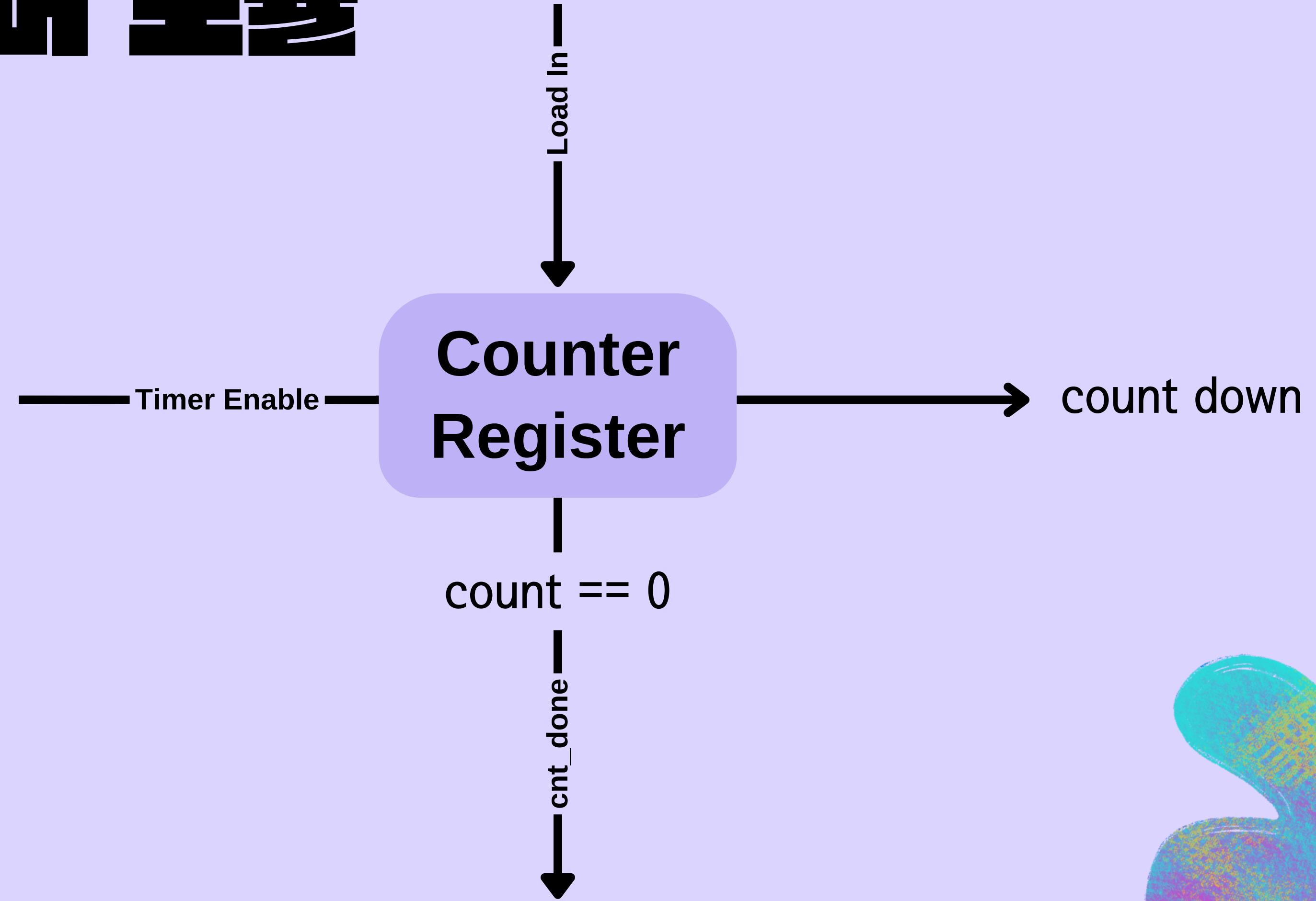
구현방법

: State 머신 사용

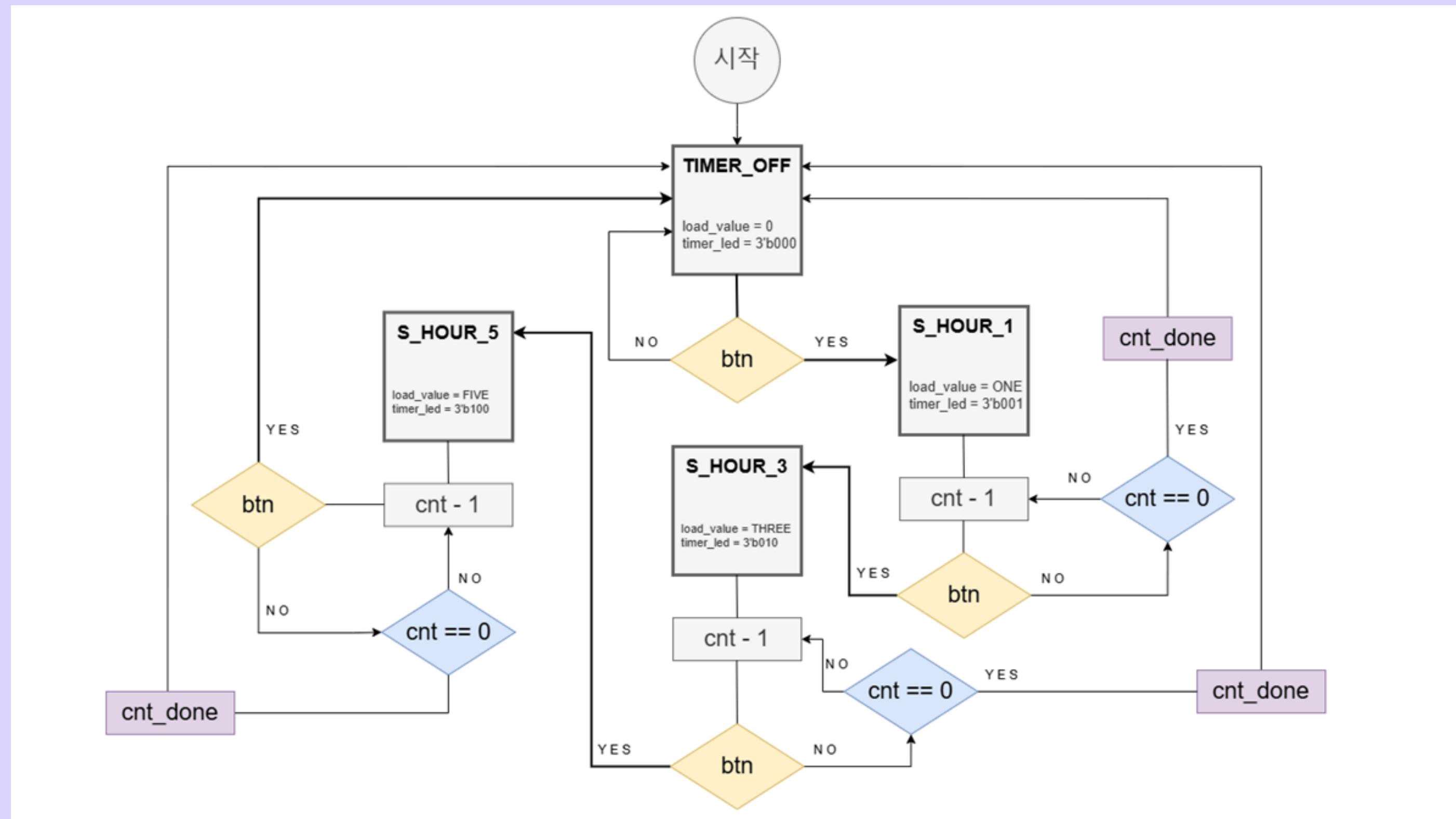
- 다운카운터로 구현



ECHOH 95



state



THE

ECHOH

SOUND

```
button_cntr timer_btn(
    .clk(clk),
    .reset_p(reset_p),
    .btn(btn),
    .btn_pedge(btn_pedge));
```

```
parameter TIMER_OFF      = 4'b0001; // 태이머 초기
parameter S_HOUR_1        = 4'b0010; // 1시간 태이머
parameter S_HOUR_3        = 4'b0100; // 3시간 태이머
parameter S_HOUR_5        = 4'b1000; // 5시간 태이머

parameter ONE_HOUR        = 1_000_000_000; // 1시간
parameter THREE_HOUR       = 2_000_000_000; // 3시간
parameter FIVE_HOUR        = 32'd3_000_000_000; // 5시간
```

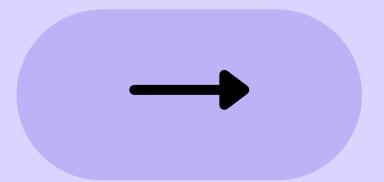
```
assign timer_end = cnt_done;
```

```
case(state)
    S_HOUR_1 : begin
        timer_en = 1;
        if(btn_pedge || remote_control_match_pulse)begin
            next_state = S_HOUR_3;
            load_en = 1;
            load_value = THREE_HOUR; //3시간
            timer_led = 3'b010;
        end
        else if (!cnt_done)begin
            load_en = 0;
            load_value = ONE_HOUR; //1시간
            timer_led = 3'b001;
        end
        else begin
            next_state = TIMER_OFF;
        end
    end
    S_HOUR_3 : begin
        timer_en = 1;
        if(btn_pedge || remote_control_match_pulse)begin
            next_state = S_HOUR_5;
            load_en = 1;
            load_value = FIVE_HOUR; // 5시간
            timer_led = 3'b100;
        end
        else if(!cnt_done)begin
            load_en = 0;
            load_value = THREE_HOUR;
            timer_led = 3'b010;
        end
        else begin
            next_state = TIMER_OFF;
        end
    end
    default : next_state = TIMER_OFF;
endcase
```

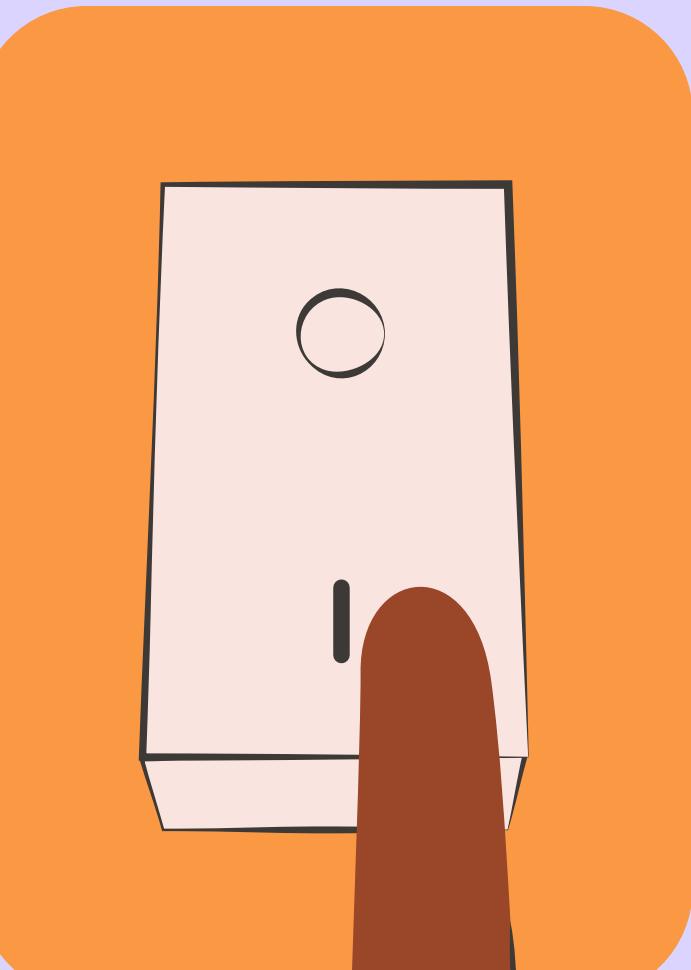
```
S_HOUR_5 : begin
    timer_en = 1;
    if(btn_pedge || remote_control_match_pulse)begin
        next_state = TIMER_OFF;
        load_en = 1;
        load_value = 0;
        timer_led = 3'b000;
    end
    else if(!cnt_done)begin
        load_en = 0;
        load_value = FIVE_HOUR;
        timer_led = 3'b100;
    end
    else begin
        next_state = TIMER_OFF;
    end
end
TIMER_OFF : begin
    timer_en = 0;
    if(btn_pedge || remote_control_match_pulse)begin
        next_state = S_HOUR_1;
        load_en = 1;
        load_value = ONE_HOUR;
        timer_led = 3'b001;
    end
    else if(!cnt_done)begin
        load_en = 0;
        load_value = 0;
        timer_led = 3'b000;
    end
    else begin
        next_state = TIMER_OFF;
    end
end
end
```

