

# DC 모터 제어

---

컴퓨터 네트워크 설계

청주대학교 전자공학과  
한철수

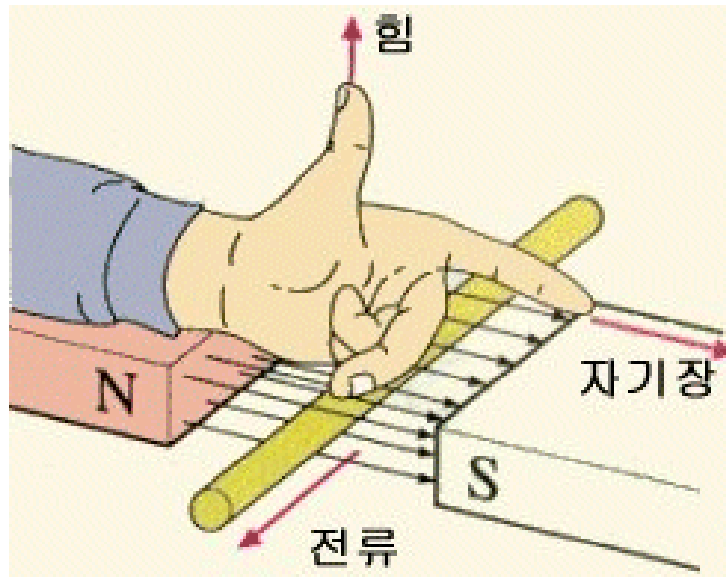
# 목차

---

- 모터의 개요
- 기본적인 DC 모터 제어 회로
- DC 모터의 양방향 제어 회로
- 제어 연습

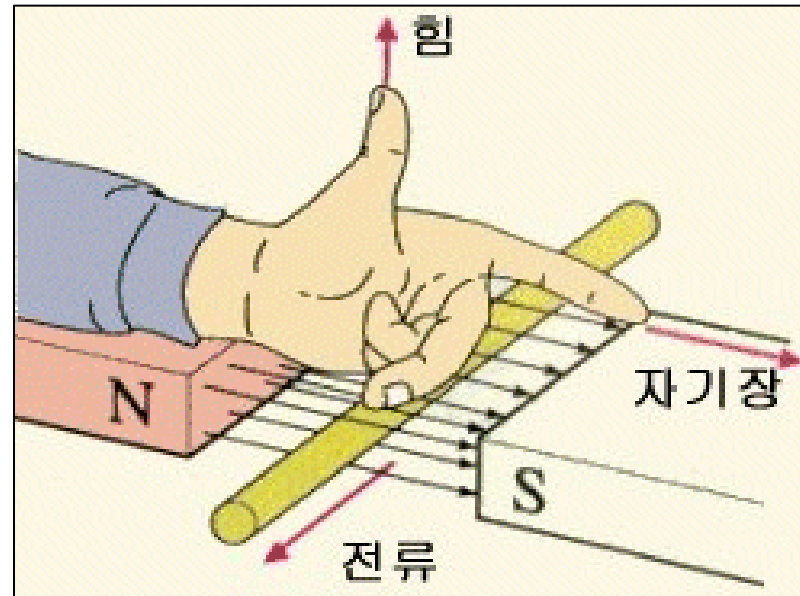
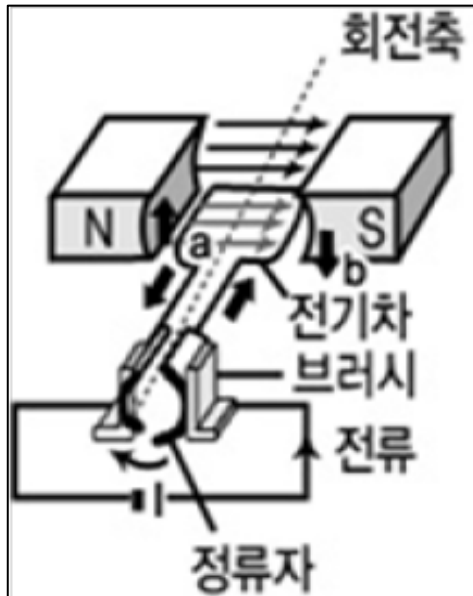
# 모터

- 모터(motor)는 전력을 이용하여 회전 운동의 힘을 얻는 장치를 말함.
  - 전동기라고도 함.
- 동작 원리
  - 자기장 내에서 전류가 흐르는 도체에는 자기력이 작용함.
    - 플레밍의 왼손 법칙
  - 모터는 이 원리를 응용해 자기장 내에서 전류를 흘려 자기력을 발생시키고, 이 힘을 회전 운동의 원천으로 함.

