

성균관대학교

S I O R

로봇학회

2022년 06월 26일

EMBEDDED

7 주 차

목차

- 모터
- 센서
- 블루투스
- EEPROM
- ○ ○
- ○ ○

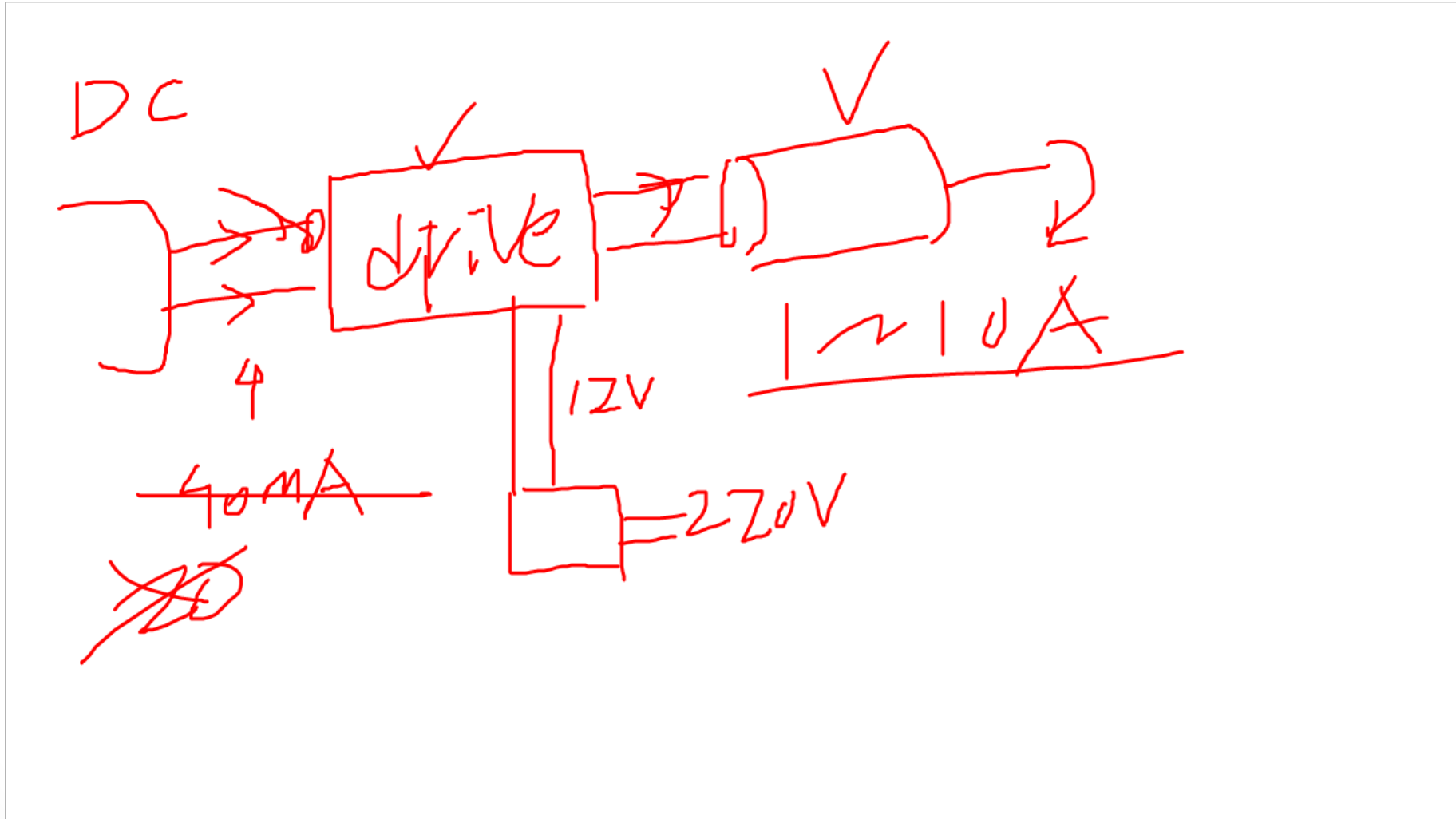
EMBEDDED

모터

DC / Servo / Stepper