개념 EPOCH

도구



입출력 포트 및 레지스터 소개
state상태



LED 조명

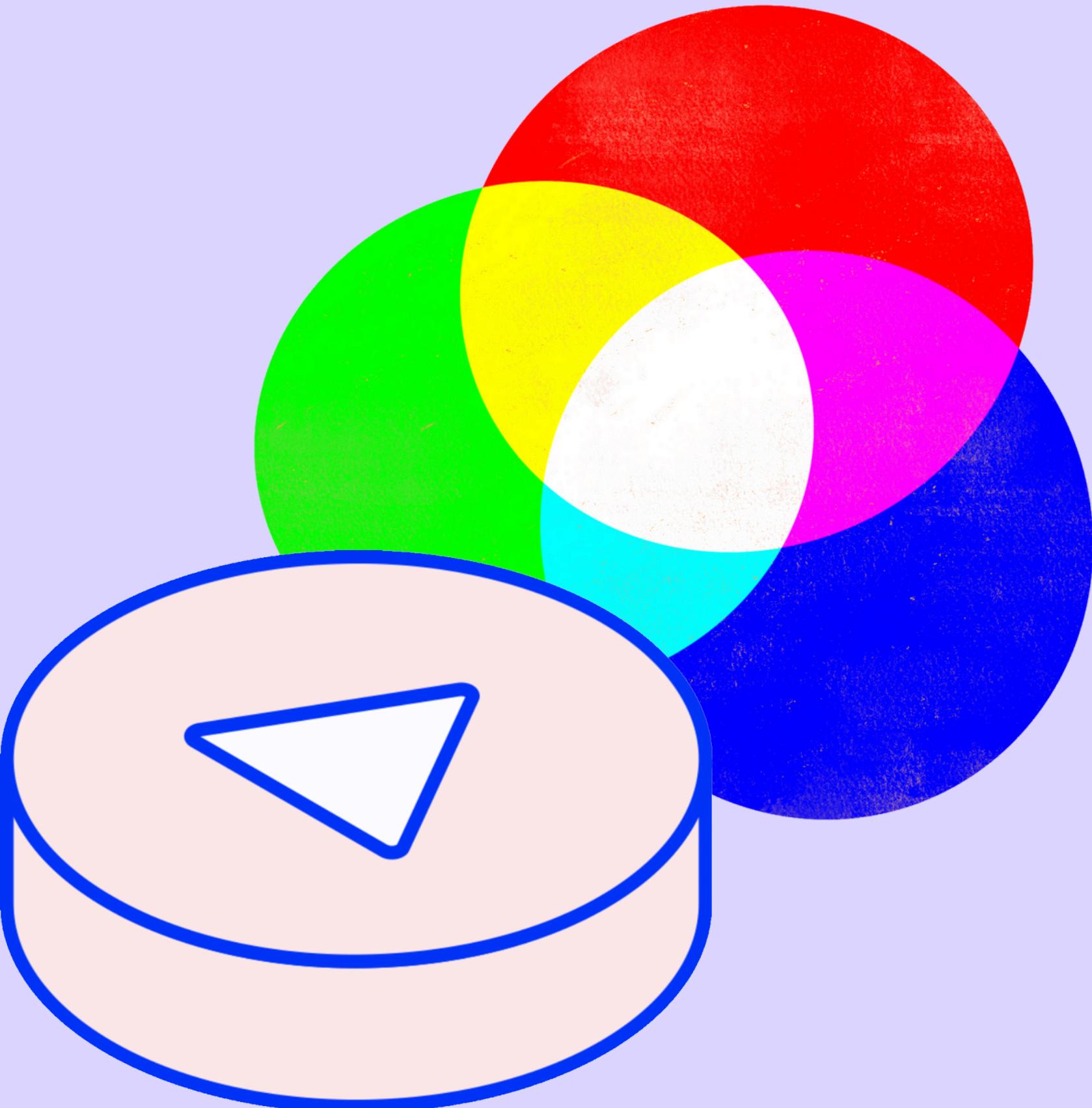
모듈



구현 목표:

버튼 하나로 RGB LED 색상 밝기 및 조절

- 짧게 누름 - 색상조절
- 길게 누름 - 밝기 조절



LED 조명

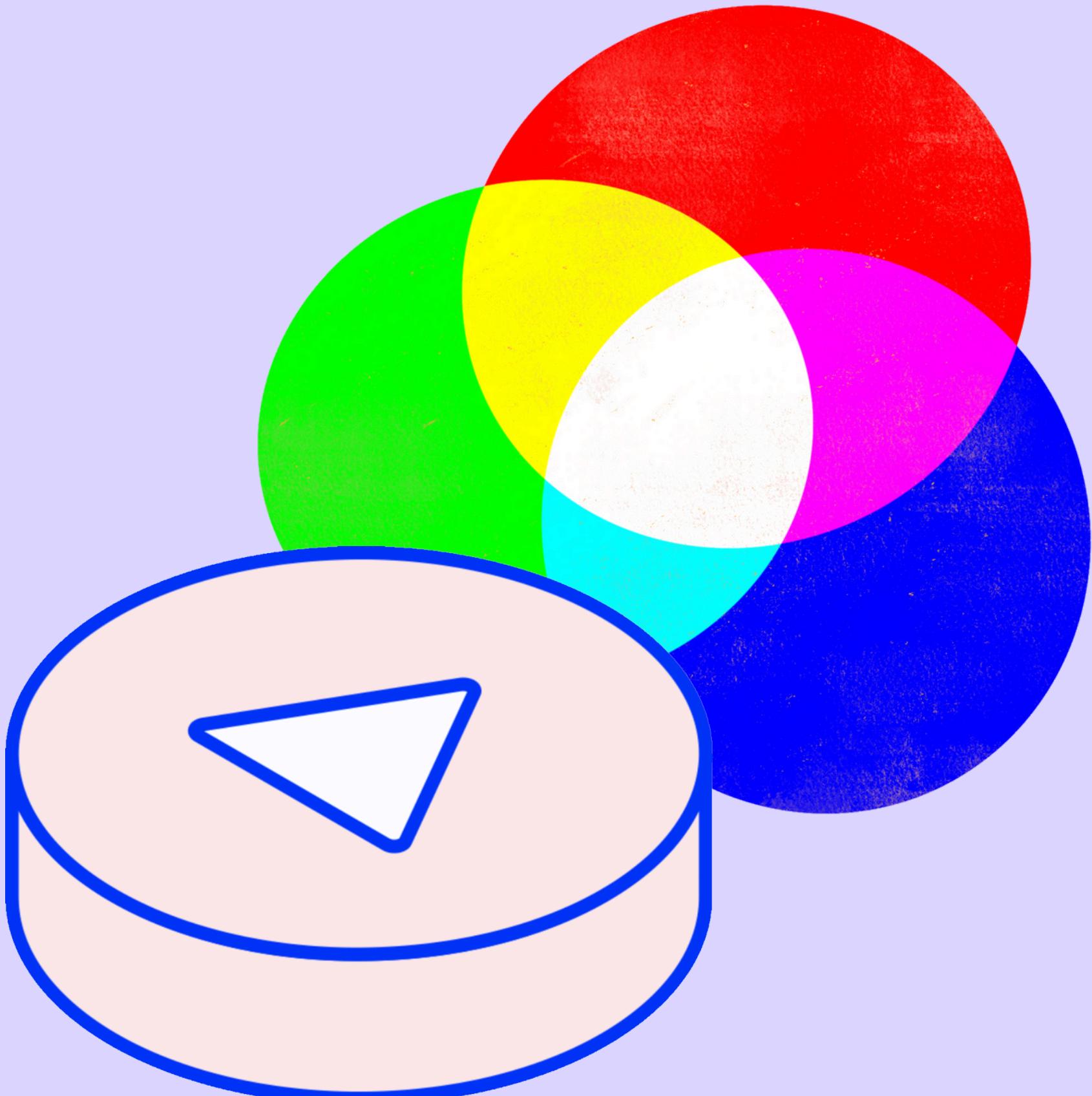
모듈



구현 방법:

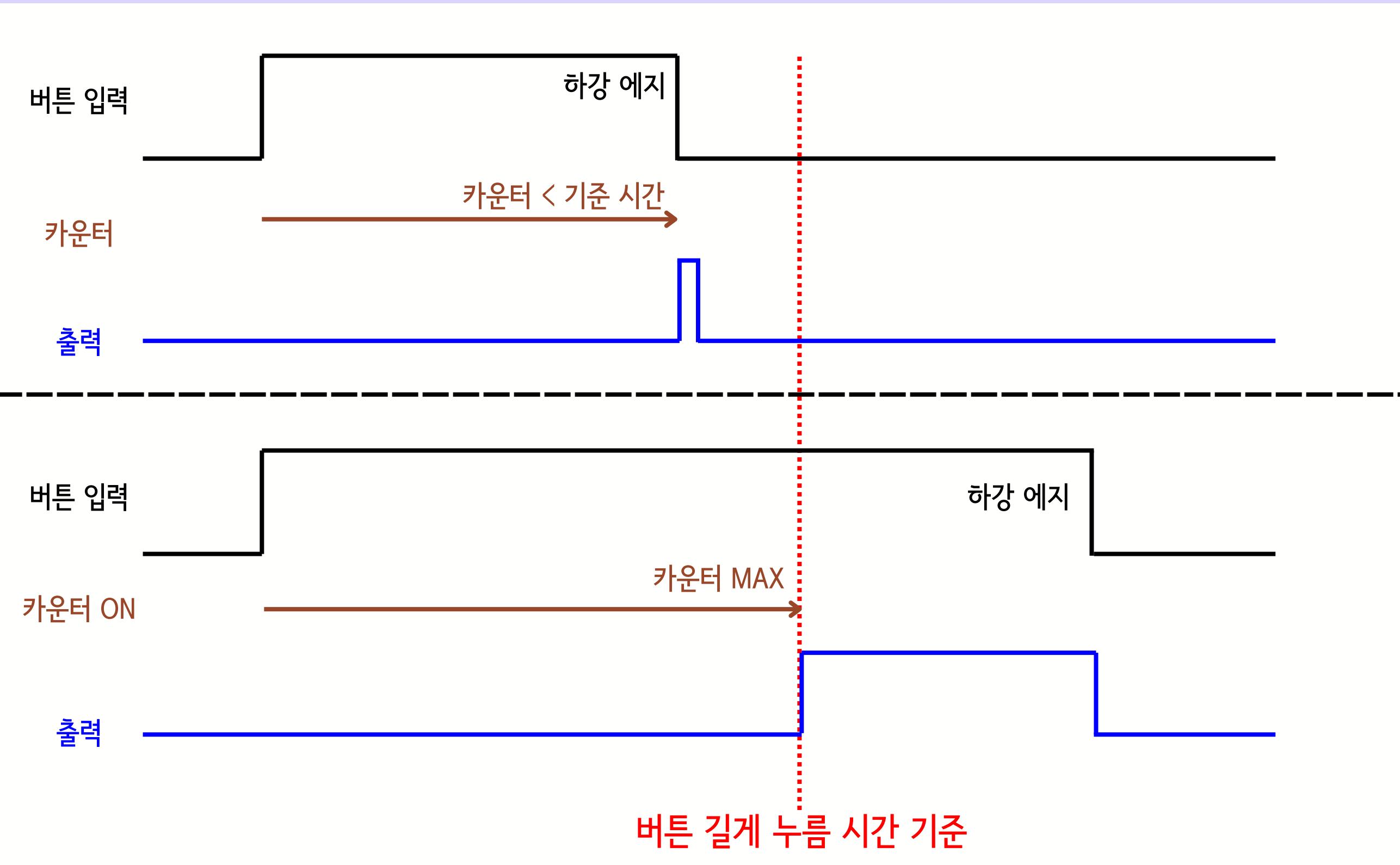
버튼 하나로 RGB LED 색상 밝기 및 조절

- 버튼 누름 시간 구분
- 레지스터 조작 - 색상 조절
- 레지스터 조작 - 밝기 조절
- LED로 PWM 출력



버튼 입력

버튼 누름시간 구분



LED 조명 모듈

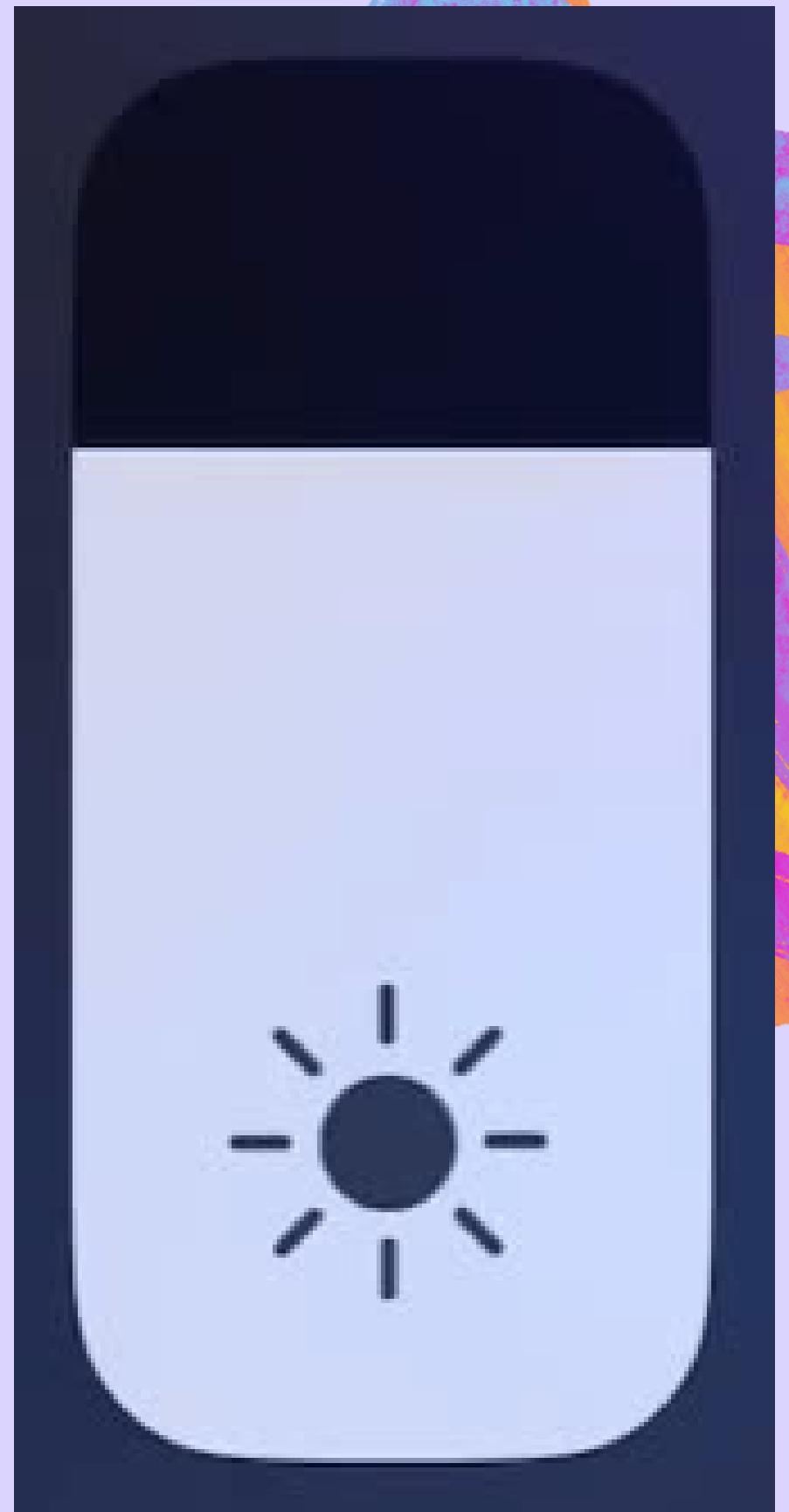


구현절차

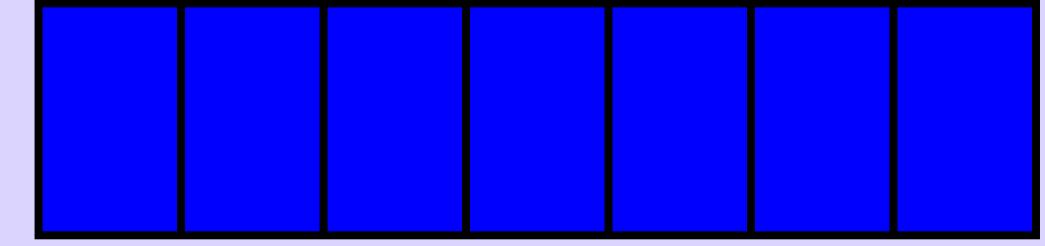
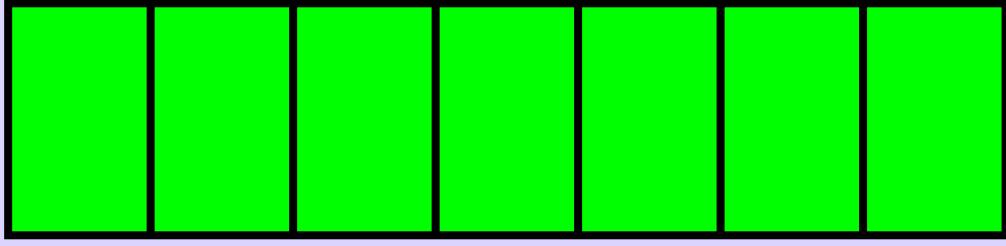
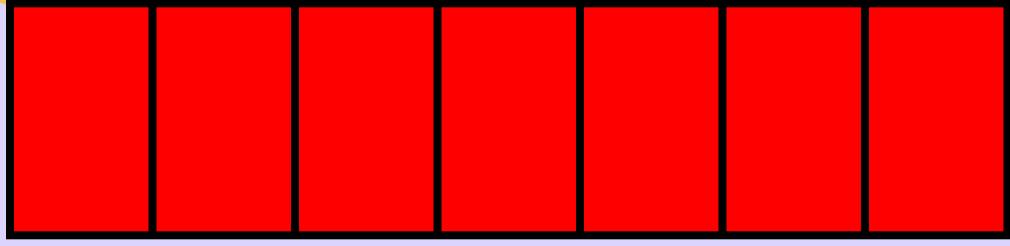
:LED 색상 및 밝기 구현

- PWM 스텝값으로 색상 및 밝기 조절 가능

알맞은 PWM 스텝값 레지스터 조작이 핵심



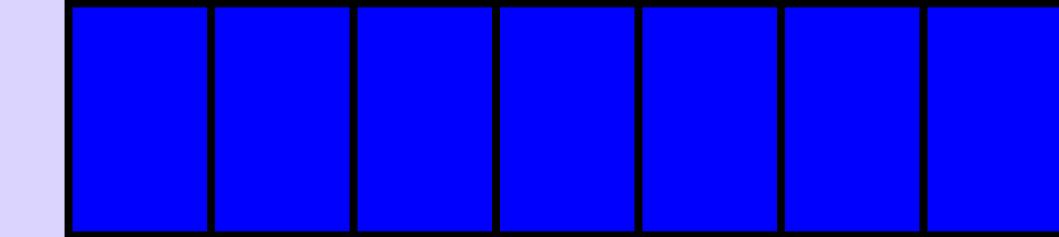
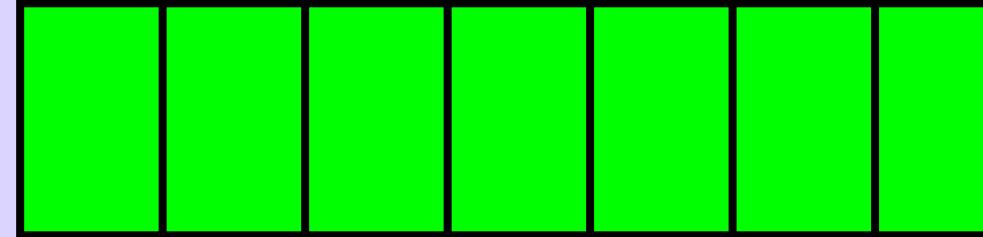
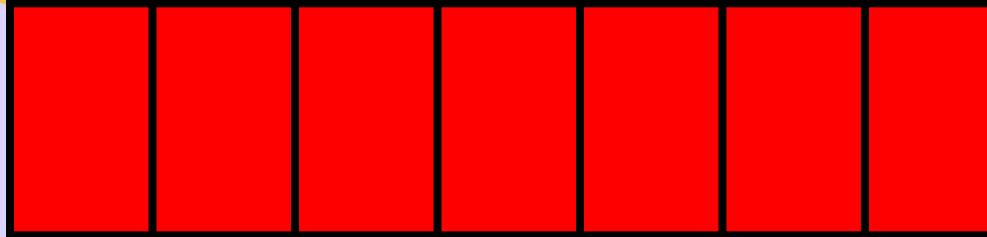
무한경차



PWM 스텝값 레지스터

구현경차

레지스터 조작 - 색상조절



	Hue	Hex	Rgb	Hsl
	0	#ff0000	rgb(255, 0, 0)	hsl(0, 100%, 50%)
	15	#ff4000	rgb(255, 64, 0)	hsl(15, 100%, 50%)
	30	#ff8000	rgb(255, 128, 0)	hsl(30, 100%, 50%)
	45	#ffbfff	rgb(255, 191, 0)	hsl(45, 100%, 50%)
	60	#ffff00	rgb(255, 255, 0)	hsl(60, 100%, 50%)
	75	#bfff00	rgb(191, 255, 0)	hsl(75, 100%, 50%)
	90	#80ff00	rgb(128, 255, 0)	hsl(90, 100%, 50%)
	105	#40ff00	rgb(64, 255, 0)	hsl(105, 100%, 50%)
	120	#0eff00	rgb(0, 255, 0)	hsl(120, 100%, 50%)
	135	#0eff40	rgb(0, 255, 64)	hsl(135, 100%, 50%)
	150	#0eff80	rgb(0, 255, 128)	hsl(150, 100%, 50%)
	165	#0effbf	rgb(0, 255, 191)	hsl(165, 100%, 50%)
	180	#0effff	rgb(0, 255, 255)	hsl(180, 100%, 50%)

	180	#0effff	rgb(0, 255, 255)	hsl(180, 100%, 50%)
	195	#00bfff	rgb(0, 191, 255)	hsl(195, 100%, 50%)
	210	#0080ff	rgb(0, 128, 255)	hsl(210, 100%, 50%)
	225	#0040ff	rgb(0, 64, 255)	hsl(225, 100%, 50%)
	240	#0000ff	rgb(0, 0, 255)	hsl(240, 100%, 50%)
	255	#4000ff	rgb(64, 0, 255)	hsl(255, 100%, 50%)
	270	#8000ff	rgb(128, 0, 255)	hsl(270, 100%, 50%)
	285	#bf00ff	rgb(191, 0, 255)	hsl(285, 100%, 50%)
	300	#ff00ff	rgb(255, 0, 255)	hsl(300, 100%, 50%)
	315	#ff00bf	rgb(255, 0, 191)	hsl(315, 100%, 50%)
	330	#ff0080	rgb(255, 0, 128)	hsl(330, 100%, 50%)
	345	#ff0040	rgb(255, 0, 64)	hsl(345, 100%, 50%)
	360	#ff0000	rgb(255, 0, 0)	hsl(0, 100%, 50%)

