C언어 – HW11

임베디드스쿨2기

Lv1과정

2021. 06. 14

차현호

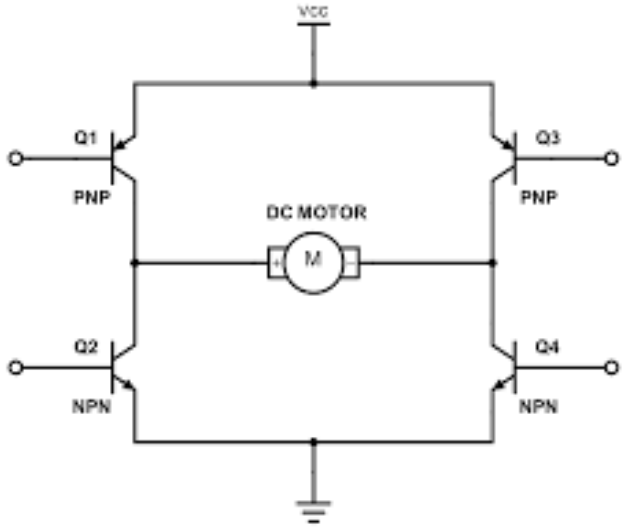
DC Motor Control

DC 모터는 전압 극성과 레벨에따라 속도와 방향이 변화하는 모터이다.
이러한 DC 모터는 아래그림처럼 두개의 선이 나와 있으며 이선에 인가되는 전압의 극성에따라 회전방향이 바뀐다.

DC모터의 방향전환을 위해서 극성을 바꾸어 주어야하는데 이는 H-Bridge라는 회로를 통해 제어할 수 있다.

DC Motor Control

H-Bridge는 아래 그림과같이 2개의 PNP, NPN 트랜지스터로 이루어진 회로이다.



출처 : https://www.build-electronic-circuits.com/h-bridge/

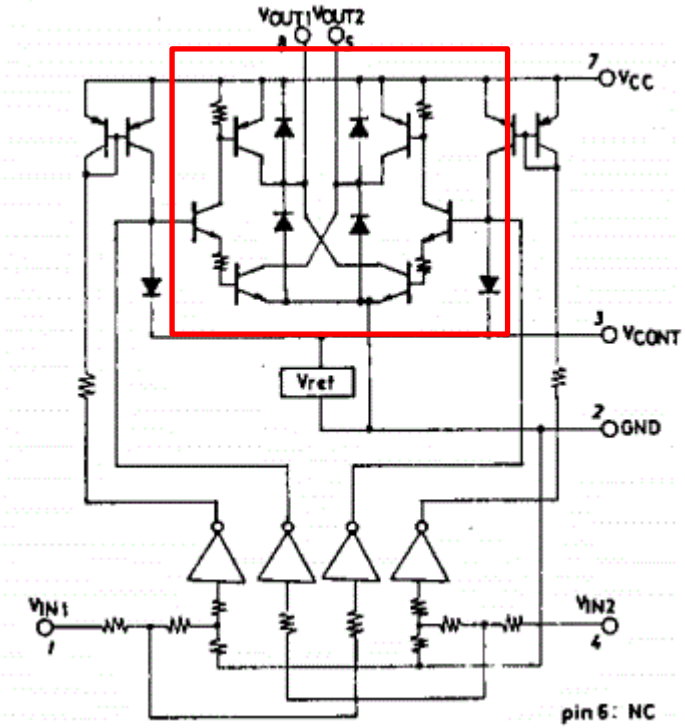
동작을 간단하게 살펴보면 각각 TR을 스위치처럼 사용하는데 양옆의 Base 단에다가 전류를 흘려주어 제어한다. 이러한 회로가 들어가있는 Driver IC 중에 이번 수업중에서는 LB1630이라는 IC를 사용하였다.

DC Motor Control

LB1630의 datasheet를 보면 이전 슬라이드에서 살펴보았던 H-Bridge 회로가 있는것을 볼 수 있다. 또한 Operating condition을 살펴보면 VCC 입력은 2.5V~6V 이며 input 전압레벨이 2V~6V이면 high -0.3V~0.7V이면 Low로 인식한다는것을 확인 할 수 있다.

