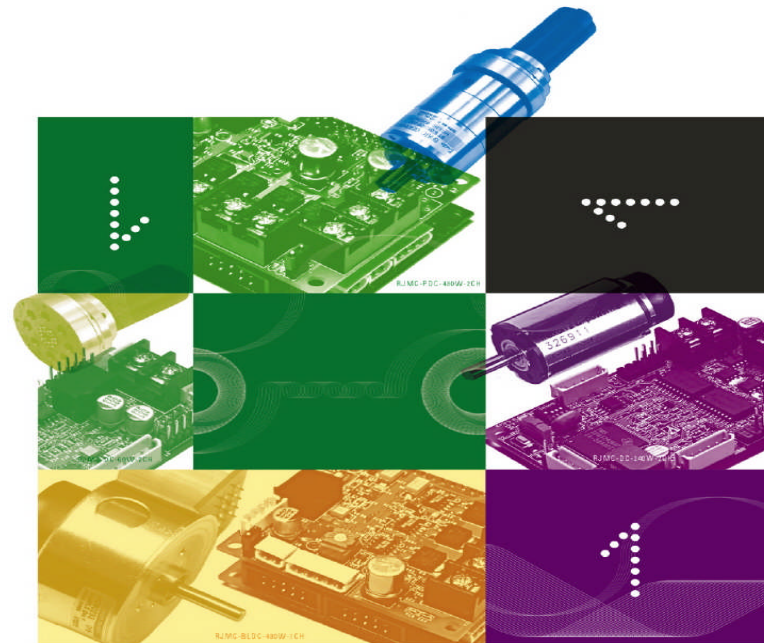


Network Based Digital Motor Controller

모터 제어기 명령어 파일



제어기 명령어 LIST



<http://www.robocube.co.kr>



Revision: April/15/2015

firmware upgrade 정보

날짜	대상제어기	수정내용	비고
2013-11-18	analog 또는 resolver 센서를 사용하는 모터 제어기 (속도값 계산관련)	1. MOTORx_SPEED_CALCULATION_SCALE_FACTOR 파라미터를 도입하여 analog 또는 resolver 센서를 사용할때 속도계산값의 스케일을 바꿀수 있도록 하였음. 이를 위하여 Ms 명령을 추가하였음.	
2013-11-19	모든 모터제어기 (02->01 제어모드 자동 전환 관련)	1. SX 명령어 두번째 파라미터의 bit28가 0이면 정지시에 02번 제어모드에서 01번 제어모드로 전환하는 기능을 사용하게 하였음. (기존에는 자동화버전에서만 무조건 실행되게 하였음)	
2013-11-19	모든 모터제어기 (통신관련)	1. SX 명령어 세번째 파라미터의 bit0가 0이면 Pa/Pb/Pp/Pq/p/Qb9에서 모터2의 데이터를 수신/송신에서 제외하도록 하였음.	
2013-11-24	모든 모터제어기 (QR/QT 명령관련)	1. QR/QT 명령에서 데이터수를 8개에서 16개 까지 할수 있도록 하였음.	
2013-11-26	모든 모터제어기 (CAN 통신관련)	1. CAN 통신에서 SX 명령어 두번째 파라미터의 bit27이 0이고 국방로봇제어기가 아닐 경우 상위제어기로 모터1의 상태값을 전송하는 기능을 추가하였음. 모터x의 상태값을 전송하는 주기와 시점 및 주소를 설정하도록 하였음. 이를 위하여 EU 명령을 도입하였음.	
2013-11-27	모든 모터제어기 (PW 명령관련)	1. DC 모터의 작동모드 00 에서 PW 명령을 사용할때 리미트센서에서 정지하는 기능을 사용하게 하였음.	
2014-1-28	모든 모터제어기 (EXT_FAULTx_port 의 사용 관련) (전류 fault 처리관련)	1. EXT_FAULTx_port 의 사용을 정립하였음. ZC 명령어로 설정하는 파라미터의 bit0/2 또는 bit1/3이 1이면 각각 모터1 또는 2의 오류상태를 출력하는 용도로 사용함. ZC 명령어로 설정하는 파라미터의 bit0 또는 bit1이 0이면 각각 EXT_FAULT1_port 또는 EXT_FAULT2_port 단자에 외부에서 사용하기 위한 3.3V 또는 5V 전압이 연결되어 있음 의미하며, 이 경우가 아닐때 ZC 명령어로 설정하는 파라미터의 bit2 또는 bit3이 0이면 각각 EXT_FAULT1_port 또는 EXT_FAULT2_port 단자에 X0 명령어로 설정하는 ON/OFF 출력이 인가됨을 의미함. 2. MOTORx_MAX_CURRENT_fault_flg=1 일때 PWM 구동 전원이 차단되므로 작동모드도 0으로 설정하고 모든 게이트도 off 하게 하였음.	

firmware upgrade 정보

날짜	대상제어기	수정내용	비고
2013-2-02	모든 모터제어기 (통신관련) (CAN 통신관련)	<ol style="list-style-type: none"> TWO_MOTOR_DATA 변수를 도입하고 SX 명령어로 설정하는 세번째 파라미터의 bit0가 00이면 모터2의 파라미터를 모든 모터제어기 명령에서 생략하도록 하였음. 두채널제어기에서 SX 명령어로 설정하는 세번째 파라미터의 bit0가 10이면서 세번째 파라미터의 bit1이 00이면 모든 모터제어기 명령에서 모터2의 파라미터를 모터1의 파라미터뒤로 옮기고 모이도록 하게 하였음. SX 명령어로 설정하는 세번째 파라미터의 bit3가 00이면 CAN 통신에서 p 명령에 대한 리턴을 하지 않도록 하였음. SX 명령어로 설정하는 세번째 파라미터의 bit20이 00이면 CAN 통신에서 주기적으로 모터2의 상태를 상위로 전송하는 기능을 추가하였음. 	
2014-2-13	모든 모터제어기 (브레이크저항 관련)	<ol style="list-style-type: none"> 브레이크저항의 작동을 시험할 수 있도록 하였음. 이를 위하여 ZP 명령을 수정하였음. 	
2014-3-09	모든 모터제어기 (CAN 통신관련)	<ol style="list-style-type: none"> CAN 모드일때 SX 명령어로 설정하는 세번째 파라미터의 bit4가 00이면 0xff 번 broadcast 주소로 상위제어기로 부터 명령어가 수신된 경우에도 값을 리턴하도록 하였음. 	
2014-3-21	모든 모터제어기 (가상원점 관련) (자동화버전 주소값 설정 관련)	<ol style="list-style-type: none"> imaginary_origin1, imaginary_origin2 변수를 도입하고 QE5AA5; QEAA55; QE55AA; PC 명령어를 도입하여 가상원점을 설정하고 이를 기준으로 위치명령값을 사용할수있도록 하였음. 자동화버전에서 SX 명령어로 설정하는 세번째 파라미터의 bit5가 00이면 로타리 스위치의 값을 모터제어기 주소를 사용하게 하였음. (부팅시에만 적용) 	
2014-3-22	모든 모터제어기 (가상원점 관련) (firmware 오류 수정)	<ol style="list-style-type: none"> QPC; Qpc; 명령을 추가하여 가상원점 기준의 위치값 또는 위치명령값을 읽을수 있도록 하였음. Pp/Pq, Qb1/2; 명령의 리턴값에서 값을 쪼갤때 인터럽트가 걸리면 값이 변경되는 문제를 수정하였음. 	이전 버전에서 Pp/Pq, Qb1/2; 명령의 리턴값이 잘못된 데이터로 리턴될 확률이 매우 작지만 있음.
2014-4-09	모든 모터제어기 (firmware 오류 수정)	<ol style="list-style-type: none"> positionx_increment 변수를 long 변수로 선언하고 long으로 처리되게 하였음. 이전 버전에서 PP/PQ/Pp/Pq/p 명령등에서 증분이 16비트를 넘어가는 경우 문제가 있었음. 	이전 버전에서 PP/PQ/Pp/Pq/p 명령등에서 증분이 16비트를 넘어가는 경우 문제가 있었음.
2014-5-25	모든 모터제어기 (CAN 통신관련)	<ol style="list-style-type: none"> Eu, Ev 명령을 도입하여, 일정주기로 CAN 통신으로 상위제어기에 데이터를 전송할 때, 변수를 지정할수 있도록 하였음. 1채널 모터제어기에서도 2채널 제어기만큼의 데이터 전송이 되게 하였음. 	

firmware upgrade 정보

날짜	대상제어기	수정내용	비고
2014-7-14	모든 모터제어기 (LSxC 포트 모니터관련)	1. Do 명령을 추가하여 LSxC 포트의 전압값을 필터일한 값을 읽을수 있도록 하였음.	
2014-9-23	모든 모터제어기 (흡동작, 원점 또는 가상원점 재설정 관련)	1. GH 등의 명령을 사용하여 원점찾기 동작을 하거나, QE 명령에 의하여 원점 또는 가상원점을 재설정을 하는 부분에서 오류가 있었던 부분을 수정하였고, 이러한 기능을 사용할 수 없는 경우에도 절대각 센서를 사용하는 경우를 제외하고는 사용 가능하게 하였음. 2. QEi; 명령을 추가하여 가상원점 설정값을 읽을수 있도록 하였음. 3. Qx134; Qx135; 명령을 추가하여 흡동작 과정을 볼수 있도록 하였음.	
2014-10-03	fault 출력단자가 있는 모든 모터제어기 (fault 출력단자를 통한 펄스출력기능)	1. Xp, Xq 명령을 추가하여 일정 변위마다 fault 출력단자를 통하여 low 펄스출력을 내보내는 기능을 추가하였음. 2. Zp; 명령을 추가하여 PWM frequency와 전류제어 frequency를 읽을 수 있도록 하였음.	
2014-10-05	fault 출력단자가 있는 모든 모터제어기 (fault 출력단자를 통한 펄스출력기능)	1. Xr; 명령을 추가하여 Xp 명령으로 펄스출력을 개시할때의 위치저장값을 읽어낼 수 있게 하였음.	
2014-10-07	BLDC에서 엔코더를 안쓰는 경우 (firmware 오류 수정)	1. BLDC에서 엔코더를 안쓰는 경우에, 모터 stall시에 설정하는 위치명령 값이 잘못 설정되는 문제가 발견되어 수정하였고, 리밋 초과 상태를 알려주는 Q2 두번째 파라미터 설정, 리밋센서를 만났을때 강제 정지하는 기능도 같은 문제가 있어서 수정하였음.	오류가 있었으나 이 문제와 관련하여 판매된 제어기는 없음.
2014-11-04	IR21771S 를 전류측정에 사용하는 모든 모터제어기 (전류 offset 측정기능)	1. PE 명령을 수행할 때 PWM출력이 on 되는데, on 직후 fault로 간주되어 다시 off 되는 문제가 있어서 수정 하였음. (2013년~2014년 사이의 수정과정에서 발생한 문제임)	오류가 있었으나 이 문제와 관련하여 판매된 제어기는 없음.
2014-11-15	모든 모터제어기 (호밍동작 기능)	1. SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.	

firmware upgrade 정보

날짜	대상제어기	수정내용	비고
2014-11-22	fault 출력단자가 있는 모든 모터제어기 (fault 출력단자를 통한 펄스출력기능)	1. 2014-10-05 일자에 처리된 Xr 명령어 처리가 잘못되어서 수정하였음.	
2014-11-22	모든 모터제어기 (흡에서의 위치 기본값 설정)	1. HOME_AT_OL 매크로와 OFFSET_AT_HOME 매크로를 도입하여 흡위치를 0으로 선택할수 있도록 하였음. 단 firware가 구별됨. (BLDC 엔코더, Hall IC only의 경우는 확인됨)	고객의 요청에 의하여 5,000,000이 원점이 아니고 0 이 원점이 되게 할수 있으나 firmware는 별도임.
2014-11-23	digital Hall IC 만을 사용하는 BLDC 모터제어기 (흡동작 관련)	1. 2014-9-23일자 수정과정에서 digital Hall IC 만을 사용하는 BLDC 모터의 흡동작에서 위치 리밋을 잘못 적용하여 수정하였음.	2014-9-23일자 수정과정에서 digital Hall IC 만을 사용하는 BLDC 모터의 흡동작에서 위치 리밋을 잘못 적용하였으나 판매 적용을 하지 않은 상태임.
2014-11-24	모든 모터제어기 (흡으로 이동 관련)	1. 흡으로의 이동을 명령하는 PH0; PHo; 명령을 추가하였음.	
2014-12-14	관성센서	1. 관성센서에서 CAN 포트에 120 오옴이 달려있지 않으면 reset을 반복하는 현상이 있어서 수정하였음. 2014년 6월 14일 이후의 S/W 버전에서는 CAN 통신선이 연결되지 않은 상태에서는 baud rate를 설정하면 리셋되는 문제가 있었음. 모터제어기에서도 2014년 6월 14일 ~ 2014년 9월 12일 S/W 버전에서는 CAN 통신선이 연결되지 않은 상태에서는 baud rate를 설정하면 리셋되는 문제가 있음. 2. 관성센서에서 SA 명령어에 의한 주소값설정에 오류가 있어서 수정함.	주소값 설정오류는 2013년 11월 02일자 수정이 잘못됨에 기인하므로 2013년 11월 02일~2014년 12월 13일 사이의 버전만 문제됨.
2015-1-10	digital Hall IC 만을 사용하는 BLDC 모터제어기 (속도제어 관련)	1. 모터제어기에서 (operation_mode_SWITCH & 0x00800000L)!=0x0L 이면 05번 속도제어일때 정지후 출발하는 경우 위치값을 초기화하는 기능을 사용하지 않도록 하였음. 2010년 11월 30일 이전 S/W 버전에서는 위치값 초기화 기능이 없고, 2010년 11월 30일 S/W 버전부터는 위치값 초기화 기능을 사용하는 것이 기본임.	속도제어를 하면서 위치값을 사용하는 경우를 대비한 조치임.
2015-1-11	관성센서 (EKF 관련)	1. 관성센서에서 QC/QP/Qp/Qq0/Qq1/Qq2/Qq3/Qq4/b1/b2/b3/b4; 명령어를 추가하여 complementary_filter 또는 EKF에 의한 값을 읽어 낼 수 있도록 하였음.	

firmware udgrade 정보

날짜	대상제어기	수정내용	비고
2015-2-07	관성센서 (EKF 관련)	1. 관성센서가 up-side-down 되어 있을 때에도 SX 명령어의 bit8을 1로 설정하면 정상적인 계산을 할수 있도록 하였음.	
2015-3-17	TMS320F28334를 사용하는 모터제어기 버전모든 모터제어기 (시퀀스제어 관련)	1. <code>get_X_hex2address()</code> 함수에서 <code>0x3f8000</code> 주소를 사용하는 오류가 있어서 수정하였음.	시퀀스제어에서 주소값으로 변수의 값을 사용하는 경우에 문제가 됨.
2015-3-20	rack 장착형 전류제어기 (제어주파수 관련)	1. <code>cpu_pamp.c</code> 에서 PWM 주파수를 100KHz 까지 설정가능하게 하였음. 브레이크포트를 사용할수있게 하였음.	
2015-3-23	모든 모터제어기 (홀으로 이동 관련)	1. <code>Zp</code> 명령을 추가하여 PWM 주파수와 모터 1/2의 전류제어 주파수, 속도 및 위치제어 주파수를 읽어낼수 있게 하였음.	

firmware udgrade 정보

날짜	대상제어기	수정내용	비고

firmware upgrade 정보

날짜	대상제어기	수정내용	비고

firmware upgrade 정보

날짜	대상제어기	수정내용	비고

※ 읽기 전에

1 명령어 구분

각 명령어 뒤에 + # ^ 표시는 명령어가 적용되는 MCU종류를 나타냅니다. 다음 표는 모터 제어기 별로 할당된 + # ^ 기호를 보여줍니다. 단 각각의 명령어는 구동모터의 종류와 제어기 모델에 따라서 실제로는 적용이 되지 않는 경우가 있습니다.

모터	출력	모델명	사용전압	+ (TMS320F2811)	# (TMS320F28334/5)	^ (TMS320F2809)
DC	25W	CUBE-DC2401-SIQ, -DIQ	24V			○
	50W	CUBE-DC2402-DII, -6II	24/36V	○		
	100W	CUBE-DC2404-DIK	24/36/48V	○		
	200W	CUBE-DC2408-SID, -DID	24/36/48V	○		
	400W	CUBE-eDC2416-SIH, -SDH	24/36/48V	○		
	800W	CUBE-eDC2432-SDH	24/36/48V	○		
BLDC	25W	CUBE-BL2401-SIQ	24V			○
	50W	CUBE-BL2402-SII, -DII	24/36V	○		
	100W	CUBE-BL2404-DIK, -DIKR	24/36/48V	○		
	100W	CUBE-BL2404-DEKR, -DEK	24/36/48V	○	○	
	200W	CUBE-BL2408-SID, -SMD	24/36/48V	○		
	200W	CUBE-BL2408-DID	24/36/48V	○	○	
	400W	CUBE-BL2416-SIH, -SDH	24/36/48V	○		
	800W	CUBE-BL2432-SDH, -SAH	24/36/48/60V	○		
	800W	CUBE-BL2432-SIDS, -SAD	24/36/48/72V		○	
	800W	CUBE-BL1008-DIB	100/200V		○	
	800W	CUBE-BL2432-SEH, -SFH	24/36/48/60/72V		○	
	1500W	CUBE-BL3804-SEB	380VDC/220VAC		○	
	6000W	CUBE-BL3816-SEB, -SAB, -SFB	380VDC/220VAC		○	
STEP	50W	CUBE-ST2402-DII, -DIIE	24/36V		○	
	100W	CUBE-ST2405-DIKE, -SEKE, -DEKE	24/36/48V		○	

2 명령어 검색방법

명령어를 찾으실 때에는 찾기(Cntl + F)를 눌러 ";" + "명령어 앞 두글자" 를 치시면 쉽게 찾으실 수 있습니다. 명령어에서 ";"는 명령어의 끝을 지정하는 문자로서 명령어 앞쪽에는 붙일 필요가 없습니다.

2008년 이전 구매 고객께서는 당사에 문의하여 주십시오.

※ 읽기 전에

3 명령어 형식

☞ 각 명령어는 맨앞에 1~3자의 명령어 문자(ASCII code)가 위치하며, 그 뒤에 필요한 파라미터 값이 ',' 를 separator로 하여 나열된 다음, 마지막에 ';' 이나 '!'로 끝나는 형태로 되어있습니다. 파라미터값은 없거나 ?인 경우도 있습니다. (예로는 PA510000,520000; 또는 EsA55A! 또는 Qs; 또는 SA?;) (단 2012년 6월 3일 이전 S/W 버전에서는 RS485의 경우에 한하여 전송된 명령의 마지막 끝이 '!' 이 되는 것을 사용하면 안됩니다.)

☞ 모터제어기로 명령이 전송되면 약 200us이내에 처리를 끝내는데, 이때 모터제어기로 전송된 명령의 마지막 끝이 ';' 이면 명령어에 대한 리턴값이 모터제어기로부터 출력됩니다. 이전에 전송된 명령의 처리가 끝난 이후에만 다음 명령어가 처리될수 있습니다. 따라서 모터제어기로부터 이전명령어에 대한 리턴값을 수신한 이후에 다음 명령을 보내야 안전합니다.

☞ 모터제어기로 전송된 명령의 마지막 끝이 '!' 이면 명령어에 대한 리턴값이 모터제어기로부터 출력되지 않습니다. 이 경우에는 이전 명령어의 마지막 값 ('!')을 보낸 다음, 반드시 최소 200uS 이후에 다음 명령의 전송을 개시하여야 합니다. (단 2012년 6월 3일 이전 S/W 버전에서는 RS485의 경우에 한하여, 전송된 명령의 마지막 끝이 '!' 이 되는 것을 사용하면 안됩니다.)

☞ 파라미터 값은 다음 표과 같은 종류의 값들이 될수 있습니다.

파라미터값 표현형식	파라미터의 의미	사용 예
dddd	16비트 또는 32비트의 부호없는 값을 십진수로 표현한 값	1234 또는 1234567 (제어기로부터 출력되는 값을 표현하는 경우일때 long 변수로서 해당값이 너무 크면 ffffffff로 출력됨)
iiii	16비트의 부호있는 값을 십진수로 표현한 값	-1234
llll	32비트의 부호있는 값을 십진수로 표현한 값	+1234567 (제어기로부터 출력되는 값을 표현하는 경우일때 long 변수로서 해당값이 너무 크면 ffffffff로 출력됨)
ffff	부호와 소숫점이 있는 실수값을 9글자의 십진수로 표현한 값	-0.3456 (부호, 소숫점 포함 총 9글자이며 소숫점 이상과 이하가 각각 한자리 이상이어야함, 단 2012년 4월 16일 S/W 버전부터 명령값을 받을때 eeee 형식과 마찬가지로 추가로 3글자의 지수표현을 덧붙여 사용할수 있습니다.)

※ 읽기 전에

파라미터값 표현형식	파라미터의 의미	사용 예
gggg	부호와 소숫점이 있는 실수값을 11글자의 십진수로 표현한 값	-0.3456 (부호, 소숫점 포함 총 11글자이며 소숫점 이상과 이하가 각각 한자리 이상이어야함)
eeee	부호와 소숫점 및 지수가 있는 실수값을 9글자의 십진수와 3글자의 지수로 표현한 값	+10.2, -0.034, 100.0, -123.E+3, 0.2E-4 등의 형식을 사용할수 있으며, 지수는 -9 ~ +9까지만 허용됨. (부호, 소숫점 포함 총 9글자와 지수표현용 3글자로 구성되며 2012년 4월 16일 S/W 버전부터 사용가능합니다.)
HL	8비트의 부호없는 값을 16진수로 표현한 값	7A (십진수로는 $7 \times 2^4 + 10 = 122$ 에 해당함)
HLHL	16비트의 부호없는 값을 16진수로 표현한 값	1234 (십진수로는 $1 \times 2^{12} + 2 \times 2^8 + 3 \times 2^4 + 4 = 4660$ 에 해당함)
HLHLHL	24비트의 부호없는 값을 16진수로 표현한 값	12345B (십진수로는 $1 \times 2^{20} + 2 \times 2^{16} + 3 \times 2^{12} + 4 \times 2^8 + 5 \times 2^4 + 11 = 1193051$ 에 해당함)
HLHLHLHL	32비트의 부호없는 값을 16진수로 표현한 값	1234567C (십진수로는 $1 \times 2^{28} + 2 \times 2^{24} + 3 \times 2^{20} + 4 \times 2^{16} + 5 \times 2^{12} + 6 \times 2^8 + 7 \times 2^4 + 12 = 305419900$ 에 해당함)
LB	8비트의 부호없는 값을 한개의 바이트로 표현한 값	[0x0F] (여기서 [0x0F]는 00001111b 한바이트를 나타내며 십진수로 15에 해당함)
HBLB	16비트의 부호없는 값을 두개의 바이트로 표현한 값으로서 HB가 상위바이트임	[0x01][0x81] (십진수로는 $1 \times 256 + 128 + 1 = 385$ 에 해당함)
HMBLB	24비트의 부호없는 값을 3개의 바이트로 표현한 값	[0x01][0x02][0x03] (십진수로는 $1 \times 65536 + 2 \times 256 + 128 + 3 = 66051$ 에 해당함)
HbMbLbHbMbLb	12비트의 부호없는 값 2개를 총 3개의 바이트로 표현한 값	[0x51][0x42][0x03] (십진수로는 $5 \times 256 + 1 \times 16 + 4 = 1300$ 과 $2 \times 256 + 0 \times 16 + 3 = 515$ 두개의 숫자를 의미함)

☞ 상기의 표에서 HB/HBLB/HMBLB는 8/16/24비트의 값을 각각 1/2/3개의 바이트로 직접 표현한 것이며, HbMbLbHbMbLb는 12비트의 수 2개를 3개의 바이트로 표현한 것이고, 나머지의 경우는 십진수 또는 16진수로 변환한 값의 각 자리를 0/1/2/3/4/5/6/7/8/9/A/B/C/D/E/F를 나타내는 ASCII code인 [0x30]/[0x31]/[0x32]/[0x33]/[0x34]/[0x35]/[0x36]/[0x37]/[0x38]/[0x39]/[0x41]/[0x42]/[0x43]/[0x44]/[0x45]/[0x46]으로 표현한 것입니다.

※ 읽기 전에

예를 들어서 Sw`dddd1,dddd2`; 라는 형식으로 기술되어 있는 명령어에 대하여, 파라미터값 `dddd1, dddd2`의 값으로 100과 120을 사용한다면 명령어가 Sw100,120;이 되는데, 이때 Sw100,120;는 다음 10개의 바이트로 표현되며 맨 앞에 있는 바이트부터 모터제어기로 전송되어야 합니다.

```
[0x53][0x77][0x31][0x30][0x30][0x2c][0x31][0x32][0x30][0x3B]
  ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑
  'S'   'w'   '1'   '0'   '0'   ','   '1'   '2'   '0'   ';'

```

모터제어기가 Sw100,120;를 수신하면 필요한 처리를 하고 리턴값으로 Sw100,120;를 되돌려 줍니다. 되돌려주는 값 역시 받은 값과 같은 형식이며 [0x53][0x77][0x31][0x30][0x30][0x2c][0x31][0x32][0x30][0x3B] 순으로 총 10바이트를 순차적으로 host로 전송합니다.

예를 들어서 SA`HLHL`; 라는 형식으로 기술되어 있는 명령어에 대하여, 파라미터값 HLHL의 값으로 FE01을 사용한다면 명령어가 SAFE01;이 되는데, 이때 SAFE01;는 다음 7개의 바이트로 표현되며 맨 앞에 있는 바이트부터 모터제어기로 전송되어야 합니다.

```
[0x53][0x41][0x46][0x45][0x30][0x31][0x3B]
  ↑      ↑      ↑      ↑      ↑      ↑      ↑
  'S'   'A'   'F'   'E'   '0'   '1'   ';'

```

예를 들어서 Pa`HBMBLBHBMBLBCS`; 라는 형식으로 기술되어 있는 명령어에 대하여, 파라미터값 `HBMBLB`의 값으로 0x123456 과 0xEDF018를 사용하면 명령어가 Pa[0x12][0x34][0x56][0xED][0xF0][0x18];이 되는데, 이때 Pa[0x12][0x34][0x56][0xED][0xF0][0x18];는 다음 9개의 바이트로 표현되며 맨 앞에 있는 바이트부터 모터제어기로 전송되어야 합니다.

```
[0x50][0x61][0x12][0x34][0x56][0xED][0xF0][0x18][0x3B]
  ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑
  'P'   'a'   [0x12][0x34][0x56][0xED][0xF0][0x18]   ';'

```

※ 읽기 전에

예를 들어서 `bHbMbLbHbMbLbHbMbLbHbMbLbCS;` 라는 형식으로 기술되어 있는 명령어에 대하여, 파라미터값 `HbMbLbHbMbLbHbMbLbHbMbLb`의 값으로 `0x123, 0x456, 0xEDF` 과 `0x018`를 사용한다면 명령어가 `b[0x12][0x34][0x56][0xED][0xF0][0x18][0x6E];`이 되는데, `[0x6E]`는 앞선 6바이트의 합을 inverse한 값(checksum)으로서, `b[0x12][0x34][0x56][0xED][0xF0][0x18][0x6E];`는 다음 9개의 바이트로 표현되며 맨 앞에 있는 바이트부터 모터제어기로 전송되어야 합니다.

```

[0x62][0x12][0x34][0x56][0xED][0xF0][0x18][0x6E][0x3B]
  ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑      ↑
  'b'  [0x12][0x34][0x56][0xED][0xF0][0x18] [0x6E]  ';'

```

☞ 명령마다 리턴값의 형식이 일부 다를수 있으므로 개별적인 명령어의 설명을 참조하기 바랍니다.

☞ 명령어를 사용할때 ?를 사용하여 현재의 값이 얼마로 되어 있는지를 확인한 다음, 필요하다면 수정된 값을 명령어에 붙여서 다시 보내는 형식으로 사용하면 실수를 줄일수 있습니다.

예로서 `Sw?;` 를 모터제어기로 전송하면 제어에 사용하는 전류의 최대치를 리턴값으로 받아 볼수 있는데, 그 리턴값이 `Sw800,900;` 이라면 모터1의 제어전류 최대치가 800, 모터2의 제어전류 최대치가 900 이라는 뜻입니다. 그 값을 1000, 1000으로 바꾸고자 하면 `Sw1000,1000;`을 모터제어기로 보내면 됩니다.

☞ 저가형 1채널 제어기가 아닌 경우에는 명령어 형식에서 모터1 및 2의 파라미터가 모두 있는 경우, 1채널형 모터제어기 일지라도 모터1에 대한 값과 모터2에 대한 값을 모두 설정해야하고, 리턴값으로도 모터1과 모터2의 값이 모두 리턴 됩니다. 그러나 명령어 뒤에 위치하는 모터1의 파라미터 설정값 뒤에 모터2의 파라미터 설정값이 따라 붙는 형식의 명령어인 경우에는, 1채널형 모터제어기일 때에 한하여 모터2의 파라미터를 생략할수 있습니다. 그러나 리턴값으로는 모터1과 모터2의 값이 모두 리턴 되며, 이 때 모터2에 대한 리턴값은 의미가 없습니다.

예로서 1채널형 모터제어기인 경우에 `Sw1310,1310;` 명령어 대신에 `Sw1310;` 를 사용할수 있습니다. 그러나 리턴값은 `Sw1310,xxxx;` 이며 두번째 파라미터값은 무시하셔야 합니다.

※ 읽기 전에

- ☞ SX 명령어로 설정하는 세번째 파라미터의 LSB(bit0)를 0으로 설정하는 경우에는 'p', 'Pa', 'Pb', 'Pp', 'Pq', 'Qb9' 명령에 한하며 모터1의 값만을 설정하고, 리턴값도 모터1에 대한 리턴값만 리턴하도록 할수 있습니다. (2013년 11월 22일 S/W 버전부터) (2014년 2월 01일 S/W 버전부터는 모든 명령에 적용됨)
- ☞ SX 명령어로 설정하는 세번째 파라미터의 bit1을 0으로 설정하는 경우에는 2채널 제어기에서 모터2에 대한 데이터를 모터1에 대한 데이터 다음에 모아서 위치하도록 파라미터 순서를 바꾸어 적용하고, 리턴값도 바꾼 순서대로 리턴됨. 단, 리턴값으로 되돌아 오는 값에서 추가로 붙여지는 파라미터에 한해서는 사용자 매뉴얼에 기술된 순서대로 리턴됨. (2014년 2월 02일 S/W 버전부터 적용됨)
- ☞ 저가형 1채널 제어기에서는 명령어에 의한 파라미터 설정값이나 리턴값에서 모두 모터2에 대한 값이 생략 되었으므로 주의를 요함.
- ☞ 명령어 중에서 QC 와 Qc, QK 와 Qk, SI 와 Si, Sw 와 Sz, SC 와 Sc, SH 와 Sh, SK 와 Sk, MX 와 Mx, MR 와 Mr 명령어는 전류의 크기를 A/D 변환값으로 표현하는 경우와 0.01A 단위로 표현하는 경우에 각각 사용하는 명령어의 짝임. 단 Si, Sh, Sk, Mx, Mr 명령어는 2013년 4월 3일 S/W 버전부터 사용이 가능하며, 2013년 4월 3일 S/W 버전부터 MX 명령어는 이전의 Sx 명령어와 같고, SK 명령어는 이전의 Sh 명령어와 같음.
- ☞ 당사에서는 CUBEMON 이라고 하는 PC utility를 제공하고 있습니다. 이를 사용하면 terminal 창을 이용하여 모터제어기로 보낼 명령을 입력하고 이에 따른 리턴값을 표시해주는 기능을 사용할수 있습니다. 또한 여러 종류로 설정된 4가지의 변수값을 1ms 단위로 실시간 모니터링해볼수 있는 기능도 이용할 수 있습니다.
- ☞ 본 설명서중에 RJM_VERx 라는 명칭이 사용되고 있는데, 그 의미는 다음과 같습니다.
 RJM_VER1 또는 RJM_VER2 또는 RJM_VER3 : 현재는 생산하고 있지 않은 초기 H/W 버전의 DC 모터제어기를 가리킴.
 RJM_VER4 : 25W ~ 100W 급의 DC 모터제어기를 가리킴.
 RJM_VER8 : 200W 급 이상의 DC 모터제어기를 가리킴.
 RJM_VER5 : 2상 STEPPING 모터제어기를 가리킴.
 RJM_VER6 또는 RJM_VER7_3PHASE_STEP : 3상 STEPPING 모터제어기를 가리킴.
 RJM_VER7_5PHASE_STEP : 5상 STEPPING 모터제어기를 가리킴.
 RJM_VER7 : BLDC 또는 AC 서보 모터제어기를 가리킴.

RS485 통신과 CAN통신에 의한 다중 마이컴 통신

- ☞ RS485 통신을 사용하여 명령과 데이터를 모터제어기로 보내려고 하면, 앞서 기술된 명령 및 데이터를 보내기 전에 주소값에 해당하는 한 바이트를 먼저 전송하여야 합니다. 주소값은 0 ~ 125 까지 허용되는데, 0 ~ 125 범위의 주소값에 128을 더한 값을 주소값으로 전송하는 경우에는 제어기로부터 리턴값이 출력되지 않습니다. 제어기로부터 리턴값이 출력되지 않게하는 방법으로는 명령어 제일마지막에 삽입하는 terminator인 ';' 대신에 '!'를 사용하는 방법도 있습니다. (단 2012년 6월 3일 이전 S/W 버전에서는 RS485의 경우에 한하여, 전송된 명령의 마지막 끝이 '!' 이 되는 것을 사용하면 안됩니다.) 또 주소값이 255인 경우에는 모든 모터 제어기가 명령 및 데이터를 수신합니다.
- ☞ RS485 통신을 사용하여 명령과 데이터를 모터제어기로 전송하고 상위제어기에서 리턴값을 받는 경우에, 리턴값의 첫번째 바이트는 Sm 명령어로 설정된 host제어기의 주소값이며, 그 이후에 리턴되는 값은 RS232C 통신을 사용할때 리턴되는 값과 같습니다.
- ☞ RS485 통신을 사용하는 경우에, 2013년 10월 18일 이전 S/W 버전에서는 바이트간 시간간격에 대한 제한조건을 지켜야하였으나 (다음페이지의 내용 참조), 2013년 10월 18일 S/W 버전부터는 바이트간 시간간격에 대한 제한조건을 지키지 않아도 되도록 upgrade 되었으며, 명령 및 데이터를 주거나 받을때 항상 checksum을 붙이도록 하는 option이 추가되었고, ROBOTIZ 형식의 통신 프로토콜이 추가되었습니다.
- ☞ CAN 통신을 사용하여 명령과 데이터를 모터제어기로 보내려고 하면, 앞서 기술된 명령 및 데이터를 8바이트 message packet 단위로 끊어서 순차적으로 보내되 같은 주소값을 사용하면 됩니다. 한개의 명령 및 데이터를 8바이트 단위로 잘라서 보낼때 마지막으로 보내는 데이터수가 8개보다 작은 경우는 남은 개수만큼만 보내어야하며, 마지막 message packet에 그 개수를 설정하여야합니다. 주소값은 0 ~ 125 까지 허용되는데, 0 ~ 125 범위의 주소값에 128을 더한 값을 주소값으로 전송하는 경우에는 제어기로부터 리턴값이 출력되지 않습니다. 제어기로부터 리턴값이 출력되지 않게하는 방법으로는 명령어 제일마지막에 삽입하는 terminator인 ';' 대신에 '!'를 사용하는 방법도 있습니다. 또 주소값이 255인 경우에는 모든 모터 제어기가 명령 및 데이터를 수신합니다.
- ☞ CAN 통신을 사용하여 명령과 데이터를 모터제어기로 전송하고 리턴값을 받는 경우에, Sm 명령어로 설정된 host제어기의 주소값을 10로 하는 message packet을 발생시킵니다. 이때 8바이트를 넘는 경우, 8바이트단위로 잘라서 순차적으로 message를 발생시킵니다. 마지막 packet은 잔여 바이트만 message packet에 포함됩니다. 리턴되는 값은 RS232C 통신을 사용할때 리턴되는 값과 같습니다.

broadcasting 또는 selected broadcasting 통신

☞ broadcasting 기능

RS485 통신 또는 CAN 통신에서 host에서 주소값으로 255(0xFF)를 사용하여 데이터를 전송하면 모든 모터제어기에서 같은 데이터를 수신함.

☞ selected broadcasting 기능

RS485 통신 또는 CAN 통신에서 host에서 주소값으로 254(0xFE)를 사용하여 데이터를 전송하면, 사전에 EcS; 명령어를 사용하여 수신설정이 된 모든 제어기에서 같은 데이터를 수신함. EcR; 명령어를 사용하면 수신설정이 해제됨.

이 기능은 2012년 6월 3일 S/W 버전부터 사용할 수 있음.

16비트 또는 20비트 CAN 통신주소 설정과 SA 명령어의 사용

- SA 명령어로 모터제어기의 주소를 설정하는데, SX 명령어 첫번째파라미터의 bit0 = 0 일때 (CAN 통신이 선택되었을 때)
2012년 10월 16일 S/W 버전부터 SX 명령어 두번째파라미터의 bit15/14 값에 따라서 CAN 통신의 주소사용방법이 달라짐.

SX 명령어 두번째파라미터의

- bit15/14 = 11 일때 11-bit 주소모드로서 주소로 0-255만 사용
- = 10 일때 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용
- = 01 일때 29-bit 주소모드로서 주소로 0-65535만 사용 (CAN2.0B), 2012년 10월 16일부터 적용
- = 00 일때 11-bit 주소모드로서 주소로 0-255만 사용

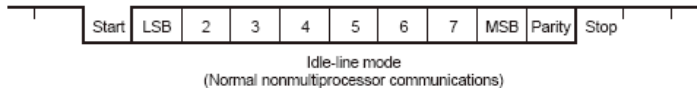
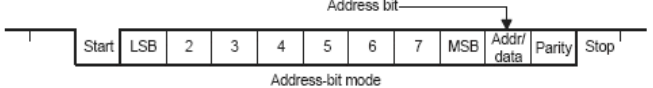
- SA 명령으로 주소값을 설정할 때 주소값의 제한사항은
2012년 10월 18일 이전 S/W 버전인 경우 또는 SX 명령어 첫번째파라미터의 bit0 = 1 인 경우 (RS485 통신이 선택되었을 때)
또는 2012년 10월 18일 S/W 버전부터로서 SX 명령어 두번째파라미터의 bit15/14 = 11 또는 00 일때는
주소값의 상위 바이트는 반드시 하위바이트의 inverse이어야만 baord 주소의 설정이 유효함.

- SA 명령으로 주소값을 설정할 때 값의 범위는
2012년 10월 18일 이전 S/W 버전인 경우 또는 SX 명령어 첫번째파라미터의 bit0 = 1 인 경우 (RS485 통신이 선택되었을 때)
: 주소값 [0000 - 00FF] (807F) 단 상위바이트는 하위바이트의 logical inverse 이어야함.
2012년 10월 18일 S/W 버전부터로서 SX 명령어 첫번째파라미터의 bit0 = 0 인 경우
: 주소값 [0000 - 00FF] (807F) - SX 명령어 두번째파라미터의 bit15/14 = 11 일때
: 주소값 [0000 - 07FF] (007F) - SX 명령어 두번째파라미터의 bit15/14 = 10 일때
: 주소값 [0000 - FFFF] (007F) - SX 명령어 두번째파라미터의 bit15/14 = 01 일때
: 주소값 [0000 - 00FF] (807F) - SX 명령어 두번째파라미터의 bit15/14 = 00 일때

- SA 명령으로 설정하는 주소값으로서 254(0xFE) 및 255(0xFF)는 bradcasting용으로 사용하는 주소이므로 126(0x8E), 127(0x8F), 254(0xFE) 및 255(0xFF)는 제어기의 주소로 사용하면 안됨.

RS485 통신을 이용한 다중 마이컴 통신의 주의사항 (2013년 10월 18일 이전의 S/W 버전에서만 해당함)

☞ TI사의 DSP제어기는 비동기통신포트(SCI)를 이용한 다중마이컴 통신을 위하여 다음과 같은 2가지 방식을 제공하고 있습니다.

	Idle-Line Multiprocessor 모드	Address-Bit Multiprocessor 모드
1-byte format	 <p>Idle-line mode (Normal nonmultiprocessor communications)</p>	 <p>Address-bit mode</p>
	RS232 통신과 같이 start-bit/stop-bit/parity-bit가 붙은 8-bit 형식	start-bit/stop-bit/parity-bit가 붙은 8-bit 형식의 통상적인 RS232 통신과 달리 address-bit가 추가되어 있음.
1-byte 값이 주소인지 데이터인지를 구별하는 방법	이전 바이트와 현재 수신된 바이트 사이의 시간간격이 10-bit 시간 이내이면 데이터로 판단하고, 10-bit 시간 이상이면 주소로 판단함.	현재 수신된 바이트의 9th bit가 1 이냐 0이냐에 따라서 주소인지, 데이터인지를 판단.
PC등 기존 기기들과의 호환성	바이트와 바이트사이의 시간간격을 지킬수 있다면 호환성 있음.	호환성 없음.
로보큐브테크 채택방식	0	X

☞ 로보큐브테크는 기존 기기들과의 호환성 유지를 위하여 TI사의 DSP제어기의 Idle-Line Multiprocessor 모드를 채택하고 있습니다. 그러나 다중 제어기간의 통신에는 다음과 같은 주의가 필요합니다.

☞ 로보큐브테크 제어기 사용자의 RS485 통신시의 준수사항. (2013년 10월 18일 이전의 S/W 버전에서만 해당함)

- : 주소가 같은 데이터 블록안에 있는 데이터를 전송할 때는 바이트와 바이트 사이의 시간 간격이 10-bit 전송시간 이내가 되도록 하여야 합니다.
- : 이를 준수하려면 주소가 같은 데이터 블록 안에 있는 데이터는 한꺼번에 준비하여야하며, 데이터블록안의 첫바이트 전송이 개시되면 나머지 데이터는 interrupt routine을 이용하여 background에서 전송하도록 하는 것이 필요합니다. (interrupt가 아니고 프로그램이 일일이 관여하여 데이터를 보내게 되면 통신이 제대로 안되거나 cpu time이 통신에 100% 점유되는 상태가 될수 있습니다.)
- : 만약 바이트와 바이트 사이의 시간 간격이 클수 밖에 없는 경우라면 통신속도를 떨어뜨려 사용하는 것도 한가지 방법입니다. (예를 들면 1.25 Mbps의 경우 10-bit의 시간은 8uS 인데 비하여 9600 bps 인 경우는 10-bit의 시간이 1.04ms 로 매우 커집니다.)
- : 같은 이유로 주소가 다른 데이터 블록 사이에는 데이터 블록과 데이터 블록 사이의 시간 간격이 10-bit 전송시간 이상이 되도록 하여야 합니다.

RS485 통신에서 추가된 checksum option 기능 (2013년 10월 18일 S/W 버전부터)

☞ 2013년 10월 18일 S/W 버전부터 적용됩니다.

☞ RS485 통신을 할때만 적용됩니다.

☞ SX 명령어로 설정되는 두번째 파라미터의 최상위 비트가 0인 경우에 RS485 통신에서 checksum option이 선택됩니다.

☞ RS485 통신에서 checksum option이 선택된 경우에는 상위제어기가 모터제어기로 명령을 전송할때, 주소를 포함한 명령 및 데이터값을 모두 더한 다음 inverse한 값을 덧붙여서 전송하여야 합니다. 리턴값 역시 checksum 값이 마지막에 덧붙여 집니다. 따라서 checksum option을 선택하지 않은 경우에 비하여 한 바이트가 많아지며, 마지막 checksum 한 바이트를 제외한 나머지의 값들은 checksum option 과 무관합니다.

☞ checksum option이 선택하지 않은 경우의 사용례

0x01(모터제어기주소),0x53(S),0x41(A),0x3F(?),0x3B(;) 를 모터제어기로 전송하는 경우 (01번 모터제어기로 SA?; 명령을 보냄), 모터제어기는 0x00(상위제어기주소),0x53(S),0x41(A),0x46(F),0x45(E),0x30(O),0x31(1),0x3B(;) 를 상위제어기로 리턴합니다.

☞ checksum option이 선택한 경우의 사용례

0x01(모터제어기주소),0x53(S),0x41(A),0x3F(?),0x3B(;),0xF0(checksum) 를 모터제어기로 전송하는 경우 (01번 모터제어기로 SA?; 명령을 보냄), 모터제어기는 0x00(상위제어기주소),0x53(S),0x41(A),0x46(F),0x45(E),0x30(O),0x31(1),0x3B(;),44(checksum) 를 상위제어기로 리턴합니다.

RS485 통신에서 추가된 ROBOTIZ 프로토콜 (2013년 10월 18일 S/W 버전부터)

- ☞ 2013년 10월 18일 S/W 버전부터 적용됩니다.
- ☞ RS485 통신에서 ROBOTIZ 형식의 통신프로토콜을 일부 지원합니다.

☞ ROBOCUBE 모터제어기의 명령어를 ROBOTIZ 형식의 통신프로토콜을 사용하여 전송하는 사용례

FF(),FF(),01(모터제어기주소),06(바이트수),99(0P),53(S),41(A),3F(?),3B(:),51(checksum) 를 모터제어기로 전송하는 경우, 모터제어기는 FF(),FF(),01(모터제어기주소),09(바이트수),00(ER),53(S),41(A),46(F),45(E),30(O),31(1),3B(:),3A(checksum) 를 상위제어기로 리턴합니다. checksum은 첫 두바이트 FF와 FF를 제외한 나머지 바이트의 총합의 complement의 최하위 바이트값임.

☞ ROBOTIZ 형식의 통신프로토콜 중에서 sync write 기능을 사용하여 p 명령에 해당하는 위치명령을 전송하는 사용례

FF(),FF(),FE(sync write),20(총바이트수),83(0P),1E(Pa 유사명령어),06(모터제어기 한개당 데이터 바이트수),
 01(모터제어기주소),00(low_byte),12(middle_byte),34(high_byte),00(low_byte),54(middle_byte),78(high_byte),
 (모터제어기 1의 위치명령값 모터1 0x341200, 모터2 0x785400)
 02(모터제어기주소),00(low_byte),72(middle_byte),A4(high_byte),00(low_byte),14(middle_byte),38(high_byte),
 (모터제어기 2의 위치명령값 모터1 0xA47200, 모터2 0x381400)
 04(모터제어기주소),00(low_byte),65(middle_byte),62(high_byte),00(low_byte),18(middle_byte),49(high_byte),
 (모터제어기 4의 위치명령값 모터1 0x626500, 모터2 0x491800)
 07(모터제어기주소),00(low_byte),47(middle_byte),30(high_byte),00(low_byte),51(middle_byte),B5(high_byte),
 (모터제어기 7의 위치명령값 모터1 0x304700, 모터2 0xB5100)

1D(checksum, 첫 두바이트 FF와 FF를 제외한 나머지 바이트의 총합의 complement의 최하위 바이트값) 를 모터제어기로 전송하는 경우, 해당제어기가 위치명령값을 수신하며, 리턴값은 없습니다. **총 전송바이트수는 256바이트를 넘으면 안됩니다.**

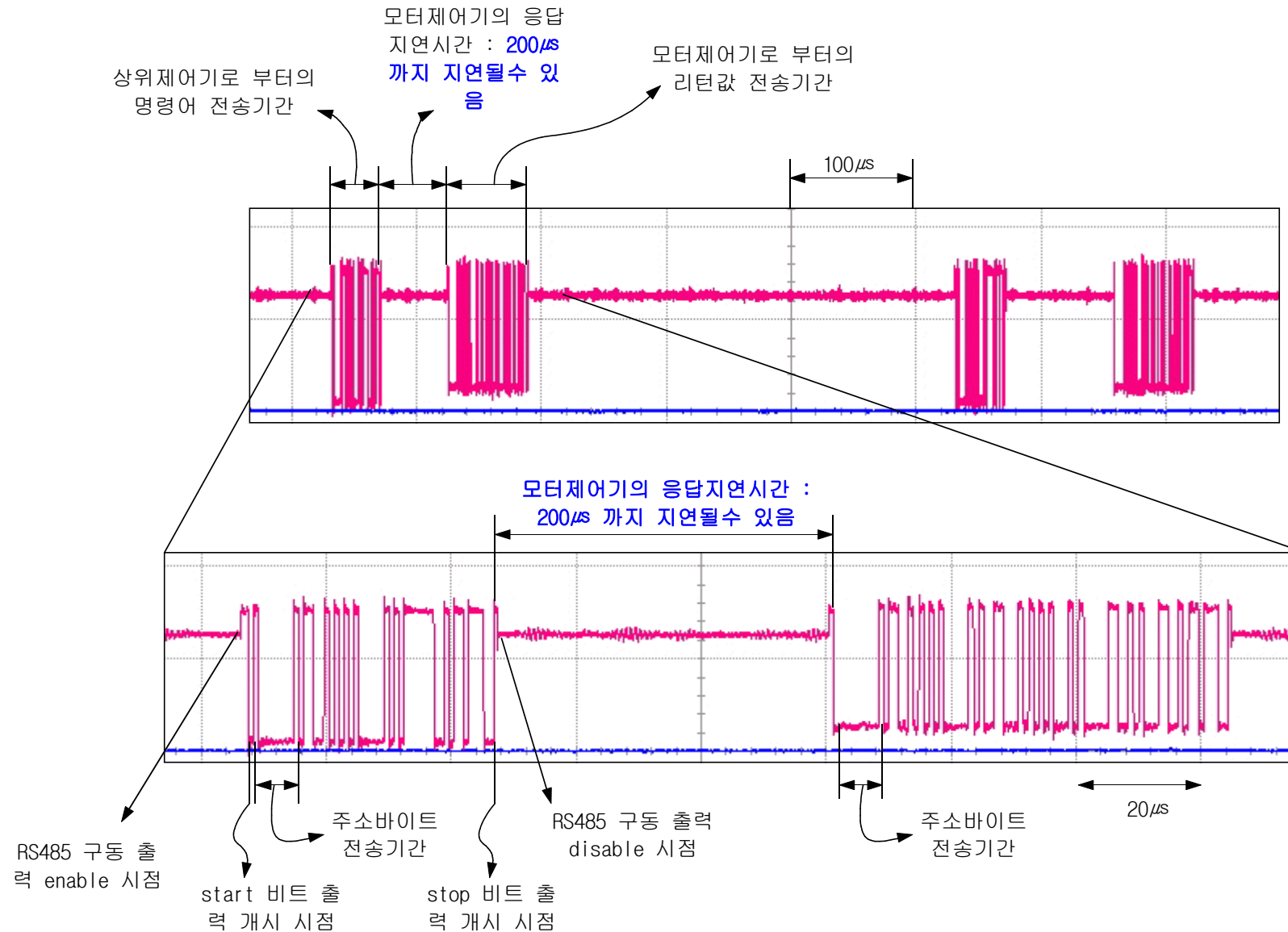
FF,FF,FE,12,83,1E,06,02,50,72,4C,70,C0,4C,01,50,70,4C,50,C0,4C,53 (모터제어기 #2에 0x4C7250,0x4CC070 위치명령과 모터제어기 #1에 0x4C7050,0x4CC050 위치명령을 전송하되, Pa 명령과 같은 작동을 하며, 리턴값은 없음.)

FF,FF,FE,12,83,9E,06,02,50,72,4C,70,C0,4C,01,50,70,4C,50,C0,4C,D3 (모터제어기 #2에 0x4C7250,0x4CC070 위치명령과 모터제어기 #1에 0x4C7050,0x4CC050 위치명령을 전송하되, p 명령과 같은 작동을 하며, 리턴값은 없음.)

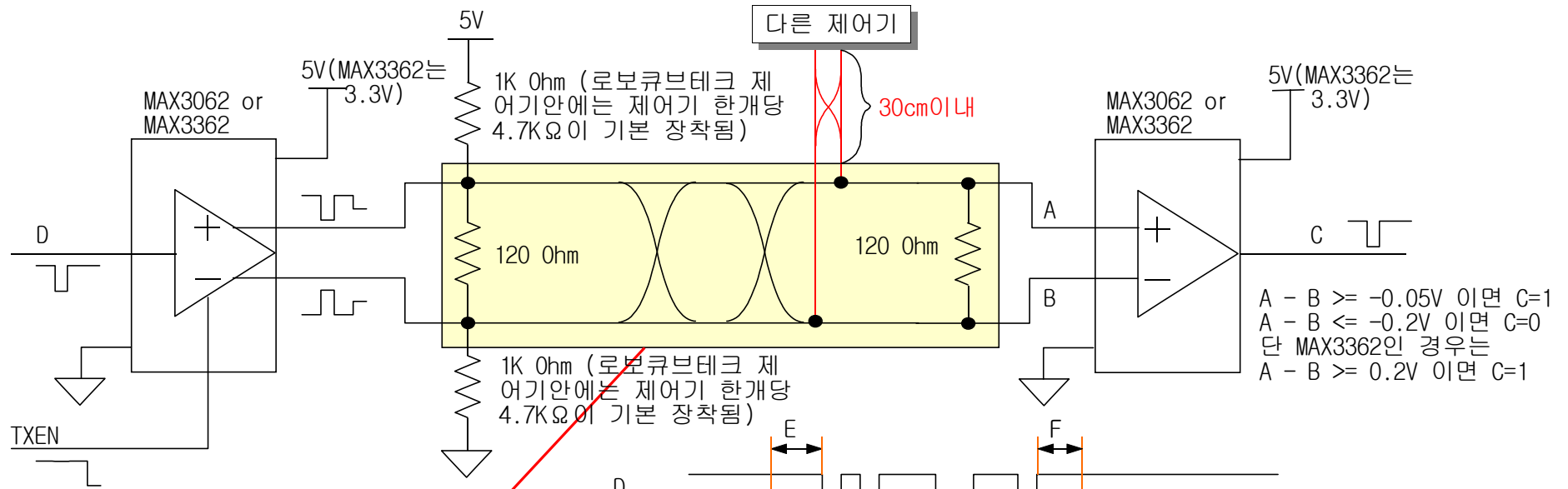
모터제어기 #2의
위치명령

모터제어기 #1의
위치명령

RS485 통신을 이용한 다중 마이크 통신의 주의사항 (1.25Mbps RS485 출력 +단자 파형)



RS485 통신을 이용한 다중 마이컴 통신의 주의사항 (통신 배선상)



제어기 외부회로 :

- (1) 통신선은 반드시 **twist pair**를 사용해야하며
- (2) 상위제어기를 포함하여 모든 모터제어기들 사이의 통신 배선은 **star** 형식으로 연결하면 안되고 반드시 **daisy-chain** 형식이어야하며
- (3) 반드시 통신배선 양쪽 끝에 120 Ω 종단저항 2개를 설치해야하며
- (4) 통신배선 중간에 연결된 제어기는 연결선이 30cm 이내 이어야하며
- (5) 통신선에 연결된 모든 모터제어기 및 상위제어기의 통신용 ground는 연결되어 있어야 하며
- (6) 통신선에 연결된 모든 모터제어기 및 상위제어기의 ground간 허용 전위차는 -7V에서 +12V 입니다.

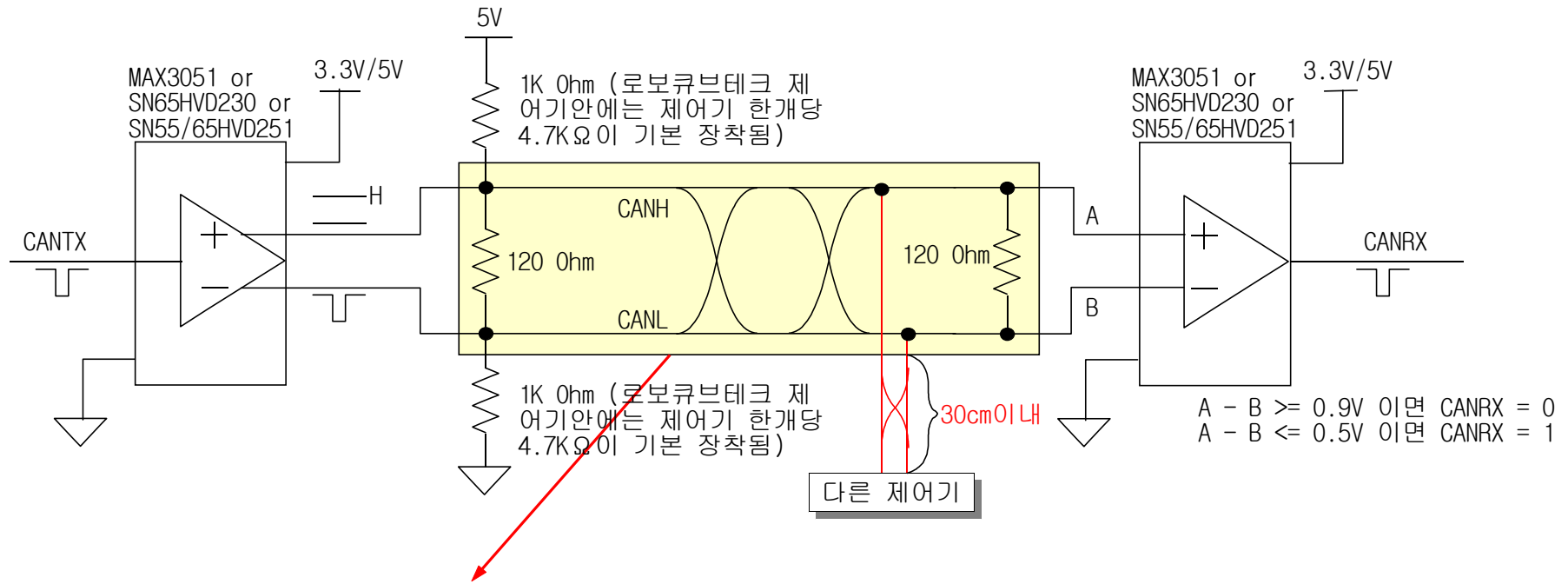
※ MAX3362를 사용한 H/W 버전에서는 pull-up/pull-down 저항값의 합성저항 크기가 1K 정도는 되어야 통신에 오류가 생기지 않으며, MAX3062를 사용한 H/W 버전에서는 합성저항크기가 좀더 커도 오류가 없음.

노이즈등의 이유로 C=low가 된 경우에, E시간이 짧으면 low->low로 또는 low->짧은 high->low로 변하는 C 신호로 인하여 수신 오류가 발생하게 됩니다.

노이즈에 의하여 C=low가 잠깐될수 있으며, 이때 MAX3062의 경우는 -0.05V 이상이면 high로 인지하므로 C=low가 지속되지 않으나, MAX3362의 경우는 0.2V 이상이어야 high로 인지되므로 1K pull-up/pull-down이 없다면 C=low가 지속되어 break detect error가 발생할수 있습니다.

제어기 명령어 LIST

CAN 통신을 이용한 다중 마이크로 통신의 주의사항 (통신 배선상)



제어기 외부회로 :

- (1) 통신선은 반드시 **twist pair**를 사용해야하며
- (2) 상위제어기를 포함하여 모든 모터제어기들 사이의 통신배선은 **star** 형식으로 연결하면 안되고 반드시 **daisy-chain**형식이어야하며
- (3) 반드시 통신배선 양쪽 끝에 120Ω 종단저항 2개를 설치해야하며
- (4) 통신배선 중간에 연결된 제어기는 연결선이 30cm 이내이어야하며
- (5) 통신선에 연결된 모든 모터제어기 및 상위제어기의 통신용 ground는 연결되어 있어야 하며
- (6) 통신선에 연결된 모든 모터제어기 및 상위제어기의 ground간 허용 전위차는 MAX3051인 경우 -7V에서 +12V이고, SN65HVD230의 경우 -7V에서 +16V이고, SN55/65HVD251의 경우는 -36V에서 +36V입니다. 순간 허용 전위차는 SN65HVD230/SN65HVD251/SN55/65HVD251이 각각 ±25V/±200V/±200V 입니다.
- (7) 통신선에 연결된 모든 모터제어기 및 상위제어기의 ground간 전위차로서 통신가능한 전위차 범위는 MAX3051와 SN55/65HVD251 인 경우 -7V에서 +12V이고, SN65HVD230 경우는 -2V에서 +7V 범위 입니다.

