마이크로프로세서 응용 보고서
(DC모터제어 전압지령&전류센싱)

자동차IT융합학과 20183376 박선재

```
/*전류제어만 있는 코드*/
#include "MPC5604P_M26V.h"
#include "freemaster.h"
#include "init_base.h"

// Macros for cleaner code
#define LPF(out, in, fct)  out = out-in>0 ? fct*(out-in)+ in :-fct*(in-out)+ in
#define FILFCT(wc, fct, ts)  fct =  ts/(wc+ts)
#define bound(in,lim)  ((in > (lim)) ? (lim) : ((in < -(lim)) ? -(lim) : in));
#define abs(x) ((x > 0) ? x : -x)
#define rad2rpm 9.54929
#define TWO_PI6.28316

// Global Variable
volatile int temp10ms=0;
int Reset = 1, DSPRUN = 0, SpdCon = 0;
float IdcPre=0.0f, IdcErr=0.0f, IdcRef=0.0f, IdcFdb=0.0f, IdcOffset=0.0f;
float VdcRef=0.0f, VdcPterm=0.0f, VdcPiterm=0.0f, VdcIterm=0.0f;
float VdcFil=0.0f, VdcPre=0.0f, vRefUlim=0.0f, PWM_Scale=0.0f;
float Kpc=0.0f, Kic=0.0f;
int PWM_B=0, PWM_Peak=0;
long int delta_m=0, mold=0, mnew = 0;
float rpm = 0., rpmFil = 0., wrTemp = 0., wr = 0., wrFil = 0., rpmRef = 0., rpmErr
=0.;
float spdPterm = 0., spdPiterm = 0., spdIterm = 0., spdPiUlim = 0., spdPiOut = 0.;
float kpSpd = 0., kiSpd = 0.;
float  iScale=0.0f,  VdcScale=0.0f,  FadcScale=0.0f,  FvdcHwScale=0.0f,  FiHwScale=0.0f,
SpeedScale=0.0f, MotorEncPulse=4.0f;
unsigned int offsetTimer=0, OFF_SW=1, ErrCode=0;
char Err1 = 0, Err2 = 0;
float OverCurrent = 0.0f, OverSpeed = 0.0f;
float VdcFdbLpfFct=0.0f, IdcOffsetLpfFct=0.0f, rpmFilLpfFct=0.0f;
uint8_t ErrReset = 0;

float IadcDATA = 0;
```

```
float Motor_R = 0.0f;
float Motor_L = 0.0f;
unsigned int Wcc = 0;
float absIdcFdb=0.0;
float VdcCmd = 0.0f;

// Function Pre-define
void init_Variable(void);
void init_PIN(void);
void init_ADC(void);
void init_FlexPWM0_sub1(void);
void ADC_Read(void);
void Errdetect(void);
void H_BridgeRun(void);
void adc_offset_cal(void);
void PWMISR(void);

int main(void)
{
	initModesAndClock();
	disableWatchdog();
	enableIrq();
	initOutputClock();
	FMSTR_Init();
	init_INTC();
	init_Linflex0();
	init_PIN();
	init_ADC();
	init_FlexPWM0_sub1();
	init_Variable();

	INTC_InstallINTCInterruptHandler(PWMISR,183,6);

	// Loop forever
	for (;;)
	{
		FMSTR_Recorder();
		FMSTR_Poll();
	}
}
```

```
void init_Variable(void)
{
        FadcScale = 5.0f/1023.0f;
        FvdcHwScale = 10.0f/40.0f;
        FiHwScale = 1.0f / 20.0f;

        iScale = (float)(FadcScale/FiHwScale);
        VdcScale = (float)(FadcScale/FvdcHwScale);

        SIU.GPDO[40].B.PDO = 1; // RESET에 1 (active low임)
        SIU.GPDO[61].B.PDO = 1; // SR
        SIU.GPDO[58].B.PDO = 1; //Gate High 1을 이 핀에 내려보냄
        SIU.GPDO[59].B.PDO = 1; //Gate Low

        PWM_Peak = 1600; //duty 중간값
        PWM_Scale = PWM_Peak/12;

        FILFCT(10. , IdcOffsetLpfFct , 10000.);
        FILFCT(10. , IdcOffsetLpfFct , 100.);

        FILFCT(100., VdcFdbLpfFct, 10000.);
}

// MPC5604P Device Configuration
void init_PIN(void)
{
        SIU.PCR[62].R = 0x0600; // FlexPWM_0 B[1]->PHASE
        SIU.PCR[58].R = 0x0300;// GateIC PWMH
        SIU.PCR[59].R = 0x0300;// GateIC PWML
        SIU.PCR[61].R = 0x0300; // GateIC SR
        SIU.PCR[40].R = 0x0300; // GateIC Reset
        SIU.PCR[29].R = 0x0100; // GateIC Err2
        SIU.PCR[30].R = 0x0100; // GateOC Err1

        SIU.PCR[23].R = 0x2000; // ADC0 AN[0] VBAT
        SIU.PCR[34].R = 0x2000; // ADC0 AN[3] IDC
}

void init_ADC(void) // 0번 모듈 , 0번 채널 , 3번 채널
{
   ADC_0.MCR.B.ABORT = 1; //진행중인 변환 중단