T T T

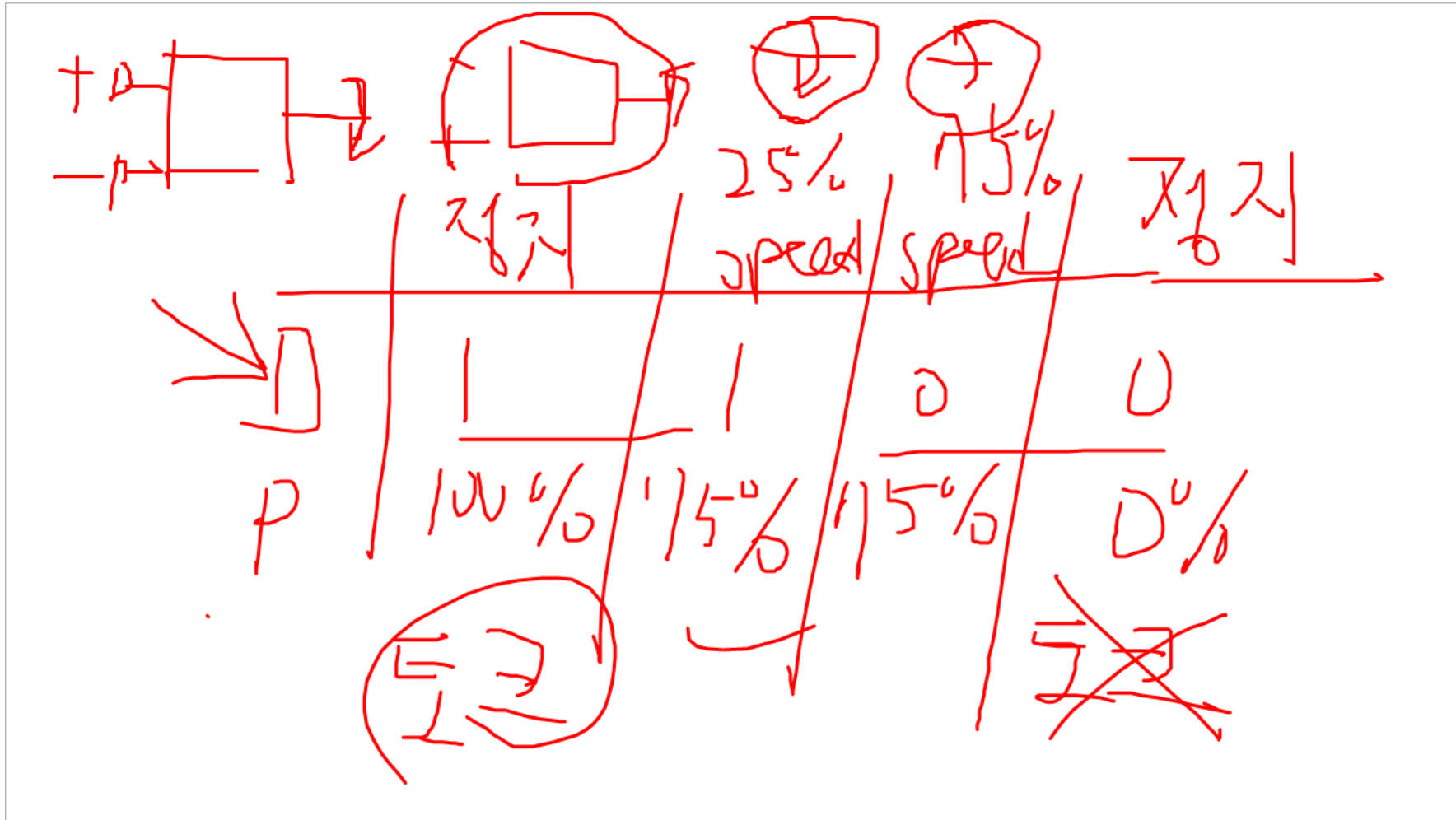
EMBEDDED

모터



EMBEDDED

모터



모터

servo

DC speed

servo pos

$0^\circ \sim 180^\circ$

0% ~ 100%



EMBEDDED

모터