CANTX=0일 때 $A - B \geq 0.9V$, CANRX = 0

CANTX=1일 때

1K pull-up/pull-down이 없으면 $A-B=0V \leq 0.5V \Rightarrow$ CANRX=1

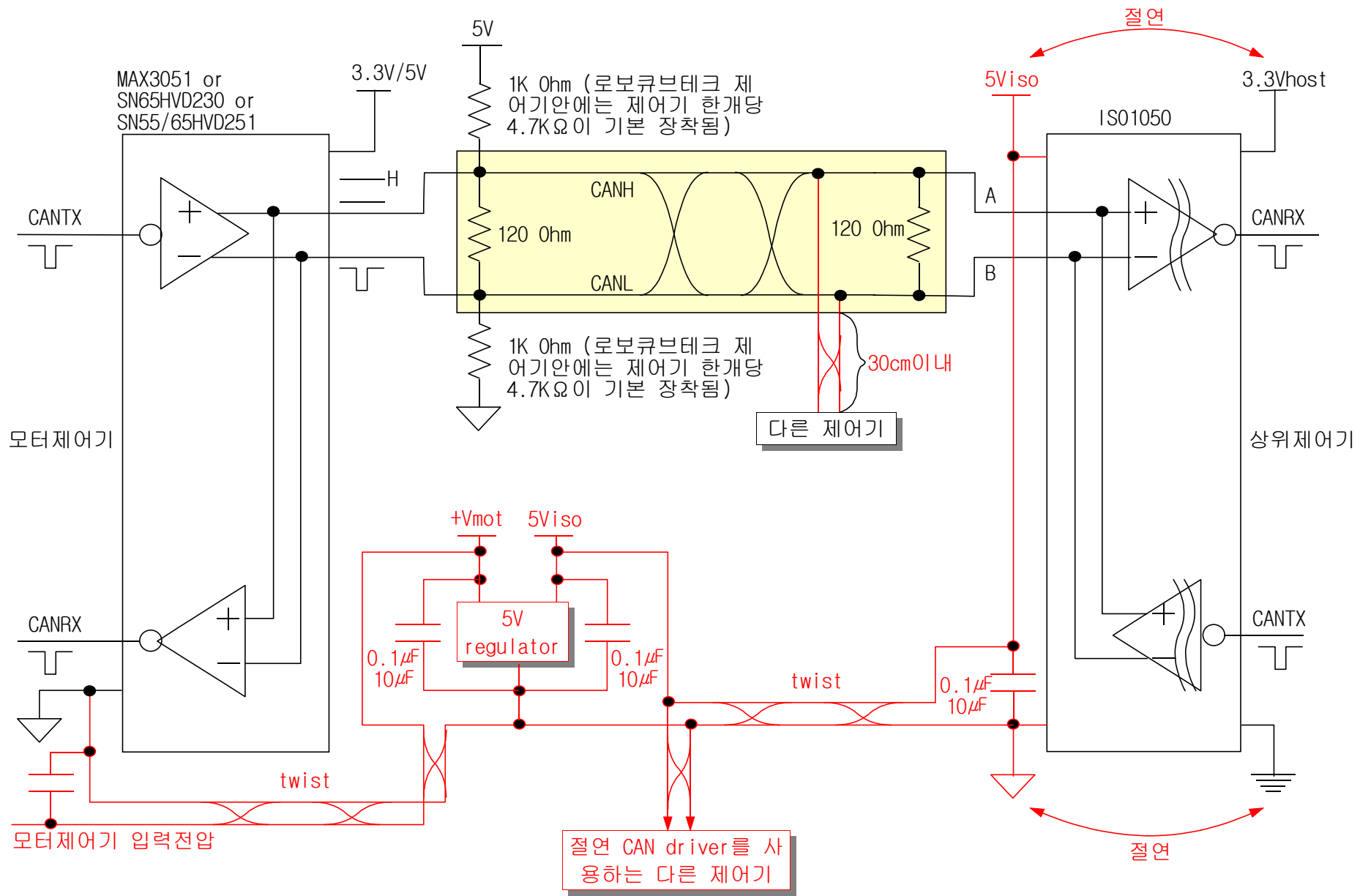
1K pull-up/pull-down이 있으면 $A-B=0.096V \leq 0.5V \Rightarrow$ CANRX=1

(1K pull-up/pull-down 이 있는 경우 noise margin이 0.096V 감소함)

※ CANH, CANL 출력단에 설치된 4.7KΩ pull-up/pull-down 저항은 RS485 통신을 할때는 통신상의 단말기가 모두 off되어 있는 경우 수신단의 H상태를 유지하기 위하여 반드시 필요하나, CAN 통신에서는 통신선상의 단말기가 많을 경우에 잘못된 값을 수신할 가능성이 존재하므로 이때는 좀더 큰 저항값을 사용하여야 함.

※ CANH, CANL에 설치된 120Ω저항이 없을때는 CAN baudrate 설정의 즉시 반영은 불가하며, 설정시 마이크로 리셋을 유발할 수 있음.

CAN 통신을 이용한 다중 마이크 컴퓨터 통신의 주의사항 (isolator를 사용할때)



제어기 명칭어 LIST

전원 및 통신배선 연결시 주의사항

※ 전원용 배선 규격

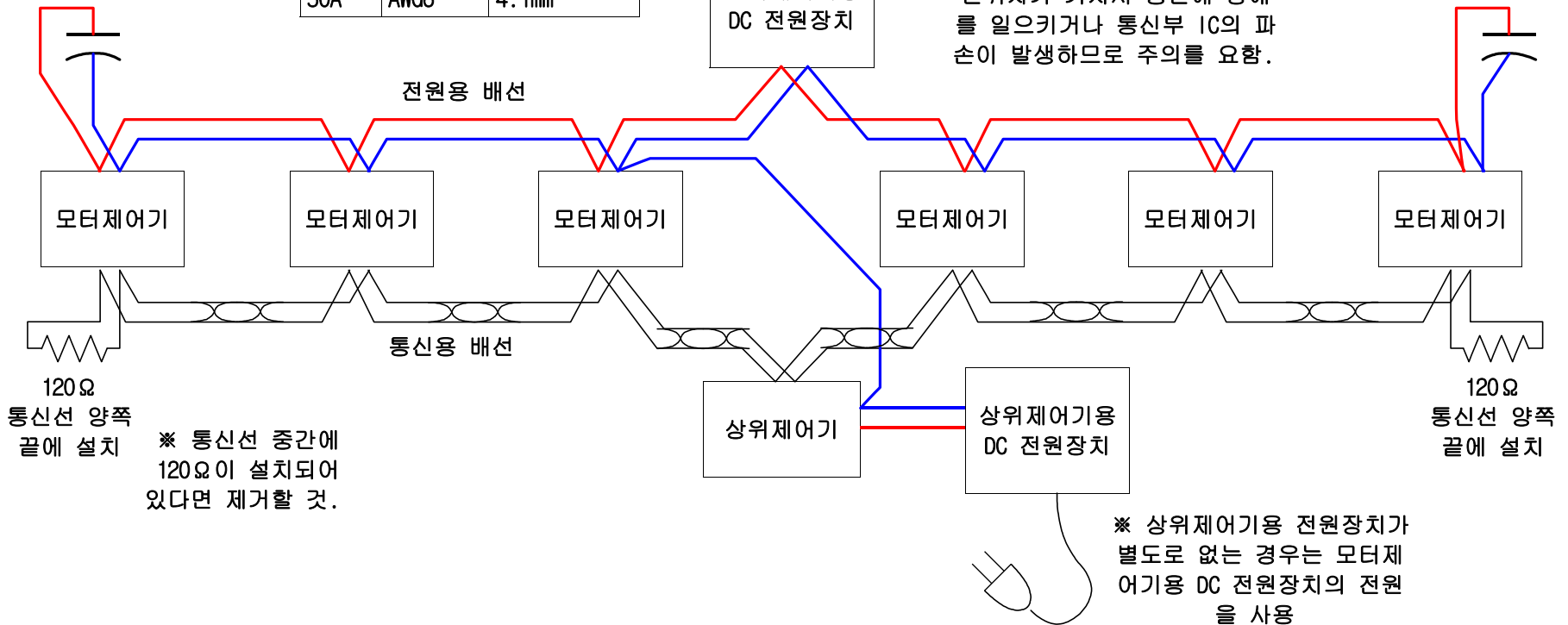
전류	배선규격	도체부 직경
2A	AWG20	0.8mm
5A	AWG16	1.3mm
10A	AWG13	1.8mm
20A	AWG10	2.6mm
50A	AWG6	4.1mm

※ 모터제어기의 개수가 많거나 전원배선의 길이가 길 때 1000uF이상의 콘덴서를 끝부분에 추가

※ 전원용 배선은 모터제어기용 DC 전원장치로부터 모터제어기에 각각 직접 연결하는 것이 좋으나, 배선이 복잡하여 그림과 같이 Daisy chain 형식으로 연결하는 경우는 주의를 요함.

※ 모터제어기의 개수가 많거나 전원배선의 길이가 길 때 1000uF이상의 콘덴서를 끝부분에 추가

※ 전원용 배선이 너무 가늘거나 너무 길면 모터제어기간의 전위차가 커져서 통신에 장애를 일으키거나 통신부 IC의 파손이 발생하므로 주의를 요함.



제어기 명령어 LIST

CAN 통신모드일때 일정시간마다 제어기/센서 상태값을 상위제어기로 전송하는 기능 (2013년 11월 25일 부터)

- ☞ CAN 통신을 사용할 때, 일정주기로 모터의 상태값을 전송함. 2013년 11월 25일 S/W 버전부터 적용됨.
 SX 명령어 두번째파라미터의 bit27 = 0 일때는 모터1의 상태값이 상위제어기로 전송되며,
 SX 명령어 세번째파라미터의 bit2 = 0 일때는 모터2의 상태값이 상위제어기로 전송됨.
 두값 모두 같은 시점에서 전송이 가능함. 데이터의 내용은 SX 명령어의 설명 바로 다음에 제시되어 있음.
 전송되는 데이터의 내용은 Eu, Ev 명령어로 바꿀 수 있음.

전송되는 주소는 모터1의 경우 기본값 0xfd 이고, 모터2의 경우 기본값 0xfc 임.

- ☞ CAN 통신을 사용하여 일정주기로 모터의 상태값을 전송할 때
 주기값과 주소는 EU 명령을 사용하여 설정이 가능하며, 전송되는 데이터의 내용은 SX 명령어 다음에 부분에 있는 내용을 참조.
 전송되는 데이터의 내용은 Eu, Ev 명령어로 바꿀 수 있음.
- ☞ CAN 통신을 사용하여 일정주기로 모터의 상태값을 전송하는 기능은
 SX 명령어 첫번째 파라미터의 bit25가 1 이거나 (1ms main loop의 동기화 경우),
 국방로봇용 제어기이거나,
 CAN 통신모드가 아닌 경우는 적용되지 않음.

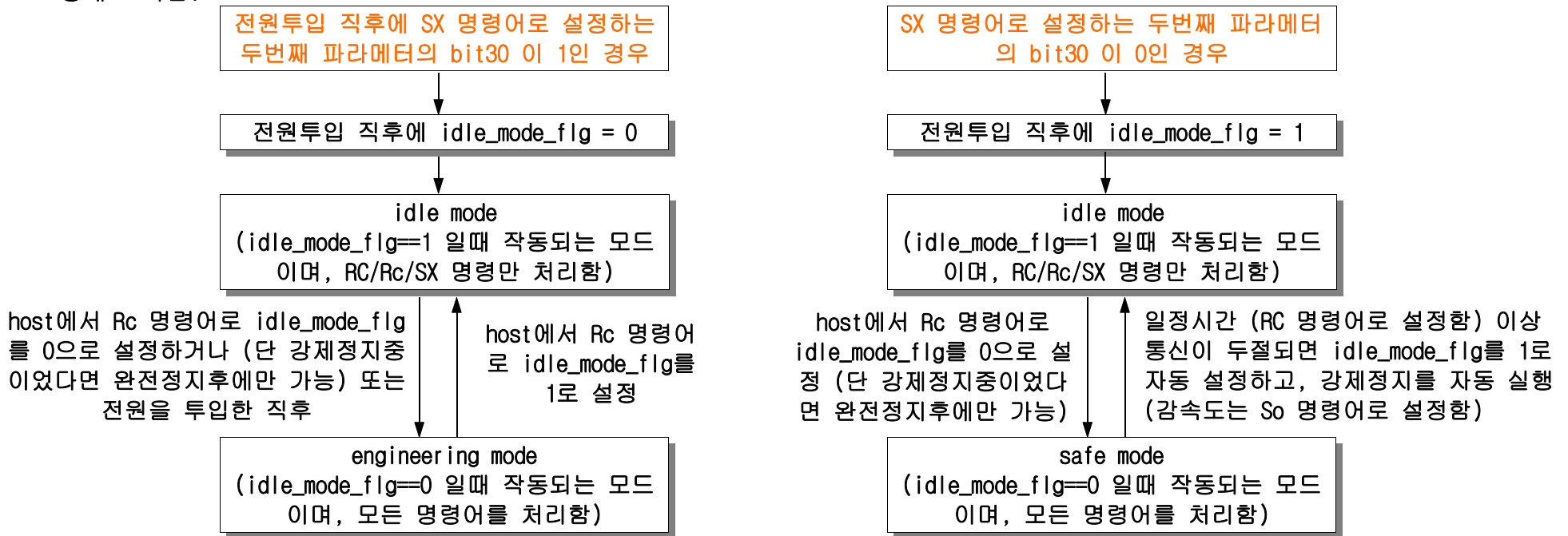
Pa/Pb/Pp/Pq/p/Qb9 명령의 명령값 및 리턴값의 데이터수를 줄이는 기능 (2013년 11월 22일 부터)

- ☞ Pa/Pb/Pp/Pq/p/Qb9 명령을 사용할 때 2채널제어기의 경우는 전송데이터량이 필요이상으로 많아지는 문제가 있어서 2013년 11월 22일 S/W 버전부터 SX 명령어로 설정하는 세번째 long word의 bit0 값이 0이면 모터2의 명령값과 리턴값을 생략함.

통신 두절시 강제 정지기능과 통신모드의 변경 (2013년 11월 05일 S/W 버전부터)

☞ 2013년 11월 05일 S/W 버전부터 적용됩니다.

- ☞ SX 명령어로 설정하는 두번째 파라미터의 bit30 이 1 이면 (2013년 10월 23일 이전의 S/W 버전 에서 사용하던 통신기능과 호환됨)
 - 전원 투입 직후 RS485/CAN/RS232C 통신에 관계없이 모든 명령어를 처리하는 상태가 됨. (engineering mode)
 - Rc 명령어로 idle mode로 진입하는 flag를 세트하면 idle mode 상태로 바뀜.
 - idle mode에서는 RC/Rc/SX 명령만 처리하며 (단 RS232C 통신인 경우는 모든 명령어를 처리함), Rc 명령어로 idle mode로 진입하는 flag를 클리어하면 engineering mode 상태로 돌아감.
- ☞ SX 명령어로 설정하는 두번째 파라미터의 bit30 이 0 이면 (safe mode 기능이 추가로 적용됨)
 - 전원 투입 직후에는 RC/Rc/SX 명령만 처리함. (idle mode) 단 RS232C 포트를 통하여 입력되는 명령어는 모두 처리함.
 - 이후에 Rc 명령어로 idle mode로 진입하는 flag를 클리어하면 safe mode 상태로 바뀜.
 - safe mode 상태에서는 RS485/CAN 통신을 지속적으로 하여야 하며, 모든 명령어를 사용할 수 있으며, 일정시간(RC 명령어로 설정함) 이상 RS485/CAN 통신이 두절되면 자동으로 강제 정지하고, idle mode로 진입하는 flag를 자동으로 세트하여 idle mode로 진입함.
 - idle mode에서는 RC/Rc/SX 명령만 처리하며, 정지상태에서 Rc 명령어로 idle mode로 진입하는 flag를 다시 클리어하여야만 safe mode 상태로 바뀜.



명령어의 분류

분류	설명	페이지	비고
통신관련 파라미터의 설정	통신관련 파라미터의 설정 - 포트 및 주소 및 특수기능1/2	30/31	
	통신관련 파라미터의 설정 - 에러 및 복구	32	
모터의 기본 파라미터 설정과 저장	모터의 기본 파라미터 설정과 저장 - 1/2	33/34	
factory default 파라미터의 설정	factory default 파라미터의 설정과 저장	35	
모터제어 파라미터의 설정	모터제어 파라미터의 설정 - 전류/속도/위치 게인1/2	36/37	
	제어 파라미터의 설정 - BLDC 모터 추가게인 설정	38	
	제어 파라미터의 설정 - 게인 스케줄링 파라미터 설정	39	
	제어 파라미터의 설정 - 기타게인 설정1/2/3	40/41/42	
	제어 파라미터의 설정 - advanced control algorithm 파라미터 설정	43	
	모터제어 파라미터의 설정 - 정지강성, 한계값, dead-time 및 펄스주기 속도계산	44	
	모터제어 파라미터의 설정 - compensator 및 filter	45	
	모터제어 파라미터의 설정 - 기타 파라미터1/2	46/47	
작동개시, 작동중지 및 비상정지, 강제정지	작동개시, 작동중지 및 비상정지 관련 명령어	48	
	강제정지 명령어	49	
모터제어 작동 실행	전류구동 명령어	50	
	속도구동 명령어	51	
	point-to-point 위치구동 명령어 - 1/2	52/53	
	연속 point-to-point 위치구동 명령어	54	
	실시간 연속 위치구동 명령어 - 1/2	55/56	
	스테핑모터 구동관련 명령어	57	
	jog 작동 명령어	58	
	homing 작동 명령어	59	
	sequence 제어 명령어	60	
	출력 강제작동 명령어	61	
입출력 조작	입력/출력 모니터 명령어 - 1/2	62/63	
초기화	초기화 명령어	64	
Fault 처리	과전압, 저전압 및 과전류 보호 명령어 - 1/2/3	65/66/67	
	과온도 보호 명령어	68	
	fault상태 모니터 명령어	69	
제어작동상태 모니터	제어작동상태 모니터 명령어	70	
	제어작동상태 변수 모니터 명령어 - 1/2	71/72	
	제어작동상태 레지스터 및 메모리 읽기/쓰기 명령어	73	
	전류측정 명령어	74	
test 구동	test 구동 명령어 - 1/2	75/76	
튜닝	위치센서튜닝 명령어	77	
	위치센서튜닝 명령어 - analog type 센서 사용 BLDC모터의 자동 파라미터 설정	78	
	위치센서튜닝 명령어 - resolver 및 삼상 analog Hall 센서	79	
text 입력/저장과 LCD 표시장치	text 입력/저장과 LCD 표시장치 관련 명령어	80	
데모작동	데모작동 명령어	81	
특정 목적의 기능	특정 목적의 기능관련 명령어 - 1/2	82/83	
F/T 센서	F/T 센서 명령어 - 1/2/3/4	84/85/86/87	
사용하지 않는 명령어	사용하지 않는 명령어	88	
레이저스캐너, 줌카메라에서 사용하는 명령어	TMS320F28016 DSP를 사용한 레이저스캐너, 줌카메라에서 사용하는 명령어	89/90/91/92/93/94	
RC 서보제어기에서 사용하는 명령어	C8051F555_noRobitiz, C8051F561, C8051F555_Robitiz RC 서보제어기에서 사용하는 명령어	95/96/97/98/99/100/101	
전류앰프에서 사용하는 명령어	고전압 전류앰프에서 사용하는 명령어	102	

명령어의 분류 (통신관련 파라미터의 설정 - 포트 및 주소 및 특수기능1)

적용 CPU	명령어	return value	operation
+#^% +#^%	;SB ; ;SB?;	SB ; (SB?; 인 경우에만 echo 있음)	RS232C 직렬통신속도값을 설정함.
+^% +^%	;Sb ; ;Sb?;	Sb ; (Sb?; 인 경우에만 echo 있음)	RS485 직렬통신속도값을 설정함.
# #	;Sb , 2; ;Sb?;	Sb , 2; (Sb?; 인 경우에만 echo 있음)	RS485 와 RS232C extra 직렬통신속도값을 설정함.
+#^ +#^	;Zbddd1; ;Zb?;	Zbddd1;	CAN 통신속도를 설정함.
+#^% +#^%	;SXHLHL; ;SX?;	SXHLHL; 또는 SXHLHLHLHL; SXHLHLHLHL,HLHLHLHL; (2012년 5월 28일 S/W 버전부터) SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터)	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+#^ +#^	;SXHLHLHLHL; ;SX?;	SXHLHLHLHL; SXHLHLHLHL,HLHLHLHL; (2012년 5월 28일 S/W 버전부터) SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터)	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+#^ +#^	;SXHLHLHLHL,HLHLHLHL; ;SX?;	SXHLHLHLHL,HLHLHLHL; SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터)	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+#^ +#^	;SXHLHLHLHL,HLHLHLHL,HLHLHLHL; ;SX?;	SXHLHLHLHL,HLHLHLHL,HLHLHLHL;	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함. 2013년 11월 22일 S/W 버전부터 사용가능한 명령임.

명령어의 분류 (통신관련 파라미터의 설정 - 포트, 주소 및 특수기능2)

적용 CPU	명령어	return value	operation
+#^%	;SAHLHL;	SAHLHL;	모터제어기의 주소를 설정함.
+#^%	;SA?;		
+#^%	;SmHLHL;	SmHLHL;	명령어에 대한 return값을 보낼 때 사용하는 host의 주소를 설정함.
+#^%	;Sm?;		
+#^	;EUHLHL,HLHL,HLHL;	EUHLHL,HLHL,HLHL;	CAN 통신을 사용할 때, 일정주기로 상위제어기에 제어기 상태값을 전송하는 기능의 파라미터를 설정함.
+#^	;EU?;		
+#^	;EuHLHL,HLHL,HLHL,HLHL;	EuHLHL,HLHL,HLHL,HLHL;	CAN 통신을 사용하여 일정주기로 상위제어기에 모터1의 상태값을 전송할 때, 전송할 모터1의 상태값 주소를 설정함.
+#^	;Eu?;		
+#^	;EvHLHL,HLHL,HLHL,HLHL;	EvHLHL,HLHL,HLHL,HLHL;	CAN 통신을 사용하여 일정주기로 상위제어기에 모터2의 상태값을 전송할 때, 전송할 모터2의 상태값 주소를 설정함.
+#^	;Ev?;		

명령어의 분류 (통신관련 파라미터의 설정 - 에러 및 복구)

적용 CPU	명령어	return value	operation
+#^%	;QE;	QEHLHL,HLHL,HL,HLHL,HLHL,HLHL;	통신에러횟수를 읽어서 host로 전송함.
+#^	;QEE;	QEEHLHL,HLHL,HLHL,HLHL,HLHL;	통신에러횟수를 읽어서 host로 전송함.
+#^%	;EC?;	ECHLHL,HLHL,HLHL;	RS232, RS485, CAN 통신과정에서 어떤 에러가 발생하였는지를 읽어서 host로 전송함.
+#^%	;ECR;	ECR;	RS232, RS485, CAN 통신에러 플래그를 없애고 에러를 복구함.
+#^	;EcS;	EcS;	RS485 또는 CAN을 통하여 모터제어기 주소 254(0xFE)로 전송되어오는 broadcasting 데이터를 수신하도록 설정함.
+#^	;EcR;	EcR;	RS485 또는 CAN을 통하여 모터제어기 주소 254(0xFE)로 전송되어오는 broadcasting 데이터를 수신하지 못하도록 설정함.
+#^ +#^	;RCddd1,ddd2; ;RC?;	RCddd1,ddd2,ddd3,ddd4;	CAN 이나 RS485 통신이 일정시간 두절되었을 때를 감지하는 time threshold 파라미터를 설정함. 2013년 10월 23일 S/W 버전부터 사용가능한 명령임.
+#^ +#^	;Rcddd1; ;Rc?;	Rcddd1;	CAN 이나 RS485 통신에서 idle mode로 진입하거나 또는 정상통신모드로 진입하는 것을 설정함. 2013년 11월 05일 S/W 버전부터 사용가능한 명령임.

명령어의 분류 (모터의 기본 파라미터 설정과 저장 - 1)

적용 CPU	명령어	return value	operation
+#^% +#^%	;STHLHL; ;ST?;	STHLHL;	모터의 기본파라미터를 쉽게 설정하기 위하여 미리 준비되어있는 motor type을 선정함.
+#^% +#^%	(저가형 1채널 버전이 아닌 경우) ;SEA55A,HLHL,HLHL,HLHL,HLHL; ;SE?;	SEHLHL,HLHL,HLHL,HLHL; (2012년 1월 24일 이전 S/W 버전) SEHLHL,HLHL,HLHL,HLHL,HLHL; (2012년 1월 24일 S/W 버전부터) SEHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (2013년 5월 19일 S/W 버전부터)	모터1회전당 펄스수와 POLE PAIR 갯수를 설정함. (저가형 1채널 버전이 아닌 경우)
+#^ +#^	(저가형 1채널 버전인 경우) ;SEA55A,HLHL,HLHLHLHL; ;SE?;	SEHLHL,HLHLHLHL,HLHL,HLHL;	모터1회전당 펄스수와 POLE PAIR 갯수를 설정함. (저가형 1채널 버전인 경우)
+#^ +#^ +#^	;XEddd1,ddd2; or ;XE 1111,1112; ;XE?;	XEddd1,ddd2; or XE 1111,1112;	스테핑모터 또는 절대각엔코더를 사용하는 경우 엔코더의 4채배된 펄스수를 설정함.
+#^% +#^%	;SGA55A,ddd1,ddd2,ddd3,ddd4; ;SG?;	SGddd1,ddd2,ddd3,ddd4;	감속비와 최대회전속도를 설정함.
+#^% +#^%	;SXHLHL; ;SX?;	SXHLHL; 또는 SXHLHLHLHL; SXHLHLHLHL,HLHLHLHL; (2012년 5월 28일 S/W 버전부터) SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터)	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.

명령어의 분류 (모터의 기본 파라미터 설정과 저장 - 2)

적용 CPU	명령어	return value	operation
+#^ +#^	;SXHLHLHLHL; ;SX?;	SXHLHLHLHL; SXHLHLHLHL,HLHLHLHL; (2012년 5월 28일 S/W 버전부터) SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터)	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+#^ +#^	;SXHLHLHLHL,HLHLHLHL; ;SX?;	SXHLHLHLHL,HLHLHLHL; SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터)	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+#^ +#^	;SXHLHLHLHL,HLHLHLHL, HLHLHLHL; ;SX?;	SXHLHLHLHL,HLHLHLHL,HLHLHLHL;	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함. 2013년 11월 22일 S/W 버전부터 사용가능한 명령임.
+#^ +#^	(F/T센서가 아닌 경우 로서 2013년 11월 22 일 S/W 버전부터) ;Msddd1,ddd2; ;Ms?;	Msddd1,ddd2;	resolver를 위치센서로 사용하는 경우, U/V/W상 전류가 한 사이클 변 하면서 움직이는 동안 발생하는 resolver 센서 신호의 사이클 수를 설정함. 2013년 11월 22일 S/W 버전부터 사용가능한 명령임.
+#^%	;ERHLHL,HLHL;	ERHLHL,HLHL;	파라미터 값을 EEPROM으로부터 읽어내어 DSP SARAM에 저장함.
+#^%	;ESHLHL,HLHL;	ESHLHL,HLHL;	DSP SRAM에 있는 파라미터 값을 page단위(8 word 또는 16 byte)로 EEPROM에 저장함.
+#^%	;EDA55A;	EDA55A;	DSP SRAM에 있는 파라미터 값을 default 값으로 바꿈. 이때 ST 명 령어로 선정된 motor type에 따라서 파라미터의 기본값이 설정됨.
+#^%	;EsA55A;	EsA55A;	DSP SRAM에 있는 작동 파라미터 값을 한꺼번에 EEPROM에 저장함.

명령어의 분류 (factory default 파라미터의 설정과 저장)

적용 CPU	명령어	return value	operation
+#^	;Z0iiii1; ~ ;ZFiiii1; (단 ;Z3HLHL; ;Z4HLHL; ;Z5HLHL; ;Z6HLHL; ;Z7HLHL; ;Z8HLHL; ;Z9HLHL; ;Z AHLHL; ;ZBHLHL; ;ZCHHL; 는 예외) (+#^) (2012년 12월 19일 이전 버전인 경우)	Z0(~F)HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,H LHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLH L; HLHL,HLHL,.....,HLHL : 16개의 FACTORY SETTING parameter 값	0 ~ F 까지의 숫자로 지정된 위치의 FACTORY SETTING parameter RAM에 값 을 저장함.
+#^	;ZOA55A,iiii1; ~ ;ZFA55A,iiii1; (단 ;Z3A55A,HLHL; ;Z4A55A,HLHL; ;Z5A55A,HLHL; ;Z6A55A,HLHL; ;Z7A55A,HLHL; ;Z8A55A,HLHL; ;Z9A55A,HLHL; ;ZAA55A,HLHL; ;ZBA55A,HLHL; ;ZCA55A,HLHL; 는 예외) (2012 년 12월 19일 S/W 버전부터)		
+#^	;Zr;	ZrHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL, HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	16개의 FACTORY SETTING parameter 값 을 읽어서 보여줌 (RAM 영역의 값).
+#^	;ZsA55A;	ZsA55A;	메모리상의 FACTORY SETTING parameter 를 EEPROM으로 저장함.
+#^	;Zp?;	Zpdddd1,dddd2; (2015년 3월 23일 이전 버전 인 경우) Zpdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd 7,dddd8; (2015년 3월 23일 버전부터)	PWM 주파수와 모터1/2의 전류제어 주 파수, 속도 및 위치제어 주파수를 읽 어서 호스트로 전송함.

명령어의 분류 (모터제어 파라미터의 설정 - 전류/속도/위치 계인1)

적용 CPU	명령어	return value	operation
+#^%	:SQ 또는 ;SqHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL;	SQ 또는 SqHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL;	모터1의 파라미터 리밋과 위치/속도/전류 제어파라미터를 설정함. (WQ 명령어로 속도 및 전류제어 파라미터를 설정하는 경우는 제외)
+#^%	:SQ?; 또는 :Sq?;		
+#^%	:SR 또는 ;SpHLHLHLHL,HLHLHLHL,HLHLHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL;	SR 또는 SpHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL;	모터2의 파라미터 리밋과 위치/속도/전류 제어파라미터를 설정함. (WR 명령어로 속도 및 전류제어 파라미터를 설정하는 경우는 제외)
+#^%	:SR?; 또는 :Sp?;		
+#^	:AQ 또는 ;AqHLHL,HLHLHLHL,HLHL,HLHLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL;	AQ 또는 AqHLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL;	모터1의 위치/속도 제어파라미터를 설정함. (WQ 명령어로 속도제어 파라미터를 설정하는 경우는 제외)
+#^	:AQ?; 또는 :Aq?;		
+#^	:AR 또는 ;ArHLHL,HLHLHLHL,HLHL,HLHLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL;	AR 또는 ArHLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL,HLHLHLHL;	모터2의 위치/속도 제어파라미터를 설정함. (WR 명령어로 속도제어 파라미터를 설정하는 경우는 제외)
+#^	:AR?; 또는 :Ar?;		
+#^	:AS 또는 ;AsHLHL,HLHLHLHL,HLHL,HLHLHLHLHL,HLHL,HLHLHLHLHL;	AS 또는 AsHLHL,HLHLHLHL,HLHL,HLHLHLHLHL;	모터1,2의 전류 제어파라미터를 설정함. (WQ, WR 명령어로 전류제어 파라미터를 설정하는 경우는 제외)
+#^	:AS?; 또는 :As?;		

명령어의 분류 (모터제어 파라미터의 설정 - 전류/속도/위치 게인2)

적용 CPU	명령어	return value	operation
+#^	;AJddd1,ddd2,ddd3,ddd4,ddd5,ddd6;	AJddd1,ddd2,ddd3,ddd4,ddd5,ddd6;	저가형 1채널 모터제어기에서 모터1의 위치/속도 제어파라미터를 설정함.
+#^	;AJ?;		

명령어의 분류 (제어 파라미터의 설정 - BLDC 모터 추가게인 설정)

적용 CPU	명령어	return value	operation
+#^ +#^	;WQdddd1,dddd2,dddd3,dddd4,dddd5; ;WQ?;	WQdddd1,dddd2,dddd3,dddd4,dddd5;	모터1의 03/06번 위치/속도제어모드에서 속도게인과 , RJM_VER7형의 3상/5상 STEP모터와 BLDC 모터1의 전류제어 파라미터를 설정함.
+#^ +#^	;WRdddd1,dddd2,dddd3,dddd4,dddd5; ;WR?;	WRdddd1,dddd2,dddd3,dddd4,dddd5;	모터2의 03/06번 위치/속도제어모드에서 속도게인과 , RJM_VER7형의 3상/5상 STEP모터와 BLDC 모터2의 전류제어 파라미터를 설정함.
+#^ +#^	;WEdddd1,dddd2; ;WE?;	WEdddd1,dddd2,dddd3,dddd4,dddd5, dddd6;	BLDC 모터1,2에 있어서 Hall_A상 상승에지에 대한 U상 전 압 positive zero crossing 포인트의 위상지연 값을 설정 하며, 모터1,2의 POLE_PAIR 수, 모터1,2의 MOTORx_POSITION_SCALE_FACTOR를 읽어서 host로 전송함.
+#^ +#^	;AWdddd1,dddd2; ;AW?;	AWdddd1,dddd2;	BLDC 모터에서 최대속도를 높이기 위한 field weakening 비례제어 게인을 설정함.
+#^ +#^	(Wafer control이 아닌 BLDC 모터의 경우로서 2012년 2월 18일 S/W 버전 부터) ;Shdddd1,dddd2,dddd3,dddd4; ;Sh?; 2013년 4월 4일 S/W 버전부터 SK/Sk 명령으로 바뀜.	Shdddd1,dddd2,dddd3,dddd4; 2013년 4월 4일 S/W 버전부터 SK/Sk 명령으로 바뀜.	BLDC모터에서 holding torque를 만드는 파라미터를 설정 함.
+#^	(Wafer control이 아닌 BLDC 모터의 경우) ;SKdddd1,dddd2,dddd3,dddd4; 또는 ;Skdddd1,dddd2,dddd3,dddd4; 또는 ;SK?; 또는 ;Sk?; 2013년 4월 4일 S/W 버전부터 SK 명령은 Sh 명령대신으로 사용되며, 전류값을 0.01A 단위로 설정할수 있는 Sk 명령은 추가되었음.		
+#^ +#^	;Gtdddd1,dddd2; ;Gt?;	Gtdddd1,dddd2;	RJM_VER7의 Hall 센서입력의 ON/OFF를 검출하는 기준전압 과 히스테리시스 진폭을 설정함.

명령어의 분류 (제어 파라미터의 설정 - 게인 스케줄링 파라미터 설정)

적용 CPU	명령어	return value	operation
+#^	;AX or ;AxA55A, dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14; ;AX?; or ;Ax?;	AX or AxA55A, dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14;	모터1의 상위 or 하위 속도에서 제어 게인을 높이거나 낮추는 파라미터를 설정함.
+#^	;AY or ;AyA55A, dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14; ;AY?; or ;Ay?;	AY or AyA55A, dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14;	모터2의 상위 or 하위 속도에서 제어 게인을 높이거나 낮추는 파라미터를 설정함.
+#^ +#^	;AZA55A, dddd1, dddd2; ;AZ?;	AZddd1, dddd2, dddd3, dddd4, dddd5, dddd6;	SQ/SR/Sq/Sr/AQ/AR/Aq/Ar/WQ/WR 등의 명령으로 설정하는 게인이 적용되는 중간 속도값을 설정함.
+#^	;AzA55A;	AzA55A;	AX/Ax/AY/Ay/AZ 명령에 의한 설정값과 advanced control algorithm의 파라미터를 EEPROM에 저장함. (단 저가형 1채널 제어기에서는 EsA55A; 명령으로 저장됨)

명령어의 분류 (제어 파라미터의 설정 - 기타 게인 설정1)

적용 CPU	명령어	return value	operation
+#^ +#^	;WDA55A, dddd1, dddd2; ;WD?;	WDddd1, dddd2;	미분값 계산을 실시하는 시간차를 설정함.
+#^ +#^	;Wdddd1, dddd2, dddd3, dddd 4, dddd5, dddd6; ;Wd?;	Wdddd1, dddd2, dddd3, dddd4, dddd5, dddd6;	inner 속도제어루프가 있는 03번 위치제어모드의 속도제어 또는 Digital Hall sensor만을 사용하는 BLDC모터의 위치/속도 제어 또는 0A번 속도제어모드 또는 속도제어로서 SX 명령어로 설정하는 두번째 파라미터의 bit11이 0인 경우에, 낮은 속도에서 위치/속도 제어게인(3번 위치제어모드의 경우는 속도게인만)을 낮게하는 파라미터를 설정함.
+#^ +#^	;WSddd1, dddd2; ;WS?;	WSddd1, dddd2;	위치제어모드에서 속도에 비례하는 damping 회전력을 발생시키는 게인을 설정함.
+#^ +#^	;Wsddd1, dddd2; ;Ws?;	Wsddd1, dddd2;	DD motor 등에서 속도에 비례하는 damping current를 발생시키는 게인을 설정함.
+#^ +#^	;Sfddd1, dddd2; ;Sf?;	Sfddd1, dddd2;	위치제어의 feedforward 게인을 설정함.

명령어의 분류 (제어 파라미터의 설정 - 기타 계인 설정2)

적용 CPU	명령어	return value	operation
+#^ +#^	(BLDC 모터의 경우) ;MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6; ;MB?;	MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6;	엔코더를 사용하지 않고 digital Hall IC만을 사용하는 BLDC 모터1의 경우, Hall 센서 A/B/C 상신호의 에지위치를 설정함.
+#^	(BLDC 모터의 경우) ;Mb?;	Mbddd1,ddd2,ddd3,ddd4,ddd5,ddd6;	analog Hall 센서를 사용하지 않는 BLDC 모터1의 경우 Hall 센서신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
+#^ +#^	(BLDC 모터의 경우) ;MEddd1,ddd2,ddd3,ddd4,ddd5,ddd6; ;ME?;	MEddd1,ddd2,ddd3,ddd4,ddd5,ddd6;	엔코더를 사용하지 않고 digital Hall IC만을 사용하는 BLDC 모터2의 경우, Hall 센서 A/B/C 상신호의 에지위치를 설정함.
+#^	(BLDC 모터의 경우) ;Me?;	Medddd1,ddd2,ddd3,ddd4,ddd5,ddd6;	analog Hall 센서를 사용하지 않는 BLDC 모터2의 경우 Hall 센서신호의 에지간 간격값을 읽어서 상위제어기로 전송함.

명령어의 분류 (제어 파라미터의 설정 - 기타 계인 설정3)

적용 CPU	명령어	return value	operation
+#^ +#^	(low resolution encoder의 DC 모터인 경우) ;MBddd1,ddd2,ddd3,ddd4; ;MB?;	MBddd1,ddd2,ddd3,ddd4;	low resolution encoder를 사용하는 DC 모터1의 경우, 엔코더신호 A/B 상신호의 에지위치를 설정함.
+#^	(low resolution encoder의 DC 모터인 경우) ;Mb?;	Mbddd1,ddd2,ddd3,ddd4;	low resolution encoder를 사용하는 DC 모터1의 경우 엔코더신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
+#^ +#^	(low resolution encoder의 DC 모터인 경우) ;MEddd1,ddd2,ddd3,ddd4; ;ME?;	MEddd1,ddd2,ddd3,ddd4;	low resolution encoder를 사용하는 DC 모터2의 경우, 엔코더신호 A/B 상신호의 에지위치를 설정함.
+#^	(low resolution encoder의 DC 모터인 경우) ;Me?;	Meddd1,ddd2,ddd3,ddd4;	low resolution encoder를 사용하는 DC 모터2의 경우 엔코더신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
+#^ +#^ +#^	(low resolution encoder의 DC 모터 또는 BLDC 모터인 경우) ;MFddd1,ddd2; (2013년 2월 9일 이전 S/W 버전) ;MFHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; ; (2013년 2월 9일 S/W 버전부터) ;MF?;	MFddd1,ddd2; (2013년 2월 9일 이전 S/W 버전) MFHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (2013년 2월 9일 S/W 버전부터)	low resolution encoder를 사용하는 DC 모터 또는 BLDC 모터인 경우에 속도신호의 변화를 제한하는 범위를 설정함.

명령어의 분류 (제어 파라미터의 설정 - advanced controll algorithm 파라미터 설정)

적용 CPU	명령어	return value	operation
+#^ +#^	;AUeeee1,eeee2,eeee3,eeee4,eeee5,eeee6; ;AU?;	AUgggg1,gggg2,gggg3,gggg4,gggg5,gggg6;	Observer에서 사용하는 모터1의 파라미터를 설정함.
+#^ +#^	;Aueeee1,eeee2,eeee3,eeee4,eeee5,eeee6; ;Au?;	Augggg1,gggg2,gggg3,gggg4,gggg5,gggg6;	Advanced controller에서 사용하는 모터1의 파라미터를 설정함.
+#^ +#^	;AVeeee1,eeee2,eeee3,eeee4,eeee5,eeee6; ;AV?;	AVgggg1,gggg2,gggg3,gggg4,gggg5,gggg6;	Observer에서 사용하는 모터2의 파라미터를 설정함.
+#^ +#^	;Aveeee1,eeee2,eeee3,eeee4,eeee5,eeee6; ;Av?;	Avgggg1,gggg2,gggg3,gggg4,gggg5,gggg6;	Advanced controller에서 사용하는 모터2의 파라미터를 설정함.
+#^ +#^	;AaHLHL,HLHL; ;Aa?;	AaHLHL,HLHL;	Advanced controller 를 activate 시키는 파라미터를 설정함.

명령어의 분류 (제어 파라미터의 설정 - 한계값, dead-time 및 펄스주기 속도계산)

적용 CPU	명령어	return value	operation
+#^% +#^%	;SLiiii1,iiii2,iiii3,iiii4; ;SL?;	SLiiii1,iiii2,iiii3,iiii4;	위치제어모드에서 위치의 한계값을 설정함.
+#^% +#^%	;Swddd1,ddd2; ;Sw?;	Swddd1,ddd2;	제어시 사용하는 전류의 리밋을 설정함.
+#^ +#^	;Szddd1,ddd2; ;Sz?;	Szddd1,ddd2;	제어시 사용하는 전류의 리밋을 0.01A 단위로 설정함.
+#^ +#^	;MRddd1,ddd2,ddd3,ddd4; ;MR?; 2013년 4월 4일 S/W 버전부터는 Mr 명령을 사용하여 0.01A 단위로 전류값을 설정할수있음.	MRddd1,ddd2,ddd3,ddd4; 2013년 4월 4일 S/W 버전부터는 Mr 명령을 사용하여 0.01A 단위로 전류값을 설정할수있음.	전류상승률을 제한하는 파라미터를 설정함.
+#^	;Mrddd1,ddd2,ddd3,ddd4; 또는 ;Mr?; 2013년 4월 4일 S/W 버전부터는 MR 명령어로 하던 설정을 Mr 명령을 사용하여 0.01A 단위의 전류값으로 설정할수있음. 2013년 4월 4일 S/W 버전부터는 SI 명령어로 하던 설정을 Si 명령을 사용하여 0.01A 단위의 전류값으로 설정할수있음.		
+#^ +#^	;AwA55A,ddd1,ddd2; ;Aw?;	Awddd1,ddd2;	사용가능한 최대크기의 PWM duty 비를 설정함.
+#^ +#^	;ADA55A,iiii1,iiii2; ;AD?;	ADiiii1,iiii2;	MOTOR 구동 Half-bridge의 dead-time을 보상하되, Z6 명령어로 설정하는 dead-time에 추가하여 보상하려는 보상량을 설정함.
+ +	;Acddd1,ddd2; ;Ac?;	Acddd1,ddd2;	엔코더펄스 카운트 방식으로 속도를 계산할 때 moving average 구간의 크기를 설정함.
# #	;Acddd1,ddd2,ddd3,ddd4; ;Ac?;	Acddd1,ddd2,ddd3,ddd4;	엔코더펄스 카운트 방식으로 속도를 계산할 때 moving average 구간의 크기와, 엔코더펄스의 주기를 사용하여 속도를 계산할때 주기계산에 사용되는 엔코더펄스개수의 2에 대한 지수를 설정함.

명령어의 분류 (제어 파라미터의 설정 - compensator, filter 및 정지강성)

적용 CPU	명령어	return value	operation
+#^ +#^	;Sgiiii1,iiii2,iiii3,iiii4; ;Sg?;	Sgiiii1,iiii2,iiii3,iiii4;	위치제어 또는 속도제어에서 phase lead 또는 phase lag compensator의 파라미터를 설정함. (이 경우는 미분제어 게인이 무시됨)
+#^% +#^%	;SFddd1,ddd2,ddd3; ;SF?ddd1;	SFddd1,ddd2,ddd3;	butterworth filter의 시정수와 변화율 상한값을 설정함.
# #	;ALddd1,ddd2,ddd3,ddd4; ;AL?;	ALddd1,ddd2,ddd3,ddd4;	28~31번 butterworth filterA의 시정수를 설정함.
+#^	;Mfddd1;	Mfddd1,ddd2;	butterworth filterA의 시정수를 읽어서 호스트로 전송함.
+#^ +#^	;ANddd1,ddd2; ;AN?;	ANddd1,ddd2;	모터1에 적용되는 Notch filter의 중심주파수와 Q factor를 설정함.
+#^ +#^	;Anddd1,ddd2; ;An?;	Anddd1,ddd2;	모터2에 적용되는 Notch filter의 중심주파수와 Q factor를 설정함. (저가형 1채널 제어기에서는 모터1에 적용됨)
+#^ +#^	;Suddd1,ddd2; ;Su?;	Suddd1,ddd2;	위치제어에서 정지위치 근방에서 강성을 높이기 위하여 제어 게인을 높게 하는 비율을 설정함.
+#^ +#^	;ABA55A,ddd1; ;AB?;	ABddd1;	위치제어에서 정지하고 있을때, jitter와 소음을 줄이거나 또는 강성을 높이기 위하여, 위치제어 게인을 낮추거나 높이기 위한 파라미터를 설정함.

명령어의 분류 (제어 파라미터의 설정 - 기타 파라미터1)

적용 CPU	명령어	return value	operation
+#^ +#^	;Dd <code>dddd1</code> ; ;Dd?;	D <code>dddd1</code> ;	포텐시오펜터로 속도명령 또는 전류명령을 받아서 움직이는 모드에서 속도명령값 또는 전류명령값의 deadband 크기를 설정함.
+#^ +#^	;DOA55A, <code>dddd1</code> , <code>dddd2</code> ; ;DO?;	DO <code>dddd1</code> , <code>dddd2</code> ;	포텐시오펜터로 속도명령 또는 전류명령을 받아서 움직이는 모드에서 속도명령값 또는 전류명령값이 0일때의 LSxC 포트 A/D 변환값을 설정함.
+#^ +#^	;Do?;	Do <code>dddd1</code> , <code>dddd2</code> ;	LSxC 포트 A/D 변환값의 필터링값을 읽어서 호스트로 전송함.
# #	(절대각센서를 사용하는 경우) ;MO <code>iiii1</code> , <code>iiii2</code> ; ;MO?;	MO <code>iiii1</code> , <code>iiii2</code> ;	절대각센서를 사용하는 경우에 BLDC 모터의 Hall A/B/C상 신호와 모터위상각신호의 위상차를 없애기 위한 위상 offset을 설정함.
+#^ +#^	;MA <code>dddd1</code> , <code>dddd2</code> , <code>dddd3</code> ; ;MA?;	MA <code>dddd1</code> , <code>dddd2</code> , <code>dddd3</code> ;	자동화용 H/W 버전에서 상위명령값과 실제값의 차이가 설정값이내에 들어왔는지를 판단하고 mission 완료를 결정하는 경계값(threshold)을 설정함.
+#^ +#^	;WP <code>dddd1</code> , <code>dddd2</code> ; ;WP?;	WP <code>dddd1</code> , <code>dddd2</code> ;	Digital Hall sensor type의 RJM_VER7 모터에 대하여 정지 희망 edge 번호를 지정함.
+#^ +#^	;Wp <code>dddd1</code> , <code>dddd2</code> , <code>dddd3</code> , <code>dddd4</code> , <code>dddd5</code> ; ;Wp?;	Wp <code>dddd1</code> , <code>dddd2</code> , <code>dddd3</code> , <code>dddd4</code> , <code>dddd5</code> ;	Digital Hall sensor type의 RJM_VER7 모터에 대하여 감속 완료후 안정화를 개시하는 지연시간과 안정화용 analog Hall sensor의 offset, 및 analog Hall 센서가 사용되는 위치오차 범위등을 설정함.

명령어의 분류 (제어 파라미터의 설정 - 기타 파라미터2)

적용 CPU	명령어	return value	operation
+#^ +#^	;Wciiii1,iiii2; ;Wc?;	Wciiii1,iiii2;	유도모터의 경우에 Id 전류의 기준값을 설정함.
+#^ +#^	;Wqdddd1,dddd2; ;Wq?;	Wqdddd1,dddd2;	모터1의 MAX_PERCENT_SLIP1, ELECTRIC_TIME_CONSTANT1 값을 설정함.
+#^ +#^	;Wrdddd1,dddd2; ;Wr?;	Wrdddd1,dddd2;	모터2의 MAX_PERCENT_SLIP2, ELECTRIC_TIME_CONSTANT2 값을 설정함.
+#^ +#^	;WTdddd1,dddd2,dddd3,dddd4; ;WT?;	WTdddd1,dddd2,dddd3,dddd4;	BLDC 모터에 있어서 cogging torque를 보상하는 파라미터를 설정함.
+#^ +#^	;WAdddd1,dddd2; ;WA?;	WAdddd1,dddd2;	속도모드에서 속도오차가 클때 속도명령값을 수정하는 속도오차의 최소크기를 설정함.
+#^ +#^	;AtA55A,dddd1,dddd2; ;At?;	Atdddd1,dddd2;	모터1과 모터2의 제어알고리즘 적용시점을 설정함.
+#^ +#^	;Addddd1,dddd2; ;Ad?;	Addddd1,dddd2;	모터1과 모터2의 마찰력을 보상하는 전류크기를 설정함.

명령어의 분류 (작동개시, 중지 및 비상정지 관련 명령어)

적용 CPU	명령어	return value	operation
+#^%	;PEHLHL;		
+#^	;PEA55A;	PEHLHL;	선택된 보드의 모터 출력을 ON 시킴.
+#^%	;PE?;		
+#^%	;SMHLHL;		
+#^%	;SM?;	SMHLHL;	위치, 속도, 전류제어 등의 작동모드를 설정함.
+#^	;RBHLHL;		
+#^	;RB?;	RBHLHL;	feedback 제어를 개시하도록 작동모드 설정에 필요한 조치를 자동으로 처리하게 함.
+#^%	;PDHLHL;		
+#^	;PDA55A;	PDHLHL;	선택된 보드의 모터 출력을 OFF 시킴.
+#^	;PD?;		
+#^%	;PPR; or ;PQR;	PPR; or PQR;	모터가 stall된 채로 정지된 상태를 해지함.
+#^%	;PRHLHL;		
+#^	;PRA55A;	PRHLHL;	선택된 보드의 fault flag를 clear 시킴.
+#^	;RrA55A;	N.A.	모터제어기를 리셋함.
+#^	;RSHLHL;		
+#^	;RS?;	RSHLHL;	비상정지시의 정지 형태를 정의함.
+#^	;RsA55A;	RsA55A;	모터1과 모터2가 명령에 의한 비상정지를 실행함.
+#^	;RsA5A5;	RsA5A5;	모터1이 명령에 의한 비상정지를 실행함.
+#^	;Rs5A5A;	Rs5A5A;	모터2가 명령에 의한 비상정지를 실행함.
+#^	;Sodddd1, dddd2;		
+#^	;So?;	Sodddd1, dddd2;	속도제어 및 위치제어 모드에서 리밋센서, 비상정지 명령 또는 비상정지 상황에 의한 강제 감속시의 감속율을 설정함.

명령어의 분류 (강제정지 명령어)

적용 CPU	명령어	return value	operation
+#^%	;PsA55A;	PsA55A;	모터1,2를 모두 현재의 위치에서 감속을 실시한 후에 정지함.
+#^	;PtA55A;	PtA55A;	
+#^%	;PsA5A5;	PsA5A5;	모터1을 현재의 위치에서 감속을 실시한 후에 정지함.
+#^	;PtA5A5;	PtA5A5;	
+#^%	;Ps5A5A;	Ps5A5A;	모터2를 현재의 위치에서 감속을 실시한 후에 정지함.
+#^	;Pt5A5A;	Pt5A5A;	
+#^	;PeA55A;	PeA55A;	digital Hall sensor만을 사용하는 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1,2 모두를 02번 위치제어모드로 바꾸고, 현재의 위치에서 감속후 정지함.
+#^	;PeA5A5;	PeA5A5;	digital Hall sensor만을 사용하는 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1을 02번 위치제어모드로 바꾸고, 현재의 위치에서 감속후 정지함.
+#^	;Pe5A5A;	Pe5A5A;	digital Hall sensor만을 사용하는 BLDC 모터에 있어서, 05번 속도제어 상태의 모터2를 02번 위치제어모드로 바꾸고, 현재의 위치에서 감속후 정지함.
+#^	;PvA55A;	PvA55A;	digital Hall sensor만을 사용하는 wafer 회전구동용 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1,2 모두를 감속한 후에, 낮은 속도가 되면 02번 위치제어모드로 바꾸고, 감속후 정지함.
+#^	;PvA5A5;	PvA5A5;	digital Hall sensor만을 사용하는 wafer 회전구동용 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1을 감속한 후에, 낮은 속도가 되면 02번 위치제어모드로 바꾸고, 감속후 정지함.
+#^	;Pv5A5A;	Pv5A5A;	digital Hall sensor만을 사용하는 wafer 회전구동용 BLDC 모터에 있어서, 05번 속도제어 상태의 모터2를 감속한 후에, 낮은 속도가 되면 02번 위치제어모드로 바꾸고, 감속후 정지함.

명령어의 분류 (전류구동 명령어)

적용 CPU	명령어	return value	operation
+ ^% + ^%	;SCiiii1,iiii2; ;SC?;	SCiiii1,iiii2; SCiiii1,iiii2,iiii3,iiii4; (RJM_VER5 STEP형 만 해당됨) SCiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (RJM_VER6 3상STEP형 만 해당됨)	전류제어 모드에서 전류명령값을 설정함.
+ ^ + ^	;Sciiii1,iiii2; ;Sc?;	Sciiii1,iiii2;	전류제어 모드에서 0.01A 단위로 전류명령값을 설정함.
+ ^ + ^	;Pciiii1,iiii2; ;Pc?;	Pciiii1,iiii2;	digital Hall sensor만을 사용하는 BLDC 모터가 아닌 경우에, 전류제어 모드에서 전류명령값을 설정하되, 리턴값으로 현재 속도값을 host로 전송함.

명령어의 분류 (속도구동 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^ +#^	;SV 1, 2; ;SVr 1, 2; ;SV?;	SV 1, 2;	속도제어 모드에서 속도명령값을 설정함.
+#^ +#^ +#^	;PV 1, 2,dddd3; ;Pvr 1, 2,dddd3; ;PV?;	PV 1, 2;	속도제어 모드에서 속도명령값을 설정하되 설정값의 checksum을 사용함.
+#^ +#^	;Pv 1, 2; ;Pv?;	Pv 1, 2;	digital Hall sensor만을 사용하는 BLDC 모터가 아닌 경우에, 속도제어 모드에서 속도명령값을 설정하되, 리턴값으로 현재 위치의 원점기준 변위값을 host로 전송함.
+#^% +#^%	;Sadddd1,dddd2,dddd3,dddd4; ;Sa?;	Sadddd1,dddd2,dddd3,dddd4;	속도제어 및 위치제어 모드에서 가속 및 감속율을 설정함.

명령어의 분류 (point-to-point 위치구동 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^% +#^%	;PA 1, 2; ;PA?; 2014년 3월 21일 S/W 버전부터	PA 1, 2;	모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함. (PC는 2014년 3월 21일 S/W 버전부터 적용)
+#^ +#^	;PC 1, 2; ;PC?;	PC 1, 2;	
+#^ +#^	2013년 7월 31일 S/W 버전부터 ;PAa 1; ;PAa?; 2014년 3월 21일 S/W 버전부터	PAa 1;	모터제어기의 작동모드에 따라서 모터1의 위치/속도/전류의 명령값을 설정함. (PAa는 2013년 7월 31일 S/W 버전부터 적용, PCa는 2014년 3월 21일 S/W 버전부터 적용)
+#^ +#^	;PCa 1; ;PCa?;	PCa 1;	
+#^ +#^	2013년 7월 31일 S/W 버전부터 ;PAb 1; ;PAb?; 2014년 3월 21일 S/W 버전부터	PAb 1;	모터제어기의 작동모드에 따라서 모터2의 위치/속도/전류의 명령값을 설정함. (PAb는 2013년 7월 31일 S/W 버전부터 적용, PCb는 2014년 3월 21일 S/W 버전부터 적용)
+#^ +#^	;PCb 1; ;PCb?;	PCb 1;	

명령어의 분류 (point-to-point 위치구동 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^% +#^%	;PBHLHLHLHL,HLHLHLHL; ;PB?;	PBHLHLHLHL,HLHLHLHL;	모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함.
+#^ +#^	;PBaHLHLHLHL; ;PBa?;	PBaHLHLHLHL;	모터제어기의 작동모드에 따라서 모터1의 위치/속도/전류의 명령값을 설정함. (2013년 7월 31일 S/W 버전부터 적용)
+#^ +#^	;PBbHLHLHLHL; ;PBb?;	PBbHLHLHLHL;	모터제어기의 작동모드에 따라서 모터2의 위치/속도/전류의 명령값을 설정함. (2013년 7월 31일 S/W 버전부터 적용)
+#^% +#^ +#^ +#^	;PaHBMBLBHBMBLBCS; ;PbHBMBLBHBMBLBCS; (2013년 1월 13일 S/W 버전부터 적용) (2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우) ;PaHBMBLBCS; ;PbHBMBLBCS;	PaHBMBLBHBMBLBCS; (현재 위치값을 리턴) PbHBMBLBHBMBLBCS; (작동모드에 따라서 현재 위치/속도/전류값을 리턴) (2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우) PaHBMBLBCS; PbHBMBLBCS;	모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함.

명령어의 분류 (연속 point-to-point 위치구동 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^	;P1 1, 2; ;P1?;	P1 1, 2;	위치 명령값의 증분을 설정하고 이동을 명령함.
+#^ +#^	;P1 1; ;P1?;	P1 1;	모터1에 대하여 위치 명령값의 증분을 설정하고 이동을 명령함.
+#^ +#^	;P2 1; ;P2?;	P2 1;	모터2에 대하여 위치 명령값의 증분을 설정하고 이동을 명령함.
+#^ +#^	;PH0; ;PH0;	PH0; PH0;	흡위치 또는 가상흡위치로의 이동명령을 수행함.
+#^% +#^%	;SSddd1,ddd2; ;SS?;	SSddd1,ddd2;	위치제어모드에서 Point-to-Point 이동시 이동속도값을 설정함.
+#^% +#^%	;Ssddd1,ddd2; ;Ss?;	Ssddd1,ddd2;	PA, PB, Pa 명령에 의한 위치구동시의 가속/감속 기간을 설정함.
+#^ +#^	;AI dddd1,ddd2,ddd3,ddd4; ;AI?;	AI dddd1,ddd2,ddd3,ddd4;	PA 또는 SV 명령에 의한 가감속시에 전류명령값의 feedforward 값을 계산하는 scale factor를 설정함.
+#^ +#^	;Ai dddd1,ddd2; ;Ai?;	Ai dddd1,ddd2;	PA 명령에 의한 가속시에 사다리꼴 모양의 가속 프로파일을 만들기 위한 가속 변화율값을 설정함.
+#^ +#^	;ATddd1; ;AT?;	ATddd1;	PA 명령의 완료 시점에서 추가로 실행하는 기능을 위한 시간파라미터를 설정함.
+#^% +#^%	;SPddd1, 2, 3; ;SP?ddd1;	SPddd1, 2, 3;	좌표점을 정의함.
+#^% +#^%	;PMddd1,ddd2,ddd3; ;PM?;	PMddd1,ddd2,ddd3;	현재위치로부터 dddd1점을 거쳐 dddd2점까지 dddd3의 속도로 이동시킴.
+#^%	;PME;	PME;	PM 에 의한 동작을 멈춤.

명령어의 분류 (실시간 연속 위치구동 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^% +#^%	;Stdddd1,dddd2,ddd d3; ;St?;	Stdddd1,dddd2,dddd3;	PP/Pp/PQ/Pq 명령어의 update 주기와 PP/Pp/PQ/Pq 명령어 미수신의 연속허용횟수를 설정함.
+#^% +#^%	;SeHLHL; ;Se?;	SeHLHL;	모터1 과 모터2 에 대하여 PM, PP, PQ, Pp, Pq command를 적용가능토록 설정함.
+#^% +#^% +#^%	;SrHLHL,HLHL; ;SrHLHL,HLHL,HLHL, HLHL; ;Sr?;	SrHLHL,HLHL; SrHLHL,HLHL,HLHL,HLHL;	Pp/Pq/p 명령어에 대한 return값의 주소를 설 정함.
+#^% +#^%	;PP 1, 2; ;PQ 1, 2;	PP 1, 2; PQ 1, 2;	모터제어기의 작동모드에 따라서 10msec/30msec 동안에 가야할 위치/속도/전류 의 명령값을 설정하고 모터가 움직이도록 함. PP인경우는 10msec, PQ인 경우는 30msec임.

명령어의 분류 (실시간 연속 위치구동 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^% +#^% +#^ +#^	;PpHBMBLBHMBLBCS; ;PqHBMBLBHMBLBCS; (2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우) ;PpHBMBLBCS; ;PqHBMBLBCS;	2006년 10월 7일 이전 S/W 버전 : PpHBMBLBHMBLBCS; PqHBMBLBHMBLBCS; 2006년 10월 7일 S/W 버전부터 : PpLBHBLBLBHBLBCS; 또는 PpHBLBHBLBHBLBCS; (2007년 9월 19일 S/W 버전부터 사용가능) PqLBHBLBLBHBLBCS; 또는 PqHBLBHBLBHBLBCS; (2007년 9월 19일 S/W 버전부터 사용가능)	모터제어기의 작동모드에 따라서 10msec/30msec 동안에 가야할 위치/속도/전류의 명령값을 설정하고 모터가 움직이도록 함. Pp인경우는 10msec, Pq인 경우는 30msec임.
+#^% +#^% +#^ +#^	;pHBMBLBHMBLBCS; (RS485의 경우) ;pHBMBLBHMBLBCS (CAN의 경우 8 byte, 1 packet) (2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우) ;pHBMBLBCS; (RS485의 경우) ;pHBMBLBCS (CAN의 경우 5 byte, 1 packet)	RS485의 경우 : pLBHBLBLBHBLBCS; 또는 pHBLBHBLBHBLBCS; (2007년 9월 19일 S/W 버전부터 사용가능) CAN의 경우 : LBHBLBLBHBLB 또는 HBLBHBLBHBLB (2007년 9월 19일 S/W 버전부터 사용가능)	모터제어기의 작동모드에 따라서 10msec 동안에 가야할 위치/속도/전류의 명령값을 설정하고 모터가 움직이도록 함.
+#^%	;PPE; or ;PQE;	PPE; or PQE;	PP, PQ, Pp, Pq, p 에 의한 동작을 멈춤.

