

모터뱅크 | GSA56-2425

사용설명서

1. GSA56-2425사용안내

1.1 제품특징

- *더블 엔코더로 축의 싱글턴절대위치 출력가능. 정전시 출력축의 절대위치를 방지가능
- *모터 드라이버 일체형
- *CAN BUS를 통해 모터 속도/위치/토크/온도 등 피드백가능
- *과열보호기능탑재
- *위치모드에서 사다리꼴가감속 지원

1.2 사용전압: DC 24V

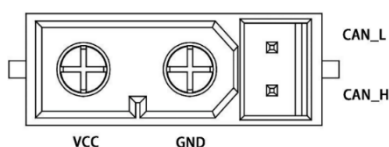
1.3 사용전류: 드라이버는 최대 41A까지 가능. 프로그램을 통해 최대 전류비율설정이 가능합니다. 기본값은 0.8입니다(즉 최대 80%까지 사용하다는 뜻입니다). 98%이상 사용하시는 것을 삼가해주시기 바랍니다.

1.4 최고속도

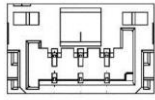
최고속도는 여러가지 요소의 제한을 받고 있습니다. 공급전압(V_{BUS}), KV값, 감속비GR 등 요소의 원인으로 하단의 서식을 통해 상한값을 얻을수 있습니다. (N_{pp} 는 극쌍수/ rotor flux)

$$V_{MAX}(rad/s) = 0.57735 * \frac{V_{BUS}}{N_{pp} \times GR \times \psi_f} (rad/s)$$

1.5 인터페이스 설명



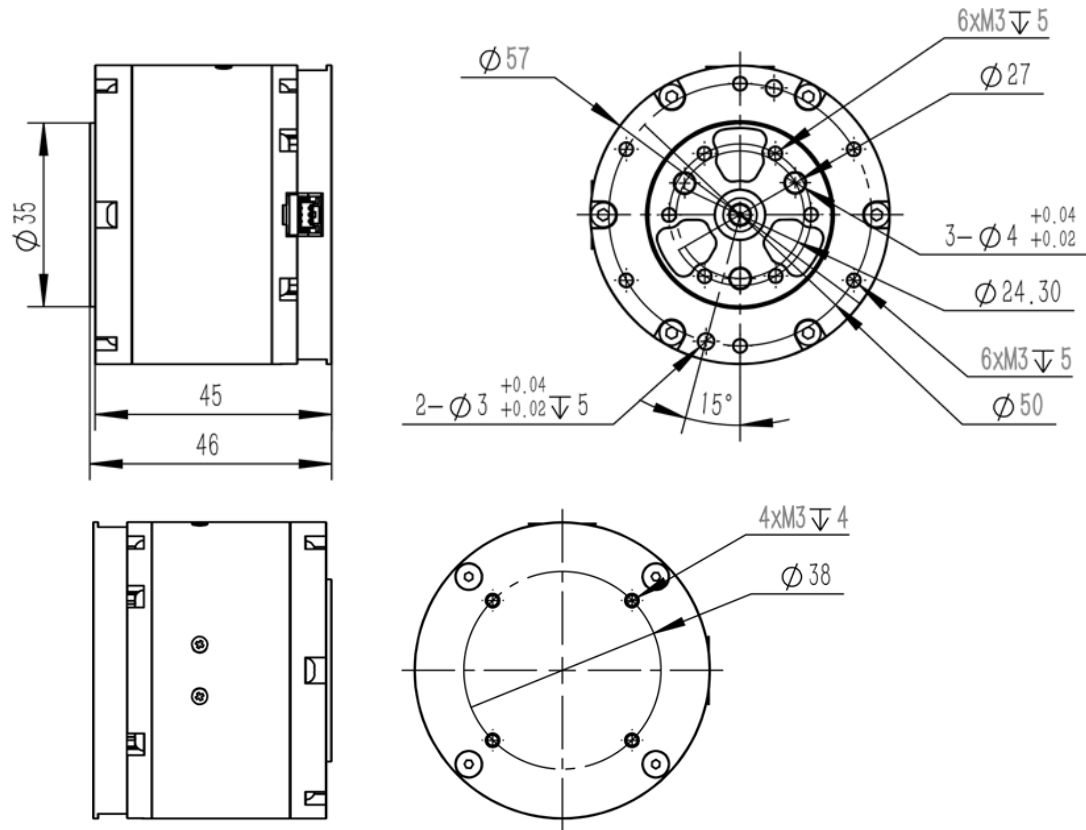
- 정격전압 24V . CAN통신포트를 통해 외부 컨트롤에 연결하면 CAN제어명령을 받을수 있어 모터 피드백을 받을수 있습니다.
- 모터에는 2개의 전원코드가 있습니다. 2개중 임의 한 개로 단독 연결시 사용가능하며 병렬 사용 가능합니다.



GND RX TX

- GH1.25연결선 -3PIN,USB-CAN전환탭을 PC에 연결 후 프로그램을 통해 스펙설정을 진행합니다

1.6 모터사이즈 및 설치참고



1.7 상태등 알림

정상

초록불 켜짐	정상작동상태
빨간불 켜짐	정지상태

에러상황

빨간불 반짝임	8: 과전압 9: 저전압 A: 과전류 B: MOS 과열 C: 모터권선과열 D: 통신실패 E: 과부하 피드백/ 프로그램을 통해 에러확인가능
---------	---

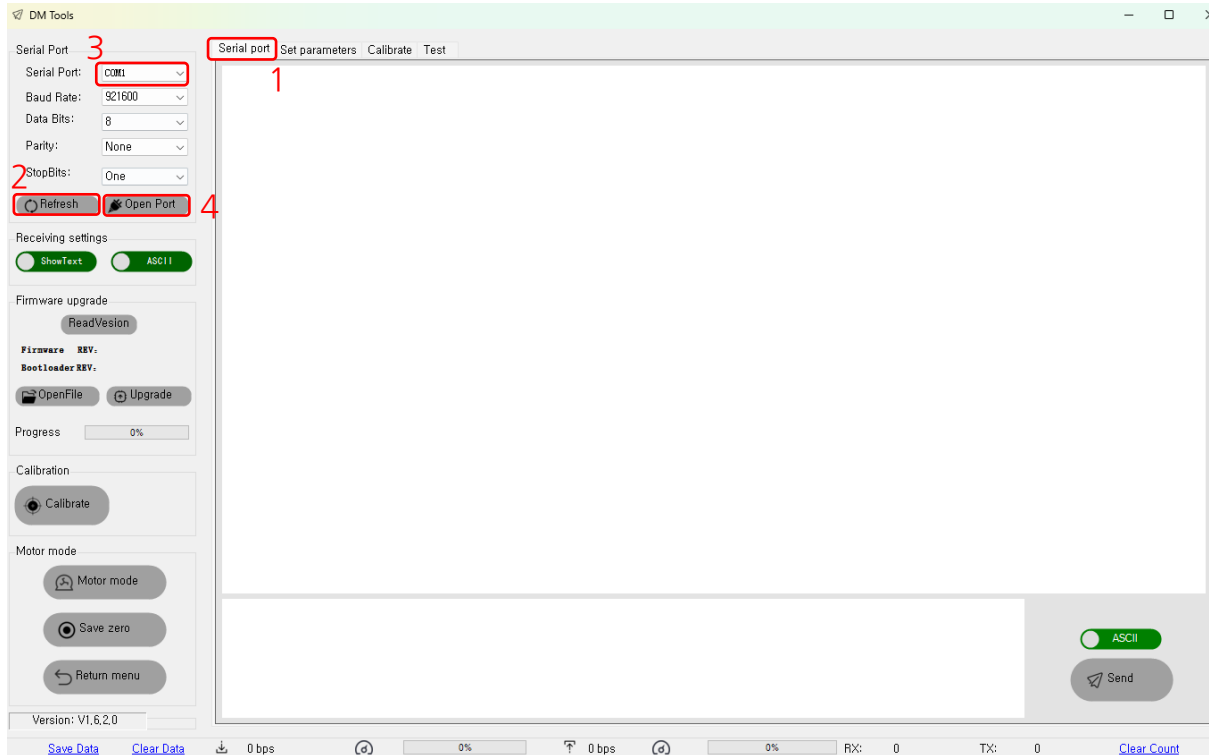
1.8 스펙

정격전압	24V
정격전류	2.5A
최대전류	7.5A
정격토크	3NM
최대토크	7NM
정격속도	120RPM
무부하최대속도	200RPM
감속비	10:1
극수	14
상 인덕턴스	340uH
상저항	650mΩ
외경	56mm
높이	46mm
무게	300g
엔코더비트	14 비트
엔코더개수	2
엔코더유형	자석엔코더(싱글)
제어타입	CAN@1Mbps
스펙조절 인터페이스	UART@921600bps
제어모드	MIT모드
	속도모드
	위치모드
보호	드라이버 과열보호. 보호온도 120 ℃, 과열시 모터는 Enter모드에서 로그아웃
	모터 과열보호. 보호온도 100 ℃, 과열시 모터는 Enter모드에서 로그아웃
	모터 과전압보호. 최대 32V. 과전압시 Enter모드에서 로그아웃
	통신분실보호. 설정한 주기내에 CAN명령을 수신못한 경우 Enter모드에서 로그아웃.
	모터 과전류보호. 최대 9.8A. 과전류시 Enter모드에서 로그아웃
	모터 저전압보호. 저전압시 Enter모드에서 로그아웃. 15V보다 높아야함.