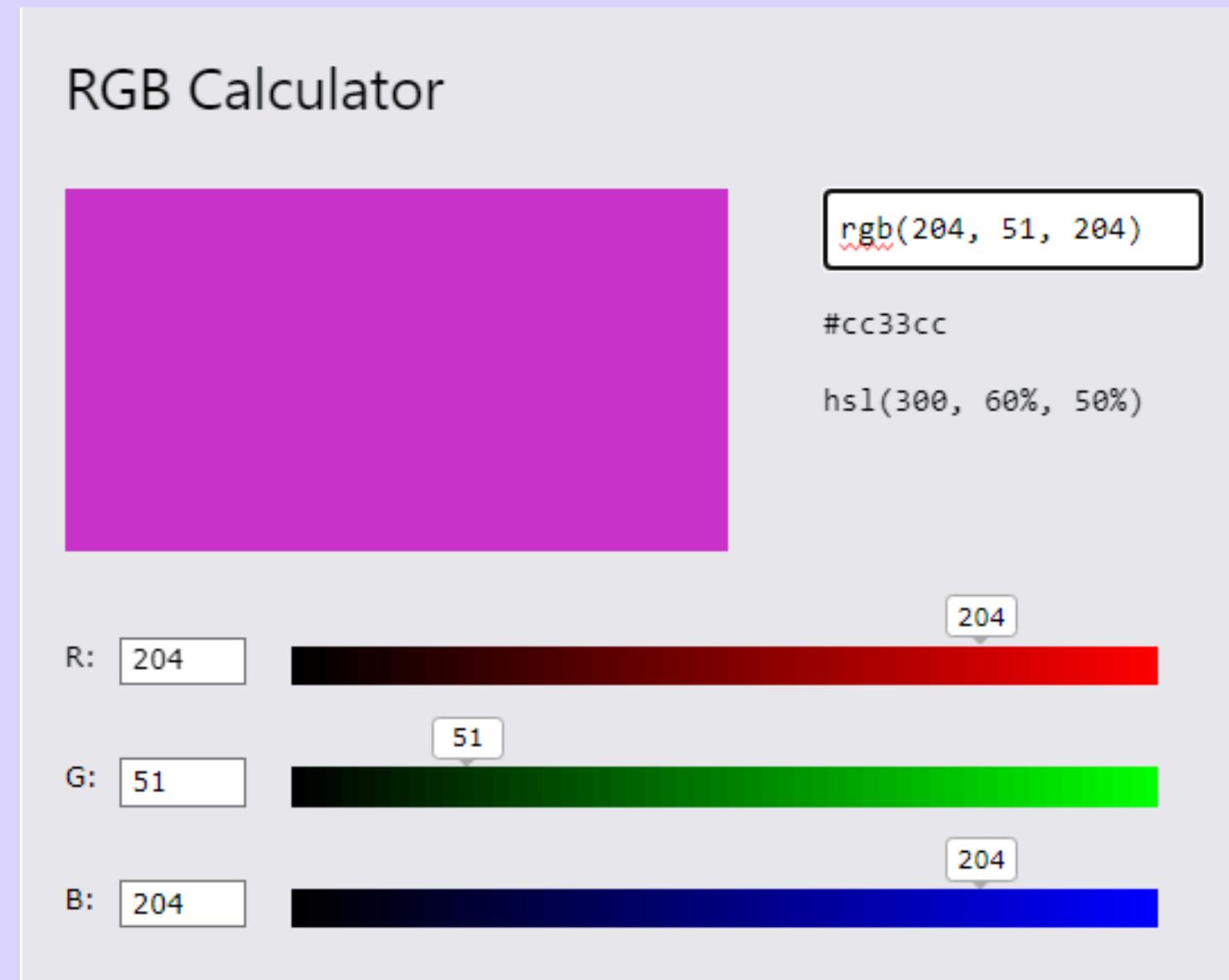
구현경차

레지스터 조작 - 색상조절

1 1 0 0 1 1 0

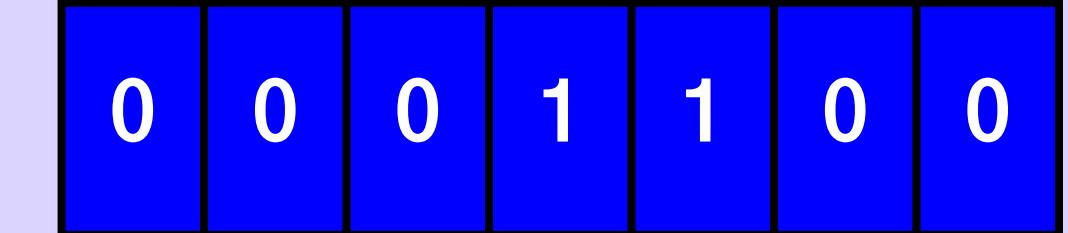
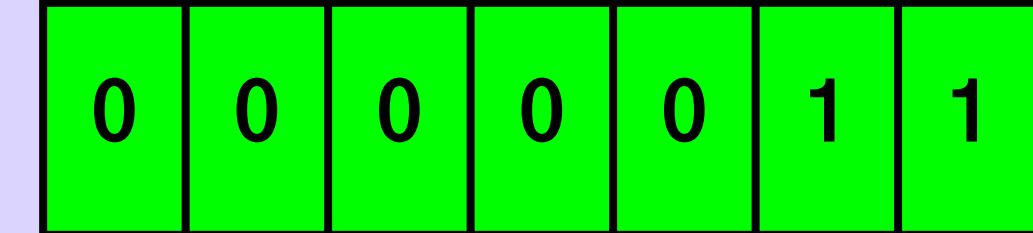
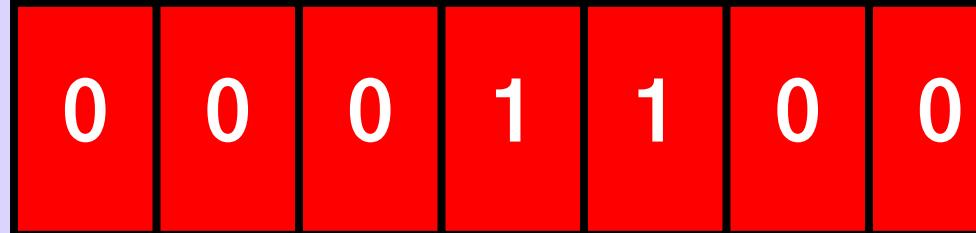
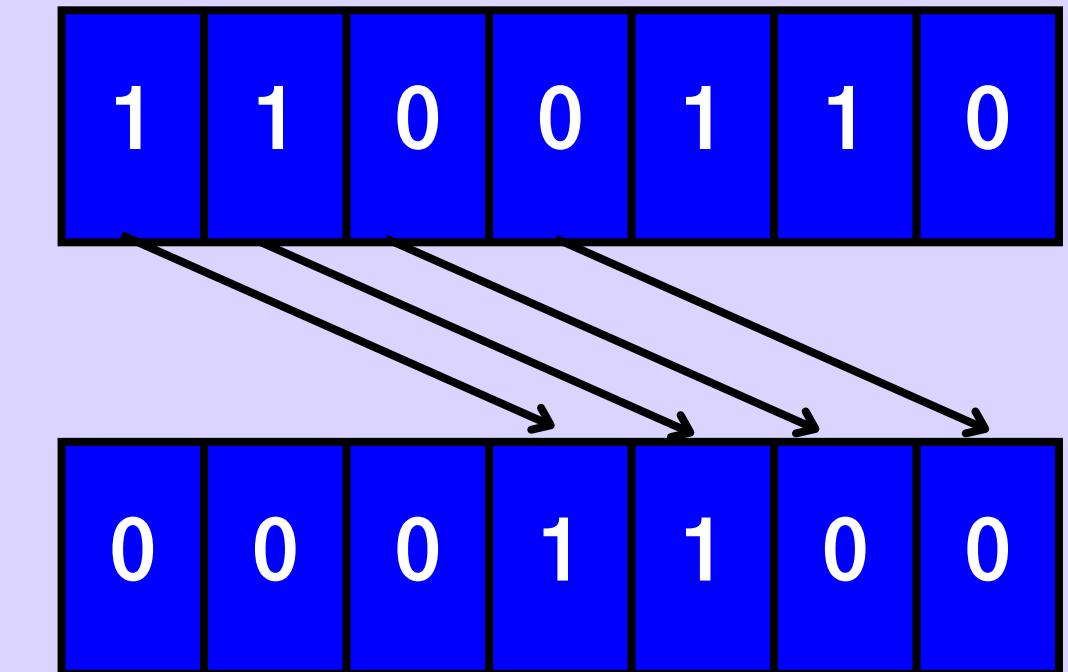
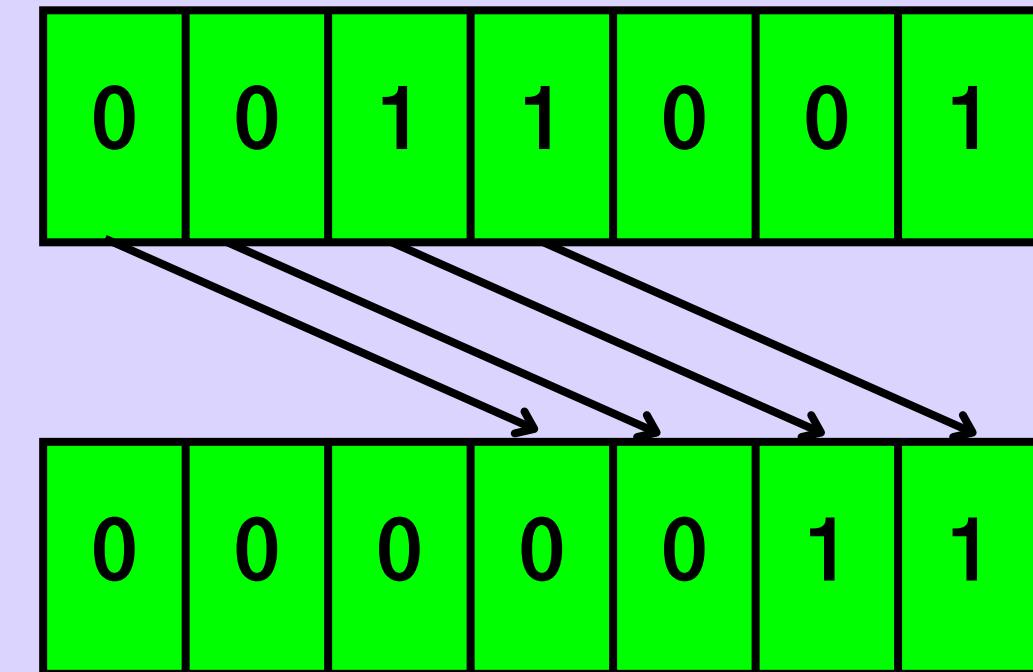
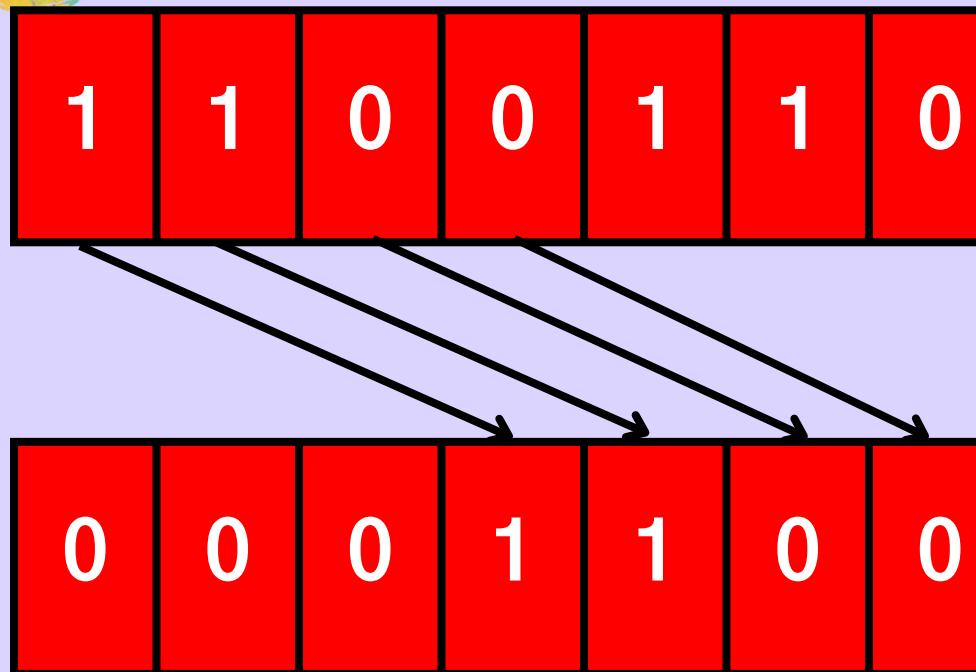
0 0 1 1 0 0 1