```

```
    ADC_0.MCR.B.PWDN = 0; //POWER DOWN MODE OFF
    ADC_0.CTR[0].R = 0x00008208;
    ADC_0.NCMR[0].R = 0x00000009; //0번채널, 3번채널 ENABLE
    ADC_0.CDR[1].R = 0x00000000; //데이터 받아오는 레지스터 초기화
    ADC_0.CDR[3].R = 0x00000000;  //데이터 받아오는 레지스터 초기화
}

void init_FlexPWM0_sub1(void)
{
        FLEXPWM_0.OUTEN.B.PWMB_EN = 0b0010;
        FLEXPWM_0.MCTRL.B.LDOK = 0b0010;
        FLEXPWM_0.MCTRL.B.RUN = 0b0010;
        // complementary PWM pair 안함. 독립적 사용
        FLEXPWM_0.SUB[1].CTRL2.B.INDEP = 1;

        FLEXPWM_0.SUB[1].INIT.R = -3200;
        FLEXPWM_0.SUB[1].VAL[0].R = 0;
        FLEXPWM_0.SUB[1].VAL[1].R = 3200;
        //compare interrupt val0, val1, val2, val3 enable CMPIE REG
        FLEXPWM_0.SUB[1].INTEN.R = 0x0001;
        // PWM B Fault Mask : Turn on
        FLEXPWM_0.SUB[1].DISMAP.B.DISB = 0;
        FLEXPWM_0.MCTRL.B.LDOK = 0b0010; //1번모듈 ldok
        FLEXPWM_0.OUTEN.B.PWMB_EN = 0b0010;
}

void PWMISR(void)
{
        ADC_Read();
        H_BridgeRun();
        if(OFF_SW) adc_offset_cal();
        //PWM 1번 모듈에서 발생한 CMPI FLAG 전부 내림
        FLEXPWM_0.SUB[1].STS.R = 0x0001;
}

// 0번 모듈 사용, 3번채널은 전류 , 0번채널은 전압
void ADC_Read(void)
{
        ADC_0.MCR.B.NSTART = 1; // Module 0 Conversion Start
        while(ADC_0.MCR.B.NSTART) asm("nop");
```

```
        if(ADC_0.CDR[0].B.VALID == 1)
        {
                IadcDATA = (float)ADC_0.CDR[3].B.CDATA;
                IdcPre = -((float) ADC_0.CDR[3].B.CDATA - 512.0f) * (5.0f/1023.0f) *
10.0f;
                //VBAT값을 읽어온거 보정
                VdcPre = (float) ADC_0.CDR[0].B.CDATA * VdcScale;
        }
        IdcFdb = IdcPre - IdcOffset;
}

void H_BridgeRun(void)
{
        if(DSPRUN)
        {
                //PI 제어
                IdcErr = IdcRef - IdcFdb;
                VdcPterm = Kpc * IdcErr;
                VdcPiterm = VdcPterm + Kic * IdcErr +VdcIterm;
                VdcRef = bound(VdcPiterm, vRefUlim);
                VdcIterm += Kic * IdcErr;
                VdcIterm = bound(VdcIterm, vRefUlim);
                //1~3ms 안에 전류가 도달해야함
                PWM_B = (int16_t)(VdcRef*PWM_Scale) + PWM_Peak;

                FLEXPWM_0.SUB[1].VAL[4].R = (unsigned short)-PWM_B;
                FLEXPWM_0.SUB[1].VAL[5].R = (unsigned short) PWM_B;
                FLEXPWM_0.MCTRL.B.LDOK = 0b0010; //1번모듈 ldok
                FLEXPWM_0.OUTEN.B.PWMA_EN = 0b0010;

                SIU.GPDO[40].B.PDO = 1;
                SIU.GPDO[58].B.PDO = 1; //Gate High
                SIU.GPDO[59].B.PDO = 1; //Gate Low
                SIU.GPDO[61].B.PDO = 1;
        }
        else
        {
                SIU.GPDO[40].B.PDO = 0;
                SIU.GPDO[58].B.PDO = 0;
                SIU.GPDO[59].B.PDO = 0;
                SIU.GPDO[61].B.PDO = 0;
```

```
                FLEXPWM_0.OUTEN.B.PWMA_EN = 0x0;

                IdcRef = 0.0;
                VdcRef = 0.0;
                VdcPterm = 0.;
                VdcIterm = 0.0;
                VdcPiterm = 0.0;
        }
}

void adc_offset_cal(void)
{
        offsetTimer++;
        if(offsetTimer > 10000)
        {
                OFF_SW = 0;
                offsetTimer = 10001;
        }
        else
        {
                DSPRUN = 0;
                OFF_SW = 1;
                ErrCode=0;
                LPF(IdcOffset, IdcPre, IdcOffsetLpfFct);
        }
}
```