2. 모터조정방법

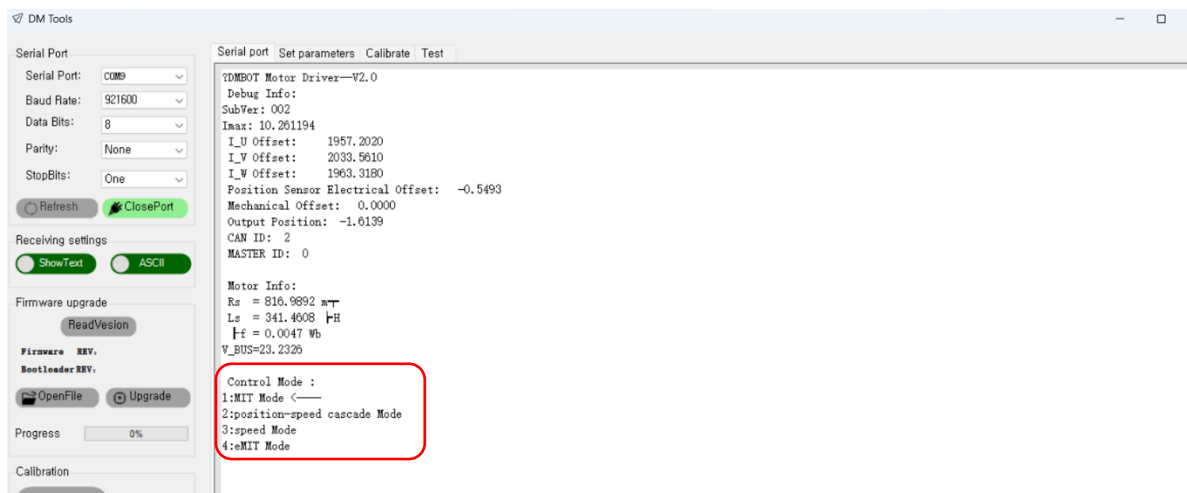
2.1 프로그램 연결

모터와 pc 및 공급전원을 연결합니다.



모터에 전원을 인가하면 하단과 같은 정보를 확인할수 있습니다

Control Mode : 현재 구동모드. 모드에 따라 사용하는 명령형식도 다릅니다(CAN통신부분 참고)



2.2보정(Calibrate)

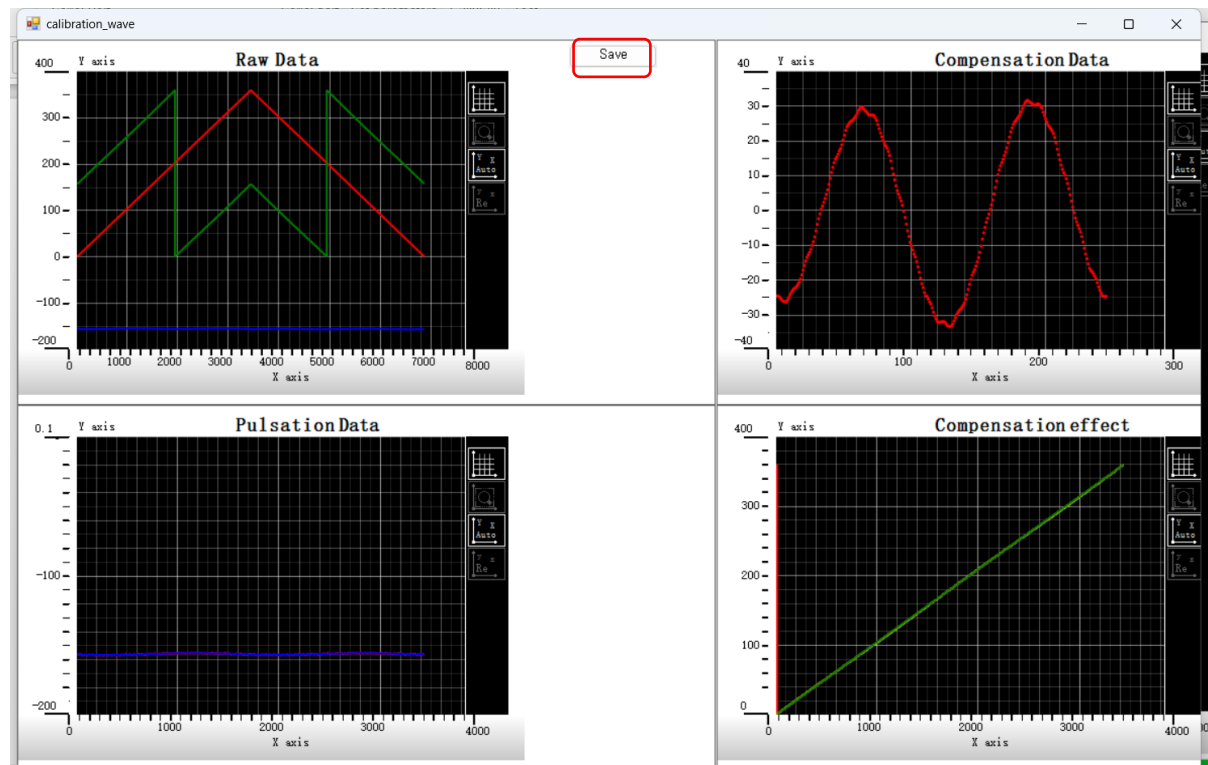
해당 작업은 센서의 설치오차에 대해 보정 진행합니다. 보정과정에서 모터는 정역으로 한바퀴 작동합니다. 모터가 자율적으로 작동할수 있도록 해줘야 합니다. **무부하 상태로 보정작업을 진행하는 것을 권장**드립니다. (부하상태 시 보정실패 가능성도 있습니다)

① 센서보정

보정(Calibrate)을 클릭하면 모터가 작동을 시작하며 완료되면 극쌍수 값이 뜹니다.모터는 정역회전을 하면서 센서데이터보정을 진행합니다.



보정완료후 그래프 팝업이 뜹니다.



② 보정데이터 점검 및 저장

Compensation Data값을 우선적으로 확인해주셔야 합니다. ± 50 을 초과하면 안됩니다.

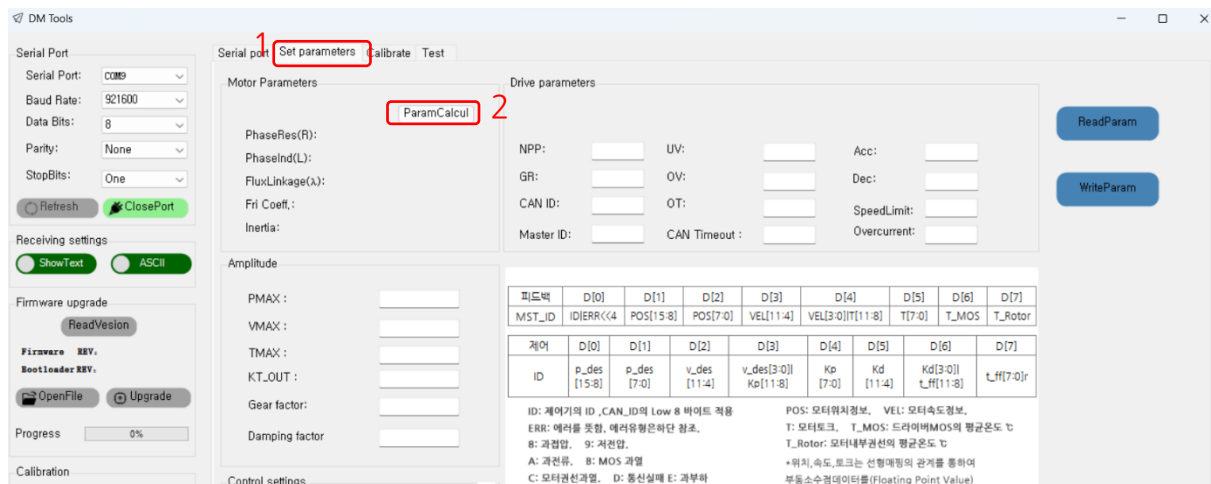
± 50 을 초과한 경우 하단 3가지 확인해주셔야 합니다

1. 극쌍수 식별 실패
2. 모터 저항력이 높음
3. 잘못된 센서설치

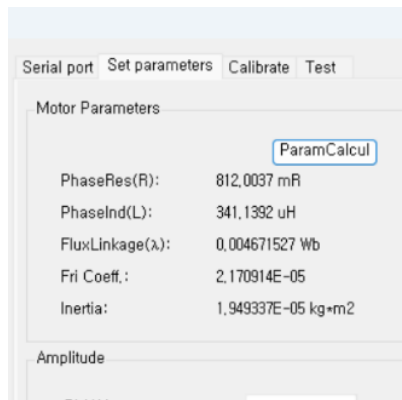
문제없을 경우 저장Save를 누르시면 보정데이터가 모터조정설정에 저장됩니다.

