

Sensorless BLDC모터를 구동해보자!!

작성자 : 허홍강

■ ■ ■ 안녕하세요! 필자는 엔지니어로서 막 발을 들여놓은지 얼마 되지않아, 많은 것을 배우고 있는 '왕초보'입니다!!

이번에 처음으로 Sensorless BLDC 모터 구동에 대한 글을 쓰게 되었습니다. 아주 간단하게 Sensorless BLDC모터를 구동하는 방법에 대해 글을 써볼까 합니다. 부족한 점이 많겠지만, 같이 열심히 배워보도록 합시다.

처음 Sensorless BLDC 모터를 손에 쥐고, 구동하려 하니 막막할 따름입니다. 이녀석을 어떻게 구동시켜야 할까? 과연 구동시킬 수 있을까...? 하는 걱정도 들었습니다.

사실 왕초보인 필자는 어릴 때 가지고 놀던 미니카에 들어가는 DC 모터를 뜯어본 기억이 전부인데 말입니다.

학교 앞 문구점 앞에 설치된 미니카용 트랙(Track)에서 미니카를 친구들 것보다 빠르게 달리게 하기 위해 모터를 구입하던 기억이 떠오르네요. BLDC 모터라... 무언가 DC와는 이름부터 다르니 다른점이 한, 두가지가 아닐 것 같은 느낌이 매우 많이 듭니다. 얼마나 많이 다른걸까요?

자! 그럼 Sensorless BLDC모터를 구동하기에 앞서, BLDC와 Sensorless BLDC에 대해서 알아봐야 하겠죠??

그럼 간단하게 알아보도록 합시다

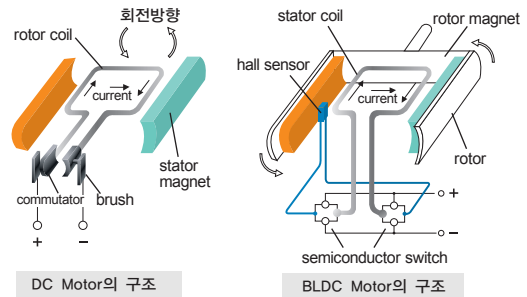
1 BLDC(Brushless DC electric motor)란?

BLDC란 무엇일까요?? 약자를 풀어서 읽어보면, Brush가 없는 DC 모터인 것이 분명합니다. 그렇다면 일반 DC 모터와는 무슨 차이가 있길래 Brush를 왜 없앴을까요? 다시 옛날에 가지고 놀던 미니카의 DC모터를 뜯어보겠습니다.

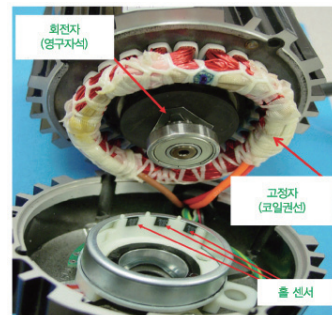
〈한참 달리고 나면 모터가 엄청 뜨겁습니다. 약간의 탄내도 올라옵니다. 모터를 뜯어보니 Brush가 탄 게 분명한 것 같습니다. 이걸 갈아줘야 다시 달릴 수 있을 것 같습니다.〉 다들 미니카를 한 번쯤 다뤄봤다면 이와 같은 일을 겪었을 것입니다.

그렇습니다. DC 모터의 단점은 Brush입니다. 소음이 많고, 일정 시간이 되면 갈아줘야 하기 때문입니다. 이와 같은 단점을 없앤 것이 BLDC(Brushless DC electric motor)인 것입니다. 그렇다면 구조는 어떻게 다른지 살펴보겠습니다.

〈그림1-1〉에서 볼 수 있듯이, DC 모터의 경우 회전체가 코일인 반면, BLDC 모터는 자석자체가 회전하는 회전체의 모습입니다. 그러므로 구조상 Brush가 필요 없어지게 된 것입니다. 그렇다면 실제 BLDC의 모습은 어떻게 생겼을까요?



〈그림1-1〉 DC와 BLDC의 구조



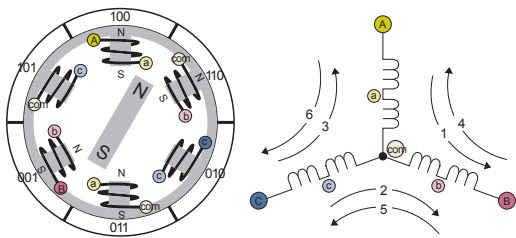
〈그림1-2〉 실제 BLDC모터 실제모습

<그림 1-2>을 보시면 <그림 1-1>에서 봤던 것처럼 코일에 전류가 흐르면 자석이 회전하게끔 만들어져 있습니다.

그리고 3개의 극성 터미널이 존재합니다 U,V,W(A,B,C) 가 바로 그것이지요.

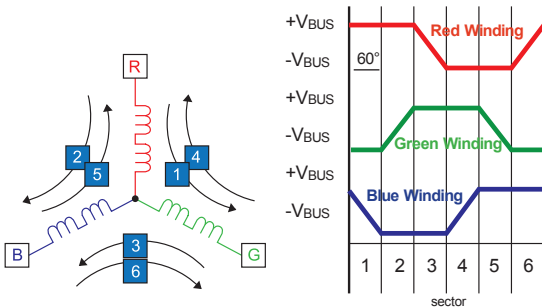
그리고 모터 회전체인 자석의 위치를 알려주는 홀 센서가 장착되어 있습니다. 이 홀 센서를 통해 모터자석의 위치를 판단할 수 있고, 모터의 회전속도 및 회전각 등을 알 수 있는 것입니다.