명령어의 분류 (스텝모터 구동관련 명령어)

적용 CPU	명령어	return value	operation
+#^% +#^% +#^%	(RJM_VER5/6 STEP형인 경우) ;SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; ;SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7,iiii8; ;SH?; 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함.	SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (RJM_VER5 STEP형 만 해당됨) SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7,iiii8; (RJM_VER6 3상 STEP형 만 해당됨) 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함.	RJM_VER5/6 STEP형의 host direct electrical angle 제어 모드에서 구동전류의 진폭과 전류 OFFSET을 설정함.
+#^	(RJM_VER5/6 STEP형인 경우) ;Shiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; 또는 ;Shiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7,iiii8; 또는 ;Sh?; 2013년 4월 4일 S/W 버전부터는 SH 명령어로 하던 설정을 Sh 명령을 사용하여 0.01A 단위의 전류값으로 설정할수있음.		
+#^% +#^%	(스텝모터의 경우) ;Shdddd1,dddd2,dddd3,dddd4; ;Sh?; 2013년 4월 4일 S/W 버전부터 SK/Sk 명령으로 바뀜.	Shdddd1,dddd2,dddd3,dddd4; 2013년 4월 4일 S/W 버전부터 SK/Sk 명령으로 바뀜.	스텝모터에서 전류명령값의 최대치와 정지시의 전류치 비율을 설정함.
+#^	(스텝모터의 경우) ;SKdddd1,dddd2,dddd3,dddd4; 또는 ;Skdddd1,dddd2,dddd3,dddd4; 또는 ;SK?; 또는 ;Sk?; 2013년 4월 4일 S/W 버전부터 SK 명령은 Sh 명령대신으로 사용되며, 전류값을 0.01A 단위로 설정할수 있는 Sk 명령은 추가되었음.		
+#^ +#^ +#^	;XEdddd1,dddd2; or ;XEiiii1,iiii2; ;XE?;	XEdddd1,dddd2; or XEiiii1,iiii2;	스텝모터 또는 절대각엔코더를 사용하는 경우 엔코더의 4채배된 펄스수를 설정함.
+#^ +#^	;Xedddd1,dddd2; ;Xe?;	Xedddd1,dddd2;	스텝모터에서 DC-DC converter의 승압전압의 크기와 turn-off에 소요되는 시간을 설정함.

명령어의 분류 (jog 작동 명령어)

적용 CPU	명령어	return value	operation
+#^%	;J;	J;	모터1의 속도모드 조그 작동을 수행함.
+#^%	;K;	K;	모터2의 속도모드 조그 작동을 수행함.
+#^%	;SJHL, dddd1, dddd2, dddd3;	SJHL, dddd1, dddd2, dddd3;	속도모드의 조그동작에서 조그방향, 조그이동속도 및 조그 동작 지속시간을 정의함.
+#^%	;SJ?;		
+#^%	;j;	j;	모터1의 위치모드 조그 작동을 수행함.
+#^%	;k;	k;	모터2의 위치모드 조그 작동을 수행함.
+#^%	;SjHL, dddd1, dddd2, dddd3;	SjHL, dddd1, dddd2, dddd3;	위치모드의 조그동작에서 조그방향, 조그이동속도 및 조그 동작 지속거리를 정의함.
+#^%	;Sj?;		

명령어의 분류 (homing 작동 명령어)

적용 CPU	명령어	return value	operation
+#^%	:GHHL,HL,iiii1,iiii2;	GHL,HL,IIII1,IIII2;	2개의 리미트센서와 1개의 홈센서를 사용한 홈 동작을 개시함. (0.01도 단위의 offset 변위)
+#^	:GH?;		
+#^	:GhHL,HL,IIII1,IIII2;	GhHL,HL,IIII1,IIII2;	2개의 리미트센서와 1개의 홈센서를 사용한 홈 동작을 개시함. (4체배 엔코더 펄스단위의 offset 변위)
+#^	:Gh?;		
+#^	:GIHL,HL,iiii1,iiii2;	GIHL,HL,IIII1,IIII2;	2개의 리미트센서 만을 사용한 홈 동작을 개시함. (0.01도 단위의 offset 변위)
+#^	:GI?;		
+#^	:GiHL,HL,IIII1,IIII2;	GiHL,HL,IIII1,IIII2;	2개의 리미트센서 만을 사용한 홈 동작을 개시함. (4체배 엔코더 펄스단위의 offset 변위)
+#^	:Gi?;		
+#^	:GJHL,HL,iiii1,iiii2;	GJHL,HL,IIII1,IIII2;	한개의 홈센서 만을 사용한 홈 동작을 개시함. (0.01도 단위의 offset 변위)
+#^	:GJ?;		
+#^	:GjHL,HL,IIII1,IIII2;	GjHL,HL,IIII1,IIII2;	한개의 홈센서 만을 사용한 홈 동작을 개시함. (4체배 엔코더 펄스단위의 offset 변위)
+#^	:Gj?;		
+#^	:GKHL,HL,iiii1,iiii2;	GKHL,HL,IIII1,IIII2;	한개의 홈센서 만을 사용한 홈 동작을 개시하되 홈센서가 ON 되어 있는 상태에서만 개시함. (0.01도 단위의 offset 변위)
+#^	:GK?;		
+#^	:GkHL,HL,IIII1,IIII2;	GkHL,HL,IIII1,IIII2;	한개의 홈센서 만을 사용한 홈 동작을 개시하되 홈센서가 ON 되어 있는 상태에서만 개시함. (4체배 엔코더 펄스단위의 offset 변위)
+#^	:Gk?;		
+#^	:SIIIII1,IIII2,IIII3,IIII4;	SIIIII1,IIII2,IIII3,IIII4;	홈찾기 동작에서 위치의 한계값을 설정함.
+#^	:SI?;		
+#^	:GTddd1,ddd2;	GTddd1,ddd2;	리미트센서와 홈센서의 ON/OFF를 검출하는 경계전압을 설정함.
+#^	:GT?;		

명령어의 분류 (sequence 제어 명령어)

적용 CPU	명령어	return value	operation
+#^	;XPXAAAAA.....AAAA;	XPXAAAAA.....AAAA;	시퀀스제어 프로그램의 현재 포인터 위치에 시퀀스제어 프로그램을 입력함.
+#^	;XP?;	XPAAAAA.....AAAA;	시퀀스제어 프로그램의 현재 포인터 위치에 있는 시퀀스제어 프로그램 한 스텝을 보여주고, 포인터 위치는 변경하지 않음.
+#^	;XP+;	XPAAAAA.....AAAA;	시퀀스제어 프로그램의 현재 포인터 위치에 있는 시퀀스제어 프로그램 한 스텝을 보여주고, 포인터 위치를 다음 스텝위치로 이동함.
+#^	;XPddd1;	XPAAAAA.....AAAA;	시퀀스제어 프로그램의 포인터 위치를 강제 설정하고, 설정된 포인터 위치의 시퀀스제어 프로그램 한 스텝을 보여주고, 포인터 위치는 변경하지 않음.
+#^	;XPp;	XPpddd1;	현재의 시퀀스제어 프로그램 포인터 위치를 보여줌.
+#^	;XPsA55A;	XPsA55A;	메모리상의 시퀀스제어 프로그램을 EEPROM으로 저장함.
+#^	;XPt;	XPtHLHL,HLHL,HLHL;	현재의 X_timer0, X_timer0_period, X_counter0의 값을 보여줌.
+#^	;XPP;	XPPHLHL,HLHL;	현재와 직전의 시퀀스제어 프로세스 카운트를 보여줌.
+#^	;XPT;	XPTHHLHL,HLHL; (2013년 8월 28일 이전 S/W 버전인 경우나 자동화버전이 아닌 경우) XPTHHLHLHLHL,HLHLHLHL; (자동화버전으로서 2013년 8월 28일 S/W 버전부터인 경우)	현재의 PORT_STATUS_FOR_X_CONTROL와 X_PORT_STATUS를 보여줌.
+#^	;XPF;	XPFHLHL;	현재의 X_STATUS_FLAG를 보여줌.
+#^	;XPS;	XPSHLHL,HLHL;	한스텝씩 시퀀스제어 프로그램을 실행시키는 모드일 때 한스텝 씩 시퀀스제어 프로그램을 실행하게 함.

명령어의 분류 (출력 강제작동 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^ +#^ +#^	;PWiiii1,iiii2; (VER4/8) ;PWiiii1,iiii2,iiii3,iiii4; (VER5) ;PWiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (VER6/7) ;PW?;	PWiiii1,iiii2; (VER4/8) PWiiii1,iiii2,iiii3,iiii4; (VER5) PWiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (VER6/7)	모터의 PWM duty 출력을 강제로 설정함.
+#^ +#^	;DXHLHL,HLHLHL; ;DX?;	DXHLHL,HLHL,HLHL,HLHL,HLHL;	ON/OFF 출력과 7-seg 표기값을 설정하고 ON/OFF 입력과 switch 입력을 읽어서 host로 전송함.
+#^ +#^	;XOHL,HL; ;XO?;	XOHL,HL;	EXT_FAULT1_port와 EXT_FAULT2_port를 강제로 set 또는 clear 함.
#^ #^	(스테핑모터의 경우) ;Xodddd1; ;Xo?;	Xodddd1;	스테핑모터에서 50V 승압회로와 역기전력 해소 pass를 ON/OFF 함.
#^ #^	(고전압 제어기인 경우) ;Xodddd1; ;Xo?;	Xodddd1;	380V 용 모터제어기에서 PRECHARGE_RELAY, MAIN_RELAY, BRAKE_RELAY를 강제 설정함.
+#^ +#^	;Xpddd1,ddd2; ;Xp?;	Xpddd1,ddd2;	변위값이 일정량 증가 또는 감소할 때 fault 출력 단자를 통하여 일정기간의 low 펄스를 내보내는 기능에서 변위값의 증분을 설정함.
+#^ +#^	;Xqddd1,ddd2; ;Xq?;	Xqddd1,ddd2;	변위값이 일정량 증가 또는 감소할 때 fault 출력 단자를 통하여 일정기간의 low 펄스를 내보내는 기능에서 low 펄스의 기간을 설정함.
+#^	;Xr; 또는 Xr?;	XrIIII1,IIII2;	Xp 명령으로 펄스개시 출력을 개시하는 순간의 위치값을 읽어서 호스트로 전송함.

명령어의 분류 (입력/출력 모니터 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^%	:QS;	QSHL,HLHL,HLHL,HLHL; (스텝핑모터가 아닌 경우) QSHL,HLHL,HLHLHLHL,HLHLHLHL; (스텝핑모터인 경우) QSHL,HLHL,HLHLHLHL; (저가형 1채널 제어기인 경우)	on/off 형식의 센서 상태를 읽어서 host로 전송함.
+#^	(절대각센서를 사용하지 않는 경우) :Qs;	QsHLHL,HLHL,HLHL,HLHL;	인공안구의 경우는 팬/틸트방향의 절대각을 읽어서 host로 전송하며, 베스트모션의 경우는 경사각 센서값을 읽어서 host로 전송함.
#	(절대각센서를 사용하는 경우) :Qs;	QsHLHLHLHL,HLHLHLHL;	절대각센서값을 읽어서 host를 전송함.
+#^%	:QA;	QAHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	하위 8 channel의 12-bit AD 변환값을 읽어서 host로 전송함.
+#^%	:QB;	QBHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	상위 8 channel의 12-bit AD 변환값을 읽어서 host로 전송함.
+#^	:Qa;	QaHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	TLV2556의 12-bit A/D 변환값을 읽어서 host로 전송함.
+	:QGAdddd1;	QGAAdd,A,A,A;	TMS320F2810/11/12를 사용한 모터제어기에서 개별 입출력 포트의 H/L 상태, 입출력 설정 상태, MUX 설정 상태를 읽어서 호스트로 전송함.
+	:QGBdddd1;	QGBdd,A,A,A;	
+	:QGDdddd1;	QGDdd,A,A,A;	
+	:QGEdddd1;	QGEdd,A,A,A;	
+	:QGFdddd1;	QGFdd,A,A,A;	
+	:QGGdddd1;	QGGdd,A,A,A;	
#^	:QGdddd1;	QGAAdd,A,A,A; QGBdd,A,A,A; QGCdd,A,A,A;	TMS320F28334/335/09를 사용한 모터제어기에서 개별 입출력 포트의 H/L 상태, 입출력 설정 상태, MUX 설정 상태를 읽어서 호스트로 전송함.

명령어의 분류 (입력/출력 모니터 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^	;QH0;	QH31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QHBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)의 H/L 상태를 읽어서 호스트로 전송함.
+#^	;QH1;	QH63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QHED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)의 H/L 상태를 읽어서 호스트로 전송함.
+#^	;QH2;	QH95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QHGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)의 H/L 상태를 읽어서 호스트로 전송함.
+#^	;Qh0;	Qh31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QhBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)가 입력(0)/출력(1) 포트로 설정된 상태를 읽어서 호스트로 전송함.
+#^	;Qh1;	Qh63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QhED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)가 입력(0)/출력(1) 포트로 설정된 상태를 읽어서 호스트로 전송함.
+#^	;Qh2;	Qh95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QhGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)가 입력(0)/출력(1) 포트로 설정된 상태를 읽어서 호스트로 전송함.
+#^	;Qg0;	Qg31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QgBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
+#^	;Qg1;	Qg63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QgED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
+#^	;Qg2;	Qg95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QgGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.

명령어의 분류 (초기화 명령어)

적용 CPU	명령어	return value	operation
+#^%	;QE55A;	QE55A;	모터1/2의 순간목표/명령/현재위치, 엔코더카운터값 등을 초기값으로 재설정함.
+#^	;QE55A,HLHL,HLHL;	QE55A;	
+#^%	;QE5A5;	QE5A5;	모터1의 순간목표/명령/현재위치, 엔코더카운터값 등을 초기값으로 재설정함.
+#^	;QE5A5,HLHL;	QE5A5;	
+#^%	;QE5A5A;	QE5A5A;	모터2의 순간목표/명령/현재위치, 엔코더카운터값 등을 초기값으로 재설정함.
+#^	;QE5A5A,HLHL;	QE5A5A;	
+#^	;QE5AA5;	QE5AA5;	모터1/2의 현재 순간목표위치값을 PC 명령에서 사용하는 가상원점으로 설정함.
+#^	;QEAA55;	QEAA55;	모터1의 현재 순간목표위치값을 PC 명령에서 사용하는 가상원점으로 설정함.
+#^	;QE55AA;	QE55AA;	모터2의 현재 순간목표위치값을 PC 명령에서 사용하는 가상원점으로 설정함.
+#^	;QEi;	QEi 1, 2;	모터1/2에 대하여 설정된 가상원점의 5,000,000에 대한 차이값을 읽어서 host로 전송함.
+#^	;RQq;	RQHLHLHLHL;	1ms 단위로 CAN 통신을 통하여 동기위치제어를 할때 모터1의 현재의 위치값을 position1_ref 변수에 저장함.
+#^	;RQ?;	RQHLHLHLHL;	
+#^	;RRr;	RRHLHLHLHL;	1ms 단위로 CAN 통신을 통하여 동기위치제어를 할때 모터2의 현재의 위치값을 position2_ref 변수에 저장함.
+#^	;RR?;	RRHLHLHLHL;	
+#^	;RIA55A;	RIA55A;	모터1/2의 적분오차값을 강제로 0으로 설정함.
+#^	;RIA5A5;	RIA5A5;	모터1의 적분오차값을 강제로 0으로 설정함.
+#^	;RI5A5A;	RI5A5A;	모터2의 적분오차값을 강제로 0으로 설정함.
+#^	;RrA55A;	N.A.	모터제어기를 리셋함.

명령어의 분류 (과전압, 저전압 및 과전류 보호 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^% +#^%	;Svdddd1,dddd2,dddd3; ;Sv?;	Svdddd1,dddd2,dddd3,dddd4,dddd5; (2011년 9월 11일 이전 S/W 버전) Svdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd7; (2011년 9월 11일 S/W 버전부터)	MOSFET의 gate를 off 시키는 전압 강하량과 과전압 및 검출시간의 크기를 설정함.
+#^ +#^	;Sddddd1,dddd2,dddd3,dddd4; ;Sd?;	Sddddd1,dddd2,dddd3,dddd4;	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 위하여 가감속률을 제한하는 파라미터를 설정함.
+#^ +#^	(2013년 3월 3일 이전 S/W 버전) ;SDdddd1,dddd2; ;SD?;	SDdddd1,dddd2;	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 시작하는 전압변동량을 설정함.
+#^ +#^ +#^	(2013년 3월 3일부터 TMS320F2811용, 2014년 2월 1일부터 TMS320F28334용 S/W 버전) ;SDdddd1,dddd2; 또는 ;SDdddd1,dddd2,dddd3,dddd4; ;SD?;	SDdddd1,dddd2,dddd3,dddd4;	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 시작하는 전압변동량과 전압제어게인을 설정함.
+#^ +#^	(2014년 2월 13일 이전 S/W 버전) ;ZPdddd1; ;ZP?;	ZPdddd1;	역기전력 해소용 전력저항의 최대 ON time을 설정함.
+#^ +#^ +#^	(2014년 2월 13일 S/W 버전부터) ;ZPdddd1; 또는 ;ZPdddd1,dddd2,dddd3; ;ZP?;	ZPdddd1,dddd2,dddd3; (2014년 2월 13일 S/W 버전부터)	역기전력 해소용 전력저항의 최대 ON time을 설정함. 2014년 2월 13일 S/W 버전 부터는 brake 저항 작동을 테스트하는 기능도 겸함.

명령어의 분류 (과전압, 저전압 및 과전류 보호 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^%	;SI dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8;	SI dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8;	구동회로와 기구를 보호하기 위한 logic에서 사용하는 전류리밋값과 time threshold를 설정함.
+#^%	;SI?; 2013년 4월 4일 S/W 버전부터	2013년 4월 4일 S/W 버전부터는 Si 명령어를 사용하여 0.01A 단위로 전류값을 설정함.	
+#^	;Si dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8; 또는 ;Si?; 2013년 4월 4일 S/W 버전부터는 SI 명령어로 하던 설정을 Si 명령어를 사용하여 0.01A 단위의 전류값으로 설정할수있음.		
+#^%	;SW dddd1, dddd2;	SW dddd1, dddd2;	과전류시에 HW적으로 회로를 차단하기 위한 과전류 설정값을 설정함.
+#^%	;SW?;		
+#^%	(2012년 10월 03일 이전 S/W 버전) ;Si dddd1, dddd2;	Si dddd1, dddd2;	RJM_VER1과 RJM_VER2의 경우에는 과전류에러 fault 진단을 위한 1.024초 동안의 과전류검출 횟수를 설정하며, 그외의 경우는 SI 명령어로 설정한 전류치보다 실제전류가 연속적으로 초과하기를 허용하는 최대시간을 설정함.
+#^%	;Si?;		
+#^	(2012년 10월 03일 S/W 버전 부터) ;Si dddd1, dddd2; 또는 ;Si dddd1, dddd2, dddd3, dddd4; ;Si?; 2013년 4월 4일 S/W 버전부터 Wi 명령어로 바뀜.	Si dddd1, dddd2, dddd3, dddd4; 2013년 4월 4일 S/W 버전부터 Wi 명령어로 바뀜.	
+#^	;Wi dddd1, dddd2; 또는 ;Wi dddd1, dddd2, dddd3, dddd4; 또는 ;Wi?; 2013년 4월 4일 S/W 버전부터 Si 명령대신으로 사용됨.		

명령어의 분류 (과전압, 저전압 및 과전류 보호 명령어 - 3)

적용 CPU	명령어	return value	operation
+#^%	;Sxddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8;	Sxddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8;	제어오차가 과도하게 큰 상태를 진단하기 위한 위치오차, 속도오차, 전류오차, 허용시간을 설정함. (RJM_VER5에는 해당없음)
+#^%	;Sx?; 2013년 4월 4일 S/W 버전부터 MX/Mx 명령어로 바뀜.	2013년 4월 4일 S/W 버전부터 MX/Mx 명령어로 바뀜.	
+#^	;MXddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8; 또는 ;Mxddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8;		2013년 4월 4일 S/W 버전부터 MX 명령은 Sx 명령대신으로 사용되며, 전류값을 0.01A 단위로 설정할수 있는 Mx 명령은 추가되었음.
+#^	;MX?; 또는 ;Mx?;		

명령어의 분류 (과온도 보호 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^	;ZTdddd1,dddd2; ;ZT?;	ZTdddd1,dddd2;	모터1과 모터2의 최대허용 권선온도상승값을 설정함.
+#^ +#^	;ZQdddd1,dddd2,dddd3,dddd4, dddd5,dddd6,dddd7,dddd8; ;ZQ?;	ZQdddd1,dddd2,dddd3,dddd4,dddd5,dddd6, dddd7,dddd8;	모터1의 특성파라미터(저항, 인덕턴스, Kt, Kb, Rth1, Rth2, Tau1, Tau2)를 설정함.
+#^ +#^	;ZRdddd1,dddd2,dddd3,dddd4, dddd5,dddd6,dddd7,dddd8; ;ZR?;	ZRdddd1,dddd2,dddd3,dddd4,dddd5,dddd6, dddd7,dddd8;	모터2의 특성파라미터(저항, 인덕턴스, Kt, Kb, Rth1, Rth2, Tau1, Tau2)를 설정함.
+#^ +#^	;Zcdddd1,dddd2; ;Zc?;	Zcdddd1,dddd2;	IRMCK201을 사용한 RJM_VER3에 있어서 모터1과 모터2의 nominal current(IRMCK201 파라미터 설정에 사용한 값)를 설정함.
+#^ +#^	;MTffff1,ffff2,ffff3; ;MT?;	MTffff1,ffff2,ffff3;	온도측정용 센서 파라미터를 설정함.
+#^ +#^	;Mtdddd1; ;Mt?;	Mtdddd1,dddd2,iiii3;	온도측정용 센서를 사용하여 측정된 저항치 및 온도값을 읽어내어 host로 전송함.
+#^	;QW;	QWiiii0,iiii1,iiii2,...,iiii5,iiii6,ii ii7;	모터1과 모터2의 input power, output power, output torque, efficiency를 읽어서 host로 전송함.
+#^	;Qw;	Qwiiii0,iiii1,iiii2,...,iiii5,iiii6,ii ii7;	모터1과 모터2의 loss power, winding temperature, housing temperature, ambient temperature를 읽어서 host로 전송함.

명령어의 분류 (fault상태 모니터 명령어)

적용 CPU	명령어	return value	operation
+#^%	;Q2;	Q2 HLHL,HLHL;	fault_status, fault1_status를 읽어서 host로 전송함.
+#^%	;Q3;	Q3 HLHL; (2013년 2월 05일 이전 S/W 버전) Q3 HLHL,HLHL; (2013년 2월 05일 S/W 버전부터)	IRMCK201의 fault_status, fault1_status를 읽어서 host로 전송함.
+#^	;Qv;	QvHLHL,HLHL;	현재의 공급전압과 전압강하의 크기를 읽어서 host로 전송함.
+#^	;QJ;	QJHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	faultBufPointer, faultBuf[0-7]을 읽어서 host로 전송함.
+#^	;Qj;	QjHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	faultBufPointer_save, faultBuf_save[0-14]을 읽어서 host로 전송함.
+#^	;QjA55A;	QjHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	faultBufPointer_save, faultBuf_save[0-14]을 0으로 clear함.
+#^	;RJI;	RJn<AAAAAAAAAAAA>;	fault code를 저장한 RAM buffer를 읽어서 fault의 내용을 문자로 보여주는 index를 가장 최근의 fault code를 지정하도록 초기화함.
+#^	;RJ;	RJn<AAAAAAAAAAAA>;	fault code를 저장한 RAM buffer를 읽어서 fault의 내용을 문자로 보여줌.
+#^	;RJR;	RJR;	fault code를 저장한 RAM buffer를 지움.
+#^	;RjI;	Rjnold<AAAAAAAAAAAA>; 또는 Rjnnew<AAAAAAAAAAAA>;	EEPROM에 저장된 fault code를 읽어서 fault의 내용을 문자로 보여주는 index를 가장 최근의 fault code를 지정하도록 초기화함.
+#^	;Rj;	Rjnold<AAAAAAAAAAAA>; 또는 Rjnnew<AAAAAAAAAAAA>;	EEPROM에 저장된 fault code를 읽어서 fault의 내용을 문자로 보여줌.
+#^	;RjR;	RjR;	EEPROM에 저장된 fault code를 지움. (단 EsA55A;를 실행하여야 지운값이 저장됨)

명령어의 분류 (제어작동상태 모니터 명령어)

적용 CPU	명령어	return value	operation
+#^%	;QP;		
+#^	;QPC; (2014년 3월 21일 S/W 버전부터)	QPHLHLHLHL,HLHLHLHL;	현재 위치값을 읽어서 host로 전송함.
+#^%	;Qp;		
+#^	;Qpc; (2014년 3월 21일 S/W 버전부터)	QpHLHLHLHL,HLHLHLHL;	순간목표위치값을 읽어서 host로 전송함.
+#^	;Qb9;	QbHBLBHBLBHBLBHBLBCS; (통상적인 리턴값)	모터제어기에서 모터1/2의 순간 위치명령값과 현재 위치값의 원점으로부터의 변화량을 읽어서 host로 전송함.
+#^	;Qn9; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL; (통상적인 리턴값) SX 명령어 세번째 파라미터의 LSB가 0인 경우는 QbHBLBHBLBCS; (2013년 11월 22일 S/W 버전부터) QnHLHL,HLHL; (2013년 11월 22일 S/W 버전부터)	
+#^	;Qq;	QqHLHLHLHL,HLHLHLHL;	option으로 부착된 엔코더카운트값을 읽어서 host로 전송함.
+#^%	;QV;	QVIIII1,IIII2;	현재 속도값을 읽어서 host로 전송함.
+#^%	(Force/Moment 센서가 아닌 경우) ;Qe;	QeIIII1,IIII2,IIII3,IIII4,IIII5,IIII6;	error, error 적분값, error 미분값을 읽어서 host로 전송함.
+#^%	;QF;	QFiiii0,iiii1,iiii2,...,iiii5,iiii6,iiii7;	filter 출력값을 읽어서 host로 전송함.
+#^%	;Q1;	Q1 HLHL;	operation_status를 읽어서 host로 전송함.
+#^%	;Q4;	Q4 HLHL,HLHL,HLHL,HLHL;	프로그램 작성 날짜, version, macro 설정을 읽어서 host로 전송함.
+#^	;Q5;	Q5 HLHL;	operation1_status를 읽어서 host로 전송함.

명령어의 분류 (제어작동상태 변수 모니터 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^	;Q6;	Q6 HLHL,HLHL; (2011년 9월 25일 이전 S/W 버전) Q6 HLHL,HLHL,HLHL,HLHL; (2011년 9월 25일 S/W 버전부터)	conditional compiling 조건을 읽어서 host로 전송함.
+#^	;Q7;	Q7 HLHL; (2013년 8월 08일 S/W 버전부터)	motor_status_word 를 읽어서 host로 전송함.
+#^% +#^%	;Qxddd1; ;Qx?;	Qxddd1;	1msec 또는 2ms 마다 프로그램에서 설정된 4word 또는 8word를 host PC로 전송하는 데이터의 종류를 선택하는 변수인 data_logging_select 변수값을 설정함.
+#^%	;Qxy;	없음	1msec 또는 2ms 마다 프로그램에서 설정된 4word 또는 8word를 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#^	;Qxyz;	없음	Qxy; 또는 Qz; 또는 Qzx; 또는 Qzy; 명령에 의한 모니터링 작동을 중지함.
+#^% +#^%	;QZHLHL,HLHL,HLHL,HLHL; ;QZ;	QZHLHL,HLHL,HLHL,HLHL;	Qz 명령어로 1ms 마다 값을 host로 전송하는 4개의 변수, 또는 Qy 명령어로 host로 전송하는 8개 변수중에서 앞쪽 4개의 변수에 대한 주소를 설정함.
+#^%	;Qz;	없음	QZ 명령어로 주소가 지정된 4word를 1msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#^	;Qzx;	없음	QZ 명령어로 주소가 지정된 4word를 2msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#^	;Qzy;	없음	QZ 명령어로 주소가 지정된 8word를 2msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.

명령어의 분류 (제어작동상태 변수 모니터 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^ +#^	;QYHLHL,HLHL,HLHL,HLHL; ;QY;	QYHLHL,HLHL,HLHL,HLHL;	Qy 명령어로 host로 전송하는 8개 변수중에서 뒷쪽 4개의 변수에 대한 주소를 설정함.
+#^	;Qy;	(RS232/485의 경우) QyHBLBHBLBHBLBHBLBHBLBHBLBHBLBHBLB; (CAN의 경우) HBLBHBLBHBLBHBLBHBLBHBLBHBLBHBLB	QZ,QY 명령어로 주소가 지정된 8word를 binary 값의 형태로 host PC에 전송함.

명령어의 분류 (제어작동상태 레지스터 및 메모리 읽기/쓰기 명령어)

적용 CPU	명령어	return value	operation
+#^%	;QRHLHL,HL;	QRHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 DSP SRAM 내용을 읽어서 host로 전송함.
+#^	;QQHLHL,HL;	QQHLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 DSP SRAM 내용을 읽어서 host로 전송함.
+#^%	;QTHLHL,HL,HLHL(,HLHL);	QTHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 데이터를 DSP의 SRAM에 전송함.
+#^	;QUHLHL; ;QUIHLHL; ;QULHLHL; ;QUuHLHL; ;QUUHLHL; ;QUhHLHL; ;QUHHLHL;	QUHLHL,ffff1; QUIHLHL,iiii1; QULHLHL,IIII1; QUuHLHL,dddd1; QUUHLHL,ddddddd1; QUhHLHL,HLHL; QUHHLHL,HLHLHLHL;	지정된 주소의 DSP SRAM에서 실수/정수값을 읽어서 host로 전송함.
+#^	;QuHLHL,ffff1; ;QUIHLHL,iiii1; ;QULHLHL,IIII1; ;QUuHLHL,dddd1; ;QUUHLHL,ddddddd1; ;QUhHLHL,HLHL; ;QUHHLHL,HLHLHLHL;	QuHLHL,ffff1; QUIHLHL,iiii1; QULHLHL,IIII1; QUuHLHL,dddd1; QUUHLHL,ddddddd1; QUhHLHL,HLHL; QUHHLHL,HLHLHLHL;	지정된 주소의 DSP SRAM에 실수/정수값을 저장함. 관성센서에서는 Qu가 아니고 QW를 사용함.
+#^%	;QrHL,HL;	QrHL,HL,HLHL;	BLDC 모터제어용 chip인 IRMCK201 chip에 있는 register의 지정된 주소로부터 register 값을 읽어서 host로 전송함. (BLDC 구동 chip IRMCK201 전용)
+#^%	;QtHL,HL,HL;	QtHL,HL,HL;	BLDC 모터제어용 chip인 IRMCK201 chip에 있는 register의 지정된 주소에 host로부터 전송된 데이터를 저장함과 동시에 DSP의 SARAM 0x0a3e-0x0abd 번지에도 저장함. (BLDC 구동 chip IRMCK201 전용)
+#^%	;QtHL,HL,HLHL;	QtHL,HL,HLHL;	
+#^%	;QtHL,HL?;	QtHL,HL,HLHL;	
+#^	;Mgffff1;	Mgffff1;	엔코더값을 analog 센서에 의한 위치값으로 바꿀때의 scale factor를 설정함.

명령어의 분류 (전류측정 명령어)

적용 CPU	명령어	return value	operation
	+#^% ;QC;	QCiiii1,iiii2; QCiiii1,iiii2,iiii3,iiii4; (2상 STEP형 만 해당됨) QCiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (3상 STEP형 만 해당됨) QCiiii1,iiii2,iiii3,iiii4,iiii5; (5상 STEP형 만 해당됨)	모터에 흐르는 filtering된 상전류값을 읽어서 host로 전송함.
	+#^% ;Qc;	Qciiii1,iiii2; Qciiii1,iiii2,iiii3,iiii4; (2상 STEP형 만 해당됨) Qciiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (3상 STEP형 만 해당됨) Qciiii1,iiii2,iiii3,iiii4,iiii5; (5상 STEP형 만 해당됨)	모터에 흐르는 filtering된 상전류값을 읽어서 0.01A 단위의 암페어값으로 환산하여 host로 전송함.
	+#^ ;QK;	QKiiii1,iiii2;	모터에 흐르는 filtering된 총전류값을 읽어서 host로 전송함.
	+#^ ;Qk;	Qkiiii1,iiii2;	모터에 흐르는 filtering된 총전류값을 읽어서 암페어값으로 환산하여 host로 전송함.
	+#^ ;MC; +#^ ;MC?;	MCiiii1,iiii2,iiii3,iiii4;	BLDC 또는 AC servo 모터의 Iq, Id 값을 읽어서 host로 전송함.
	+#^ ;Mc; +#^ ;Mc?;	Mciiii1,iiii2,iiii3,iiii4;	BLDC 또는 AC servo 모터의 Iq, Id 값을 읽어서 0.01A 단위의 암페어값으로 환산하여 host로 전송함.
	+#^ (F/T센서가 아닌 경우로서 2013년 4월 07일 S/W 버전부터) ;MS?;	MSffff1,ffff2;	내부에서 사용하는 전류값을 0.01A 단위로 변환하는 scale factor을 읽어서 호스트로 전송함.

명령어의 분류 (test 구동 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^ +#^	;Tsddd1,ddd2; ;Ts?;	Tsddd1,ddd2;	모터1의 step 구동을 위한 주기와 진폭을 설정하고 step 구동을 실행하도록 함.
+#^ +#^	;Ttddd1,ddd2; ;Tt?;	Ttddd1,ddd2;	모터2의 step 구동을 위한 주기와 진폭을 설정하고 step 구동을 실행하도록 함.
+#^ +#^	;Tuddd1,ddd2; ;Tu?;	Tuddd1,ddd2;	모터1의 사인파 구동을 위한 주기와 진폭을 설정하고 사인파 구동을 실행하도록 함.
+#^ +#^	;Tvddd1,ddd2; ;Tv?;	Tvddd1,ddd2;	모터2의 사인파 구동을 위한 주기와 진폭을 설정하고 사인파 구동을 실행하도록 함.
+#^ +#^	;T0ddd1,ddd2; ;T0?;	T0ddd1,ddd2;	07번 전류제어모드의 Ts/Tt/Tu/Tv 명령에 의한 test 구동시에 변위의 중심위치가 한쪽으로 흐르는 것을 방지하기 위한 전류 offset을 설정함.
+#^ +#^	;S1ddd1,ddd2; ;S1?	S1ddd1,ddd2;	0E/0F 위치제어모드에서 모터1의 sine파 변위 주기 및 진폭을 설정함.
+#^ +#^	;S2ddd1,ddd2; ;S2?	S2ddd1,ddd2;	0E/0F 위치제어모드에서 모터2의 sine파 변위 주기 및 진폭을 설정함.
+#^% +#^%	(RJM_VER5/6 STEP형이 아닌 경우) ;SHiiii1,iiii2; ;SH?; 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함.	SHiiii1,iiii2; 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함.	host direct electrical angle 제어 모드에서 구동전류 (RJM_VER7형의 3상/5상 STEP의 경우로서 작동모드 09번이 아닐때) 또는 구동전압의 진폭 (BLDC 구동 chip IRMCK201 경우) 또는 PWM duty (DC 모터를 제외한 기타의 경우)를 설정함.
+#^	(RJM_VER5/6 STEP형이 아닌 경우) ;Shiiii1,iiii2; 또는 ;Sh?; 2013년 4월 4일 S/W 버전부터는 SH 명령어로 하던 설정을 Sh 명령을 사용하여 0.01A 단위의 전류값으로 설정할수있음.		

명령어의 분류 (test 구동 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^ +#^	;WXiiii1,iiii2; ;WX?;	WXiiii1,iiii2;	RJM_VER7 모터의 electrical angle을 강제로 설정함.
+#^%	;Qdddd1;	Qdddd1;	Sleep_1micro(dddd1), Sleep_10micro(dddd1), Sleep_msec(1) 함수실행결과의 지연시간동안 CS_EEPROM_N_port에 low pulse를 출력함.

명령어의 분류 (위치센서튜닝 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^	;WKdddd1; ;WK?;	WKdddd1,dddd2;	encoder/digital_Hall_IC_only type BLDC 모터의 자동 파라미터 설정을 개시함.
+#^ +#^	;WCdddd1,dddd2; ;WC?;	WCdddd1,dddd2;	analog Hall sensor type의 RJM_VER7 모터에 대하여 위상 각 측정용 파라미터의 설정을 개시함.
+#^ +#^ +#^	;Wkdddd1,dddd2; (2011년 6월 14일 이전 S/W 버전) ;Wkdddd1,dddd2,dddd3,dddd4,dddd5; (2011년 6월 14일 S/W 버전부터) ;Wk?;	Wkdddd1,dddd2; (2011년 6월 14일 이전 S/W 버전) Wkdddd1,dddd2,dddd3,dddd4,dddd5; (2011년 6월 14일 S/W 버전부터)	RJM_VER7 BLDC 모터제어기에서, 모드8의 host direct angle control을 사용하여 자동튜닝 등의 구동을 할 때, 사용하는 PWM duty 값을 설정함.
+#^ +#^	;WGdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; ;WG?;	WGdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;	analog Hall sensor type의 RJM_VER7 모터1의 센서 scale 과 offset을 설정함.
+#^ +#^	;WHdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; ;WH?;	WHdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;	analog Hall sensor type의 RJM_VER7 모터2의 센서 scale 과 offset을 설정함.
# #	;WNdddd1,dddd2; ;WN?;	WNdddd1,dddd2;	RJM_VER7 analog Hall sensor type에서 구동가능한 영역에서의 analog Hall sensor wave 총갯수를 설정함..
+#^% +#^%	;SZdddd1,dddd2; ;SZ?;	SZdddd1,dddd2;	Hall sensor pulse의 지정된 edge를 기준으로 하여 엔코더 Z펄스의 위치를 앤코더펄스 갯수로 읽어 냄. (BLDC 구동 chip IRMCK201 전용)

명령어의 분류 (위치센서튜닝 명령어 - analog type 센서 사용 BLDC모터의 자동 파라미터 설정)

적용 CPU	명령어	return value	operation
+#^ +#^	(제일메티컬의 경우) ;WJdddd1,dddd2,dddd3,dddd4,dddd5; ;WJ?;	WJdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;	analog Hall sensor type BLDC 모터의 linear interpolation 방식 자동 파라미터 설정을 개시함.
#^ #^	(나노모션의 linear Hall sensor의 경우) ;WJdddd1; ;WJ?;	WJdddd1,dddd2;	analog Hall sensor type BLDC 모터의 linear interpolation 방식 자동 파라미터 설정을 개시함.
+#^ +#^	(그 외의 경우) ;WJdddd1,dddd2,dddd3,dddd4,dddd5; ;WJ?;	WJdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;	analog Hall sensor type BLDC 모터의 linear interpolation 방식 자동 파라미터 설정을 개시함.
+#^ +#^	;WLdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd7; ;WL?dddd1;	WLdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd7;	3상 analog Hall 센서를 사용하는 경우에 A/B/C 상신호의 교차점값을 설정함.
+#^	;WLPdddd1;	WLP;	3상 analog Hall 센서를 사용하는 경우에 지정된 삼상 wave sector 번호부터 A/B/C 상신호의 교차점값을 자동 설정함.
+#^	;WLQ;	WLQ;	3상 analog Hall 센서를 사용하는 자석가동자의 경우에 -방향쪽 센서와 +방향쪽 센서의 위상차를 자동 설정함.
+#^	;WLR;	WLR;	3상 analog Hall 센서를 사용하는 자석가동자의 경우에 순위가 높은 제어기의 +방향쪽 센서와 순위가 낮은 현재 제어기의 -방향쪽 센서 사이의 위상차를 자동 설정함.

명령어의 분류 (위치센서튜닝 명령어 - resolver 및 삼상 analog Hall 센서)

적용 CPU	명령어	return value	operation
+#^	;A0ddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	A0ddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	모터1에 사용되는 resolver (cos-sin analog type) 또는 삼상 analog Hall 센서의 offset/gain/phase를 보정하는 파라미터를 설정함.
+#^	;A0?;		
+#^	;Aoddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	Aoddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	모터2에 사용되는 resolver (cos-sin analog type) 또는 삼상 analog Hall 센서의 offset/gain/phase를 보정하는 파라미터를 설정함.
+#^	;Ao?;		

명령어의 분류 (text 입력/저장과 LCD 표시장치 관련 명령어)

적용 CPU	명령어	return value	operation
+#^	;XTAAAAA.....AAAA;	XTAAAAA.....AAAA;	text의 현재 포인터 위치에 text를 저장하며 저장된 text를 보여줌.
+#^	;XT?dddd1;	XTdddd1,AAAAA.....AAAA;	text의 포인터 위치를 강제 설정하고 해당 위치에 있는 text 한 줄을 보여주고, 포인터 위치는 강제 설정된 값을 유지함.
+#^	;XT?;	XTAAAAA.....AAAA;	text의 현재 포인터 위치에 있는 text 한 줄을 보여주고, 포인터 위치는 변경하지 않음.
+#^	;XT+;	XTAAAAA.....AAAA;	text의 현재 포인터 위치에 있는 text 한 줄을 보여주고, 포인터 위치를 다음 줄 위치로 이동함.
+#^	;XTp;	XTpdddd1;	현재의 text 포인터 위치값을 보여줌.
+#^	;XTsA55A;	XTsA55A;	메모리상의 text를 EEPROM으로 저장함.
+#^	;DZHLHL,HLHL,HLHL,HLHL;	DZHLHL,HLHL,HLHL,HLHL;	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 값의 주소를 설정함.
+#^	;DZ?;		
+#^	;DFHLHL,HLHL,HLHL,HLHL;	DFHLHL,HLHL,HLHL,HLHL;	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 값의 type과 표시형식을 설정함.
+#^	;DF?;		
+#^	;DSffff1,ffff2,ffff3,ffff4;	DSffff1,ffff2,ffff3,ffff4;	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 값의 scale factor를 설정함.
+#^	;DS?;		
+#^	;DTdddd1,AAAAAAAAA;	DTdddd1,AAAAAAAAA;	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시되는 값의 타이틀을 설정함.
+#^	;DT?dddd1;		

명령어의 분류 (데모작동 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^ +#^	;DDddd1,ddd2,ddd3; or ;DDddd1,ddd2,ddd3,ddd4; ;DD?;	DDddd1,ddd2,ddd3; or DDddd1,ddd2,ddd3,ddd4;	일정스텝으로 이동을 반복하는 데모작동의 파라미터를 설정함.
+#^ +#^	;DEHLHL; ;DE?;	DEHLHL;	일정스텝으로 이동을 반복하는 데모작동의 작동모드를 설정함.

명령어의 분류 (특정 목적의 기능관련 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^ +#^	;WMHLHL; ;WM?;	WMHLHL;	Wafer 회전위치제어시에 위치모드로 정지한 한 상태에서 속도모드로 작동모드를 변경하며, 이에 따른 변수초기화 작동을 실시함.
+#^ +#^	(Wafer control의 경우로서 2012년 2월 18일 S/W 버전부터) ;Shddd1,ddd2,ddd3,ddd4; ;Sh?; 2013년 4월 4일 S/W 버전부터 SK/Sk 명령어로 바뀜.	Shddd1,ddd2,ddd3,ddd4; 2013년 4월 4일 S/W 버전부터 SK/Sk 명령어로 바뀜.	BLDC모터에서 holding torque를 만드는 파라미터를 설정함.
+#^	(Wafer control의 경우) ;SKddd1,ddd2,ddd3,ddd4; 또는 ;Skddd1,ddd2,ddd3,ddd4; 또는 ;SK?; 또는 ;Sk?; 2013년 4월 4일 S/W 버전부터 SK 명령어는 Sh 명령어대신으로 사용되며, 전류값을 0.01A 단위로 설정할수 있는 Sk 명령어는 추가되었음.		
# #	;MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8,ddd9; ;MB?;	MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8,ddd9;	brake actuator의 동작관련 파라미터를 설정함.
# #	;Mbddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7; ;Mb?;	Mbddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	brake actuator의 동작관련 파라미터를 설정함.
# #	;MEddd1,ddd2,ddd3,ddd4; ;ME?;	MEddd1,ddd2,ddd3,ddd4;	brake actuator의 동작관련 파라미터를 설정함.
+# +#	;MBiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7; ;MB?;	MBiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7;	EPS의 동작관련 파라미터를 설정함.

명령어의 분류 (특정 목적의 기능관련 명령어 - 2)

적용 CPU	명령어	return value	operation
#	;MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;	자석가동자 제어기의 동작관련 파라미터를 설정함.
#	;MB?;		
#	;Mbddd1,11112;	Mbddd1,11112;	자석가동자 제어기의 동작관련 파라미터를 설정함.
#	;Mb?;		

명령어의 분류 (F/T 센서 명령어 - 1)

적용 CPU	명령어	return value	operation
+#^%	;Qb;	QbHBLBHBLBHBLBHBLBCLCS; (6축 F/T 센서의 경우는 4 byte가 추가됨)	channel 8-11(F2809는 0-3)의 A/D 변환값을 읽어서 host로 전송함. (6축 F/T 센서의 경우는 2채널이 추가됨)
+#^	;Qn; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)	
+#^	;Qb1;	QbHBLBHBLBHBLBHBLBCLCS; (6축 F/T 센서의 경우는 4 byte가 추가됨)	Butterworth filter #8-#11의 필터링한 값과 공급전압의 크기를 필터링한 값을 읽어서 host로 전송함. (6축 F/T 센서의 경우는 2채널이 추가됨)
+#^	;Qn1; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)	
+#^	;Qb2;	QbHBLBHBLBHBLBHBLBCLCS; (6축 F/T 센서의 경우는 4 byte가 추가됨)	F/T sensor에서 bridge amplifier #1-#4의 offset 값과 공급전압의 크기를 필터링한 값을 읽어서 host로 전송함. (6축 F/T 센서의 경우는 2채널이 추가됨)
+#^	;Qn2; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)	
+#^	;Qb3;	QbHBLBHBLBHBLBHBLBCLCS; (6축 F/T 센서의 경우는 4 byte가 추가됨)	F/T sensor에서 계산된 모멘트값과 힘값을 읽어서 host로 전송함.
+#^	;Qn3; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)	
+#^	;Qb4;	QbHBLBHBLBHBLBHBLBCLCS; (6축 F/T 센서의 경우는 4 byte가 추가됨)	F/T sensor의 D/A 출력 전압(bridge amplifier의 영점 조정용)을 A/D 변환한 값을 읽어서 host로 전송함.
+#^	;Qn4; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)	

명령어의 분류 (F/T 센서 명령어 - 2)

적용 CPU	명령어	return value	operation
+#^	;Qb5;	QbHBLBHBLBHBLBHBLBCS; (6축 F/T 센서의 경우는 4 byte가 추가됨)	F/T sensor에서 모멘트값과 힘값을 계산한 후, 추가 필터링한 값을 읽어서 host로 전송함.
+#^	;Qn5; (2013년 3월 26일 S/W 버전부터)	QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)	
+#^	(Force/Moment 센서인 경우) ;Qe;	Qeiiii0,iiii1,iiii2,iiii3; or Qeiiii0,iiii1,iiii2,iiii3,iiii4,iiii5; (6축 F/T 센서의 경우)	X-moment, Y-moment, Z-force, extra load cell값에 대하여 추가 필터링한 값을 읽어서 host로 전송함. (6축의 경우는 X-moment, Y-moment, Z-force, X-force, Y-force, Z-moment 임)
+#^	;Qf;	Qfiiii0,iiii1,iiii2,iiii3; or Qfiiii0,iiii1,iiii2,iiii3,iiii4,iiii5; (6축 F/T 센서의 경우)	X-moment, Y-moment, Z-force, extra load cell를 읽어서 host로 전송함. (6축의 경우는 X-moment, Y-moment, Z-force, X-force, Y-force, Z-moment 임)
+#^%	;b;	RS485의 경우 : bHbMbLbHbMbLbHbMbLbHbMbLbCS; CAN의 경우 : bHbMbLbHbMbLbHbMbLbHbMbLb; (6축 F/T 센서의 경우는 3 byte가 추가됨)	Butterworth filter 8-11 (6축 F/T 센서의 경우는 8-13)의 필터링된 값을 읽어서 host로 전송함.
+#^	(Force/Moment 센서인 경우) ;SIA55A,dddd1,dddd2;	SIdddd1,dddd2;	F/T 센서에서 AD channel 8/9/10/11(내장형의 경우는 0/1/2/3) 전압의 평균값을 설정하기 위한 filtering을 실시하고, 온도값과 함께 force_data_offset 변수값을 설정함.
+#^	;SI?;		
+#^	(Force/Moment 센서인 경우) ;SI5AA5,dddd1,dddd2;	SIdddd1,dddd2;	F/T 센서에서 AD channel 8/9/10/11(내장형의 경우는 0/1/2/3) 전압의 평균값을 설정하기 위한 filtering을 실시하고, 온도증가에 따른 전압상승의 기울기를 설정함.
+#^	;SI?;		

명령어의 분류 (F/T 센서 명령어 - 3)

적용 CPU	명령어	return value	operation
+#^ +#^	(Force/Moment 센서인 경우) ;SIA5A5,0,0; ;SI?;	SIdddd1,dddd2;	F/T 센서에서 no load 상태일때 AD channel 8/9/10/11(내장형의 경우는 0/1/2/3)의 변환값이 2048이 되도록 D/A 변환기를 조절하여 만듦. (F/T센서의 앰프 offset을 2048로 맞추어주는 작동이며 총 5초가 걸림)
+#^ +#^	(F/T센서의 경우) ;MOdddd1,dddd2,dddd3,dddd4; ;MO?;	MOdddd1,dddd2,dddd3,dddd4;	F/T 센서에서 TLV5630 또는 DAC124S085 D/A converter의 출력값을 설정함.
+#^ +#^	;MTffff1,ffff2,ffff3; ;MT?;	MTffff1,ffff2,ffff3;	온도측정용 센서 파라미터를 설정함.
+#^ +#^	;Mtdddd1; ;Mt?;	Mtdddd1,dddd2,iiii3;	온도측정용 센서를 사용하여 측정된 저항치 및 온도값을 읽어내어 host로 전송함.
+^ +^	;MUffff1,ffff2,ffff3; ;MU?;	MUffff1,ffff2,ffff3;	온도측정용 두번째 센서 파라미터를 설정함.
+^ +^	;Mudddd1; ;Mu?;	Mudddd1,dddd2,iiii3;	온도측정용 두번째 센서를 사용하여 측정된 저항치 및 온도값을 읽어내어 host로 전송함.
+#^ +#^	(F/T센서의 경우) ;MPffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12; ;MP?;	MPffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12;	4개의 strain guage 신호를 증폭하고 AD 변환된 값을 이용하여 X-moment, Y-moment, Z-force를 계산하는 변환행렬 값을 설정함.
+#^	;Qa1;	Qiiii0;	load_cell 값을 읽어서 host로 전송함.

명령어의 분류 (F/T 센서 명령어 - 4)

적용 CPU	명령어	return value	operation
+#^ +#^	(F/T센서의 경우) ;Mpffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12; ;Mp;	Mpffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12;	6개의 strain guage 신호를 증폭하고 AD 변환된 값을 이용하여 X-moment, Y-moment, Z-force, X-force를 계산하는 변환행렬 값을 설정함.
+#^ +#^	(F/T센서의 경우) ;Mqffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12; ;Mq;	Mqffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12;	6개의 strain guage 신호를 증폭하고 AD 변환된 값을 이용하여 Y-force, Z-moment를 계산하는 변환행렬 값을 설정함.
+#^ +#^	(F/T센서의 경우) ;MSiiii1,iiii2,iiii3,iiii4; ;MS?;	MSiiii1,iiii2,iiii3,iiii4;	F/T 센서에서 증폭후 A/D 변환된 스트레인 값의 온도증가에 따른 변화율을 설정함.
+#^ +#^	(F/T센서의 경우) ;MViiii1,iiii2; ;MV?;	MViiii1,iiii2;	F/T 센서에서 온도보상의 기준온도와 온도보상 기울기를 자동 계산할 때 사용한 또 다른 온도값을 설정함.
+#^ +#^	(F/T센서의 경우) ;Moiiii1,iiii2,iiii3,iiii4,iiii5; ;Mo?;	Moiiii1,iiii2,iiii3,iiii4,iiii5;	F/T 센서에서 수직력의 auto zero rate, maximum allowed drift, X 모멘트 coupling, Y 모멘트 coupling, FORCE_SENSOR_ERROR_DETECTION_TIME 파라미터를 설정함.
+^ +^	(F/T센서의 경우) ;Msddd1,ddd2; ;Ms?;	Msddd1,ddd2;	삼성메디컬용 F/T 센서의 X-moment, Z-force coupling을 처리하는 파라미터를 설정함.
+^ +^	(F/T센서의 경우) ;MBddd1,ddd2; ;MB?;	MBddd1,ddd2;	F/T 센서 drift offset 제거 동작관련 파라미터를 설정함.

명령어의 분류 (사용하지 않는 명령어)

적용 CPU	명령어	return value	operation
+#^ +#^	;WBddd1,ddd2; ;WB?;	WBddd1,ddd2;	RJM_VER7 제어시에 100KHz interrupt routine에서 사용하는 Butterworth filter의 시정수를 설정함.
% %	;Scddd1,ddd2; ;Sc?;	Scddd1,ddd2;	전류치를 암페어로 환산하는 scale factor를 설정함.
% %	;PWiiii1,iiii2; ;PW?;	PWiiii1,iiii2;	모터의 PWM duty 출력을 강제로 설정함.
% %	;SViiii1,iiii2; ;SV?;	SViiii1,iiii2;	속도제어 모드에서 속도명령값을 설정함.

명령어의 분류 (TMS320F28016 레이저스캐너+, 줌카메라#에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+#	;QS;	QSHLHL;	on/off 형식의 센서 상태를 읽어서 host로 전송함.
+#	;QA;	QAHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	하위 8 channel의 12-bit AD 변환값을 읽어서 host로 전송함.
+#	;QB;	QBHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	상위 8 channel의 12-bit AD 변환값을 읽어서 host로 전송함.
+#	;Qxdddd1;	Qxdddd1;	1msec 마다 프로그램에서 설정된 4word를 host PC로 전송하는 데이터의 종류를 선택하는 변수인 data_logging_select 변수값을 설정함.
+#	;Qx?;		
+#	;Qxy;	없음	1msec 마다 프로그램에서 설정된 4word를 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#	;QZHLHL,HLHL,HLHL,HLHL;	QZHLHL,HLHL,HLHL,HLHL;	Qz 명령어로 1ms 마다 값을 host로 전송하는 4개의 변수, 또는 Qy 명령어로 host로 전송하는 8개 변수중에서 앞쪽 4개의 변수에 대한 주소를 설정함.
+#	;QZ;		
+#	;Qz;	없음	QZ 명령어로 주소가 지정된 4word를 1msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#	;Q1;	Q1 HLHL;	operation_status를 읽어서 host로 전송함.
+#	;Q2;	Q2 HLHL,HLHL;	fault_status, fault1_status를 읽어서 host로 전송함.
+#	;Q4;	Q4 HLHL,HLHL,HLHL,HLHL;	프로그램 작성 날짜, version, macro 설정을 읽어서 host로 전송함.
+#	;QRHLHL,HL;	QRHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 DSP SRAM 내용을 읽어서 host로 전송함.
+#	;QTHLHL,HL,HLHL(,HLHL);	QTHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 데이터를 DSP의 SRAM에 전송함.

명령어의 분류 (TMS320F28016 레이저스캐너+, 줌카메라#에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +	;PI 11111,11112; ;PI?;	PI 11111,11112;	위치 명령값의 증분을 설정하고 이동을 명령함.
+ +	;P1 11111; ;P1?;	P1 11111;	모터1에 대하여 위치 명령값의 증분을 설정하고 이동을 명령함.
+ +	;P2 11111; ;P2?;	P2 11111;	모터2에 대하여 위치 명령값의 증분을 설정하고 이동을 명령함.
+ +	;PA 11111,11112; ;PA?;	PA 11111,11112;	모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함.
+ +	;Pa 11111; ;Pa?;	Pa 11111;	모터1의 위치명령값을 설정함. (레이저스캐너에서만 적용됨)
+ +	;PW 11111,11112,11113; ;PW?;	PW 11111,11112,11113;	모터1의 PWM duty 출력을 강제로 설정함.
# #	;PP 11111,11112; ;PP?;	PP 11111,11112;	줌위치와 초점위치를 설정하고 이동을 명령함. (줌카메라에서만 적용됨)

명령어의 분류 (TMS320F28016 레이저스캐너+, 줌카메라#에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +#	:SAHLHL; :SA?;	SAHLHL;	모터제어기의 주소를 설정함.
+ +#	:SMHLHL; :SM?;	SMHLHL;	위치, 속도, 전류제어 등의 작동모드를 설정함.
+ +#	:SXHLHL; :SX?;	SXHLHL;	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+ +	:SQddd1,ddd2,ddd3; :SQ?;	SQddd1,ddd2,ddd3;	모터1의 P, I, D 제어파라미터를 설정함.
# #	:SfHLHL; :Sf?;	SfHLHL,HLHL;	자화전자의 AF 카메라의 초점구동 명령을 보내고, 자화전자의 AF 카메라로부터의 값을 host로 전송함. (줌카메라에서만 적용됨)
# #	:Siddd1; :Si?;	Siddd1;	IRIS 명령을 설정함. (줌카메라에서만 적용됨)
# #	:Snddd1; :Sn?;	SnHLHL;	자화전자의 AF 카메라의 ND 필터 및 셔터의 구동명령을 보내고, 상태값을 host로 전송함. (줌카메라에서만 적용됨)
# #	:SVIIII1,IIII2; :SV?;	SVIIII1,IIII2;	속도제어 모드에서 속도명령값을 설정함. (줌카메라에서만 적용됨)
+ +	:SViiii1; :SV?;	SViiii1;	속도제어 모드에서 속도명령값을 설정함. (레이저스캐너에서만 적용됨)
+ +#	:Sadddd1,ddd2,ddd3,ddd4; :Sa?;	Sadddd1,ddd2,ddd3,ddd4;	속도제어 및 위치제어 모드에서 가속 및 감속율을 설정함.
+ +#	:SSddd1,ddd2; :SS?;	SSddd1,ddd2;	위치제어모드에서 Point-to-Point 이동시 이동속도값을 설정함.

명령어의 분류 (TMS320F28016 레이저스캐너+, 줌카메라#에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +#	:Ssdddd1,dddd2; :Ss?;	Ssdddd1,dddd2;	PA, PB, Pa 명령에 의한 위치구동시의 가속/감속 기간을 설정함.
# #	:SHiiii1,iiii2; :SH?;	SHiiii1,iiii2;	스텝모터의 구동전류의 진폭을 설정함. (줌카메라에서만 적용됨)
+ +	:SHiiii1; :SH?;	SHiiii1;	01번 제어 모드에서 구동전압의 진폭을 설정함. (레이저스캐너에서만 적용됨)
+ +#	:Shdddd1,dddd2,dddd3,dddd4; :Sh?;	Shdddd1,dddd2,dddd3,dddd4;	스테핑모터에서 전류명령값의 최대치와 정지시의 전류치 비율을 설정함. (줌카메라에서만 적용됨)
+ +	:SDdddd1; :SD?;	SDdddd1;	속도에 따른 damping을 만들기 위한 계인을 설정함. (레이저스캐너에서만 적용됨)
+ +	:Stdddd1,dddd2,dddd3,dddd4; :St?;	Stdddd1,dddd2,dddd3,dddd4;	position1_cmd_step_max, position1_cmd_increment, position1_cmd_change_time, laser_ON_time, laser_OFF_time을 설정함. (레이저스캐너에서만 적용됨)
+ +#	:SFdddd1,dddd2,dddd3; :SF?dddd1;	SFdddd1,dddd2,dddd3;	butterworth filter의 시정수와 변화율 상한값을 설정함.
# #	:SPdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd7,dddd8; :SP?dddd1;	SPdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd7,dddd8;	줌카메라의 줌설정위치번호, 줌설정위치번호에서의 줌위치값, 줌설정위치번호에서의 6개 초점위치번호에서의 초점위치값을 설정함. (줌카메라에서만 적용됨)
+ +#	:SmHLHL; :Sm?;	SmHLHL;	명령어에 대한 return값을 보낼 때 사용하는 host의 주소를 설정함.

명령어의 분류 (TMS320F28016 레이저스캐너+, 줌카메라#에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +#	;Zbddd1; ;Zb?;	Zbddd1;	CAN 통신속도를 설정함.
+ +#	;ZZ; ;ZZ?;	ZZddd1;	흡위치를 찾는 일을 중단함. (줌카메라에서만 적용됨)
+ +	;WEddd1; ;WE?;	WEddd1;	BLDC 모터1에 있어서 Hall_A상 상승에지에 대한 U상 전압 positive zero crossing 포인트의 위상지연 값을 설정함.
+ +	;WXiiii1; ;WX?;	WXiiii1;	RJM_VER7 모터의 electrical angle을 강제로 설정함.
+ +#	;ERHLHL,HLHL;	ERHLHL,HLHL;	파라미터 값을 EEPROM으로부터 읽어내어 DSP SRAM에 저장함. (EEPROM으로부터 읽은 값을 0x8200 - 0x87FF 번지에 저장함.)
+ +#	;ESHLHL,HLHL;	ESHLHL,HLHL;	DSP SRAM에 있는 파라미터 값을 page단위(8 word 또는 16 byte)로 EEPROM에 저장함. (0x8200 - 0x87FF 번지의 값을 EEPROM에 저장함.)
+ +#	;EsA55A;	EsA55A;	DSP SRAM에 있는 작동 파라미터 값을 한꺼번에 EEPROM에 저장함.
+ +#	;EDA55A;	EDA55A;	DSP SRAM에 있는 파라미터 값을 default 값으로 바꿈. 이때 ST 명령어로 선정된 motor type에 따라서 파라미터의 기본값이 설정됨.
+ +#	;EC?;	ECHLHL,HLHL,HLHL;	RS232, RS485, CAN 통신과정에서 어떤 에러가 발생하였는지를 읽어서 host로 전송함.
+ +#	;ECR;	ECR;	RS232, RS485, CAN 통신에러 플래그를 없애고 에러를 복구함.