2.3 사양설정

보정이 완료된 후 모터의 사양을 확인합니다. 사양설정을 누른 후 사양측정버튼을 클릭하시면 모터는 자동식별을 시작합니다. ***무부하상태를 권장드리며 모터를 고정시켜줘야 합니다.**



식별 완료되면 모터사양이 자동으로 뜹니다,

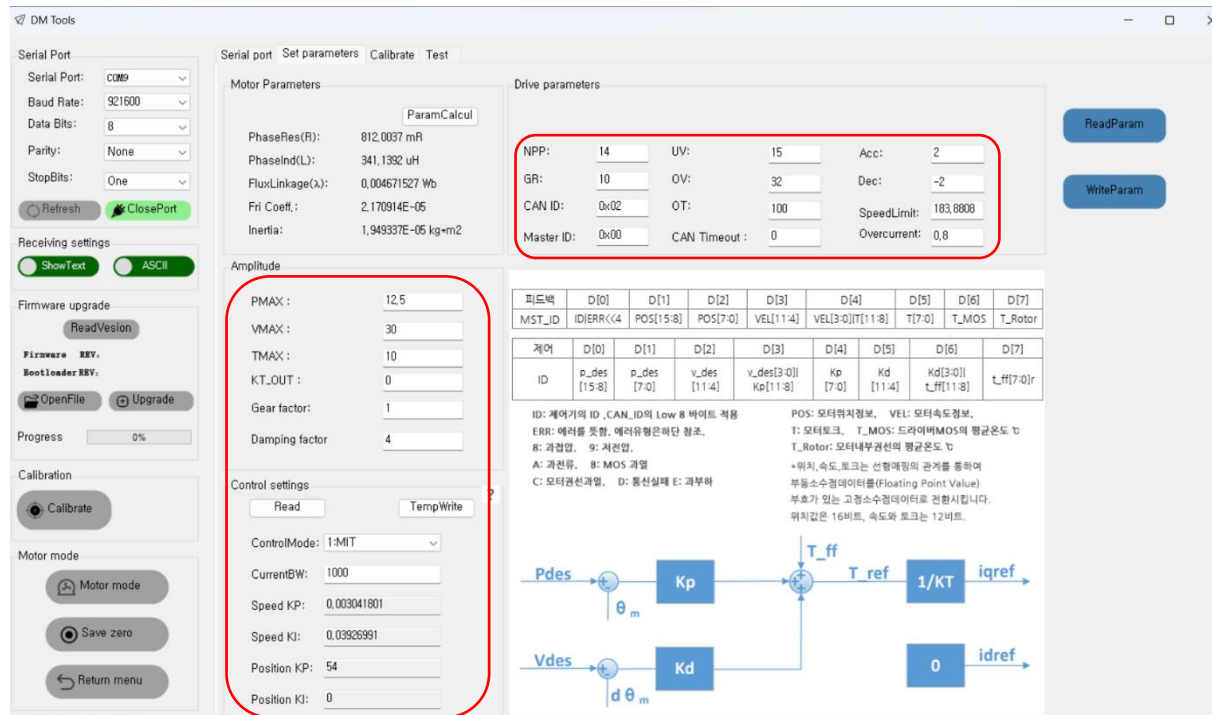
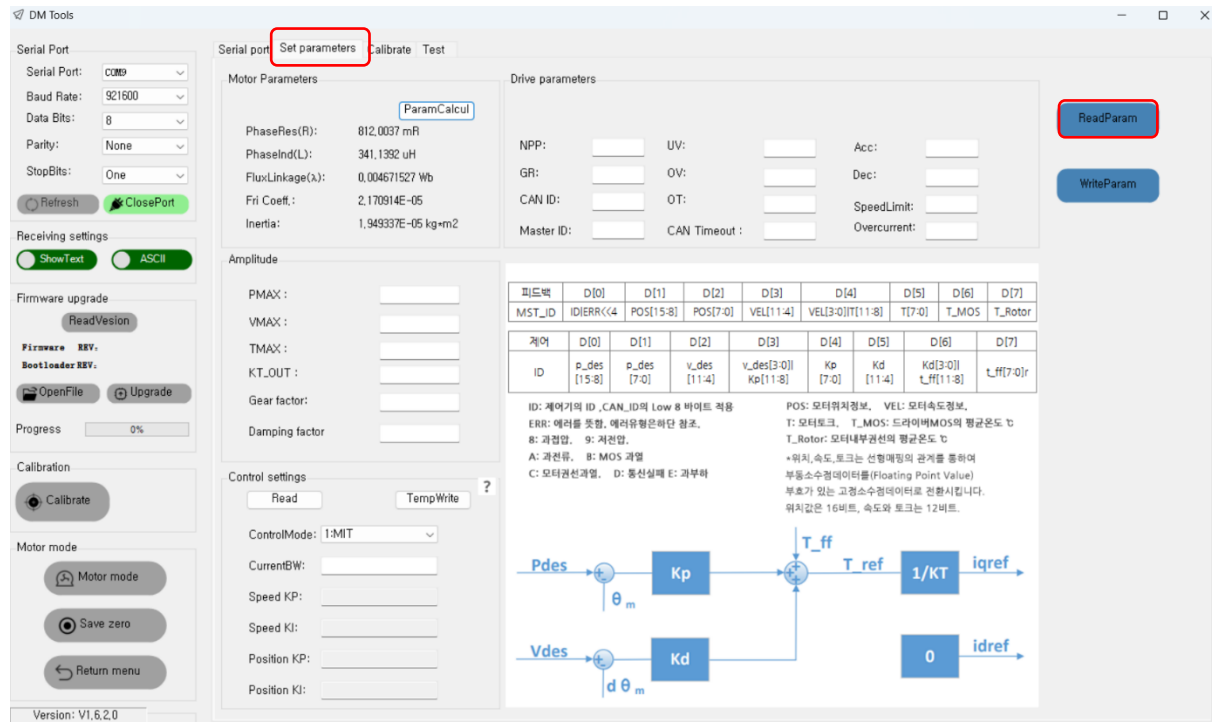


사양측정부분에서 마이너스값이 나올 경우 다시 측정하셔야 합니다

2.4 사양읽기/쓰기

2.4.1 사양읽기

사양설정에서 사양읽기를 누르면 모터에 저장된 사양이 뜹니다.



읽은 사양은 하단내용이 포함됩니다.

***드라이버사양: 드라이버의 사양설정부분**

극쌍수: 모터의 극쌍수. 보정단계에서 얻을 수 있습니다

저전압: 전원전압이 설정값 보다 낮을 경우 프로그램은 더이상 작동하지 않습니다. 공급전압>15v

과전압: 프로그램의 전압 상한값 설정. 모터에 전원을 인가 시 공급전압을 측정합니다. 공급전압이 설정값 보다 높을 경우 프로그램은 작동하지 않습니다. (전원인가시에만 한번 작동)

가/감속: MIT모드외에 사용. 프로그램의 가/감속을 제한하는데 사용

감속비: 모터감속비. 모터출력속도/위치와 연관이 됩니다. 모터토크피드백에도 간접적으로 영향됩니다.

과열: 사용환경에 따라 설정. 100℃를 초과할 경우 프로그램이 중단되거나 모터가 고장날 수 있습니다

CAN_ID: CAN의 명령을 수신하는 프레임ID, 16진수.

Master ID: 프로그램이 피드백에 사용되는 프레임ID, 16진수.

CAN Timeout: CAN통신시간 설정. 정수32비트, 일정수량의 counting cycle후 CAN명령을 측정하지 못하여 모터에 대한 보호기능을 실시한다는 뜻입니다. 1개의 counting cycle당 50us.

속도제한: 속도모드에서만 사용가능. 모터 최대속도를 제한하는데 사용, rad/s

과전류: 모터의 최대 상전류를 제한하는데 사용, 백분율

***모터사양: 모터사양의 사양설정부분**

해당 사양은 드라이버에서 자동식별되는 사양입니다.

***진폭제어: 구동명령 사양범위 설정**

PMAX : MIT모드에서 모터가 피드백을 수신하는 매핑값. 매핑규칙은 CAN부부 참조.

VMAX : PMAX와 동일.

TMAX : PMAX와 동일.

KT_OUT : 모터토크계수, 모터사양이 정확한 경우 0으로 설정.

기어계수 : 기어토크계수

***제어설정**

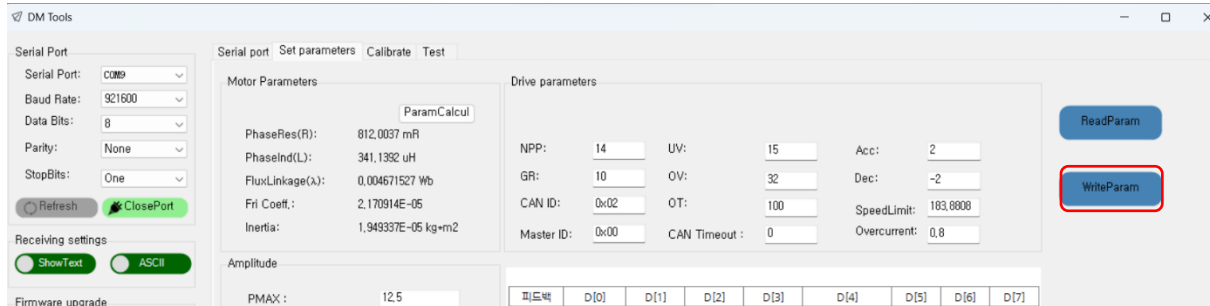
제어모드: MIT모드, 속도위치모드(사다리꼴가감속), 속도모드

전류대역폭: 전류루프의 이득계수, 기본값 1000.

속도KP/KI, 위치KP/KI: 속도루프, 위치루프의 사양.

2.4.2 사양쓰기

드라이버사양, 진폭제어, 제어설정 등 사양을 점검후 수정할 사양에 대해 변경을 진행합니다. 완료후 사양쓰기를 클릭하면 수정완료된 사양을 프로그램에 저장합니다.

