2ms CAN 통신/제어기 2CH	
요구조건	2ms(500hz) 제어 B/W , 12 Axis
구성방법	- 2ch 모터 driver 6개 사용 - CAN 2 Port (각 port당 3개 제어기 연결)
명령	전류명령
응답	엔코더값
명령 크기	1 Message(8bytes)/ 1 Motor Driver-2ch
응답 크기	1 Message(8bytes)/ 1 Motor Driver-2ch
명령 range	0x4C4B40 + (-1023 ~ +1023) (전류명령)
응답 range	0~9,999,999 (제어기 reset시 기본 값 5,000,000=0x4C4B40)
명령데이터 구성	[1 byte char][6 bytes binary][1byte checksum]
응답데이터 구성	binary data

2ms CAN 통신/제어기 2CH - can 메시지 예제	
1)명령 예제 - 명령어 p - big-endian	모터제어기에 각 채널에 전류명령 각각 516(채널 1) , -436(채널 2) 를 전송할때, 생성해야할 can message data * 편이상 byte데이터를 hex 표현 val1=5000000+516 = 5000516 = 0x4C4D44 val2=5000000-436 = 4999564 = 0x4C498C CAN_MSG.DATA[0]='p' CAN_MSG.DATA[1]=0x4C CAN_MSG.DATA[2]=0x4D CAN_MSG.DATA[3]=0x44 CAN_MSG.DATA[4]=0x4C CAN_MSG.DATA[5]=0x49 CAN_MSG.DATA[6]=0x8C CAN_MSG.DATA[7]= Check sum, [1]~[6] 데이터 합의 Inverse, (BYTE)(~(0x4C+0x4D+0x44+0x4C+0x49+0x8C) & 0xFF) CAN_MSG.LEN=8 cf.) 0~3(채널 1번 데이터), 4~7(채널 2번 데이터)
2)응답예제 - big-endian	-응답이 다음과 같을때, * 편이상 byte데이터를 hex 표현 * 응답데이터 주소를 상위, 하위주소 순으로 설정한경우 CAN_MSG.DATA[0]=0x00 CAN_MSG.DATA[1]=0x4C CAN_MSG.DATA[2]=0x4B CAN_MSG.DATA[3]=0x3F CAN_MSG.DATA[4]=0x00 CAN_MSG.DATA[5]=0x4C CAN_MSG.DATA[6]=0x4B CAN_MSG.DATA[7]=0x41 CAN_MSG.LEN=8 cf) 0x4C4B3F =4,999,999 , cf) 0x4C4B41 =5,000,001

2ms CAN 통신/제어기 2CH	
제어기 설정예	

2) 통신 주기 및 통신 에러허용횟수	- 명령 : St
	- 기능 : 2채널 모터의 경우, 각 채널당 데이터 전송간격,통신에러 허용횟수 설정
	Ex) 2ms 간격으로 데이터가 들어온다고 알려주기
	
	String="St2,2,5;" 을 CAN으로 전송
	
	CAN_MSG.DATA[0]='S'
	CAN_MSG.DATA[1]='t'
	CAN_MSG.DATA[2]='2' //2번 채널 데이터 2ms간격으로 host->제어기 전송
	CAN_MSG.DATA[3]=','
	CAN_MSG.DATA[4]='2' //2번 채널 데이터 2ms간격으로 host->제어기 전송
	CAN_MSG.DATA[5]=','
	CAN_MSG.DATA[6]='5' //통신에러 허용횟수
	CAN_MSG.DATA[7]=';'
	CAN_MSG.LEN=8

3) 응답 주소 설정	- 명령 : Sr - 기능 : 전류명령 수신후 HOST로 전송되는 데이터를 가진 주소 설정 Ex) 엔코더 데이터 주소 * 제어기 및 버전별로 다름 * 일반 DC 2채널 제어기 기준으로 설명 채널 1번 엔코더 주소 829B, 829A (상위,하위 주소 순으로 설정할 경우 : big-endian) 채널 2번 엔코더 주소 82AD, 82AC (상위,하위, 주소 순으로 설정할 경우: big-endian)

1st msessage packet	2nd msessage packet
CAN_MSG.DATA[0]='S'	CAN_MSG.DATA[0]='2'
CAN_MSG.DATA[1]='r'	CAN_MSG.DATA[1]='9'
CAN_MSG.DATA[2]='8'	CAN_MSG.DATA[2]='A'
CAN_MSG.DATA[3]='2'	CAN_MSG.DATA[3]=','
CAN_MSG.DATA[4]='9'	CAN_MSG.DATA[4]='8'
CAN_MSG.DATA[5]='B'	CAN_MSG.DATA[5]='2'
CAN_MSG.DATA[6]=','	CAN_MSG.DATA[6]='A'
CAN_MSG.DATA[7]='8'	CAN_MSG.DATA[7]='D'
CAN_MSG.LEN=8	CAN_MSG.LEN=8

3rd msessage packet	1st, 2nd, 3rd msessage packet 을 연속으로 제어기에 전송
CAN_MSG.DATA[0]=',' CAN_MSG.DATA[1]='8' CAN_MSG.DATA[2]='2' CAN_MSG.DATA[3]='A' CAN_MSG.DATA[4]='C' CAN_MSG.DATA[5]=';' CAN_MSG.DATA[6]='₩0' CAN_MSG.DATA[7]='₩0' CAN_MSG.LEN=6	- 정상적으로 전송되면, 응답은 같은 데이터가 host로 전송됨.

2ms CAN 통신/제어기 2CH				
제어기 설정 및 구동	1) 통신 주기 및 통신에러허용횟수 설정	"St2,2,5;"	전송	* 2ms, 2ms, 5회
	2) p 명령 활성화	"Se0303;"	전송	* 주기 명령 수신 및 동작 기능 활성화
	3) 응답 주소 설정	"Sr829A,829B,82AC,82AD;"	전송	* 주소는 ASCII-HEX * 제어기 종류별로 Address는 다를수 있음. * 본 예제는 2811 DSP기반, 2CH, DC 모터, Encoder 버전의 제어기 경우임. * 각 제어기의 엔코더 위치값 저장 변수 address
	4) 서보 ON	"PE0001;"	전송	* 제어기 주소가 1번일때 PE0001 * 제어기 주소가 2번이면 PE0002 * 제어기 주소가 10번이면 PE000A
	5) 작동모드 설정	"SM0707;"	전송	
	6) 전류명령을 주고 응답을 받는 LOOP 실행	"p....."	전송	
	7) 주기명령 종료	"PPE!"	전송	* ! 를 붙여서 보내면 응답하지 않음. * 더이상 전송할 데이터가 없을 경우 반드시 보낸다. * "p....."를 보내다, 보내지 않으면, 통신에러 허용횟수 시간만큼 이전데이터를 기준으로 보간 이동하다, 오류횟수를 넘으면, 마지막 받은 데이터값으로 제어기 명령이 수정됨.
	8) PPE후 다시 "p....."를 보내면 재동작			*다시 명령을 보낼때, 급격한 전류명령을 보내지 말것 * p 명령은 가감속 rate를 상위제어기에서 담당하는 것으로 판단하고, 내부적으로 rate를 별도로 조정하지 않음.