여기서 궁금한 점이 있을 것입니다. DC모터와 BLDC가 구조적인 차이점이 있으니, 구동하는데도 각각의 차이점이 있지 않을까요? 물론 그렇습니다. 일반 DC 모터의 경우 Brush의 양 터미널에 (+,-)를 입력하면 간단하게 구동이 되지만, BLDC 모터의 경우 홀 센서의 출력에 따라 각각의 터미널에 전류를 흘려주어야 하므로 반드시 구동용 드라이버가 필요합니다.



<그림1-3> BLDC모터의 내부모습

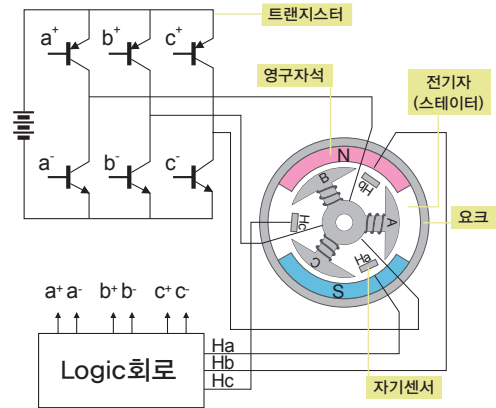
<그림 1-3>은 일반적인 BLDC 모터의 내부구조입니다. BLDC모터는 영구자석으로 이루어진 회전자와 권선으로 이루어진 스테이터들로 이루어져 있으며, 전류가 흐르는 권선으로부터 자기장 관계에 의해 회전자를 회전시켜 전기적 에너지를 기계적인 에너지로 변환되게 됩니다.



<그림1-4> 홀 센서 신호에 따른 전압인가 방법

<그림 1-3>의 오른쪽에 나타난 화살표의 방향은 각각의 전류 흐름 방향이고 이를 좀더 자세히 표현한 것은 아래 <그림 1-4>와 같습니다.

<그림 1-4 >와 같이 BLDC모터는 총 6-STEP으로 이루어져 있으며, 홀 센서의 신호에 따라 각각 다른 방향으로 전압을 인가해 주어야 합니다. 위와 같이 전류를 흘려주게 되면, 모터는 회전체(영구자석)가 순서대로 각 전자석에게 밀리고 당겨져 회전체가 회전하게 되는 것입니다.



<그림1-5> 3상 전류 제어용 인버터

<그림1-5>는 3상 전류제어용 인버터로, 위와 같은 회로 조합을 통해 BLDC모터를 제어하게 됩니다.

여기까지 매우 간단하게 BLDC에 대해 알아보았습니다. 다음 장에서는 Sensorless BLDC에 대해 알아보도록 하겠습니다.

2 Sensorless BLDC란?

드디어 Sensorless BLDC 모터에 대해 알아보는 시간이 되었습니다! Sensorless BLDC모터란 무엇인가? 단순히 해석하고자 하면, BLDC모터에 Sensor가 없는 것이므로, 홀 센서가 없는 것일까? 그렇습니다. 홀 센서가 장착되지 않은 BLDC모터입니다. 그렇다면 왜 홀 센서를 장착하지 않은 것일까요? 필자의 생각으로는, 소형 BLDC모터를 개발하다 보니 홀 센서까지 제거하여 크기를 줄이다가 개발된 것이 아닐까 생각이 듭니다.

즉, 기존의 BLDC모터에 비해 크기를 줄이기 위한 것이 아닐까 조심스럽게 추측해 봅니다. 본론으로 넘어와서, Sensorless BLDC

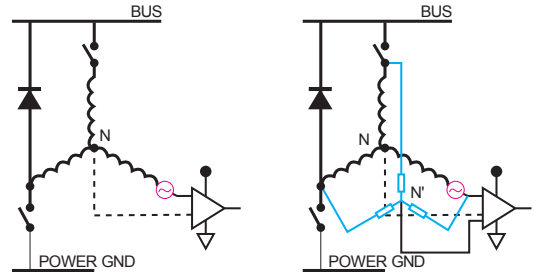
모터를 구동하기 위해서는 무엇을 알아야 할까요? 기존 BLDC 모터와 구동하는데 있어서 차이점은 무엇일까요? 그럼 앞서간 다른 BLDC모터에 대해 다시 한번 떠올려 볼까요? 기존 BLDC모터의 경우에는, 각 홀 센서의 출력 신호를 가지고, 모터의 회전자(영구자석)의 위치를 파악하여 3개의 터미널에 순서대로 각기 다른 전압을 인가하여 회전자와 전자석 사이의 밀고 당기는 힘을 이용하여 회전하였습니다. 그렇다면, "Sensorless BLDC모터의 경우 홀 센서가 없으면, 모터의 회전체의 위치는 어떻게 알고 전압을 순서대로 인가하지? 홀 센서가 없는데 회전체의 위치를 알고 모터에 정확한 신호를 준다? 어떤 방법일까?" 하는 의문이 생길 것입니다.