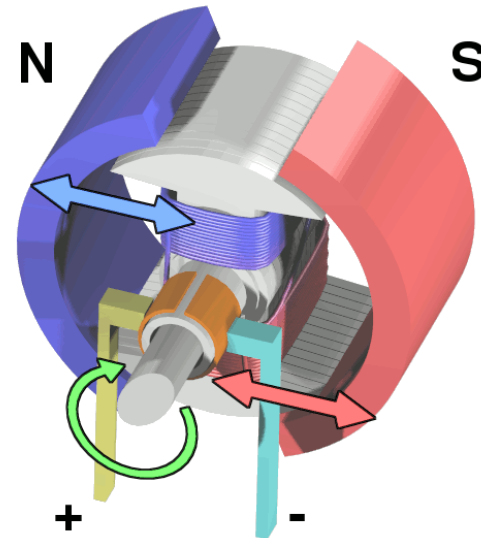
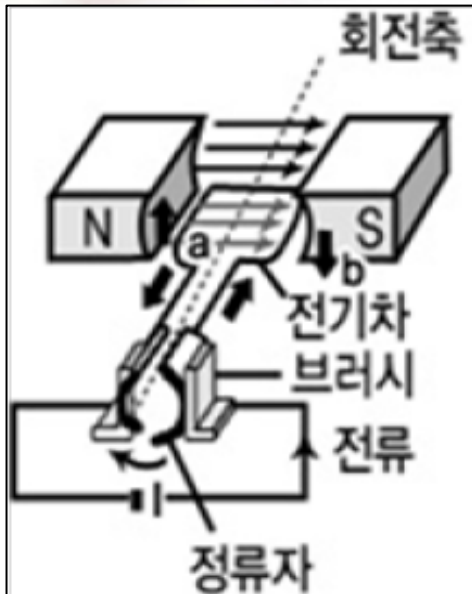


# DC 모터

- DC 모터는 직류(DC) 전류를 이용하는 모터를 말함.
  - 소형 장난감, 컴퓨터 팬, 전기 면도기 등에 주로 이용됨.