SERVO

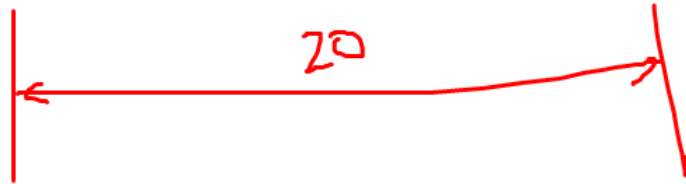


10 WM
(50 Hz)

duty = 1.5ms 90°
1ms 0° / 2ms 180°

EMBEDDED

모터



$\rightarrow 0^\circ$



$\rightarrow 90^\circ$

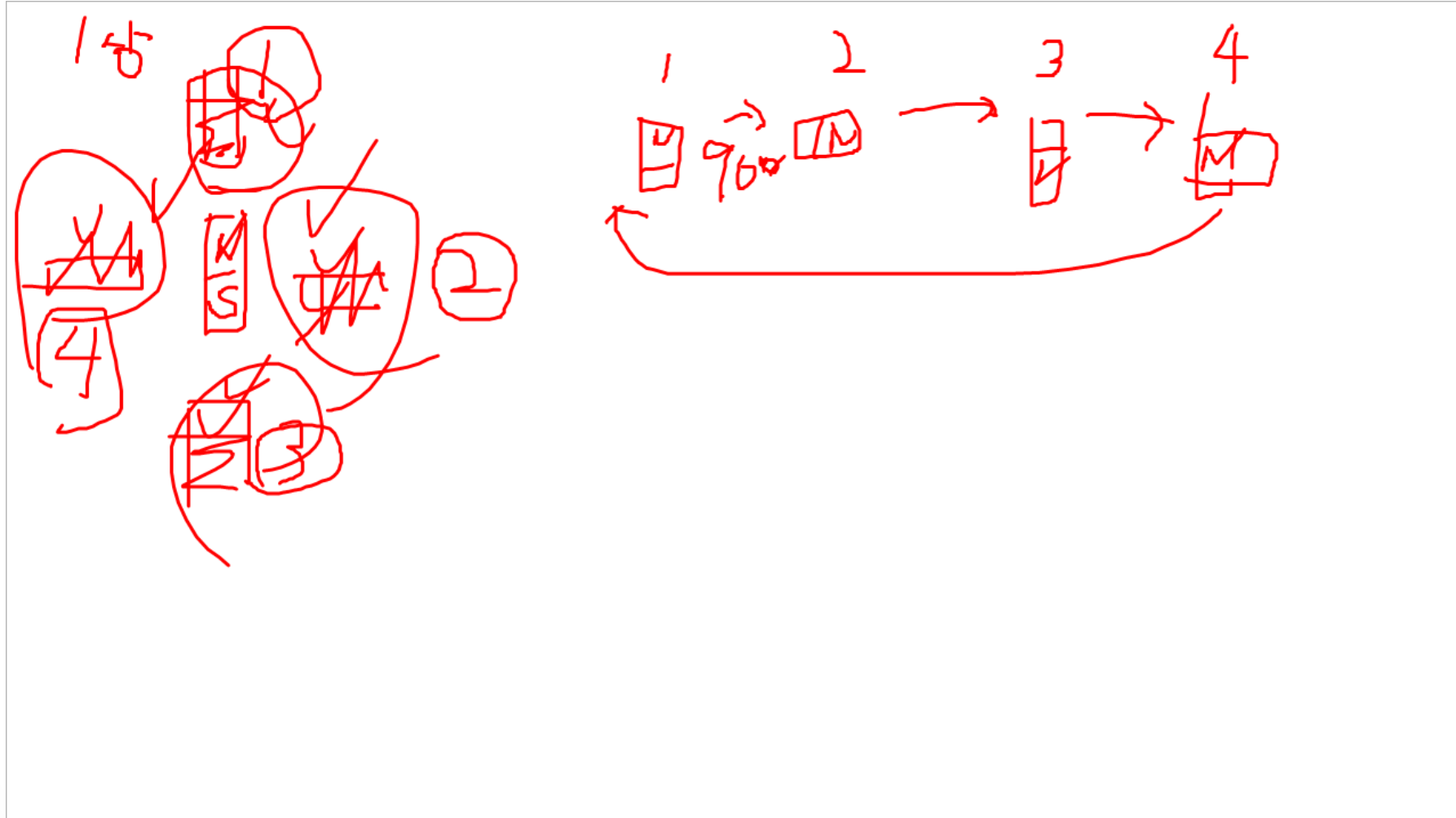


$\rightarrow 180^\circ$



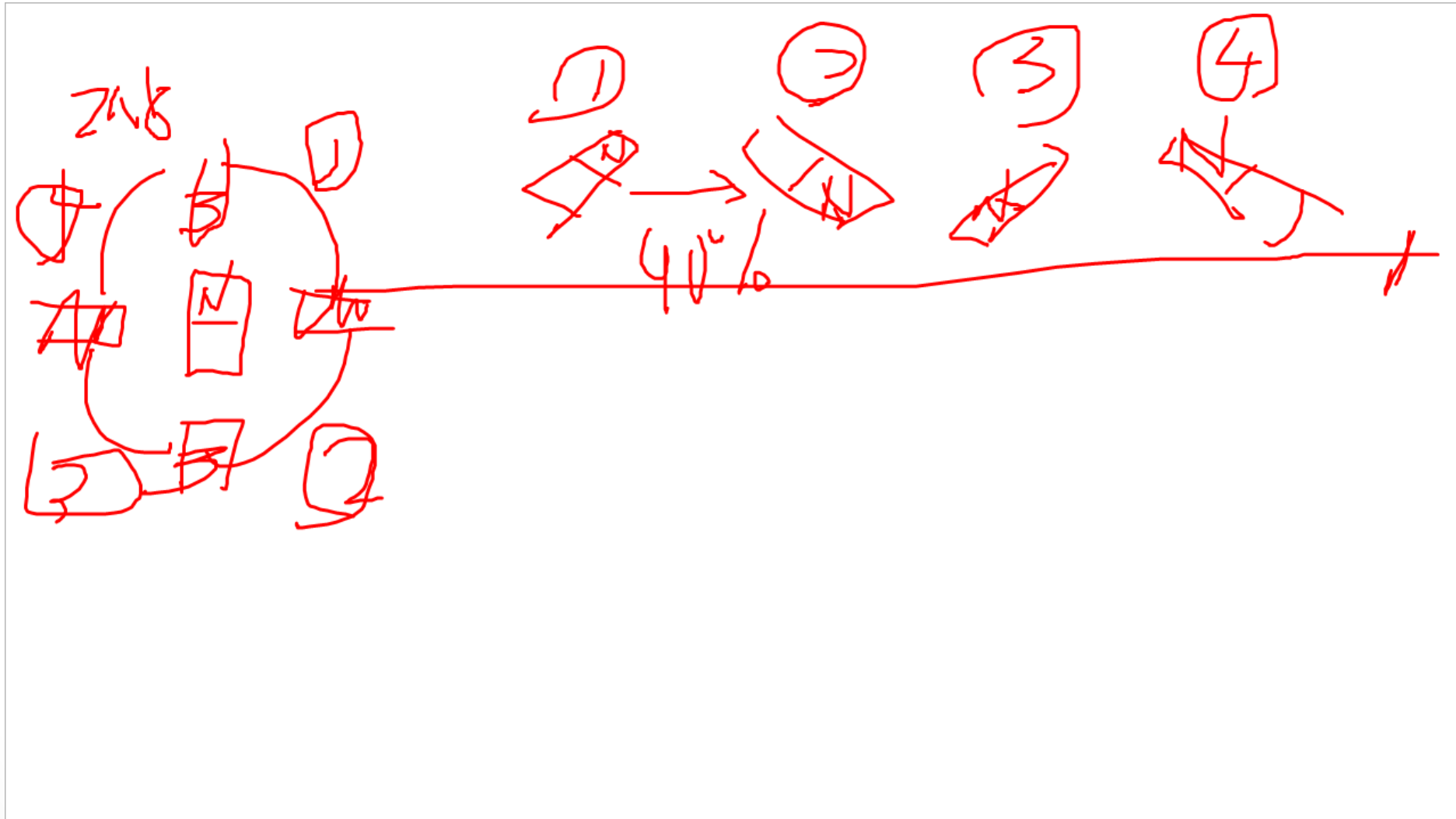
EMBEDDED

모터



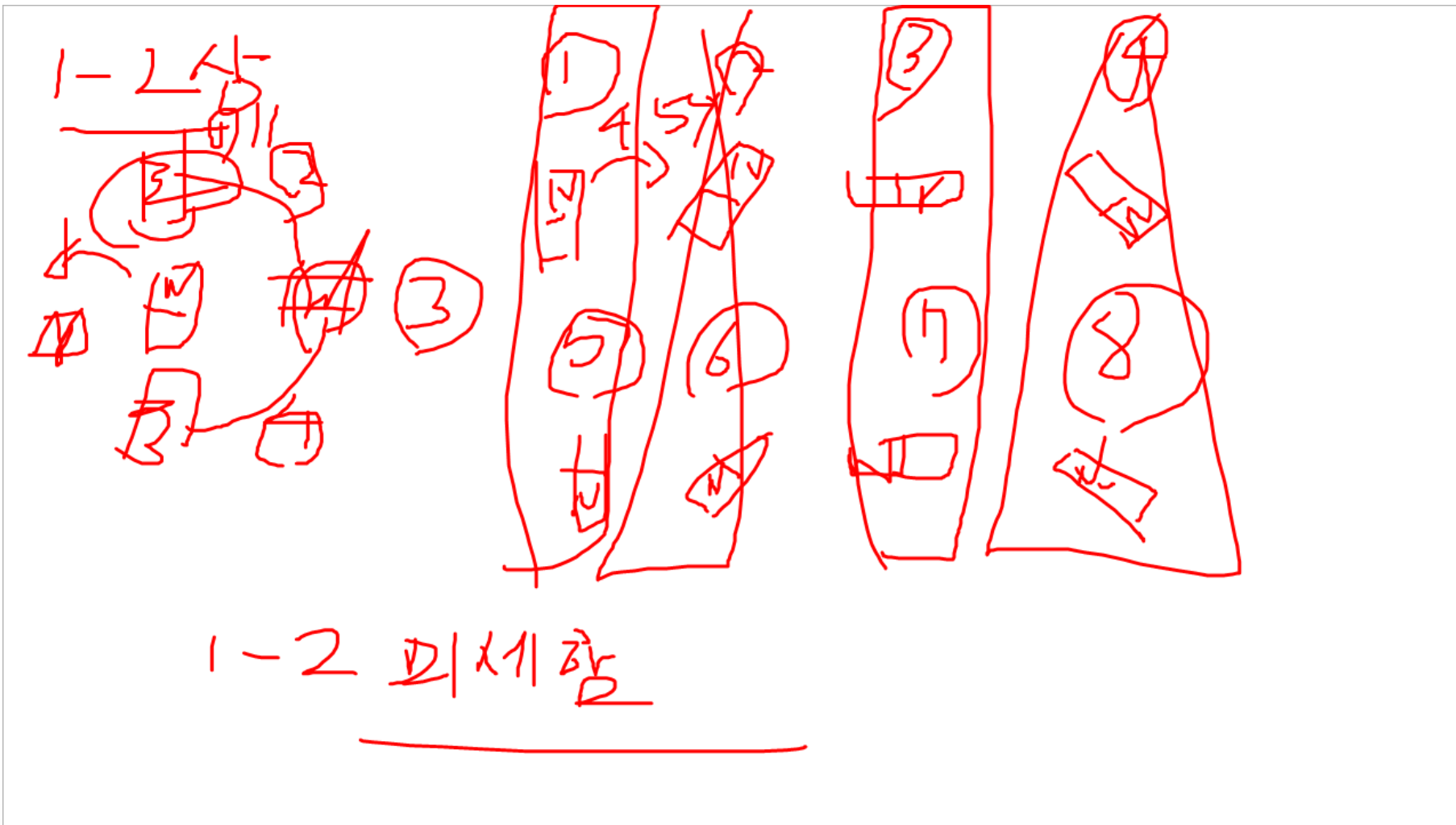
EMBEDDED