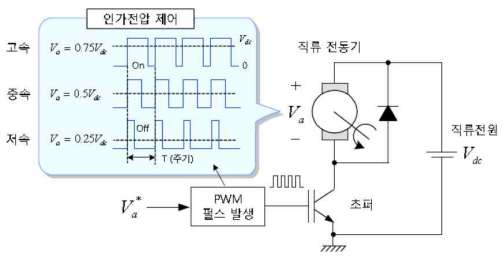


그림 2.48 직류 전동기 구동회로

=> 직류 전동기는 코일로 감겨져 있어 전류를 이용함.

=> 인덕터 충전전류로 인한 기기의 손상을 방지하기 위해 부하와 병렬로 연결된 활류 다이오드가 존재함.

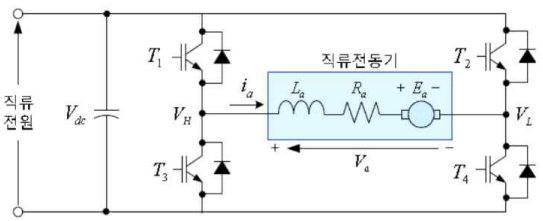


그림 2.49 H-브리지 회로

=> 모터의 속도가 무한대로 흐르지 않게 하기 위해 발전기에 역기전력이 흘러 속도가 제한됨.

=> 역방향, 정방향의 모터제어가 모두 가능함.