일단 결론부터 얘기하면, 역기전력(BACK EMF)을 이용하는 방법입니다. BLDC 모터가 회전할 때, 각 권선은 렌츠의 법칙(Lenz's Law)에 따라서 각 권선은 인가되는 주 전압과 반대의 극성을 갖는 역기전력(BEMF)을 발생시키게 됩니다. 이들 역기전력의 극성은 인가 전압의 방향과 반대가 되며, 주로 3개의 큰 모터 파라미터에 의해 결정됩니다.

- 고정자 권선의 턴 수
- 회전자의 각 속도
- 회전자 자석에 의해 발생하는 자계

(더 많은 내용이 있지만, 간단하게 구동하는 방법만을 소개하므로, 복잡한 내용은 생략하도록 하겠습니다.)

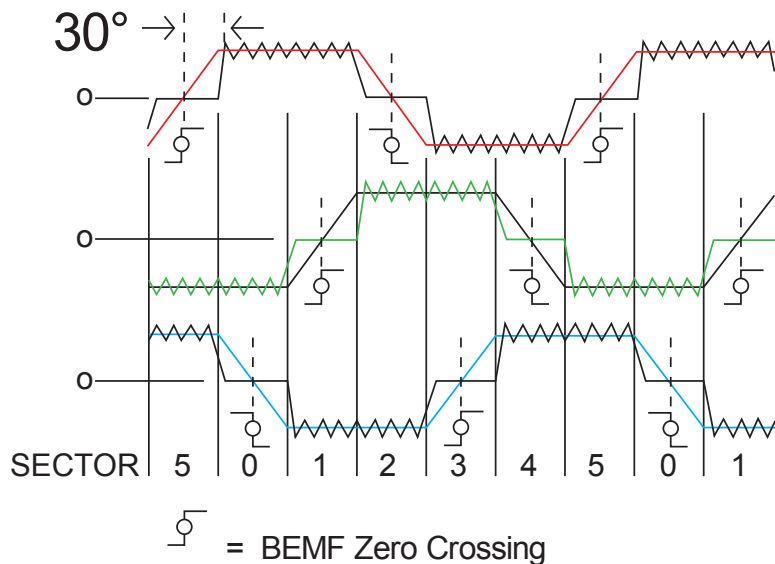
〈그림 2-1〉은 각각의 U,V,W (이하A,B,C) 에 순서대로 전압을 인가했을 때 발생하는 Back-EMF Zero Crossing의 위치를 나타낸 것입니다. 그렇다면 역기전력은 어떻게 이용하느냐가 문제인데, 보통 Sensorless BLDC 모터를 접하면, 중성점(Neutral Point)이 나와있는 것과 그렇지 않은 것들 두 가지를 접하게 됩니다.



〈그림2-2〉 중성점이 나와있는 경우 (우), 중성점이 없는 가상의 중성점인 경우(좌)

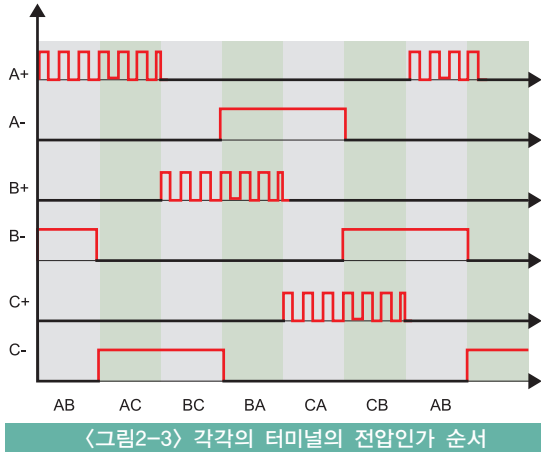
〈그림2-2〉에 나와있는것 같이 왼쪽의 중성점이 있는 경우와, 오른쪽처럼 중성점이 없는 경우 두 가지의 방법이 있습니다.

왼쪽과 같은 경우에는 중성점을 이용하여, 직접 Back-EMF를 검출할 수 있는 반면, 오른쪽과 같이 중성점이 나와있지 않은 경우, 가상의 중성점을 구성하여 간접적으로 검출하여야 합니다. 그리고 간접적으로 검출한 Back-EMF와 각각의 U,V,W (이하A,B,C) 와의 비교기를 통해 홀센서와 유사한 출력을 얻고, 그 출력에 따라



〈그림2-1〉 Zero Crossing 검출

순서대로 전압을 인가하게 되는 것입니다.



드디어 BLDC모터와 Sensorless BLDC모터에 대해 간단하게 알아본 것 같군요!!

다음장에서는 제가 수행한 방법을 소개하도록 하겠습니다.

3 Sensorless BLDC 모터 구동

1장과 2장을 통해 간략히 BLDC모터와 Sensorless BLDC 모터에 대해 알아보았습니다. 1장과 2장을 통해 제 설명이 잘 전달 되었는지 궁금하군요!!

이제 BLDC 모터를 구동할 시간입니다. 그럼 저와 함께 Sensorless BLDC모터 구동을 해보도록 할까요??