Qxddd1 에 의하여 전송되는 변수값 (TMS320F28016 레이저스캐너+인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
1 +	위치값 (position1) -8192 ~ 8192 범위의 값임.	속도값 (speed1_RPM) 단위 크기는 RPM	0.1ms 동안의 위치값증분의 filtering 값 (vfiltA1) 단위 크기는 A/D 1bit	위상각 ((phase_angle1 * 36000L) >> 14) 단위 크기는 0.01 도
2 +	전압명령값 (voltage1_cmd)	U상 구동 PWM duty 값 (motor1_PWM1_set)	V상 구동 PWM duty 값 (motor1_PWM2_set)	W상 구동 PWM duty 값 (motor1_PWM3_set)
7 +	구동전압의 위상값 (electrical_angle1_cnt >> 8)	Hall 센서의 A/D 변환값 (ADCRESULT0_reg >> 4)	(ADCRESULT7_reg >> 4)	(ADCRESULT15_reg >> 4)
8 +	위치값 (position1) -8192 ~ 8192 범위의 값임.	위상각 ((phase_angle1 * 36000L) >> 14) 단위 크기는 0.01 도	구동전압의 위상값 (electrical_angle1_cnt >> 8)	전압명령값 (voltage1_cmd)
9 +	영상의 VSYNC 신호와 timer를 결합한 값 (VSYNC_IN_port * -1000 + VSYNC_timer) -8192 ~ 8192 범위의 값임.	위치명령값 (position1_cmd) -8192 ~ 8192 범위의 값임.	위치값 (position1) -8192 ~ 8192 범위의 값임.	레이저 ON/OFF 출력값 (LASER_OUT_port * 2000)
20 +	작동모드 (operation_mode)	mainLoopStep_cnt	mainLoop_execution_time 단위 크기는 1%	maximum_interrupt_service_time 단위 크기는 1%

명령어의 분류 (C8051F555_noRobitiz+, C8051F561#, C8051F555_Robitiz^ RC 서보제어기에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+#^	;QS;	QSHLHL;	on/off 형식의 상태를 읽어서 host로 전송함.
+#	;QA;	QAHLHL,HLHL,HLHL;	potentiometer, current, voltage의 12-bit A/D 변환값을 읽어서 host로 전송함.
+#	;QC;	QCHLHL,HLHL,HLHL;	current_cmd, current A/D 변환값, current A/D 변환값을 필터링한 값을 읽어서 host로 전송함.
+#	;Qxddd1;	Qxddd1;	1msec 마다 프로그램에서 설정된 4word를 host PC로 전송하는 데이터의 종류를 선택하는 변수인 data_logging_select 변수값을 설정함.
+#	;Qx?;		
+#	;Qxy;	없음	1msec 마다 프로그램에서 설정된 4word를 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#	;QZHLHL,HLHL,HLHL,HLHL;	QZHLHL,HLHL,HLHL,HLHL;	Qz 명령어로 1ms 마다 값을 host로 전송하는 4개의 변수에 대한 주소를 설정함.
+#	;QZ;		
+#	;Qz;	없음	QZ 명령어로 주소가 지정된 4word를 1msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+#	;Q1;	Q1 HLHL;	operation_status를 읽어서 host로 전송함.
+#^	;Q4;	Q4 HLHL,HLHL,HLHL,HLHL;	프로그램 작성 날짜, version, macro 설정을 읽어서 host로 전송함.
#	;QRHLHL,HL;	QRHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 SRAM 내용을 읽어서 host로 전송함.
#	;QrHLHL,HL;	QrHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 flash ROM 내용을 읽어서 host로 전송함.
#	;QTHLHL,HL,HLHL(,HLHL);	QTHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 데이터를 SRAM에 전송함.

명령어의 분류 (C8051F555_noRobitiz+, C8051F561#, C8051F555_Robitiz^ RC 서보제어기에서 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +	;PAiiii1; ;PA?;	PAiiii1; or PAdddd1;	위치명령값을 설정함.
+ +	;pAiiii1; ;pA?;	pAiiii1; or pAdddd1;	위치명령값을 설정함.
+ +	;Paiiii1; ;Pa?;	Paiiii1; or PAdddd1;	순간위치명령값을 설정함.
+ +	;paiiii1; ;pa?;	paiiii1; or padddd1;	순간위치명령값을 설정함.
+ +	;Ppiiii1; ;Pp?;	Ppiiii1; or Ppddd1;	St 명령어로 설정된 시간동안 이동할 위치명령값을 설정함.
+ +	;ppiiii1; ;pp?;	ppiiii1; or ppddd1;	St 명령어로 설정된 시간동안 이동할 위치명령값을 설정함.
+ +	;PWiiii1; ;PW?;	PWiiii1;	PWM duty 출력을 강제로 설정함.
+#^	;FrA55A;	FrA55A;	flash ROM에 있는 파라미터를 SRAM으로 읽어냄.
+#^	;FeA55A;	FeA55A;	flash ROM에 있는 파라미터를 모두 지움.
+#^	;FsA55A;	FsA55A;	SRAM에 있는 파라미터를 flash ROM으로 저장함.
+#^	;FdA55A;	FdA55A;	SRAM에 있는 파라미터를 기본값으로 설정함.

명령어의 분류 (C8051F555_noRobitiz+, C8051F561#, C8051F555_Robitiz^ RC 서보제어기에서 사용하는 명령어)

+#^ +#^	:SAHLHL; :SA?;	SAHLHL;	모터제어기의 주소를 설정함.
+# +#	:SMHLHL; :SM?;	SMHLHL;	위치, 전류제어 등의 작동모드를 설정함.
+#^ +#^	:SXHLHL; :SX?;	SXHLHL;	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+# +#	:SCiiii1; :SC?;	SCiiii1;	전류명령값을 설정함.
+#^ +#^	:SLiiii1,iiii2; :SL?;	SLiiii1,iiii2; or SLdddd1,dddd2;	위치 한계값을 설정함.
+#^ +#^	:S0dddd1; :S0?;	S0dddd1;	위치 offset 값을 설정함.
+#^	:SoA;	Sodddd1;	전류 offset 값을 자동으로 설정함.
+#^ +#^	:Sodddd1; :So?;	Sodddd1;	전류 offset 값을 수동으로 설정함.
+#^ +#^	:SDdddd1; :SD?;	SDdddd1;	속도 damping 계인을 설정함.
+#^ +#^	:Sddddd1; :Sd?;	Sdddd1;	위치오차의 미분값 계산을 위한 Δt 를 설정함.
+#^ +#^	:Stdddd1; :St?;	Stdddd1;	Pp 또는 pp 명령으로 설정된 위치명령값까지 이동하는데 걸리는 시간을 설정함.
+#^ +#^	:SSdddd1; :SS?;	SSdddd1;	PA 또는 pA 명령으로 설정된 위치명령값까지 이동하는 동안의 속도를 설정함.
+#^ +#^	:SVdddd1; :SV?;	SVdddd1;	PWM duty 크기를 제한하기 시작하는 공급전압의 threshold 값

명령어의 분류 (C8051F555_noRobitiz+, C8051F561#, C8051F555_Robitiz^ RC 서보제어기에서 사용하는 명령어)

+#^ +#^	;SQddd1,ddd2,ddd3; ;SQ?;	SQddd1,ddd2,ddd3;	P, I, D 게인값을 설정함.
+#^ +#^	;Swddd1; ;Sw?;	Swddd1;	제어에 사용하는 전류의 최대치를 설정함.
+#^ +#^	;SRddd1; ;SR?;	SRddd1;	Robotiz protocol을 사용하는 경우 Status_Return_Level 값을 설정함.
+#^ +#^	;SWiiii1; ;SW?;	SWiiii1;	방향에 따라서 같은 PWM duty에 대하여 출력의 크기가 다른 경우에 출력의 크기가 같게 되도록 보정하는 factor를 설정함.
+#^ +#^	;SFddd1; ;SF?;	SFddd1;	포텐시오미터마다 scale factor가 다른 경우에 이를 보정하기 위한 factor를 설정함.
+#^ +#^	;SBddd1; ;SB?;	SBddd1;	통신속도를 설정함.

PC에서 로봇운동제어기를 통하여 모터제어기(Robitiz protocol 사용)로 명령을 보낼때

Dr01,SA?; 명령을 로봇운동제어기로 보내면, 01번 모터제이기로부터 SAFE01; 값이 리턴됨.

Dr01,So2048; 명령을 로봇운동제어기로 보내면, 01번 모터제이기로부터 So2048; 값이 리턴되면서 전류 offset 값이 2048로 설정됨.

원가를 설정하기 전에는 반드시 명령어+?;를 사용하여 현재값을 확인하고 나서, 필요시에만 바꾸도록 명령을 보낼 것.

설정된 파라미터를 저장하고 싶으면 Dr01,FsA55A; 명령을 로봇운동제어기로 보내면 됨.

Qxddd1 에 의하여 전송되는 변수값 (C8051F561#인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0 +#^	순간위치명령값 (position1_set) -8192 ~ 8192 범위의 값임.	position1_at_1ms -8192 ~ 8192 범위의 값임.	speed1_RPM_10 단위 크기는 0.1 RPM	current1_cmd 전류감지저항으로 5m0hm을 사용하는 경우는 2048일때 10A 이며, 0.10hm을 사용하는 경우는 2048 일때 2.5A임.
2 +#^	전류명령값 (current1_cmd) 전류감지저항으로 5m0hm을 사용하는 경우는 2048일때 10A 이며, 0.10hm을 사용하는 경우는 2048 일때 2.5A임.	권선전류의 A/D 변환값 (current1_AD) 전류감지저항으로 5m0hm을 사용하는 경우는 2048일때 10A 이며, 0.10hm을 사용하는 경우는 2048 일때 2.5A임.	ADCRESULT[1] 전류감지저항으로 5m0hm을 사용하는 경우는 2048일때 10A 이며, 0.10hm을 사용하는 경우는 2048 일때 2.5A임.	motor1_PWM_set -2048 ~ 2048 범위의 값임.
3 #	순간위치명령값 (position1_set) -8192 ~ 8192 범위의 값임.	position1_at_1ms -8192 ~ 8192 범위의 값임.	position1_by_potentiometer -8192 ~ 8192 범위의 값임.	position_encoder 펄스단위의 값으로서 Robotiz EX106의 경우 128 pulse/rev임.
4 #	순간위치명령값 (position1_set) -8192 ~ 8192 범위의 값임.	position1_at_1ms -8192 ~ 8192 범위의 값임.	position1_by_potentiometer -8192 ~ 8192 범위의 값임.	position1_error -8192 ~ 8192 범위의 값임.
5 #	순간위치명령값 (position1_set) -8192 ~ 8192 범위의 값임.	position1_at_1ms -8192 ~ 8192 범위의 값임.	position1_by_potentiometer -8192 ~ 8192 범위의 값임.	position1_adjustment -8192 ~ 8192 범위의 값임.
20 +#^	공급전압값 (vfiltA1) 149 이면 1V에 해당함.	PORT_STATUS bit0 : MOT_EN_port ^ 1 bit1 : MOT_FAULT_port ^ 1 bit7 : AD_error_flg	mainLoop_execution_time 단위 크기는 1%	maximum_interrupt_service_time 단위 크기는 1%
21 +#^	motor1_error -8192 ~ 8192 범위의 값임.	motor1_integral	motor1_derivative	current1_cmd 전류감지저항으로 5m0hm을 사용하는 경우는 2048일때 10A 이며, 0.10hm을 사용하는 경우는 2048 일때 2.5A임.

명령어의 분류 (Robotiz protocol을 사용하는 경우 처리하는 Robotiz 명령어)

byte2	byte3	byte4	byte5	기능
0xfe or =address	byte3부터 총 바이트 갯수	0x83	0x1e	sync write 기능을 수행함.
		0x03	0x03	ID 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x04	Baud rate 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x06	CW/CCW angle_limit 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x08	CCW angle_limit 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x0e	처리하지 않음
		0x03	0x10	Status_Return_Level 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x18	torque enable 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x1e	goal position 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x20	moving speed 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x03	0x22	torque_limit 를 write함. Status_Return_Level!=2이면 설정값의 return을 실행함.
		0x02	0x00	YEAR_MONTH 를 return 함.
		0x02	0x02	MAJORMACRO 를 return 함.
		0x02	0x03	ID 를 return 함.
		0x02	0x04	Baud rate 를 return 함.
		0x02	0x06	CW/CCW angle_limit 를 return 함.
		0x02	0x08	CCW angle_limit 를 return 함.
		0x02	0x0e	처리하지 않음
		0x02	0x10	Status_Return_Level 를 return 함.
		0x02	0x18	torque enable 를 return 함.
		0x02	0x1e	goal position 를 return 함.
		0x02	0x20	moving speed 를 return 함.
		0x02	0x22	torque_limit 를 return 함.
		0x02	0x24	present position [(max-min)/(1024 or 4096 or 16384 bit)] 를 return 함.
		0x02	0x26	present speed [114RPM/1023bit] 를 return 함.
		0x02	0x28	present current [5mOhm : 10A/1023bit, 0.1ohm : 2.5A/1023bit] with bias 를 return 함.
		0x02	0x2a	present voltage [0.1V/bit] 를 return 함.
		0x99	byte5부터 byte3-2 개수만큼의 데이터를 Robocube 제어기 명령어로 인식하여 처리함.	

명령어의 분류 (C8051F555_noRobitiz+, C8051F561#, C8051F555_Robitiz^ RC 서보제어기에서 사용하는 기본값)

모터	사용자	파라미터	비고
RC ROBOROBO 2011-8-23	+		C8051F555 V1.3
RC RX28 2011-8-23	^	SL+08191,-08192;S002150;So02060;SD00250;Sd00001;St00050;SS00050;SV00014;SQ00002,0000,00020;Sw01229;SR00002;SW+00256;SF18000;	C8051F555 V1.2
RC RX64 2011-8-23	#	SL+08191,-08192;S002150;So02090;SD00100;Sd00002;St00050;SS00050;SV00014;SQ00001,0000,00005;Sw00819;SR00002;SW+00256;SF18220;	C8051F561 V1.2, bipolar drive
RC EX106 2011-8-23	#	SL+08191,-08192;S002150;So02072;SD00200;Sd00002;St00050;SS00050;SV00014;SQ00001,0000,00002;Sw00819;SR00002;SW+00256;SF18000;	C8051F561 V1.2, bipolar drive

명령어의 분류 (전류앰프가 사용하는 명령어)

+	:QA;	QAHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	하위 8 channel의 12-bit AD 변환값을 읽어서 host로 전송함.
+	:QB;	QBHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	상위 8 channel의 12-bit AD 변환값을 읽어서 host로 전송함.
+	:QC;	QCHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;	12-bit DA 변환 출력 설정값을 읽어서 host로 전송함.
+	:Qodddd1,dddd2; :Qo?dddd1;	Qodddd1,dddd2;	선택된 12-bit DA 변환 출력값을 강제로 설정함.
+	:QH0;	QH31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QHBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)의 H/L 상태를 읽어서 호스트로 전송함.
+	:QH1;	QH63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QHED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)의 H/L 상태를 읽어서 호스트로 전송함.
+	:QH2;	QH95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QHGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)의 H/L 상태를 읽어서 호스트로 전송함.
+	:Qh0;	Qh31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QhBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)가 입력(0)/출력(1) 포트에 설정된 상태를 읽어서 호스트로 전송함.
+	:Qh1;	Qh63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QhED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)가 입력(0)/출력(1) 포트에 설정된 상태를 읽어서 호스트로 전송함.
+	:Qh2;	Qh95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QhGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)가 입력(0)/출력(1) 포트에 설정된 상태를 읽어서 호스트로 전송함.

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+	;Qg0;	Qg31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QgBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
+	;Qg1;	Qg63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QgED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
+	;Qg2;	Qg95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; QgGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
+	;QGAdddd1;	QGA dd , A , A , A ;	TMS320F2810/11/12를 사용한 모터제어기에서 개별 입출력 포트의 H/L 상태, 입출력 설정 상태, MUX 설정 상태를 읽어서 호스트로 전송함.
+	;QGBdddd1;	QGB dd , A , A , A ;	
+	;QGDdddd1;	QGD dd , A , A , A ;	
+	;QGEdddd1;	QGE dd , A , A , A ;	
+	;QGFdddd1;	QGF dd , A , A , A ;	
+	;QGGdddd1;	QGG dd , A , A , A ;	
+	;QE;	QE HLHL , HLHL , HL , HLHL , HLHL , HLHL ;	통신에러횟수를 읽어서 host로 전송함.
+	;QEE;	QEE HLHL , HLHL , HLHL , HLHL , HLHL ;	통신에러횟수를 읽어서 host로 전송함.
+	;Qxdddd1;	Qx dddd1 ;	1msec 또는 2ms 마다 프로그램에서 설정된 4word 또는 8word를 host PC로 전송하는 데이터의 종류를 선택하는 변수인 data_logging_select 변수값을 설정함.
+	;Qx?;		
+	;Qxy;	없음	1msec 또는 2ms 마다 프로그램에서 설정된 4word 또는 8word를 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+	;Qxyz;	없음	Qxy; 또는 Qz; 또는 Qzx; 또는 Qzy; 명령에 의한 모니터링 작동을 중지함.

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+	;QZHLHL,HLHL,HLHL,HLHL; ;QZ;	QZHLHL,HLHL,HLHL,HLHL;	Qz 명령어로 1ms 마다 값을 host로 전송하는 4개의 변수, 또는 Qy 명령어로 host로 전송하는 8개 변수중에서 앞쪽 4개의 변수에 대한 주소를 설정함.
+	;Qz;	없음	QZ 명령어로 주소가 지정된 4word를 1msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+	;Qzx;	없음	QZ 명령어로 주소가 지정된 4word를 2msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+	;Qzy;	없음	QZ 명령어로 주소가 지정된 8word를 2msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
+	;QYHLHL,HLHL,HLHL,HLHL; ;QY;	QYHLHL,HLHL,HLHL,HLHL;	Qy 명령어로 host로 전송하는 8개 변수중에서 뒷쪽 4개의 변수에 대한 주소를 설정함.
+	;Qy;	(RS232/485의 경우) QyHBLBHBLBHBLBHBLBHBLBHBLBHBLBHBLB; (CAN의 경우) HBLBHBLBHBLBHBLBHBLBHBLBHBLBHBLB	QZ,QY 명령어로 주소가 지정된 8word를 binary 값의 형태로 host PC에 전송함.
+	;Q1;	Q1 HLHL;	operation_status를 읽어서 host로 전송함.
+	;Q2;	Q2 HLHL,HLHL;	fault_status, fault1_status를 읽어서 host로 전송함.
+	;Q3;	Q3 HLHL; (2013년 2월 05일 이전 S/W 버전) Q3 HLHL,HLHL; (2013년 2월 05일 S/W 버전부터)	IRMCK201의 fault_status, fault1_status를 읽어서 host로 전송함.
+	;Q4;	Q4 HLHL,HLHL,HLHL,HLHL;	프로그램 작성 날짜, version, macro 설정을 읽어서 host로 전송함.

명령어의 분류 (전류애플가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+	;QRHLHL,HL;	QRHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 DSP SRAM 내용을 읽어서 host로 전송함.
+	;QQHLHL,HL;	QQHLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 DSP SRAM 내용을 읽어서 host로 전송함.
+	;QTHLHL,HL,HLHL (,HLHL);	QTHLHL,HL,HLHL(,HLHL);	지정된 주소에서 시작하여 지정된 양만큼의 데이터를 DSP의 SRAM에 전송함.
+	;QUHLHL; ;QIHLHL; ;QULHLHL; ;QUHLHL; ;QUHLHL; ;QUHLHL; ;QUHLHL; ;QUHLHL;	QUHLHL,ffff1; QIHLHL,iiii1; QULHLHL,IIII1; QUHLHL,dddd1; QUHLHL,ddddddd1; QUHLHL,HLHL; QUHLHL,HLHLHLHL;	지정된 주소의 DSP SRAM에서 실수/정수값을 읽어서 host로 전송함.
+	;QuHLHL,ffff1; ;QiHLHL,iiii1; ;QuHLHL,IIII1; ;QuHLHL,dddd1; ;QuHLHL,dddddd dd1; ;QuHLHL,HLHL; ;QuHLHL,HLHLHL HL;	QuHLHL,ffff1; QiHLHL,iiii1; QuHLHL,IIII1; QuHLHL,dddd1; QuHLHL,ddddddd1; QuHLHL,HLHL; QuHLHL,HLHLHLHL;	지정된 주소의 DSP SRAM에 실수/정수값을 저장함.

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+	;PWiiii1;	PWiiii1;	모터의 PWM duty 출력을 강제로 설정함.
+	;PW?;		
+	;PEHLHL;	PEHLHL;	선택된 보드의 모터 출력을 ON 시킴.
+	;PEA55A;		
+	;PE?;		
+	;PRHLHL;	PRHLHL;	선택된 보드의 fault flag를 clear 시킴.
+	;PRA55A;		
+	;PDHLHL;	PDHLHL;	선택된 보드의 모터 출력을 OFF 시킴.
+	;PDA55A;		
+	;PD?;		
+	;SAHLHL;	SAHLHL;	모터제어기의 주소를 설정함.
+	;SA?;		
+	;SMHLHL;	SMHLHL;	작동모드를 설정함.
+	;SM?;		
+	;SMHLHL;	SMHLHL;	작동모드를 설정함.
+	;SM?;		
+	;SXHLHLHLHL,HLHLHLHL;	SXHLHLHLHL,HLHLHLHL;	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.
+	;SX?;		
+	;Svddd1,ddd2,ddd3;	Svddd1,ddd2,ddd3,ddd4,ddd5,ddd6,d ddd7;	MOSFET의 gate를 off 시키는 전압 강하량과 과전압 및 검출시 간의 크기를 설정함.
+	;Sv?;		

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +	;SCiiii1; ;SC?;	SCiiii1;	전류제어 모드에서 전류명령값을 설정함.
+ +	;Sciiii1; ;Sc?;	Sciiii1;	전류제어 모드에서 0.01A 단위로 전류명령값을 설정함.
+ +	;SdHLHL; ;Sd?;	SdHLHL;	DA 출력 모드를 설정함.
+ +	;S1dddd1, dddd2; ;S1?	S1dddd1, dddd2;	OE/OF 위치제어모드에서 모터1의 sine파 변위 주기 및 진폭을 설정함.
+ +	;S2dddd1, dddd2; ;S2?	S2dddd1, dddd2;	OE/OF 위치제어모드에서 모터2의 sine파 변위 주기 및 진폭을 설정함.
+ + +	;SDdddd1, dddd2; 또는 ;SDdddd1, dddd2, dddd3, dddd4; ;SD?;	SDdddd1, dddd2, dddd3, dddd4;	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 시작하는 전압변동량과 전압제어게인을 설정함.
+ +	;SI dddd1, dddd2; ;SI?;	SI dddd1, dddd2;	구동회로와 기구를 보호하기 위한 logic에서 사용하는 전류리미트값과 time threshold를 설정함.
+ +	;Si dddd1, dddd2; ;Si?;	Si dddd1, dddd2;	구동회로와 기구를 보호하기 위한 logic에서 사용하는 전류리미트값을 0.01A 단위로 설정하며, time threshold를 설정함.
+ +	;Sw dddd1; ;Sw?;	Sw dddd1;	제어시 사용하는 전류의 리미트를 설정함.
+ +	;SBiiii1; ;SB?;	SBiiii1; (SB?; 인 경우에만 echo 있음)	RS232C 직렬통신속도값을 설정함.
+ +	;SF dddd1, dddd2, dddd3; ;SF? dddd1;	SF dddd1, dddd2, dddd3;	butterworth filter의 시정수와 변화율 상한값을 설정함.

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +	;Wi dddd1, dddd2; ;Wi?;	Wi dddd1, dddd2;	SI 명령어로 설정한 전류치보다 실제전류가 연속적으로 초과하기를 허용하는 최대시간을 설정함.
+	;Rr A55A;	N.A.	모터제어기를 리셋함.
+ +	;RBHLHL; ;RB?;	RBHLHL;	feedback 제어를 개시하도록 작동모드 설정에 필요한 조치를 자동으로 처리하게 함.
+	;RJl;	RJn<AAAAAAAAAAAA>;	fault code를 저장한 RAM buffer를 읽어서 fault의 내용을 문자로 보여주는 index를 가장 최근의 fault code를 지정하도록 초기화함.
+	;RJ;	RJn<AAAAAAAAAAAA>;	fault code를 저장한 RAM buffer를 읽어서 fault의 내용을 문자로 보여줌.
+	;RJr;	RJR;	fault code를 저장한 RAM buffer를 지움.
+	;Rjl;	RjnoId<AAAAAAAAAAAA>; 또는 Rjnnew<AAAAAAAAAAAA>;	EEPROM에 저장된 fault code를 읽어서 fault의 내용을 문자로 보여주는 index를 가장 최근의 fault code를 지정하도록 초기화함.
+	;Rj;	RjnoId<AAAAAAAAAAAA>; 또는 Rjnnew<AAAAAAAAAAAA>;	EEPROM에 저장된 fault code를 읽어서 fault의 내용을 문자로 보여줌.
+	;RjR;	RjR;	EEPROM에 저장된 fault code를 지움. (단 EsA55A;를 실행하여야 지운값이 저장됨)
+ +	;RCddd1, dddd2; ;RC?;	RCddd1, dddd2, dddd3, dddd4;	CAN 이나 RS485 통신이 일정시간 두절되었을 때를 감지하는 time threshold 파라미터를 설정함. 2013년 10월 23일 S/W 버전부터 사용가능한 명령임.
+ +	;Rcddd1; ;Rc?;	Rcddd1;	CAN 이나 RS485 통신에서 idle mode로 진입하거나 또는 정상통신모드로 진입하는 것을 설정함. 2013년 11월 05일 S/W 버전부터 사용가능한 명령임.

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+	;Z0A55A, iiii1 ; ~ ;ZFA55A, iiii1 ; (단 ;Z3A55A, HLHL ; ;Z4A55A, HLHL ; ;Z5A55A, HLHL ; ;Z6A55A, HLHL ; ;Z7A55A, HLHL ; ;Z8A55A, HLHL ; ;Z9A55A, HLHL ; ;ZAA55A, HLHL ; ;ZBA55A, HLHL ; ;ZCA55A, HLHL ; 는 예외)	ZO(~F) HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL ; HLHL, HLHL, HLHL, HLHL ; 16개의 FACTORY SETTING parameter 값	0 ~ F 까지의 숫자로 지정된 위치의 FACTORY SETTING parameter RAM에 값을 저장함.
+	;Zr;	Zr HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL, HLHL ;	16개의 FACTORY SETTING parameter 값을 읽어서 보여줌 (RAM 영역의 값).
+	;ZsA55A;	ZsA55A;	메모리상의 FACTORY SETTING parameter 를 EEPROM으로 저장함.
+	;Zp?;	Zp dddd1, dddd2 ;	PWM 주파수와 모터 1/2의 전류제어 주파수, 속도 및 위치제어 주파수를 읽어서 호스트로 전송함.
+	;ZP dddd1 ; 또는		
+	;ZP dddd1, dddd2, dddd3 ;	ZP dddd1, dddd2, dddd3 ;	역기전력 해소용 전력저항의 최대 ON time을 설정함. brake
+	;ZP?;		저항 작동을 테스트하는 기능도 겸함.
+	;Zb dddd1 ;	Zb dddd1 ;	CAN 통신속도를 설정함.
+	;Zb?;		

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+ +	;AwA55A, dddd1; ;Aw?;	Awdddd1;	사용가능한 최대크기의 PWM duty 비를 설정함.
+ +	;Abddd1, dddd2; ;Ab?;	Abddd1, dddd2;	전류제어의 PI 계인을 설정함.
+ +	;Ayddd1, dddd2; ;Ay?;	Ayddd1, dddd2;	외부로부터 인가되는 전류명령 입력의 offset과 scale factor를 설정함.
+ +	;Axddd1, llll2, dddd3, dddd4; ;Ax?;	Axddd1, llll2, dddd3, dddd4;	총전류 측정과 전압 측정에 사용하는 offset과 scale factor를 설정함.
+ +	;ADA55A, iiii1, iiii2; ;AD?;	ADiiii1, iiii2, iiii3;	MOTOR 구동 Half-bridge의 dead-time을 보상하되, Z6 명령어로 설정하는 dead-time에 추가하여 보상하려는 보상량을 설정함.
+ +	;MS?;	MSffff1;	내부에서 사용하는 전류값을 0.01A 단위로 변환하는 scale factor을 읽어서 호스트로 전송함.
+ +	;MRddd1, dddd2; ;MR?;	MRddd1, dddd2;	전류상승률을 제한하는 파라미터를 설정함.
+ +	;Mrddd1, dddd2; ;Mr?;	Mrddd1, dddd2;	전류상승률을 제한하는 파라미터를 0.01A 단위로 설정함.
+ +	;Mxddd1, dddd2; ;Mx?;	Mxddd1, dddd2;	제어오차가 과도하게 큰 상태를 진단하기 위한 전류오차, 허용 시간을 설정함.
+ +	;MTffff1, ffff2, ffff3; ;MT?;	MTffff1, ffff2, ffff3;	온도측정용 센서 파라미터를 설정함.
+ +	;Mtddd1; ;Mt?;	Mtddd1, dddd2, iiii3;	온도측정용 센서를 사용하여 측정된 저항치 및 온도값을 읽어내어 host로 전송함.
+ +	;MFddd1, dddd2; ;MF?;	MFddd1, dddd2;	팬을 작동시키는 온도와 히스테리시스를 설정함.

명령어의 분류 (전류앰프가 사용하는 명령어)

적용 CPU	명령어	return value	operation
+	;Tsddd1,ddd2;	Tsddd1,ddd2;	코일의 전류에 대하여 step 구동 위한 주기와 진폭을 설정하고 step 구동을 실행하도록 함.
+	;Ts?;		
+	;Tuddd1,ddd2;	Tuddd1,ddd2;	코일의 전류에 대하여 사인파 구동 위한 주기와 진폭을 설정하고 사인파 구동을 실행하도록 함.
+	;Tu?;		
+	;ERHLHL,HLHL;	ERHLHL,HLHL;	파라미터 값을 EEPROM으로부터 읽어내어 DSP SARAM에 저장함.
+	;ESHLHL,HLHL;	ESHLHL,HLHL;	DSP SRAM에 있는 파라미터 값을 page단위(8 word 또는 16 byte)로 EEPROM에 저장함.
+	;EDA55A;	EDA55A;	DSP SRAM에 있는 파라미터 값을 default 값으로 바꿈. 이때 ST 명령어로 선정된 motor type에 따라서 파라미터의 기본값이 설정됨.
+	;EsA55A;	EsA55A;	DSP SRAM에 있는 작동 파라미터 값을 한꺼번에 EEPROM에 저장함.
+	;EC?;	ECHLHL,HLHL,HLHL;	RS232, RS485, CAN 통신과정에서 어떤 에러가 발생하였는지를 읽어서 host로 전송함.
+	;ECR;	ECR;	RS232, RS485, CAN 통신에러 플래그를 없애고 에러를 복구함.
+	;EcS;	EcS;	RS485 또는 CAN을 통하여 모터제어기 주소 254(0xFE)로 전송되어오는 broadcasting 데이터를 수신하도록 설정함.
+	;EcR;	EcR;	RS485 또는 CAN을 통하여 모터제어기 주소 254(0xFE)로 전송되어오는 broadcasting 데이터를 수신하지 못하도록 설정함.
+	;EUHLHL,HLHL,HLHL;	EUHLHL,HLHL,HLHL;	CAN 통신을 사용할 때, 일정주기로 상위제어기에 제어기 상태값을 전송하는 기능의 파라미터를 설정함.
+	;EU?;		
+#^	;EuHLHL,HLHL,HLHL,HLHL;	EuHLHL,HLHL,HLHL,HLHL;	CAN 통신을 사용하여 일정주기로 상위제어기에 모터1의 상태값을 전송할 때, 전송할 모터1의 상태값 주소를 설정함.
+#^	;Eu?;		
+#^	;EvHLHL,HLHL,HLHL,HLHL;	EvHLHL,HLHL,HLHL,HLHL;	CAN 통신을 사용하여 일정주기로 상위제어기에 모터2의 상태값을 전송할 때, 전송할 모터2의 상태값 주소를 설정함.
+#^	;Ev?;		

명령어	구분	설명
;QS; (+#^%)	operation	on/off 형식의 센서 상태를 읽어서 host로 전송함.
	return value	<p>QSHL,HLHL,HLHL,HLHL; (스텝핑모터가 아닌 경우) HL is two-byte hex code of a byte. QSHL,HLHL,HLHLHLHL,HLHLHLHL; (스텝핑모터인 경우) HL is two-byte hex code of a byte. QSHL,HLHL,HLHLHLHL; (저가형 1채널 제어기인 경우) HL is two-byte hex code of a byte.</p> <p>1st byte : bit0 - 2 : low(0)/high(1) status of LS1A/B/C : bit3 - 5 : low(0)/high(1) status of LS2A/B/C : bit6 : low(0)/high(1) status of ENC1Z of 모터1 : bit7 : low(0)/high(1) status of ENC2Z of 모터2</p> <p>2nd word : bit0 : SEL_EEPROM_N_port : bit1 : MOT_EN_N_port : bit2 : RXEN_N_port : bit3 : TXEN_port : bit4 : OFF12V_port (intelligent power IC를 사용하는 일부모델은 해당없음) : bit5 : DISABLE1_IN_port (intelligent power IC를 사용하지 않는 일부모델은 해당없음) : bit6 : DISABLE2_IN_port (intelligent power IC를 사용하지 않는 일부모델은 해당없음) : bit7 : CS_IRMCK1_port (RJM_VER3 모델에만 해당됨) : bit8 : CS_IRMCK2_port (RJM_VER3 모델에만 해당됨) : bit9 : : bit10 : : bit11 : LS1D_port (일부모델은 해당없음, 모델별 입력컨넥터 포트 사양참조) : bit12 : LS2D_port (일부모델은 해당없음, 모델별 입력컨넥터 포트 사양참조) : bit13 : EXT_FAULT1_port : bit14 : EXT_FAULT2_port : bit15 :</p> <p>3rd word or long word : 모터1 encoder (4채배 pulse, 스텝핑모터와 저가형 1채널 제어기인 경우는 32-bit, 그외는 16-bit) 4th word or long word : 모터2 encoder (4채배 pulse, 스텝핑모터인 경우는 32-bit, 그외는 16-bit)</p>

명령어	구분	설명
	comment	<p>두번째 리턴값의 bit1에 해당하는 MOT_EN_N_port의 값이 0이면 모터가 enable 되어 있는 것이고, 1이면 모터가 disable 되어 있는 것임.</p> <p>두번째 리턴값의 bit5/6가 1이면 과전류 또는 과온도가 검출되었음을 의미함. (2013년 1월 16일 이전의 S/W 버전에서는 0일때가 과전류 또는 과온도가 검출되었음을 의미하는 경우도 있으므로 전원 투입직후의 값을 비교하여 판단하여야함)</p> <p>스텝핑모터의 경우는 엔코더 입력단자가 설치되어 있는 경우에 한하여 엔코더 값을 32bit로 확장한 값을 리턴함.</p> <p>스텝핑모터에서 2013년 12월 19일 이전의 S/W 버전에서는 리미트센서가 정상적으로 검출되지 않는 문제가 있음.</p> <p>1st byte의 bit0~bit5 는 SX 명령어 두번째 파라미터의 bit0~bit7의 설정에 따라서, 실제포트의 H/L 상태가 아니고 반전된 상태값이 될수 있으며 이 기능은 2012년 5월 28일 S/W 버전부터 적용됨.</p>

명령어	구분	설명
(절대각센서를 사용하지 않는 경우) ;Qs; (+#^)	operation	인공안구의 경우는 팬/틸트방향의 절대각을 읽어서 host로 전송하며, 베스트모션의 경우는 경사각 센서값을 읽어서 host로 전송함.
	return value	QsHLHL,HLHL,HLHL,HLHL; HL is two-byte hex code of a byte. HLHL는 word값을 4개의 hex code로 바꾼 값인데 high byte:low byte 순으로 표현됨. 인공안구의 경우(F2811/F28334만 해당) 1st word : 팬방향의 절대각을 14bit로 표현한 값 (bit) 2nd word : 팬방향의 360도 회전수 3rd word : 틸트방향의 절대각을 14bit로 표현한 값 (bit) 4th word : 틸트방향의 360도 회전수 베스트모션의 경우(F28334만 해당) 1st word : 경사각센서1의 경사각을 16bit로 표현한 값 (bit) 2nd word : 경사각센서1의 1bit status 값 3rd word : 경사각센서2의 경사각을 16bit로 표현한 값 (bit) 4th word : 경사각센서2의 1bit status 값
	comment	인공안구의 경우는 팬/틸트방향의 14bit 절대각을 읽은 값이며, 베스트모션의 경우 16bit의 경사각센서를 읽은 값임. 인공안구의 경우에 360도 회전수는 signed integer 임을 주의할 것.

명령어	구분	설명
(절대각센서를 사용하는 경우)	operation	절대각센서값을 읽어서 host를 전송함.
	return value	QsHLHLHLHL,HLHLHLHL; HL is two-byte hex code of a byte.
	comment	F28334로서 절대각센서를 사용하는 경우에 한함.
;Qs; (#)	operation	하위 8 channel의 12-bit A/D 변환값을 읽어서 host로 전송함.
	return value	QAHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; HL is two-byte hex code of a byte. HLHL는 word값을 4개의 hex code로 바꾼 값인데 high byte:low byte 순으로 표현됨.
	comment	각각의 값이 갖는 의미는 모델별로 다름.
;QA; (+#^%)	operation	상위 8 channel의 12-bit A/D 변환값을 읽어서 host로 전송함.
	return value	QBHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; HL is two-byte hex code of a byte. HLHL는 word값을 4개의 hex code로 바꾼 값인데 high byte:low byte 순으로 표현됨.
	comment	각각의 값이 갖는 의미는 모델별로 다름.
;QB; (+#^%)	operation	TLV2556의 12-bit A/D 변환값을 읽어서 host로 전송함.
	return value	QaHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL;
	comment	TLV2556 12-bit A/D 변환기를 가지거나 RJM_VER7에서만 실제값이 리턴됨.
;Qa; (+#^)	operation	load_cell 값을 읽어서 host로 전송함.
	return value	Qi i i i 0;
	comment	F/T 센서에서만 적용되는 명령어임. 2013년 5월 10일 S/W 버전부터 사용가능한 명령어임.
;Qa1; (+#^)	operation	load_cell 값을 읽어서 host로 전송함.
	return value	Qi i i i 0;
	comment	F/T 센서에서만 적용되는 명령어임. 2013년 5월 10일 S/W 버전부터 사용가능한 명령어임.

명령어	구분	설명
;Qb; (+#^%) ;Qn; (+#^%) (2013 년 3월 26일 S/W 버전부터)	operation	channel 8-11(F2809는 0-3)의 A/D 변환값을 읽어서 host로 전송함. (6축 F/T 센서의 경우는 2채널이 추가됨)
	return value	Qb HBLBHBLBHBLBHBL CS; (6축 F/T 센서의 경우는 4 byte가 추가됨) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 8 byte의 합 inverse를 binary값 한 바이트로 표현한 것임. Qn HLHL,HLHL,HLHL,HLHL ; Qn HLHL,HLHL,HLHL,HLHL,HLHL,HLHL ; (6축 F/T 센서의 경우) ☞ F2809가 아닌 경우: 1st word : A/D channel 8의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #1) 2nd word : A/D channel 9의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #2) 3rd word : A/D channel 10의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #3) 4th word : A/D channel 11의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #4) 5th word : A/D channel 12의 A/D값 (bit, 6축 F/T sensor의 경우 bridge amplifier #5) 6th word : A/D channel 13의 A/D값 (bit, 6축 F/T sensor의 경우 bridge amplifier #6) ☞ F2809인 경우: 1st word : A/D channel 0의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #1) 2nd word : A/D channel 1의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #2) 3rd word : A/D channel 2의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #3) 4th word : A/D channel 3의 A/D값 (bit, F/T sensor의 경우 bridge amplifier #4) 5th word : A/D channel 4의 A/D값 (bit, 6축 F/T sensor의 경우 bridge amplifier #5) 6th word : A/D channel 5의 A/D값 (bit, 6축 F/T sensor의 경우 bridge amplifier #6)
	comment	F/T 센서의 경우에 A/D 변환값은 4개(6축 F/T 센서의 경우는 6개)의 strain-guage bridge에서 증폭된 전압을 A/D 변환한 값임. 별도의 load cell을 외부에 장착하는 S/W 버전에서는 1st word 값이 별도 load cell의 값에 해당함.

명령어	구분	설명
;Qb1; (+#^) ;Qn1; (+#^) (2013년 3월 26일 S/W 버전부터)	operation	Butterworth filter #8-#11의 필터링한 값과 공급전압의 크기를 필터링한 값을 읽어서 host로 전송함. (6축 F/T 센서의 경우는 2채널이 추가됨)
	return value	QbHBLBHBLBHBLBHBLBHBLBCS; (6축 F/T 센서의 경우는 4 byte가 추가됨) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 10 byte의 합의 inverse를 binary값 한 바이트로 표현한 것임. QnHLHL,HLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우) 1st word : butterworth filter #8의 A/D 변환값을 필터링한 값 (bit) 2nd word : butterworth filter #9의 A/D 변환값을 필터링한 값 (bit) 3rd word : butterworth filter #10의 A/D 변환값을 필터링한 값 (bit) 4th word : butterworth filter #11의 A/D 변환값을 필터링한 값 (bit) 5th word : butterworth filter #12의 A/D 변환값을 필터링한 값 (bit) (6축 F/T sensor의 경우만 해당됨) 6th word : butterworth filter #13의 A/D 변환값을 필터링한 값 (bit) (6축 F/T sensor의 경우만 해당됨) 마지막 word : DC_BUS_VOLT_filtered (65 bit/V) (6축 F/T sensor가 아니면 5th word 임)
	comment	F/T 센서인 경우 butterworth filter #8-#11(6축 F/T 센서의 경우는 #8-#13)는 bridge amplifier #1-#4(6축 F/T 센서의 경우는 #1-#6)의 전압값을 필터링함. 별도의 load cell을 외부에 장착하는 S/W 버전에서는 1st word 값이 별도 load cell의 값에 해당함. 2014년 3월 22일 이전 S/W 버전에서는 Qb1; 의 리턴값중 마지막 값이 불연속적으로 튀는 현상이 있을수 있음.

명령어	구분	설명
;Qb2; (+#^) ;Qn2; (+#^) (2013년 3월 26일 S/W 버전부터)	operation	F/T sensor에서 bridge amplifier #1-#4의 offset 값과 공급전압의 크기를 필터링한 값을 읽어서 host로 전송함. (6축 F/T 센서의 경우는 2채널이 추가됨)
	return value	QbHBLBHBLBHBLBHBLBHLBCS; (6축 F/T 센서의 경우는 4 byte가 추가됨) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 10 byte의 합의 inverse를 binary값 한 바이트로 표현한 것임. QnHLHL,HLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우) 1st word : bridge amplifier #1의 offset 값 (bit) 2nd word : bridge amplifier #2의 offset 값 (bit) 3rd word : bridge amplifier #3의 offset 값 (bit) 4th word : bridge amplifier #4의 offset 값 (bit) 5th word : bridge amplifier #5의 offset 값 (bit) (6축 F/T sensor의 경우만 해당됨) 6th word : bridge amplifier #6의 offset 값 (bit) (6축 F/T sensor의 경우만 해당됨) 마지막 word : DC_BUS_VOLT_filtered (65 bit/V) (6축 F/T sensor가 아니면 5th word 임)
	comment	F/T 센서인 경우에만 적용됨. offset 값은 SI 명령어로 설정이 가능하며, 필터 시정수를 크게하여 일정시간 bridge amplifier #1-#4(6축 F/T 센서의 경우는 #1-#6))의 전압값을 filtering한 값으로 설정됨. 별도의 load cell을 외부에 장착하는 S/W 버전에서는 1st word 값이 별도 load cell의 값에 해당함. 2014년 3월 22일 이전 S/W 버전에서는 Qb2; 의 리턴값중 마지막 값이 불연속적으로 튀는 현상이 있을수 있음.

명령어	구분	설명
;Qb3; (+#^) ;Qn3; (+#^) (2013년 3월 26일 S/W 버전부터)	operation	F/T sensor에서 계산된 모멘트값과 힘값을 읽어서 host로 전송함.
	return value	QbHBLBHBLBHBLBHBLBCS; (6축 F/T 센서의 경우는 4 byte가 추가됨) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 8 byte의 합 inverse를 binary값 한 바이트로 표현한 것임. QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우) 1st word : X-moment (0.01Nm) 2nd word : Y-moment (0.01Nm) 3rd word : Z-force (0.1N) 4th word : DC_BUS_VOLT_filtered (65 bit/V) or 첫번째 bridge-amplifier A/D 변환값을 필터링하고 offset을 제거한 값 or X-force (0.1N, 6축 F/T 센서의 경우) 5th word : Y-force (0.1N, 6축 F/T 센서의 경우) 6th word : Z-moment (0.01Nm, 6축 F/T 센서의 경우)
	comment	F/T 센서인 경우에만 적용됨. 주의 : Qb3; 명령에 대한 마지막 리턴값은 첫번째 amplifier와 나머지 3 amplifier로 분리하여 사용할때, DC_BUS_VOLT_filtered/ 값이 아니고 첫번째 amplifier 출력의 A/D 변환값에서 offset을 뺀 값임. Qn3; 명령에 대한 리턴값이 FFFF이면 -1에 해당함.

명령어	구분	설명
;Qb4; (+#^) ;Qn4; (+#^) (2013년 3월 26일 S/W 버전부터)	operation	F/T sensor의 D/A 출력 전압(bridge amplifier의 영점 조정용)을 A/D 변환한 값을 읽어서 host로 전송함.
	return value	QbHBLBHBLBHBLBHBLBCS; (6축 F/T 센서의 경우는 4 byte가 추가됨) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 8 byte의 합의 inverse를 binary값 한 바이트로 표현한 것임. QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우) 1st word : bridge amplifier #1의 영점조정용 D/A 출력전압의 A/D 변환값 2nd word : bridge amplifier #2의 영점조정용 D/A 출력전압의 A/D 변환값 3rd word : bridge amplifier #3의 영점조정용 D/A 출력전압의 A/D 변환값 4th word : bridge amplifier #4의 영점조정용 D/A 출력전압의 A/D 변환값 5th word : bridge amplifier #5의 영점조정용 D/A 출력전압의 A/D 변환값 (6축 F/T 센서의 경우) 6th word : bridge amplifier #6의 영점조정용 D/A 출력전압의 A/D 변환값 (6축 F/T 센서의 경우)
	comment	F/T 센서인 경우에만 적용됨. 별도의 load cell을 외부에 장착하는 S/W 버전에서는 1st word 값이 별도 load cell의 값에 해당함.

명령어	구분	설명
;Qb5; (+#^) ;Qn5; (+#^) (2013년 3월 26일 S/W 버전부터)	operation	F/T sensor에서 모멘트값과 힘값을 계산한 후, 추가 필터링한 값을 읽어서 host로 전송함.
	return value	<p>QbHBLBHBLBHBLBHBLBCS; (6축 F/T 센서의 경우는 4 byte가 추가됨) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 8 byte의 합 inverse를 binary값 한 바이트로 표현한 것임.</p> <p>QnHLHL,HLHL,HLHL,HLHL; QnHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (6축 F/T 센서의 경우)</p> <p>1st word : X-moment의 추가필터링값 (0.01Nm) 2nd word : Y-moment의 추가필터링값 (0.01Nm) 3rd word : Z-force의 추가필터링값 (0.1N) 4th word : DC_BUS_VOLT_filtered (65 bit/V) or 첫번째 bridge-amplifier A/D 변환값을 필터링하고 offset을 제거한 후 추가필터링값 값 or X-force의 추가필터링값 (0.1N, 6축 F/T 센서의 경우) 5th word : Y-force의 추가필터링값 (0.1N, 6축 F/T 센서의 경우) 6th word : Z-moment의 추가필터링값 (0.01Nm, 6축 F/T 센서의 경우)</p>
	comment	<p>F/T 센서인 경우에만 적용됨. 추가필터의 시정수는 SF9,xxxx,xxxx; 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할수 있음. 주의 : Qb3; 명령에 대한 마지막 리턴값은 첫번째 amplifier와 나머지 3 amplifier로 분리하여 사용할때, DC_BUS_VOLT_filtered/ 값이 아니고 첫번째 amplifier 출력의 A/D 변환값에서 offset을 뺀 값임. Qn5; 명령에 대한 리턴값이 FFFF이면 -1에 해당함. 2012년 3월 9일 S/W 버전부터 사용가능한 명령임.</p>

명령어	구분	설명
;Qb9; (+#^) ;Qn9; (+#^) (2013년 3월 26일 S/W 버전부터)	operation	모터제어기에서 모터1/2의 순간 위치명령값과 현재 위치값의 원점으로부터의 변화량을 읽어서 host로 전송함.
	return value	<p>QbHBLBHBLBHBLBHLBCS; (통상적인 리턴값) HBLB 는 16bit word의 상위바이트 한개, 하위바이트 한개를 각각 binary 형태로 표현한 값. CS 는 앞선 8 byte의 합의 inverse를 binary값 한 바이트로 표현한 것임.</p> <p>QnHLHL,HLHL,HLHL,HLHL; (통상적인 리턴값) 1st word : 원점으로부터 모터1의 순간 위치명령값 변화량 2nd word : 원점으로부터 모터1의 현재 위치값 변화량 3rd word : 원점으로부터 모터2의 순간 위치명령값 변화량 4th word : 원점으로부터 모터2의 현재 위치값 변화량</p> <p>SX 명령어 세번째 파라미터의 LSB가 0인 경우는 QbHBLBHBLBCS; (2013년 11월 22일 S/W 버전부터) QnHLHL,HLHL; (2013년 11월 22일 S/W 버전부터) 로서 모터1의 값만을 리턴함.</p>
	comment	2012년 12월 10일 이전 S/W 버전에서는 리턴값중에서 4th word에 문제가 있어 사용에 주의를 바람. Qn9; 명령에 대한 리턴값이 FFFF이면 -1에 해당함.

명령어	구분	설명
;b; (+#^%)	operation	Butterworth filter 8-11(6축 F/T 센서의 경우는 8-13)의 필터링된 값을 읽어서 host로 전송함.
	return value	<p>RS485의 경우 : bHbMbLbHbMbLbHbMbLbHbMbLbCS; (6축 F/T 센서의 경우는 3 byte가 추가됨)</p> <p>CAN의 경우 : bHbMbLbHbMbLbHbMbLbHbMbLb; (6축 F/T 센서의 경우는 3 byte가 추가됨)</p> <p>HbMbLb 는 binary 형태의 12bit word를 가리키며 HbMbLbHbMbLb 로서 2개의 12bit가 모이면 8bit 단위의 3byte가 됨.</p> <p>CS 는 앞선 6 byte의 합의 inverse를 binary값 한 바이트로 표현한 것임.</p> <p>1st 12-bit : butterworth filter #8의 필터링된 값</p> <p>2nd 12-bit : butterworth filter #9의 필터링된 값</p> <p>3rd 12-bit : butterworth filter #10의 필터링된 값</p> <p>4th 12-bit : butterworth filter #11의 필터링된 값</p> <p>5th 12-bit : butterworth filter #12의 필터링된 값 (6축 F/T 센서의 경우)</p> <p>6th 12-bit : butterworth filter #13의 필터링된 값 (6축 F/T 센서의 경우)</p>
	comment	Butterworth filter의 시정수를 바꾸려면 SF8,dddd1,dddd2; 명령을 사용하면 됨.

명령어	구분	설명
;QH0; (+#^)	operation	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)의 H/L 상태를 읽어서 호스트로 전송함.
	return value	QH31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QHBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 상태가 low, 1이면 포트의 상태가 high 임.
;QH1; (+#^)	operation	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)의 H/L 상태를 읽어서 호스트로 전송함.
	return value	QH63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QHED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 상태가 low, 1이면 포트의 상태가 high 임.
;QH2; (+#^)	operation	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)의 H/L 상태를 읽어서 호스트로 전송함.
	return value	QH95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QHGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 상태가 low, 1이면 포트의 상태가 high 임. TMS320F2809를 사용한 제어기에서는 해당사항 없음.

명령어	구분	설명
;Qh0; (+#^)	operation	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)가 입력(0)/출력(1) 포트로 설정된 상태를 읽어서 호스트로 전송함.
	return value	Qh31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QhBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 설정이 입력, 1이면 포트의 설정이 출력임.
;Qh1; (+#^)	operation	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)가 입력(0)/출력(1) 포트로 설정된 상태를 읽어서 호스트로 전송함.
	return value	Qh63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QhED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 설정이 입력, 1이면 포트의 설정이 출력임.
;Qh2; (+#^)	operation	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)가 입력(0)/출력(1) 포트로 설정된 상태를 읽어서 호스트로 전송함.
	return value	Qh95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QhGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 설정이 입력, 1이면 포트의 설정이 출력임. TMS320F2809를 사용한 제어기에서는 해당사항 없음.

명령어	구분	설명
;Qg0; (+#^)	operation	입출력 포트 0-31(TMS320F2810/11/12를 사용한 모터제어기에서는 B포트, A포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
	return value	Qg31 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QgBA bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 설정이 범용입출력, 0이 아니면 포트의 설정이 주변장치목적임.
;Qg1; (+#^)	operation	입출력 포트 32-63(TMS320F2810/11/12를 사용한 모터제어기에서는 E포트, D포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
	return value	Qg63 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QgED bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 설정이 범용입출력, 0이 아니면 포트의 설정이 주변장치목적임.
;Qg2; (+#^)	operation	입출력 포트 64-95(TMS320F2810/11/12를 사용한 모터제어기에서는 G포트, F포트)의 MUX 설정 상태(0이면 범용입출력)를 읽어서 호스트로 전송함.
	return value	Qg95 bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F28334/335/09의 경우) QgGF bbbbbbbb bbbbbbbb bbbbbbbb bbbbbbbb; (TMS320F2810/11/12의 경우)
	comment	MSB에서 LSB 순으로 8비트씩 나열되어 있으며, TMS320F2810/11/12의 경우는 B포트에서 A포트 순으로 나열되었음. 0이면 포트의 설정이 범용입출력, 0이 아니면 포트의 설정이 주변장치목적임. TMS320F2809를 사용한 제어기에서는 해당사항 없음.

명령어	구분	설명
;QGAddd1;(+) ;QGBddd1;(+) ;QGDddd1;(+) ;QGEddd1;(+) ;QGFddd1;(+) ;QGGddd1;(+)	operation	TMS320F2810/11/12를 사용한 모터제어기에서 개별 입출력 포트의 H/L 상태, 입출력 설정 상태, MUX 설정 상태를 읽어서 호스트로 전송함. 1st word : 포트번호 [0 - 15]
	return value	QGA dd , A,A,A ; (A 포트그룹의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGB dd , A,A,A ; (B 포트그룹의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGD dd , A,A,A ; (D 포트그룹의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGE dd , A,A,A ; (E 포트그룹의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGF dd , A,A,A ; (F 포트그룹의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGG dd , A,A,A ; (G 포트그룹의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태)
	comment	H0이면 high, L0이면 low, I0이면 input, O0이면 output, G0이면 범용입출력, P0이면 주변장치목적임을 나타냄.
;QGddd1; (#^)	operation	TMS320F28334/335/09를 사용한 모터제어기에서 개별 입출력 포트의 H/L 상태, 입출력 설정 상태, MUX 설정 상태를 읽어서 호스트로 전송함. 1st word : 포트번호 [0 - 95] (TMS320F2809를 사용한 제어기에서는 63번 까지만 유효함)
	return value	QGA dd , A,A,A ; (A 포트그룹(0-31)의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGB dd , A,A,A ; (B 포트그룹(32-63)의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태) QGC dd , A,A,A ; (C 포트그룹(64-95)의 dd번째 포트에 대한 H/L 상태, 입출력 설정 상태, MUX 설정 상태)
	comment	H0이면 high, L0이면 low, I0이면 input, O0이면 output, G0이면 범용입출력, 1/2/3 이면 주변장치목적임을 나타냄.

명령어	구분	설명
;QE; (+#^%)	operation	통신에러횟수를 읽어서 host로 전송함.
	return value	QEHLHL, HLHL, HL, HLHL, HLHL, HLHL; 1st word : Pa_receive_error_cnt (회) 2nd word : Pp_receive_error_cnt (회) 3rd byte : PP_command_update_skip_cnt (회) 4th word : RS232C 통신에러 횟수 (회) 5th word : CAN 통신에러 횟수 (회) 6th word : RS485 통신에러 횟수 (회)
	comment	1st word는 Pa 명령어를 수신했을 때 잘못된 checksum으로 수신되어, 값을 무시하고 Pa 명령어 끝에 :으로 회신한 횟수를 가리킴. 2nd word는 Pp 명령어를 수신했을 때 잘못된 checksum으로 수신되어, 값을 무시하고 Pp 명령어 끝에 :으로 회신한 횟수를 가리킴. 3rd byte는 관절제어 PCB로 주어진 주기로 Pp 명령어가 보내어지지 않고 건너뛴 횟수를 가리킴. 4th word는 RS232C 통신도중에 RX-ERROR/BRKDKT/FE/OE/PE 에러가 발생하여 시리얼 통신 모듈을 자동으로 초기화한 횟수를 가리킴. 5th word는 CAN 통신도중에 FE/BE/SA1/CRCE/SE/ACKE/BO/EP/EW/bus-off 에러가 발생하여 CAN 통신 모듈을 자동으로 초기화한 횟수를 가리킴. 6th word는 RS485 통신도중에 RX-ERROR/BRKDKT/FE/OE/PE 에러가 발생하여 시리얼 통신 모듈을 자동으로 초기화한 횟수를 가리킴.

명령어	구분	설명
;QEE; (+#^)	operation	통신에러횟수를 읽어서 host로 전송함.
	return value	QEEHLHL, HLHL, HLHL, HLHL, HLHL; 1st word : RS232C 통신에러 횟수 (회) 2nd word : CAN 통신에러 횟수 (회) 3rd word : RS485 통신에러 횟수 (회) 4th word : RS485 또는 CAN 통신의 수신에러 횟수 (회) 5th word : RS485 또는 CAN 통신의 송신에러 횟수 (회)
	comment	1st word는 RS232C 통신도중에 RX-ERROR/BRKDKT/FE/OE/PE 에러가 발생하여 시리얼 통신 모듈을 자동으로 초기화한 횟수를 가리킴. 2nd word는 CAN 통신도중에 FE/BE/SA1/CRCE/SE/ACKE/BO/EP/EW/bus-off 에러가 발생하여 CAN 통신 모듈을 자동으로 초기화한 횟수를 가리킴. 3rd word는 RS485 통신도중에 RX-ERROR/BRKDKT/FE/OE/PE 에러가 발생하여 시리얼 통신 모듈을 자동으로 초기화한 횟수를 가리킴. 4th word의 하위바이트는 RS485 또는 CAN 통신도중에 수신시간 초과에러가 발생한 횟수를 가리킴. 상위바이트는 2nd word나 3rd word의 값과 같음. (단 bus-off error는 제외) 5th word의 하위바이트는 RS485 통신도중에 TXEN 신호의 high 상태가 시간 초과되어 발생한 에러 횟수 또는 CAN 통신도중에 송신시간 초과에러가 발생한 횟수를 가리킴. 상위바이트는 RS485 통신도중에 송신시간 초과에러가 발생한 횟수를 가리킴. (단 TXEN 신호의 high 상태 시간초과 에러는 제외)

명령어	구분	설명
;QE55A; (+#^%) or ;QE55A,HLHL,HLHL; (+#^)	operation	모터1/2의 순간목표/명령/현재위치, 엔코더카운터값 등을 초기값으로 재설정함. 1st word : 모터1의 위치 기준값 5,000,000에 더해지는 위치 offset 값이며 FFFF는 -1에 해당함. 2nd word : 모터2의 위치 기준값 5,000,000에 더해지는 위치 offset 값이며 FFFF는 -1에 해당함.
	return value	QE55A;
	comment	위치의 초기값은 5,000,000 4채배 pulse이며 엔코더카운터값 및 제어오차의 초기값은 0, 4채배 pulse임. 순간목표위치값(positionx_set), 명령위치값(positionx_cmd), 현재위치값(positionx)임. 절대각센서를 사용하는 경우, 홀센서가 없는 경우, feedback-type 스텝모터에서는 명령의 사용이 불가함. 위치 offset 값은 2012년 4월 10일 S/W 버전부터 적용되며, 2014년 9월 21일 이전 S/W 버전에서는 일부 digital/analog Hall 센서만을 사용하는 BLDC 모터의 경우는 문제가 있을수 있음. 2014년 9월 21일 S/W 버전부터는 feedback-type 스텝모터에도 QE55A; 명령의 사용이 가능하며, 위치 offset 값은 스텝모터가 아닌 경우로서 엔코더를 사용하는 경우에만 적용됨. 2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스텝모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 초기화후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 5,000,000을 원점으로 하는 위치명령을 사용할수도 있음. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.