여기서 input port는 모터의 회전방향을 제어하는 핀이며 아래의 진리표를 통해 확인 할 수 있다.

Equivalent Circuit



Allowble Operating Conditions at Ta=25°C

			unit
Supply Voltage	V_{CC}	2.5 to 6.0	V
Input "H"-Level Voltage	V_{IH}	2.0 to 6.0	V
Input "L"-Level Voltage	V_{IL}	-0.3 to +0.7	V

Truth Table

IN1	IN2	OUT1	OUT2	MOTOR
H	L	H	L	Forward
L	H	L	H	Reverse
H	H	off	off	Standby
L	L	off	off	Standby

DC Motor Control

DC 모터 컨트롤을 위한 아래의 AVR 코드를 살펴보자.

```
void init_dc_motor(void)
{
    DDRB = 0xFF;
    DDRD = 0xFF;

    // Timer/Counter
    // WGM22, WGM21, WGM20 ==> Fast PWM, OCRA
    TCCR2A |= (1 << WGM21) | (1 << WGM20);
    // COM2A1, COM2B1 ==>
    // Clear OC2B on compare match, set OC2B at BOTTOM,
    // (non-inverting mode).
    // 매칭이 되면 클리어
    TCCR2A |= (1 << COM2A1) | (1 << COM2B1);
    TCCR2B |= (1 << WGM22);
    // 64 분주 => 16000000 / 64 = 250000
    TCCR2B |= (1 << CS22);
}
```

위 코드는 DC motor를 init 하는 부분이다 먼저 DDRB = 0xFF, DDRD = 0xFF를 통해 PortB와 PortD의 핀설정을 output으로 설정한다.

DC Motor Control

다음으로 TCCR레지스터의 설정을 살펴보자

TCCR2A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
(0xB0)	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 17-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

위그림에서도 확인 할 수 있듯이 TCCR레지스터를 통해 Timer/Counter Mode를 설정 할 수 있다. 우리의 코드는 WGM0, WGM1을 1로 셋팅하였으므로 Fast PWM mode가 설정 된것을 확인 할 수 있다.

DC Motor Control

다음으로 TCCR2A레지스터에서 COM2A1비트와 COM2B1비트를 1로 셋팅하는 것을 볼 수 있으며 이는 카운터값이 설정한 값에 도달하였을 때 OC2A, OC2B핀을 0으로 하고 Bottom값에서 설정값까지는 1로 출력하는것을 셋팅하였다.

Table 17-3. Compare Output Mode, Fast PWM Mode(1)

COM2A1	COM2A0	Description
0	0	Normal port operation, OC2A disconnected.
0	1	WGM22 = 0: Normal port operation, OC0A disconnected. WGM22 = 1: Toggle OC2A on compare match.
1	0	Clear OC2A on compare match, set OC2A at BOTTOM, (non-inverting mode).
1	1	Set OC2A on compare match, clear OC2A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2A equals TOP and COM2A1 is set. In this case, the compare match is ignored, but the set or clear is done at BOTTOM. See Section 17.7.3 “Fast PWM Mode” on page 122 for more details.

Table 17-6. Compare Output Mode, Fast PWM Mode(1)

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on compare match, set OC2B at BOTTOM, (non-inverting mode).
1	1	Set OC2B on compare match, clear OC2B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 is set. In this case, the compare match is ignored, but the set or clear is done at BOTTOM. See Section 17.7.4 “Phase Correct PWM Mode” on page 123 for more details.

DC Motor Control

마지막으로 TCCR2B 레지스터를 살펴 보면 WGM22 비트와 CS22 비트를 1로 셋팅한것을 볼 수 있다.

우선 CS22비트를 1로 세팅하고 나머지 CS21, CS20을 0으로 세팅하면 현재 시스템 클럭 16MHz를 64분주한 250kHz의 속도로 카운터를 올리겠다는 의미이다.

TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	–	–	**WGM22**	**CS22**	**CS21**	**CS20**	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 17-9. Clock Select Bit Description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{T2S}/(no prescaling)
0	1	0	clk_{T2S}/8 (from prescaler)
0	1	1	clk_{T2S}/32 (from prescaler)
1	0	0	clk_{T2S}/64 (from prescaler)
1	0	1	clk_{T2S}/128 (from prescaler)
1	1	0	clk_{T2S}/256 (from prescaler)
1	1	1	clk_{T2S}/1024 (from prescaler)

DC Motor Control

Motor Init부분 이후에 모터를 직접적으로 컨트롤하는 부분에 대해 살펴보자.

```
// count * PI / 180;
// 1rad = 56.x degree
// 360 degree = 2PI rad
// 1 degree = 2 * PI / 360 rad
//          = PI / 180 rad
// 호도법(각도와 라디안의 관계)
// 이유: math 라이브러리의 sin, cos, tan 등은 radian을 기반으로 동작하기 때문
t = (double)count / 180.0 * 3.141592;
// 255 * (0 ~ 360 degree)
// sin의 특성: -1 ~ 1
// -255 ~ 255
speed = 255 * sin(t);
// speed는 벡터(속도)
velocity = (unsigned int)fabs(speed);
// velocity는 스칼라(속력)
```

먼저 t 변수는 라디안 값을 나타낸다. Count값을 라디안으로 변환하기 위해 1 degree = 2 * π / 360 식을 이용하였다.

이렇게 구한 라디안값 t를 math 라이브러리에 있는 sin()함수에 매개변수로 전달해준다.
이렇게 나온 sin값은 -1~1사잇값을 왔다갔다 하는데 진폭을 255곱해줘 -255~255사이로 확대해준다.
또한 절대값을 함수를 사용하여 속력를 구한다.

DC Motor Control

속도와 속력을 정의한다음에 속도가 0보다 클때(정방향이라고 정의)와 같거나 작을때(역방향이라고 정의)의 조건문을 작성해 주었다.

두개의 조건문을 살펴보면 Port D와 Port B의 값을 앞서 살펴본 LB1630드라이버의 진리표대로 한쪽은 High 다른 한쪽은 Low로 주고 있다.

```
// 양방향
// OCA와 OCB를 통해 정방향, 역방향을 제어함
if (speed > 0)
{
    PORTD |= 0x08;
    PORTB &= ~0x08;
    OCR2A = 255 - velocity;
}
else
{
    PORTD &= ~0x08;
    PORTB |= 0x08;
    OCR2A = velocity;
}

count++;
// 속도의 변화율이 가속도이므로
// 딜레이가 크면 가속도가 작고 작으면 가속도가 크다.
// dv / dt
// (나중 속도 - 이전 속도) / 시간(짧다고 가정) = 근사 가속도
// 이와 같이 디지털로 미적분을 수행하는 기법을 수치해석이라고 한다.
_delay_ms(20);
```

그다음 OCR2A = 255 – velocity 부분은 PWM의 Duty비를 조정하여 DC모터의 RPM을 조절하는 부분이다.

마지막으로 20ms마다 count값을 변화시켜 속도 및 속력을 변화시킨다.

DC Motor Control

앞서 math 라이브러리에 있는 sine 함수의 구현은 어떻게 되어있을지 생각해보자.
Sine함수를 구현할때는 매클로린 급수를 이용하여 구할 수 있다.

매클로린 급수란 x = 0에서 모든 차수의 도함수를 갖는 함수 f가 주어질 때, 함수 f에 의하여 생성되는 매클로린급수는 아래 그림과 같다.

$$f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\cdots+\frac{f^{(n-1)}(0)}{(n-1)!}x^{n-1}+\cdots$$

만약 함수 f(x)가 sinx 함수라면 아래 그림과 같이 표현된다.

$$\sin x=\sum_{n=0}^{\infty}(-1)^n\frac{x^{2n+1}}{(2n+1)!}=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\ \ldots\ (-\infty<x<\infty)$$

위 그림을 보면 맨 오른쪽 급수는 앞서나온 매클로린 급수의 f(x)부분에 sinx를 를 넣었을때 나온 급수이며 이는 일정한 패턴을 보이기때문에 중간식처럼 일반화 할 수 있다.

이렇게 일반화한 식을 이용하여 코드를 작성하여 sin()함수를 구현 할 수 있다.