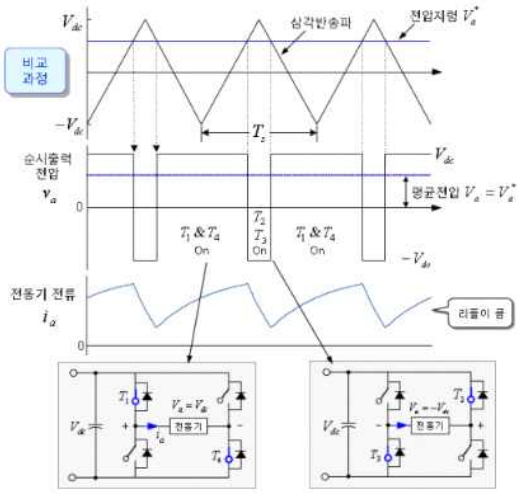


그림 2.50 바이폴라 변조 방식

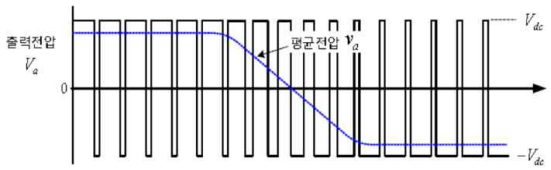


그림 2.51 바이폴라 변조 방식에서의 출력 전압

=> 바이폴라 변조 방식에서 +전압 또는 -전압

=> 부드러움

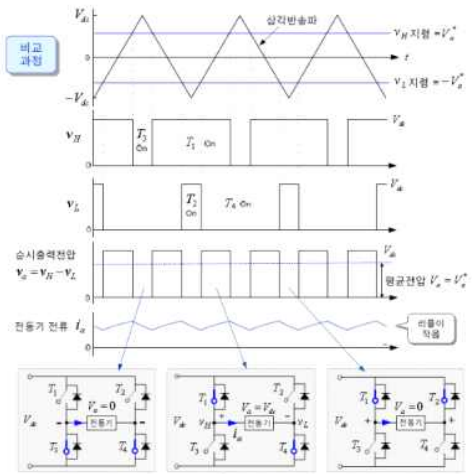


그림 2.51 유니폴라 변조 방식

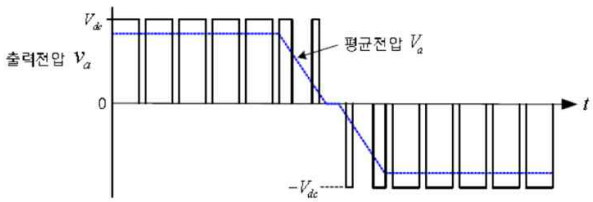
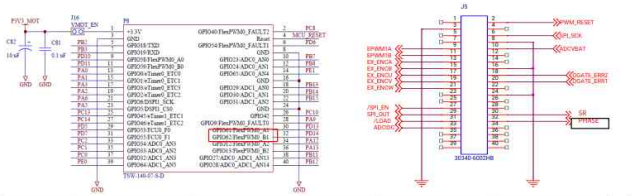


그림 2.53 유니폴라 변조 방식에서의 출력전압

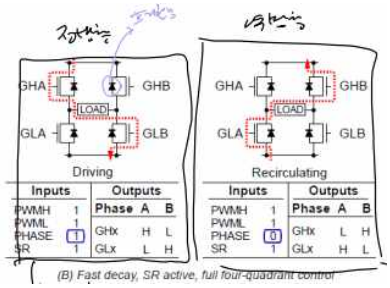
=> 유니폴라 변조 방식 +전압/0 또는 -전압/ 0

=> 부드럽지 않음 (정/역방향이 빈번하게 진행)