명령어	구분	설명
;QE5A5; (+#^%) or ;QE5A5,HLHL; (+#^)	operation	모터1의 순간목표/명령/현재위치, 엔코더카운터값 등을 초기값으로 재설정함. 1st word : 모터1의 위치 기준값 5,000,000에 더해지는 위치 offset 값이며 FFFF는 -1에 해당함.
	return value	QE5A5;
	comment	위치의 초기값은 5,000,000 4체배 pulse이며 엔코더카운터값 및 제어오차의 초기값은 0, 4체배 pulse임. 순간목표위치값(positionx_set), 명령위치값(positionx_cmd), 현재위치값(positionx)임. 절대각센서를 사용하는 경우, 홀센서가 없는 경우, feedback-type 스텝모터에서는 명령의 사용이 불가함. 위치 offset 값은 2012년 4월 10일 S/W 버전부터 적용되며, 2014년 9월 21일 이전 S/W 버전에서는 일부 digital/analog Hall 센서만을 사용하는 BLDC 모터의 경우는 문제가 있을수 있음. 2014년 9월 21일 S/W 버전부터는 feedback-type 스텝모터에도 QE5A5; 명령의 사용이 가능하며, 위치 offset 값은 스텝모터가 아닌 경우로서 엔코더를 사용하는 경우에만 적용됨. 2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스텝모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 초기화후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 5,000,000을 원점으로 하는 위치명령을 사용할수도 있음. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.

명령어	구분	설명
;QE5A5A; (+#^%) or ;QE5A5A,HLHL; (+#^)	operation	모터2의 순간목표/명령/현재위치, 엔코더카운터값 등을 초기값으로 재설정함. 1st word : 모터2의 위치 기준값 5,000,000에 더해지는 위치 offset 값이며 FFFF는 -1에 해당함.
	return value	QE5A5A;
	comment	위치의 초기값은 5,000,000 4체배 pulse이며 엔코더카운터값 및 제어오차의 초기값은 0, 4체배 pulse임. 순간목표위치값(positionx_set), 명령위치값(positionx_cmd), 현재위치값(positionx)임. 절대각센서를 사용하는 경우, 홀센서가 없는 경우, feedback-type 스텝모터에서는 명령의 사용이 불가함. 위치 offset 값은 2012년 4월 10일 S/W 버전부터 적용되며, 2014년 9월 21일 이전 S/W 버전에서는 일부 digital/analog Hall 센서만을 사용하는 BLDC 모터의 경우는 문제가 있을수 있음. 2014년 9월 21일 S/W 버전부터는 feedback-type 스텝모터에도 QE5A5A; 명령의 사용이 가능하며, 위치 offset 값은 스텝모터가 아닌 경우로서 엔코더를 사용하는 경우에만 적용됨. 2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스텝모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 초기화후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 5,000,000을 원점으로 하는 위치명령을 사용할수도 있음. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.
;QEi; (+#^)	operation	모터1/2에 대하여 설정된 가상원점의 5,000,000에 대한 차이값을 읽어서 host로 전송함.
	return value	QEi 1 1 1 1, 1 1 1 2; 1st long int : 모터1에 대하여 설정된 가상원점의 5,000,000에 대한 차이값. 2nd long int : 모터2에 대하여 설정된 가상원점의 5,000,000에 대한 차이값.
	comment	2014년 9월 21일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;QE5AA5; (+#^)	operation	모터1/2의 현재 순간목표위치값을 PC 명령에서 사용하는 가상원점으로 설정함.
	return value	QE5AA5;
	comment	이 명령어를 입력하는 순간의 위치값을 가상 원점으로 설정하며, 이 가상 원점은 PC 명령을 사용할 때만 적용됨. 가상원점은 PC 명령에서 5,000,000을 위치명령값으로 주었을 때의 위치에 해당함. 이 명령어는 2014년 3월 21일 S/W 버전부터 적용됨. 2014년 9월 21일 S/W 버전부터는 흡동작을 하였을때 원점값이 5,000,000과 조금 다른 값이 될 수있는데 (스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우), 흡동작을 할 때도 PC 명령에서 사용하는 가상원점을 자동 설정함. 전원을 처음 투입할때도 가상원점을 설정함. 가상원점위치값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.
;QEAA55; (+#^)	operation	모터1의 현재 순간목표위치값을 PC 명령에서 사용하는 가상원점으로 설정함.
	return value	QEAA55;
	comment	이 명령어를 입력하는 순간의 위치값을 가상 원점으로 설정하며, 이 가상 원점은 PC 명령을 사용할 때만 적용됨. 가상원점은 PC 명령에서 5,000,000을 위치명령값으로 주었을 때의 위치에 해당함. 이 명령어는 2014년 3월 21일 S/W 버전부터 적용됨. 2014년 9월 21일 S/W 버전부터는 흡동작을 하였을때 원점값이 5,000,000과 조금 다른 값이 될 수있는데 (스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우), 흡동작을 할 때도 PC 명령에서 사용하는 가상원점을 자동 설정함. 전원을 처음 투입할때도 가상원점을 설정함. 가상원점위치값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.
;QE55AA; (+#^)	operation	모터2의 현재 순간목표위치값을 PC 명령에서 사용하는 가상원점으로 설정함.
	return value	QE55AA;
	comment	이 명령어를 입력하는 순간의 위치값을 가상 원점으로 설정하며, 이 가상 원점은 PC 명령을 사용할 때만 적용됨. 가상원점은 PC 명령에서 5,000,000을 위치명령값으로 주었을 때의 위치에 해당함. 이 명령어는 2014년 3월 21일 S/W 버전부터 적용됨. 2014년 9월 21일 S/W 버전부터는 흡동작을 하였을때 원점값이 5,000,000과 조금 다른 값이 될 수있는데 (스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우), 흡동작을 할 때도 PC 명령에서 사용하는 가상원점을 자동 설정함. 전원을 처음 투입할때도 가상원점을 설정함. 가상원점위치값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.

명령어	구분	설명
;QP; (+#^%) 2014년 3월 21일 S/W 버전부터 ;QPC; (+#^)	operation	현재 위치값을 읽어서 host로 전송함.
	return value	QPHLHLHLHL,HLHLHLHL; 1st long word : 모터1의 현재 위치값 (4체배 pulse, 32bit) 2nd long word : 모터2의 현재 위치값 (4체배 pulse, 32bit) QPC; 인 경우에 리턴값은 QE5AA5; QEAA55; QE55AA; 등으로 설정한 가상원점을 위치값 5,000,000으로 보았을 때의 현재 위치값을 의미함. (2014년 3월 21일 S/W 버전부터 사용가능함.)
	comment	엔코더카운터값을 읽어서 32bit long word로 변환한 값을 읽어냄.
;Qp; (+#^%) 2014년 3월 21일 S/W 버전부터 ;Qpc; (+#^)	operation	순간목표위치값을 읽어서 host로 전송함.
	return value	QpHLHLHLHL,HLHLHLHL; 1st long word : 모터1의 순간목표위치값 (4체배 pulse, 32bit) 2nd long word : 모터2의 순간목표위치값 (4체배 pulse, 32bit)
	comment	순간목표위치값(positionx_set)을 읽어서 32bit long word로 변환한 값을 읽어냄. 순간목표위치값은 1ms의 제어주기마다 설정되는 순간순간의 목표위치값을 가리킴. Qpc; 인 경우에 리턴값은 QE5AA5; QEAA55; QE55AA; 등으로 설정한 가상원점을 순간위치명령값 5,000,000으로 보았을 때의 현재 순간위치명령값을 의미함. (2014년 3월 21일 S/W 버전부터 사용가능함.)
;Qq; (+#^)	operation	option으로 부착된 엔코더카운트값을 읽어서 host로 전송함.
	return value	QqHLHLHLHL,HLHLHLHL; 1st long word : 모터1의 현재 엔코더카운트값 (4체배 pulse, 32bit) 2nd long word : 모터2의 현재 엔코더카운트값 (4체배 pulse, 32bit)
	comment	option으로 부착된 엔코더카운터값을 읽어서 32bit long word로 변환한 값을 읽어냄.

명령어	구분	설명
;QV; (+#^%)	operation	현재 속도값을 읽어서 host로 전송함.
	return value	QV <i>iiii1,iiii2</i> ; 1st integer: 모터1의 현재 속도값 (rpm) 2nd integer : 모터2의 현재 속도값 (rpm)
	comment	1msec 동안 입력된 엔코더 펄스수를 1개의 ring buffer에 1ms 마다 저장하고, 이를 1ms 마다 모두 더함으로써, 10msec 동안 입력된 엔코더 펄스수를 RPM으로 환산한 값임. 4채배된 4096 pulse의 엔코더 경우에 1000rpm 이면 683 pulse 가 10ms 동안에 입력됨. 이때 속도값의 분해능은 1.465rpm 이며, 시정수는 6ms 정도임. <i>iiii1,iiii2</i> 는 부호가 있는 5자리 십진수임. <i>iiii1,iiii2</i> 는 부호가 있는 7자리 십진수임.
;Qv; (+#^)	operation	현재의 공급전압과 전압강하의 크기를 읽어서 host로 전송함.
	return value	Qv <i>HLHL,HLHL</i> ; 1st word: 현재의 filtering된 모터용 공급전압 크기 (0.01V) 2nd word : 현재의 filtering된 모터용 공급전압의 전압강하 크기 (0.01V)
	comment	공급전압에 대한 필터의 시정수는 1ms 임. 공급전압을 IR2175S로 측정하는 경우는 2ms.

명령어	구분	설명
;QC; (+#^%)	operation	모터에 흐르는 filtering된 상전류값을 읽어서 host로 전송함.
	return value	<p>QCiiii1,iiii2; or QCiiii1,iiii2,iiii3,iiii4; (2상 STEP형 만 해당됨) or QCiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (3상 STEP형 만 해당됨) or QCiiii1,iiii2,iiii3,iiii4,iiii5; (5상 STEP형 만 해당됨)</p> <p>1st integer : 모터1 전류값 (bit) (STEP형은 coil1, RJM_VER3(BLDC)/7은 모터1의 +축 공급전류) 2nd integer : 모터2 전류값 (bit) (STEP형은 coil2, RJM_VER3(BLDC)/7은 모터2의 +축 공급전류) 3rd integer : coil3 전류값 (bit) (STEP형은 coil3) 4th integer : coil4 전류값 (bit) (STEP형은 coil4) 5rd integer : coil5 전류값 (bit) (STEP형은 coil5) 6th integer : coil6 전류값 (bit) (STEP형은 coil6)</p>
	comment	<p>$1A \times 25m\Omega \times 11(\text{앰프 증폭률}) = 0.275V$이므로 $3.3/3.0V$는 $3.3/3.0V \div 0.275V = 12/10.9A$에 해당하며 $85.25/375.375$ bit/A. (DC ver1.1) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$이므로 $3.3/3.0V$는 $3.3/3.0V \div 0.25V = 13.2/12A$에 해당하며 $77.5/341.25$ bit/A. (DC V1.2부터) $1A \times 2m\Omega \times 50(\text{앰프 증폭률}) = 0.1V$이므로 $3.3/3.0V$는 $3.3/3.0V \div 0.1V = 33/30A$에 해당하며 $31/136.5$ bit/A. (Piggyback-DC/RJM_VER8부터) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) \times (3.3/5) = 0.165V$이므로 $3.3/3.0V$는 $3.3/3.0V \div 0.165V = 20/18.18A$에 해당하 며 $51.15/225.225$ bit/A. (Piggyback-BLDC형) $1A \times 10m\Omega \times 50(\text{앰프 증폭률}) = 0.5V$이므로 $3.3/3.0V$는 $3.3/3.0V \div 0.5V = 6.6/6A$에 해당하며 $155/682.5$ bit/A. (mini-DC)</p>

명령어	구분	설명
	comment	<p> $1A \times 20(10/10)m\Omega \times 50(\text{앰프 증폭률}) = 1.0/0.5/0.5V$이므로 $3.3/5.0/5.0V$는 $3.3/5.0/5.0V \div 1.0/0.5/0.5V = 3.3/10/10A$에 해당하며 $310/409.5/409.5 \text{ bit/A}$. (STEP형) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$이므로 $5.0V$는 $5.0V \div 0.25V = 20A$에 해당하며 204.75 bit/A. (Sensor-BLDC) </p> <p> 3상 STEP형은 모터1(coil1,coil2,coil3)/모터2(coil4,coil5,coil6) 으로 코일이 사용됨. iiii1,iiii2,iiii3,iiii4,iiii5,iiii6은 부호가 있는 십진수임. SF 명령을 사용하여 필터 시정수를 조정할 수 있으며, 사용 필터 채널은 상전류값 순서대로 0/1/2/3/4/5/임. </p> <p> 2008년 6월 이후의 모든 F2811/F28334 H/W 버전에서는 다음과 같이 통일 되었음. 2mΩ인 경우는 81.9/81.9 bit/A (RJM_VER3 : IRMCK201을 사용한 BLDC제어기, RJM_VER8 : 고출력 DC) 5mΩ인 경우는 204.75/204.75 bit/A (RJM_VER7 : IRMCK201을 사용하지 않은 BLDC제어기) 10mΩ인 경우는 409.5/409.5 bit/A (RJM_VER4 : mini DC, RJM_VER5 : step motor) IRMCK201을 사용한 경우는 IRMCK201의 read group Iq 값이며, 4096일 때 정격전류값(RMS치)에 해당함. 2009년 7월 15일 S/W 버전부터는 IRMCK201을 사용한 경우도 전류감지저항에 따라서 scale factor가 정해짐. </p>

명령어	구분	설명
;Qc; (+#^%)	operation	모터에 흐르는 filtering된 상전류값을 읽어서 0.01A 단위의 암페어값으로 환산하여 host로 전송함.
	return value	<p>Qciiii1,iiii2; or Qciiii1,iiii2,iiii3,iiii4; (2상 STEP형 만 해당됨) or Qciiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (3상 STEP형 만 해당됨) or Qciiii1,iiii2,iiii3,iiii4,iiii5; (5상 STEP형 만 해당됨)</p> <p>1st integer : 모터1 코일 전류값 (0.01A) (STEP형은 coil1, RJM_VER3(BLDC)/7은 모터1의 +측 공급전류) 2nd integer : 모터2 코일 전류값 (0.01A) (STEP형은 coil2, RJM_VER3(BLDC)/7은 모터2의 +측 공급전류) 3rd integer : coil3 전류값 (0.01A) (STEP형은 coil3) 4th integer : coil4 전류값 (0.01A) (STEP형은 coil4) 5rd integer : coil5 전류값 (0.01A) (STEP형은 coil5) 6th integer : coil6 전류값 (0.01A) (STEP형은 coil6)</p>
	comment	<p>3상 STEP형은 모터1(coil1,coil2,coil3)/모터2(coil4,coil5,coil6) 으로 코일이 사용됨. iiiii1,iiii2,iiii3,iiii4,iiii5,iiii6은 부호가 있는 십진수임. SF 명령을 사용하여 필터 시정수를 조정할 수 있으며, 사용 필터 채널은 상전류값 순서대로 0/1/2/3/4/5/임. 2009년 7월 15일 S/W 버전부터 Z3 명령어로 전류감지저항값이 정상적으로 설정되어 있어야함. 2009년 7월 15일 S/W 버전부터 IRMCK201을 사용한 경우는 Zc 명령어로 nominal current가 정상적으로 설정되어 있어야함.</p>

명령어	구분	설명
;QK; (+#^)	operation	모터에 흐르는 filtering된 총전류값을 읽어서 host로 전송함.
	return value	QKiiii1,iiii2;
	comment	스케일은 Qc 명령에 준함. SF 명령을 사용하여 필터 시정수를 조정할 수 있으며, 사용 필터 채널은 총전류값 순서대로 14/15임. F2811/F28334 에서만 가능한 명령임. 2008년 6월 이후의 모든 F2811/F28334 H/W 버전에서는 다음과 같이 통일 되었음. 2mΩ인 경우는 81.9/81.9 bit/A (RJM_VER3 : IRMCK201을 사용한 BLDC제어기, RJM_VER8 : 고풍력 DC) 5mΩ인 경우는 204.75/204.75 bit/A (RJM_VER7 : IRMCK201을 사용하지 않은 BLDC제어기) 10mΩ인 경우는 409.5/409.5 bit/A (RJM_VER4 : mini DC, RJM_VER5 : step motor)
;Qk; (+#^)	operation	모터에 흐르는 filtering된 총전류값을 읽어서 암페어값으로 환산하여 host로 전송함.
	return value	Qkiiii1,iiii2;
	comment	스케일은 Qc 명령에 준함. SF 명령을 사용하여 필터 시정수를 조정할 수 있으며, 사용 필터 채널은 총전류값 순서대로 14/15임. F2811/F28334 에서만 가능한 명령임. 2009년 7월 15일 S/W 버전부터 Z3 명령어로 전류감지저항값이 정상적으로 설정되어 있어야함. 2009년 7월 15일 S/W 버전부터 IRMCK201을 사용한 경우는 Zc 명령어로 nominal current가 정상적으로 설정되어 있어야함.

명령어	구분	설명
(Force/Moment 센서가 아닌 경우) ;Qe; (+#^%)	operation	error, error 적분값, error 미분값을 읽어서 host로 전송함.
	return value	Qeiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; 1st long integer : 모터1의 위치에러 (4체배 pulse) 2nd long integer : 모터1의 위치에러 적분값 (4체배 pulse) 3rd long integer : 모터1의 위치에러 미분값 (4체배 pulse) 4th long integer : 모터2의 위치에러 (4체배 pulse) 5th long integer : 모터2의 위치에러 적분값 (4체배 pulse) 6th long integer : 모터2의 위치에러 미분값 (4체배 pulse)
	comment	단위는 4체배된 엔코더 펄스수임.
(Force/Moment 센서인 경우) ;Qe; (+#^)	operation	X-moment, Y-moment, Z-force, extra load cell값에 대하여 추가 필터링한 값을 읽어서 host로 전송함. (6축의 경우는 X-moment, Y-moment, Z-force, X-force, Y-force, Z-moment 임)
	return value	Qeiiii0,iiii1,iiii2,iiii3; or Qeiiii0,iiii1,iiii2,iiii3,iiii4,iiii5; (6축 F/T 센서의 경우) iiii0 : X-moment의 추가필터링값 (0.01Nm 단위) iiii1 : Y-moment의 추가필터링값 (0.01Nm 단위) iiii2 : Z-force의 추가필터링값 (0.1N 단위) iiii3 : 첫번째 amplifier에 연결된 load cell의 증폭후 A/D 변환값의 추가필터링값 - offset (bit 단위, 4095일때 3V) or X-force의 추가필터링값 (6축 F/T 센서의 경우) iiii4 : Y-force의 추가필터링값 (6축 F/T 센서의 경우) iiii5 : Z-moment의 추가필터링값 (6축 F/T 센서의 경우)
	comment	FT 센서용 amp 회로인 경우에만 해당됨. 추가필터의 시정수는 SF9,xxxx,xxxx; 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할수 있음. iiii3는 6축 F/T 센서가 아닌 경우로서 외장형일때 첫번째 amplifier와 나머지 3개의 amplifier를 분리하여 사용하는 경우에만 의미있는 값이며, 이렇게 사용하는 것은 주문 사양임. 2012년 3월 9일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;QF; (+#^%)	operation	filter 출력값을 읽어서 host로 전송함.
	return value	<p>QFiiii0,iiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7;</p> <p>0th integer : CMPR1의 filtering된 값 (bit, 모터1의 PWM 출력값) 1st integer : CMPR3의 filtering된 값 (bit, 모터1의 전류 제한값 또는 설정값) 2nd integer : CMPR2의 filtering된 값 (bit, 모터2의 PWM 출력값) 3rd integer : T1CMPR의 filtering된 값 (bit, 모터2의 전류 제한값 또는 설정값) 4th integer : A/D 변환한 MOTOR1의 전류치를 FILTERING한 값 (bit, DC V1.1 : 85.25/375.375 bit/A, DC V1.2부터 : 77.5/341.25 bit/A, Piggyback-DC/RJM_VER8부터 : 31/136.5 bit/A, BLDC : 51.15/225.225 bit/A, mini-DC : 155/682.5 bit/A, STEP : 310/682.5 bit/A, sensor BLDC : 204.75 bit/A) 5th integer : A/D 변환한 MOTOR2의 전류치를 FILTERING한 값 (bit, DC V1.1 : 85.25/375.375 bit/A, DC V1.2부터 : 77.5/341.25 bit/A, Piggyback-DC/RJM_VER8부터 : 31/136.5 bit/A, BLDC : 51.15/225.225 bit/A, mini-DC : 155/682.5 bit/A, STEP : 310/682.5 bit/A, sensor BLDC : 204.75 bit/A) 6th integer : 모터용 공급전압값을 FILTERING한 값 (bit) (DC V1.2이전 : 414/956 bit, DC V1.2부터/mini-DC/Piggyback-DC : 281.8/650 bit/10V, Piggyback-BLDC : 724.8 bit/10V Sensor-BLDC : 638.9 bit/10V) 7th integer :</p>
	comment	<p>filtered_value = butterworth_filter(filter_number, value_to_be_filtered); 의 형식으로 사용하면 됨. filter_number는 0 - 7중의 하나를 선택하며, 한가지 신호에 한개의 필터번호만 사용하여야함.</p>

명령어	구분	설명
;Qf; (+#^)	operation	X-moment, Y-moment, Z-force, extra load cell를 읽어서 host로 전송함. (6축의 경우는 X-moment, Y-moment, Z-force, X-force, Y-force, Z-moment 임)
	return value	Qfiiii0,iiii1,iiii2,iiii3; or Qfiiii0,iiii1,iiii2,iiii3,iiii4,iiii5; (6축 F/T 센서의 경우) iiii0 : X-moment (0.01Nm 단위) iiii1 : Y-moment (0.01Nm 단위) iiii2 : Z-force (0.1N 단위) iiii3 : 첫번째 amplifier에 연결된 load cell의 증폭후 A/D 변환값 - offset (bit 단위, 4095일때 3V) or X-force (6축 F/T 센서의 경우) iiii4 : Y-force (6축 F/T 센서의 경우) iiii5 : Z-moment (6축 F/T 센서의 경우)
	comment	FT 센서용 amp 회로인 경우에만 해당됨. iiii3는 6축 F/T 센서가 아닌 경우로서 외장형일때 첫번째 amplifier와 나머지 3개의 amplifier를 분리하여 사용하는 경우에만 의미있는 값이며, 이렇게 사용하는 것은 주문 사양임. 삼성의료기의 FT 센서인 경우는 무조건 4개의 값만 host로 전송됨.

명령어	구분	설명
;QW; (+#^)	operation	모터1과 모터2의 input power, output power, output torque, efficiency를 읽어서 host로 전송함.
	return value	QWiiii0,iiii1,iiii2,...,iiii5,iiii6,iiii7; iii0 : 모터1의 input power (0.1W 단위) iii1 : 모터1의 output power (0.1W 단위) iii2 : 모터1의 output torque (mNm 단위) iii3 : 모터1의 efficiency (% 단위) iii4 : 모터2의 input power (0.1W 단위) iii5 : 모터2의 output power (0.1W 단위) iii6 : 모터2의 output torque (mNm 단위) iii7 : 모터2의 efficiency (% 단위)
	comment	사전에 Z3 명령에 의하여 전류감지 저항값을 설정하여야 하며, ZQ/ZR 명령어에 의하여 모터의 특성파라메터가 설정되어 있어야 하며, SX 명령어로 bit3(밑에서 4번째 비트)가 세트되어 있어야 계산이 실시됨. IRMCK201을 사용한 RJM_VER3의 경우는 Zc 명령에 의한 nominal current 값도 설정되어 있어야함. efficiency는 시정수 8ms의 필터링이 된 값임. IRMCK201을 사용한 RJM_VER3의 경우는 시정수 8ms의 필터링이 된 전류치를 사용함.

명령어	구분	설명
;Qw; (+#^)	operation	모터1과 모터2의 loss power, winding temperature, housing temperature, ambient temperature를 읽어서 host로 전송함.
	return value	Qwiiii0,iiii1,iiii2,...,iiii5,iiii6,iiii7; iiii0 : 모터1의 손실전력 (0.1W 단위) iiii1 : 모터1의 권선온도 (0.1도 단위) iiii2 : 모터1의 모터케이스온도 (0.1도 단위) iiii3 : 모터1의 주변온도 (0.1도 단위) iiii4 : 모터2의 손실전력 (0.1W 단위) iiii5 : 모터2의 권선온도 (0.1도 단위) iiii6 : 모터2의 모터케이스온도 (0.1도 단위) iiii7 : 모터2의 주변온도 (0.1도 단위)
	comment	사전에 Z3 명령에 의하여 전류감지 저항값을 설정하여야 하며, ZQ/ZR 명령어에 의하여 모터의 특성파라메터가 설정되어 있어야 하며, SX 명령어로 bit3(밑에서 4번째 비트)가 세트되어 있어야 계산이 실시됨. IRMCK201을 사용한 RJM_VER3의 경우는 Zc 명령에 의한 nominal current 값도 설정되어 있어야함. 주변온도는 0도로 세팅되어 있음.

명령어	구분	설명
;QJ; (+#^)	operation	faultBufPointer, faultBuf[0-7]을 읽어서 host로 전송함.
	return value	QJHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; 1st word : faultBufPointer [0-7] 2nd word - 9th word : faultBuf[0] - faultBuf[7]
	comment	전원을 켜 이후에 발생한 오류에 대하여 flash 코드를 저장한 것임. 전원이 차단되면 이 flash 코드는 사라짐. 2012년 2월 10일 S/W 버전부터 사용가능한 명령임. 2012년 8월 18일 이전 S/W 버전에서는 오류코드의 값이 >>4 되어서 리턴되는 오류가 있었음.
;Qj; (+#^)	operation	faultBufPointer_save, faultBuf_save[0-14]을 읽어서 host로 전송함.
	return value	QjHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; 1st word : faultBufPointer_save [0-14] 2nd word - 16th word : faultBuf_save[0] - faultBuf_save[14]
	comment	오류가 발생한 경우, flash 코드를 오류발생 순서대로 EEPROM에 저장한 값임. 따라서 과거의 flash 코드도 가장 최근 것이 전원 투입/차단 횟수에 관계없이 남아 있음. 에러코드의 상위 8비트가 0 일때는 이번의 전원투입이후에 발생한 오류코드임을 의미하며, 0이 아닌값이면 클수록 더 이전에 발생한 오류코드임을 의미함. 2012년 2월 10일 S/W 버전부터 사용가능한 명령임. 2012년 8월 18일 이전 S/W 버전에서는 오류코드의 값이 >>4 되어서 리턴되는 오류가 있었음. 2013년 1월 16일 이전 S/W 버전에서는 정상적인 전원 off 시에도 저전압오류를 검출하고 오류코드가 저장되는 경우가 있으며, 각종 오류코드의 저장 자체가 제대로 안되는 경우도 있음. 설정된 값은 EEPROM에 저장된 값임.
;QjA55A; (+#^)	operation	faultBufPointer_save, faultBuf_save[0-14]을 0으로 clear 함.
	return value	QjHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; 1st word : faultBufPointer_save [0-14] 2nd word - 16th word : faultBuf_save[0] - faultBuf_save[14]
	comment	2012년 2월 10일 S/W 버전부터 사용가능한 명령임. EEPROM에 저장된 값까지 clear하려면 QjA55A; 실행후에 EsA55A; 명령으로 저장하여야함. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Qdddd1; (+#^%)	operation	Sleep_1micro(dddd1), Sleep_10micro(dddd1), Sleep_msec(1) 함수실행결과의 지연시간동안 CS_EEPROM_N_port에 low pulse를 출력함. 1st word : 지연시간 [0-100]
	return value	Qdddd1;
	comment	
;QD; (+#^%)	operation	diagnosis mode를 개시함.
	return value	QD;
	comment	특별한 조치는 없음.
;QX; (+#^%)	operation	diagnosis mode를 종료함.
	return value	QX;
	comment	특별한 조치는 없음.

명령어	구분	설명
;Qxddd1; (+#^%) ;Qx?; (+#^%)	operation	1msec 또는 2ms 마다 프로그램에서 설정된 4word 또는 8word를 host PC로 전송하는 데이터의 종류를 선택하는 변수인 data_logging_select 변수값을 설정함. 1st word : data_logging_select 변수값
	return value	Qxddd1;
	comment	data_logging_select 값에 따라서 host PC로 전송하는 값을 선택하고 보낼 데이터를 SCITX0[]에 저장하는 것은 cpu_joio.c에 이미 반영되어 있어야함. data_logging_select 값에 따라 전송되는 변수값은 다음 페이지에 list되었음. 2012년 1월 07일 S/W 버전부터는 data_logging_select 값이 4096~8191인 경우는 0~4095인 경우와 같은 데이터가 전송되는데, 4096~6143일 때는 4word의 데이터를 2ms 마다 13 byte로 나누어서 보내며, 6144~8191일 때는 8word의 데이터를 2ms 마다 16 byte로 나누어서 보냄. 2012년 6월 16일 S/W 버전부터는 data_logging_select 값이 500~599인 경우는 20KHz ~ 100KHz interrupt routine에서 4096word를 capture하며, PC로의 전송은 Qx9; 에 의하여 개시됨. 단 이 기능은 F/T 센서의 경우는 해당없으며, Qx6;로 바뀌어서 실행됨.
;Qxy; (+#^%)	operation	1msec 또는 2ms 마다 프로그램에서 설정된 4word 또는 8word를 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
	return value	none.
	comment	전송되는 데이터는 Qxy; 명령을 전송해온 RS232C/RS485/CAN 채널로만 전송됨. 모터제어기의 RS232C 채널이 115200 baud rate의 전송속도로 설정이 되어 있을 때만 유효함. 전송데이터의 종류는 Qx 명령어에 의하여 선택됨. Qxy; 명령어를 한번 보내면 전송개시, 한번 더 보내면 전송을 정지함. 4 word의 값에 해당하는 총 8바이트를 binary 형태로 separator (;) 없이 1ms 마다 host로 전송하는데, 이때 한 word의 값은 low-byte, high-byte 순으로 전송함. 2012년 1월 07일 S/W 버전부터는 data_logging_select 값에 따라서 1ms당 4word 또는 2ms당 4word 또는 2ms당 8word를 8byte 또는 13byte 또는 16byte로 나누어서 보냄. 2012년 12월 27일 S/W 버전부터는 Qxyz; 명령어로도 전송을 정지할수 있음.

명령어	구분	설명
;Qxyz; (+#^)	operation	Qxy; 또는 Qz; 또는 Qzx; 또는 Qzy; 명령에 의한 모니터링 작동을 중지함.
	return value	none.
	comment	2012년 12월 27일 S/W 버전부터 사용가능한 명령어임.

Qx, Qz, Qzx, Qzy 에 의한 실시간 데이터 모니터링

Qx0; ~ Qx2047; 는 Qx4096; ~ Qx6143; 과 같은 값을 PC로 전송하되,
 Qx0; ~ Qx2047;는 1ms당 4word를 8byte로 나누어서 기존방식으로 PC에 전송하며,
 Qx4096; ~ Qx6143;는 2ms당 4word를 보내는데 한 word당 3byte로 나누어서 보내고 마지막에는 0xaa를 붙여 보내기 때문에 총 13byte를 전송함.
 (첫바이트는 word의 0~5번 비트까지, 두번째바이트는 word의 6~11번 비트까지, 세번째바이트는 word의 12~15번 비트까지를 가지고 있음)
 Qx6144; ~ Qx8191; 의 경우는 2ms당 8word를 16byte로 나누어서 기존방식으로
 PC에 전송하나 현재는 Qx6144; 인 경우에만 정상적인 8word가 PC로 전송됨.

Qz; 를 보내면 1ms당 4word를 8byte로 나누어서 기존방식으로 PC에 전송하며,
 Qzx; 를 보내면 2ms당 4word를 13byte로 나누어서 보내며 (데이터 형식은 Qx4096; ~ Qx6143; 의 경우와 같음)
 Qzy; 를 보내면 2ms당 8word를 16byte로 나누어서 기존방식으로
 PC에 전송하며 8 word의 주소는 QZ 로 설정하는 4 word와 QY로 설정하는 4word를 합해서 설정됨

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0	모터1의 순간위치명령값 (position1_set)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S1/Ts/Tu 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임	모터1의 현재 위치값(position1)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S1/Ts/Tu 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임	모터2의 순간위치명령값 (position2_set)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S2/Tt/Tv 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임	모터2의 현재 위치값(position2)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S2/Tt/Tv 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임
1	모터1의 순간속도명령값 (speed1_cmd_old / SPEED_SCALE_FACTOR, 03번 모드에서는 (speed1_long_cmd + speed1_long_cmd_feedforward) / 10, 06/08/09/0A번 모드에서는 speed1_long_cmd_old / 10) 단위크기는 RPM	모터1의 현재 속도값 (speed1_RPM, digital Hall 센서만을 사용하는 경우는 필터링한 값임) 단위크기는 RPM	모터2의 순간속도명령값 (speed2_cmd_old / SPEED_SCALE_FACTOR, 03번 모드에서는 (speed2_long_cmd + speed2_long_cmd_feedforward) / 10, 06/08/09/0A번 모드에서는 speed2_long_cmd_old / 10) 단위크기는 RPM	모터2의 현재 속도값 (speed2_RPM, digital Hall 센서만을 사용하는 경우는 필터링한 값임) 단위크기는 RPM
2	모터1의 전류명령값 (current1_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 U상 전류의 A/D 변환값 (current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 V상 전류의 A/D 변환값 (current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 W상 전류의 A/D 변환값 (current3_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
3	모터2의 전류명령값 (current2_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 U상 전류의 A/D 변환값 (current4_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 V상 전류의 A/D 변환값 (current5_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 W상 전류의 A/D 변환값 (current6_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당

Qx0; 에 대한 return 값은 위치센서로 analog 또는 digital potentiometer을 사용하는 경우에 /4를 하지 않고 원래값을 보내줌.
또 S1/S2/Ts/Tt/Tu/Tv 명령에 의한 구동일 경우에도 /4를 하지 않고 원래값을 보내줌.

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
4	모터1의 현재 총전류의 A/D 변환값 (total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current1_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 총전류의 A/D 변환값 (total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current2_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
5	resolver를 사용하는 경우는 모터1의 필터링된 cos*4	모터1의 필터링된 sin*4	모터1의 AD2_12_bit_Value (원점)	모터1의 phase_angle1_old
5	유도모터의 경우는 ld1_AD	lq1_AD	lmr1>>16	speed1_RPM
5	그외는 모터1의 Hall 센서 A상 전압의 A/D 변환값 (0 - 4095)	모터1의 Hall 센서 B상 전압의 A/D 변환값 (0 - 4095)	모터1의 Hall 센서 C상 전압의 A/D 변환값 (0 - 4095)	모터1의 Hall 센서 A상 상승에지 기준 위상각 (0 - 16383)
6	resolver를 사용하는 경우는 모터1의 필터링된 cos*4	모터1의 필터링된 sin*4	모터1의 electrical_angle1_cnt>>8	모터1의 phase_angle1
6	유도모터의 경우는 speed1_RPM	(4096L*lq1_AD)/(int)(lmr1>>16)	FluxRate1	electrical_angle1_cnt>>8
6	그외는 모터2의 Hall 센서 A상 전압의 A/D 변환값 (0 - 4095)	모터2의 Hall 센서 B상 전압의 A/D 변환값 (0 - 4095)	모터2의 Hall 센서 C상 전압의 A/D 변환값 (0 - 4095)	모터2의 Hall 센서 A상 상승에지 기준 위상각 (0 - 16383)

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
7	resolver를 사용하는 경우는 모터1의 필터링된 cos*4	모터1의 필터링된 sin*4	모터1의 AD2_12_bit_Value (원점)	모터1의 (position1 & 0x3ffL)<<4; // 위상각 (linear resolver)
7	그외는 모터1의 전류명령값 (current1_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 U상 전류의 A/D 변환값을 필터링한 값 (current1_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 V상 전류의 A/D 변환값을 필터링한 값 (current2_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 W상 전류의 A/D 변환값을 필터링한 값 (current3_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
8	resolver를 사용하는 TPC_FOCUSING 경우는 모터1의 현재 위치값(position1)을 4로 나눈값 단위크기는 0.01953125μm	모터1의 현재 위치값 (position1_ENCODER)을 4로 나눈값 단위크기는 0.01953125μm	Hall1C 입력 전압을 필터링한 값 (모터1의 외부로부터의 전류명령값)	phase_angle1_old + ENC1Z_port * 1000;
8	그외는 모터2의 전류명령값 (current2_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 U상 전류의 A/D 변환값을 필터링한 값 (current4_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 V상 전류의 A/D 변환값을 필터링한 값 (current5_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 W상 전류의 A/D 변환값을 필터링한 값 (current6_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
10	resolver를 사용하는 경우는 모터2의 필터링된 cos*4	모터2의 필터링된 sin*4	모터2의 AD10_12_bit_Value (원점)	모터2의 phase_angle2_old
10	그외는 모터1의 입력 power (Pin1) 단위크기는 0.1W	모터1의 출력 power (Pout1) 단위크기는 0.1W	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 효율 단위크기는 1%

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
11	resolver를 사용하는 경우는 모터2의 필터링된 cos*4	모터2의 필터링된 sin*4	모터2의 electrical_angle2_cnt>>8	모터2의 phase_angle2
11	그외는 모터2의 입력 power(Pin2) 단위크기는 0.1W	모터2의 출력 power(Pout2) 단위크기는 0.1W	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 효율 단위크기는 1%
12	resolver를 사용하는 경우는 모터2의 필터링된 cos*4	모터2의 필터링된 sin*4	모터2의 AD10_12_bit_Value (원점)	모터2의 (position2 & 0x3ffL)<<4; // 위상각 (linear resolver)
12	그외는 모터1의 입력 power(Pin1) 단위크기는 0.1W	모터1의 출력 토크(Tout1) 단위크기는 mN	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 효율 단위크기는 1%
13	resolver를 사용하는 TPC_FOCUSING 경우는 모터2의 현재 위치값(position1)을 4로 나눈값 단위크기는 0.01953125μm	모터2의 현재 위치값 (position1_ENCODER)을 4로 나눈값 단위크기는 0.01953125μm	Hall2C 입력 전압을 필터링한 값 (모터2의 외부로부터의 전류명령값)	phase_angle2_old + ENC2Z_port * 1000;
13	모터2의 입력 power(Pin2) 단위크기는 0.1W	모터2의 출력 토크(Tout2) 단위크기는 mN	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 효율 단위크기는 1%
14	공급전압 65 가 1V 에 해당함.	모터1의 현재 총전류의 A/D 변환값(total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 부호있는 전류진폭 2/5/10mΩ의 경우 81.9/204.75/409.5 가 전류진폭 1A에 해당
15	공급전압 65 가 1V 에 해당함.	모터2의 현재 총전류의 A/D 변환값(total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 부호있는 전류진폭 2/5/10mΩ의 경우 81.9/204.75/409.5 가 전류진폭 1A에 해당

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
16	모터1의 손실 power(Pin1-Pout1) 단위크기는 0.1W	모터1의 권선 온도(Twinding1) 단위크기는 0.1도	모터1의 하우징 온도(Thousing1) 단위크기는 0.1도	모터1의 분위기 온도(Tambient1) 단위크기는 0.1도
17	모터2의 손실 power(Pin2-Pout2) 단위크기는 0.1W	모터2의 권선 온도(Twinding2) 단위크기는 0.1도	모터2의 하우징 온도(Thousing2) 단위크기는 0.1도	모터2의 분위기 온도(Tambient2) 단위크기는 0.1도
18	모터1의 Hall 센서 A상 전압의 A/D 변환값 (0 - 4095)	모터1의 Hall 센서 B상 전압의 A/D 변환값 (0 - 4095)	모터1의 Hall 센서 C상 전압의 A/D 변환값 (0 - 4095)	모터1의 U상 출력 위상각 단위크기는 1 deg
19	모터2의 Hall 센서 A상 전압의 A/D 변환값 (0 - 4095)	모터2의 Hall 센서 B상 전압의 A/D 변환값 (0 - 4095)	모터2의 Hall 센서 C상 전압의 A/D 변환값 (0 - 4095)	모터2의 U상 출력 위상각 단위크기는 1 deg
20	motor1_control_service_time + motor2_control_service_time * 100 단위크기는 1ms의 1%	mainLoopStep_cnt 0-9 범위의 값으로서 단위크기는 1ms	mainLoop_execution_time 단위크기는 1ms의 1%	maximum_interrupt_service_time 단위크기는 PWM 주기의 1%
21	motor1_error 위치제어 모드에서의 단위크기는 1 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함) 속도제어 모드에서의 단위크기는 0.1 rpm 전류제어 모드에서는 해당없음	motor1_integral 위치제어 모드에서의 단위크기는 1 encoder pulse-msec (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함) 속도제어 모드에서의 단위크기는 0.1 rpm-sec 전류제어 모드에서는 해당없음.	motor1_derivative 위치제어 모드에서의 단위크기는 1 encoder pulse/ms (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함) 속도제어 모드에서의 단위크기는 0.1 rpm/ms 전류제어 모드에서는 해당없음	current1_cmd (단 01번 위치제어 모드에서는 motor1_PWM_set, 03번 위치제어모드에서는 speed1_long_cmd) 위치/속도제어 모드에서의 단위크기는 AD 변환된 전류값의 단위크기와 같음 전류제어 모드에서는 해당없음
22	motor2_error	motor2_integral	motor2_derivative	current2_cmd (단 01번 위치제어 모드에서는 motor2_PWM_set, 03번 위치제어모드에서는 speed2_long_cmd)

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
23	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilTA[10] - DC_BUS_VOLT_filtered_at_power_ up 65 가 1V 에 해당함	speed1_RPM 단위크기는 RPM	vfilTA[10] 65 가 1V 에 해당함	BRAKE_RELAY_port * 1000 1000 이면 on, 0 이면 off
23	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilTA[10] 65 가 1V 에 해당함	DC_BUS_VOLT_filtered_at_power_ up 65 가 1V 에 해당함	speed1_RPM 단위크기는 RPM	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off
24	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilTA[10]	speed1_RPM	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 1000 이면 on, 0 이면 off
24	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilTA[10]	전압오차 brake_action_error	브레이크저항 구동 PWM duty T1CMPR_reg	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off
25	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilTA[10]	DC_BUS_VOLT_filtered_at_power_ up	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 1000 이면 on, 0 이면 off
25	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilTA[10] - DC_BUS_VOLT_filtered_at_power_ up	vfilTA[10]- DC_BUS_VOLT_slow_filtered	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
30	(fault1_status >> 11) & 0x0001	controller_heat_dissipation1/100 단위크기는 0.1W	controller_heat_dissipation_limit/100 단위크기는 0.1W	current1_control_limit - current1_control_limit_cnt
31	(fault1_status >> 11) & 0x0001	controller_heat_dissipation2/100 단위크기는 0.1W	controller_heat_dissipation_limit/100 단위크기는 0.1W	current2_control_limit - current2_control_limit_cnt
32	controller_heat_dissipation1/100 단위크기는 0.1W	controller_heat_dissipation2/100 단위크기는 0.1W	current1_control_limit - current1_control_limit_cnt	current2_control_limit - current2_control_limit_cnt
33	(fault1_status >> 8) & 0x0001	모터1의 권선 온도(Twinding1) 단위크기는 0.1도	모터1의 권선 온도 리밧 (TempLimit1) 단위크기는 0.1도	current1_control_limit - current1_control_limit_cnt
34	(fault1_status >> 9) & 0x0001	모터2의 권선 온도(Twinding2) 단위크기는 0.1도	모터2의 권선 온도 리밧 (TempLimit2) 단위크기는 0.1도	current2_control_limit - current2_control_limit_cnt
35	LS1A_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 1000)	LS1B_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 2000)	LS1C_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 3000)	LS1D_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 + EXT_FAULT1_port * 5000 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 4000)
36	LS2A_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 1000)	LS2B_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 2000)	LS2C_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 3000)	LS2D_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 + EXT_FAULT2_port * 5000 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 4000)

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
37(모터1), 38(모터2)	positionx_plus_limit_over_flg * 1 + positionx_minus_limit_over_flg * 10 + motorx_emergency_stop_flg * 100 + motorx_emergency_stop_K_flg * 1000 + motorx_immediate_stop_action_flg * 10000	positionx_deceleration_at_limit_flg * 1 + speedx_deceleration_at_limit_flg * 10	speedx_cmd_old / SPEED_SCALE_FACTOR	모터x의 순간위치명령값 (positionx_set)을 4로 나눈값
39	모터1의 순간위치명령값 (position1_set) 단위크기는 1 encoder pulse임	모터1의 현재 위치값(position1) 단위크기는 1 encoder pulse임	모터2의 순간위치명령값 (position2_set) 단위크기는 1 encoder pulse임	모터2의 현재 위치값(position2) 단위크기는 1 encoder pulse임
43	fault_status	fault1_status	IRMCK201_fault_status	MOT_EN_N_port * 1000 + operation_mode
44	faultBufPointer	faultBuf[faultBufPointer]	faultBuf[faultBufPointer-1]	faultBuf[faultBufPointer-2]
45	fault_save_timer	faultBufPointer_save	faultBuf_save[faultBufPointer_save]	faultBuf_save[faultBufPointer_save-1]
46	fault_status	fault_save_timer	speed1_set	operation_mode
47	fault_status	fault1_status	IRMCK201_fault_status	DC_BUS_VOLT

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
48	HALL_Sensor 1_A_AD_value 2014년 2월 17일부터 (((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION & RJM_VER7_DRV8332 & BLDC_TWO_MOTOR_SENSOR)==1)&&(A NGLE_RESOLVING_METHOD!=0) 일때 는 필터링된 값	HALL_Sensor 1_B_AD_value 2014년 2월 17일부터 (((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION & RJM_VER7_DRV8332 & BLDC_TWO_MOTOR_SENSOR)==1)&&(A NGLE_RESOLVING_METHOD!=0) 일때 는 필터링된 값	HALL_Sensor 1_C_AD_value 2014년 2월 17일부터 (((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION & RJM_VER7_DRV8332 & BLDC_TWO_MOTOR_SENSOR)==1)&&(A NGLE_RESOLVING_METHOD!=0) 일때 는 필터링된 값	phase_angle1
49	HALL_Sensor 2_A_AD_value 2014년 2월 17일부터 (((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION & RJM_VER7_DRV8332 & BLDC_TWO_MOTOR_SENSOR)==1)&&(A NGLE_RESOLVING_METHOD!=0) 일때 는 필터링된 값	HALL_Sensor 2_B_AD_value 2014년 2월 17일부터 (((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION & RJM_VER7_DRV8332 & BLDC_TWO_MOTOR_SENSOR)==1)&&(A NGLE_RESOLVING_METHOD!=0) 일때 는 필터링된 값	HALL_Sensor 2_C_AD_value 2014년 2월 17일부터 (((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION & RJM_VER7_DRV8332 & BLDC_TWO_MOTOR_SENSOR)==1)&&(A NGLE_RESOLVING_METHOD!=0) 일때 는 필터링된 값	phase_angle2
50	DC_BUS_VOLT_filtered_at_power_ up	vfiltA[10]	DC_BUS_VOLT_slow_filtered	fault_status
51	total_current1_AD_filtered	motor1_current_max	current1_max_limit	fault_status
52	total_current2_AD_filtered	motor2_current_max	current2_max_limit	fault_status

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
53	motor1_current_max	current1_over_timer	current1_over_flg + current1_over_K_flg * 10	fault_status
54	motor2_current_max	current2_over_timer	current2_over_flg + current2_over_K_flg * 10	fault_status
55	current1_cmd	lq1_AD	motor1_moving_flg * 1000 + ldref1 + ldref1_by_field_weakening + (MOTORx_PA_end_processing_time_set - MOTOR1_PA_end_timer)	ld1_AD
56	current2_cmd	lq2_AD	motor2_moving_flg * 1000 + ldref2 + ldref2_by_field_weakening + (MOTORx_PA_end_processing_time_set - MOTOR2_PA_end_timer)	ld2_AD
60	TPC 초점구동의 경우는 ADC_RESULT2_reg>>2	vfiltA[2]	current1_cmd	speed1_cmd / SPEED_SCALE_FACTOR
60	자동화용 H/W 버전의 경우는 ADC_RESULT14_reg>>2 또는 ADC_RESULT6_reg>>2;	vfiltA[15] or vfiltA[11]	current1_cmd	speed1_cmd / SPEED_SCALE_FACTOR
61	TPC 초점구동의 경우는 ADC_RESULT10_reg>>2	vfiltA[5]	current2_cmd	speed2_cmd / SPEED_SCALE_FACTOR
61	Hall 센서형 전류센서를 사용하는 경우는 current_offset_measurement_timer	CURRENT1_SEN_P_at_power_up	CURRENT1_SEN_N_at_power_up	CURRENT2_SEN_P_at_power_up
62	current_offset_measurement_timer	CURRENT1_SEN_P_at_power_up	CURRENT1_SEN_N_at_power_up	TOTAL_CURRENT1_at_power_up

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
63	Hall 센서형 전류센서를 사용하는 경우는 CANRX0_Buf[250]	CANRX0_Buf[251]	CANRX0_Buf[252]	current1_AD
63	그외는 current_offset_measurement_timer	CURRENT2_SEN_P_at_power_up	CURRENT2_SEN_N_at_power_up	TOTAL_CURRENT2_at_power_up
64	current_offset_measurement_timer	CURRENT1_SEN_P	CURRENT1_SEN_N	total_current1_AD
65	Hall 센서형 전류센서를 사용하는 경우는 CURRENT1_SEN_P	CURRENT1_SEN_N	CURRENT2_SEN_P	CURRENT1_SEN_P + CURRENT1_SEN_N + CURRENT2_SEN_P - (CURRENT1_SEN_P_at_power_up + (CURRENT1_SEN_N_at_power_up + CURRENT2_SEN_P_at_power_up)
65	그외는 current_offset_measurement_timer	CURRENT2_SEN_P	CURRENT2_SEN_N	total_current2_AD
66	automatic_boot_up_timer	operation_mode	speed1_RPM	position1
67	automatic_boot_up_timer	phase_angle1	position1_cmd	position1
68	total_current1_AD_filtered	motor1_current_max	current1_max_limit (=max(current1_limit, current1_control_limit) * 1.25)	min(current1_limit, current1_control_limit) * 1.25
69	total_current2_AD_filtered	motor2_current_max	current2_max_limit (=max(current2_limit, current2_control_limit) * 1.25)	min(current2_limit, current2_control_limit) * 1.25

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
70	vfiltA[0] (모터1의 cos 입력 * 4)	vfiltA[1] (모터1의 sin 입력 * 4)	position1 - POSITION1_MID	Mg 명령에 의한 파라미터값이 0.0이고, INTERPOLATOR1_ERR_port가 있는 경우는 INTERPOLATOR1_ERR_port * 1000
70	vfiltA[0] (모터1의 cos 입력 * 4)	vfiltA[1] (모터1의 sin 입력 * 4)	position1 - POSITION1_MID	Mg 명령에 의한 파라미터값이 1.0인 경우는 position1_ENCODER - POSITION1_MID 단 2013년 6월 08일 이전 S/W 버전에서는 INTERPOLATOR1_ERR_port * 1000
70	vfiltA[0] (모터1의 cos 입력 * 4)	vfiltA[1] (모터1의 sin 입력 * 4)	position1 - POSITION1_MID	그외는 (position1_ENCODER - POSITION1_MID) * encoder_to_analog_sensor_scale_factor(Mg 명령에 의한 파라미터값) 단 2013년 6월 08일 이전 S/W 버전에서는 INTERPOLATOR1_ERR_port * 1000
71	vfiltA[3] (모터2의 cos 입력 * 4)	vfiltA[4] (모터2의 sin 입력 * 4)	position2 - POSITION2_MID	Mg 명령에 의한 파라미터값이 0이고, INTERPOLATOR2_ERR_port가 있는 경우는 INTERPOLATOR2_ERR_port * 1000
71	vfiltA[3] (모터2의 cos 입력 * 4)	vfiltA[4] (모터2의 sin 입력 * 4)	position2 - POSITION2_MID	Mg 명령에 의한 파라미터값이 1.0인 경우는 position2_ENCODER - POSITION2_MID 단 2013년 6월 08일 이전 S/W 버전에서는 INTERPOLATOR2_ERR_port * 1000
71	vfiltA[3] (모터2의 cos 입력 * 4)	vfiltA[4] (모터2의 sin 입력 * 4)	position2 - POSITION2_MID	그외는 (position2_ENCODER - POSITION2_MID) * encoder_to_analog_sensor_scale_factor(Mg 명령에 의한 파라미터값) 단 2013년 6월 08일 이전 S/W 버전에서는 INTERPOLATOR2_ERR_port * 1000

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
72	모터1의 전류명령값 (current1_cmd) 단위크기는 0.01A	모터1의 U상 전류의 A/D 변환값 (current1_AD) 단위크기는 0.01A	모터1의 V상 전류의 A/D 변환값 (current2_AD) 단위크기는 0.01A	모터1의 W상 전류의 A/D 변환값 (current3_AD) 단위크기는 0.01A
73	모터2의 전류명령값 (current2_cmd) 단위크기는 0.01A	모터2의 U상 전류의 A/D 변환값 (current4_AD) 단위크기는 0.01A	모터2의 V상 전류의 A/D 변환값 (current5_AD) 단위크기는 0.01A	모터2의 W상 전류의 A/D 변환값 (current6_AD) 단위크기는 0.01A
74	모터1의 현재 총전류의 A/D 변환값 (total_current1_AD) 단위크기는 0.01A	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current1_AD_filtered) 단위크기는 0.01A	모터2의 현재 총전류의 A/D 변환값 (total_current2_AD) 단위크기는 0.01A	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current2_AD_filtered) 단위크기는 0.01A
78	모터1의 전압진폭 Vamp1	모터1의 Id 전류 설정치 ldref1_by_field_weakening	모터1의 Iq 전류 리밋 lq1_limit	모터1의 Iq 전류 설정치 current1_cmd
79	모터2의 전압진폭 Vamp2	모터2의 Id 전류 설정치 ldref2_by_field_weakening	모터2의 Iq 전류 리밋 lq2_limit	모터2의 Iq 전류 설정치 current2_cmd
88	모터1의 03/0A번 제어모드에서 속도의 PID 공통 조정계인 * I 조정계인 * 1000	모터1의 03/0A번 제어모드에서 속도의 P 조정계인 * 1000	모터1의 03/0A번 제어모드에서 속도의 I 조정계인 * 1000	모터1의 03/0A번 제어모드에서 속도의 D 조정계인 * 1000
89	모터1의 03/0A번 제어모드에서 속도의 PID 공통 조정계인 * I 조정계인 * 1000	모터2의 03/0A번 제어모드에서 속도의 P 조정계인 * 1000	모터2의 03/0A번 제어모드에서 속도의 I 조정계인 * 1000	모터2의 03/0A번 제어모드에서 속도의 D 조정계인 * 1000
91	스텝모터의 경우는 lead_angle1_set	lead_angle1	lead_angle1_set - lead_angle1	0
91	그외는 current1_cmd	Iq1_AD	current1_cmd - Iq1_AD	phase_angle1

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
92	스텝모터의 경우는 speed1_cmd_old / (7L * MOTOR1_NUMBER_of_POLE_PAIR)	(speed1_RPM_count * 600000L) / MOTOR1_PULSE_PER_REV_ENC	첫번째 변수 - 두번째 변수	motor1_error
92	그외는 speed1_long_cmd_old 또는 (speed1_cmd_old * 10) / SPEED_SCALE_FACTOR	speed1_RPM_10	첫번째 변수 - 두번째 변수	motor1_error
93	스텝모터의 경우는 (position1_set - POSITION1_MID) / MOTOR1_POSITION_SCALE_FACTOR	(position1 - POSITION1_MID) / MOTOR1_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor1_error
93	그외는 position1_set - POSITION1_MID	(position1 - POSITION1_MID) / MOTOR1_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor1_error
95	스텝모터의 경우는 lead_angle2_set	lead_angle2	lead_angle2_set - lead_angle2	0
95	그외는 current2_cmd	lq2_AD	current2_cmd - lq2_AD	phase_angle2
96	스텝모터의 경우는 speed2_cmd_old / (7L * MOTOR2_NUMBER_of_POLE_PAIR)	(speed2_RPM_count * 600000L) / MOTOR2_PULSE_PER_REV_ENC	첫번째 변수 - 두번째 변수	motor2_error
96	그외는 speed2_long_cmd_old 또는 (speed2_cmd_old * 10) / SPEED_SCALE_FACTOR	speed2_RPM_10	첫번째 변수 - 두번째 변수	motor2_error

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
97	스텝모터의 경우는 (position2_set - POSITION2_MID) / MOTOR2_POSITION_SCALE_FACTOR	(position2 - POSITION2_MID) / MOTOR2_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor2_error
97	그외는 position2_set - POSITION2_MID	(position2 - POSITION2_MID) / MOTOR2_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor2_error
98	모터1의 PID 공통 조정계인 * I 조정계인 * I 위치제어 추가조정 계인 * 1000	모터1의 P 조정계인 * 1000	모터1의 I 조정계인 * 1000	모터1의 D 조정계인 * 1000
99	모터2의 PID 공통 조정계인 * I 조정계인 * I 위치제어 추가조정 계인 * 1000	모터2의 P 조정계인 * 1000	모터2의 I 조정계인 * 1000	모터2의 D 조정계인 * 1000
101	모터1의 U상 PWM 출력값 (0 - PWM_PERIOD, PWM_PERIOD/2 일때 50% duty)	모터1의 V상 PWM 출력값 (0 - PWM_PERIOD, PWM_PERIOD/2 일때 50% duty)	모터1의 W상 PWM 출력값 (0 - PWM_PERIOD, PWM_PERIOD/2 일때 50% duty)	모터1의 Hall센서 A상 상승에지 기준 위상각 (0 - 16383), IRS2336을 사용한 경우는 fault 시에 10000을 더함

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
102	((RJM_VER7_ENCODER & IRF7779 & IRS2336)==1)&&(ABSOLUTE_ENC==3)인 경우는 eQEP1_POSLAT_reg_save >> 14	eQEP1_POSLAT_reg_save & 0x3fff	phase_angle1	(position1 - POSITION1_MID) / (MOTOR1_POSITION_SCALE_FACTOR * 4)
102	그외 (((RJM_VER7_ENCODER & IRF7779)==1) ((RJM_VER8 & IRF7779)==1))&&(ABSOLUTE_ENC==2)인 경우는 absENCA_duty PWM 절대각센서를 사용하는 경우로서 절대각센서 A의 low인 기간의 duty ratio (-800 - +800)	absENCB_duty PWM 절대각센서를 사용하는 경우로서 절대각센서 B의 low인 기간의 duty ratio (-7500 ~ +7500)	absPOSITION1>>2 PWM 절대각센서를 사용하는 경우로서 절대각 계산값 / 4 (4채배된 4096펄스의 엔코더와 각변위 단위가 같음)	(position1 - POSITION1_MID) / (MOTOR1_POSITION_SCALE_FACTOR * 4) PWM 절대각센서를 사용하는 경우로서 모터1의 각변위값 / 4 (4채배된 4096펄스의 엔코더 펄스 단위)
102	그외의 경우는 모터2의 U상 PWM 출력값 (0 - PWM_PERIOD, PWM_PERIOD/2 일때 50% duty)	모터2의 V상 PWM 출력값 (0 - PWM_PERIOD, PWM_PERIOD/2 일때 50% duty)	모터2의 W상 PWM 출력값 (0 - PWM_PERIOD, PWM_PERIOD/2 일때 50% duty)	모터2의 Hall센서 A상 상승에지 기준 위상각 (0 - 16383)
103	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 Hall 센서 A상 전압의 A/D 변환값 (0 - 4095)	모터1의 U상 출력 위상각 단위크기는 1 deg	cogging_torque_compensation1 0/4096/8192/16384이면 6.25/12.5/25% reduction
104	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 Hall 센서 A상 전압의 A/D 변환값 (0 - 4095)	모터2의 U상 출력 위상각 단위크기는 1 deg	cogging_torque_compensation2 0/4096/8192/16384이면 6.25/12.5/25% reduction
105	current1_cmd	Iq1	Vq1	Vd1
106	current2_cmd	Iq2	Vq2	Vd2

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
110	motor1_error	current1_cmd	$(\text{position1_set} - \text{POSITION1_MID}) - (\text{position1_at_1ms} - \text{POSITION1_MID}) / \text{MOTOR1_POSITION_SCALE_FACTOR}$	$(\text{position1_at_1ms} - \text{POSITION1_MID}) / \text{MOTOR1_POSITION_SCALE_FACTOR}$
111	motor2_error	current2_cmd	$(\text{position2_set} - \text{POSITION2_MID}) - (\text{position2_at_1ms} - \text{POSITION2_MID}) / \text{MOTOR2_POSITION_SCALE_FACTOR}$	$(\text{position2_at_1ms} - \text{POSITION2_MID}) / \text{MOTOR2_POSITION_SCALE_FACTOR}$
112	EPS의 경우는 모터1의 위치명령값 (position1_cmd)을 4로 나눈값	모터1의 순간위치명령값 (position1_set)을 4로 나눈값	모터1의 현재위치값(position1)을 4로 나눈값	모터1의 vehicle_driving_motor_speed
113	position1_cmd - position1_set	speed1_RPM (Hall IC only 제어의 경우는 필터링한 값)	speed1_current	speed1_residual
114	speed1 command	speed1_RPM (Hall IC only 제어의 경우는 필터링한 값)	current1_cmd	$(\text{MOTOR1_INERTIAL_RATIO} * (\text{long long})\text{motor1_acc_rate}) >> 12$
115	speed2 command	speed2_RPM (Hall IC only 제어의 경우는 필터링한 값)	current2_cmd	$(\text{MOTOR2_INERTIAL_RATIO} * (\text{long long})\text{motor2_acc_rate}) >> 12$
116	notch filter MOTOR1_uk1_0	notch filter MOTOR1_yk1_0	notch filter MOTOR2_uk1_0	notch filter MOTOR2_yk1_0

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
120	모터1의 순간속도명령값 단위크기는 RPM임	모터1의 실제속도값 단위크기는 RPM임	모터1의 속도에 따른 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터1의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임
121	모터1의 순간속도명령값 단위크기는 RPM임	모터1의 속도계산에 사용되는 1ms동안의 위치변화량 단위크기는 1ms당 펄스수임	모터1의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터1의 속도계산에 사용되는 1ms동안의 위치변화량을 제한한 값 (이값으로 속도계산을 실시함) 단위크기는 1ms당 펄스수임
122	모터2의 순간속도명령값 단위크기는 RPM임	모터2의 실제속도값 단위크기는 RPM임	모터2의 속도에 따른 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터2의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임
123	모터2의 순간속도명령값 단위크기는 RPM임	모터2의 속도계산에 사용되는 1ms동안의 위치변화량 단위크기는 1ms당 펄스수임	모터2의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터2의 속도계산에 사용되는 1ms동안의 위치변화량을 제한한 값 (이값으로 속도계산을 실시함) 단위크기는 1ms당 펄스수임
130	ADC_RESULT0_reg	ADC_RESULT1_reg	ADC_RESULT2_reg	ADC_RESULT3_reg
131	ADC_RESULT4_reg	ADC_RESULT5_reg	ADC_RESULT6_reg	ADC_RESULT7_reg
132	ADC_RESULT8_reg	ADC_RESULT9_reg	ADC_RESULT10_reg	ADC_RESULT11_reg
133	ADC_RESULT12_reg	ADC_RESULT13_reg	ADC_RESULT14_reg	ADC_RESULT15_reg
134	motor1_home_searching_status	position1 - POSITION1_MID	position1_set - POSITION1_MID	position1_at_1ms - POSITION1_MID
135	motor2_home_searching_status	position2 - POSITION2_MID	position2_set - POSITION2_MID	position2_at_1ms - POSITION2_MID
508	current1_cmd	lq1	current1_cmd - lq1	phase_angle1_old
509	current2_cmd	lq2	current2_cmd - lq2	phase_angle2_old

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER7인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0x800+32	모터1의 순간위치명령값 (position1_set) 단위크기는 1 encoder pulse임	모터1의 위치제어 오차 단위크기는 1 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함)	모터1의 위치제어 출력값 (01번 제어모드에서는 PWM값, 02번 제어모드에서는 전류명령값, 03번 제어모드에서는 속도명령값, 스테핑모터에서는 lead angle 명령값)	모터1의 속도명령값 (03/0A번 제어모드일때로서 feedforward 성분 포함) 단위크기는 RPM*10
	모터1의 실제 속도값 단위크기는 RPM*10	모터1의 속도제어오차 (03/0A번 제어모드일때) 단위크기는 RPM*10	모터1의 전류명령값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 실제전류값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
0x800+33	모터2의 순간위치명령값 (position2_set)	모터2의 위치제어 오차	모터2의 위치제어 출력값	모터2의 속도명령값 (03/0A번 제어모드일때로서 feedforward 성분 포함)
	모터2의 실제 속도값	모터2의 속도제어오차 (03/0A번 제어모드일때)	모터2의 전류명령값	모터2의 실제전류값

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0	모터1의 순간위치명령값 (position1_set)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S1/Ts/Tu 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임	모터1의 현재 위치값(position1)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S1/Ts/Tu 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임	모터2의 순간위치명령값 (position2_set)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S2/Tt/Tv 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임	모터2의 현재 위치값(position2)을 4로 나눈값 단위크기는 4 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함), 단 S2/Tt/Tv 명령에 의한 작동중일 때는 단위크기는 1 encoder pulse임
1	모터1의 순간속도명령값 (speed1_cmd_old / SPEED_SCALE_FACTOR, 03번 모드에서는 (speed1_long_cmd + speed1_long_cmd_feedforward) / 10, 0A번 모드에서는 speed1_long_cmd_old / 10) 단위크기는 RPM	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터2의 순간속도명령값 (speed2_cmd_old / SPEED_SCALE_FACTOR), 03번 모드에서는 (speed2_long_cmd + speed2_long_cmd_feedforward) / 10, 0A번 모드에서는 speed2_long_cmd_old / 10) 단위크기는 RPM	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM
2	모터1의 전류명령값 (current1_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 권선전류의 A/D 변환값 (current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 전류명령값 (current2_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 권선전류의 A/D 변환값 (current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
3	모터1의 권선전류의 A/D 변환값 (current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 권선전류의 A/D 변환값을 filtering한 값 (current1_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 권선전류의 A/D 변환값 (current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 권선전류의 A/D 변환값을 filtering한 값 (current2_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당

Qx0; 에 대한 return 값은 위치센서로 analog 또는 digital potentiometer을 사용하는 경우에 /4를 하지 않고 원래값을 보내줌.
또 S1/S2/Ts/Tt/Tu/Tv 명령에 의한 구동일 경우에도 /4를 하지 않고 원래값을 보내줌.