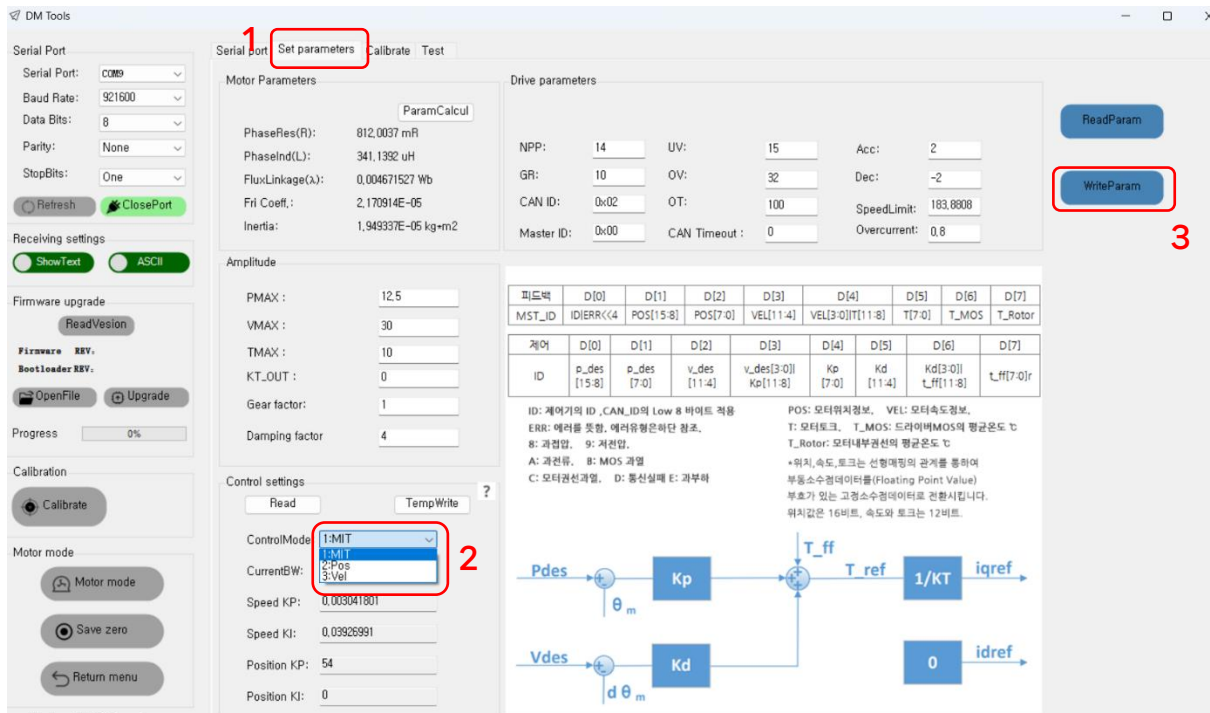


2.5 모드설정

해당 기능은 CAN인터페이스에 연결한 경우에만 사용가능합니다. CAN커넥터를 프로그램에 연결해주셔야 합니다.

2.5.1 제어모드의 선택과 확인

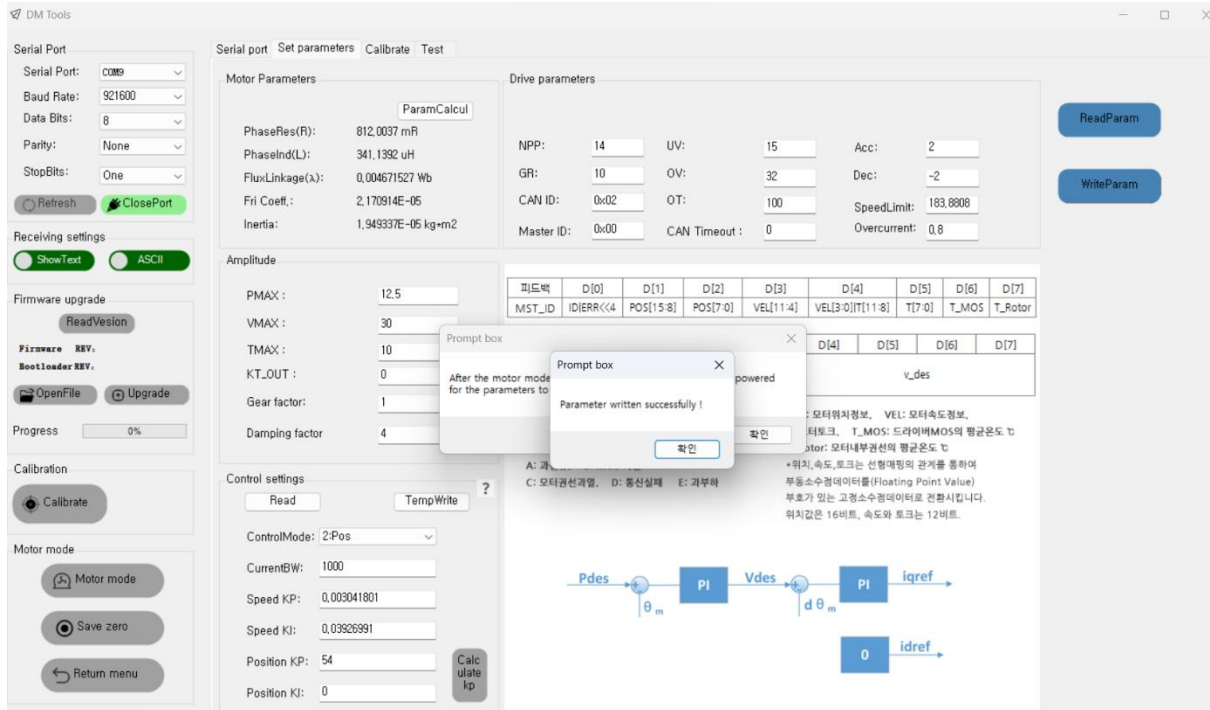
사양설정을 클릭후, 제어설정에서 제어모드를 설정클릭하면 MIT모드,속도위치모드(pos),속도모(vel) 드 세가지 선택이 가능합니다. 선택후 사양쓰기를 클릭하시면 제어모드가 설정됩니다.



제어모드 수정 시 팝업창이 2개 뜹니다.

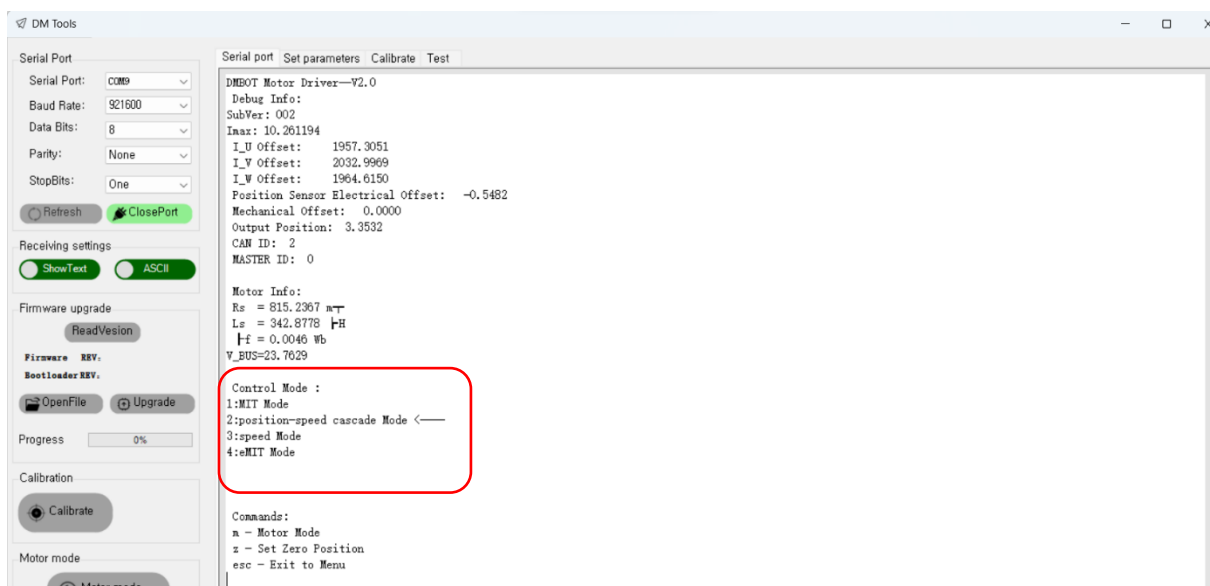
"모드변경 클릭후 전원을 다시 인가하셔야 사양이 유효합니다."

"사양쓰기성공"



제어모드를 변경할때마다 전원을 다시 인가해주셔야 합니다. 시리얼포트에서 표기된 데이터 혹은 사양 설정페이지에서 다시 읽은 사양을 통해 드라이버의 제어모드가 판단이 됩니다

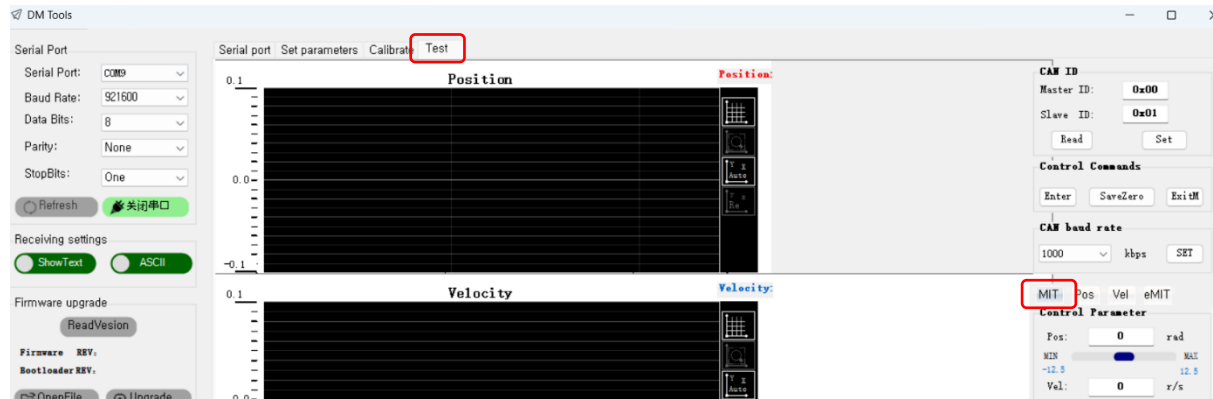
*시리얼포트에서 화살표가 표시된 모드가 현재 진행하는 제어모드입니다



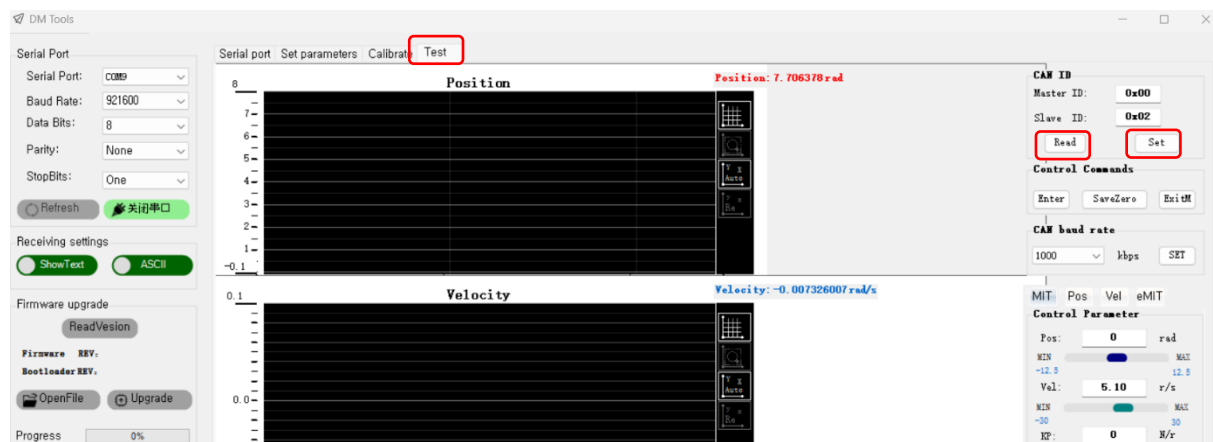
사양설정페이지에서 사양을 다시 읽은 후의 정보에 따라 드라이버 제어모드를 판단합니다.