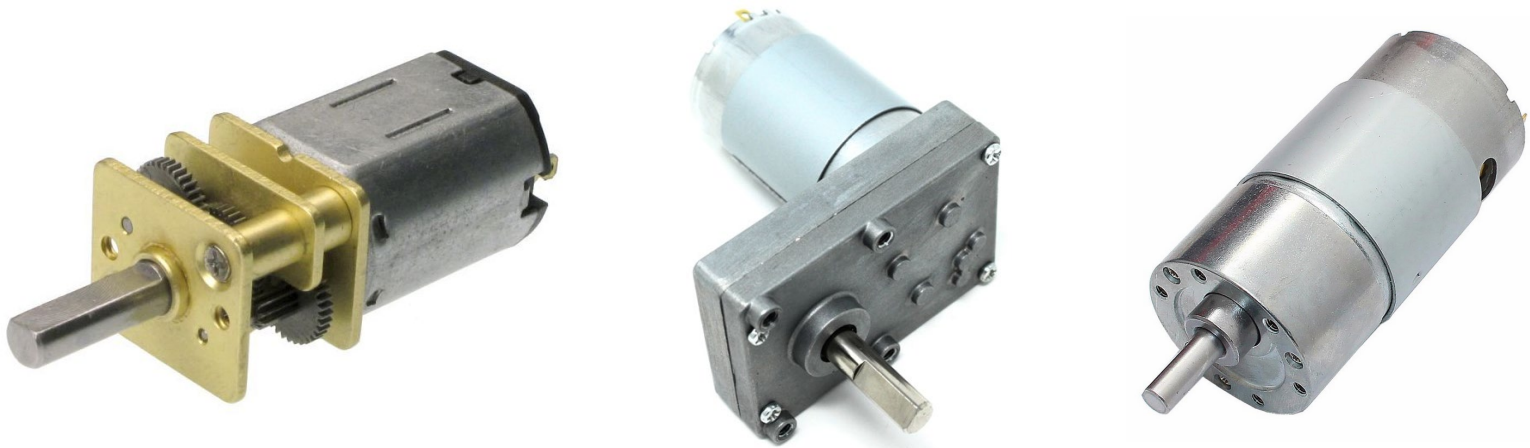


# DC 모터의 내부

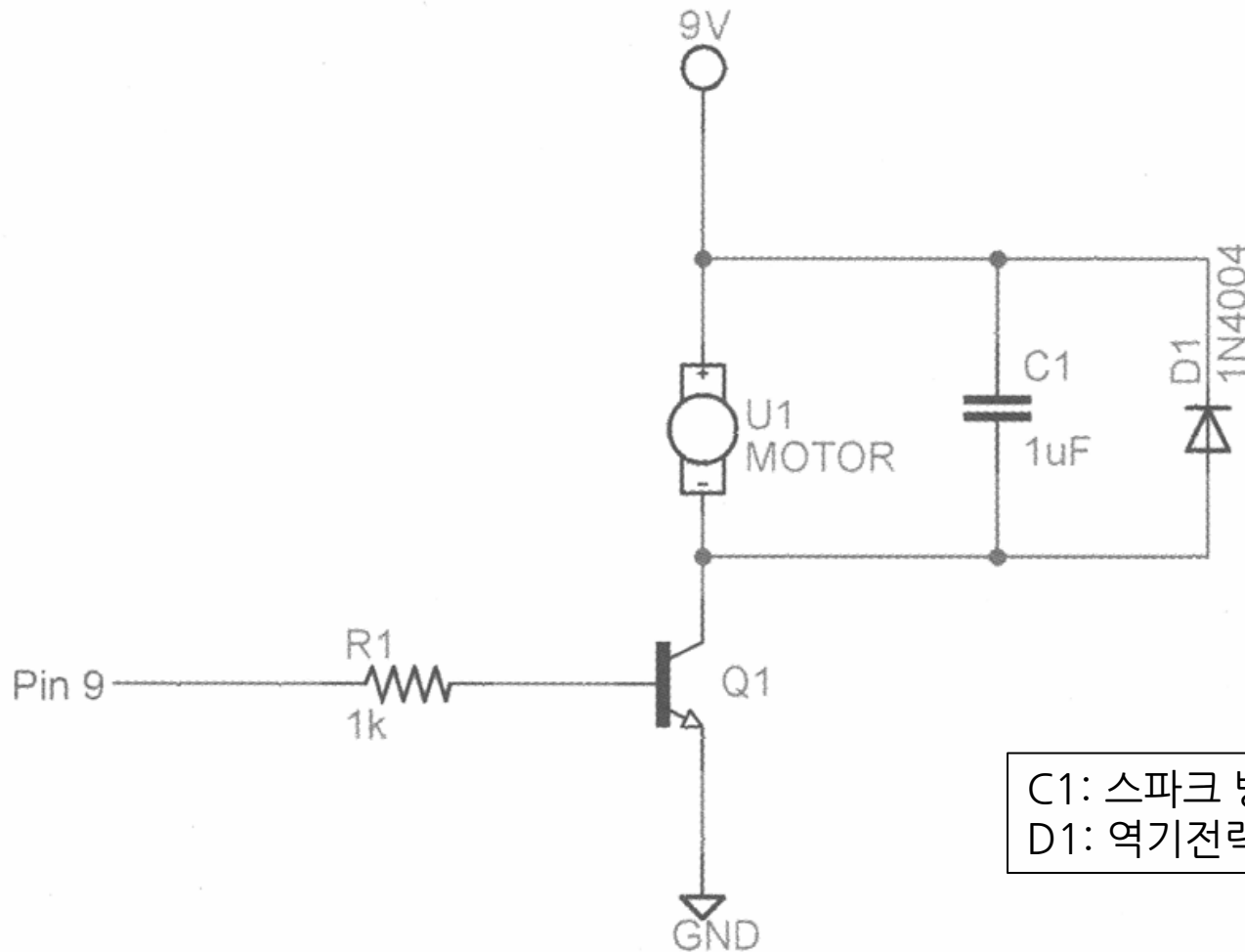


# 기타 모터

- AC 모터
  - 교류(AC) 전류를 이용하는 모터를 말함.
  - AC 220V의 고전압을 모터에 공급하여 모터를 회전시키는 세탁기, 선풍기, 냉장고 등에 주로 이용됨.
- 기어드 모터 (감속 모터)
  - 회전 속도를 줄이기 위해 기어를 추가한 모터를 말함.
    - DC 모터, AC 모터 모두 기어를 추가하여 기어드 모터로 사용 가능함.
  - 회전 속도가 줄어드는 대신 힘을 증대 시킬 수 있음.