모터



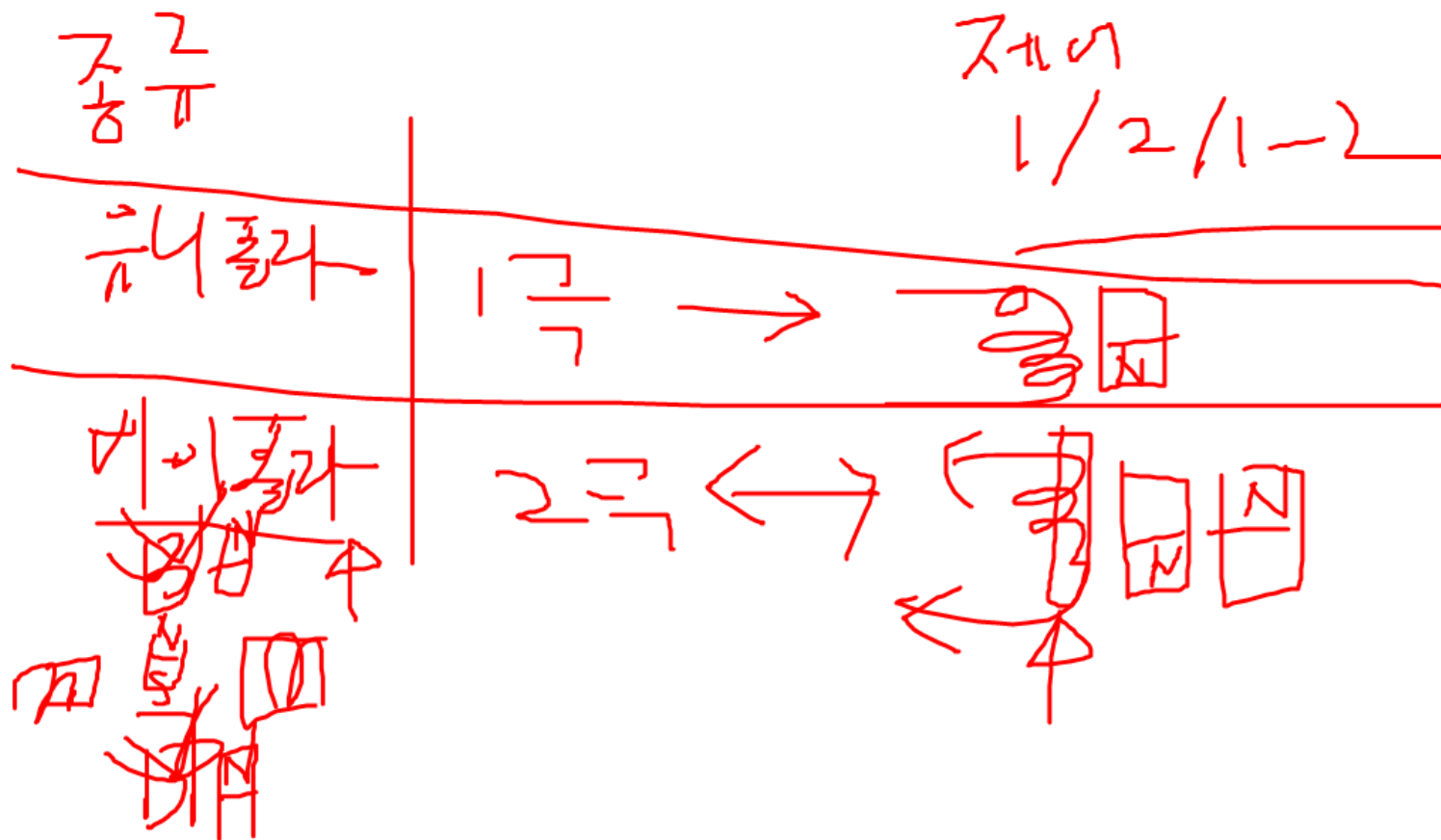
EMBEDDED

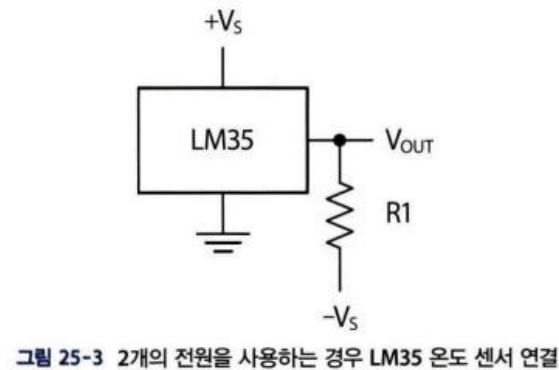
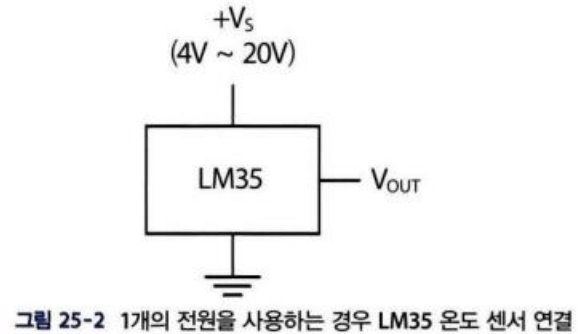
모터



EMBEDDED

모터





```