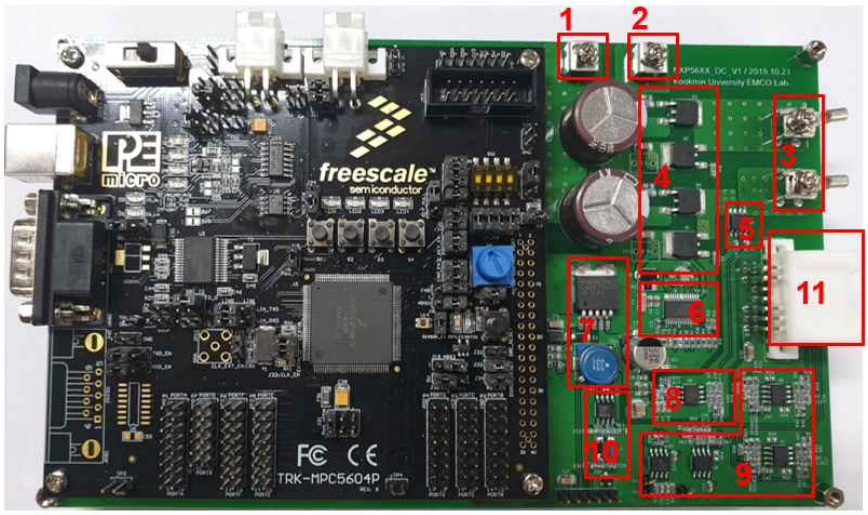


Port No.	Name	Function	Port No.	Name	Function
PD10	EPWM1A	GPIO output	PC8	PWM_RESET	GPIO output
PD11	EPWM1B	GPIO output	PD6	SPI_CLK	DSPI0 SCK
PA0	EX_ENCA	eTimer0 ETC0	PB7	ADCVBAT	ADC0 AN[0]
PA1	EX_ENCB	eTimer0 ETC1	PB13	OGATE_ERR2	GPIO input
PC14	/SPI_EN	DSPI0 CS0	PB14	OGATE_ERR1	GPIO input
PD5	SPI_OUT	DSPI0 SOUT	PD13	PHASE	FlexPWM0_B1
PD7	/LOAD	GPIO output	PD14	SR	GPIO output
PC2	ADCIDC	ADC0 AN[3]			

=> Port의 Pin 설정



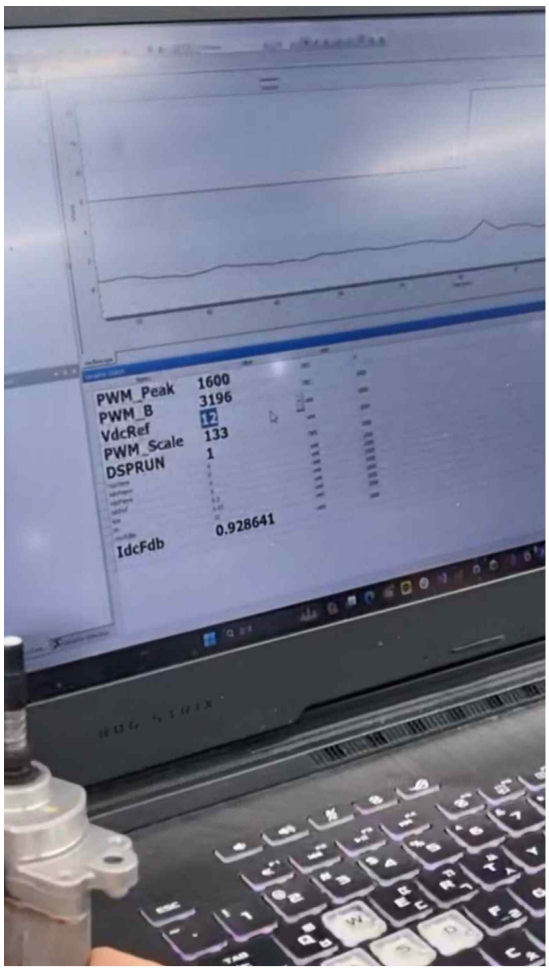
=> 바이폴라 모드로 정방향/역방향 세팅



1	12V
2	GND
3	모터 출력
4	MOSFET
5	전류센서
6	Gate Driver
7	Regulator
8	Op-Amp
9	비교기
10	CAN
11	센서 커넥터