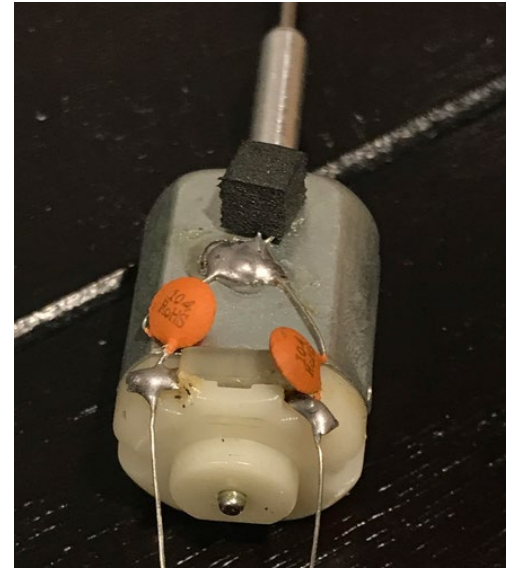
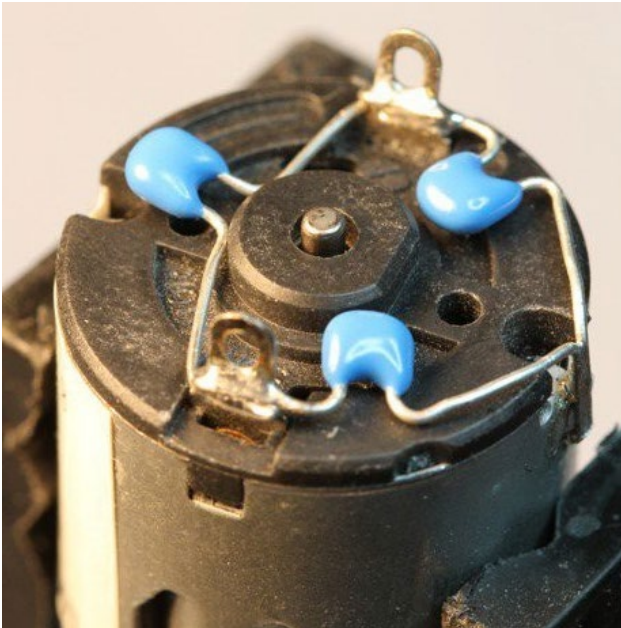
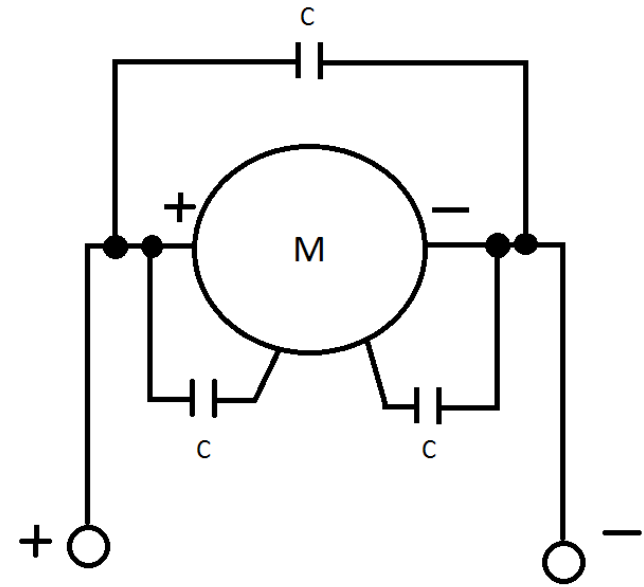
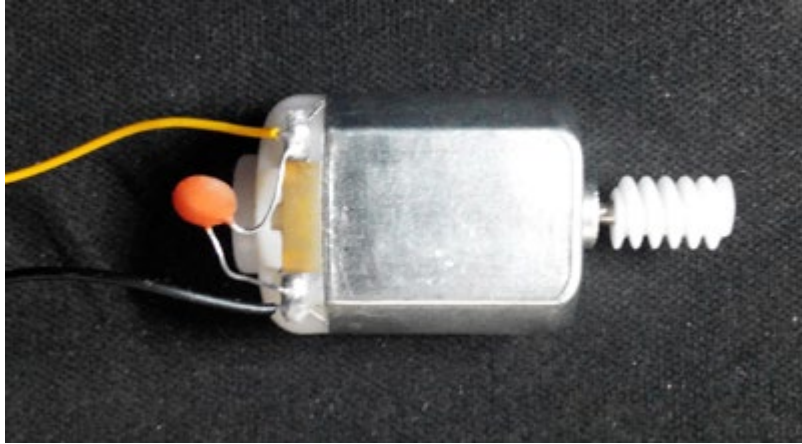