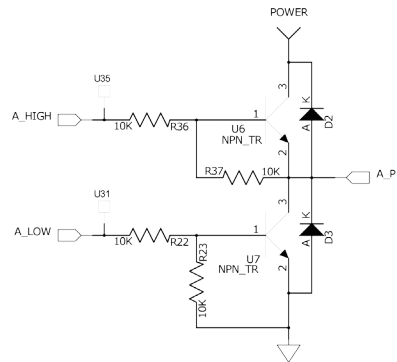
우선 Sensorless BLDC 모터를 구동하기에 앞서 모터를 선정해 볼까요?? 필자는 HDD Spindle Sensorless BLDC 모터를 선택하겠습니다. 종종 각 포탈게시판에 하드드라이버를 분해한 뒤 모터 돌리는 동영상을 보고, 한번 해 봐야지 했었는데, 드디어 해보게

되는군요!! 모터의 배선을 보니, 4선으로써 3선은 모터의 각 상인 U,V,W(A,B,C) 임에 틀림없고, 나머지 한 선은 중성점일 것입니다. 이 모터의 데이터 시트를 찾아봤지만 인터넷에는 데이터시트가 존재하지 않는군요! 어떤 터미널이 중성점인지 어떻게 알아낼까? 생각끝에 멀티메터로 찍어보았습니다.

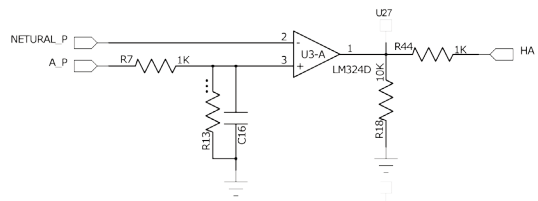
각각의 터미널끼리 저항값을 재보니, 한 터미널에서 2.9Ω이 나왔습니다. 만약 중성점일 경우, 각각의 터미널선 가운데 있으므로,



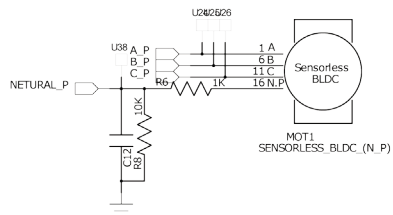
저항성분이 절반 밖에 되지 않을 것입니다. 그러므로 분명 이 녀석이 중성점일 것입니다. 이제 중성점의 위치를 알았으므로, 모터를 구동시킬 모터 구동부가 있어야 할 것입니다. 이 모터를 만능기판에 놓고 테스트해 본 결과, 큰 전류량을 필요로 하는 모터는 아니므로, 간단하게 범용 npn 트랜지스터인 2N2222을 사용하여 브릿지 회로를 구성하였습니다. 그리고, Back-EMF를 검출하기 위해 LM2904를 사용하여 비교기 회로를 구성하였습니다. (필자는 메인 MCU로 ATmega8을 사용하였습니다.)



<그림3-1> 브릿지 회로

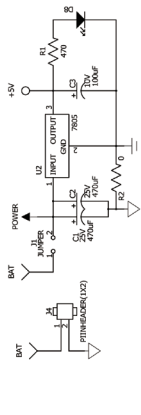


<그림3-2> Back-EMF 검출용 비교기 회로

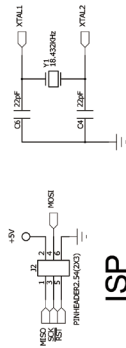


<그림3-3> 모터 연결 회로

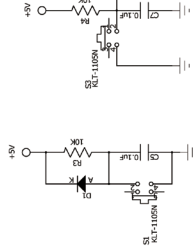
회로도 (다운로드 : <http://forum.ntrex.co.kr/>)



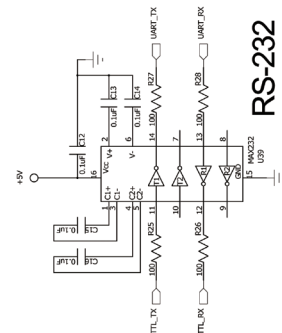
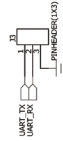
POWER MANAGEMENT