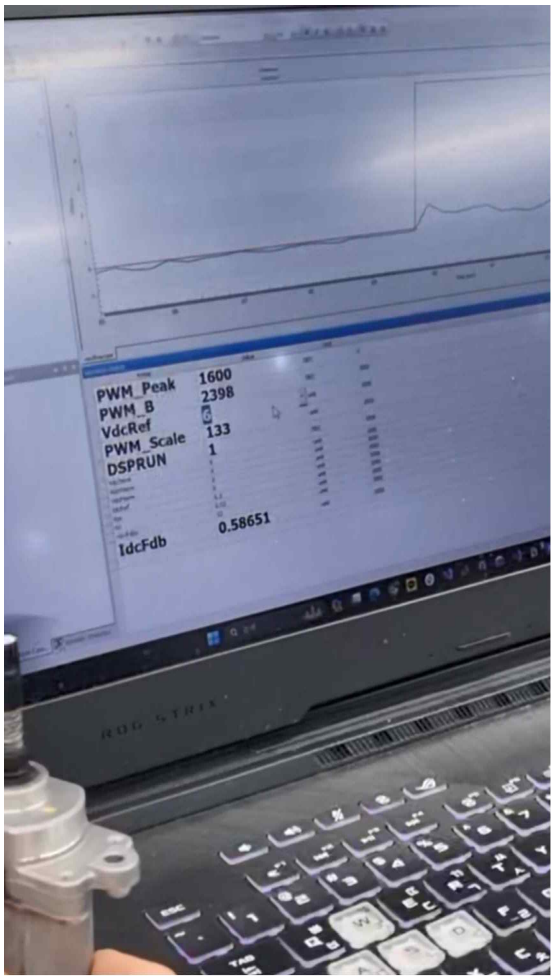
=> 모터 구동 보드와 제어 보드 연결 및 각 부품의 역할

=> 최대 전압 12V, 저항 12옴



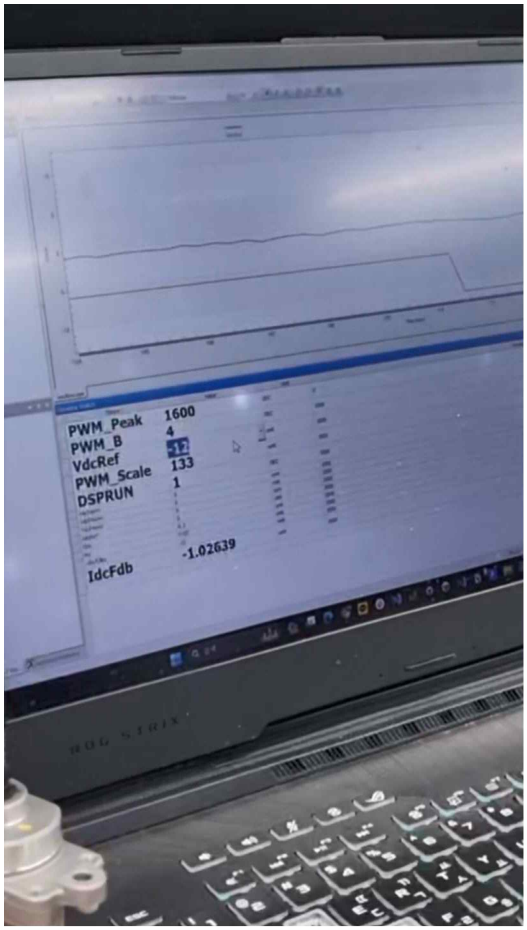
=> 12V 전압지령 (1A 근사값이 센싱)