# 기본적인 DC 모터 제어 회로



C1: 스파크 방지용 커패시터  
D1: 역기전력 제거용 다이오드

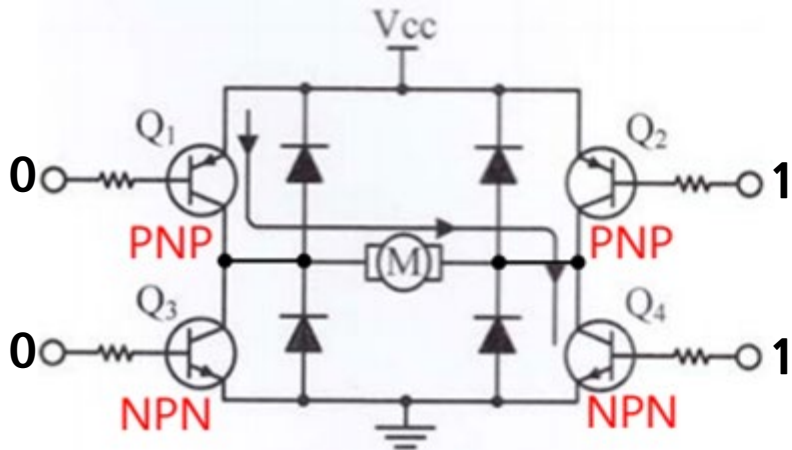
# 스파크 방지용 커패시터



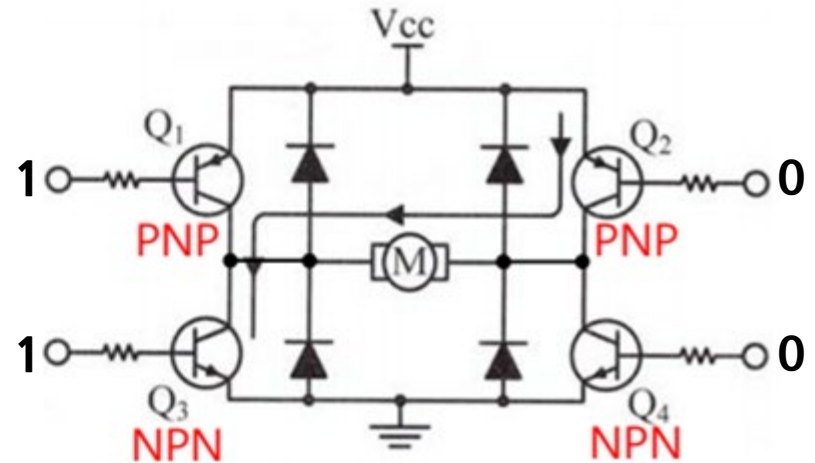


# DC 모터의 양방향 제어 회로

- H 브리지(H bridge) 회로를 이용해 DC 모터를 양방향으로 제어할 수 있음.
  - DC 모터는 전류 방향에 따라서 회전 방향이 달라지는데, H 브리지 회로를 이용하면 모터의 전류 방향을 바꿀 수 있음.