2.5.2 MIT모드

1) 2.5.1의 제어모드선택 부분 내용을 참조하여 test에서 MIT모드로 선택합니다. 제어모드를 확인후 프로그램에서 대응되는 MIT버튼을 클릭합니다.



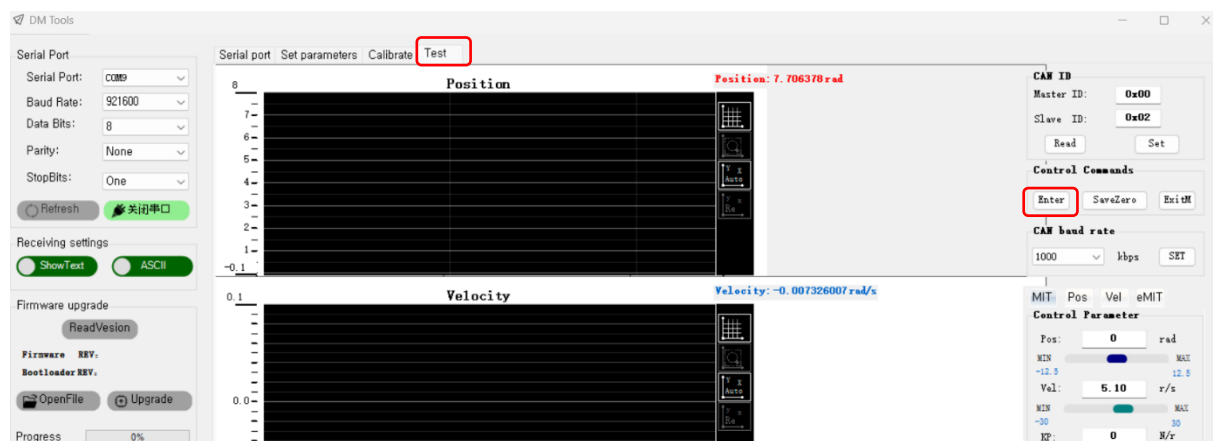
2) CAN ID 확인(시리얼포트, 사양설정페이지 등을 통해 확인가능, 테스트페이지에서 읽기->설정 클릭해도 됩니다.)



3) MIT모드에는 3가지 제어모드가 있습니다: 속도, 위치, 토크

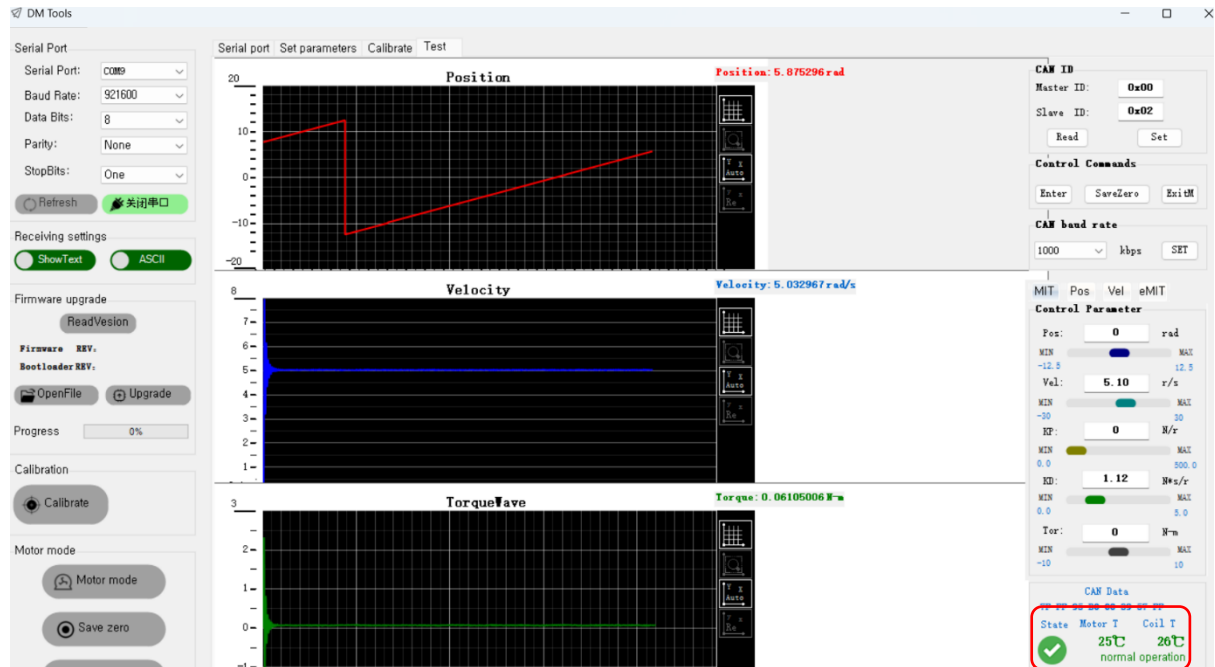
속도모드

Step1: 모터모드에서 "Enter"버튼 클릭하면 드라이버는 초록색으로 모터모드가 진행됩니다.



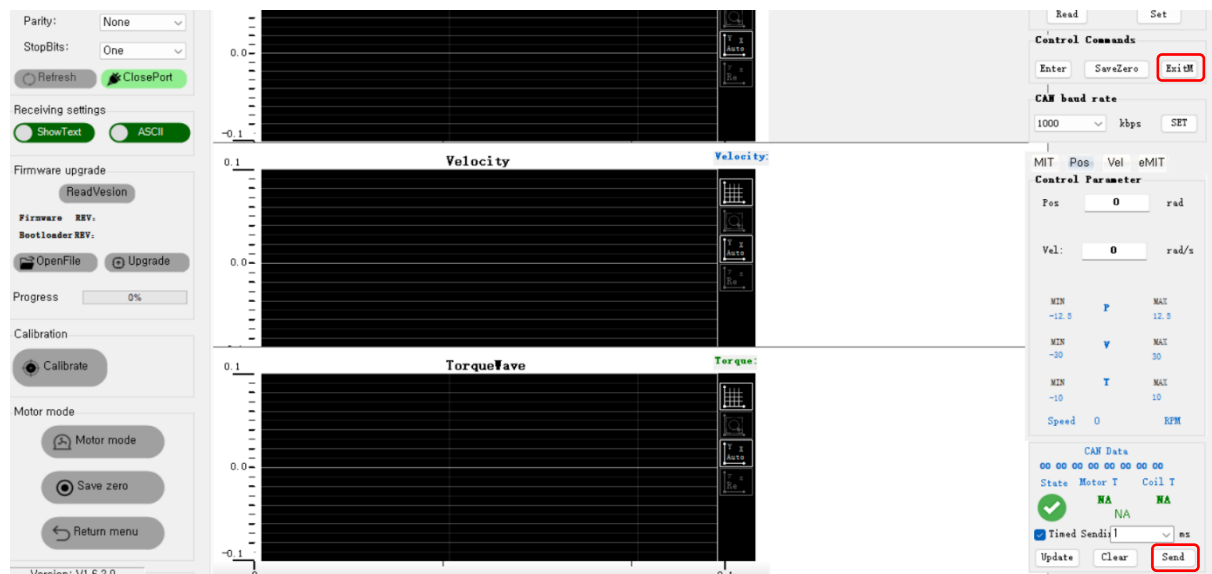
Step2: 속도값을 입력하여 모터가 속도값대로 구동할수 있도록 합니다

예: 목표속도: 5r/s, KD값은 1N*s/r 나머지는 모두 0으로 입력,센딩타임을 체크후 업데이트 및 보내기 버튼을 클릭하면 사양그래프 변화를 확인할수 있습니다. **모터를 고정시켜줘야 합니다.**



원하시는 사양에 따라 제어사양을 변경하여 그래프 변화를 확인하고 싶으시면 직접 사양을 변경후 센딩타임 및 업로드버튼을 클릭하시면 됩니다.