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
4	모터1의 현재 총전류의 A/D 변환값 (total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current1_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 총전류의 A/D 변환값 (total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current2_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
5	LOW_RESOLUTION_ENCODER==1일 때 모터1의 엔코더 A상 신호 * 4095	모터1의 엔코더 B상 신호 * 2000	phase_angle1 엔코더 A/B상 한사이클당 16384	speed1_RPM 단위크기는 RPM
6	LOW_RESOLUTION_ENCODER==1일 때 모터2의 엔코더 A상 신호 * 4095	모터2의 엔코더 B상 신호 * 2000	phase_angle2 엔코더 A/B상 한사이클당 16384	speed2_RPM 단위크기는 RPM
7	LOW_RESOLUTION_ENCODER==1일 때 모터1의 정속구간에서의 1ms당 엔코더 펄스수*10000 (speed1_set)	모터1의 1ms당 phase_angle1 변화량 (speed1_RPM_1ms_for_save_alt)	모터2의 정속구간에서의 1ms당 엔코더 펄스수*10000 (speed2_set)	모터2의 1ms당 phase_angle2 변화량 (speed2_RPM_1ms_for_save_alt)
8	LOW_RESOLUTION_ENCODER==1일 때 Hall_IC1_per iod[0]	Hall_IC1_per iod[1]	Hall_IC1_per iod[2]	Hall_IC1_per iod[3]

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
10	모터1의 입력 power(Pin1) 단위크기는 0.1W	모터1의 출력 power(Pout1) 단위크기는 0.1W	모터1의 현재 속도값 (speed1_RPM)	모터1의 효율 단위크기는 1%
11	모터2의 입력 power(Pin2) 단위크기는 0.1W	모터2의 출력 power(Pout2) 단위크기는 0.1W	모터2의 현재 속도값 (speed2_RPM)	모터2의 효율 단위크기는 1%
12	모터1의 입력 power(Pin1) 단위크기는 0.1W	모터1의 출력 토크(Tout1) 단위크기는 mN	모터1의 현재 속도값 (speed1_RPM)	모터1의 효율 단위크기는 1%
13	모터2의 입력 power(Pin2) 단위크기는 0.1W	모터2의 출력 토크(Tout2) 단위크기는 mN	모터2의 현재 속도값 (speed2_RPM)	모터2의 효율 단위크기는 1%
14	공급전압 65 가 1V 에 해당함.	모터1의 현재 총전류의 A/D 변환 값(total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 부호있는 전류진폭 2/5/10mΩ의 경우 81.9/204.75/409.5 가 전류진폭 1A에 해당
15	공급전압 65 가 1V 에 해당함.	모터2의 현재 총전류의 A/D 변환 값(total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 부호있는 전류진폭 2/5/10mΩ의 경우 81.9/204.75/409.5 가 전류진폭 1A에 해당
16	모터1의 손실 power(Pin1-Pout1) 단위크기는 0.1W	모터1의 권선 온도(Twinding1) 단위크기는 0.1도	모터1의 하우징 온도(Thousing1) 단위크기는 0.1도	모터1의 분위기 온도(Tambient1) 단위크기는 0.1도
17	모터2의 손실 power(Pin2-Pout2) 단위크기는 0.1W	모터2의 권선 온도(Twinding2) 단위크기는 0.1도	모터2의 하우징 온도(Thousing2) 단위크기는 0.1도	모터2의 분위기 온도(Tambient2) 단위크기는 0.1도
20	motor1_control_service_time + motor2_control_service_time * 100 단위크기는 1ms의 1%	mainLoopStep_cnt 0-9 범위의 값으로서 단위크기는 1ms	mainLoop_execution_time 단위크기는 1ms의 1%	maximum_interrupt_service_time 단위크기는 PWM 주기의 1%

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
21	motor1_error 위치제어 모드에서의 단위크기는 1 encoder pulse 속도제어 모드에서의 단위크기는 0.1 rpm 전류제어 모드에서는 해당없음.	motor1_integral 위치제어 모드에서의 단위크기는 1 encoder pulse-msec 속도제어 모드에서의 단위크기는 0.1 rpm-sec 전류제어 모드에서는 해당없음.	motor1_derivative 위치제어 모드에서의 단위크기는 1 encoder pulse/ms 속도제어 모드에서의 단위크기는 0.1 rpm/ms 전류제어 모드에서는 해당없음.	current1_cmd (단 01번 위치제어 모드에서는 motor1_PWM_set, 03번 위치제어모드에서는 speed1_long_cmd) 위치/속도제어 모드에서의 단위크기는 AD 변환된 전류값의 단위크기와 같음 전류제어 모드에서는 해당없음.
22	motor2_error	motor2_integral	motor2_derivative	current2_cmd (단 01번 위치제어 모드에서는 motor2_PWM_set, 03번 위치제어모드에서는 speed2_long_cmd)
23	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilTA[10] - DC_BUS_VOLT_filtered_at_power_up 65 가 1V 에 해당함	speed1_RPM 단위크기는 RPM	vfilTA[10] 65 가 1V 에 해당함	BRAKE_RELAY_port * 1000 1000 이면 on, 0 이면 off
23	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilTA[10] 65 가 1V 에 해당함	DC_BUS_VOLT_filtered_at_power_up 65 가 1V 에 해당함	speed1_RPM 단위크기는 RPM	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER7인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
24	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilta[10]	speed1_RPM	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 1000 이면 on, 0 이면 off
24	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilta[10]	전압오차 brake_action_error	브레이크저항 구동 PWM duty T1CMPR_reg	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off
25	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilta[10]	DC_BUS_VOLT_filtered_at_power_up	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 1000 이면 on, 0 이면 off
25	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilta[10] - DC_BUS_VOLT_filtered_at_power_up	vfilta[10] - DC_BUS_VOLT_slow_filtered	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off
30	(fault1_status >> 11) & 0x0001	controller_heat_dissipation1/100 단위크기는 0.1W	controller_heat_dissipation_limit/100 단위크기는 0.1W	current1_control_limit - current1_control_limit_cnt
31	(fault1_status >> 11) & 0x0001	controller_heat_dissipation2/100 단위크기는 0.1W	controller_heat_dissipation_limit/100 단위크기는 0.1W	current2_control_limit - current2_control_limit_cnt

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
32	controller_heat_dissipation1/100 단위크기는 0.1W	controller_heat_dissipation2/100 단위크기는 0.1W	current1_control_limit - current1_control_limit_cnt	current2_control_limit - current2_control_limit_cnt
33	(fault1_status >> 8) & 0x0001	모터1의 권선 온도(Twinding1) 단위크기는 0.1도	모터1의 권선 온도 리밋 (TempLimit1) 단위크기는 0.1도	current1_control_limit - current1_control_limit_cnt
34	(fault1_status >> 9) & 0x0001	모터2의 권선 온도(Twinding2) 단위크기는 0.1도	모터2의 권선 온도 리밋 (TempLimit2) 단위크기는 0.1도	current2_control_limit - current2_control_limit_cnt
35	LS1A_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 1000)	LS1B_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 2000)	LS1C_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 3000)	LS1D_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 + EXT_FAULT1_port * 5000 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 4000)
36	LS2A_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 1000)	LS2B_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 2000)	LS2C_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 3000)	LS2D_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 + EXT_FAULT2_port * 5000 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 4000)

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
37(모터1), 38(모터2)	positionx_plus_limit_over_flg * 1 + positionx_minus_limit_over_flg * 10 + motorx_emergency_stop_flg * 100 + motorx_emergency_stop_K_flg * 1000 + motorx_immediate_stop_action_flg * 10000	positionx_deceleration_at_limit_flg * 1 + speedx_deceleration_at_limit_flg * 10	speedx_cmd_old / SPEED_SCALE_FACTOR	모터x의 순간위치명령값 (positionx_set)을 4로 나눈값 위치센서로 analog 또는 digital potentiometer을 사용하는 경우에 /4를 하지 않고 원래값을 보내줌.
39	모터1의 순간위치명령값 (position1_set) 단위크기는 1 encoder pulse임	모터1의 현재 위치값 (position1) 단위크기는 1 encoder pulse임	모터2의 순간위치명령값 (position2_set) 단위크기는 1 encoder pulse임	모터2의 현재 위치값(position2) 단위크기는 1 encoder pulse임
43	fault_status	fault1_status	IRMCK201_fault_status	MOT_EN_N_port * 1000 + operation_mode
44	faultBufPointer	faultBuf[faultBufPointer]	faultBuf[faultBufPointer-1]	faultBuf[faultBufPointer-2]
45	fault_save_timer	faultBufPointer_save	faultBuf_save[faultBufPointer_save]	faultBuf_save[faultBufPointer_save-1]
46	fault_status	fault_save_timer	speed1_set	operation_mode
47	fault_status	fault1_status	IRMCK201_fault_status	DC_BUS_VOLT
50	DC_BUS_VOLT_filtered_at_power_up	vfiltA[10]	DC_BUS_VOLT_slow_filtered	fault_status
51	total_current1_AD_filtered	motor1_current_max	current1_max_limit	fault_status
52	total_current2_AD_filtered	motor2_current_max	current2_max_limit	fault_status

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
53	motor1_current_max	current1_over_timer	current1_over_flg + current1_over_K_flg * 10	fault_status
54	motor2_current_max	current2_over_timer	current2_over_flg + current2_over_K_flg * 10	fault_status
60	모터1의 외부 analog 입력 전압을 A/D 변환한 값	모터1의 외부 analog 입력 전압을 A/D 변환한 값의 offset을 제거하고 필터링한 값	current1_cmd 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	speed1_cmd / SPEED_SCALE_FACTOR 단위크기는 RPM
61	모터2의 외부 analog 입력 전압을 A/D 변환한 값	모터2의 외부 analog 입력 전압을 A/D 변환한 값의 offset을 제거하고 필터링한 값	current2_cmd 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	speed2_cmd / SPEED_SCALE_FACTOR 단위크기는 RPM
62	CURRENT1_SEN_P_at_power_up	TOTAL_CURRENT1_at_power_up	CURRENT2_SEN_P_at_power_up	TOTAL_CURRENT2_at_power_up
64	CURRENT1_SEN_P	total_current1_AD	CURRENT2_SEN_P	total_current2_AD
66	automatic_boot_up_timer	operation_mode	speed1_RPM	position1
67	automatic_boot_up_timer	operation_mode	speed2_RPM	position2
68	total_current1_AD_filtered	motor1_current_max	current1_max_limit (=max(current1_limit, current1_control_limit) * 1.25)	min(current1_limit, current1_control_limit) * 1.25
69	total_current2_AD_filtered	motor2_current_max	current2_max_limit (=max(current2_limit, current2_control_limit) * 1.25)	min(current2_limit, current2_control_limit) * 1.25

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
70	vfiltA[0] (모터1의 cos 입력 * 4)	vfiltA[1] (모터1의 sin 입력 * 4)	position1 - POSITION1_MID	position1_ENCODER - POSITION1_MID
71	vfiltA[3] (모터2의 cos 입력 * 4)	vfiltA[4] (모터2의 sin 입력 * 4)	position2 - POSITION2_MID	position2_ENCODER - POSITION2_MID
72	모터1의 전류명령값 (current1_cmd) 단위크기는 0.01A	모터1의 권선전류의 A/D 변환값 (current1_AD) 단위크기는 0.01A	모터2의 전류명령값 (current2_cmd) 단위크기는 0.01A	모터2의 권선전류의 A/D 변환값 (current2_AD) 단위크기는 0.01A
73	모터1의 권선전류의 A/D 변환값 (current1_AD) 단위크기는 0.01A	모터1의 권선전류의 A/D 변환값을 filtering한 값 (current1_AD_filtered) 단위크기는 0.01A	모터2의 권선전류의 A/D 변환값 (current2_AD) 단위크기는 0.01A	모터2의 권선전류의 A/D 변환값을 filtering한 값 (current2_AD_filtered) 단위크기는 0.01A
74	모터1의 현재 총전류의 A/D 변환값 (total_current1_AD) 단위크기는 0.01A	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current1_AD_filtered) 단위크기는 0.01A	모터2의 현재 총전류의 A/D 변환값 (total_current2_AD) 단위크기는 0.01A	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current2_AD_filtered) 단위크기는 0.01A
88	모터1의 03/0A번 제어모드에서 속도의 PID 공통 조정계인 * I 조정계인 * 1000	모터1의 03/0A번 제어모드에서 속도의 P 조정계인 * 1000	모터1의 03/0A번 제어모드에서 속도의 I 조정계인 * 1000	모터1의 03/0A번 제어모드에서 속도의 D 조정계인 * 1000
89	모터1의 03/0A번 제어모드에서 속도의 PID 공통 조정계인 * I 조정계인 * 1000	모터2의 03/0A번 제어모드에서 속도의 P 조정계인 * 1000	모터2의 03/0A번 제어모드에서 속도의 I 조정계인 * 1000	모터2의 03/0A번 제어모드에서 속도의 D 조정계인 * 1000

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
91	current1_cmd	current1_AD	current1_cmd - current1_AD	0
92	speed1_long_cmd_old 또는 (speed1_cmd_old * 10) / SPEED_SCALE_FACTOR	speed1_RPM_10	첫번째 변수 - 두번째 변수	motor1_error
93	position1_set - POSITION1_MID	(position1 - POSITION1_MID) / MOTOR1_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor1_error
95	current2_cmd	current2_AD	current2_cmd - current2_AD	0
96	speed2_long_cmd_old 또는 (speed2_cmd_old * 10) / SPEED_SCALE_FACTOR	speed2_RPM_10	첫번째 변수 - 두번째 변수	motor2_error
97	position2_set - POSITION2_MID	(position2 - POSITION2_MID) / MOTOR2_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor2_error
98	모터1의 PID 공통 조정게인 * I 조정게인 * I 위치제어 추 가조정게인 * 1000	모터1의 P 조정게인 * 1000	모터1의 I 조정게인 * 1000	모터1의 D 조정게인 * 1000
99	모터2의 PID 공통 조정게인 * I 조정게인 * I 위치제어 추 가조정게인 * 1000	모터2의 P 조정게인 * 1000	모터2의 I 조정게인 * 1000	모터2의 D 조정게인 * 1000
111	BRAKE 제어기의 경우는 모터1의 속도값(speed1_RPM)	모터1의 전류값(current1_AD)	모터1의 speed1_cmd_old / SPEED_SCALE_FACTOR	모터1의 current1_control_limit
112	BRAKE 제어기의 경우는 모터1의 속도값(speed1_RPM)	모터1의 전류값(current1_AD)	모터1의 포텐시오미터 위치 A/D 값	모터1의 압력센서 A/D값
114	speed1 command	speed1_RPM (Hall IC only 제어 의 경우는 필터링한 값)	current1_cmd	(MOTOR1_INERTIAL_RATIO * (long long)motor1_acc_rate) >> 12
115	speed2 command	speed2_RPM (Hall IC only 제어 의 경우는 필터링한 값)	current2_cmd	(MOTOR2_INERTIAL_RATIO * (long long)motor2_acc_rate) >> 12
116	notch filter MOTOR1_uk1_0	notch filter MOTOR1_yk1_0	notch filter MOTOR2_uk1_0	notch filter MOTOR2_yk1_0

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER2/3/4/8인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
130	ADC_RESULT0_reg	ADC_RESULT1_reg	ADC_RESULT2_reg	ADC_RESULT3_reg
131	ADC_RESULT4_reg	ADC_RESULT5_reg	ADC_RESULT6_reg	ADC_RESULT7_reg
132	ADC_RESULT8_reg	ADC_RESULT9_reg	ADC_RESULT10_reg	ADC_RESULT11_reg
133	ADC_RESULT12_reg	ADC_RESULT13_reg	ADC_RESULT14_reg	ADC_RESULT15_reg
134	motor1_home_searching_status	position1 - POSITION1_MID	position1_set - POSITION1_MID	position1_at_1ms - POSITION1_MID
135	motor2_home_searching_status	position2 - POSITION2_MID	position2_set - POSITION2_MID	position2_at_1ms - POSITION2_MID
508	current1_cmd	offset을 제거한 전류 A/D 변환값	current1_cmd - offset을 제거한 전류 A/D 변환값	0
509	current2_cmd	offset을 제거한 전류 A/D 변환값	current2_cmd - offset을 제거한 전류 A/D 변환값	0
0x800+32	모터1의 순간위치명령값 (position1_set) 단위크기는 1 encoder pulse 임	모터1의 위치제어 오차 단위크기는 1 encoder pulse (엔코더를 안쓰는 경우는 16384 펄스의 엔코더로 간주함)	모터1의 위치제어 출력값 (01번 제어모드에서는 PWM값, 02번 제어모드에서는 전류명령값, 03번 제어모드에서는 속도명령값, 스테핑모터에서는 lead angle 명령값)	모터1의 속도명령값 (03/0A번 제어모드일때로서 feedforward 성분 포함) 단위크기는 RPM*10
	모터1의 실제 속도값 단위크기는 RPM*10	모터1의 속도제어오차 (03/0A번 제어모드일때) 단위크기는 RPM*10	모터1의 전류명령값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 실제전류값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
0x800+33	모터2의 순간위치명령값 (position2_set)	모터2의 위치제어 오차	모터2의 위치제어 출력값	모터2의 속도명령값 (03/0A번 제어모드일때로서 feedforward 성분 포함)
	모터2의 실제 속도값	모터2의 속도제어오차 (03/0A번 제어모드일때)	모터2의 전류명령값	모터2의 실제전류값

Qxddd1 에 의하여 전송되는 변수값 (전류앰프인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0	공급전압 단위크기는 0.01V	총전류 단위크기는 0.01A	전류명령값 단위크기는 0.01A	전류측정값 단위크기는 0.01A
2	전류명령값 56mV/A, x20B	전류측정값 56mV/A, x20B		
3	전류측정값 56mV/A, x20B	전류측정값을 필터링한 값 56mV/A, x20B		
4	전류측정값 (ADCRESULT0_reg>>4) 56mV/A, x20B	외부전류명령값 (ADCRESULT2_reg>>4)	전류 reference 값 (ADCRESULT8_reg>>4)	1.5V reference 값 (ADCRESULT9_reg>>4)
5	전류측정값 ((ADCRESULT0_reg>>4) - (ADCRESULT8_reg>>4)) 56mV/A, x20B	CURRENT1_SEN	CURRENT1_SEN_at_power_up	전류측정값
6	CMPR1_reg	CMPR2_reg	T1CMPR_reg	전류측정값 단위크기는 0.01A
7	공급전압을 필터링한 값 단위크기는 0.01V	전류명령값 단위크기는 0.01A	Iq1 단위크기는 0.01A	전류측정값 단위크기는 0.01A
8	공급전압을 필터링한 값 단위크기는 0.01V	NTC_resistance	ADCRESULT15_reg >> 4	temperature_FT_sensor
18	TOTAL_CURRENT_ON_PERIOD	TOTAL_CURRENT_PERIOD	DC_BUS_VOLT_ON_PERIOD	DC_BUS_VOLT_PERIOD
19	TOTAL_CURRENT_ON_PERIOD	TOTAL_CURRENT	DC_BUS_VOLT_ON_PERIOD	DC_BUS_VOLT
20	motor1_control_service_time + motor2_control_service_time * 100 단위크기는 1ms의 1%	mainLoopStep_cnt 0-9 범위의 값으로서 단위크기는 1ms	mainLoop_execution_time 단위크기는 1ms의 1%	maximum_interrupt_service_time 단위크기는 PWM 주기의 1%

Qxddd1 에 의하여 전송되는 변수값 (전류앰프인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
23	vfiltA[10] 단위크기는 0.01V	DC_BUS_VOLT_filtered_at_power_up 단위크기는 0.01V	전류측정값 단위크기는 0.01A	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000
24	vfiltA[10] 단위크기는 0.01V	brake_action_error	T1CMPR_reg	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000
25	vfiltA[10] - DC_BUS_VOLT_filtered_at_power_up 단위크기는 0.01V	vfiltA[10] - DC_BUS_VOLT_slow_filtered 단위크기는 0.01V	power_resistor_ON_timer	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 1000/2000 이면 on, 0 이면 off
50	DC_BUS_VOLT_filtered_at_power_up 단위크기는 0.01V	vfiltA[10] 단위크기는 0.01V	DC_BUS_VOLT_slow_filtered 단위크기는 0.01V	fault_status
51	MOSFET_BRAKEDOWN_VOLTAGE	MOSFET_GATE_SHUTDOWN_HIGH_VOLTAGE	DC_BUS_VOLT_filtered_at_power_up 단위크기는 0.01V	vfiltA[10] 단위크기는 0.01V
52	current1_absolute_limit 단위크기는 0.01A	current1_max_limit 단위크기는 0.01A	current1_control_limit 단위크기는 0.01A	current1_AD 단위크기는 0.01A
53	motor1_current_max	current1_over_timer	current1_over_flg + current1_over_K_flg * 10	fault_status
68	total_current1_AD_filtered	motor1_current_max	current1_max_limit (=max(current1_limit, current1_control_limit) * 1.25)	min(current1_limit, current1_control_limit) * 1.25

Qxddd1 에 의하여 전송되는 변수값 (전류앰프인 경우)

RJM_VER2/3/4/8인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
80	current1_cmd	lq1	(PWM_PERIOD>>2) - CMPR1_reg	(PWM_PERIOD>>2) - CMPR2_reg
81	current1_cmd	current1_error	motor1_integral_INT>>8	motor1_PWM_set>>2
82	current1_cmd	lq1	motor1_PWM_set>>2	motor1_duty_compensation>>2
83	current1_cmd_ext	lq1	motor1_PWM_set>>2	motor1_duty_compensation>>2
130	ADC_RESULT0_reg	ADC_RESULT1_reg	ADC_RESULT2_reg	ADC_RESULT3_reg
131	ADC_RESULT4_reg	ADC_RESULT5_reg	ADC_RESULT6_reg	ADC_RESULT7_reg
132	ADC_RESULT8_reg	ADC_RESULT9_reg	ADC_RESULT10_reg	ADC_RESULT11_reg
133	ADC_RESULT12_reg	ADC_RESULT13_reg	ADC_RESULT14_reg	ADC_RESULT15_reg

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0	모터1의 순간위치명령값 (position1_set)을 4로 나눈값 단위크기는 4/16384 바퀴임, 단 S1/Ts/Tu 명령에 의한 작동중일 때는 단위크기는 1/16384 바퀴임	모터1의 현재 위치값(position1)을 4로 나눈값 단위크기는 4/16384 바퀴임, 단 S1/Ts/Tu 명령에 의한 작동중일 때는 단위크기는 1/16384 바퀴임	모터2의 순간위치명령값 (position2_set)을 4로 나눈값 단위크기는 4/16384 바퀴임, 단 S2/Tt/Tv 명령에 의한 작동중일 때는 단위크기는 1/16384 바퀴임	모터2의 현재 위치값(position2)을 4로 나눈값 단위크기는 4/16384 바퀴임, 단 S2/Tt/Tv 명령에 의한 작동중일 때는 단위크기는 1/16384 바퀴임
1	모터1의 순간속도명령값 (speed1_cmd_old / (70L * MOTOR1_NUMBER_of_POLE_PAIR)) 단위크기는 RPM	모터1의 현재 속도값 ((speed1_RPM_count * 60000L) / MOTOR1_PULSE_PER_REV_ENC) 단위크기는 RPM	모터2의 순간속도명령값 (speed2_cmd_old / (70L * MOTOR2_NUMBER_of_POLE_PAIR)) 단위크기는 RPM	모터2의 현재 속도값 ((speed2_RPM_count * 60000L) / MOTOR2_PULSE_PER_REV_ENC) 단위크기는 RPM
2	모터1의 A상 전류명령값 (current1_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 A상 권선전류의 A/D 변환값(current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 B상 전류명령값 (current2_cmd), 단 3/5상인 경우는 current2_AD 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 B상 권선전류의 A/D 변환값(current2_AD), 단 3/5상인 경우는 current3_AD 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
3	모터1의 A상 권선전류의 A/D 변환값(current1_AD), 단 3상일때는 current2_cmd, 단 5상일때는 current1_cmd 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 A상 권선전류의 A/D 변환값을 filtering한 값 (current1_AD_filtered), 단 3/5상일때는 current4_AD 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 B상 권선전류의 A/D 변환값(current2_AD), 단 3/5상일때는 current5_AD 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 B상 권선전류의 A/D 변환값을 filtering한 값 (current2_AD_filtered), 단 3상일때는 current6_AD, 단 5상일때는 Iq1_AD 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
4	모터1의 현재 총전류의 A/D 변환값(total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current1_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 총전류의 A/D 변환값(total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current2_AD_filtered) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
7	모터2의 A상(RJM_VER6는 모터1의 C상) 전류명령 (current3_cmd), 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 A상(RJM_VER6는 모터1의 C상) 권선전류의 A/D 변환값 (current3_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 B상(RJM_VER6는 모터2의 A상) 전류명령값(current4_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 B상(RJM_VER6는 모터2의 A상) 권선전류의 A/D 변환값 (current4_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
8	RJM_VER6의 경우는 모터2의 B상 전류명령값 (current5_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 B상 권선전류의 A/D 변환값(current5_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 C상 전류명령값 (current6_cmd) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 C상 권선전류의 A/D 변환값(current6_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
8	RJM_VER5의 경우는 모터1의 순간위치명령값 (position1_set)을 4로 나눈값 16384 일때 한바퀴임	모터1의 엔코더에 의한 현재 위치값(position1_ENC)을 한바퀴당 16384로 환산한 후에 4로 나눈값 16384 일때 한바퀴임	모터2의 순간위치명령값 (position2_set)을 4로 나눈값 16384 일때 한바퀴임	모터2의 엔코더에 의한 현재 위치값(position2_ENC)을 한바퀴당 16384로 환산한 후에 4로 나눈값 16384 일때 한바퀴임
14	공급전압 65 가 1V 에 해당함.	모터1의 현재 총전류의 A/D 변환값(total_current1_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 현재 속도값 (speed1_RPM) 단위크기는 RPM	모터1의 부호있는 전류진폭 2/5/10mΩ의 경우 81.9/204.75/409.5 가 전류진폭 1A에 해당
15	공급전압 65 가 1V 에 해당함.	모터2의 현재 총전류의 A/D 변환값(total_current2_AD) 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 현재 속도값 (speed2_RPM) 단위크기는 RPM	모터2의 부호있는 전류진폭 2/5/10mΩ의 경우 81.9/204.75/409.5 가 전류진폭 1A에 해당
20	motor1_control_service_time + motor2_control_service_time * 100 단위크기는 1ms의 1%	mainLoopStep_cnt 0-9 범위의 값으로서 단위크기는 1ms	mainLoop_execution_time 단위크기는 1ms의 1%	maximum_interrupt_service_time 단위크기는 PWM 주기의 1%

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
21	motor1_error 02번 위치제어 모드에서의 단위 크기는 1 encoder pulse	motor1_integral 02번 위치제어 모드에서의 단위 크기는 1 encoder pulse-msec	motor1_derivative 02번 위치제어 모드에서의 단위 크기는 1 encoder pulse/ms	03번 작동모드일 때는 speed1_long_cmd 그외는 lead_angle1_set 4096일때 90도
22	motor2_error	motor2_integral	motor2_derivative	03번 작동모드일 때는 speed2_long_cmd 그외는 lead_angle2_set
23	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfiltA[10] - DC_BUS_VOLT_filtered_at_power_up 65 가 1V 에 해당함	speed1_RPM 단위크기는 RPM	vfiltA[10] 65 가 1V 에 해당함	BRAKE_RELAY_port * 1000 + DC_DC_converter_voltage_transient_timer 1000 이면 on, 0 이면 off
23	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfiltA[10] 65 가 1V 에 해당함	DC_BUS_VOLT_filtered_at_power_up 65 가 1V 에 해당함	speed1_RPM 단위크기는 RPM	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 + DC_DC_converter_voltage_transient_timer 1000/2000 이면 on, 0 이면 off
24	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfiltA[10]	speed1_RPM	(1 - CH_PUMP_EN_N_port) * 500 + power_resistor_ON_timer	BRAKE_RELAY_port * 1000 + DC_DC_converter_voltage_transient_timer 1000 이면 on, 0 이면 off
24	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfiltA[10]	전압오차 brake_action_error	브레이크저항 구동 PWM duty T1CMPR_reg	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 + DC_DC_converter_voltage_transient_timer 1000/2000 이면 on, 0 이면 off

RJM_VER5/6인 경우

dddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
25	2013년 3월 3일 이전 또는 TMS320F28334 S/W 버전 vfilTA[10]	DC_BUS_VOLT_filtered_at_power_up	$(1 - CH_PUMP_EN_N_port) * 500 + power_resistor_ON_timer$	BRAKE_RELAY_port * 1000 + DC_DC_converter_voltage_transient_timer 1000 이면 on, 0 이면 off
25	2013년 3월 3일자 TMS320F2811 S/W 버전부터 vfilTA[10] - DC_BUS_VOLT_filtered_at_power_up	vfilTA[10] - DC_BUS_VOLT_slow_filtered	$(1 - CH_PUMP_EN_N_port) * 500 + power_resistor_ON_timer$	BRAKE_RELAY_port * 1000 + BRAKE_RELAY_PWM_ON_flg * 2000 + DC_DC_converter_voltage_transient_timer 1000/2000 이면 on, 0 이면 off
26	스텝핑모터로서 2013년 6월 16일 이후 S/W 버전으로서 2채널 자동화 버전의 경우 vfilTA[18] - DC_BUS_VOLT_filtered_at_power_up 65 가 1V 에 해당함	speed2_RPM 단위크기는 RPM	vfilTA[18] 65 가 1V 에 해당함	BRAKE_RELAY2_port * 1000 + DC_DC_converter_voltage_transient_timer 1000 이면 on, 0 이면 off
26	스텝핑모터로서 2013년 6월 16일 이후 S/W 버전으로서 그외의 경우 vfilTA[10] - DC_BUS_VOLT_filtered_at_power_up 65 가 1V 에 해당함	speed2_RPM 단위크기는 RPM	vfilTA[10] 65 가 1V 에 해당함	BRAKE_RELAY_port * 1000 + DC_DC_converter_voltage_transient_timer 1000 이면 on, 0 이면 off

RJM_VER5/6인 경우				
dddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
30	(fault1_status >> 11) & 0x0001	controller_heat_dissipation1/100 단위크기는 0.1W	controller_heat_dissipation_limit/100 단위크기는 0.1W	current1_control_limit - current1_control_limit_cnt
31	(fault1_status >> 11) & 0x0001	controller_heat_dissipation2/100 단위크기는 0.1W	controller_heat_dissipation_limit/100 단위크기는 0.1W	current2_control_limit - current2_control_limit_cnt
32	controller_heat_dissipation1/100 단위크기는 0.1W	controller_heat_dissipation2/100 단위크기는 0.1W	current1_control_limit - current1_control_limit_cnt	current2_control_limit - current2_control_limit_cnt
33	(fault1_status >> 8) & 0x0001	모터1의 권선 온도(Twinding1) 단위크기는 0.1도	모터1의 권선 온도 리밧 (TempLimit1) 단위크기는 0.1도	current1_control_limit - current1_control_limit_cnt
34	(fault1_status >> 9) & 0x0001	모터2의 권선 온도(Twinding2) 단위크기는 0.1도	모터2의 권선 온도 리밧 (TempLimit2) 단위크기는 0.1도	current2_control_limit - current2_control_limit_cnt

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
35	LS1A_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 1000)	LS1B_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 2000)	LS1C_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 3000)	LS1D_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 + EXT_FAULT1_port * 5000 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 4000)
36	LS2A_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 1000)	LS2B_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 2000)	LS2C_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 3000)	LS2D_port에 작용하는 전압의 A/D 변환값 또는 ON/OFF 입력값 + EXT_FAULT2_port * 5000 (analog의 경우는 4095 일때 3V, ON/OFF의 경우는 0 또는 4000)
37(모터1), 38(모터2)	positionx_plus_limit_over_flg * 1 + positionx_minus_limit_over_flg * 10 + motorx_emergency_stop_flg * 100 + motorx_emergency_stop_K_flg * 1000 + motorx_immediate_stop_action_flg * 10000	positionx_deceleration_at_limit_flg * 1 + speedx_deceleration_at_limit_flg * 10	speedx_cmd_old / SPEED_SCALE_FACTOR	모터x의 순간위치명령값 (positionx_set)을 4로 나눈값

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
39	모터1의 순간위치명령값 (position1_set) 단위크기는 1 encoder pulse임	모터1의 현재 위치값(position1) 단위크기는 1 encoder pulse임	모터2의 순간위치명령값 (position2_set) 단위크기는 1 encoder pulse임	모터2의 현재 위치값(position2) 단위크기는 1 encoder pulse임
43	fault_status	fault1_status	IRMCK201_fault_status	MOT_EN_N_port * 1000 + operation_mode
44	faultBufPointer	faultBuf[faultBufPointer]	faultBuf[faultBufPointer-1]	faultBuf[faultBufPointer-2]
45	fault_save_timer	faultBufPointer_save	faultBuf_save[faultBufPointer_save]	faultBuf_save[faultBufPointer_save-1]
46	fault_status	fault_save_timer	speed1_set	operation_mode
47	fault_status	fault1_status	IRMCK201_fault_status	DC_BUS_VOLT
50	DC_BUS_VOLT_filtered_at_power_up	vfilTA[10]	DC_BUS_VOLT_slow_filtered	fault_status
51	total_current1_AD_filtered	motor1_current_max	current1_max_limit	fault_status
52	total_current2_AD_filtered	motor2_current_max	current2_max_limit	fault_status
53	motor1_current_max	current1_over_timer	current1_over_flg + current1_over_K_flg * 10	fault_status
54	motor2_current_max	current2_over_timer	current2_over_flg + current2_over_K_flg * 10	fault_status
60	모터1의 외부 analog 입력 전압을 A/D 변환한 값	모터1의 외부 analog 입력 전압을 A/D 변환한 값의 offset을 제거하고 필터링한 값	current1_cmd 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	speed1_cmd / SPEED_SCALE_FACTOR 단위크기는 RPM
61	모터2의 외부 analog 입력 전압을 A/D 변환한 값	모터2의 외부 analog 입력 전압을 A/D 변환한 값의 offset을 제거하고 필터링한 값	current2_cmd 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	speed2_cmd / SPEED_SCALE_FACTOR 단위크기는 RPM

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
62	CURRENT1_SEN_P_at_power_up	CURRENT1_SEN_N_at_power_up	CURRENT2_SEN_P_at_power_up	CURRENT2_SEN_N_at_power_up
64	CURRENT1_SEN_P	CURRENT1_SEN_N	CURRENT2_SEN_P	CURRENT2_SEN_N
66	automatic_boot_up_timer	operation_mode	speed1_RPM	position1
67	automatic_boot_up_timer	operation_mode	speed2_RPM	position2
68	total_current1_AD_filtered	motor1_current_max	current1_max_limit (=max(current1_limit, current1_control_limit) * 1.25)	min(current1_limit, current1_control_limit) * 1.25
69	total_current2_AD_filtered	motor2_current_max	current2_max_limit (=max(current2_limit, current2_control_limit) * 1.25)	min(current2_limit, current2_control_limit) * 1.25
70	vfiltA[0] (모터1의 cos 입력 * 4)	vfiltA[1] (모터1의 sin 입력 * 4)	position1 - POSITION1_MID	자동화버전 스텝핑모터의 경우 INTERPOLATOR1_ERR_port * 1000 그외는 position1_ENCODER - POSITION1_MID
71	vfiltA[3] (모터2의 cos 입력 * 4)	vfiltA[4] (모터2의 sin 입력 * 4)	position2 - POSITION2_MID	자동화버전 스텝핑모터의 경우 INTERPOLATOR2_ERR_port * 1000 그외는 position2_ENCODER - POSITION2_MID

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
72	모터1의 A상 전류명령값 (current1_cmd) 단위크기는 0.01A	모터1의 A상 권선전류의 A/D 변환값(current1_AD) 단위크기는 0.01A	모터1의 B상 전류명령값 (current2_cmd), 단 3/5상일때는 current2_AD 단위크기는 0.01A	모터1의 B상 권선전류의 A/D 변환값(current2_AD), 단 3/5상일때는 current3_AD 단위크기는 0.01A
73	모터1의 A상 권선전류의 A/D 변환값(current1_AD), 단 3상일때는 current2_cmd, 단 5상일때는 current1_cmd 단위크기는 0.01A	모터1의 A상 권선전류의 A/D 변환값을 filtering한 값 (current1_AD_filtered), 단 3/5상일때는 current4_AD 단위크기는 0.01A	모터1의 B상 권선전류의 A/D 변환값(current2_AD), 단 3/5상일때는 current5_AD 단위크기는 0.01A	모터1의 B상 권선전류의 A/D 변환값을 filtering한 값 (current2_AD_filtered), 단 3상일때는 current6_AD, 단 5상일때는 Iq1_AD 단위크기는 0.01A
74	모터1의 현재 총전류의 A/D 변환값(total_current1_AD) 단위크기는 0.01A	모터1의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current1_AD_filtered) 단위크기는 0.01A	모터2의 현재 총전류의 A/D 변환값(total_current2_AD) 단위크기는 0.01A	모터2의 현재 총전류의 A/D 변환값을 필터링한 값 (total_current2_AD_filtered) 단위크기는 0.01A
91	lead_angle1_set	lead_angle1	lead_angle1_set - lead_angle1	0
92	speed1_cmd_old / (7L * MOTOR1_NUMBER_of_POLE_PAIR)	(speed1_RPM_count * 600000L) / MOTOR1_PULSE_PER_REV_ENC	첫번째 변수 - 두번째 변수	motor1_error
93	(position1_set - POSITION1_MID) / MOTOR1_POSITION_SCALE_FACTOR	(position1 - POSITION1_MID) / MOTOR1_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor1_error
95	lead_angle2_set	lead_angle2	lead_angle2_set - lead_angle2	0
96	speed2_cmd_old / (7L * MOTOR2_NUMBER_of_POLE_PAIR)	(speed2_RPM_count * 600000L) / MOTOR2_PULSE_PER_REV_ENC	첫번째 변수 - 두번째 변수	motor2_error
97	(position2_set - POSITION2_MID) / MOTOR2_POSITION_SCALE_FACTOR	(position2 - POSITION2_MID) / MOTOR2_POSITION_SCALE_FACTOR	첫번째 변수 - 두번째 변수	motor2_error

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
98	모터1의 PID 공통 조정게인 * I 조정게인 * I 위치제어 추가조정 게인 * 1000	모터1의 P 조정게인 * 1000	모터1의 I 조정게인 * 1000	모터1의 D 조정게인 * 1000
99	모터2의 PID 공통 조정게인 * I 조정게인 * I 위치제어 추가조정 게인 * 1000	모터2의 P 조정게인 * 1000	모터2의 I 조정게인 * 1000	모터2의 D 조정게인 * 1000
114	speed1_cmd_old / (70L * MOTOR1_NUMBER_of_POLE_PAIR)	speed1_RPM	lead_angle1_set	(MOTOR1_INERTIAL_RATIO * (long long)motor1_acc_rate) >> 12
115	speed2_cmd_old / (70L * MOTOR2_NUMBER_of_POLE_PAIR)	speed2_RPM	lead_angle2_set	(MOTOR2_INERTIAL_RATIO * (long long)motor2_acc_rate) >> 12
116	notch filter MOTOR1_uk1_0	notch filter MOTOR1_yk1_0	notch filter MOTOR2_uk1_0	notch filter MOTOR2_yk1_0
130	ADC_RESULT0_reg	ADC_RESULT1_reg	ADC_RESULT2_reg	ADC_RESULT3_reg
131	ADC_RESULT4_reg	ADC_RESULT5_reg	ADC_RESULT6_reg	ADC_RESULT7_reg
132	ADC_RESULT8_reg	ADC_RESULT9_reg	ADC_RESULT10_reg	ADC_RESULT11_reg
133	ADC_RESULT12_reg	ADC_RESULT13_reg	ADC_RESULT14_reg	ADC_RESULT15_reg
134	motor1_home_searching_status	position1 - POSITION1_MID	position1_set - POSITION1_MID	position1_at_1ms - POSITION1_MID
135	motor2_home_searching_status	position2 - POSITION2_MID	position2_set - POSITION2_MID	position2_at_1ms - POSITION2_MID
508	lead_angle1_set	lead_angle1	lead_angle1_set - lead_angle1	0
509	lead_angle2_set	lead_angle2	lead_angle2_set - lead_angle2	0

Qxddd1 에 의하여 전송되는 변수값 (RJM_VER5/6인 경우)

RJM_VER5/6인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
0x800 +32	모터1의 순간위치명령값 (position1_set) 단위크기는 1 encoder pulse임	모터1의 위치제어 오차 단위크기는 1 encoder pulse (엔 코더를 안쓰는 경우는 16384 펄 스의 엔코더로 간주함)	모터1의 위치제어 출력값 (lead angle 명령값) 4096일때 90도	모터1의 lead angle 값 4096일때 90도
	모터1의 실제 속도값 단위크기는 RPM*10	모터1의 lead angle 제어오차 4096일때 90도	모터1의 전류명령값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터1의 실제전류값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당
0x800 +33	모터2의 순간위치명령값 (position2_set) 단위크기는 1 encoder pulse임	모터2의 위치제어 오차 단위크기는 1 encoder pulse (엔 코더를 안쓰는 경우는 16384 펄 스의 엔코더로 간주함)	모터2의 위치제어 출력값 (lead angle 명령값) 4096일때 90도	모터2의 lead angle 값 4096일때 90도
	모터2의 실제 속도값 단위크기는 RPM*10	모터2의 lead angle 제어오차 4096일때 90도	모터2의 전류명령값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당	모터2의 실제전류값 2/5/10mΩ의 경우 81.9/204.75/409.5 가 1A에 해당

Qxddd1 에 의하여 전송되는 변수값 (외장형 F/T 센서의 경우)

RJM_VER8 외장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
2	strain guage bridge1의 증폭된 전압값 (ADCRESULT8_reg >> 4)	strain guage bridge2의 증폭된 전압값 (ADCRESULT9_reg >> 4)	strain guage bridge3의 증폭된 전압값 (ADCRESULT10_reg >> 4)	strain guage bridge4의 증폭된 전압값 (ADCRESULT11_reg >> 4)
3	strain guage bridge1의 증폭된 전압값을 필터링한 값 (vfilt[8])	strain guage bridge2의 증폭된 전압값을 필터링한 값 (vfilt[9])	strain guage bridge3의 증폭된 전압값을 필터링한 값 (vfilt[10])	strain guage bridge4의 증폭된 전압값을 필터링한 값 (vfilt[11])
4	strain guage bridge1의 증폭된 전압값을 필터링하고 offset(온도 offset 제외)을 제거한 값 (vfilt[8] - force_data_offset[0])	strain guage bridge2의 증폭된 전압값을 필터링하고 offset(온도 offset 제외)을 제거한 값 (vfilt[9] - force_data_offset[1])	strain guage bridge3의 증폭된 전압값을 필터링하고 offset(온도 offset 제외)을 제거한 값 (vfilt[10] - force_data_offset[2])	strain guage bridge4의 증폭된 전압값을 필터링하고 offset(온도 offset 제외)을 제거한 값 (vfilt[11] - force_data_offset[3])
5	온도보상 안된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상 안된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상 안된 compression force 단위크기는 0.1N	온도센서 온도값 단위크기는 0.01도
6	strain guage bridge1의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[8] - force_data_offset[0] - temp_offset[0])	strain guage bridge2의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[9] - force_data_offset[1] - temp_offset[1])	strain guage bridge3의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[10] - force_data_offset[2] - temp_offset[2])	strain guage bridge4의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[11] - force_data_offset[3] - temp_offset[3])
7	온도보상된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상된 compression force 단위크기는 0.1N	온도센서 온도값 단위크기는 0.01도
8	온도보상된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도센서 저항값 단위크기는 0hm	온도보상된 compression force 단위크기는 0.1N	온도센서 온도값 단위크기는 0.01도

6/7/8 번항은 온도센서가 장착된 H/W 버전에서만 해당됨.

Qxddd1 에 의하여 전송되는 변수값 (외장형 F/T 센서의 경우)

RJM_VER8 외장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
10	온도보상 안된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상 안된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상 안된 compression force 단위크기는 0.1N	온도보상 안된 독립 1채널의 load_cell 값 단위크기는 0.1N
11	온도보상된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상된 compression force 단위크기는 0.1N	온도보상된 독립 1채널의 load_cell 값 단위크기는 0.1N
14	공급전압 65 가 1V 에 해당함.			
15	공급전압 65 가 1V 에 해당함.			
20	operation_mode	mainLoopStep_cnt 0-9 범위의 값으로서 단위크기는 1ms	mainLoop_execution_time 단위크기는 1ms의 1%	maximum_interrupt_service_time 단위크기는 10us의 1%
43	fault_status	fault1_status	IRMCK201_fault_status	MOT_EN_N_port * 1000 + operation_mode
44	faultBufPointer	faultBuf[faultBufPointer]	faultBuf[faultBufPointer-1]	faultBuf[faultBufPointer-2]
45	fault_save_timer	faultBufPointer_save	faultBuf_save[faultBufPointer_save]	faultBuf_save[faultBufPointer_save-1]
46	fault_status	fault_save_timer	speed1_set	operation_mode
130	ADC_RESULT0_reg	ADC_RESULT1_reg	ADC_RESULT2_reg	ADC_RESULT3_reg
131	ADC_RESULT4_reg	ADC_RESULT5_reg	ADC_RESULT6_reg	ADC_RESULT7_reg
132	ADC_RESULT8_reg	ADC_RESULT9_reg	ADC_RESULT10_reg	ADC_RESULT11_reg
133	ADC_RESULT12_reg	ADC_RESULT13_reg	ADC_RESULT14_reg	ADC_RESULT15_reg

Qxddd1 에 의하여 전송되는 변수값 (내장형 F/T 센서의 경우)

RJM_VER7 내장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
2	strain guage bridge1의 증폭된 전압값 (ADCRESULT0_reg >> 4)	strain guage bridge2의 증폭된 전압값 (ADCRESULT1_reg >> 4)	strain guage bridge3의 증폭된 전압값 (ADCRESULT2_reg >> 4)	strain guage bridge4의 증폭된 전압값 (ADCRESULT3_reg >> 4)
3	strain guage bridge1의 증폭된 전압값을 필터링한 값 (vfilt[8])	strain guage bridge2의 증폭된 전압값을 필터링한 값 (vfilt[9])	strain guage bridge3의 증폭된 전압값을 필터링한 값 (vfilt[10])	strain guage bridge4의 증폭된 전압값을 필터링한 값 (vfilt[11])
4	strain guage bridge1의 증폭된 전압값을 필터링하고 offset을 제거한 값 (vfilt[8] - force_data_offset[0])	strain guage bridge2의 증폭된 전압값을 필터링하고 offset을 제거한 값 (vfilt[9] - force_data_offset[1])	strain guage bridge3의 증폭된 전압값을 필터링하고 offset을 제거한 값 (vfilt[10] - force_data_offset[2])	strain guage bridge4의 증폭된 전압값을 필터링하고 offset을 제거한 값 (vfilt[11] - force_data_offset[3])
5	온도보상 안된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상 안된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상 안된 compression force 단위크기는 0.1N	온도센서 온도값 단위크기는 0.01도
6	strain guage bridge1의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[8] - force_data_offset[0] - temp_offset[0])	strain guage bridge2의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[9] - force_data_offset[1] - temp_offset[1])	strain guage bridge3의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[10] - force_data_offset[2] - temp_offset[2])	strain guage bridge4의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[11] - force_data_offset[3] - temp_offset[3])
7	온도보상된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상된 compression force 단위크기는 0.1N	온도센서 온도값 단위크기는 0.01도
8	온도보상된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도센서 저항값 단위크기는 0hm	온도보상된 compression force 단위크기는 0.1N	온도센서 온도값 단위크기는 0.01도

Qxddd1 에 의하여 전송되는 변수값 (내장형 F/T 센서의 경우)

RJM_VER7 내장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
10	온도보상 안된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상 안된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상 안된 compression force 단위크기는 0.1N	온도보상 안된 load_cell
11	온도보상된 X 방향 모멘트값 (X_moment) 단위크기는 0.01Nm	온도보상된 Y 방향 모멘트값 (Y_moment) 단위크기는 0.01Nm	온도보상된 compression force 단위크기는 0.1N	온도보상된 load_cell
12	6축인 경우는 strain guage bridge5의 증폭된 전압값 (ADCRESULT4_reg >> 4)	strain guage bridge6의 증폭된 전압값 (ADCRESULT5_reg >> 4)	DA output5 4095 가 3V 에 해당함.	DA output6 4095 가 3V 에 해당함.
13	6축인 경우는 strain guage bridge5의 증폭된 전압값을 필터링한 값 (vfilt[12])	strain guage bridge6의 증폭된 전압값을 필터링한 값 (vfilt[13])	strain guage bridge5의 증폭된 전압값을 필터링하고 offset을 제거한 값 (vfilt[12] - force_data_offset[4])	strain guage bridge6의 증폭된 전압값을 필터링하고 offset을 제거한 값 (vfilt[13] - force_data_offset[5])
14	공급전압 65 가 1V 에 해당함.			
15	6축이 아닌 경우는 공급전압 65 가 1V 에 해당함.			
15	6축인 경우는 온도보상 안된 X 방향 force값 (X_force) 단위크기는 0.1N	온도보상 안된 Y 방향 force값 (Y_force) 단위크기는 0.1N	온도보상 안된 Z 방향 모멘트값 (Z_moment) 단위크기는 0.01Nm	온도센서 온도값 단위크기는 0.01도

Qxddd1 에 의하여 전송되는 변수값 (내장형 F/T 센서의 경우)

RJM_VER7 내장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
16	6축인 경우는 strain guage bridge5의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[12] - force_data_offset[4] - temp_offset[4])	strain guage bridge6의 증폭된 전압값을 필터링하고 offset(온도 offset 포함)을 제거한 값 (vfilt[13] - force_data_offset[5] - temp_offset[5])	온도센서 온도값 단위크기는 0.01도	온도센서 온도값 증가량 단위크기는 0.01도
17	6축인 경우는 온도보상된 X 방향 force값 (X_force) 단위크기는 0.1N	온도보상된 Y 방향 force값 (Y_force) 단위크기는 0.1N	온도보상된 Z 방향 모멘트값 (Z_moment) 단위크기는 0.01Nm	온도센서 온도값 단위크기는 0.01도
18	temperature_FT_sensor_thermocouple[0] - filtered hot junction 온도 단위크기는 0.01℃	temperature_FT_sensor_thermocouple[1] - filtered hot junction 온도 단위크기는 0.01℃	temperature_FT_sensor_thermocouple[2] - filtered hot junction 온도 단위크기는 0.01℃	temperature_FT_sensor_thermocouple[3] - filtered hot junction 온도 단위크기는 0.01℃
19	temperature_FT_sensor_thermocouple[4] - unfiltered cold junction 온도 단위크기는 0.01℃	temperature_FT_sensor_thermocouple[5] - unfiltered cold junction 온도 단위크기는 0.01℃	temperature_FT_sensor_thermocouple[6] - unfiltered cold junction 온도 단위크기는 0.01℃	temperature_FT_sensor_thermocouple[7] - unfiltered cold junction 온도 단위크기는 0.01℃
20	operation_mode	mainLoopStep_cnt 0-9 범위의 값으로서 단위크기는 1ms	mainLoop_execution_time 단위크기는 1ms의 1%	maximum_interrupt_service_time 단위크기는 10us의 1%

Qxddd1 에 의하여 전송되는 변수값 (내장형 F/T 센서의 경우)

RJM_VER7 내장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
43	fault_status	fault1_status	IRMCK201_fault_status	MOT_EN_N_port * 1000 + operation_mode
44	faultBufPointer	faultBuf[faultBufPointer]	faultBuf[faultBufPointer-1]	faultBuf[faultBufPointer-2]
45	fault_save_timer	faultBufPointer_save	faultBuf_save[faultBufPointer_save]	faultBuf_save[faultBufPointer_save-1]
46	fault_status	fault_save_timer	speed1_set	operation_mode
130	ADC_RESULT0_reg	ADC_RESULT1_reg	ADC_RESULT2_reg	ADC_RESULT3_reg
131	ADC_RESULT4_reg	ADC_RESULT5_reg	ADC_RESULT6_reg	ADC_RESULT7_reg
132	ADC_RESULT8_reg	ADC_RESULT9_reg	ADC_RESULT10_reg	ADC_RESULT11_reg
133	ADC_RESULT12_reg	ADC_RESULT13_reg	ADC_RESULT14_reg	ADC_RESULT15_reg

Qxddd1 에 의하여 전송되는 변수값 (내장형 F/T 센서의 경우)

RJM_VER7 내장형 F/T 센서인 경우								
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수	다섯번째 변수	여섯번째 변수	일곱번째 변수	여덟번째 변수
6144	ADCRESULT0_reg >> 4	ADCRESULT1_reg >> 4	ADCRESULT2_reg >> 4	ADCRESULT3_reg >> 4	ADCRESULT4_reg >> 4	ADCRESULT5_reg >> 4	ADCRESULT7_reg >> 4	ADCRESULT15_reg >> 4
6146	ADCRESULT0_reg >> 4	ADCRESULT1_reg >> 4	ADCRESULT2_reg >> 4	ADCRESULT3_reg >> 4	ADCRESULT4_reg >> 4	ADCRESULT5_reg >> 4	ADCRESULT7_reg >> 4	ADCRESULT15_reg >> 4
6147	vfilt[8]	vfilt[9]	vfilt[10]	vfilt[11]	vfilt[12]	vfilt[13]	temperature_FT_sensor	NTC_resistance
6148	vfilt[8]-force_data_offset[0]	vfilt[9]-force_data_offset[1]	vfilt[10]-force_data_offset[2]	vfilt[11]-force_data_offset[3]	vfilt[12]-force_data_offset[4]	vfilt[13]-force_data_offset[5]	temperature_FT_sensor	NTC_resistance
6149	온도보상 안된 X_moment	온도보상 안된 Y_moment	온도보상 안된 foot_force	온도보상 안된 X_force	온도보상 안된 Y_force	온도보상 안된 Z_moment	temperature_FT_sensor	NTC_resistance
6150	웁셋, 온도 보상된 vfilt[8]	웁셋, 온도 보상된 vfilt[9]	웁셋, 온도 보상된 vfilt[10]	웁셋, 온도 보상된 vfilt[11]	웁셋, 온도 보상된 vfilt[12]	웁셋, 온도 보상된 vfilt[13]	temperature_FT_sensor	NTC_resistance
6151	온도보상된 X_moment	온도보상된 Y_moment	온도보상된 foot_force	온도보상된 X_force	온도보상된 Y_force	온도보상된 Z_moment	temperature_FT_sensor	NTC_resistance
6152	DA output0	DA output1	DA output2	DA output3	DA output4	DA output5	temperature_FT_sensor	NTC_resistance
6153	온도보상되고 추가 필터링된 X_moment	온도보상되고 추가 필터링된 Y_moment	온도보상되고 추가 필터링된 foot_force	온도보상되고 추가 필터링된 X_force	온도보상되고 추가 필터링된 Y_force	온도보상되고 추가 필터링된 Z_moment	temperature_FT_sensor	NTC_resistance

Ts/Tt 명령에 의한 step파 test와 Tu/Tv 명령에 의한 sin파 test시의 Qxddd1 사용

Ts/Tt/Tu/Tv 명령에 의한 test시의 모니터링				
ddd1 값	위치제어모드 (01/02/03/04/0E/0F 모드)	속도제어모드 (05/06/0A 모드)	전류제어모드 (07 모드)	comment
0	위치값	-	-	TMS320F28334/5에서 03번모드의 속도제어 튜닝은 0A번 모드에서 실시할것
1	속도값	-	-	
75/76	speedx_RPM_10/speedx_RPM_per iod_10/speedx_RPM_count_10/speedx_RPM_combined_10		-	TMS320F28334/5에서 카운트방식 /펄스주기방식/조합방식 속도값
2/3	상전류값			DC의 경우는 Qx2;에서 모터1/2의 상전류가 모두 표시됨
4	총전류값			
93/97	위치제어 상황			
92/96	속도제어 상황			
91/95	전류제어상황			
55/56	$\text{currentx_cmd}/\text{Iqx_AD}/\text{motorx_moving_flg} * 1000 + \text{Idrefx} + (\text{MOTORx_PA_end_processing_time_set} - \text{MOTORx_PA_end_timer})/\text{Idx_AD}$			BLDC의 경우 전류값
80/81	$\text{motorx_error}/\text{speedx_long_cmd} + \text{speedx_long_cmd_feedforward}/\text{motorx_error_speed}/\text{currentx_cmd}$	-	-	TMS320F28334/5에서 03번 위치 제어모드 일때 위치/속도제어루프의 오차 및 제어출력값
82/83	$\text{motorx_error_speed}/\text{motorx_integral_speed}/\text{motorx_derivative_speed}/\text{currentx_cmd}$		-	TMS320F28334/5에서 03번 위치 /0A번 속도제어모드 일때 속도 제어루프의 오차 및 제어출력값
86/87	$\text{speedx_long_cmd_feedforward}/\text{speedx_long_cmd} + \text{speedx_long_cmd_feedforward}/\text{motorx_error_speed}/\text{currentx_cmd}$		-	TMS320F28334/5에서 03번 위치 /0A번 속도제어모드 일때
114/115	속도명령/속도/전류명령/전류offset		-	가속에 따른 전류 offset
77	speedx_RPM_10/gain_factorx		-	속도에 따른 게인 조정

Ts/Tt 명령에 의한 step파 test와 Tu/Tv 명령에 의한 sin파 test시의 Qxddd1 사용

Ts/Tt/Tu/Tv 명령에 의한 test시의 모니터링				
ddd1 값	위치제어모드 (01/02/03/04/0E/0F 모드)	속도제어모드 (05/06/0A 모드)	전류제어모드 (07 모드)	comment
51/52	총전류/상전류/전류리밋1/오류코드			과전류보호
53/54	상전류/과전류타이머/과전류flag/오류코드			과전류보호
68/69	총전류/상전류/전류리밋high/전류리밋low			과전류보호

명령어	구분	설명
;QZHLHL,HLHL,HLHL,HLHL; ;QZ; (+#^%)	operation	Qz 명령어로 1ms 마다 값을 host로 전송하는 4개의 변수, 또는 Qy 명령어로 host로 전송하는 8개 변수중에서 앞쪽 4개의 변수에 대한 주소를 설정함. 1st word : 첫번째 변수의 주소 2nd word : 두번째 변수의 주소 3rd word : 세번째 변수의 주소 4th word : 네번째 변수의 주소
	return value	QZHLHL,HLHL,HLHL,HLHL;
	comment	QRHLHL,HL; 명령어로 설정가능한 변수의 주소범위를 벗어나면 reset됨. 설정가능한 주소범위는 다음과 같음. 1st/2nd/3rd/4th : RAM address (TMS320F810/11/12) [0000 - 017F] : M0 RAM block (.stack) [0180 - 01BF] : M0 RAM block (.bss) [01C0 - 06FF] : M0/1 RAM block (.ebss) [0700 - 07FF] : M1 RAM block (asm에서 사용) [0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용) [E000 - FFFF] : H0 RAM block (.esysmem) [E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨. 1st/2nd/3rd/4th : RAM address (TMS320F28334/28335) [0000 - 004F] : M0 RAM block (stack for boot) [0050 - 03FF] : M0 RAM block (no use) [0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss) [0880 - 793F] : peripheral memory [8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용) [C000 - DFFF] : L4/L5 RAM block (.ebss) [E000 - FFFF] : L6/L7 RAM block (.esysmem) 사용 가능한 주소영역과 각 주소의 변수값의미는 S/W 버전마다 다를수 있음.