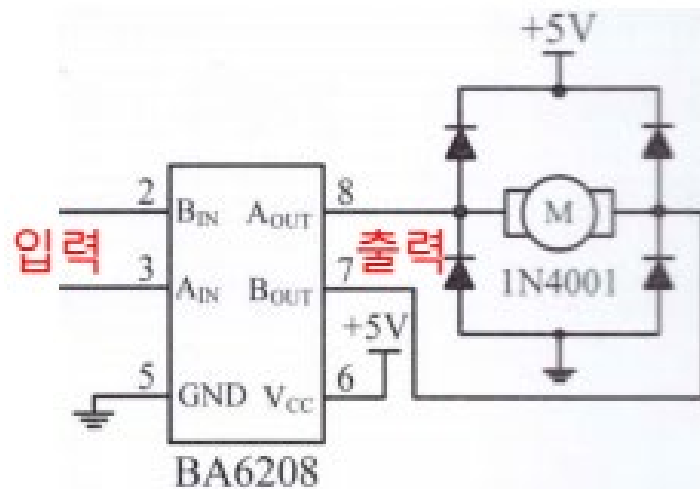
(a) 모터 정방향 회전



(b) 모터 역방향 회전

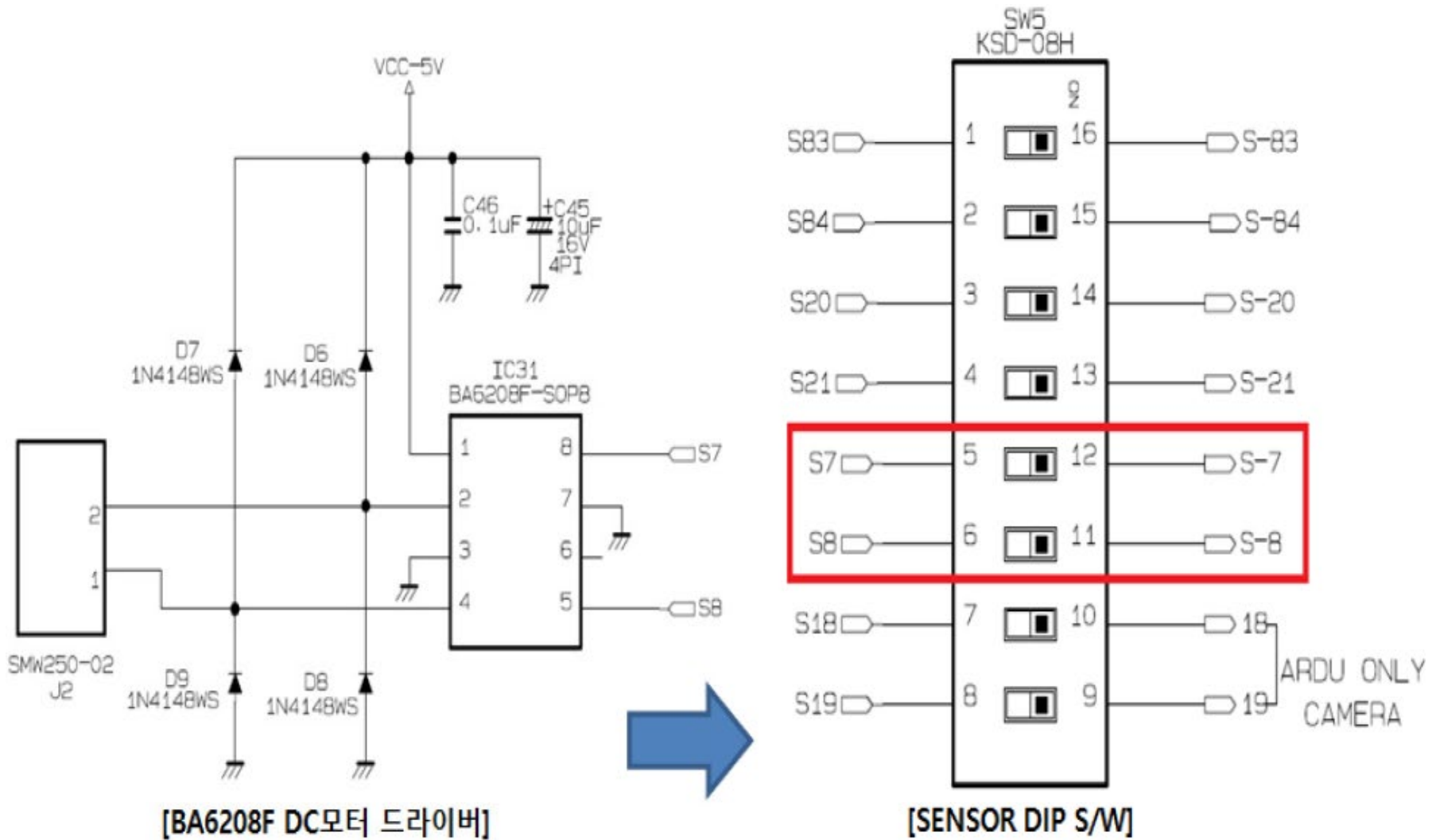
# DC 모터 드라이버 IC를 이용한 모터 제어

- DC 모터를 제어하기 위한 회로를 단일 IC 형태로 만든 것을 DC 모터 드라이버 IC라고 함.
- 키트에서는 BA6208을 이용함.