프로그램에서는 모터와 드라이버의 온도 및 모터 운전상태도 실시간으로 표시됩니다. 피드백프레임을 통해확인도 가능합니다. 피드백 프레임 통해서도 확인이 가능합니다. 피드백프레임 양식 및 고장원인은 4.1 피드백프레임 부분 참고바랍니다



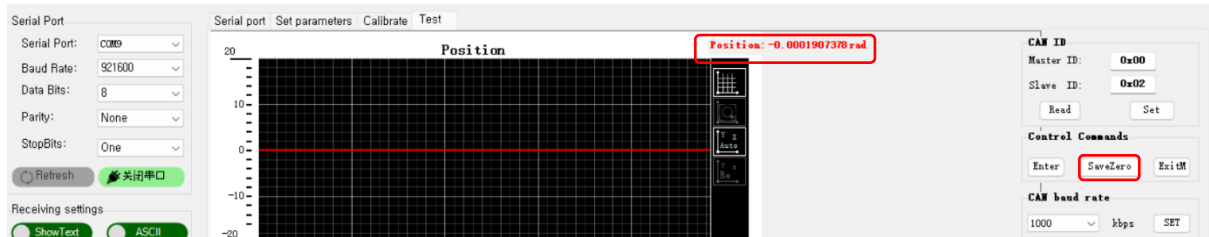
Step3: 프로그램 조정상태를 중지하고 싶으시면 정지 및 Enter버튼을 클릭합니다. 드라이버가 빨간색일 때 모터모드에서 중지됩니다.

위치제어

Step1: 모터모드에서 "Enter"버튼 클릭하면 드라이버는 초록색으로 모터모드가 진행됩니다.

Step2: 목표값을 설정하여 모터가 지정위치로 도달하게 합니다

*모터의 시작위치를 기억하셔야 합니다. 제어사양에서 위치 사양에 목표값을 설정시 시작위치와 갭차이가 너무 크면 모터에 충격을 부여하게 됩니다. 제어사양의 원점저장을 사용하여 모터 현재위치를 원점으로 설정을 하면 위치사양설정에 편리합니다.



예시: 목표위치를 3.14rad, KP를 2N/r, KD 를 1N*s/r ,나머지 0 센딩타임을 체크후 업데이트 및 보내기 버튼을 클릭하면 사양그래프 변화를 확인할수 있습니다. **모터를 고정시켜줘야 합니다.**

원하시는 사양에 따라 제어사양을 변경하여 그래프 변화를 확인하고 싶으시면 직접 사양을 변경후 센딩타임 및 업로드버튼을 클릭하시면 되십니다.

프로그램에서는 모터와 드라이버의 온도 및 모터 운전상태도 실시간으로 표시됩니다. 피드백프레임을 통해확인도 가능합니다. 피드백 프레임 통해서도 확인이가능합니다. 피드백프레임 양식 및 고장원인은 4.1 피드백프레임 부분 참고바랍니다.

Step3: 프로그램 조정상태에서 중지하고 싶으시면 정지 및 Enter버튼을 클릭합니다. 드라이버가 빨간색 일 때 모터모드에서 중지됩니다

MIT토크제어

Step1: 모터모드에서 "Enter"버튼 클릭하면 드라이버는 초록색으로 모터모드가 진행됩니다.

Step1: 목표토크값을 입력후 모터를 조정합니다.

*무부하에서는 최소한의 토크로 설정하더라도 모터는 최대속도까지 가속도를 진행합니다

예: 목표사양; 토크를 1N-m로 설정후 나머지는 모두 0으로 설정. 센딩타임,업데이트 및 보내기를 클릭하면 그래프확인이 가능합니다. **모터를 고정시켜줘야 합니다.**

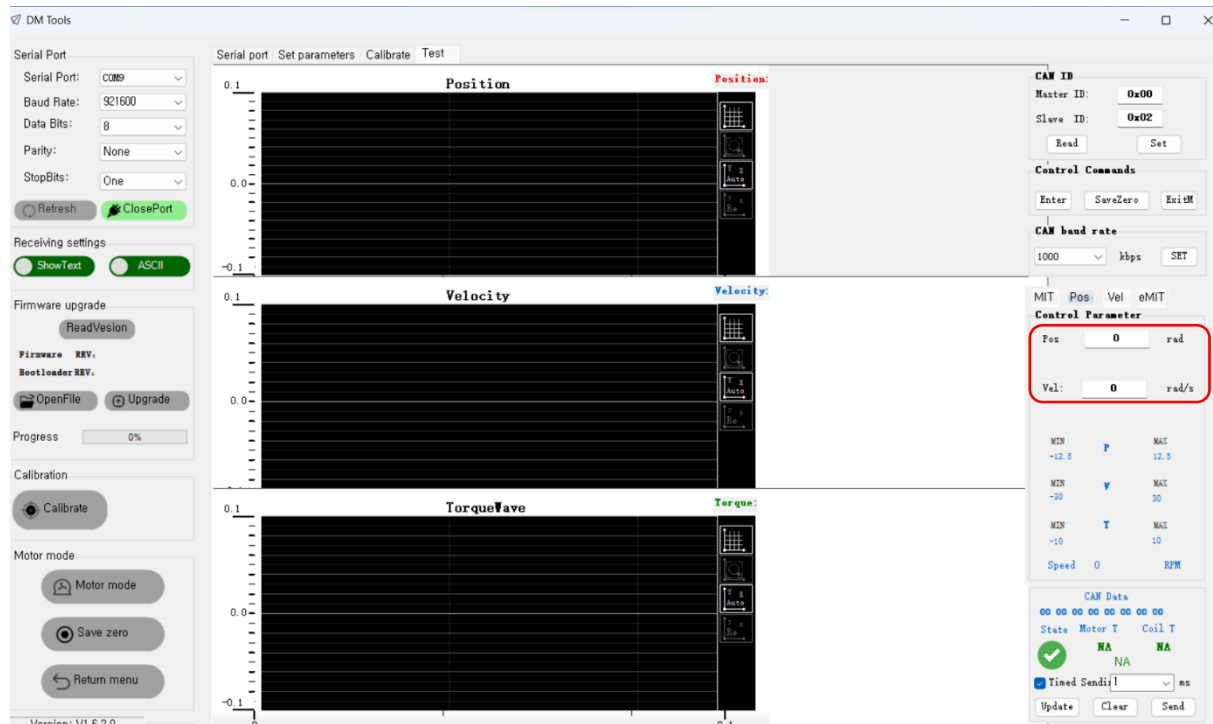
원하시는 사양에 따라 제어사양을 변경하여 그래프 변화를 확인하고 싶으시면 직접 사양을 변경후 센딩타임 및 업로드버튼을 클릭하시면 되십니다.

프로그램에서는 모터와 드라이버의 온도 및 모터 운전상태도 실시간으로 표시됩니다. 피드백프레임을 통해확인도 가능합니다. 피드백 프레임 통해서도 확인이가능합니다. 피드백프레임 양식 및 고장원인은 4.1 피드백프레임 부분 참고바랍니다.

Step3: 프로그램 조정상태에서 중지하고 싶으시면 정지 및 Enter버튼을 클릭합니다. 드라이버가 빨간색 일 때 모터모드에서 중지됩니다.

2.5.3 위치속도모드

- 1) 2.5.1의 제어모드선택 부분 내용을 참조하여 위치속도모드로 선택합니다. 제어모드를 확인후 프로그램에서 대응되는 위치버튼을 클릭합니다.
- 2) CAN ID 확인(시리얼포트,사양설정페이지 등을 통해 확인가능, 테스트페이지에서 읽기->설정 클릭해도 되십니다.)
- 3) 모터모드에서 "Enter"버튼 클릭하면 드라이버는 초록색으로 모터모드가 진행됩니다.
- 4) 사양설정. 모터는 설정한 속도로 지정한 위치에 도착



사양설정전에 모터의 초기위치를 확인하셔합니다. 해당 위치를 참고하여 사양에 대한 설정을 진행합니다

예시: 사양설정, 위치 10rad, 속도 5rad/s. 샌딩타임,업데이트 및 보내기를 클릭하면 그래프확인이 가능합니다. **모터를 고정시켜줘야 합니다.**

원하시는 사양에 따라 제어사양을 변경하여 그래프 변화를 확인하고 싶으시면 직접 사양을 변경후 샌딩타임 및 업로드버튼을 클릭하시면 되십니다.

프로그램에서는 모터와 드라이버의 온도 및 모터 운전상태도 실시간으로 표시됩니다. 피드백프레임을 통해확인도 가능합니다. 피드백 프레임 통해서도 확인이가능합니다. 피드백프레임 양식 및 고장원인은 4.1 피드백프레임 부분 참고바랍니다.

- 5) 프로그램 조정상태에서 중지하고 싶으시면 정지 및 Enter버튼을 클릭합니다. 드라이버가 빨간색 일 때 모터모드에서 중지됩니다.

2.5.3 속도모드

- 1) 2.5.1의 제어모드선택 부분 내용을 참조하여 속도모드로 선택합니다. 제어모드를 확인후 프로그램에서 대응되는 속도버튼을 클릭합니다.
- 2) CAN ID 확인(시리얼포트, 사양설정페이지 등을 통해 확인가능, 테스트페이지에서 읽기->설정 클릭해도 되십니다.)
- 3) 모터모드에서 "Enter"버튼 클릭하면 드라이버는 초록색으로 모터모드가 진행됩니다.
- 4) 사양설정. 모터는 설정한 속도로 구동합니다.



예: 사양설정 속도 5rad/s. 업데이트 및 보내기 버튼을 클릭하면 되십니다. **모터를 고정시켜줘야 합니다.**

원하시는 사양에 따라 제어사양을 변경하여 그래프 변화를 확인하고 싶으시면 직접 사양을 변경후 센딩타임 및 업로드버튼을 클릭하시면 되십니다.