1 1 0 0 1 1 0



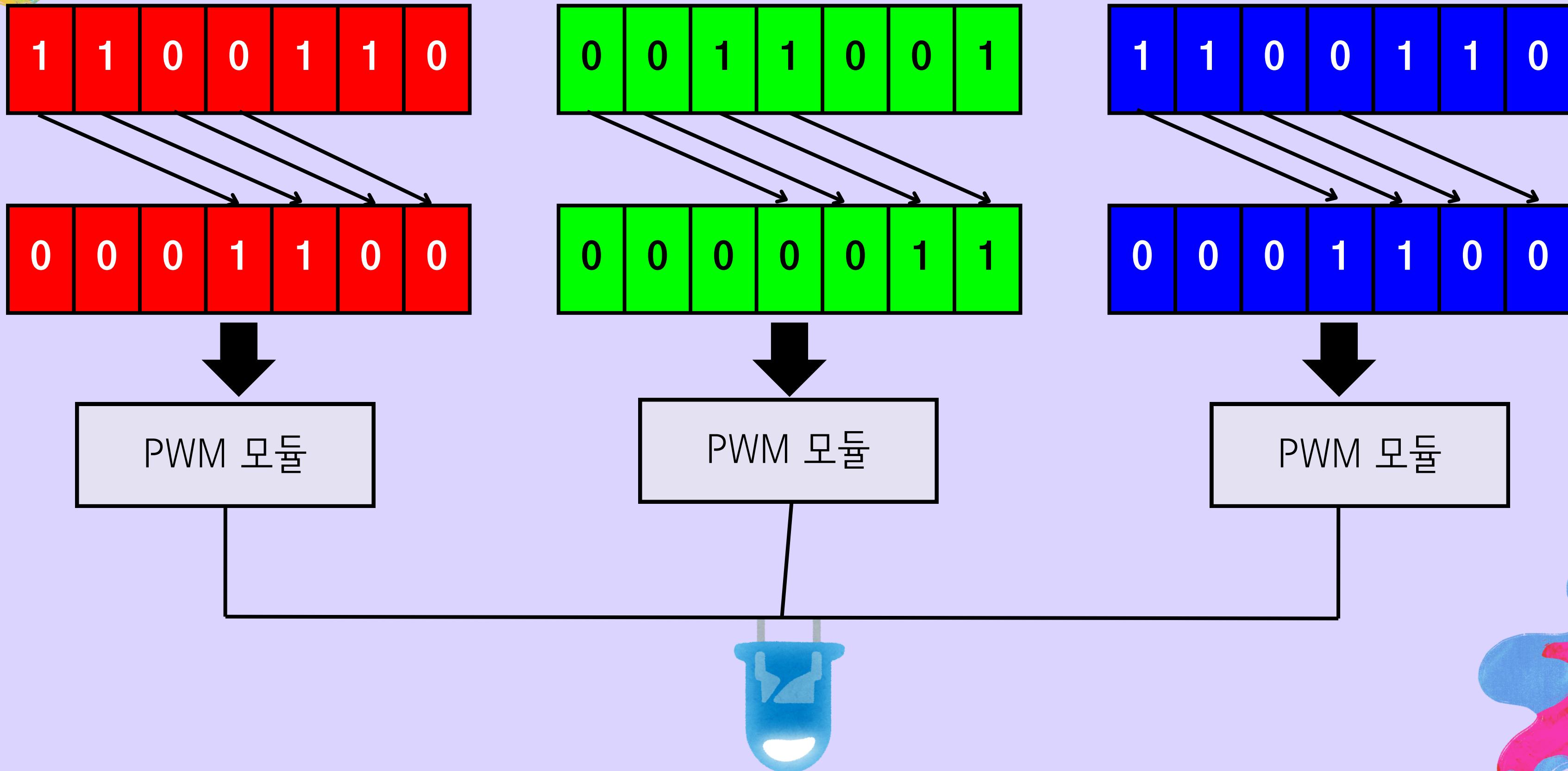
구현경차

레지스터 조작 - 밝기조절



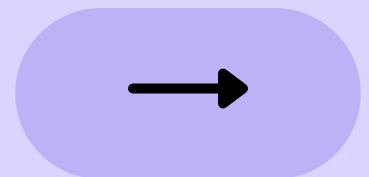
주제설명

RGB 출력으로 PWM 출력



IR 수신 모듈

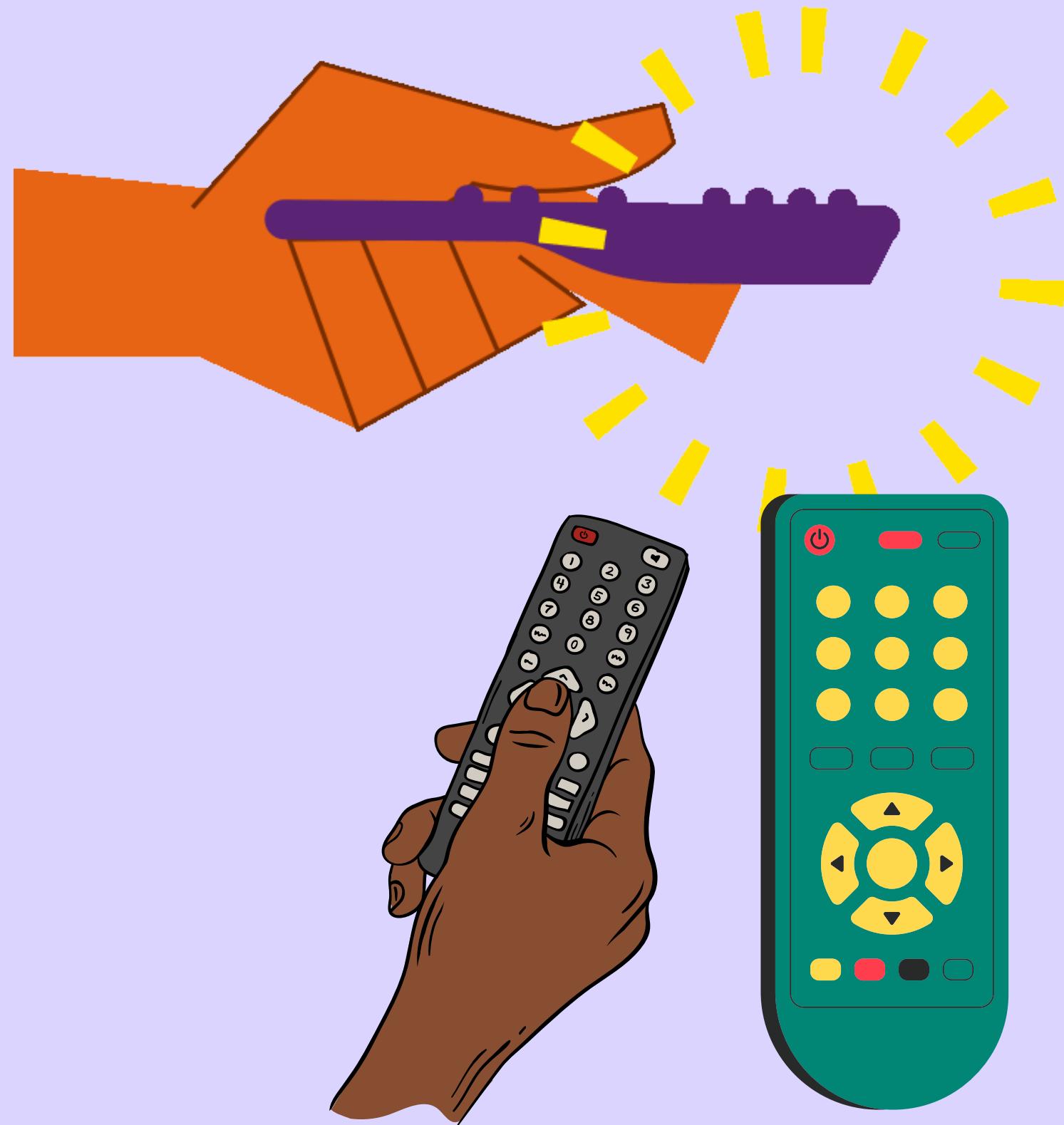
모듈



구현목표:

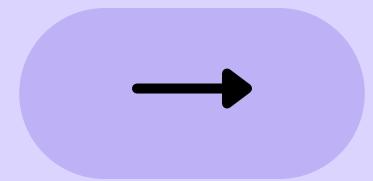
IR 수신기 데이터를 가공하여 출력

- IR 수신기 출력 파형 분석
- 직렬 입력 병렬 출력(SIPO 레지스터)
- 의도하지 않은 신호 차단



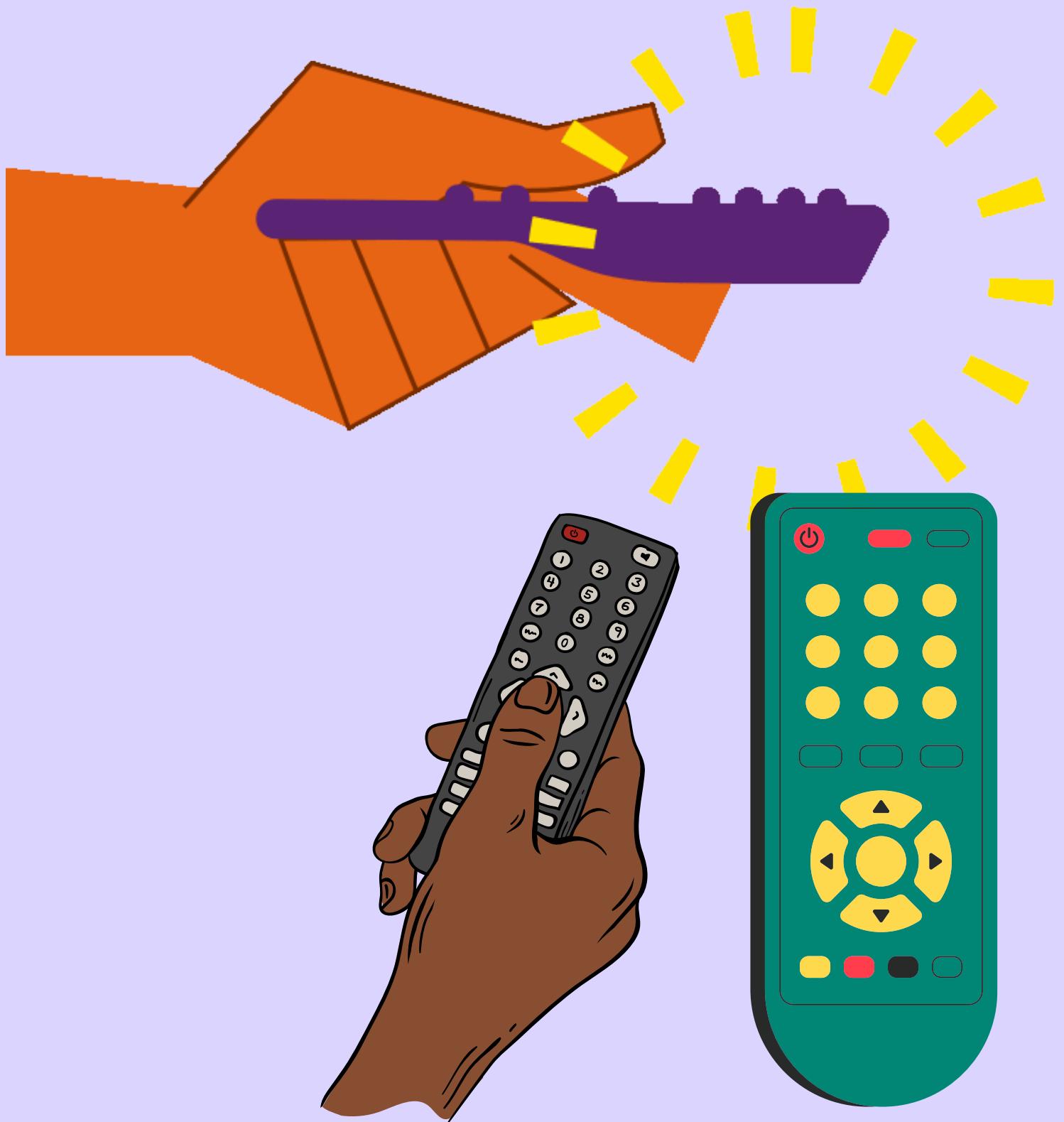
IR관련 주제

모듈



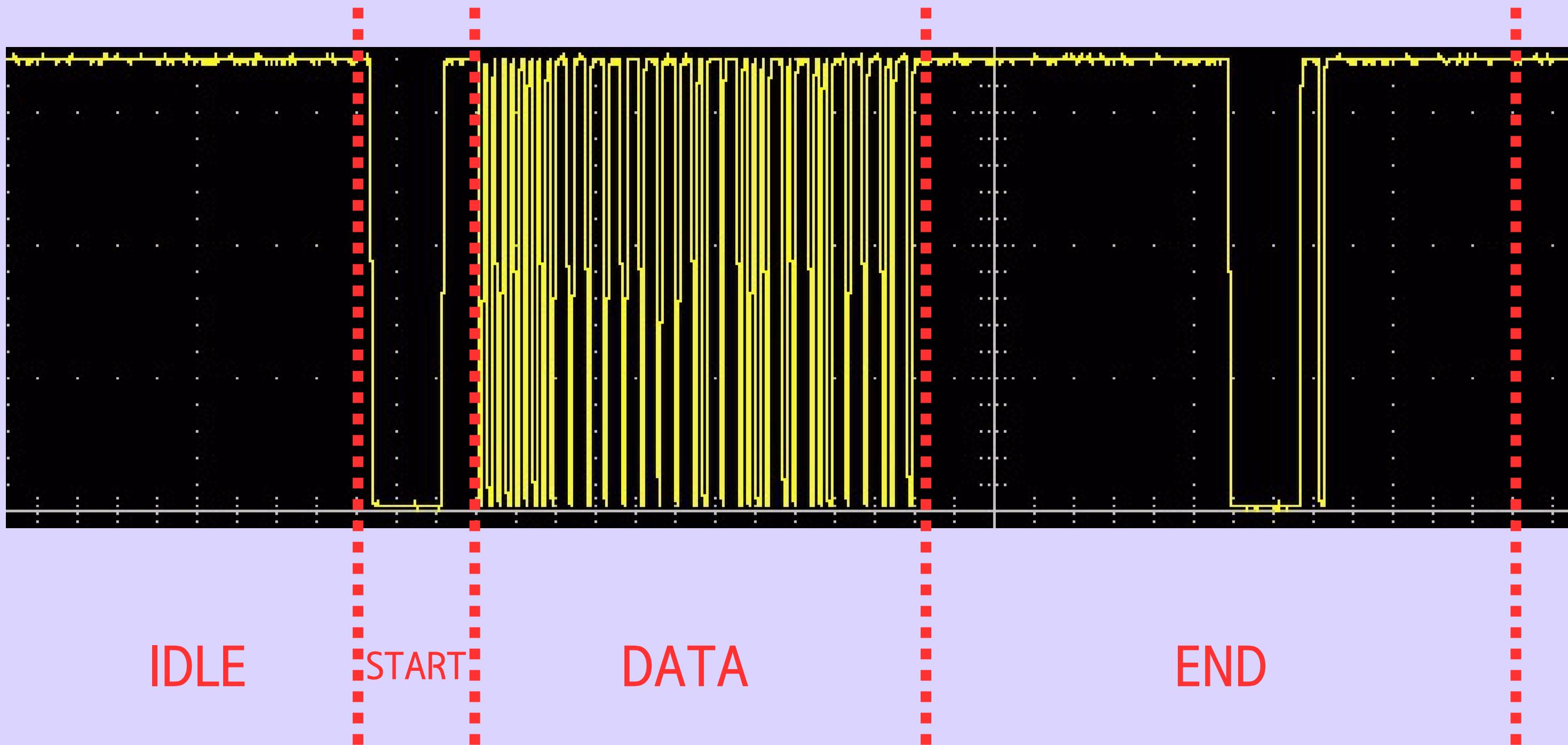
구현절차

- IR 수신기 출력 읽기
- 데이터 가공
- 버튼 길게 누름 탐지
- 데이터 출력



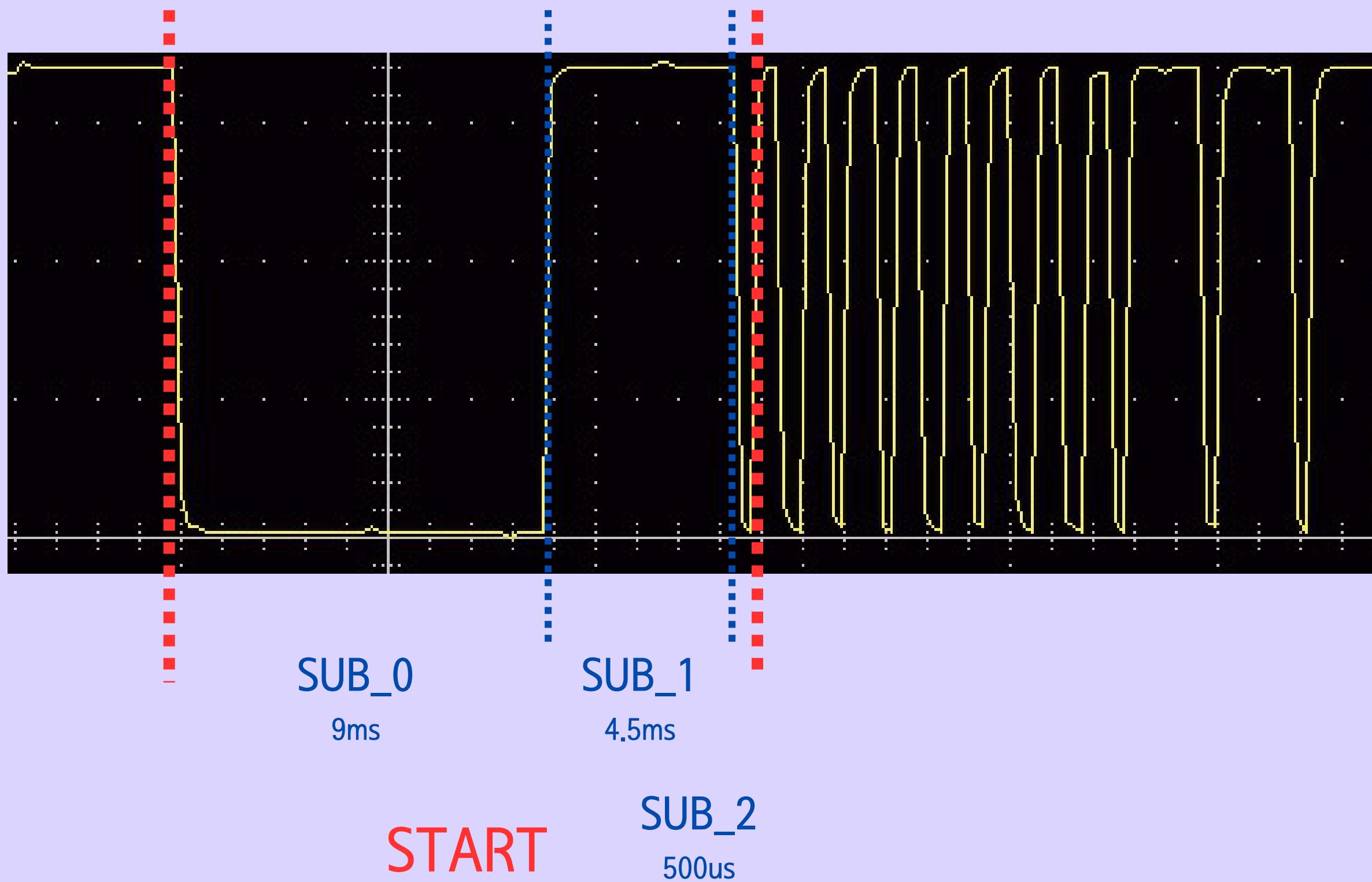
State 상태도

IR 수신기 출력 신호 분석



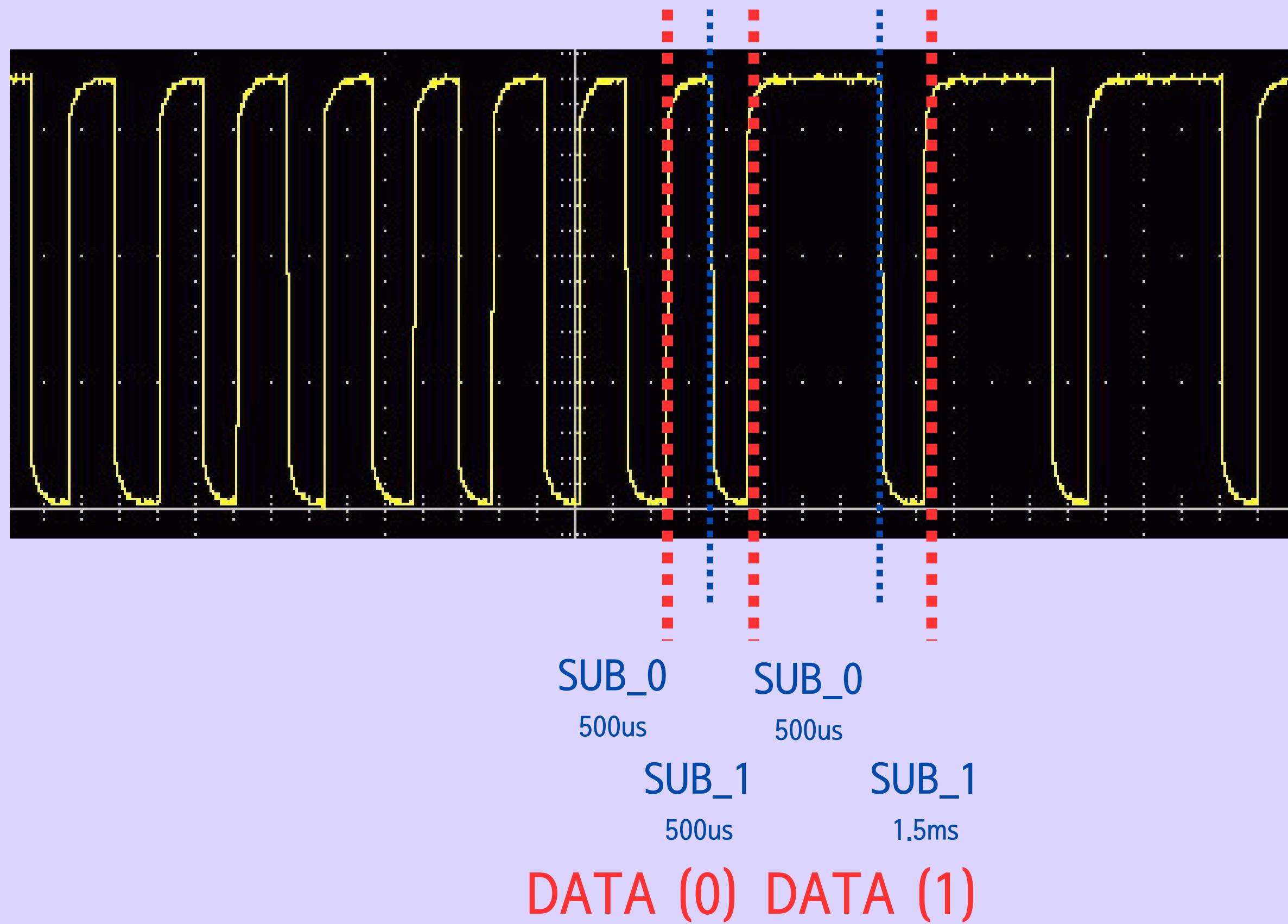
State 상태도

IR 수신기 출력 신호 분석