=> 정방향 모터구동



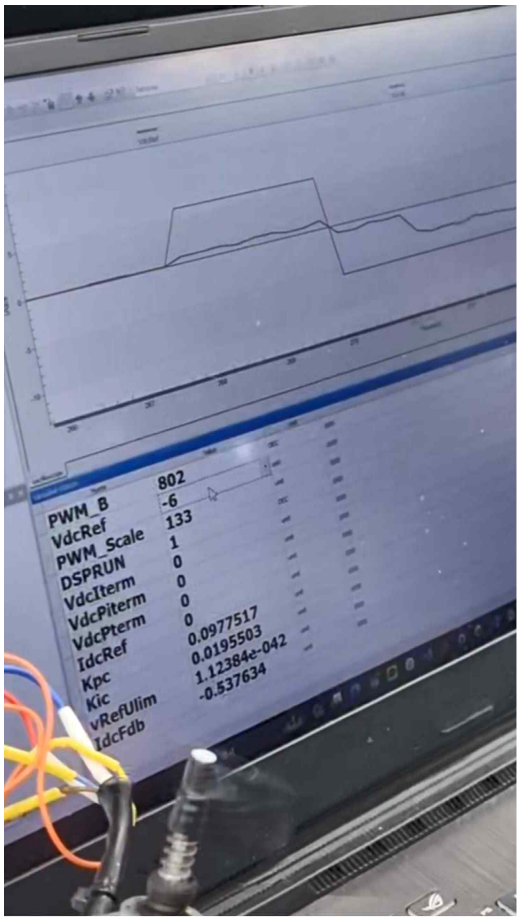
=> 6V 전압지령 (0.5A 근사값이 센싱)

=> 정방향 모터구동



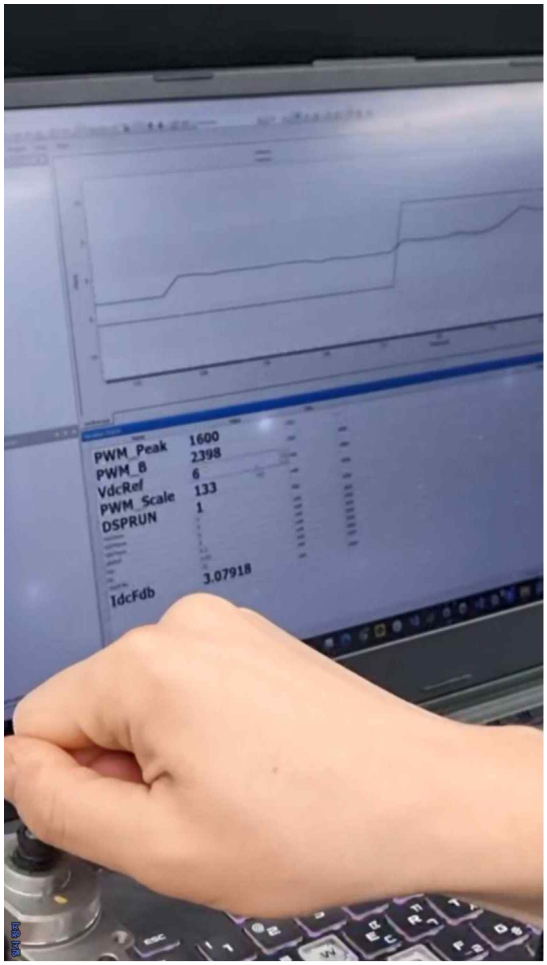
=> -12V 전압지령 (-1A 근사값이 센싱)