while(1)
{
    // 온도 센서의 출력 전압을 ADC를 거쳐 읽는다.
    read = read_ADC();

    // 0에서 1023 사이의 값을 0V에서 5V 사이 값으로 변환한다.
    input_voltage = read * 5.0 / 1023.0;

    // 10mV에 1℃이므로 100을 곱해서 현재 온도를 얻는다.
    temperature = input_voltage * 100.0;
}
```

25.2 초음파 거리 센서

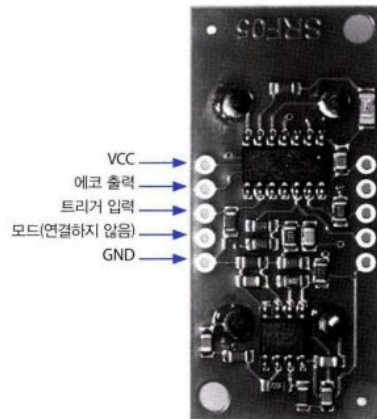
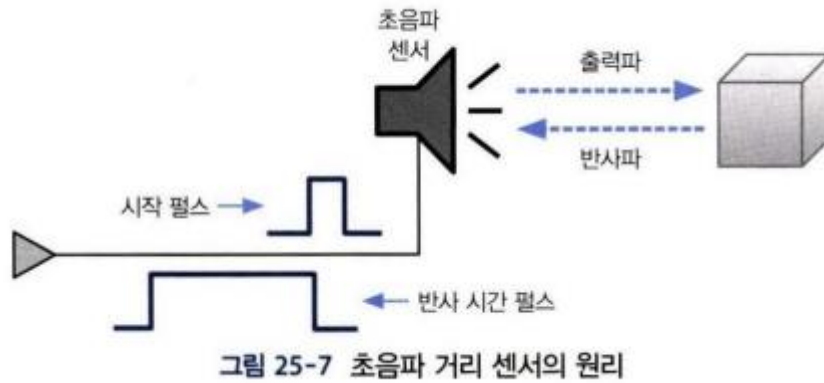