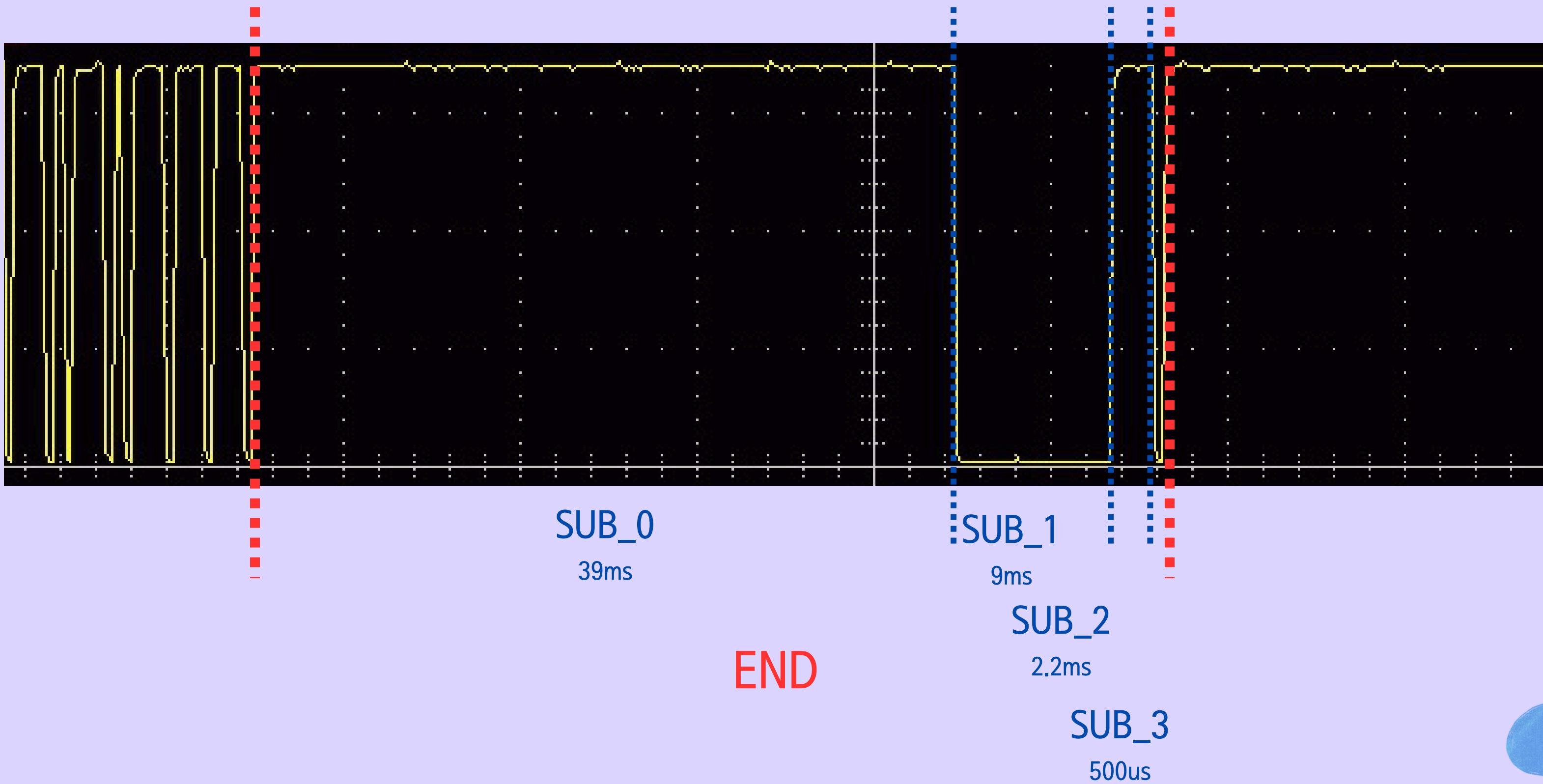
State 상태도

IR 수신기 출력 신호 분석



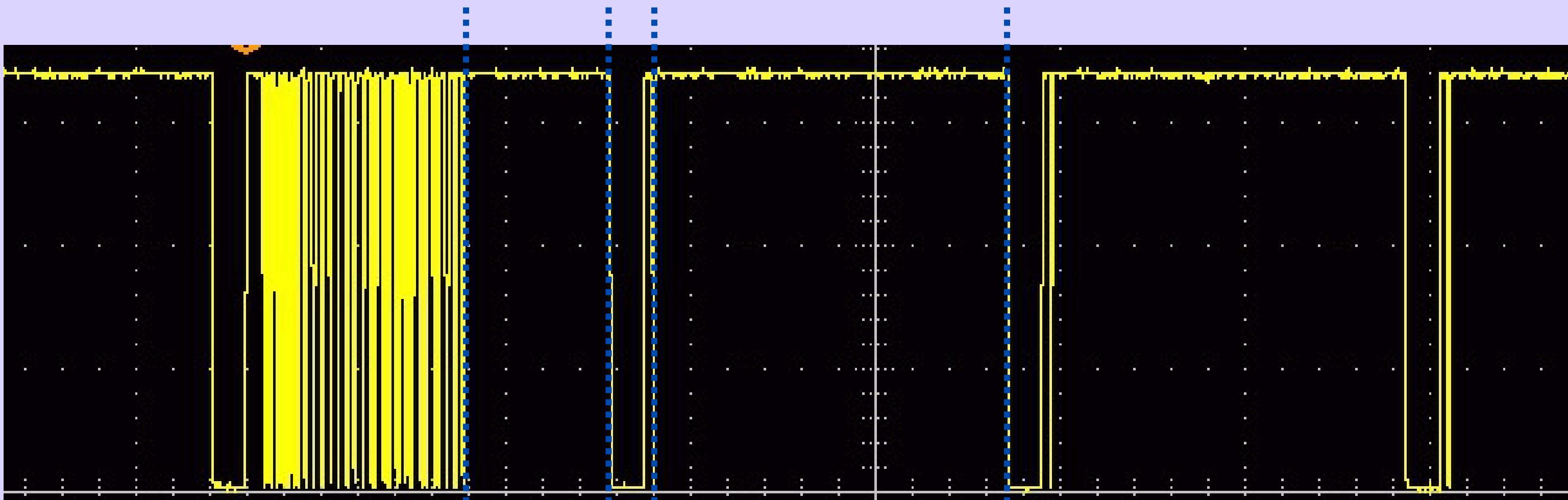
State 상태도

IR 수신기 출력 신호 분석



State 상태도

IR 수신기 출력 신호 분석



SUB_0
39ms

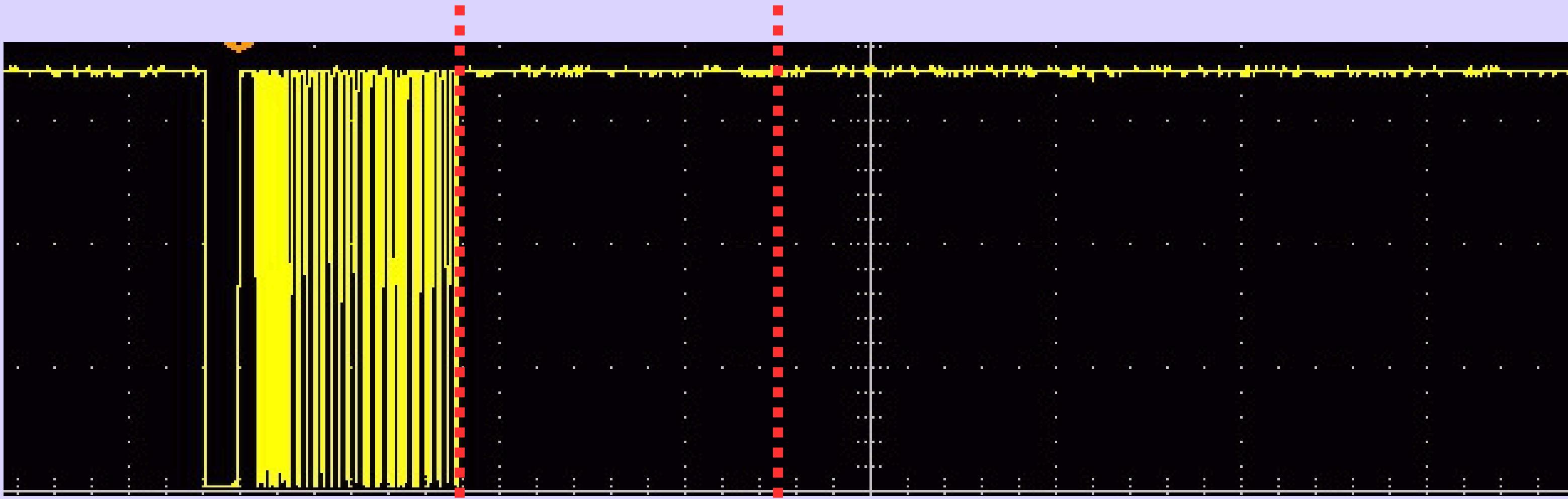
SUB_0
96ms

END

END

State 상태도

IR 수신기 출력 신호 분석



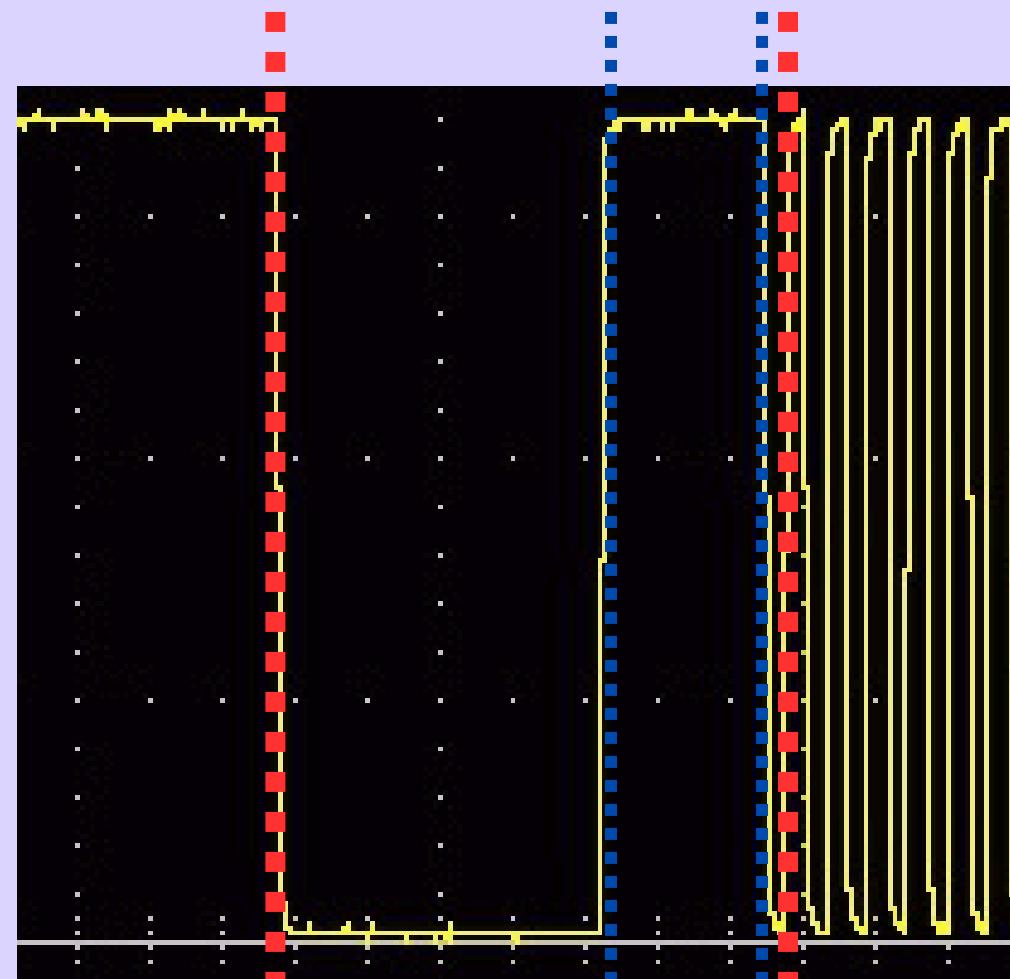
SUB_0
>96ms

END

IDLE

State 상태도

IR 수신기 출력 신호 분석

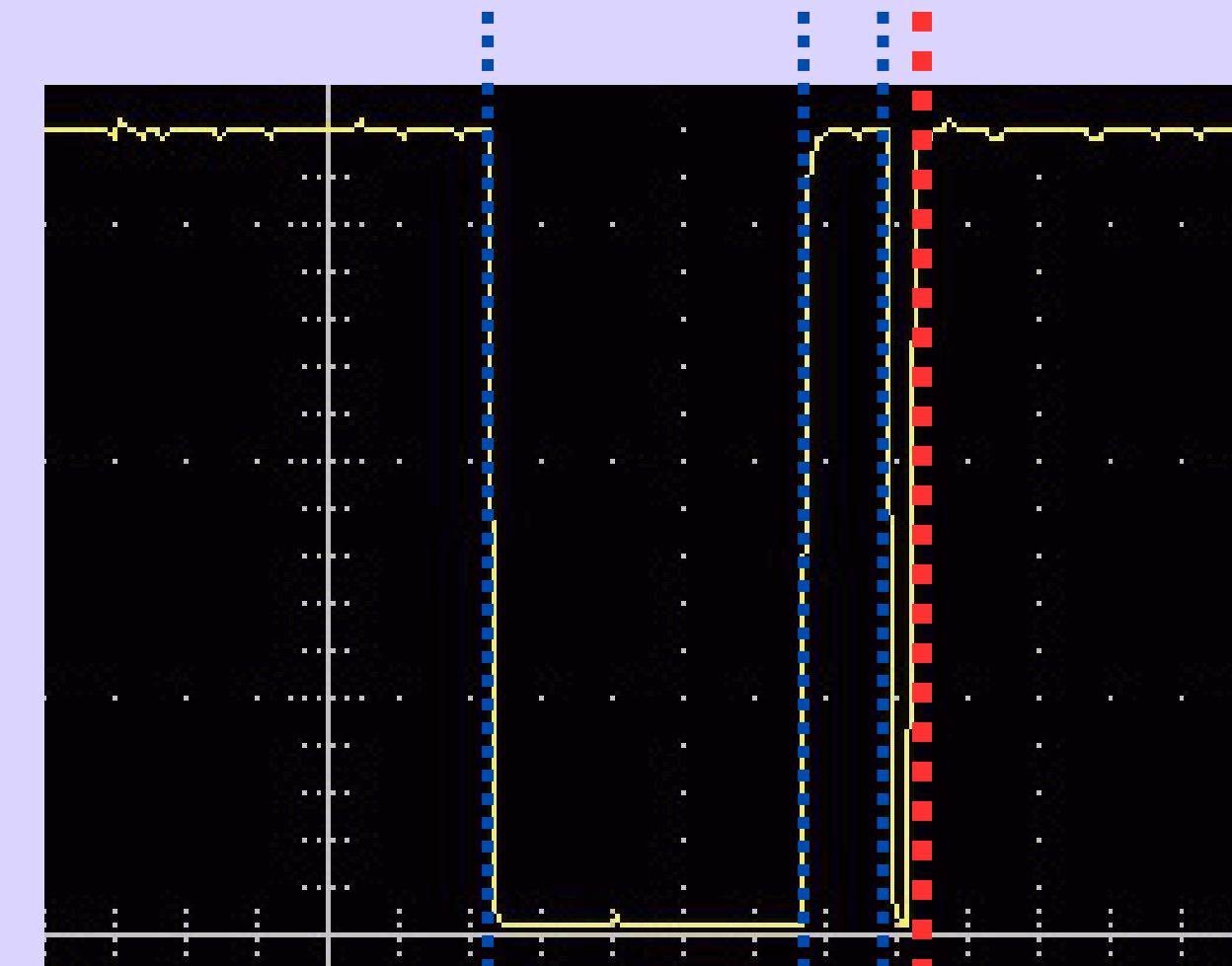


START

9ms

SUB_1
4.5ms

SUB_2
500us