=> 역방향 모터구동

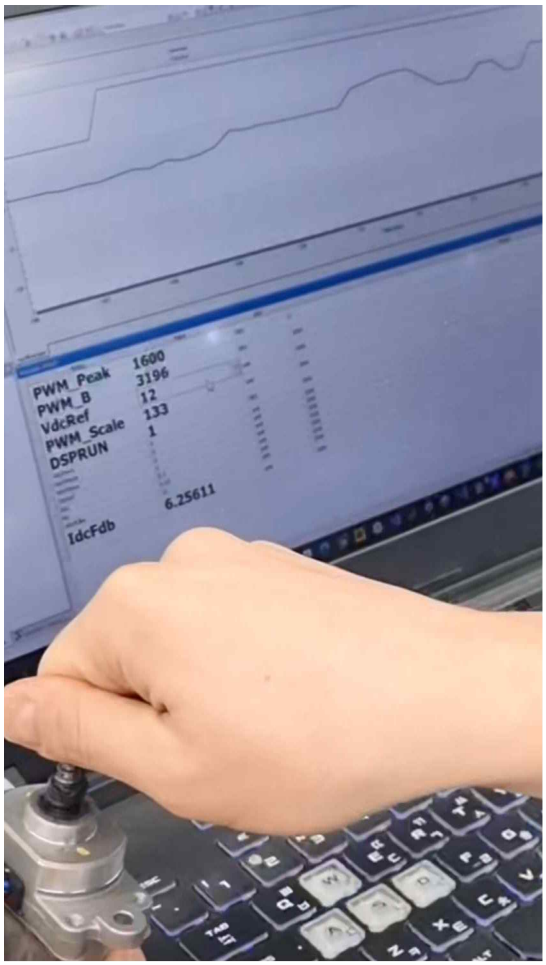


=> -6V 전압지령 (-0.5A 근사값이 센싱)

=> 역방향 모터구동



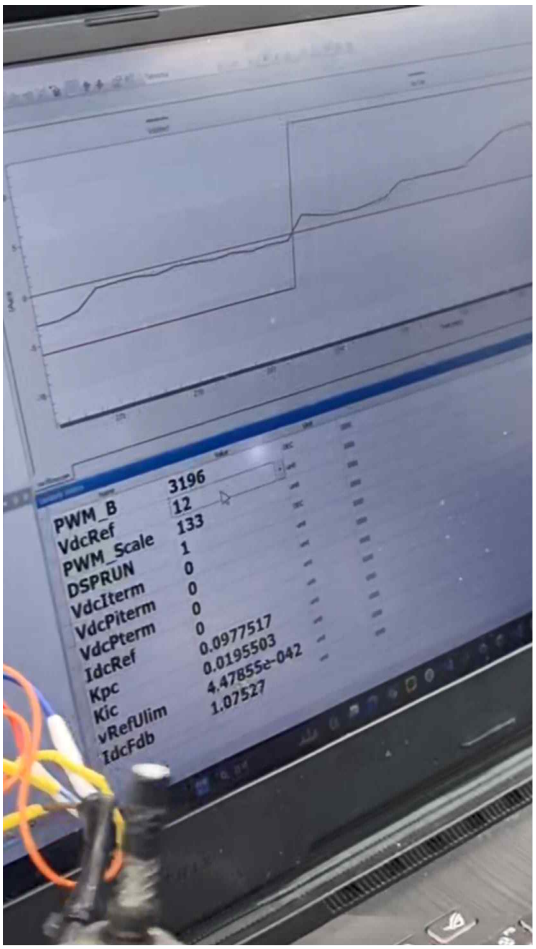
=> 6v 전압지령으로 모터 구동 시 강제로 모터 구동을 막음으로 원하는 출력이 나오지 않아 전류가 급격히 증가함. (3A 근사값)



=> 12v 전압지령으로 모터 구동 시 강제로 모터 구동을 막음으로 원하는 출력이 나오지 않아 전류가 급격히 증가함. (6A 근사값)

=> 파워 서플라이의 값도 대폭 증가하게 됨.

=> 12v 전압지령으로 모터 구동 중 -6v의 전압지령으로 전류 그래프가 급격히 하강. 그만큼의 에너지를 이용하는게 회생제동



=> -6v 전압지령으로 모터 구동 중 12v의 전압지령으로 전류 그래프가 급격히 상승. 그만큼의 에너지를 이용하는게 회생제동