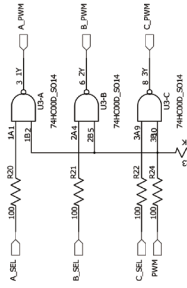
ISP



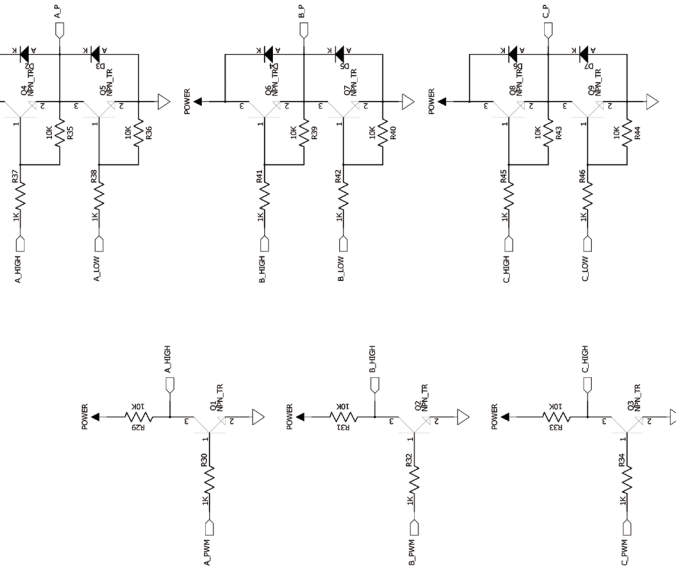
RESET



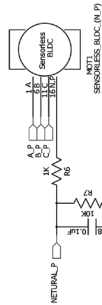
RS-232



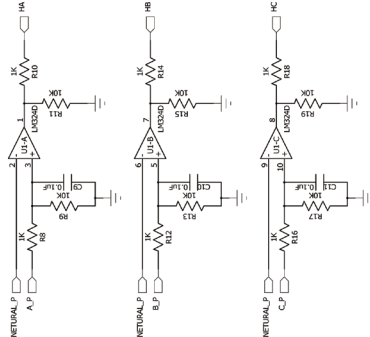
PWM DIVISION



SPINDLE MOTOR



VIRTURE MIDDLE POINT

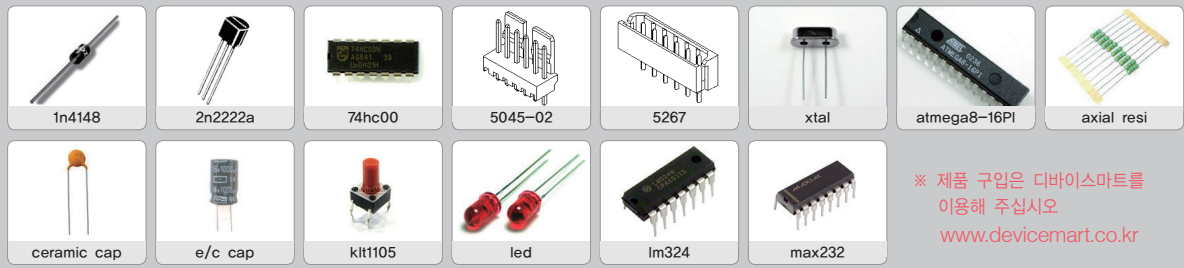


MCU

COMPANY:	NTREX	
TITLE:	NT-SpindletestKIT	
CODE:	SIZE:	REV:
DRAWING NO.		

NT - SpindletestKIT Parts List

	Part	Package	Description	EA	Location	디바이스마트 제품명
IC	ATmega8-16PI	DIP	MCU	1	U4	ATmega8-16PI
	KIA7805	DIP	Line driver	1	U39	MAX232N
	LM324D	DIP	OP-AMP	1	U1	LM324N
	74HC08D	DIP	NAND GATE	1	U3	74HC00
X-TAL	CRYSTAL - 18.432MHz	ATS	18.432MHZ	1	Y1	X-TAL 18.432M (49/U Type) Fundamental
Regulator	7805	TO-220	5V Regulator	1	U2	7805(TO-220) 방열판 세트
DIODE	1N4148	DIP	DIODE	7	D1,D2,D3,D4,D5,D6,D7	1N4148
	3파이 LED RED	DIP	LED	1	D8	3A4SCA-8
SW	KLT-1105B	DIP	SWITCH	2	S1,S3	KLT-1105B
Connector	Molex 53014-02	DIP	molex 53014 - 2PIN	1	J4	Molex 53014-02
	Molex 5045-02	DIP	molex 5045 - 2PIN	1	J1	Molex 5045-02
	PINHEADER 2X3	DIP	2.54mm	1	J2	핀헤더
	PINHEADER 1X3	DIP	2.54mm	1	J3	핀헤더
MOTOR	SPINDLE MOTOR		Sensorless BLDC	1		HDD Spindle Sensorless BLDC
	Molex 5267-04		Motor Con	1	MOT1	Molex 5267-04
Transistor	2N2222A	TO-92	NPN TR	9	Q1,Q2,Q3,Q4,Q5, Q6,Q7,Q8,Q9	2N2222A
Resistor	1/4W Axial Resistor 471J	DIP	470Ω	1	R1	1/4W Axial Resistor 471J
	1/4W Axial Resistor 101J	DIP	100Ω	8	R20,R21,R22,R24,R25, R26,R27,R28	1/4W Axial Resistor 101J
	1/4W Axial Resistor 1K	DIP	1KΩ	17	R5,R6,R8,R10,R12,R14, R16,R18,R30,R32,R34, R37,R38,R41,R42,R45, R46	1/4W Axial Resistor 1K
	1/4W Axial Resistor 10K	DIP	10KΩ	19	R3,R4,R7,R9,R11,R13,R15,R1 7,R19,R23,R29,R31,R33,R35, R36,R39,R40,R43,R44	1/4W Axial Resistor 10K
	1/4W Axial Resistor 0	DIP	0Ω	1	R2	1/4W Axial Resistor 0
CAP	E/C 25V 470uF	DIP	470uF E/C	2	C1,C2	E/C 25V 470uF (85°C)
	E/C 10V 100uF	DIP	100uF E/C	1	C3	E/C 10V 100uF (85°C)
	Ceramic 0.1μF, 50V	DIP		14	C5,C7,C8,C9,C10,C11, C12,C13,C14,C15,C16, C17,C18,C19	Ceramic 0.1μF, 50V
	Ceramic 22pF, 50V	DIP		2	C4,C6	Ceramic 22pF, 50V