END

39ms

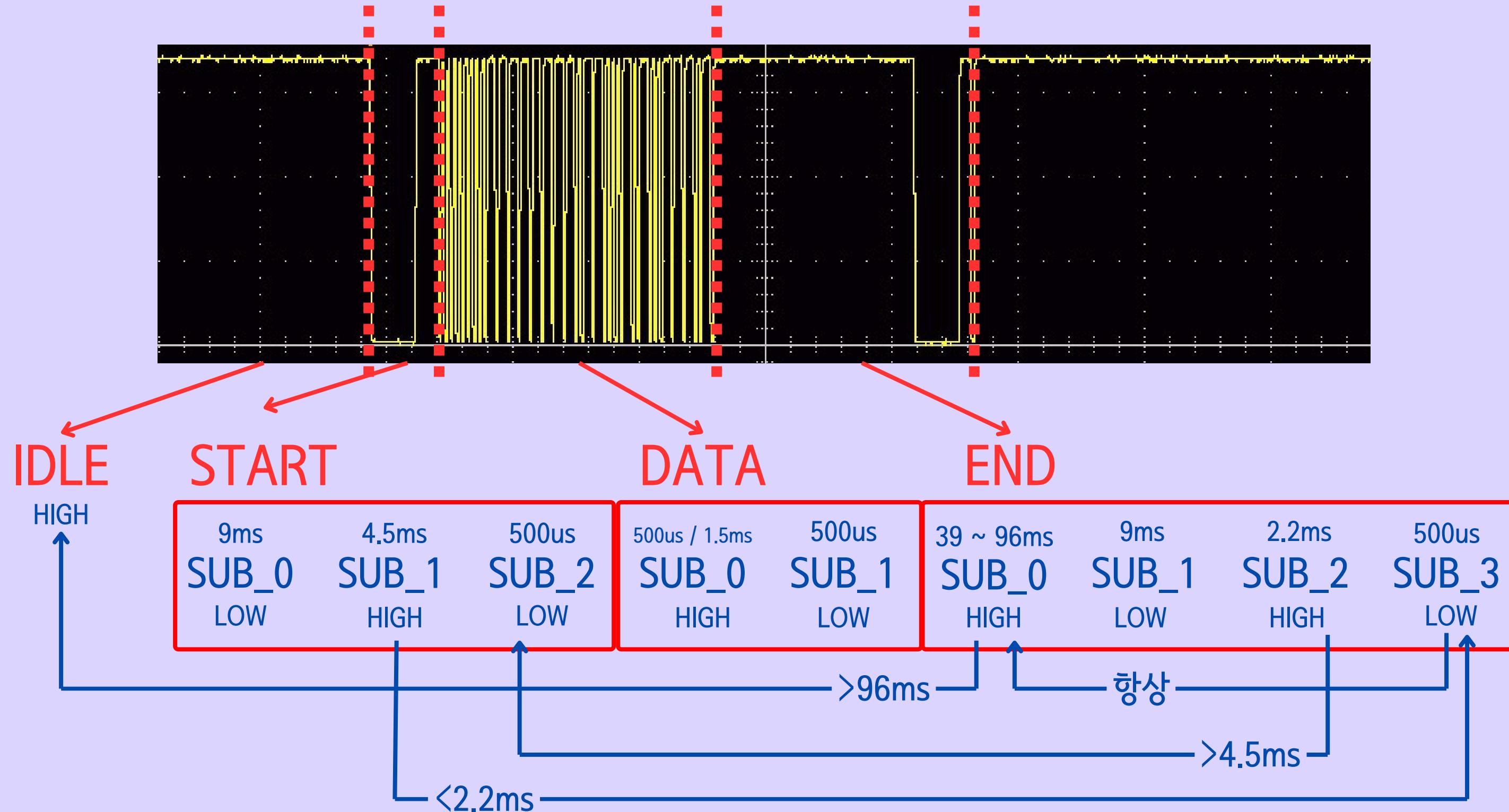
SUB_1
9ms

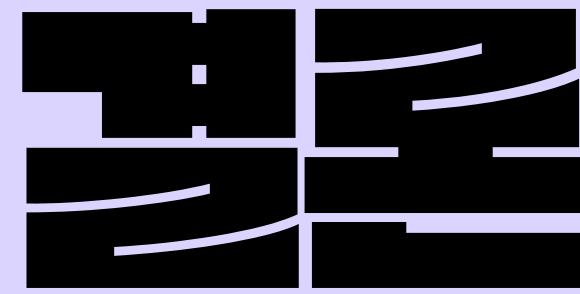
SUB_2
2.2ms

SUB_3
500us

State 상태도

상태 머신 설계





선풍기 바람제기 조작

- 적절한 전압 공급의 중요성을 알게 되었다.
- 모듈에 공급할 전압 확인(DC모터는 5V)

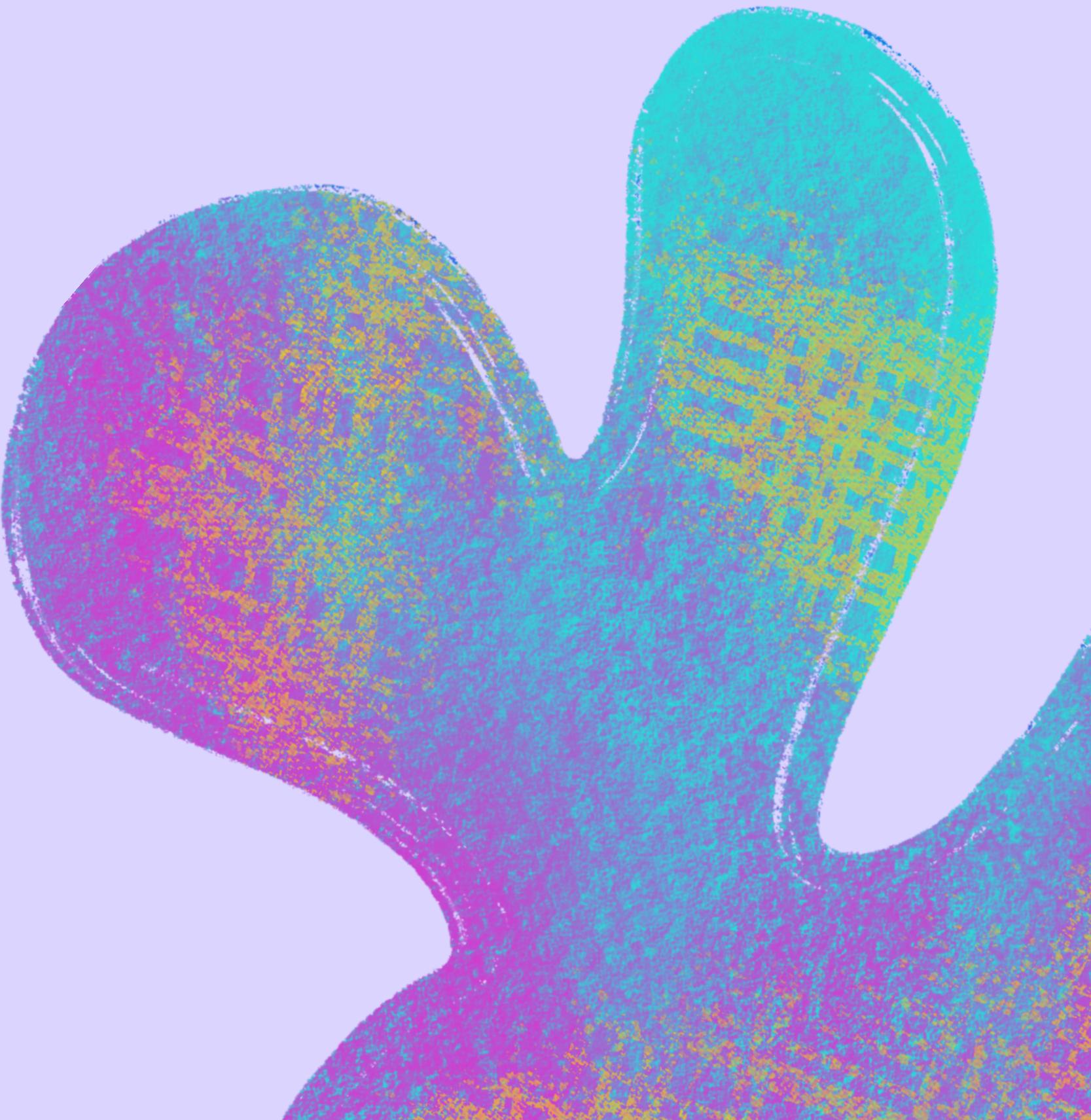
카운터 / OFF

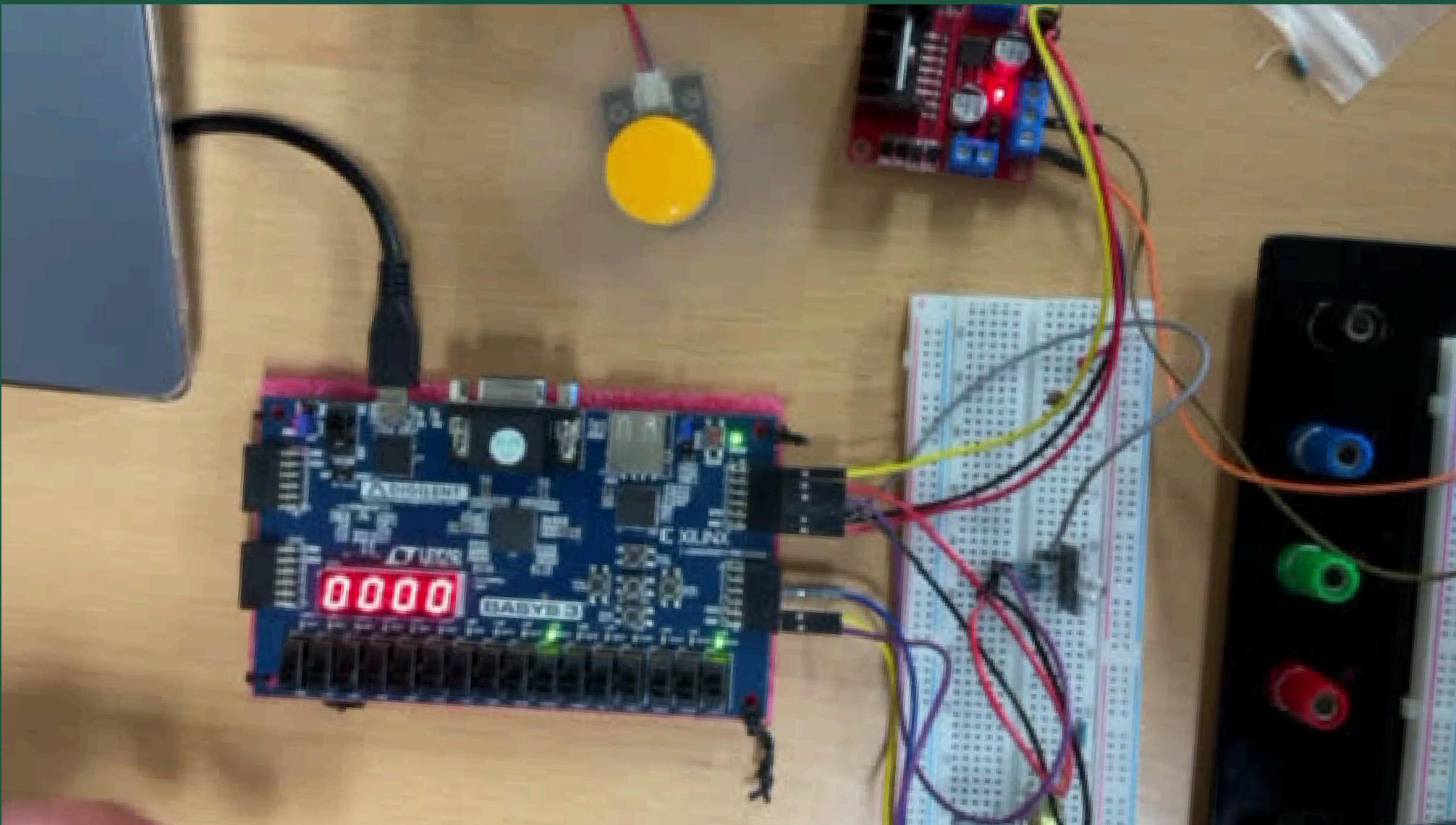
- IF 조건문의 중요성을 알게 되었다.
- 선언의 순서.

리모컨 수신 모듈

- State Machine 설계 (중복 하위 상태 합성 실패)
- 의도하지 않은 신호 수신

동화시대





정의당
정의당