명령어	구분	설명
;Qz; (+#^%)	operation	QZ 명령어로 주소가 지정된 4word를 1msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
	return value	none.
	comment	전송되는 데이터는 Qz; 명령을 전송해온 RS232C/RS485/CAN 채널로만 전송됨. 모터제어기의 RS232C 채널이 115200 baud rate의 전송속도로 설정이 되어 있을 때만 유효함. 전송데이터의 주소는 QZ 명령어에 의하여 설정됨. Qz; 명령어를 한번 보내면 전송개시, 한번 더 보내면 전송을 정지함. 1ms당 4word를 8byte로 나누어서 보냄. 2012년 12월 27일 S/W 버전부터는 Qxyz; 명령어로도 전송을 정지할수 있음.
;Qzx; (+#^)	operation	QZ 명령어로 주소가 지정된 4word를 2msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
	return value	none.
	comment	전송되는 데이터는 Qzx; 명령을 전송해온 RS232C/RS485/CAN 채널로만 전송됨. 모터제어기의 RS232C 채널이 115200 baud rate의 전송속도로 설정이 되어 있을 때만 유효함. 전송데이터의 주소는 QZ 명령어에 의하여 설정됨. Qz; 또는 Qzx; 명령어를 한번 보내면 전송개시, 한번 더 보내면 전송을 정지함. 2012년 1월 08일 S/W 버전부터 적용되며 2ms당 4word를 13byte로 나누어서 보냄. 2012년 12월 27일 S/W 버전부터는 Qxyz; 명령어로도 전송을 정지할수 있음.
;Qzy; (+#^)	operation	QZ/QY 명령어로 주소가 지정된 8word를 2msec 마다 host PC로 전송을 시작하게 하거나 전송을 중지 시킴.
	return value	none.
	comment	전송되는 데이터는 Qzy; 명령을 전송해온 RS232C/RS485/CAN 채널로만 전송됨. 모터제어기의 RS232C 채널이 115200 baud rate의 전송속도로 설정이 되어 있을 때만 유효함. 전송데이터의 주소는 QZ 명령어와 QY 명령어에 의하여 설정됨. Qz; 또는 Qzy; 명령어를 한번 보내면 전송개시, 한번 더 보내면 전송을 정지함. 2012년 1월 08일 S/W 버전부터 적용되며 2ms당 8word를 16byte로 나누어서 보냄. 2012년 12월 27일 S/W 버전부터는 Qxyz; 명령어로도 전송을 정지할수 있음.

명령어	구분	설명
;QYHLHL,HLHL,HLHL,HLHL; ;QY; (+#^)	operation	Qy 명령어로 host로 전송하는 8개 변수중에서 뒷쪽 4개의 변수에 대한 주소를 설정함. 1st word : 다섯번째 변수의 주소 2nd word : 여섯번째 변수의 주소 3rd word : 일곱번째 변수의 주소 4th word : 여덟번째 변수의 주소
	return value	QYHLHL,HLHL,HLHL,HLHL;
	comment	QRHLHL,HL; 명령어로 설정가능한 변수의 주소범위를 벗어나면 reset됨. 설정가능한 주소범위는 다음과 같음. 1st/2nd/3rd/4th : RAM address (TMS320F810/11/12) [0000 - 017F] : M0 RAM block (.stack) [0180 - 01BF] : M0 RAM block (.bss) [01C0 - 06FF] : M0/1 RAM block (.ebss) [0700 - 07FF] : M1 RAM block (asm에서 사용) [0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용) [E000 - FFFF] : H0 RAM block (.esystem) [E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨. 1st/2nd/3rd/4th : RAM address (TMS320F28334/28335) [0000 - 004F] : M0 RAM block (stack for boot) [0050 - 03FF] : M0 RAM block (no use) [0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss) [0880 - 793F] : peripheral memory [8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용) [C000 - DFFF] : L4/L5 RAM block (.ebss) [E000 - FFFF] : L6/L7 RAM block (.esystem) 사용 가능한 주소영역과 각 주소의 변수값의미는 S/W 버전마다 다를수 있음.

명령어	구분	설명
;Qy; (+#^)	operation	QZ,QY 명령어로 주소가 지정된 8word를 binary 값의 형태로 host PC에 전송함.
	return value	RS232/485의 경우 : QyHBLBHBLBHBLBHBLBHBLBHBLBHBLBHBLB; CAN의 경우 : HBLBHBLBHBLBHBLBHBLBHBLBHBLBHBLB HBLB 는 binary 형태의 16bit word의 상위바이트 한개, 하위바이트 한개를 가리킴. 1st 2-byte : QZ로 주소가 지정된 첫번째 변수의 값 2nd 2-byte : QZ로 주소가 지정된 두번째 변수의 값 3rd 2-byte : QZ로 주소가 지정된 세번째 변수의 값 4th 2-byte : QZ로 주소가 지정된 네번째 변수의 값 5th 2-byte : QY로 주소가 지정된 첫번째 변수의 값 6th 2-byte : QY로 주소가 지정된 두번째 변수의 값 7th 2-byte : QY로 주소가 지정된 세번째 변수의 값 8th 2-byte : QY로 주소가 지정된 네번째 변수의 값
	comment	HBLB 는 16bit word의 상위,하위바이트를 가리키며 16bit의 binary 값임.

명령어	구분	설명
;Q1; (+#^%)	operation	operation_status를 읽어서 host로 전송함.
	return value	Q1 HLHL; bit0 : motor1_start_to_move_trajectory_flg bit1 : motor1_start_to_move_to_position_flg bit2 : motor1_moving_flg bit3 : motor1_feedback_mode bit4 : motor1_home_searching_status bit0 bit5 : motor1_home_searching_status bit1 bit6 : motor1_home_searching_status bit2 bit7 : motor1_home_searching_status bit3 bit8 : motor2_start_to_move_trajectory_flg bit9 : motor2_start_to_move_to_position_flg bit10 : motor2_moving_flg bit11 : motor2_feedback_mode bit12 : motor2_home_searching_status bit0 bit13 : motor2_home_searching_status bit1 bit14 : motor2_home_searching_status bit2 bit15 : motor2_home_searching_status bit3
	comment	motorx_start_to_move_trajectory_flg=1 이면 PM 명령이 수행중인 상태임. motorx_start_to_move_to_position_flg=1 이면 PP/PQ/Pp/Pq 명령이 수행중 또는 대기중 상태임. motor1_home_searching_status 는 홈동작 진행/성공여부임. (0이면 홈동작명령 실시된 바 없음, 1이면 성공, 2이면 실패, 3이면 리밋 찾는 중, 4이면 홈을 찾는 중, 5이면 홈을 지나치는 중, 6이면 offset 이동 중, 7이면 리밋센서가 꺼지기를 기다리는 중 (2개의 리밋센서만 사용할 때), 8이면 홈을 찾는중 (한개의 홈센서만 사용할 때), 9이면 홈센서가 꺼지기를 기다리는 중 (한개의 홈센서만 사용할 때) 2014년 9월 23일 S/W 버전 부터는 10이면 PsA5A5! 등의 명령을 내리는 순간, 11이면 PsA5A5! 등의 명령을 수행하는 중, 12~111이면 100ms를 기다리는 중, 112이면 위치값을 초기화하는 순간, 113이면 offset 이동을 명령하는 순간을 의미하는데, 그 값이 16이상일 때는 15로 표기함.

명령어	구분	설명
	comment	motor1_moving_flg, motor2_moving_flg는 작동모드 1/2/3/4/8에서만 유효함. 2012년 5월 22일 S/W 버전 부터는 작동모드 5/6에서도 유효하며, 모터의 실제움직임을 반영한 값이 아니고, 모터의 속도가 0이 되도록 제어하고 있는 상태임을 나타냄. 2013년 1월 10일부터 2013년 4월 26일 이전 S/W 버전에서는 리턴값에서 Q1 다음의 space가 생략되었음.

Q1의 return 값 (내부 변수명은 operation_status)								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	motor2_home_se arching_status bit2	motor2_home_se arching_status bit2	motor2_home_se arching_status bit1	motor2_home_se arching_status bit0	motor2_feedbac k_mode	motor2_moving_ flg	motor2_start_t o_move_to_posi tion_flg	motor2_start_t o_move_traject ory_flg
bit7 - bit0	motor1_home_se arching_status bit2	motor1_home_se arching_status bit2	motor1_home_se arching_status bit1	motor1_home_se arching_status bit0	motor1_feedbac k_mode	motor1_moving_ flg	motor1_start_t o_move_to_posi tion_flg	motor1_start_t o_move_traject ory_flg

명령어	구분	설명
;Q2; (+#^%)	operation	fault_status, fault1_status를 읽어서 host로 전송함.
	return value	<p>Q2 HLHL,HLHL;</p> <p>1st word bit0 : 모터1의 gate kill fault (DC/BLDC) 1st word bit1 : 모터2의 gate kill fault (DC/BLDC) 1st word bit2 : 공급전압이 허용 low voltage 이하로 낮게 공급된 fault (DC/BLDC) 1st word bit3 : 공급전압이 허용 maximum voltage를 초과한 fault (DC/BLDC) 1st word bit4 : 모터1의 maximum current를 초과한 fault (DC/BLDC) 1st word bit5 : 모터2의 maximum current를 초과한 fault (DC/BLDC) 1st word bit6 : 모터1의 제어 오차가 허용치를 넘는 fault (DC/BLDC) 1st word bit7 : 모터2의 제어 오차가 허용치를 넘는 fault (DC/BLDC) 1st word bit8 : current1_over_flg (DC/BLDC) 1st word bit9 : current1_over_K_flg (DC/BLDC) 1st word bit10 : motor1_stall_flg (DC/BLDC) (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 1st word bit11 : motor1_stall_K_flg (DC/BLDC) (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 1st word bit12 : current2_over_flg (DC/BLDC) 1st word bit13 : current2_over_K_flg (DC/BLDC) 1st word bit14 : motor2_stall_flg (DC/BLDC) (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 1st word bit15 : motor2_stall_K_flg (DC/BLDC) (20143년 10월 07일 이전의 S/W 버전은 comment 참조)</p> <p>2nd word bit0 : motor1_position_plus_limit_over (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 2nd word bit1 : motor2_position_plus_limit_over (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 2nd word bit2 : motor1_position_minus_limit_over (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 2nd word bit3 : motor2_position_minus_limit_over (20143년 10월 07일 이전의 S/W 버전은 comment 참조) 2nd word bit4 : IRMCK201 fault 2nd word bit5 : RS232C communication fault 2nd word bit6 : RS485 communication fault 2nd word bit7 : CAN communication fault</p>

명령어	구분	설명
	return value	<p>2nd word bit8 : 모터1의 권선온도가 ZI 명령어로 설정된 허용최대온도를 초과함.</p> <p>2nd word bit9 : 모터2의 권선온도가 ZI 명령어로 설정된 허용최대온도를 초과함.</p> <p>2nd word bit10 : 모터공급전압이 Z6로 설정된 전압의 125%를 초과함.</p> <p>2nd word bit11 : 모터제어기의 bridge1에서 발생하는 발열량이 방열판에서 감당 가능한 범위를 초과함.</p> <p>2nd word bit12 : 모터제어기의 bridge2에서 발생하는 발열량이 방열판에서 감당 가능한 범위를 초과함.</p> <p>2nd word bit13 : 12V용 EPS 모터제어기에 있어서 모터1의 과부하로 인하여 모터구동 전압이 10V 이하로 떨어짐.</p> <p>2nd word bit14 : 과전류/과온도 검출기능이 있는 모터구동 IC를 사용하는 경우에 한하며, 모터1에서 과전류/과온도 fault가 발생함.</p> <p>2nd word bit15 : 과전류/과온도 검출기능이 있는 모터구동 IC를 사용하는 경우에 한하며, 모터2에서 과전류/과온도 fault가 발생함.</p>
	comment	<p>각각의 fault에 대한 처리는 별도의 document에 기술되었음.</p> <p>1st word bit0/1/2/3/4/5/6/7 은 fault 플래그를 지우려면 PRHLHL; 명령어를 보내면 됨.</p> <p>1st word bit8/9/12/13은 fault 상태가 제거되고 SI 명령어로 설정되는 시간을 경과하면 자동으로 지워짐.</p> <p>1st word bit10/11/14/15는 fault 플래그를 지우려면 PPR; 명령어를 보내면 됨. 20143년 10월 07일 이전의 S/W 버전에서는 BLDC에서 엔코더를 안쓰는 경우 stall 조건에서 정지하는 명령위치 설정 오류가 있음.</p> <p>2nd word bit0/1/2/3 은 fault 상태가 제거되면 바로 지워짐. 20143년 10월 07일 이전의 S/W 버전에서는 BLDC에서 엔코더를 안쓰는 경우 2nd word bit0/1/2/3 의 값에 오류가 있음.</p> <p>2nd word bit2/3의 경우 2013년 11월 24일 이전 S/W 버전에서는 위치값이 0보다 작아서 음수가 된 경우 값이 1로 설정되는 경우가 있음.</p> <p>2nd word bit4 는 IRMCK201에 fault가 발생하였거나 또는 추가의 fault가 있음을 알려주는 비트이며 Q3; 명령어를 사용하여 fault 상태를 알 수 있음. 이 IRMCK201을 사용한 경우에는 fault 플래그가 한번 발생하면 강제로 지우지 않는 한 남아있으며, fault 플래그를 없애고 fault를 복구하려면 PRHLHL; 명령어를 보내면 됨.</p> <p>2nd word bit5/6/7 는 RS232, RS485, CAN 통신과정에서 fault가 발생하였음을 알려주는 비트이며, fault의 상태를 알려면 EC?; 명령어를 보내면 됨. 이 fault 플래그는 한번 발생하면 강제로 지우지 않는 한 남아있으며, fault 플래그를 없애고 fault를 복구하려면 ECR; 명령어를 보내면 됨.</p>

명령어	구분	설명																		
		<p>2nd word bit8/9 는 권선의 온도가 허용온도를 초과한 에러로서, 전류제어를 사용하는 상태일때는 전류값의 최대치를 줄이는 action을 자동으로 실시하게되며, 온도가 허용온도 이하로 내려가면 지연시간후 에러에서 복구됨.</p> <p>2nd word bit11/12 는 모터제어기의 발열량이 Z9 명령어로 설정하는 방열판용량으로부터 계산된 발열 허용범위를 초과한 에러로서, 전류제어를 사용하는 상태일때는 전류값의 최대치를 줄이는 action을 자동으로 실시하게되며, 발열량이 허용량 이하로 내려가면 시간지연후 에러에서 복구됨.</p> <p>2nd word bit13 은 모터의 과부하로 인하여 모터구동 전압이 10V 이하로 떨어질 때 발생함. 전류제어에 사용하는 전류크기의 제한값 (Sw 명령어로 설정됨)을 점차 작게하며, 모터구동전압이 10V 이상이되면 회복 시킴.</p> <p>2nd word bit14/15 은 과전류/과온도 검출기능이 있는 모터구동 IC를 사용하는 경우에 한하며, 과전류 fault 조건에서 발생하여 모터구동출력을 off 시킴.</p>																		
	comment	<table border="1"> <thead> <tr> <th></th> <th>bit14가 1이면</th> <th>bit15가 1이면</th> </tr> </thead> <tbody> <tr> <td>50W 모터제어기 모델 (L6206, L6207, L6226, L6227을 사용한 경우)</td> <td>과전류 또는 과온도가 검출되어 출력이 차단되었음.</td> <td>과전류 또는 과온도가 검출되어 출력이 차단되었음.</td> </tr> <tr> <td>100W 모터제어기 모델 (DRV8432/DRV8332를 사용한 경우)</td> <td>과전류 또는 과온도가 검출되어 출력이 차단되었음.</td> <td>과온도 경고상태가 검출되었음.</td> </tr> <tr> <td>IRS2336 IGBT/MOSFET 구동IC를 사용하는 경우</td> <td>IR21771 전류검출 IC를 사용하는 경우는 과전류가 검출되어 출력이 차단되었음.</td> <td>IRS2336에 의한 과전류가 검출되어 출력이 차단되었음.</td> </tr> <tr> <td>IRAM136-3063B IGBT를 사용하는 모델</td> <td></td> <td>과온도 경고상태가 검출되었음.</td> </tr> <tr> <td>그외에 IR21771 전류검출 IC를 사용하거나 과전류검출회로를 내장한 모터제어기의 경우</td> <td>과전류가 검출되어 출력이 차단되었음.</td> <td>-</td> </tr> </tbody> </table> <p>2013년 1월 10일부터 2013년 4월 26일 이전 S/W 버전에서는 리턴값에서 Q1 다음의 space가 생략되었음.</p>		bit14가 1이면	bit15가 1이면	50W 모터제어기 모델 (L6206, L6207, L6226, L6227을 사용한 경우)	과전류 또는 과온도가 검출되어 출력이 차단되었음.	과전류 또는 과온도가 검출되어 출력이 차단되었음.	100W 모터제어기 모델 (DRV8432/DRV8332를 사용한 경우)	과전류 또는 과온도가 검출되어 출력이 차단되었음.	과온도 경고상태가 검출되었음.	IRS2336 IGBT/MOSFET 구동IC를 사용하는 경우	IR21771 전류검출 IC를 사용하는 경우는 과전류가 검출되어 출력이 차단되었음.	IRS2336에 의한 과전류가 검출되어 출력이 차단되었음.	IRAM136-3063B IGBT를 사용하는 모델		과온도 경고상태가 검출되었음.	그외에 IR21771 전류검출 IC를 사용하거나 과전류검출회로를 내장한 모터제어기의 경우	과전류가 검출되어 출력이 차단되었음.	-
	bit14가 1이면	bit15가 1이면																		
50W 모터제어기 모델 (L6206, L6207, L6226, L6227을 사용한 경우)	과전류 또는 과온도가 검출되어 출력이 차단되었음.	과전류 또는 과온도가 검출되어 출력이 차단되었음.																		
100W 모터제어기 모델 (DRV8432/DRV8332를 사용한 경우)	과전류 또는 과온도가 검출되어 출력이 차단되었음.	과온도 경고상태가 검출되었음.																		
IRS2336 IGBT/MOSFET 구동IC를 사용하는 경우	IR21771 전류검출 IC를 사용하는 경우는 과전류가 검출되어 출력이 차단되었음.	IRS2336에 의한 과전류가 검출되어 출력이 차단되었음.																		
IRAM136-3063B IGBT를 사용하는 모델		과온도 경고상태가 검출되었음.																		
그외에 IR21771 전류검출 IC를 사용하거나 과전류검출회로를 내장한 모터제어기의 경우	과전류가 검출되어 출력이 차단되었음.	-																		

Q2 명령에 의한 fault code의 요약

Q2의 return 값중 첫번째 워드 (내부 변수명은 fault_status)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	motor2_stall_K_flg	motor2_stall_flg	current2_over_K_flg	current2_over_flg	motor1_stall_K_flg	motor1_stall_flg	current1_over_K_flg	current1_over_flg
		flash code 35		flash code 34		flash code 25		flash code 24
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	motor2_excessive_error_flg	motor1_excessive_error_flg	MOTOR2_MAX_CURRENT_fault_flg	MOTOR1_MAX_CURRENT_fault_flg	MAX_VOLTAGE_fault_flg	LOW_VOLTAGE_fault_flg	MOTOR2_gate_kill_fault_flg	MOTOR1_gate_kill_fault_flg
	flash code 33	flash code 23	flash code 32	flash code 22	flash code 12	flash code 11	fault code 31	fault code 21

Q2의 return 값중 두번째 워드 (내부 변수명은 fault1_status)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	모터제어기의 모터구동 IC가 스스로 판단한 과온도 에러 (2012년 3월 29일 버전부터 적용)	모터제어기의 모터구동 IC가 스스로 판단한 과전류 에러 (2012년 3월 29일 버전부터 적용)	모터1의 과부하로 인한 전압강하 에러 (2811 EPS제어기에만 해당됨)	모터제어기의 모터구동 bridge2 과발열 에러 (2010년 9월 22일 버전부터 적용)	모터제어기의 모터구동 bridge1 과발열 에러 (2010년 9월 22일 버전부터 적용)	공급전압이 MOSFET BREAKDOWN VOLTAGE에 접근한 에러	모터2의 권선 과온도 에러	모터1의 권선 과온도 에러
	flash code 29	flash code 28	flash code 27	flash code 19	flash code 18	flash code 17	flash code 36	flash code 26
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	CAN 통신 error	RS485 통신 error	RS232C 통신 error	IRMCK201 error 또는 추가의 fault	motor2_position_minus_limit_over	motor1_position_minus_limit_over	motor2_position_plus_limit_over	motor1_position_plus_limit_over
				flash code 13				

명령어	구분	설명
;Q3; (+#^%)	operation	IRMCK201의 fault_status, fault2_status를 읽어서 host로 전송함.
	return value	<p>Q3 HLHL; (2013년 2월 05일 이전 S/W 버전) Q3 HLHL,HLHL; (2013년 2월 05일 S/W 버전부터)</p> <p>1st word bit0 : 모터1의 gate kill fault (BLDC IRMCK201) 1st word bit1 : 모터1의 low voltage fault (BLDC IRMCK201) 1st word bit2 : 모터1의 over voltage fault (BLDC IRMCK201) 1st word bit3 : 모터1의 over speed fault (BLDC IRMCK201) 1st word bit4 : 모터1의 execution time fault (BLDC IRMCK201) 1st word bit5 : 모터1의 analog Hall sensor fault (RJM_VER7) 1st word bit6 : 모터1의 encoder fault (RJM_VER7) 1st word bit7 : low control voltage fault (고전압제어기의 경우) 1st word bit8 : 모터2의 gate kill fault (BLDC IRMCK201) 1st word bit9 : 모터2의 low voltage fault (BLDC IRMCK201) 1st word bit10 : 모터2의 over voltage fault (BLDC IRMCK201) 1st word bit11 : 모터2의 over speed fault (BLDC IRMCK201) 1st word bit12 : 모터2의 execution time fault (BLDC IRMCK201) 1st word bit13 : 모터2의 analog Hall sensor fault (RJM_VER7) 1st word bit14 : 모터2의 encoder fault (RJM_VER7) 1st word bit15 : current offset fault at operation mode setting</p> <p>2nd word bit0 : iSENSOR(SPI 통신을 사용하는 전류센서)를 사용하는 경우에 센서연결이 안 되어있는 fault 2nd word bit1 : RS485/CAN 통신을 하던 중에 일정시간 이상 통신이 두절된 fault (2013년 10월 23일 S/W 버전부터 적용됨) 2nd word bit2 : 모터1을 위한 엔코더 interpolator IC에서 오류가 발생한 fault 2nd word bit3 : 모터2를 위한 엔코더 interpolator IC에서 오류가 발생한 fault</p>

명령어	구분	설명
		<p>2nd word bit4 : 모터1에 대하여 SM 명령등에 의하여 처리 불가능한 작동모드를 설정하였을 때 (2014년 8월 16일 S/W 버전부터 적용됨)</p> <p>2nd word bit5 : 모터2에 대하여 SM 명령등에 의하여 처리 불가능한 작동모드를 설정하였을 때 (2014년 8월 16일 S/W 버전부터 적용됨)</p> <p>2nd word bit6 : iSENSOR(SPI 통신을 사용하는 전류센서)를 사용하는 경우에 센서연결이 안 되어있는 fault가 한번이상 발생하였을때 세트됨 (2014년 8월 16일 S/W 버전부터 적용됨)</p>
	comment	<p>각각의 fault나 에러에 대한 처리는 별도의 document에 기술되었음.</p> <p>1st WORD bit5/6/7/13/14/15 와 연관관 오류는 IRMCK201을 사용하지 않는 제어기에서도 발생할수 있는 fault임.</p> <p>RS485/CAN 통신을 하던 중에 일정시간 이상 통신이 두절된 fault는 Rc 명령에 의하여 fault를 clear 할수 있으며, SX 명령어 두번째파라미터의 bit30 이 0인 경우에는 통신 두절 fault 가 발생하였을 때 강제 정지 되며, 이때의 감속도는 So 명령으로 설정되는 감속도에 따름.</p> <p>2013년 1월 10일부터 2013년 4월 26일 이전 S/W 버전에서는 리턴값에서 Q1 다음의 space가 생략되었음.</p>

Q3 명령에 의한 fault code의 요약

Q3의 return 값중 첫번째 워드 (내부 변수명은 IRMCK201_fault_status)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	current offset fault at operation mode setting	모터2의 ENCODER fault	모터2의 analog Hall sensor fault	모터2의 execution time fault (IRMCK201을 사용한 경우에 한함)	모터2의 over speed fault (IRMCK201을 사용한 경우에 한함)	모터2의 over voltage fault (IRMCK201을 사용한 경우에 한함)	모터2의 low voltage fault (IRMCK201을 사용한 경우에 한함)	모터2의 gate kill fault (IRMCK201을 사용한 경우에 한함)
	flash code 58	flash code 57	flash code 56	flash code 55	flash code 54	flash code 53	fault code 52	fault code 51
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	low control voltage fault (고전압제어기의 경우)	모터1의 ENCODER fault	모터1의 analog Hall sensor fault	모터1의 execution time fault (IRMCK201을 사용한 경우에 한함)	모터1의 over speed fault (IRMCK201을 사용한 경우에 한함)	모터1의 over voltage fault (IRMCK201을 사용한 경우에 한함)	모터1의 low voltage fault (IRMCK201을 사용한 경우에 한함)	모터1의 gate kill fault (IRMCK201을 사용한 경우에 한함)
	flash code 48	flash code 47	flash code 46	flash code 45	flash code 44	flash code 43	fault code 42	fault code 41

Q3의 return 값중 두번째 워드 (내부 변수명은 fault2_status, 2013년 2월 5일 버전부터 적용)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8								
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0					모터2를 위한 엔코더 interpolator IC에서 오류가 발생한 fault	모터1을 위한 엔코더 interpolator IC에서 오류가 발생한 fault	RS485/CAN 통신을 하던 중에 일정시간이상 통신이 두절된 fault	iSENSOR를 사용하는 경우에 센서연결이 안되어있는 fault

명령어	구분	설명
;Q4; (+#^%)	operation	프로그램 작성 날짜, version, macro 설정을 읽어서 host로 전송함. (2008년 10월6일 개정)
	return value	Q4 HLHL,HLHL,HLHL,HLHL; 1st word high byte : Programming Year 1st word low byte : Programming Month 2nd word high byte : Programming Date 2nd word low byte : Programming Hour 3rd word : Major Macro Definition 4th word : Sub Macro Definition
	comment	1st word가 0608 이면 2006년 8월을 가리킴. 2nd word가 3004 이면 30일의 4시 작성 S/W 버전임을 가리킴. 3rd word의 highest 4bit : 1이면 : BIPEDAL_ROBOT 용 S/W : 2이면 : ROBOT_JOINT_MOTOR 용 S/W- : 3이면 : ROBOT_INERTIAL_SENSOR 용 S/W- : 4이면 : ROBOT_EYE 용 S/W- : 5이면 : SYSTEM MONITORING 용 S/W : 6이면 : HUMANOID 용 S/W : 7이면 : PTC HEATER 용 S/W : 8이면 : MAGNETIC BEARING 용 S/W : 9이면 : ABS CONTROL 용 S/W : a이면 : RACK type POWER AMP 용 S/W 3rd word의 next 4bit : 0/1/2/3/4/5/6/7/8/9 : TMS320LF2406A / TMS320F2810 / TMS320F2811 or 12 / TMS320LF2403A / C8051F411 / TMS320LF2407A / TMS320F28335 / TMS320F28334 / TMS320F2809 / C8051F506

명령어	구분	설명
	comment	<p>3rd word의 bit70이 1이면 : RS232C_MODE (F2811/F28334의 경우는 항상 1)</p> <p>3rd word의 bit60이 1이면 : RS485_MODE (2811에서 EtherNet-IP or USB 사용시는 선택 불가능)</p> <p>3rd word의 bit50이 1이면 : CAN_MODE</p> <p>3rd word의 bit40이 1이면 : COM_TEST_MODE</p> <p>3rd word의 bit3/20이 3 일때 CAN 11-bit 주소모드로서 주소로 0-255만 사용 2 일때 CAN 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용 1 일때 CAN 29-bit 주소모드로서 주소로 0-65535만 사용, 2012년 10월 16일부터 적용 0 일때 CAN 11-bit 주소모드로서 주소로 0-255만 사용</p> <p>3rd word의 last 2bit : 0/1 : 기타 통신모드 없음/EtherNet-IP or USB (2811/28334 일때만)</p> <p>(모터제어기의 경우)</p> <p>4th word의 highest 4bit : 1/2/3/4/5/6 : ONE_MOTOR_MODE / TWO_MOTOR_MODE / BLDC_ONE_MOTOR_MODE / BLDC_TWO_MOTOR_MODE / BLDC_ONE_MOTOR_SENSOR / BLDC_TWO_MOTOR_SENSOR</p> <p>4th word의 next 4bit : 1/2/3/4/5/6/7/8 : RJM_VER1/2/3/4/5/6/7/8 12/13/14 : RJM_VER8_from_VER7/RJM_VER7_3PHASE_STEP/RJM_VER7_5PHASE_STEP RJM_VER7_3PHASE_STEP/RJM_VER7_5PHASE_STEP는 2013년 7월 31일부터 적용.</p> <p>4th word의 next 4bit : 1/2/3/4/5/6/7/8/9 : VER7_ENCODER+digital Hall / VER7_analog Hall only / VER7_digital Hall only / JEIL_MEDICAL / NANO_MOTION_RJM_VER6 / NANO_MOTION_LM_ANALOG / BEST_MOTION / RJM_VER7_force_sensor / RJM_VER8_force_sensor RJM_VER7_force_sensor와 RJM_VER8_force_sensor는 2013년 4월 27일부터 적용.</p> <p>4th word의 lowest 4bit : 0/1/2/3/4/5/6 : CLOCK_60MHz / CLOCK_80MHz / CLOCK_100MHz / CLOCK_120MHz / CLOCK_120MHz_1 / CLOCK_150MHzHz / CLOCK_90MHzHz</p>

명령어	구분	설명
	comment	<p>(관성센서의 경우)</p> <p>4th word의 highest 4bit : 1/2/3/4 : Ver1 / Ver2(Analog Device MEMS chip 채용) / Ver3(Fujitsu Gyro, Kionix Acc 채용) / Ver4(Fujitsu Gyro, VTI Acc 채용)</p> <p>4th word의 lowest 4bit : 1/2/3/4/5 : CLOCK_80MHz / CLOCK_100MHz / CLOCK_120MHz / CLOCK_120MHz_1 / CLOCK_150MHz</p> <p>2013년 1월 10일부터 2013년 4월 26일 이전 S/W 버전에서는 리턴값에서 Q1 다음의 space가 생략되었음.</p>

Q4 명령에 의한 리턴값의 요약

Q4의 return 값중 세번째 워드

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	1 : BIPEDAL_ROBOT 용 S/W 2 : ROBOT_JOINT_MOTOR 용 S/W- 3 : ROBOT_INERTIAL_SENSOR 용 S/W- 4 : ROBOT_EYE 용 S/W- 5 : SYSTEM MONITORING 용 S/W 6 : HUMANOID 용 S/W 7 : PTC HEATER 용 S/W 8 : MAGNETIC BEARING 용 S/W 9 : ABS CONTROL 용 S/W a : RACK type POWER AMP 용 S/W				0 : TMS320LF2406A 1 : TMS320F2810 2 : TMS320F2811 or 12 3 : TMS320LF2403A 4 : C8051F411 5 : TMS320LF2407A 6 : TMS320F28335 7 : TMS320F28334 8 : TMS320F2809 9 : C8051F506 10 : TMS320F28062			
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	RS232C_MODE	RS485_MODE	CAN_MODE	COM_TEST_MODE	3 : CAN 11-bit 주소모드로서 주소로 0-255만 사용 2 : CAN 11-bit 주소모드로서 주소로 0-2047 사용 1 : CAN 29-bit 주소모드로서 주소로 0-65535만 사용 0 : CAN 11-bit 주소모드로서 주소로 0-255만 사용		0 : 기타 통신모드 없음 1 : EtherNet-IP or USB (2811/28334 일때만) 2 : ETHERCAT (2015년 3월 17일 S/W 버전 부터 ETHERCAT H/W option이 장착되고 firmware도 ETHERCAT version 탑재되었을 때만)	

Q4 명령에 의한 리턴값의 요약

Q4의 return 값중 네번째 워드 (모터제어기의 경우)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	1 : ONE_MOTOR_MODE 2 : TWO_MOTOR_MODE 3 : BLDC_ONE_MOTOR_MODE 4 : BLDC_TWO_MOTOR_MODE 5 : BLDC_ONE_MOTOR_SENSOR 6 : BLDC_TWO_MOTOR_SENSOR				1 : RJM_VER1 (초창기버전 DC) 2 : RJM_VER2 (초창기버전 DC) 3 : RJM_VER3 (DC/BLDC 공용의 초기버전, bit15-bit8 값에 따라서 1/2번 일대는 DC, 3/4이면 BLDC임) 4 : RJM_VER4 (50/100W DC) 5 : RJM_VER5 (50/100W STEP) 6 : RJM_VER6 (50/100W 삼상 STEP) 7 : RJM_VER7 (BLDC) 8 : RJM_VER8 (200W 이상의 DC) 12 : RJM_VER8_from_VER7 (BLDC 제어기 H/W로 DC 모터를 사용) 13 : RJM_VER7_3PHASE_STEP (BLDC 제어기 H/W로 3상 STEP 모터를 사용) 14 : RJM_VER7_5PHASE_STEP (BLDC 제어기 H/W로 5상 STEP 모터를 사용)			
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	1 : VER7_ENCODER+digital Hall 2 : VER7_analog Hall only 3 : VER7_digital Hall only 4 : JEIL_MEDICAL 5 : NANO_MOTION_RJM_VER6 6 : NANO_MOTION_LM_ANALOG 7 : BEST_MOTION 8 : RJM_VER7_force_sensor 9 : RJM_VER8_force_sensor				0 : CLOCK_60MHz 1 : CLOCK_80MHz 2 : CLOCK_100MHz 3 : CLOCK_120MHz 4 : CLOCK_120MHz_1 5 : CLOCK_150MHz 6 : CLOCK_90MHz			

Q4 명령에 의한 리턴값의 요약

Q4의 return 값중 네번째 워드 (관성센서의 경우)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	1 : Ver1 2 : Ver2(Analog Device MEMS chip 채용) 3 : Ver3(Fujitsu Gyro, Kionix Acc 채용) 4 : Ver4(Fujitsu Gyro, VTI Acc 채용)							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0					1 : CLOCK_80MHz 2 : CLOCK_100MHz 3 : CLOCK_120MHz 4 : CLOCK_120MHz_1 5 : CLOCK_150MHz			

명령어	구분	설명
;Q5; (+#^)	operation	operation1_status를 읽어서 host로 전송함.
	return value	Q5 HLHL; bit0 : position1_plus_limit_over_flg bit1 : position1_minus_limit_over_flg bit2 : position1_deceleration_at_limit_flg bit3 : speed1_deceleration_at_limit_flg bit4 : position1_by_PA_cmd_execute_flg bit5 : bit6 : linear_interpolation_execute_flg bit7 : positionX_by_PA_cmd_execute_flg bit0 : position2_plus_limit_over_flg bit1 : position2_minus_limit_over_flg bit2 : position2_deceleration_at_limit_flg bit3 : speed2_deceleration_at_limit_flg bit12 : position2_by_PA_cmd_execute_flg bit13 : bit14 : bit15 :
	comment	2013년 1월 10일부터 2013년 4월 26일 이전 S/W 버전에서는 리턴값에서 Q1 다음의 space가 생략되었음.

Q5 명령에 의한 operation1 status의 요약

Q5의 return 값								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8				position2_by_P A_cmd_execute_ flg	speed2_deceler ation_at_limit _flg	position2_dece leration_at_li mit_flg	position2_minu s_limit_over_f lg	position2_plus _limit_over_fl g
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	positionX_by_P A_cmd_execute_ flg	linear_interpo lation_execute _flg		position1_by_P A_cmd_execute_ flg	speed1_deceler ation_at_limit _flg	position1_dece leration_at_li mit_flg	position1_minu s_limit_over_f lg	position1_plus _limit_over_fl g

명령어	구분	설명
(저가형 1채널 제어가 아닌 경우) ;Q6; (+#^)	operation	conditional compiling 조건을 읽어서 host로 전송함.
	return value	<p>Q6 HLHL,HLHL; (2011년 9월 25일 이전 S/W 버전) Q6 HLHL,HLHL,HLHL,HLHL; (2011년 9월 25일 S/W 버전부터) Q6 HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (2013년 4월 27일 S/W 버전부터)</p> <p>1st word의 highest 4bit : ENCODER1_SCALE_FACTOR (모터1의 실제 엔코더 펄스수는 MOTOR1_PULSE_PER_REV * ENCODER1_SCALE_FACTOR 이며 ENCODER1_SCALE_FACTOR는 600, 7500, 72000을 나누어서 나누어지는 수이어야하며, 통상값은 1로서 엔코더의 펄스수가 32767을 넘는 경우에만 1이 아닌 값을 설정함) 1st word의 next 4bit : ENCODER2_SCALE_FACTOR (모터2의 실제 엔코더 펄스수는 MOTOR2_PULSE_PER_REV * ENCODER2_SCALE_FACTOR 이며 ENCODER2_SCALE_FACTOR는 600, 7500, 72000을 나누어서 나누어지는 수이어야하며, 통상값은 1로서 엔코더의 펄스수가 32767을 넘는 경우에만 1이 아닌 값을 설정함) 단 2013년 4월 27일 S/W 버전부터는 1st word의 highest 6bit는 1 : 자기부상회전기, 2 : ELECTRONIC_POWER_STEER, 3 : slit stage, 4 : LINEAR_MOTOR_CHO, 5 : 자석가동자, 6 : 고전압 안정화 구동 모터제어기, 7 : BRAKE_ACTUATOR, 8 : 해양로봇제어기 9 : RJM_VER7로 DC 모터를 구동하면서 상위제어기의 D/A 출력으로 전류구동을 하는 경우 next 1bit는 DEAD_TIME_COMPENSATION, next 1bit는 TWO_MOTOR_TOGETHER</p> <p>2014년 1월 27일 이전 S/W 버전에서는 1st word의 next 4bit : SEQUENCE_CONTROL (0 = no sequence control program, 1 = sequence control program, 2 = 프로와택) 2014년 1월 27일 S/W 버전부터는 1st word의 next 2bit : ETHERCAT_UPGRADE (0 = no ETHERCAT_UPGRADE, 1 = ETHERCAT_UPGRADE) 1st word의 next 2bit : SEQUENCE_CONTROL (0 = no sequence control program, 1 = sequence control program, 2 = 프로와택)</p>

명령어	구분	설명
	return value	<p>1st word의 last 4bit : RJM_VER7_UV_current (RJM_VER7만 해당, 0 = no U/V current sensing, 1 = U/V current at AD3/4/11/12, 2 = U/V/W current at AD13/14/15, 3 = same as 1 but trapezoidal, 4 = same as 2 but trapezoidal, 5 = same as 1 but total current + trapezoidal, 6 = same as 2 but total current + trapezoidal, 7 = 5상 스테핑모터의 경우, 8 = 삼상 스테핑모터의 경우)</p> <p>2nd word의 bit15 : FAULHABER_TYPE (0 = U 단자대 사용형식, 1 = Faulhaber type 스크류 터미널 형태)</p> <p>2nd word의 bit14 : NO_DIP_SWITCH (0 = with DIP switch, 1 = no DIP switch)</p> <p>2nd word의 bit13 : SWAP_LIMIT_SW (RJM_VER3 TMS320F2810/11으로서 2007년 8월경에 제작된 모델에만 해당, 0 = original type, 1 = swap limit switch1/switch2)</p> <p>2nd word의 bit12 : ADS7818 (RJM_VER3만 해당, 0 = 없는 경우, 1 = 있는 경우)</p> <p>2nd word의 bit11 : CURRENT_LOOP_50KHZ (RJM_VER5/6/4/8만 해당, 0 = 1KHz(RJM_VER5) / 100KHz(RJM_VER4/8) CURRENT_LOOP, 1 = 50KHz(RJM_VER5/6 TWO MOTOR MODE, RJM_VER4/8) / 100KHz(RJM_VER5/6 ONE MOTOR MODE) CURRENT_LOOP)</p> <p>2nd word의 bit10 : STEP_VER1P0 (RJM_VER5만 해당, 0 = I_FAULTx at F8/F9, 1 = I_FAULTx at E0/E1)</p> <p>2nd word의 bit9 : RJM_VER7_L6234 (RJM_VER7만 해당, 0 = no L6234, 1 = L6234)</p> <p>2nd word의 bit8 : NEW_GAIN_RULE (0 = old gain, 1 = new gain)</p> <p>2nd word의 bit7/6 : ANGLE_RESOLVING_METHOD bit1/0 (RJM_VER7 analog type만 해당, 000 = no analog Hall sensor, 001 = 아크사인/아크코사인법, 010 = 직선보간법, 011 = COS/SIN resolver)</p> <p>2nd word의 bit5 : WAFER_ROTATION (0 = non wafer control, 1 = NANO motion wafer control, RJM_VER7 digital Hall IC only 인 경우에만 해당)</p> <p>2nd word의 bit4 : DD_MOTOR (0 = Direct Drive Motor가 아닌 경우, 1 = Direct Drive Motor인 경우)</p> <p>2nd word의 bit3 : AC_INDUCTION_MOTOR (0 = AC induction motor가 아닌 경우, 1 = AC induction motor인 경우)</p> <p>2nd word의 bit2 : NO_HALL_BUT_HOME_SENSOR bit0 (00 = 통상의 경우, 01 = Encoder를 사용하지만 Hall IC는 없고 대신하여 홀센서를 사용하는 경우)</p> <p>2nd word의 bit1 : HIGH_RESOLUTION_ANGLE (0 = 통상의 경우, 1 = sin/cos 계산시에 분해능을 16배 높임)</p> <p>2nd word의 bit0 : AUTOMATION (0 = 통상의 경우, 1 = 자동화용 H/W 버전의 경우)</p>

명령어	구분	설명
	return value	<p>3rd word의 bit15 : RJM_VER7_MODULE (0 = 통상의 경우, 1 = RJM_VER7 MOSFET 모듈을 사용한 H/W 버전으로 서 DC 제어용으로 프로그램된 경우)</p> <p>3rd word의 bit14 : BIPOLAR_DRIVE (0 = 통상의 경우, 1 = DC 모터로서 bipolar 구동을 하는 경우)</p> <p>3rd word의 bit13 : RJM_VER4_PWM_AT_CH1245 (0 = PWM at PWM1+/PWM1-/PWM2+/PWM2-, 1 = PWM1+/PWM2+/PWM4+/PWM5+)</p> <p>3rd word의 bit12/11 : VERSION_380V bit1/0 (000 = IRAM136-3063B, 001 = IXYS-MUBW, 010 = MITA15WB1200, 011 = FP100R12KT4)</p> <p>3rd word의 bit10 : EtherNetIP_or_USB (0 = 통상의 경우, 1 = LAN/USB 를 사용하는 경우)</p> <p>3rd word의 bit9 : DIFFERENTIAL_HALL_IC (0 = 통상의 경우, 1 = 380V H/W 버전에서 differential Hall IC 를 사용하는 경우)</p> <p>3rd word의 bit8 : RJM_VER7_Hall_IC_ENC (0 = 통상의 경우, 1 = BLDC에서 엔코더를 위치센서로 사용하는 경우)</p> <p>3rd word의 bit7 : RJM_VER7_Hall_IC_analog (0 = 통상의 경우, 1 = BLDC에서 analog Hall 센서를 위치센서로 사용하는 경우)</p> <p>3rd word의 bit6 : RJM_VER7_Hall_IC_only (0 = 통상의 경우, 1 = BLDC에서 digital Hall 센서를 위치센서로 사용하는 경우)</p> <p>3rd word의 bit5/4 : ABSOLUTE_ENC bit1/0 (000 = 통상의 경우, 001 = Netzer 19bit Absolute Encoder 위치센서로 사용하는 경우, 010 = PWM type Absolute Encoder, 011 = Heidenheim 26bit Absolute Encoder)</p> <p>3rd word의 bit3 : NO_HALL_BUT_HOME_SENSOR의 bit1 (00 = 통상의 경우, 01 = Encoder를 사용하지만 Hall IC는 없고 대신하여 홀센서를 사용하는 경우) - 2013년 4월 27일 S/W 버전부터 적용됨.</p> <p>3rd word의 bit2 : LOW_RESOLUTION_ENCODER (0 = 통상의 경우, 1 = low resolution encoder를 위치센서로 사용하는 경우)</p> <p>3rd word의 bit1/0 : POTENTIOMETER (0 = potentiometer를 위치센서로 사용하지 않는 경우, 1 = LSxC 포트에 연결된 potentiometer 를 사용한 위치제어의 경우, 2 = digital potentiometer를 사용하는 경우)</p> <p>4th word의 highest 8bit : ENCODER1_SCALE_FACTOR</p> <p>4th word의 next 8bit : ENCODER2_SCALE_FACTOR</p> <p>5th word의 bit15 : RJM_VER8 from_VER7 (0 = 통상의 경우, 1 = BLDC 모터제어기 H/W에 DC 모터제어 S/W를 탑재한 경우)</p>
		<p>Copyright© 2008-2015 Robocubetech. All rights Reserved.</p> <p>5th word의 bit14 : RJM_VER7_DRV8332 (0 = 통상의 경우, 1 = T1사의 DRV8332를 구동칩으로 사용한 경우)</p>

명령어	구분	설명
	return value	<p>5th word의 bit13 : KOMOTEK_380V (0 = DC전원을 사용하는 모터제어기, 1 = AC전원을 사용하는 모터제어기)</p> <p>5th word의 bit12 :</p> <p>5th word의 bit11 : IRS2336 (0 = 통상의 경우, 1 = 모터구동용 MOSFET 또는 IGBT의 게이트구동 칩으로 IRS2336을 사용하는 경우)</p> <p>5th word의 bit10 : IRF7779 (0 = 통상의 경우, 1 = IR사의 150V MOSFET IRF7779를 구동칩으로 사용한 경우)</p> <p>5th word의 bit9 : DRV8432 (0 = 통상의 경우, 1 = TI사의 DRV8432 or DRV8332를 모터구동칩으로 사용하는 모터제어기)</p> <p>5th word의 bit8 :</p> <p>5th word의 bit7 : ANGLE_RESOLVING_METHOD bit2 (RJM_VER7 analog type만 해당, 000 = no analog Hall sensor, 001 = 아크사인/아크코사인법, 010 = 직선보간법, 011 = COS/SIN resolver) - 2013년 4월 27일 S/W 버전부터 적용됨.</p> <p>5th word의 bit6 : VERSION_380V bit2 (000 = IRAM136-3063B, 001 = IXYS-MUBW, 010 = MITA15WB1200, 011 = FP100R12KT4) - 2013년 4월 27일 S/W 버전부터 적용됨.</p> <p>5th word의 bit5 : ABSOLUTE_ENC bit2 (000 = 통상의 경우, 001 = Netzer 19bit Absolute Encoder 위치센서로 사용하는 경우, 010 = PWM type Absolute Encoder, 011 = Heidenheim 26bit Absolute Encoder) - 2013년 4월 27일 S/W 버전부터 적용됨.</p> <p>5th word의 bit4 :</p> <p>5th word의 bit3 : IR21771S (0 = 전류센서로 IR21771S를 사용하지 않는 경우, 1 = 전류센서로 IR21771S를 사용하는 경우)</p> <p>5th word의 bit2/1/0 : CURRENT_SENSING (AC전원을 사용하는 경우에 전류센싱방법을 지정하며, 0 = by IR2175S, 1 = by IR21771S, 2 = by Hall sensor (C8051), 3 = by Hall sensor (TMS320F28027))</p> <p>6th word의 bit15 : PITCH_51mm_200um (0 = 통상의 경우, 1 = 51mm pitch의 shaft 모터와 200um pitch의 analog resolver를 사용한 경우)</p>

명령어	구분	설명
	return value	<p>6th word의 bit14 : PITCH_18mm_20um (0 = 통상의 경우, 1 = 18mm pitch의 보이스코일 모터와 20um pitch의 analog resolver를 사용한 경우)</p> <p>6th word의 bit13 : USE_INTERPOLATOR (0 = 통상의 경우, 1 = 내장형 interpolator를 사용하는 경우)</p> <p>6th word의 bit12 : ANALOG_HALL_at_LS1C (0 = 통상의 경우, 1 = 자기부상회전에서 analog Hall 센서를 LS1C 포트에 인터페이스하는 경우)</p> <p>6th word의 bit11 : PRESSURE_SENSING (0 = 통상의 경우, 1 = 브레이크제어기에서 압력제어방식을 사용하는 경우)</p> <p>6th word의 bit10 : WIDE_PAN_TILT_TYPE (0 = 통상의 경우, 1 = 팬/틸트 모터제어기에서 팬각이 큰 경우)</p> <p>6th word의 bit9 :</p> <p>6th word의 bit8 : SAMSUNG_MEDICAL_4_SENSOR (0 = 통상의 경우, 1 = F/T 센서에서 4-full bridge를 사용한 삼성납품용의 경우)</p> <p>6th word의 bit7 : TLV5630_DAC (0 = 통상의 경우, 1 = F/T 센서에서 TLV5630_DAC를 사용하는 경우)</p> <p>6th word의 bit6 : DAC124S085 (0 = 통상의 경우, 1 = F/T 센서에서 DAC124S085를 사용하는 경우)</p> <p>6th word의 bit5 : VER_44PHI_V1P0 (0 = 통상의 경우, 1 = 44mm 직경의 F/T 센서에서 최초 PCB버전의 경우)</p> <p>6th word의 bit4 : USE_THREE_AMP (0 = 통상의 경우, 1 = F/T 센서에서 full bridge 4개 중에 3개만 사용하여 2개의 모멘트와 1개의 힘을 측정하며 나머지 한개의 full bridge로 별도의 load cell을 interface하도록 사용하는 경우)</p> <p>6th word의 bit3/2 : LINEAR_TEMP_SENSOR (00 = 통상의 경우, 01 = F/T 센서에서 선형 온도센서를 사용하는 경우, 10 = T type thermocouple(MAX31855)을 사용하는 경우)</p> <p>6th word의 bit1 : SIX_AXIS (0 = 통상의 경우, 1 = F/T 센서에서 6-full bridge를 사용하여 3개의 힘과 3개의 모멘트를 측정하는 경우)</p> <p>6th word의 bit0 : DA_2_3_SWAP (0 = 통상의 경우, 1 = F/T 센서에서 DC 변환기 2,3번 출력을 swap한 경우)</p>
	comment	<p>2011년 9월 25일 S/W 버전부터 3/4th word가 추가되었음.</p> <p>2013년 4월 27일 S/W 버전부터 5/6th word가 추가되었음.</p> <p>2013년 1월 10일부터 2013년 4월 26일 이전 S/W 버전에서는 리턴값에서 Q1 다음의 space가 생략되었음.</p>

Q6 명령에 의한 conditional compiling 조건의 요약

Q6의 첫번째 return 값									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
bit15-8	special_application_identification						DEAD_TIME_COMPENSATION	TWO_MOTOR_TOGETHER	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit7-0	ETHERCAT_UPGRADE		SEQUENCE_CONTROL		RJM_VER7_UV_current				
Q6의 두번째 return 값									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
bit15-8	FAULHABER_TYPE	NO_DIP_SWITCH	SWAP_LIMIT_SW	ADS7818	CURRENT_LOOP_50KHZ	STEP_VER1P0	RJM_VER7_L6234	NEW_GAIN_RULE	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit7-0	ANGLE_RESOLVING_METHOD bit1/0		WAFER_ROTATION	DD_MOTOR	AC_INDUCTION_MOTOR	NO_HALL_BUT_HOME_SENSOR bit0	HIGH_RESOLUTION_ANGLE	AUTOMATION or AUTOMATION_DRV8332 or AUTOMATION_DRV8432	
Q6의 세번째 return 값									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
bit15-8	RJM_VER7_for_VER8	BIPOLAR_DRIVE	RJM_VER4_PWM_AT_CH 1245	0 or VERSION_380V bit1/0		EtherNetIP_or_USB or 1	0 or DIFFERENTIAL_HALL_IC	RJM_VER7_Hall_IC_ENC	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit7-0	RJM_VER7_Hall_IC_analog	RJM_VER7_Hall_IC_only	ABSOLUTE_ENC bit1/0		NO_HALL_BUT_HOME_SENSOR bit1	LOW_RESOLUTION_ENCODER	POTENTIOMETER bit1/0		
Q6의 네번째 return 값									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
bit15-8	ENCODER1_SCALE_FACTOR								
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit7-0	ENCODER2_SCALE_FACTOR								
Q6의 다섯번째 return 값									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
bit15-8	RJM_VER8_from_VER7	RJM_VER7_DRV8332	0 or KOMOTEK_380V	-	IRS2336	0 or IRF7779	DRV8432	-	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit7-0	ANGLE_RESOLVING_METHOD bit2	0 or VERSION_380V bit2	ABSOLUTE_ENC bit2	-	IR21771S	0 or CURRENT_SENSING			
Q6의 여섯번째 return 값									
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
bit15-8	PITCH_51mm_200um	PITCH_18mm_20um	0 or USE_INTERPOLATOR	ANALOG_HALL_at_LSI_C	0 or PRESSURE_SENSING	WIDE_PAN_TILT_TYPE	-	SAMSUNG_MEDICAL_4_SENSOR	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
bit7-0	TLV5630_DAC	DAC124S085	VER_44PHI_V1P0	USE_THREE_AMP	LINEAR_TEMP_SENSOR		SIX_AXIS	DA_2_3_SWAP	

붉은색 부분이 2013년 4월 27일자 S/W 버전부터 추가 또는 변경되었음.



명령어	구분	설명
(저가형 1채널 제어기인 경우) ;Q6; (+#^)	operation	conditional compiling 조건을 읽어서 host로 전송함.
	return value	Q6 HLHL,HLHL,HLHL; 1st word : 모터 type (0 = BLDC motor, 1 = induction motor, 2 = DC motor) 2nd word : 센서 type (0 = 엔코더, 1 = digital 3상 Hall sensor, 2 = analog 3상 Hall sensor, 4 = potentiometer, 5 = PWM type potentiometer, 6 = SPI type poentniometer type 1, 7 = SPI type poentniometer type 2, 8 = low resolution encoder, 9 = absolute encoder (Netzer 19bit), 10 = absolute encoder (Heidenheim 26bit)) 3rd word : BLDC의 영구자석 위치 검출 type (0 = digital 3상 Hall sensor 를 사용하는 경우, 1 = digital 3상 Hall sensor 가 없고 홀센서를 사용하는 경우, 2 = digital 3상 Hall sensor와 홀센서가 모두 없는 경우)

명령어	구분	설명
;Q7; (+#^)	operation	motor_status_word 를 읽어서 host로 전송함. (2013년 8월 08일 S/W 버전부터)
	return value	Q7 HLHL; bit0 : motor1_moving_flg bit1 : motor1_fault_flg bit2 : motor1_home_searching_status_abstract bit0 bit3 : motor1_home_searching_status_abstract bit1 bit4 : motor2_moving_flg bit5 : motor2_fault_flg bit6 : motor2_home_searching_status_abstract bit0 bit7 : motor2_home_searching_status_abstract bit1 bit8 : bit9 : bit10 : bit11 : bit12 : bit13 : bit14 : bit15 :
	comment	motor1_home_searching_status_abstract, motor2_home_searching_status_abstract 는 홈동작 진행/성공여부 임. (0이면 홈동작명령 실시된 바 없음, 1이면 성공, 2이면 실패, 3이면 홈동작 중) motor1_fault_flg=1 이면 Q2 명령의 리턴값에서 ((fault_status & 0x055d) (fault1_status & 0x6d10))!=0 일때 임. 자세한 오류내역은 Q2 명령어 참조. motor2_fault_flg=1 이면 Q2 명령의 리턴값에서 ((fault_status & 0x50ae) (fault1_status & 0x9610))!=0 일때 임. 자세한 오류내역은 Q2 명령어 참조. motor1_moving_flg, Q1 명령어 리턴값의 bit 2와 같음. motor2_moving_flg, Q1 명령어 리턴값의 bit 10와 같음. 2013년 8월 08일 S/W 버전부터 사용 가능한 명령어임.