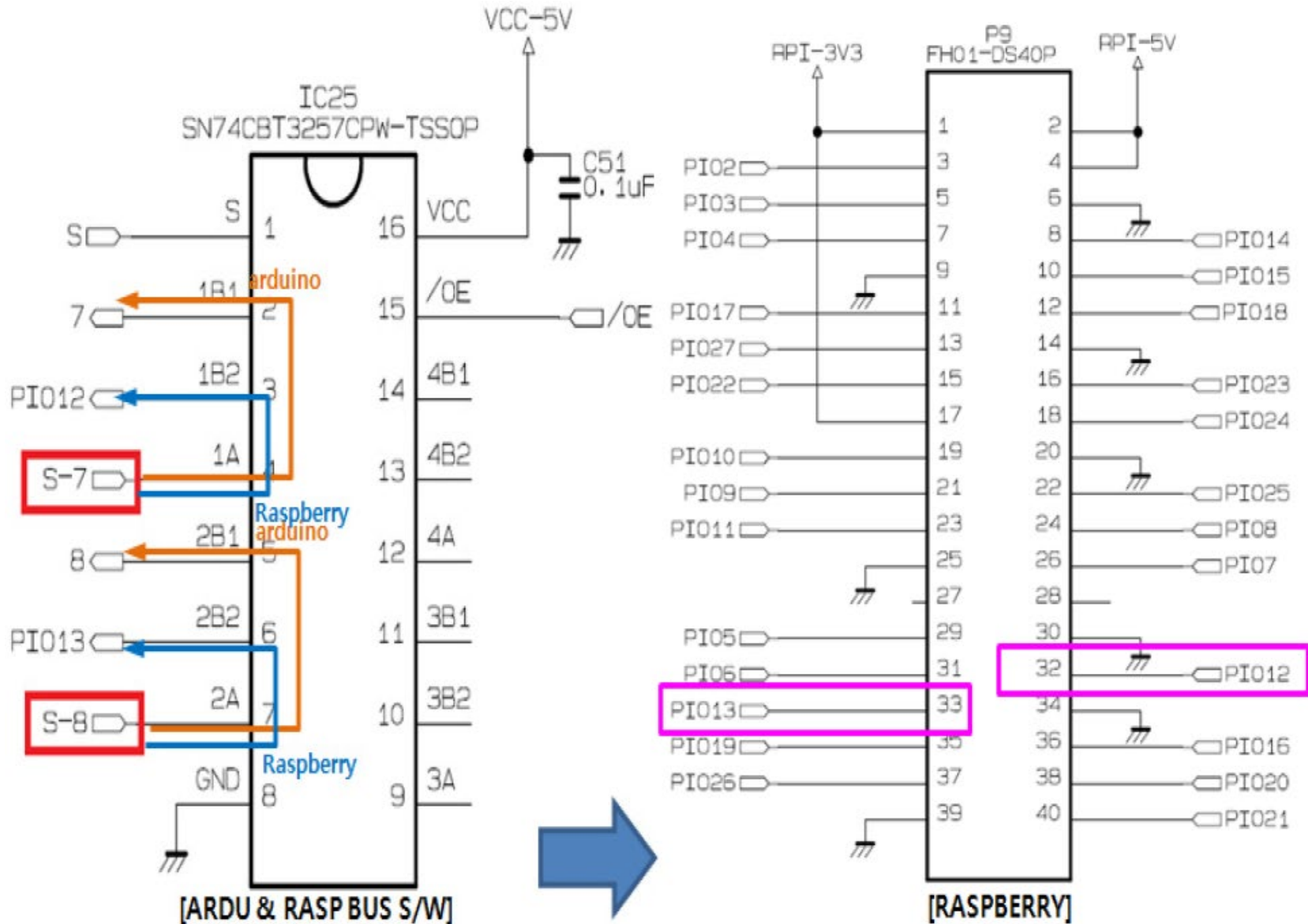


입력		출력		모터 동작
A <sub>IN</sub>	B <sub>IN</sub>	A <sub>OUT</sub>	B <sub>OUT</sub>	
High	Low	High	Low	정방향 회전
Low	High	Low	High	역방향 회전
High	High	Low	Low	브레이크 정지
Low	Low	open	open	정지

# 키트의 DC 모터 회로 연결 (1/3)



# 키트의 DC 모터 회로 연결 (2/3)



# 키트의 DC 모터 회로 연결 (3/3)

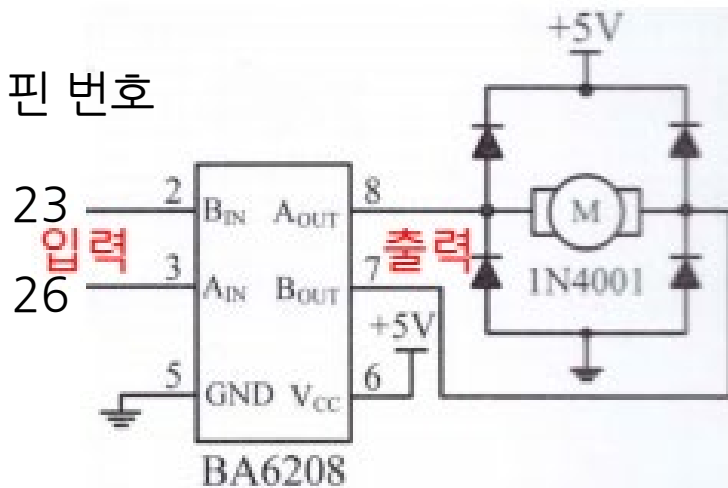
$B_{IN}$

5	21	GPIO.21	IN	1	29	30			0v		
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34			0v		
19	24	GPIO.24	IN	1	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21