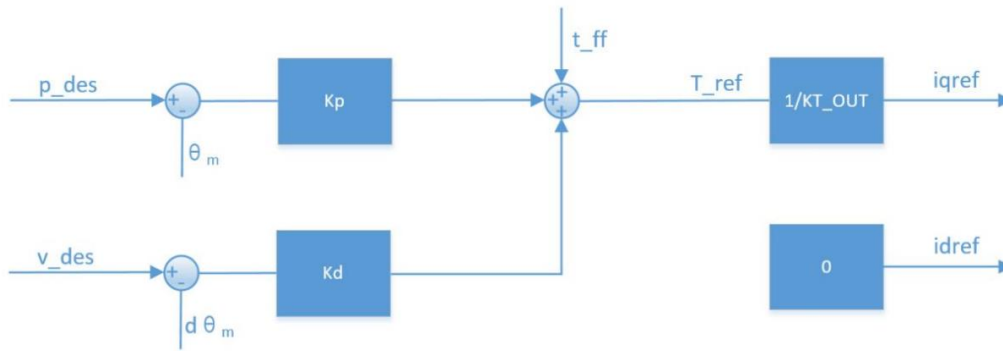
프로그램에서는 모터와 드라이버의 온도 및 모터 운전상태도 실시간으로 표시됩니다. 피드백프레임을 통해확인도 가능합니다. 피드백 프레임 통해서도 확인이가능합니다. 피드백프레임 양식 및 고장원인은 4.1 피드백프레임 부분 참고바랍니다.

- 5) 프로그램 조정상태에서 중지하고 싶으시면 정지 및 Enter버튼을 클릭합니다. 드라이버가 빨간색 일 때 모터모드에서 중지됩니다.

3. 작동모드

3.1 MIT모드

MIT모드는 원활하게 전환을 동시에 제어범위(P_MAX,V_MAX,T_MAX)도 유연하게 설정할수 있습니다. 전자속도제어기(esc)는 수신받은 CAN데이터를 제어변수로 전환 후 연산하여 나온 토크값을 루프 전류의 전류설정합니다. 루프전류는 조절규칙에 따라 최종적으로 설정한 토크전류에 도달할 수 있습니다



MIT모드에 따라 여러가지 제어모드가 파생됩니다.

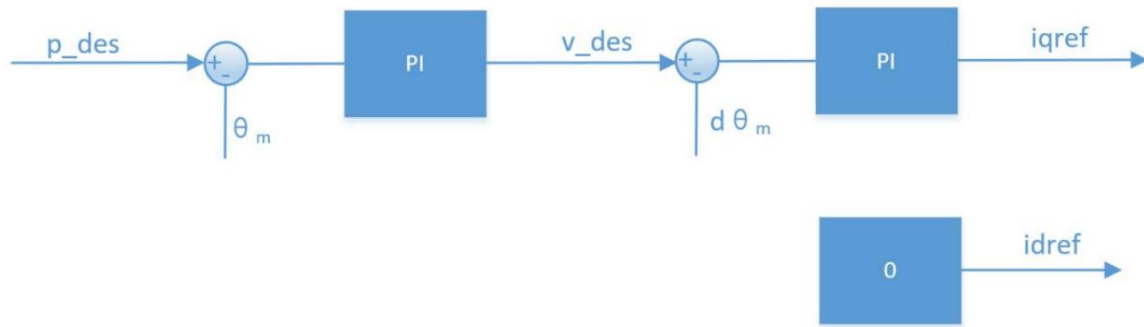
예: $k_p=0, k_d \neq 0$ 이 아닐 때 , v_{des} 로 설정하면 정속운전이 가능합니다

$k_p=0, k_d=0$, t_{ff} 를 목표로 설정하면 목표토크출력이 가능합니다.

*위치에 대해 제어시 k_d 값은 0으로 설정하시면 안됩니다. 모터가 고장날 가능성이 있습니다.

3.2 위치속도모드

위치 다단제어모드는 3단제어모드를 사용하고 있습니다. 위치루프(최외환)의 출력은 속도루프의 목표이며 속도루프의 출력은 전류루프의 목표로서 실제전류의 출력을 제어하는데 사용됩니다. 제어방식은 하단 그림 참조.

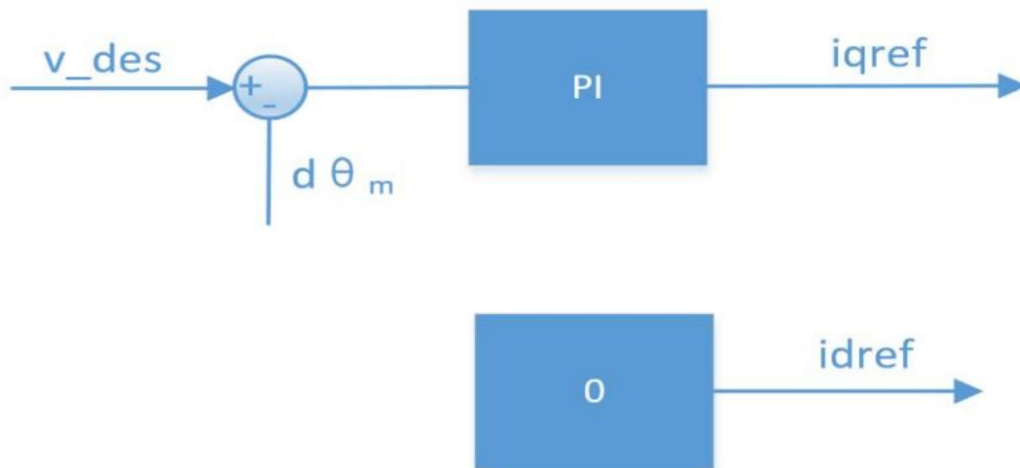


p_des 는 제어하려는 목표위치, v_des 는 작동중 최대절대속도값을 제한하는데 사용됩니다. 프로그램을 통해 위치 다단제어모드를 제어하는 경우 제어정밀도가 높으며 상대적으로 유연하지만 응답시간이 다소 길입니다. 추가로 가감속속도설정도 가능합니다.

* p_des , v_des 단위는 rad 와 rad/s, 데이터타입은 float, 댐핑팩터(damping factor)는 꼭 0이 아닌 양수로 설정하셔야 합니다.

3.3 속도모드

속도모드는 설정한 속도에서 안정적으로 운행할수 있게끔 하는 모드 입니다



* v_des 단위 rad/s, 데이터타입은 float, 프로그램을 사용하여 자동으로 스펙을 계산할 경우 댐핑팩터(damping factor)는 꼭 0이아닌 양수로 설정하셔야 합니다. 보편적으로 2.0~10.0사이의 값을 취합니다. 추천값: 4.0

4. CAN통신

앞단계에서 보정, 목표설정, 사양설정을 완료하시면 사용이 바로 가능합니다. 제어부분은 CAN표준 프레임양식을 기준으로 합니다. 고정 보드속도 1Mbps, 기능에 따라 수신프레임과 송신(피드백)프레임으로 나눕니다.

수신프레임: 수신받은 제어데이터로 모터에 명령을 내려 제어

송신프레임: 모터가 상위제어기로 모터상태를 송신하는 데이터

모터의 모드에 따라 수신프레임 양식 정의 및 프레임ID는 다르지만 송신프레임은 모든 모드에서 동일합니다

4.1송신프레임

송신프레임ID는 MasterID를 통해 설정할수 있으며 기본값은 0입니다. 주요로 모터의 위치,속도,토크에 관한 내용을 피드백합니다

	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	ID ERR<<4	POS[15:8]	POS[7:0]	VEL[11:4]	VEL[3:0] T[11:8]	T[7:0]	T_MOS	T_Rotor

ID: 제어기의 ID ,CAN_ID의 Low 8 바이트 적용

ERR: 에러

8: 과접압

9: 저전압

A: 과전류

B: MOS 과열

C: 모터권선과열

D: 통신실패

E: 과부하

POS 모터위치

VEL 모터속도

T 모터토크

T_MOS MOS의평균온도 °C

T_Rotor 모터내부권선의 평균온도 °C

위치,속도,토크는 선형매핑의 관계를 통하여 부동소수점데이터를(Floating Point Value) 부호가 있는 고정소수점데이터로 변환시킵니다. 위치값은 16비트, 속도와 토크는 12비트.

예시: 모터 속도가 25.0rad/s, 설정한 속도범위 VMAX=45rad/s일 때

송신데이터는 $VEL=25.0/(45-(-45))*2^{12}+2^{11}=3185=0xC71$

4.2MIT모드 제어프레임

	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	p_des [15:8]	p_des [7:0]	v_des [11:4]	v_des[3:0] Kp[11:8]	Kp [7:0]	Kd [11:4]	Kd[3:0] t_ff[11:8]	t_ff[7:0]r

프레임ID=설정한 CAN ID값

P_des: 위치목표

V_des: 속도목표

Kp: 위치비례계수

Kd: 위치미분계수

T_ff : 토크목표값

p_des,v_des,t_ff의 범위는 프로그램을 통해 설정이 가능합니다. Kp 범위[0,500], Kd 범위[0,5].

표준CAN데이터는 1 프레임당 8바이트만 있습니다. MIT의 제어명령형식은 Position, Velocity, Kp, Kd, Torque 5개의 스펙을 8바이트에 조합을 시킵니다. Position은 2바이트 16비트, Velocity 12비트,Kp12비트,Kd12비트.

MIT명령은 부동소수점데이터를 같은 비율로 정수로 변환하여 드라이버에 송신합니다. 드라이버는 수신한 정수를 동일비율로 부도소수점데이터로 변환합니다. 이부분에서는 uint_to_float 、float_to_uint 2개의 변환함수를 통하여 진행하며 실현방식은 하단과 같습니다.

```

1. float uint_to_float(int x_int, float x_min, float x_max, int bits){
2.     /// converts unsigned int to float, given range and number of bits ///
3.     float span = x_max - x_min;
4.     float offset = x_min;
5.     return ((float)x_int)*span/((float)((1<<bits)-1)) + offset;
6. }

```

```

1. int float_to_uint(float x, float x_min, float x_max, int bits){

2.     /// Converts a float to an unsigned int, given range and number of bits
///
3.     float span = x_max - x_min;
4.     float offset = x_min;
5.     return (int) ((x-offset)*((float)((1<<bits)-1))/span);
6. }

```

해당 변환함수는 우선 동일비율로 변환할 최소 및 최대값을 확인해야 하며 해당 값은 사양설정페이지에서 조회됩니다. KP,KD의 최대 및 최소값은 0.0~500.0, 0.0~5.0입니다. Pos, Vel, Torque 는 하단과 같이 ± 12.5 , ± 45 , ± 18 로 설정합니다. 해당 3개의 값은 모터의 실제사양에 따라 조정가능합니다. 단 제어명령을 송신시 꼭 설정값과 동일해야 하며 다를 시 동일비율로 축소될수 있습니다.