※ 제품 구입은 디바이스마트를 이용해 주십시오
www.devicemart.co.kr

소스코드

개발툴 – AVR GCC

```

#include <avr/io.h>
#include <stdio.h>
#include <avr/interrupt.h>
#include <util/delay.h>

#define A_H_ON PORTC = ((PORTC | 0x01) & 0xF9)
#define B_H_ON PORTC = ((PORTC | 0x02) & 0xFA)
#define C_H_ON PORTC = ((PORTC | 0x04) & 0xFC)

#define A_L_ON PORTC = ((PORTC | 0x08) & 0xCF)
#define B_L_ON PORTC = ((PORTC | 0x10) & 0xD7)
#define C_L_ON PORTC = ((PORTC | 0x20) & 0xE7)

void motor(void);

//main
volatile unsigned char g_Motor_state;

//pwm
volatile unsigned int g_Spindle_speed;
volatile unsigned int g_Spindle_speed_renewal;

int main(void)
{
    //GPIO Initializing
    DDRC |= (1<<DDC0)|(1<<DDC1)|(1<<DDC2)|(1<<DDC3)|(1<<DDC4)|(1<<DDC5);
    PORTC &= ((0<<PORTC0)|(0<<PORTC1)|(0<<PORTC2)|(0<<PORTC3)|(0<<PORTC4)|(0<<PORTC5));
    DDRD &= (0<<DDD2)|(0<<DDD3)|(0<<DDD4);
    PORTD |= (1<<PORTD2)|(1<<PORTD3)|(1<<PORTD4);

    DDRD &= (0<<DDD5); //START Switch
    PORTD |= (1<<PORTD5); //START Switch

    DDRB |= (1<<DDB1); //PWM
    PORTB |= (1<<PORTB1); //PWM

    //PWM Initializing
    TCCR1A = (1<<COM1A1)|(1<<COM1B1)|(1<<WGM11); //9bit phase correct PWM 15.6kHz
    TCCR1B = (1<<CS10); //1 prescaler
    TIMSK |= (1<<OCIE1A);

    sei();

    g_Spindle_speed_renewal = 200;

    while(1)
    {
        if((PIND&0x20)==0)
        {
            g_Spindle_speed = 250;
            g_start_flag = 1;
        }

        if(g_Spindle_speed>510)
        {
            g_Spindle_speed = 510;
        }
        if(g_Spindle_speed<100)
        {
            g_Motor_state = 0;
            OCR1A = 0;
        }
        motor();
    }
    return 0;
}

void motor(void)