$A_{IN}$

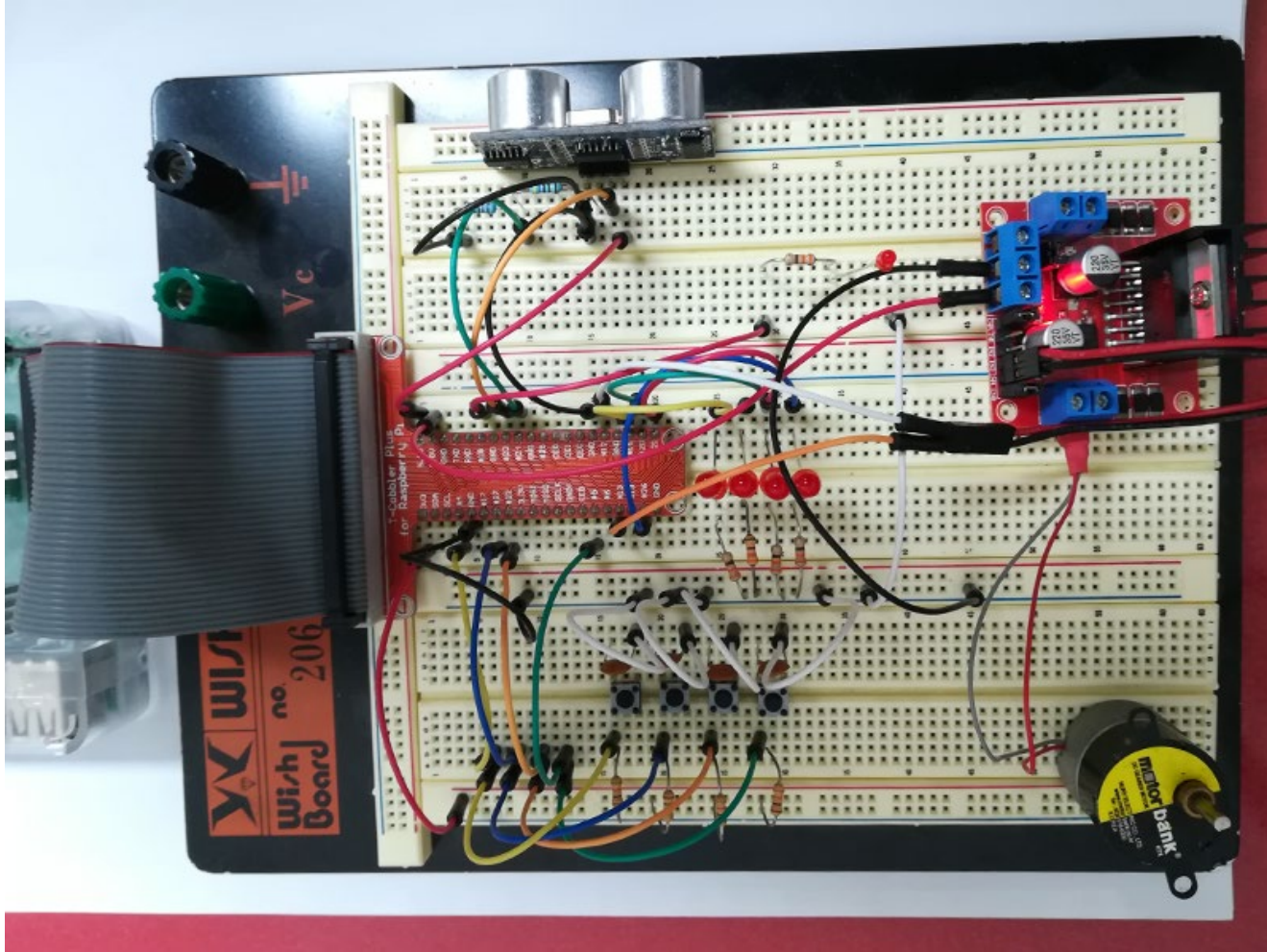
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
-----Pi 3-----										

wPi 핀 번호





# 브레드보드를 이용하는 경우



# 예제 1

- 1초 간격으로 아래와 같이 DC 모터 제어하기
  - 정방향 회전 → 역방향 회전 → 정지 → 이후 같은 패턴으로 반복

```
#include <wiringPi.h>

int main()
{
    const int motor_a=26;
    const int motor_b=23;

    wiringPiSetup();

    pinMode(motor_a, OUTPUT);
    pinMode(motor_b, OUTPUT);
    digitalWrite(motor_a, LOW);
    digitalWrite(motor_b, LOW);
```

```
    while(1){
        digitalWrite(motor_a, HIGH);
        digitalWrite(motor_b, LOW);
        delay(1000);

        digitalWrite(motor_a, LOW);
        digitalWrite(motor_b, HIGH);
        delay(1000);

        digitalWrite(motor_a, LOW);
        digitalWrite(motor_b, LOW);
        delay(1000);
    }

    return 0;
}
```

# 예제 2

- 파란색 버튼을 한번 누르면 모터 회전, 다시 누르면 모터 정지

```
#include <wiringPi.h>

int main()
{
    const int motor_a=26;
    const int motor_b=23;
    const int button=21;

    wiringPiSetup();
    pinMode(button, INPUT);
    pinMode(motor_a, OUTPUT);
    pinMode(motor_b, OUTPUT);
    digitalWrite(motor_a, LOW);
    digitalWrite(motor_b, LOW);

    int input_prev=HIGH;
    int motor=0;
```

```
    while(1){
        int input_cur=digitalRead(button);

        if(input_prev==HIGH && input_cur==LOW){
            motor=(motor==0?1:0);

            if(motor==1){
                digitalWrite(motor_a, HIGH);
                digitalWrite(motor_b, LOW);
            }
            else{
                digitalWrite(motor_a, LOW);
                digitalWrite(motor_b, LOW);
            }
        }

        if(input_prev!=input_cur)
            input_prev=input_cur;

        delay(20);
    }

    return 0;
}
```



# 질문

---

Q & A