MIT제어모드 송신함수

```

1. void ctrl_motor(CAN_HandleTypeDef* hcan,uint16_t id, float _pos, float _vel,
float _KP, float _KD, float _torq)
2. {
3.     uint16_t pos_tmp,vel_tmp,kp_tmp,kd_tmp,tor_tmp;
4.     pos_tmp = float_to_uint(_pos, P_MIN, P_MAX, 16);
5.     vel_tmp = float_to_uint(_vel, V_MIN, V_MAX, 12);
6.     kp_tmp = float_to_uint(_KP, KP_MIN, KP_MAX, 12);
7.     kd_tmp = float_to_uint(_KD, KD_MIN, KD_MAX, 12);
8.     tor_tmp = float_to_uint(_torq, T_MIN, T_MAX, 12);
9.
10.    hcan->pTxMsg->StdId = id;
11.    hcan->pTxMsg->IDE = CAN_ID_STD;
12.    hcan->pTxMsg->RTR = CAN_RTR_DATA;
13.    hcan->pTxMsg->DLC = 0x08;
14.    hcan->pTxMsg->Data[0] = (pos_tmp >> 8);
15.    hcan->pTxMsg->Data[1] = pos_tmp;
16.    hcan->pTxMsg->Data[2] = (vel_tmp >> 4);
17.    hcan->pTxMsg->Data[3] = ((vel_tmp&0xF)<<4)|(kp_tmp>>8);
18.    hcan->pTxMsg->Data[4] = kp_tmp;
19.    hcan->pTxMsg->Data[5] = (kd_tmp >> 4);
20.    hcan->pTxMsg->Data[6] = ((kd_tmp&0xF)<<4)|(tor_tmp>>8);
21.    hcan->pTxMsg->Data[7] = tor_tmp;
22.
23.    HAL_CAN_Transmit(hcan, 100);
24. }

```

해석함수 수신(3가지모두 동일)

```

1. void HAL_CAN_RxCpltCallback(CAN_HandleTypeDef* _hcan)

```

```

2. {
3.  if(HAL_GetTick() - FlashTimer>500){
4.
5.   HAL_GPIO_TogglePin(LED1_GPIO_Port,LED1_Pin);
6.   FlashTimer = HAL_GetTick();
7.  }
8.
9.  //ignore can1 or can2.
10.  if(_hcan->pRxMsg->StdId == 0)
11.   {
12.    p_int=(_hcan->pRxMsg->Data[1]<<8)|_hcan->pRxMsg->Data[2];
13.    v_int=(_hcan->pRxMsg->Data[3]<<4)|(_hcan->pRxMsg->Data[4]>>4);
14.    t_int=((_hcan->pRxMsg->Data[4]&0xF)<<8)|_hcan->pRxMsg->Data[5];
15.    position = uint_to_float(p_int, P_MIN, P_MAX, 16); // (-12.5,12.5)
16.    velocity = uint_to_float(v_int, V_MIN, V_MAX, 12); // (-45.0,45.0)
17.    torque   = uint_to_float(t_int, T_MIN, T_MAX, 12); // (-18.0,18.0)
18.  }
19.  /*#### add enable can it again to solve can receive only one ID problem!!!
####*/
20.  __HAL_CAN_ENABLE_IT(&hcan1, CAN_IT_FMP0);
21. }

```

4.3위치속도모드하에 제어프레임

	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x100+ID	p_des				v_des			

ID 는 설정한 CAN ID 값에서 0x100 뺀셈.

P_des : 위치목표, 플로트형 로우비트가 앞 하이비트가 뒤

V_des : 속도목표, 플로트형 로우비트가 앞 하이비트가 뒤

송신명령의 CAN ID는 0x100+ID입니다. 속도목표는 사다리꼴형 가속도의 최고속도이며 정속부분의 속도값입니다.

위치모드 함수 송신

```

1. void ctrl_motor2(CAN_HandleTypeDef* hcan,uint16_t id, float _pos, float _vel
)
2. {
3.     uint8_t *pbuf,*vbuf;
4.     pbuf=(uint8_t*)&_pos;
5.     vbuf=(uint8_t*)&_vel;
6.
7.     hcan->pTxMsg->StdId = id;
8.     hcan->pTxMsg->IDE = CAN_ID_STD;
9.     hcan->pTxMsg->RTR = CAN_RTR_DATA;
10.    hcan->pTxMsg->DLC = 0x08;
11.    hcan->pTxMsg->Data[0] = *pbuf;
12.    hcan->pTxMsg->Data[1] = *(pbuf+1);
13.    hcan->pTxMsg->Data[2] = *(pbuf+2);
14.    hcan->pTxMsg->Data[3] = *(pbuf+3);
15.    hcan->pTxMsg->Data[4] = *vbuf;
16.    hcan->pTxMsg->Data[5] = *(vbuf+1);
17.    hcan->pTxMsg->Data[6] = *(vbuf+2);
18.    hcan->pTxMsg->Data[7] = *(vbuf+3);
19.
20.    HAL_CAN_Transmit(hcan, 100);
21. }

```

위치모드 해석함수와 MIT모드는 동일합니다.

4.4속도모드 제어 프레임

	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0x200+ID	v_des							

ID 는 설정한 CAN ID 값에서 0x200 옵셋.

V_des: 속도목표, 플로트형 로우비트가 앞 하이비트가 뒤

송신명령의 CAN ID는 0x200+ID.

속도모드 함수 송신

```

1. void ctrl_motor3(CAN_HandleTypeDef* hcan,uint16_t id, float _vel)
2. {
3.     uint8_t *vbuf;
4.     vbuf=(uint8_t*)&_vel;
5.
6.     hcan->pTxMsg->StdId = id;
7.     hcan->pTxMsg->IDE = CAN_ID_STD;
8.     hcan->pTxMsg->RTR = CAN_RTR_DATA;
9.     hcan->pTxMsg->DLC = 0x04;
10.    hcan->pTxMsg->Data[0] = *vbuf;
11.    hcan->pTxMsg->Data[1] = *(vbuf+1);
12.    hcan->pTxMsg->Data[2] = *(vbuf+2);
13.    hcan->pTxMsg->Data[3] = *(vbuf+3);
14.
15.    HAL_CAN_Transmit(hcan, 100);

```

```

16. }

```

속도모드 해석함수와 MIT모드는 동일합니다.

4.5 모터모드 시작

전원인가후 Enter 명령을 송신해야 제어가 가능합니다. Enter프레임은 제어프레임에 포함됩니다. 모든 모드에서 Enter 데이터의 정의는 동일합니다.

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFC

4.6 모터모드 종료

Enter 프레임은 제어프레임에 포함됩니다. 데이터정의는 하단과 같습니다.

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFD

4.7 원점위치저장프레임은 제어프레임에 포함됩니다. 데이터정의는 하단과 같습니다.

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFE

부록: 사양설명

모터사양	상저항 (R)	상과상 사이의 저항값, 단위 mR
	상인덕턴스 (L)	상과상 사이의 인덕턴스, 단위 uH
	쇄교자속 λ /magnetic linkage	코일이나 각 권선과 쇄교하는 자속의 총수 단위Wb
	점성계수coefficient of viscosity	모터의 점성계수는 참고용으로 사용되지 않음, N*m*s/rad
	관성모멘트Moment of Inertia	모터의 관성모멘트, 최대한 무부하를 유지. kg*m ²

구동사양	극쌍수	모터의 극쌍수. 수동입력가능.보정을 통해 자동입력가능.
	감속비	감속기의 감속비.
	CAN ID	CAN통신에 사용되는 모터ID
	Master ID	모터가 CAN통신에 CAN StdID
	저전압	저전압보호값,단위V
	과전압	과전압보호값,단위V
	과열	모터과열보호값 °C
	CAN Timeout	CAN통신 타임아웃주기 설정, 단위 50us
	가속도	모터 가속도 설정값, 단위Krad/s ²
	감속도	모터 감속도 설정값, 단위 Krad/s ² , 최대한 가속도와 동일해야 함
	속도제한	모터의 이론상 최대속도. 감속기가 있을시 감속전의 값, 단위 rad/s
	과전류	모터 상전류 최대값의 과전류설정비율

진폭제어	PMAX	MIT모드에서 송신된 명령의 최대위치값 및 모든 모드하에서 위치 피드백에 사용된 최대값, 단위 rad
	VMAX	MIT모드에서 송신된 명령의 최대속도값 및 모든 모드하에서 속도 피드백에 사용된 최대값, 단위 rad/s
	TMAX	MIT모드에서 송신된 명령의 최대토크값 및 모든 모드하에서 토크 피드백에 사용된 최대값, 단위 Nm
	KT_OUT	드라이버에서 사용된 출력토크계수값, 예0.0이면 기본적으로 시스템의 이론값을 사용, 단위 Nm/A
	기어계수	기어토크의 전동효율, 기본값 1
	댐핑팩터(damping factor)	속도루프 댐핑팩터 설정값.기본값 4.0,추천범위 2.5~20 .

제어설정	제어모드	3가지모드 포함. 즉 MIT모드.위치속도모드(PV)와 속도모드
	전류대역폭	전류루프의 제어대역폭. 실제에 따라 조절.
	속도 KP	속도루프의 비례사양. 자동계산.
	속도 KI	속도루프의 적분계수. 자동계산.
	위치 KP	위치루프의 비례사양, 위치속도모드시 유효. kp계산을 클릴하여 자동계산가능.
	위치 KI	속도루프의 적분계수. 0.0으로 설정하는 것을 추천.