```

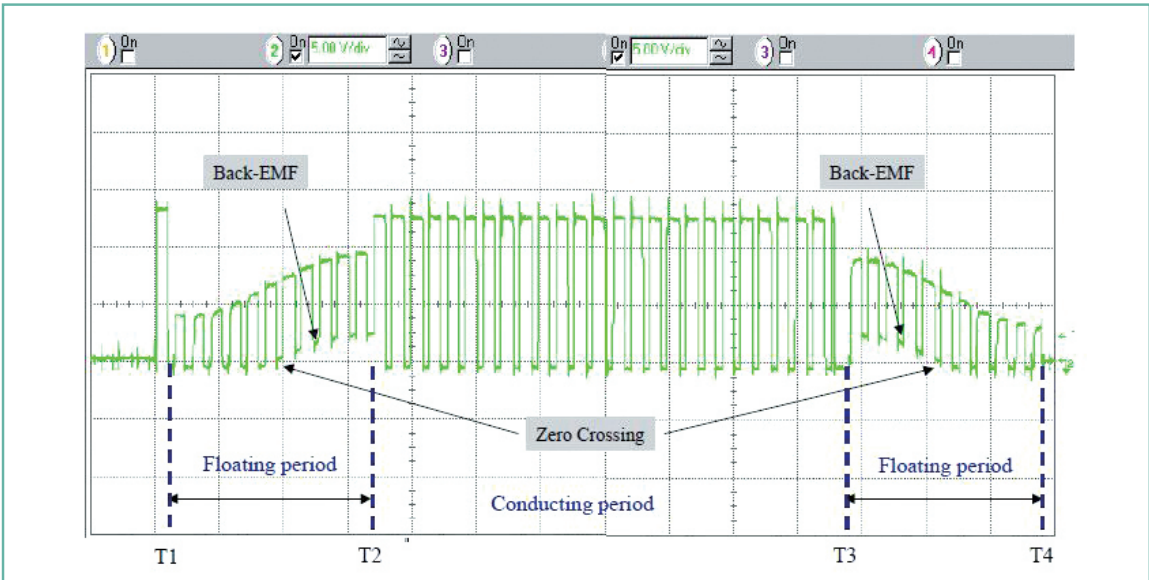
```

{
    switch(g_Motor_state)
    {
        case 0:
            if(g_Spindle_speed>100) g_Motor_state = 1;
            break;
        case 1:
            OCR1A = 200;
            if(((PIND & 0x1C)>>2) == 3)g_Motor_state = 2;
            A_H_ON;
            C_L_ON;
            _delay_ms(15);
            if(((PIND & 0x1C)>>2) == 1)g_Motor_state = 2;
            B_L_ON;
            _delay_ms(15);
            if(((PIND & 0x1C)>>2) == 5)g_Motor_state = 2;
            C_H_ON;
            B_L_ON;
            _delay_ms(15);
            if(((PIND & 0x1C)>>2) == 4)g_Motor_state = 2;
            A_L_ON;
            _delay_ms(15);
            if(((PIND & 0x1C)>>2) == 6)g_Motor_state = 2;
            B_H_ON;
            A_L_ON;
            _delay_ms(15);
            if(((PIND & 0x1C)>>2) == 2)g_Motor_state = 2;
            C_L_ON;
            _delay_ms(15);
            break;
        case 2:
            OCR1A = 300;
            if(((PIND & 0x1C)>>2) == 1)
            {
                A_H_ON;
                B_L_ON;
            }
            if(((PIND & 0x1C)>>2) == 2)
            {
                B_H_ON;
                C_L_ON;
            }
            if(((PIND & 0x1C)>>2) == 3)
            {
                A_H_ON;
                C_L_ON;
            }
            if(((PIND & 0x1C)>>2) == 4)
            {
                C_H_ON;
                A_L_ON;
            }
            if(((PIND & 0x1C)>>2) == 5)
            {
                C_H_ON;
                B_L_ON;
            }
            if(((PIND & 0x1C)>>2) == 6)
            {
                B_H_ON;
                A_L_ON;
            }
            break;
    }
}

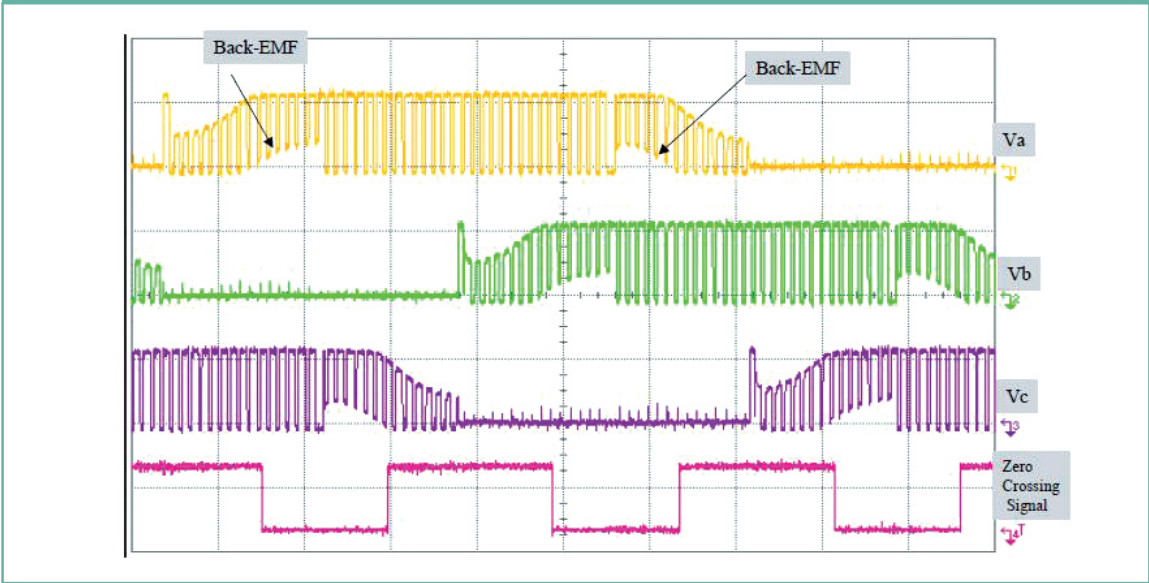
ISR(TIMER1_COMPA_vect){

```

· if(((PIND & 0x1C)>>2) == 3) : HA, HB, HC가 각각 PORTD의 2,3,4 핀에 물려있으므로, 각각의 상태를 확인하기 위해서는 0x1C와 & 연산을 해 주어야 한다. 또한 >> 2는 오른쪽으로 2번 쉬프트 연산을 통해 각각의 상태가 1~6까지의 6스텝 상태를 확인 할 수 있다.