그림 25-9 SRF05 초음파 센서 모듈 연결 핀

코드 25-3 초음파 센서로 거리 측정하기

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>
#include "UART.h"

#define PRESCALER 1024 // 분주비

void Timer_init(void){
    TCCR0B |= (1 << CS02) | (1 << CS00); // 분주비 1024로 설정
}

uint8_t measure_distance(void)
{
    // 트리거 핀으로 펄스 출력
    PORTC |= (1 << PC0); // HIGH 값 출력
    _delay_us(10); // 10마이크로초 대기
    PORTC &= ~(1 << PC0); // LOW 값 출력

    // 에코 핀이 HIGH가 될 때까지 대기

    TCNT0 = 0;
    while(!(PINC & 0x02))
        if(TCNT0 > 250) return 255; // 장애물이 없는 경우

    // 에코 핀이 LOW가 될 때까지의 시간 측정
    TCNT0 = 0;
    while(PINC & 0x02){
        if (TCNT0 > 250){ // 측정 불가능
            TCNT0 = 0;
            break;
        }
    }

    // 에코 핀의 펄스폭을 마이크로초 단위로 계산
    double pulse_width = TCNT0 * PRESCALER * 1000000.0 / F_CPU;

    return pulse_width / 58; // 센티미터 단위 거리 반환
}

int main(void)
{
    uint8_t distance;
    DDRC |= 0x01; // 트리거 핀 출력으로 설정
    DDRC &= ~0xFD; // 에코 핀 입력으로 설정

    UART_INIT(); // UART 통신 초기화
    Timer_init(); // 타이머 초기화

    while(1)
    {
        distance = measure_distance(); // 거리 측정

        UART_printString("Distance(cm) = ");
        UART_print8bitNumber(distance);
        UART_transmit('\n');

        _delay_ms(1000);
    }
}
```

EMBEDDED

블루투스

마이크로컨트롤러와 함께 사용되는 블루투스는 2.0 대부분
마스터와 슬레이브 구성, 마스터에는 최대 7개의 슬레이브 연결가능
마스터와 슬레이브 사이의 통신만이 가능(슬레이브끼리 불가)

여기서는 HC-06모듈 사용
VCC, GND, RX, TX 핀 사용

EMBEDDED EEPROM

표 27-1 ATmega328의 메모리

	플래시 메모리	SRAM	EEPROM
크기(Kbyte)	32	2	1
주 용도	프로그램 저장	데이터 저장	데이터 저장
비휘발성	○	×	○
프로그램 실행 중 변경 가능	불가능	가능	가능
속도	중간	가장 빠름	가장 느림
수명	10,000회 쓰기	반영구적	100,000회 쓰기

```
uint8_t eeprom_read_byte (const uint8_t *__p);
uint16_t eeprom_read_word (const uint16_t *__p);
uint32_t eeprom_read_dword (const uint32_t *__p);
float eeprom_read_float (const float *__p);
void eeprom_read_block (void *__flash, const void *__eeprom, size_t __n);

void eeprom_write_byte (uint8_t *__p, uint8_t __value);
void eeprom_write_word (uint16_t *__p, uint16_t __value);
void eeprom_write_dword (uint32_t *__p, uint32_t __value);
void eeprom_write_float (float *__p, float __value);
void eeprom_write_block (const void *__flash, void *__eeprom, size_t __n);

void eeprom_update_byte (uint8_t *__p, uint8_t __value);
void eeprom_update_word (uint16_t *__p, uint16_t __value);
void eeprom_update_dword (uint32_t *__p, uint32_t __value);
void eeprom_update_float (float *__p, float __value);
void eeprom_update_block (const void *__flash, void *__eeprom, size_t __n);
```

성균관대학교

Thank You

로봇동아리