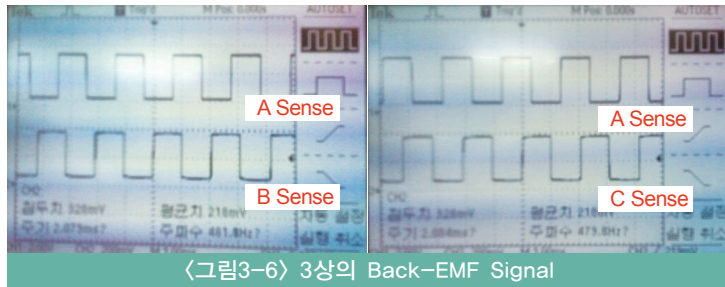
〈그림3-4〉 각 모터 터미널전압과 Back-EMF



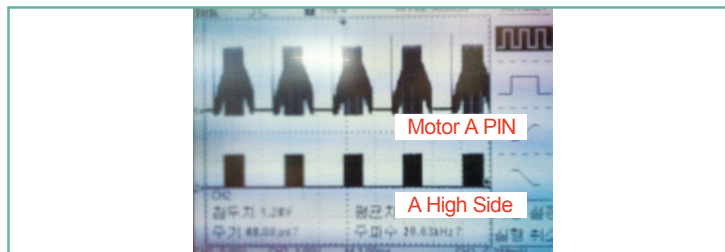
〈그림3-5〉 3상의 Back-EMF와 Zero Crossing

우선 Sensorless BLDC모터를 구동하게 되면 각각의 ABC파형은 〈그림3-5〉와 같이 Zero Crossing과 Back-EMF가 발생되는 floating 구간 사이에 Conducting 구간이 존재할 것입니다. 〈그림 3-1,2,3〉과 같이 회로는 구성했지만, 막상 Signal을 주려고 하니 막막합니다. Back-EMF 검출용 회로를 구성했는데 어떻게 써야 하나 조사해본 결과 매우 다양한 방법이 있었습니다. 그 중 가장 간단한 가정을 하나 세워 보았습니다.

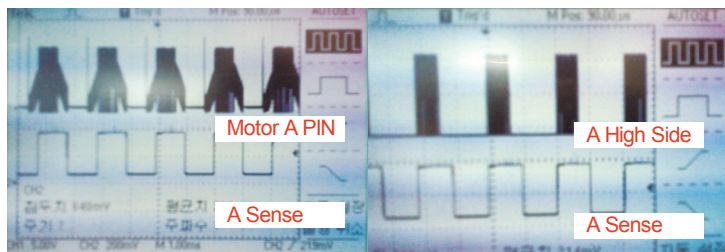
〈그림3-5〉와 같이 처음 모터를 돌리기 전에 일정 방향으로 6-Step의 펄스를 순서대로 넣어준다면, Hall Sensor Type의 BLDC 모터 처럼 Back-EMF 검출용 비교기 회로에서 Signal을 발생시킬 것이고 어느 순간 6-Step의 어느 한 스텝과 정확히 일치하는 부분이 있을 것입니다. 그 때부터 6-Step에 맞춰 Signal을 입력한다면, 모터는 제대로 회전할 것입니다. 그럼 테스트 결과를 보도록 할까요?



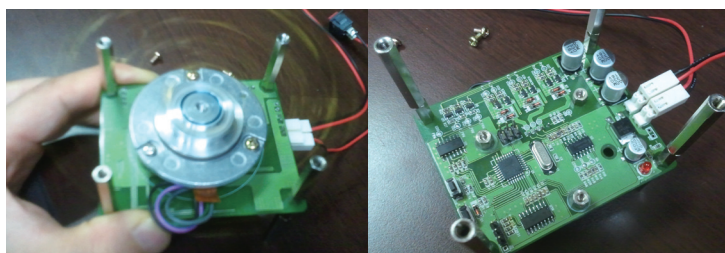
〈그림3-6〉 3상의 Back-EMF Signal



〈그림3-6〉 모터의 A terminal과 브릿지회로의 A high side input



〈그림3-7〉 모터의 A terminal, 브릿지회로의 A high side input, Back-EMF 검출기의 A signal



〈그림3-8〉 모터의 구동모습(좌), 테스트를 위해 제작한 모터 드라이버 (우)

〈그림3-6〉은 Back-EMF 검출회로를 통해 발생하는 출력 화면입니다. A,B,C 각각의 Back-EMF는 60° 를 간격으로 순차적으로 발생됨을 확인 할 수 있습니다. 그렇다면 이 Signal에 맞춰서 6-Step펄스를 입력해 준다면, 모터는 정상적으로 회전할 것입니다.

〈그림3-6〉을 살펴보면, A의 High side 입력에 따라 모터 A Terminal에는 〈그림3-4〉과 같이 Back-EMF와 Zero Crossing이 일어나는 매우 유사한 펄스를 확인할 수 있었습니다.

〈그림3-7〉을 통해 확인한 결과, A 펄스가 입력될 때, A의 Back-EMF 비교기 출력이 출력됨을 확인할 수 있었습니다.

〈그림3-8〉은 제가 구동시킨 모터가 잘 돌아가는 사진과, 모터를 돌리기 위해 제작한 간단한 회로의 모터 드라이버입니다. 간단하게 글을 쓰다 보니 생략하고 넘어간 부분이 많은 것 같습니다.

하지만 독자 여러분들 모두 Sensorless BLDC 모터를 구동할 수 있으리라 믿습니다. 여기까지 왕초보의 “Sensorless BLDC 모터를 구동하자” 를 읽어주신 모든분께 감사의 인사를 올립니다. 감사합니다.

〈사진〉(c)엔티렉스 〈저작권〉(c)엔티렉스, 무단 전재-재배포금지 허홍강 연구원(hhkgkd@ntrex.co.kr)

※참고문헌

- Microchip AN1160 (Sensorless BLDC Control with Back-EMF filtering Using a Majority Function) -2009 Microchip Technology Inc.
- Direct Back EMF Detection Method for Sensorless Brushless DC(BLDC) Motor Drives - Jianwen Shao (Virginia Polytechnic Institute and the State University -September, 2003)

※디바이스마트를 통해서 모터구동용 테스트 기판을 판매할 예정입니다.