Q7의 return 값 (내부 변수명은 motor_status_word)								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8								
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	motor2_home_se arching_status _abstract bit1	motor2_home_se arching_status _abstract bit0	motor2_fault_f lg	motor2_moving_ flg	motor1_home_se arching_status _abstract bit1	motor1_home_se arching_status _abstract bit0	motor1_fault_f lg	motor1_moving_ flg

명령어	구분	설명
<pre> :QRHLHL,HL; (+#^%) :QQHLHL,HL; (+#^) </pre>	operation	<p>지정된 주소에서 시작하여 지정된 양만큼의 DSP SRAM 내용을 읽어서 host로 전송함.</p> <p>first word : RAM address (TMS320F810/11/12)</p> <p>[0000 - 017F] : M0 RAM block (.stack) [0180 - 01BF] : M0 RAM block (.bss)</p> <p>[01C0 - 06FF] : M0/1 RAM block (.ebss) [0700 - 07FF] : M1 RAM block (asm에서 사용)</p> <p>[0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용)</p> <p>[E000 - FFFF] : H0 RAM block (.esysmem)</p> <p>first word : RAM address (TMS320F28334/28335)</p> <p>[0000 - 004F] : M0 RAM block (stack for boot) [0050 - 03FF] : M0 RAM block (no use)</p> <p>[0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss)</p> <p>[0880 - 793F] : peripheral memory [8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용)</p> <p>[C000 - DFFF] : L4/L5 RAM block (.ebss) [E000 - FFFF] : L6/L7 RAM block (.esysmem)</p> <p>second byte : number of words [00 - 07]</p> <p>2013년 11월 24일 S/W 버전부터는 00 - 0F 범위에서 설정이 가능함.</p>
	return value	<p>QRHLHL,HL,HLHL(,HLHL); 괄호안의 값이 추가로 리턴되는 값.</p> <p>QQHLHL(,HLHL); 괄호안의 값이 추가로 리턴되는 값.</p>
	comment	<p>return 값은 입력으로 들어온 값에다 RAM을 읽은 값들이 뒤에 덧붙여짐.</p> <p>number of words 가 00이면 한 개의 word를 의미하며, 07일 때 최대 8 word 까지 읽을 수 있음.</p> <p>2013년 11월 24일 S/W 버전부터는 number of words 가 0F일 때 최대 16 word 까지 읽을 수 있음.</p> <p>7010(0880) - 7531(783F) 영역의 주소에 있는 데이터를 읽을 때, 주소값이 정확치 않으면 DSP가 reset됨.</p> <p>TMS320F810/11/12의 경우 [E000 - FFFF] 주소 영역은 실제주소로는 [3F8000 - 3F9FFF] 주소영역에 해당함.</p> <p>[0600 - 07FF], [E000 - FFFF] 영역은 cpu_asm.asm에서 선언된 변수들이 저장되는 영역임.</p> <p>[01C0 - 06FF] 영역은 C에서 선언된 전역변수들이 저장되는 곳임.</p> <p>사용 가능한 주소영역과 각 주소의 변수값의미는 S/W 버전마다 다를수 있음.</p> <p>E200 - E9FF 범위를 읽으면 EEPROM에 저장되는 파라미터를 읽을수 있음.</p>

명령어	구분	설명
;QTHLHL,HL,HLHL (,HLHL); (+#^%)	operation	<p>지정된 주소에서 시작하여 지정된 양만큼의 데이터를 DSP의 SRAM에 전송함.</p> <p>first word : RAM address (TMS320F810/11/12)</p> <p>[0000 - 017F] : M0 RAM block (.stack) [0180 - 01BF] : M0 RAM block (.bss)</p> <p>[01C0 - 06FF] : M0/1 RAM block (.ebss) [0700 - 07FF] : M1 RAM block (asm에서 사용)</p> <p>[0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용)</p> <p>[E000 - FFFF] : H0 RAM block (.esysmem)</p> <p>first word : RAM address (TMS320F28334/28335)</p> <p>[0000 - 004F] : M0 RAM block (stack for boot) [0050 - 03FF] : M0 RAM block (no use)</p> <p>[0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss)</p> <p>[0880 - 793F] : peripheral memory [8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용)</p> <p>[C000 - DFFF] : L4/L5 RAM block (.ebss) [E000 - FFFF] : L6/L7 RAM block (.esysmem)</p> <p>second byte : number of words [00 - 07]</p> <p>2013년 11월 24일 S/W 버전부터는 00 - 0F 범위에서 설정이 가능함.</p> <p>following word : DSP의 SRAM에 쓸 데이터.</p>
	return value	QTHLHL,HL,HLHL(,HLHL); 괄호안의 값이 추가로 리턴되는 값.
	comment	<p>return 값은 RAM에 값을 설정한 후 다시 읽은 값들이 뒤에 덧붙여짐.</p> <p>number of words 가 00이면 한 개의 word를 의미하며, 07일 때 최대 8 word 까지 쓸 수 있음.</p> <p>2013년 11월 24일 S/W 버전부터는 number of words 가 0F일 때 최대 16 word 까지 읽을 수 있음.</p> <p>7010(0880) - 7531(783F) 영역의 주소에 데이터를 쓸 때, 주소값이 정확치 않으면 DSP가 reset됨.</p> <p>TMS320F810/11/12의 경우 [E000 - FFFF] 주소 영역은 실제주소로는 [3F8000 - 3F9FFF] 주소영역에 해당함.</p> <p>[0600 - 07FF], [E000 - FFFF] 영역은 cpu_asm.asm에서 선언된 변수들이 저장되는 영역임.</p> <p>[01C0 - 06FF] 영역은 C에서 선언된 전역변수들이 저장되는 곳임.</p> <p>사용 가능한 주소영역과 각 주소의 변수값의미는 S/W 버전마다 다를수 있음.</p> <p>E200 - E9FF 범위에 값을 쓰면 EEPROM에 저장하기 위한 파라미터를 바꿀수 있으나 EEPROM에 실제로 저장하려면 ES 명령어나 Es 명령을 사용하여야함.</p>

명령어	구분	설명
;QUHLHL; (+#^) ;QUIHLHL; (+#^) ;QULHLHL; (+#^) ;QUuHLHL; (+#^) ;QUUHLHL; (+#^) ;QUhHLHL; (+#^) ;QUHHLHL; (+#^) 	operation	지정된 주소의 DSP SRAM에서 실수/정수값을 읽어서 host로 전송함. first word : RAM address (TMS320F810/11/12) [0000 - 017F] : M0 RAM block (.stack) [0180 - 01BF] : M0 RAM block (.bss) [01C0 - 06FF] : M0/1 RAM block (.ebss) [0700 - 07FF] : M1 RAM block (asm에서 사용) [0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용) [E000 - FFFF] : H0 RAM block (.esysmem) [E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨. first word : RAM address (TMS320F28334/28335) [0000 - 004F] : M0 RAM block (stack for boot) [0050 - 03FF] : M0 RAM block (no use) [0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss) [0880 - 793F] : peripheral memory [8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용) [C000 - DFFF] : L4/L5 RAM block (.ebss) [E000 - FFFF] : L6/L7 RAM block (.esysmem)
	return value	QUHLHL,ffff1; QUIHLHL,iiii1; QULHLHL,iiii1; QUuHLHL,ddd1; QUUHLHL,ddddddd1; QUhHLHL,HLHL; QUHHLHL,HLHLHLHL; ", " 이후의 값이 추가로 리턴되는 값.
	comment	7010(0880) - 7531(783F) 영역의 주소에 있는 데이터를 읽을 때, 주소값이 정확치 않으면 DSP가 reset됨. TMS320F810/11/12의 경우 [E000 - FFFF] 주소 영역은 실제주소로는 [3F8000 - 3F9FFF] 주소영역에 해당함. [0600 - 07FF], [E000 - FFFF] 영역은 cpu_asm.asm에서 선언된 변수들이 저장되는 영역임. [01C0 - 06FF] 영역은 C에서 선언된 전역변수들이 저장되는 곳임. 사용 가능한 주소영역과 각 주소의 변수값의미는 S/W 버전마다 다를수 있음. E200 - E9FF 범위를 읽으면 EEPROM에 저장되는 파라미터를 읽을수 있음.

명령어	구분	설명
<pre> ;QuHLHL,ffff1; (+#^) ;QuiHLHL,iiii1; (+#^) ;QuLHLHL,IIII1; (+#^) ;QuuHLHL,dddd1; (+#^) ;QuUHLHL,ddddddd1; (+#^) ;QhHLHL,HLHL; (+#^) ;QuHLHL,HLHLHLHL; (+#^) </pre> <p>관성센서에서는 Qu가 아니고 QW를 사용함.</p>	operation	<p>지정된 주소의 DSP SRAM에 실수/정수값을 저장함.</p> <p>first word : RAM address (TMS320F810/11/12)</p> <p>[0000 - 017F] : M0 RAM block (.stack) [0180 - 01BF] : M0 RAM block (.bss)</p> <p>[01C0 - 06FF] : M0/1 RAM block (.ebss) [0700 - 07FF] : M1 RAM block (asm에서 사용)</p> <p>[0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용)</p> <p>[E000 - FFFF] : H0 RAM block (.esysmem) [E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨.</p> <p>first word : RAM address (TMS320F28334/28335)</p> <p>[0000 - 004F] : M0 RAM block (stack for boot) [0050 - 03FF] : M0 RAM block (no use)</p> <p>[0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss)</p> <p>[0880 - 793F] : peripheral memory [8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용)</p> <p>[C000 - DFFF] : L4/L5 RAM block (.ebss) [E000 - FFFF] : L6/L7 RAM block (.esysmem)</p> <p>2nd number : DSP의 SRAM에 쓸 데이터.</p>
	return value	<pre> QuHLHL,ffff1; QuiHLHL,iiii1; QuLHLHL,IIII1; QuuHLHL,dddd1; QuUHLHL,ddddddd1; QhHLHL,HLHL; QuHLHL,HLHLHLHL; </pre>
	comment	<p>7010(0880) - 7531(783F) 영역의 주소에 데이터를 쓸 때, 주소값이 정확치 않으면 DSP가 reset됨.</p> <p>TMS320F810/11/12의 경우 [E000 - FFFF] 주소 영역은 실제주소로는 [3F8000 - 3F9FFF] 주소영역에 해당함. [0600 - 07FF], [E000 - FFFF] 영역은 cpu_asm.asm에서 선언된 변수들이 저장되는 영역임.</p> <p>[01C0 - 06FF] 영역은 C에서 선언된 전역변수들이 저장되는 곳임.</p> <p>사용 가능한 주소영역과 각 주소의 변수값의미는 S/W 버전마다 다를수 있음.</p> <p>E200 - E9FF 범위에 값을 쓰면 EEPROM에 저장하기 위한 파라미터를 바꿀수 있으나 EEPROM에 실제로 저장하려면 ES 명령어나 Es 명령을 사용하여야함.</p> <p>관성센서에서는 Qu가 아니고 QW를 사용함.</p>

명령어	구분	설명
;QrHL,HL; (+#^%)	operation	BLDC 모터제어용 chip인 IRMCK201 chip에 있는 register의 지정된 주소로부터 register 값을 읽어서 host로 전송함. (BLDC 구동 chip IRMCK201 전용) first byte : register address [00 - 43] second byte : motor number [00 - 01]
	return value	QrHL,HL,HLHL;
	comment	return 값은 입력으로 들어온 값에다 IRMCK201 register를 읽은 값이 뒤에 덧붙여짐. register에서 읽힌 값이 DSP의 SARAM에 저장되지는 않음. return된 세번째 word의 low byte가 첫번째 바이트로 지정된 주소의 register값임. BLDC인 경우에 한하여만 사용이 가능한 명령임.
;QtHL,HL,HL; (+#^%) ;QtHL,HL,HLHL; (+#^%) ;QtHL,HL?; (+#^%)	operation	BLDC 모터제어용 chip인 IRMCK201 chip에 있는 register의 지정된 주소에 host로부터 전송된 데이터를 저장함과 동시에 DSP의 SARAM 0x0a3e-0x0abd 번지에도 저장함. (BLDC 구동 chip IRMCK201 전용) first byte : register address [00 - 7D] second byte : motor number [00 - 01] third byte : register에 저장할 값 [byte 또는 word]
	return value	QtHL,HL,HL; 또는 QtHL,HL,HLHL;
	comment	return 값은 호스트에서 입력으로 들어온 값임. 00-7D번지 값은 DSP의 SARAM 0x0a3e-0x0abd 번지에도 저장됨. ?인 경우는 DSP의 SARAM 0x0a3e-0x0abd 번지에 저장된 word로 echo를 하며, echo된 word의 low byte가 첫번째 바이트로 지정된 주소의 값에 해당함. BLDC인 경우에 한하여만 사용이 가능한 명령임.

Qr, Qt 명령을 사용한 IRMCK201 BLDC motor 구동 IC의 parameter의 읽기와 쓰기

- ◎ BLDC모터1/2의 parameter는 DSP의 SARAM 0x0a3e-0x0abd번지 (F2811의 경우는 0x3f823e-0x3f82bd, F28334의 경우는 0xE23e-E2bd) 에 IRMCK201의 register 0x00-0x7d 번지에 해당하는 값이 각각 저장된다.
- ◎ BLDC모터1/2의 Hall sensor에 대한 Z pulse의 offset(ZEntCnt)는 같은 종류의 모터라도 모터의 조립상태마다 다른 값인데 DSP의 SARAM 0x0a41/0x0a81번지 (F2811의 경우는 0x3f8241/0x3f8281, F28334의 경우는 0xE241/E281) 에 저장된다.
- ◎ BLDC모터1/2의 parameter는 Qtxx,xx,xx; 또는 Qtxx,xx,xxxx; 명령어로 바꿀수가 있으며 이때 DSP의 SARAM 0x0a3e-0x0abd번지 (F2811의 경우는 0x3f823e-0x3f82bd, F28334의 경우는 0xE23e-E2bd) 뿐아니라 IRMCK201의 write register 0x00-0x7d 번지로도 값이 각각 저장된다. DSP의 EEPROM에 저장하려면 ESOA00,0100; 명령어를 사용하여야 한다.
- ◎ BLDC모터1/2의 parameter는 Qrxx,xx; 명령어로 읽을 수가 있으며 IRMCK201의 read register 0x00-0x43 번지의 값이 읽혀져서 host로 전송된다. 이때 DSP의 SARAM에 저장되지 않으며 IRMCK201의 read register의 주소는 IRMCK201의 write register 주소와 그 내용이 다르므로 주의하여야 한다.
- ◎ 전원이 최초 투입되면 BLDC모터의 parameter는 EEPROM에서 DSP의 0x0a3e-0x0abd번지 (F2811의 경우는 0x3f823e-0x3f82bd, F28334의 경우는 0xE23e-E2bd)로 읽혀들인 다음, 그 중에서 일부 parameter가 IRMCK201의 register 0x00-0x7d 번지로 각각 저장된다.
- ◎ DSP의 SARAM 0x0a3e-0x0abd번지 (F2811의 경우는 0x3f823e-0x3f82bd, F28334의 경우는 0xE23e-E2bd)에 있는 값은 QT 명령어로 강제 설정하지 않는 한 IRMCK201의 register에 있는 값과 항상 같다.

IRMCK201 BLDC motor 구동 IC의 parameter (for reading)

address	register name	description	비고
0x00	EncCntR(L)	Current value of 16-bit Quadrature Decoder counter.	
0x01	EncCntR(M)		
0x03	EncStatus	spare PwrOnHalIC PwrOnHalIB PwrOnHalIA spare HalIC HalIB HalIA (LSB)	
0x07	SysStatus	Start Stop spare PwrID PwrID GateKill FocEnbR PwmEnbR (LSB)	
0x08	RevCode(L)	IC Revision Code. Revision code format is "XX.XX", where each "X" is a 4-bit hexadecimal number.	
0x09	RevCode(M)		
0x0a	DcBusStatus(L)	DC Bus Voltage. Data range is 0 - 4095, which corresponds to a DC bus voltage between 0 and 500 volts. 0x0b bit4 : Brake signal status. 0 = Brake signal active.	
0x0b	DcBusStatus(M)		
0x0c	IfbVRawCurr(L)	Offset-corrected V and W phase raw current from the IR2175 current sensor. Values range from 0 - 4096, where 2048 corresponds to 0 current. The current feedback scale factors IdScl and IqScl in the CurrentFeedbackConfig write register group and the current sense resistor value determine the full scale current value.	
0x0d	IfbWRawCurr(L)		
0x0e	IfbVRawCurr(M)		
	IfbWRawCurr(M)		
0x0f	Id(L)	Synchronous or rotating frame direct and quadrature current values in 2's complement representation. The full scale current values range from -16384 to 16383.	
0x10	Id(M)		
0x11	Iq(L)	Synchronous or rotating frame direct and quadrature current values in 2's complement representation. The full scale current values range from -16384 to 16383.	
0x12	Iq(M)		
0x13	Ud(L)	Synchronous or rotating frame direct and quadrature voltage values in 2's complement representation. Data ranges are $\pm VdLim$ for Ud and $\pm VqLim$ for Uq.	
0x14	Ud(M)		
0x15	Uq(L)	Synchronous or rotating frame direct and quadrature voltage values in 2's complement representation. Data ranges are $\pm VdLim$ for Ud and $\pm VqLim$ for Uq.	
0x16	Uq(M)		
0x17	UaStatFrmVolt(L)	Stationary frame Alpha and Beta voltage output component values. Data range is $\pm VdLim$ or $\pm VqLim$ (as specified in the CurrentLoopConfig write register group), whichever is larger.	
0x18	UbStatFrmVolt(L)		
	UaStatFrmVolt(M)		
0x19	UbStatFrmVolt(M)		

IRMCK201 BLDC motor 구동 IC의 parameter (for reading)

address	register name	description	비고
0x1e	FltStatus	spare spare spare ExecTmFlt OvrSpdFlt OvFlt LvFlt GatekillFlt (LSB)	
0x26	Spd(L)	Current motor speed in SPEED units.	
0x27	Spd(M)	(See the description of SpdScl in the VelocityControl write register group.)	
0x30	lfbVOffset(L)	Current feedback offset values from the last IFB Offset calculation. These values are automatically applied to each current feedback measurement value whenever the lfbOfsEnb bit in the SystemControl write register group is set.	이 값을 읽어서 write resister 0x10/11/12에 쓰고 저장하여야함.
0x31	lfbWOffset(M)		
0x32	lfbVOffset(L)		
	lfbWOffset(M)		
0x34	EncCnt32bR_LW(L)	Current value of 32-bit Quadrature Decoder counter.	
0x35	EncCnt32bR_LW(M)		
0x36	EncCnt32bR_HW(L)		
0x37	EncCnt32bR_HW(M)		
0x38	EeStatus	0x38 bit0 : I2C EEPROM Interface busy bit.	
0x39	EeDataR	EEPROM Data Register. Contains the data from the last EEPROM read operation.	
0x3a	EeAddrR	EEPROM Address read register shows the value stored in EEPROM at the offset of the EeAddrW write register (0x5D).	
0x3b	RegMapVer	Current register map version code.	
0x3c	ElecAngleStatus(L)	Electrical angle.	
0x3d	ElecAngleStatus(M)		
0x3e	SpdRef(L)	Speed PI controller reference input.	
0x3f	SpdRef(M)		
0x40	SpdErr(L)	Speed PI controller error.	
0x41	SpdErr(M)		
0x42	IqRefR(L)	Speed PI controller output.	
0x43	IqRefR(M)		

IRMCK201 BLDC motor 구동 IC의 parameter (for writing)

address	register name	EC 45 50W flat 모터	EC-powermax 30 100W	MAXON 136210 250W	FAULHARBER 4490 200W	
0x00	EncCntW(L)	0 (0x0000)	0 (0x0000)	0 (0x0000)	0 (0x0000)	Hall sensor의 값에 따라서 반드시 수동 초기화하고 저장해야 함.
0x01	EncCntW(M)					
0x03	MaxEncCnt(L)	4095 (0x0fff)	3999 (0x0f9f)	1999 (0x07cf)	1999 (0x07cf)	4체배된 값 - 1
0x04	MaxEncCnt(M)					
0x06	ZEncCnt(L)	277 (0x0115)	1583 (0x062f)	1833 (0x0729)	1750 (0x06d6)	Z 펄스의 위치값
0x07	ZEncCnt(M)					
0x09	EncAngScI(L)	32767 (0x7fff)	8389 (0x20c5)	8389 (0x20c5)	8389 (0x20c5)	
0x0a	EncAngScI(M)					
0x0b	EncCtrl	15 (0x0f)	15 (0x0f)	15 (0x0f)	15 (0x0f)	
0x0c	PwmCtrl	178 (0xb2) or 50(0x32)	178 (0xb2) or 50(0x32)	178 (0xb2) or 50(0x32)	178 (0xb2) or 50(0x32)	50(0x32)는 gatekill 기능을 안쓰는 경우.
0x0d	PwmPeriodConfig	4296 (0x10c8)	4296 (0x10c8)	4296 (0x10c8)	4296 (0x10c8)	
0x0e		PwmPeriod : 200 PwmConfig : 1	PwmPeriod : 200 PwmConfig : 1	PwmPeriod : 200 PwmConfig : 1	PwmPeriod : 200 PwmConfig : 1	
0x0f	PwmDeadTm	20(0x14)	50 (0x32)	50 (0x32)	50 (0x32)	20 : 0.6 microsec
0x10	IbOffsetV(L)	IbOffsetV : 0 IbOffsetW : 0	IbOffsetV : 0 IbOffsetW : 0	IbOffsetV : 0 IbOffsetW : 0	IbOffsetV : 0 IbOffsetW : 0	반드시 수동 초기화하고 저장해야함.
0x11	IbOffsetW(L)					
0x12	IbOffsetV(M)					
	IbOffsetW(M)					
0x13	IdScI(L)	5405 (0x151d)	4019 (0x0fb3)	1134 (0x046e)	3486 (0x0d9e)	
0x14	IdScI(M)					
0x15	IqScI(L)	5405 (0x151d)	4019 (0x0fb3)	1134 (0x046e)	3486 (0x0d9e)	
0x16	IqScI(M)					
0x17	SysCtrl	0	0	0	0	

IRMCK201 BLDC motor 구동 IC의 parameter (for writing)

address	register name	EC 45 50W flat 모터	EC-powermax 30 100W	MAXON 136210 250W	FAULHARBER 4490 200W	
0x18	IqRefW(L)	0	0	0	0	
0x19	IqRefW(M)					
0x1a	Kplreg(L)	2037 (0x07f5)	755 (0x02f3)	1219 (0x04c3)	1067 (0x042b)	
0x1b	Kplreg(M)					
0x1c	KxIreg(L)	899 (0x0383)	476 (0x01dc)	595 (0x0253)	641 (0x0281)	
0x1d	KxIreg(M)					
0x1e	IdRef(L)	0	0	0	0	
0x1f	IdRef(M)					
0x20	SlipGn(L)	0	0	0	0	
0x21	SlipGn(M)					
0x22	VqLim(L)	914 (0x0392)	1421 (0x058d)	1428 (0x0594)	1429 (0x0595)	
0x23	VqLim(M)					
0x26	VdLim(L)	0 (0x00c0)	0 (0x00c0)	0 (0x00c0)	0 (0x00c0)	
0x27	VdLim(M)					
0x31	VelCtrl	15 (0x0f)	15 (0x0f)	15 (0x0f)	13 (0x0d)	15 : SpdLpRate 7
0x32	KpSreg(L)	392 (0x0188)	57 (0x0039)	187 (0x0066)	187 (0x0066)	
0x33	KpSreg(M)					
0x34	KxSreg(L)	169 (0x00a9)	21 (0x015)	81 (0x0051)	69 (0x0045)	
0x35	KxSreg(M)					
0x36	SregLimP(L)	5000 (0x1388)	12000 (0x2ee0)	8000 (0x1f40)	8000 (0x1f40)	
0x37	SregLimP(M)					
0x38	SregLimN(L)	-5000 (0xec78)	-12000 (0xd120)	-8000 (0xe0c0)	-8000 (0xe0c0)	
0x39	SregLimN(M)					

IRMCK201 BLDC motor 구동 IC의 parameter (for writing)

address	register name	EC 45 50W flat 모터	EC-powermax 30 100W	MAXON 136210 250W	FAULHARBER 4490 200W	
0x3a	SpdScI (L)	30517 (0x7735)	31250 (0x7a12)	62499 (0xf423)	62499 (0xf423)	TargetSpd 값이 rpm 이 되도록 설정함.
0x3b	SpdScI (M)					
0x3c	TargetSpd(L)	0	0	0	0	[rpm]
0x3d	TargetSpd(M)					
0x3e	SpdAccRate	16 (0x10)	64 (0x40)	32 (0x20)	32 (0x20)	작으면 위치제어 문제 생김. 1=83/SpdLpRate [Krpm/s]
0x3f	SpdDecRate	16 (0x10)	64 (0x40)	32 (0x20)	32 (0x20)	
0x42	FaultCtrl	2, 1	2, 1	2, 1	2, 1	
0x44	ModScI (L)	601 (0x0259)	601 (0x0259)	601 (0x0259)	601 (0x0259)	
0x45	ModScI (M)					
0x4e	PwmDataSel (L)	0	0	0	0	
0x4f	PwmDataSel (M)					
0x72	HaIICBA001(L)	85 (0x0055)	333 (0x014d)	333 (0x014d)	667 (0x029b)	
0x73	HaIICBA001(M)					
0x74	HaIICBA010(L)	427 (0x01ab)	1667 (0x0683)	1667 (0x0683)	1333 (0x0535)	
0x75	HaIICBA010(M)					
0x76	HaIICBA011(L)	0 (0x0000)	0 (0x0000)	0 (0x0000)	1000 (0x03e8)	
0x77	HaIICBA011(M)					
0x78	HaIICBA100(L)	256 (0x0100)	1000 (0x03e8)	1000 (0x03e8)	0 (0x0000)	
0x79	HaIICBA100(M)					
0x7a	HaIICBA101(L)	171 (0x00ab)	667 (0x029b)	667 (0x029b)	333 (0x014d)	
0x7b	HaIICBA101(M)					
0x7c	HaIICBA110(L)	341 (0x0155)	1333 (0x0535)	1333 (0x0535)	1667 (0x0683)	
0x7d	HaIICBA110(M)					

IRMCK201 BLDC motor 구동 IC의 parameter (for writing)

address	register name	EC 45 50W flat 모터	EC-powermax 30 100W			
0x50	SysConfig	0	0	0	0	
0x52	VdSfwdCtrl(L)					
0x53	VqSfwdCtrl(L)					
0x54	VdSfwdCtrl(M) VqSfwdCtrl(M)					
0x55	ElecAngW(L)					
0x56	ElecAngW(M)					
0x58	EncCnt32bW_LW(L)					
0x59	EncCnt32bW_LW(M)					
0x5a	EncCnt32bW_HW(L)					
0x5b	EncCnt32bW_HW(M)					
0x5c	EeCtrl					
0x5d	EeAddrW					
0x5e	EeDataW					

명령어	구분	설명
;PWiiii1,iiii2; (%) ;PW?; (%)	operation	모터의 PWM duty 출력을 강제로 설정함. 1st integer : 모터1의 PWM duty값 [-PWM_PERIOD ~ PWM_PERIOD] (0) 2nd integer : 모터2의 PWM duty값 [-PWM_PERIOD ~ PWM_PERIOD] (0)
	return value	PWiiii1,iiii2;
	comment	PWM 주기값은 -PWM_PERIOD에서 PWM_PERIOD 범위에서만 유효한 값이며 PWM_PERIOD는 SG 명령어로 설정 가능함. 단 SG 명령어를 사용할 때는 특별한 sequence로 명령어를 내려보내야 하며 ST 명령어의 comment를 참조할 것. SM 명령에 의한 작동 모드가 0000일 때만 PW 명령으로 CMPR1/2_reg의 값에 적용되는데, CMPR1/2_reg의 값들을 직접 읽으려면 QR7417,01; 명령을 사용하면 됨. PW 명령어로 함부로 값을 설정하면 모터나 모터제어기에 손상이 갈수 있으므로 사용에 주의를 요함. iiii1,iiii2는 부호가 있는 십진수임. SM 명령에 의한 작동모드설정이 0000 일때만 사용이 가능한 명령임.

명령어	구분	설명
	operation	<p>모터의 PWM duty 출력을 강제로 설정함.</p> <p>1st - 6th integer : motor의 PWM duty값 [-PWM_PERIOD ~ PWM_PERIOD] (0)</p> <p>VER4/8의 경우 : 1st/2nd 는 모터1/2의 PWM duty 설정값</p> <p>VER5의 경우 : 1st/2nd 는 모터1의 PWM duty 설정값, 3rd/4th 는 모터2의 PWM duty 설정값</p> <p>VER6/7의 경우 : 1st/2nd/3rd 는 모터1의 PWM duty 설정값, 4th/5th/6th 는 모터2의 PWM duty 설정값</p>
	return value	PWiiii1,iiii2; or PWiiii1,iiii2,iiii3,iiii4; or PWiiii1,iiii2,iiii3,iiii4,iiii5,iiii6;
;PWiiii1,iiii2; (VER4/8) (+#^) ;PWiiii1,iiii2,iiii3,iiii4; (VER5) (+#^) ;PWiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (VER6/7) (+#^) ;PW?; (+#^) 	comment	<p>PWM 주기값은 -PWM_PERIOD에서 PWM_PERIOD 범위에서만 유효한 값임.</p> <p>SM 명령에 의한 작동 모드가 0000일 때만 PW 명령으로 CMPR1/2/3/4/5/6_reg의 값에 적용되는데, PW 명령어로 항부로 값을 설정하면 모터나 모터제어기에 손상이 갈수 있으므로 사용에 주의를 요함.</p> <p>CMPR1/2/3/4/5/6_reg의 값들을 직접 읽으려면 QR7417,02; 및 QR7517,02; 명령을 사용하면 됨. (2811의 경우) TMS320F28334/5의 경우는 주소가 달라서 6809/6849/6889/68C9/6909/6949 주소의 값을 읽으면 됨. iiii1,iiii2,iiii3,iiii4,iiii5,iiii6는 부호가 있는 십진수임.</p> <p>2011년 7월 25일 이전의 S/W 버전에서는 bipolar 구동 형태의 firmware인 경우, [-PWM_PERIOD/4 ~ PWM_PERIOD/4] 범위의 값만을 사용하여야하며, 이 범위를 벗어나는 값을 사용하는 경우에는 출력이 잠깐 켜지다가 off 됨.</p> <p>2011년 8월 28일 이전의 S/W 버전에서는 작동 모드가 0000일 때 PW 명령으로 PWM 출력을 조절하는 경우, 출력단자의 쇼트나 과전류보호를 하지 않음. 2011년 8월 28일 S/W 버전부터는 이러한 보호로 인하여 예민하게 보호될 수 있으므로 Sv 명령과 SI 명령을 사용하여 적절한 보호 threshold를 설정하여야함.</p> <p>2013년 1월 10일 이전의 S/W 버전에서는 SM 명령에 의한 작동모드설정이 0000 일때만 사용이 가능하며, 2013년 1월 10일 S/W 버전부터는 모터1또는 모터2의 작동모드가 00으로 설정되면, 00으로 설정된 모터에 한하여 PW 명령의 사용이 가능함.</p> <p>2013년 2월 07일 S/W 이전 버전에서는 BLDC의 경우 작동모드 0일때 PWM 출력이 50% duty가 아닌 상태가 기본값이었음.</p> <p>H/W 버전에 따라서는 100%의 duty를 설정하면 모터구동출력이 잠깐 ON 되었다가 차단되는 경우가 있으므로, 0~±90% 범위의 duty를 사용하기를 권장함.</p>

명령어	구분	설명
;PA#####; (+#^%) ;PA?; (+#^%) 2014년 3월 21일 S/W 버전부터 다음 명령어도 사용 가능함. ;PC#####; (+#^%) ;PC?; (+#^%)	operation	모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함. 1st long integer : 모터1의 위치명령값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [5000000-32000/60000 ~ 5000000+32000/60000] or 전류명령값 [5000000-1023/4095 ~ 5000000+1023/4095] 2nd long integer : 모터2의 위치명령값 [position2_minus_limit ~ position2_plus_limit] or 속도명령값 [5000000-32000/60000 ~ 5000000+32000/60000] or 전류명령값 [5000000-1023/4095 ~ 5000000+1023/4095] PC 명령인 경우는 위치제어모드인 경우에 한하여 QE5AA5; QEAA55; QE55AA; 명령어로 설정된 위치를 5,000,000 값에 해당하는 위치로 처리하며, 2014년 3월 21일 S/W 버전부터 사용가능함.
	return value	PA#####; or PC#####;
	comment	position mode 로 설정된 모터는 위치명령값으로 설정되며, 모터1의 경우는 position1_minus_limit에서 position1_plus_limit 범위에서만 유효하고, 모터2의 경우는 position2_minus_limit에서 position2_plus_limit 범위에서만 유효함. 이 경우는 위치명령값의 위치로 이동하되 SS####1,####2; 명령어로 설정된 속도로 이동함. 위치명령값은 5000000 이 기본값으로서 원점위치를 의미하며 항상 양수이어야함. speed mode 로 설정된 모터는 위치명령값이 아니고 속도명령값으로 설정되며, 설정값을 5000000-32000/60000 ~ 5000000+32000/60000 범위에서 주면 속도명령값은 -32000/60000 ~ 32000/60000 범위에서 설정됨. current mode 로 설정된 모터는 설정전류값을 설정하게 되며, 설정값을 5000000-1023/4095 ~ 5000000+1023/4095 범위에서 주면 전류명령값은 -1023/4095 ~ 1023/4095 범위에서 설정됨. 이 명령어는 십진수를 long integer로, long integer를 십진수로 바꾸는 32bit 나누기 연산을 여러번 수행하므로, 명령어 처리에 약 300microsec(return을 안받으면 150micro)의 시간이 더 걸리기 때문에, PA 대신 PB 나 Pa 명령어를 사용하는 것이 더 효과적임. 모터2개를 구동하는 모터제어기의 경우로서 위치제어모드(SM 명령어로 설정)로 설정된 경우에, 모터2개가 연동하여 linear interpolation에 의한 2축 직선운동을 하도록 설정(SX 명령어 첫번째 파라미터의 bit1으로 설정)하였다면, PA 명령어로 두 모터의 위치명령값을 설정함으로써 linear interpolation에 의한 2축 직선운동이 개시됨.

명령어	구분	설명
	comment	<p>위치명령값의 분해능</p> <ul style="list-style-type: none"> : 엔코더를 feedback 제어용으로 사용하는 DC/BLDC의 경우는 엔코더의 4체배후 분해능과 같음 : digital Hall 센서만을 사용하는 BLDC의 경우는 모터 1회전당 16384 펄스 (pole수에 무관) : 2상/3상 analog Hall 센서만을 사용하는 BLDC는 모터 1회전당 16384 펄스 (pole수에 무관) : 스텝핑모터로서 lead_anglex_control_flg가 0인 경우 모터 1회전당 16384 펄스 (pole수에 무관) (내부에서는 16384 * pole pair수가 한바퀴에 해당함) : 스텝핑모터로서 lead_anglex_control_flg가 1인 경우 엔코더의 4체배후 분해능과 같음 (내부에서는 16384 * pole pair수가 한바퀴에 해당함) <p>2013년 1월 19일 S/W 버전부터 00번 작동모드에서는 PA 명령을 사용하여도 ';'을 리턴하도록 하였음. 2014년 9월 13일 이전 S/W 버전에서는 lead angle제어를 사용하는 경우 위치명령값설정에 문제가 있었음. 이 문제는 PA/PB/PC/Pa/Pb 등 위치값을 설정하는 명령어에서 모두 발생함.</p>

명령어	구분	설명
;PAa#####; (+#^) ;PAa?; (+#^) 2014년 3월 21일 S/W 버전부터 다 음 명령어도 사용 가능함.	operation	모터제어기의 작동모드에 따라서 모터1의 위치/속도/전류의 명령값을 설정함. 1st long integer : 모터1의 위치명령값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [5000000-32000/60000 ~ 5000000+32000/60000] or 전류명령값 [5000000-1023/4095 ~ 5000000+1023/4095] PC 명령인 경우는 위치제어모드인 경우에 한하여 QE5AA5; QEAA55; QE55AA; 명령어로 설정된 위치를 5,000,000 값에 해당하는 위치로 처리하며, 2014년 3월 21일 S/W 버전부터 사용가능함.
	return value	PAa#####; or PCa#####;
	comment	comment는 PA#####; 또는 PC#####; 명령어의 모터1에 대한 부분과 같음. PAa는 2013년 7월 31일 S/W 버전부터 사용가능한 명령어임. PCa는 2014년 3월 21일 S/W 버전부터 사용가능한 명령어임.
;PAb#####; (+#^) ;PAb?; (+#^) 2014년 3월 21일 S/W 버전부터 다 음 명령어도 사용 가능함.	operation	모터제어기의 작동모드에 따라서 모터2의 위치/속도/전류의 명령값을 설정함. 1st long integer : 모터2의 위치명령값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [5000000-32000/60000 ~ 5000000+32000/60000] or 전류명령값 [5000000-1023/4095 ~ 5000000+1023/4095] PC 명령인 경우는 위치제어모드인 경우에 한하여 QE5AA5; QEAA55; QE55AA; 명령어로 설정된 위치를 5,000,000 값에 해당하는 위치로 처리하며, 2014년 3월 21일 S/W 버전부터 사용가능함.
	return value	PAb#####; or PCb#####;
	comment	comment는 PA#####; 또는 PC#####; 명령어의 모터2에 대한 부분과 같음. PAb는 2013년 7월 31일 S/W 버전부터 사용가능한 명령어임. PCb는 2014년 3월 21일 S/W 버전부터 사용가능한 명령어임.
;PCa#####; (+#^) ;PCa?; (+#^) ;PCb#####; (+#^) ;PCb?; (+#^)		

명령어	구분	설명
;PBHLHLHLHL,HLHLHLHL; (+#^%) ;PB?: (+#^%)	operation	<p>모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함.</p> <p>1st long integer : 모터1의 위치명령값 [00000000 - FFFFFFFF] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF]</p> <p>2nd long integer : 모터2의 위치명령값 [00000000 - FFFFFFFF] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF]</p>
	return value	PBHLHLHLHL,HLHLHLHL;
	comment	<p>전송데이터 형식과 설정값 범위를 빼고는 PA명령과 같음.</p> <p>position mode 로 설정된 모터는 위치명령값으로 설정되며 위치명령값의 위치로 이동하되 SSddd1,ddd2; 명령어로 설정된 속도로 이동함.</p> <p>speed mode 로 설정된 모터는 위치명령값이 아니고 속도명령값으로 설정되며, 설정값을 004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60 범위에서 주면 속도명령값은 -32000/60000 ~ 32000/60000 범위에서 설정됨.</p> <p>current mode 로 설정된 모터는 설정전류값을 설정하게 되며, 설정값을 004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF 범위에서 주면 전류명령값은 -1023/4095 ~ 1023/4095 범위에서 설정됨.</p> <p>모터2개를 구동하는 모터제어기의 경우로서 위치제어모드(SM 명령어로 설정)로 설정된 경우에, 모터2개가 연동하여 linear interpolation에 의한 2축 직선운동을 하도록 설정(SX 명령어 첫번째 파라미터의 bit1으로 설정)하였다면, PB 명령어로 두 모터의 위치명령값을 설정함으로써 linear interpolation에 의한 2축 직선운동이 개시됨. (2013년 1월 10일 S/W 버전부터 사용가능함.)</p> <p>2013년 1월 19일 S/W 버전부터 00번 작동모드에서는 PB 명령을 사용하여도 ';'을 리턴하도록 하였음.</p>

명령어	구분	설명
;PBaHLHLHLHL; (+#^)	operation	모터제어기의 작동모드에 따라서 모터1의 위치/속도/전류의 명령값을 설정함. 1st long integer : 모터1의 위치명령값 [00000000 - FFFFFFFF] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF]
	return value	PBaIIII1;
	comment	comment는 PAIIII1,IIII2; 명령어의 모터1에 대한 부분과 같음. 2013년 7월 31일 S/W 버전부터 사용가능한 명령어임.
;PBbHLHLHLHL; (+#^)	operation	모터제어기의 작동모드에 따라서 모터2의 위치/속도/전류의 명령값을 설정함. 1st long integer : 모터2의 위치명령값 [00000000 - FFFFFFFF] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF]
	return value	PBbIIII1;
	comment	comment는 PBIIII1,IIII2; 명령어의 모터2에 대한 부분과 같음. 2013년 7월 31일 S/W 버전부터 사용가능한 명령어임.

명령어	구분	설명
;PaHBMBLBHBMBLBC S; (+#%) or ;PbHBMBLBHBMBLBC S; (+#^) (2013년 1월 13일 S/W 버전부터 적용) (2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우) ;PaHBMBLBCS; (+#^) ;PbHBMBLBCS; (+#^) 	operation	모터제어기의 작동모드에 따라서 위치/속도/전류의 명령값을 설정함. 1st 24-bit long word : 모터1의 위치명령값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF] 2nd 24-bit long word : 모터2의 위치명령값 [position2_minus_limit ~ position2_plus_limit] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF] Pa 와 ; 를 빼고는 binary 값임. 2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우는 2nd 24-bit long word 값은 생략됨.
	return value	PaHBMBLBHBMBLBCS; or PbHBMBLBHBMBLBCS; HBMBLB 는 binary 형태의 24bit word를 가리키며 상위,중간,하위바이트를 가리킴. CS 는 앞선 6byte의 합의 inverse를 binary 한 바이트로 표현한 것임. Pa 명령인 경우 return 값은 현재의 위치값임. Pb 명령은 2013년 1월 13일 S/W 버전부터 사용 가능하며 작동모드에 따라서 현재위치/현재속도/현재전류 값이 리턴됨. checksum이 일치되지 않으면 수신값을 무시하고 return 값의 마지막은 :임. 이 경우 내부에서 수신에러 카운트 Pa_receive_error_cnt를 하나 증가시킴. 2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우 PaHBMBLBCS; PbHBMBLBCS; 로서 모터1의 값만을 리턴함. TMS320f28334를 사용한 모터제어기에서 2013년 10월 19일부터 2014년 3월 22일 이전 까지의 S/W 버전은 모터1의 값을 설정하는데 있어서 오류가 있음.

명령어	구분	설명
	comment	<p>전송데이터 형식을 빼고는 PA명령과 같음.</p> <p>position mode 로 설정된 모터는 위치명령값으로 설정되며 위치명령값의 위치로 이동하되 SSdddd1, dddd2; 명령어로 설정된 속도로 이동함. 위치명령값은 4C4B40 이 기본값으로서 원점위치를 의미하며 항상 양수이어야함.</p> <p>speed mode 로 설정된 모터는 위치명령값이 아니고 속도명령값으로 설정되며, 설정값을 004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60 범위에서 주면 속도명령값은 -32000/60000 ~ 32000/60000 범위에서 설정됨.</p> <p>current mode 로 설정된 모터는 설정전류값을 설정하게 되며, 설정값을 004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF 범위에서 주면 전류명령값은 -1023/4095 ~ 1023/4095 범위에서 설정됨.</p> <p>모터2개를 구동하는 모터제어기의 경우로서 위치제어모드(SM 명령어로 설정)로 설정된 경우에, 모터2개가 연동하여 linear interpolation에 의한 2축 직선운동을 하도록 설정(SX 명령어 첫번째 파라미터의 bit1으로 설정)하였다면, Pa/Pb 명령으로 두 모터의 위치명령값을 설정함으로써 linear interpolation에 의한 2축 직선운동이 개시됨. (2013년 1월 10일 S/W 버전부터 사용가능함.)</p>

명령어	구분	설명
;PI#####; (+#^) ;PI?; (+#^)	operation	위치 명령값의 증분을 설정하고 이동을 명령함. 1st integer : 모터1의 위치명령값의 증분 [-2147483648 ~ 2147483647] 2nd integer : 모터2의 위치명령값의 증분 [-2147483648 ~ 2147483647]
	return value	PI#####; 1st long integer : 모터1의 위치명령값 2nd long integer : 모터2의 위치명령값
	comment	position mode 로 설정된 경우에 한하며, 모터1의 경우는 position1_minus_limit에서 position1_plus_limit 범위에서만 유효하고, 모터2의 경우는 position2_minus_limit에서 position2_plus_limit 범위에서만 유효함. 이 경우는 증분의 위치명령값 위치로 이동하되 SS####1,####2; 명령어로 설정된 속도로 이동함. 리턴값은 증분값이 아니고 증감된 현재의 위치명령값임. 이 명령어는 십진수를 long integer로, long integer를 십진수로 바꾸는 32bit 나누기 연산을 여러번 수행하므로, 명령어 처리에 약 300microsec(return을 안받으면 150micro)의 시간이 걸림. 2013년 1월 19일 S/W 버전부터 00번 작동모드에서는 PI 명령을 사용하여도 ';'을 리턴하도록 하였음.

명령어	구분	설명
;P1 1; (+#^) ;P1?; (+#^)	operation	모터1에 대하여 위치 명령값의 증분을 설정하고 이동을 명령함. 1st integer : 모터1의 위치명령값의 증분 [-2147483648 ~ 2147483647]
	return value	P1 1; 1st long integer : 모터1의 위치명령값
	comment	position mode 로 설정된 경우에 한하며, position1_minus_limit에서 position1_plus_limit 범위에서만 유효함. 증분의 위치명령값 위치로 이동하되 SSdddd1,dddd2; 명령어로 설정된 속도로 이동함. 리턴값은 증분값이 아니고 증감된 현재의 위치명령값임. 이 명령어는 십진수를 long integer로, long integer를 십진수로 바꾸는 32bit 나누기 연산을 여러번 수행하므로, 명령어 처리에 약 150microsec의 시간이 더 걸리기 때문에 주의를 요함. 2013년 1월 19일 S/W 버전부터 00번 작동모드에서는 P1 명령을 사용하여도 ';'을 리턴하도록 하였음.
;P2 1; (+#^) ;P2?; (+#^)	operation	모터2에 대하여 위치 명령값의 증분을 설정하고 이동을 명령함. 1st integer : 모터2의 위치명령값의 증분 [-2147483648 ~ 2147483647]
	return value	P2 1; 1st long integer : 모터2의 위치명령값
	comment	position mode 로 설정된 경우에 한하며, position2_minus_limit에서 position2_plus_limit 범위에서만 유효함. 증분의 위치명령값 위치로 이동하되 SSdddd1,dddd2; 명령어로 설정된 속도로 이동함. 리턴값은 증분값이 아니고 증감된 현재의 위치명령값임. 이 명령어는 십진수를 long integer로, long integer를 십진수로 바꾸는 32bit 나누기 연산을 여러번 수행하므로, 명령어 처리에 약 150microsec의 시간이 더 걸리기 때문에 주의를 요함. 2013년 1월 19일 S/W 버전부터 00번 작동모드에서는 P2 명령을 사용하여도 ';'을 리턴하도록 하였음.

명령어	구분	설명
;PEHLHL; (+#^%) ;PEA55A; (+#^) ;PE?; (+#^%)	operation	선택된 보드의 모터 출력을 ON 시킴. 1st word : board address or A55A
	return value	PEHLHL;
	comment	<p>보드의 주소값이 일치하지 않으면 명령이 무시됨. 주소값이 A55A인 경우에는 2012년 2월 8일 S/W 버전 부터 사용가능하며 주소값이 일치하지 않더라도 모터 출력을 ON 시킴.</p> <p>fault flag가 set 되어 있으면 명령이 무시됨. (단 위치 리밋을 벗어난 fault와 통신 fault에 의해서는 PE 명령이 무시되지 않음)</p> <p>PE? 에 대한 리턴값은, 모터 출력이 ON되어 있을 때는 board address이며, OFF 되어 있을 때는 0000 임. 72V 이상의 고전압 H/W 버전에서는 PE 명령어로 서보를 enable 시킨후에 SM 명령어로 작동모드를 선택해야, 0.55초후에 전류측정 offset을 정상적으로 설정하게 되며, 이후에 PAL나 SV같은 구동명령을 사용 수 있음.</p> <p>precharge 릴레이가 있는 72V 이상의 고전압 H/W 버전에서는 2012년 6월 30일 S/W 버전부터 precharge 릴레이를 먼저 ON 한 다음 100ms 후에 PWM출력이 enable 됨.</p> <p>PEA55A; 명령은 2012년 2월 8일 S/W 버전부터 사용가능함.</p> <p>IR21771S을 사용하는 경우 PE 명령을 수행할 때 PWM출력이 on 되는데, on 직후 fault로 간주되어 다시 off 되는 문제가 있어서 수정 하였음. (2013년~2014년 사이의 수정과정에서 발생한 문제임) 2014년 11월 04일 S/W 버전부터 수정되었음.</p>

명령어	구분	설명
;PRHLHL; (+#^%) ;PRA55A; (+#^)	operation	선택된 보드의 fault flag를 clear 시킴. 1st word : board address or A55A
	return value	PRHLHL;
	comment	<p>보드의 주소값이 일치하지 않으면 명령이 무시됨. 주소값이 A55A인 경우에는 2012년 2월 8일 S/W 버전 부터 사용가능하며 주소값이 일치하지 않더라도 모터 출력을 ON 시킴.</p> <p>모터 작동의 중지를 일으킨 fault flag와 관련 timer/counter를 clear 시킴. IRMCK201을 사용할 때는 IRMCK201 내부의 fault flag도 clear 시켜서 다시 모터를 기동시킬 수 있게 하며, 홈동작중이거나 홈동작 중 에러가 발생한 경우에 홈동작을 중지하고 새로운 홈동작을 개시할 수 있는 상태로 설정함.</p> <p>fault flag중에서 motorx_stall_flg, motorx_stall_K_flg 는 PPR; 명령어에 의하여 clear 되며, currentx_over_flg, currentx_over_K_flg는 과전류상태가 벗어나야만 clear되며, 위치 리밋을 벗어난 fault는 리밋 안쪽으로 들어와야만 clear되며, 통신 fault는 ECR; 명령에 의하여 clear됨.</p> <p>보드의 주소값이 일치하지 않으면 명령이 무시됨. PRA55A; 명령은 2012년 2월 8일 S/W 버전부터 사용가능함.</p>

명령어	구분	설명
;PDHLHL; (+#^%) ;PDA55A; (+#^) ;PD?; (+#^)	operation	선택된 보드의 모터 출력을 OFF 시킴. 1st word : board address or A55A
	return value	PDHLHL;
	comment	<p>보드의 주소값이 일치하지 않으면 명령이 무시됨. 주소값이 A55A인 경우에는 2012년 2월 8일 S/W 버전 부터 사용가능하며 주소값이 일치하지 않더라도 모터 출력을 ON 시킴. 2011년 11월 20일 S/W 버전부터는 모터가 disable 될때 작동모드도 0으로 강제설정되게 하였음. PD? 에 대한 리턴값은, 모터 출력이 OFF되어 있을 때는 board address이며, ON 되어 있을 때는 0000 임. precharge 릴레이가 있는 72V 이상의 고전압 H/W 버전에서는 2012년 6월 30일 S/W 버전부터 PWM출력을 먼저 disable 한 다음 200ms 후에 precharge 릴레이를 끄. PDA55A; 명령은 2012년 2월 8일 S/W 버전부터 사용가능함.</p> <p>만약 에너지를 저장할수 있는 제어대상을 구동하던 중에, 제어대상이 큰 에너지를 저장하고 있는 상태에서 PD 명령을 수행하면, 저장된 에너지가 기전력을 발생시켜서 모터 공급전압의 강한 상승을 유발하는 경우에 모터제어기가 손상될수 있으므로, PD 명령의 사용에 주의를 기하여야함. 이 경우에는 PD 명령을 사용하지 않더라도 구동중의 fault 상황에 의하여 모터출력이 off 될수 있는데, 충분히 큰 용량의 brake저항을 사용하여 에너지를 열로 환원할 수 있게 구성하여야함.</p>

명령어	구분	설명
;PMddd1,ddd2,ddd3; (+#^%) ;PM?; (+#^%)	operation	현재위치로부터 dddd1점을 거쳐 dddd2점까지 dddd3의 속도로 이동시킴. ddd1 : 첫번째 경유지 point number [0 - 150] ddd2 : 최종 도착지 point number [1 - 151] ddd3 : point 당 움직임시간 [1 - 10000] (ms)
	return value	PMddd1,ddd2,ddd3; ddd1 : 경유중 point number [0 - 150] ddd2 : 최종 도착지 point number [1 - 151] ddd3 : point 당 움직임시간 [1 - 10000] (ms) ddd2 > ddd1 이어야함
	comment	(point간 움직임시간 * POINT간 거리)이 32768*65535를 넘으면 비정상적 작동 유발. SM 명령에 의한 작동모드가 위치제어모드인 01/02/03 중에 하나이어야하고, SeHLHL; 명령으로 해당모터의 motorx_start_to_move_trajectory_flg가 세트 가능토록 설정된 모터 (전원 투입시 기본값은 세트 가능토록 하는 것임) 만 명령이 유효함. PM 명령의 수행전에 SP 명령어로 이동좌표점이 설정되어 있거나, EROB00,0500; 명령어로 이전에 EEPROM에 저장된 이동좌표점을 불러내어야 함. 2012년 12월 03일 S/W 버전부터 좌표점의 개수가 192개에서 168개로 줄었음. 2013년 4월 14일 S/W 버전부터 좌표점의 개수가 168개에서 152개로 줄었음. RJM_VER5/6 에서는 의미없는 명령임. 자석가동자의 경우 사용하면 안되면 명령임.
;PME; (+#^%)	operation	PM 에 의한 동작을 멈춤.
	return value	PME;
	comment	PM 의 명령이 일단 실행되면 최종 도착지 point까지 이동을 하는데 도중에 멈추려면 PME; 를 실행하여야 함. RJM_VER5/6 에서는 의미없는 명령임. 자석가동자의 경우 사용하면 안되면 명령임.

명령어	구분	설명
;PP#####; (+#^%) ;PQ#####; (+#^%)	operation	<p>모터제어기의 작동모드에 따라서 10msec/30msec 동안에 가야할 위치/속도/전류의 명령값을 설정하고 모터가 움직이도록 함. PP인 경우는 10msec, PQ인 경우는 30msec임.</p> <p>1st long integer : 모터1의 위치명령값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [5000000-32000/60000 ~ 5000000+32000/60000] or 전류명령값 [5000000-1023/4095 ~ 5000000+1023/4095]</p> <p>2nd long integer : 모터2의 위치명령값 [position2_minus_limit ~ position2_plus_limit] or 속도명령값 [5000000-32000/60000 ~ 5000000+32000/60000] or 전류명령값 [5000000-1023/4095 ~ 5000000+1023/4095]</p>
	return value	PP#####; or PQ#####;
	comment	<p>SeHLHL; 명령으로 해당모터의 motorx_start_to_move_to_position_flg가 세트 가능토록 설정된 모터 (전원 투입시 기본값은 세트 가능토록 하는 것임) 만 명령이 유효하며, 10msec/30msec 이전에 다음 명령이 입력되어야 연속으로 모터가 움직임.</p> <p>10msec/30msec 명령갱신 주기는 St 명령으로 변경이 가능함.</p> <p>position mode 로 설정된 모터는 위치명령값으로 설정되며, 모터1의 경우는 position1_minus_limit에서 position1_plus_limit 범위에서만 유효하고, 모터2의 경우는 position2_minus_limit에서 position2_plus_limit 범위에서만 유효함. 이 경우는 위치명령값의 위치로 이동하되 명령갱신주기 만큼의 시간 후에 도달하도록 위치명령값이 1ms 단위로 쪼개져서 증분됨. 위치명령값은 5000000 이 기본값으로서 원점위치를 의미하며 항상 양수이어야함.</p> <p>speed mode 로 설정된 모터는 위치명령값이 아니고 속도명령값으로 설정되며, 설정값을 5000000-32000/60000 ~ 5000000+32000/60000 범위에서 주면 속도명령값은 -32000/60000 ~ 32000/60000 범위에서 설정됨.</p>

명령어	구분	설명
	comment	<p>current mode 로 설정된 모터는 설정전류값을 설정하게 되며, 설정값을 5000000-1023/4095 ~ 5000000+1023/4095 범위에서 주면 전류명령값은 -1023/4095 ~ 1023/4095 범위에서 설정됨.</p> <p>이 명령어는 십진수를 long integer로, long integer를 십진수로 바꾸는 32bit 나누기 연산을 여러번 수행하므로, 명령어 처리에 약 300microsec(return을 안받으면 150microsec)의 시간이 더 걸리기 때문에, PP/PQ 대신 Pp/Pq 명령어를 사용하는 것이 더 효과적임.</p> <p> x는 최대 7자리의 10진수임.</p> <p>2014년 4월 9일 이전 S/W 버전에서는 PP/PQ 명령의 증분이 -32768 ~ +32767 범위를 넘어가면 안됨.</p> <p>RJM_VER5/6 에서는 의미없는 명령임.</p>

명령어	구분	설명
;PpHBMBLBHMBLBC S; (+#^%) ;PqHBMBLBHMBLBC S; (+#^%) (2013년 11월 22 일 이후 S/W 버전 으로서 SX 명령어 세번째 파라미터 의 LSB가 0인 경 우) ;PpHBMBLBCS; (+#^) ;PqHBMBLBCS; (+#^)	operation	모터제어기의 작동모드에 따라서 10msec/30msec 동안에 가야할 위치/속도/전류의 명령값을 설정하고 모터가 움직이도록 함. Pp인 경우는 10msec, Pq인 경우는 30msec임. 1st 24-bit long word : 모터1의 목표위치값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF] 2nd 24-bit long word : 모터2의 목표위치값 [position2_minus_limit ~ position2_plus_limit] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF] Pp/Pq 와 ; 를 빼고는 binary 값임. 2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우는 2nd 24-bit long word 값은 생략됨.
	return value	2006년 10월 7일 이전 S/W 버전 : PpHBMBLBHMBLBCS; PqHBMBLBHMBLBCS; HBMBLB 는 binary 형태의 24bit word를 가리키며 상위,중간,하위바이트를 가리킴. CS 는 앞선 6byte의 합의 inverse를 binary 한 바이트로 표현한 것임. return 값은 현재의 위치값임. checksum이 일치되지 않으면 수신값을 무시하고 return 값의 마지막은 :임. 이 경우 내부에서 수신에러 카운트 Pp_receive_error_cnt를 하나 증가시킴. 2006년 10월 7일 S/W 버전부터 : PpLBHBLBLBHBLBHBLBCS; 또는 PpHBLBHBLBHBLBHBLBCS;(2007년 9월 19일 S/W 버전부터 사용가능) PqLBHBLBLBHBLBHBLBCS; 또는 PqHBLBHBLBHBLBHBLBCS;(2007년 9월 19일 S/W 버전부터 사용가능) LB 는 binary 형태의 8bit 바이트 한개를 가리킴. HBLB 는 binary 형태의 16bit word의 상위바이트 한개, 하위바이트 한개를 가리킴. CS 는 앞선 8byte 의 합의 inverse를 binary 한 바이트로 표현한 것임.

명령어	구분	설명
	return value	<p>Pp/Pq 명령의 리턴값 변수주소를 설정하는 명령으로 SrHLHL,HLHL;을 사용하였을 때는, Pp/Pq 명령에 대한 return 값으로 첫번째 바이트는 LIMIT_SWITCH_IN, 두번째 바이트는 high byte of fault_status, 세번째 바이트는 low byte of fault_status, 네번째 바이트는 low byte of fault1_status, 다섯번째 바이트는 high byte of *PpPq_return_value_ADDRESS1, 여섯번째 바이트는 low byte of *PpPq_return_value_ADDRESS1, 일곱번째 바이트는 high byte of *PpPq_return_value_ADDRESS2, 여덟번째 바이트는 low byte of *PpPq_return_value_ADDRESS2 임.</p> <p>2007년 9월 19일 S/W 버전부터 Pp/Pq 명령의 리턴값 변수주소를 설정하는 명령으로 SrHLHL,HLHL,HLHL,HLHL;을 사용하였을 때는, Pp/Pq 명령에 대한 return 값으로 1st/2nd/3rd/4th word에 의하여 지정된 4개 주소내의 값이 됨.</p> <p>2013년 8월 08일 S/W 버전부터 Pp/Pq 명령을 수신할때 CS 값이 ((앞선 6byte 의 합 의 inverse) & 0x00ff) 와 같을 때는 앞페이지에 기술된 형식으로 리턴값이 만들어져서 host로 전송하는데, CS 값이 ((앞선 6byte 의 합 의 inverse에 1을 더한 값) & 0x00ff) 와 같을 때는 리턴값이 다음과 같음. 첫번째 바이트는 motor_status_word 의 하위바이트, 두번째 바이트는 highest byte of position1, 세번째 바이트는 middle byte of position1, 네번째 바이트는 lowest byte of position1, 다섯번째 바이트는 highest byte of position2, 여섯번째 바이트는 middle byte of position2, 일곱번째 바이트는 lowest byte of position2, 여덟번째 바이트는 LIMIT_SWITCH_IN</p> <p>CS 값이 ((앞선 6byte 의 합 의 inverse에 2를 더한 값) & 0x00ff) 와 같을 때는 리턴값이 다음과 같음. 첫번째 바이트는 motor_status_word 의 하위바이트, 두번째 바이트는 high byte of LS1C 포트의 전압값, 세번째 바이트는 low byte of LS1C 포트의 전압값, 네번째 바이트는 high byte of LS2C 포트의 전압값, 다섯번째 바이트는 low byte of LS2C 포트의 전압값, 여섯번째 바이트는 high byte of fault_status, 일곱번째 바이트는 low byte of fault_status, 여덟번째 바이트는 LIMIT_SWITCH_IN</p> <p>LIMIT_SWITCH_IN 은 Q5; 명령어를 참조하고, fault_status 는 Q2; 명령어를 참조하며, motor_status_word 의 하위바이트는 Q7 명령어의 리턴값 하위바이트와 같음.</p>

명령어	구분	설명
	return value	<p>2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우 CS 값이 ((앞선 3byte 의 합 의 inverse에 1을 더한 값) & 0x00ff) 와 같을 때는 리턴값이 다음과 같음. 첫번째 바이트는 motor_status_word 의 하위바이트, 두번째 바이트는 highest byte of position1, 세번째 바이트는 middle byte of position1, 네번째 바이트는 lowest byte of position1, 다섯번째 바이트는 high byte of current1_AD, 여섯번째 바이트는 low byte of current1_AD, 일곱번째 바이트는 low byte of operation_status, 여덟번째 바이트는 low byte of LIMIT_SWITCH_IN 로서 모터1의 값만을 리턴함.</p> <p>LIMIT_SWITCH_IN 은 QS; 명령어를 참조하고, fault_status 는 Q2; 명령어를 참조하며, motor_status_word 의 하위바이트는 Q7 명령어의 리턴값 하위바이트와 같고, operation_status는 Q1; 명령어를 참조.</p>
	comment	<p>checksum이 일치되지 않으면 수신값을 무시하고 return 값의 마지막은 :임. 이 경우 내부에서 수신에러 카운트 Pp_receive_error_cnt를 하나 증가시킴. 전송데이터 형식을 빼고는 PP/PQ 명령과 같음.</p> <p>SeHLHL; 명령으로 해당모터의 motorx_start_to_move_to_position_flg가 세트 가능토록 설정된 모터 (전원 투입시 기본값은 세트 가능토록 하는 것임) 만 명령이 유효하며, 10msec/30msec 이전에 다음 명령이 입력되어야 연속으로 모터가 움직임. 10msec/30msec 명령갱신 주기는 St 명령으로 변경이 가능함.</p> <p>position mode 로 설정된 모터는 위치명령값으로 설정되며, 위치명령값의 위치로 이동하되 명령갱신주기 만큼의 시간 후에 도달하도록 위치명령값이 1ms 단위로 쪼개져서 증분됨. 위치명령값은 4C4B40 이 기본값으로서 원점위치를 의미하며 항상 양수이어야함.</p> <p>speed mode 로 설정된 모터는 위치명령값이 아니고 속도명령값으로 설정되며, 설정값을 004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60 범위에서 주면 속도명령값은 -32000/60000 ~ 32000/60000 범위에서 설정됨.</p> <p>current mode 로 설정된 모터는 설정전류값을 설정하게 되며, 설정값을 004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF 범위에서 주면 전류명령값은 -1023/4095 ~ 1023/4095 범위에서 설정됨. 뒤에 더 있음.</p>

명령어	구분	설명
	comment	<p>2014년 3월 22일 이전 S/W 버전에서는 리턴값중 positionx, current1_AD, fault_status 값이 불연속적으로 튀는 현상이 있을수 있음.</p> <p>2014년 4월 9일 이전 S/W 버전에서는 Pp/Pq 명령의 증분이 -32768 ~ +32767 범위를 넘어가면 안됨.</p> <p>RJM_VER5/6 에서는 의미없는 명령임.</p>
;PPE; (+#^%)	operation	PP, PQ, Pp, Pq, p 에 의한 동작을 멈춤.
	return value	PPE; or PQE;
;PQE; (+#^%)	comment	<p>PP, PQ, Pp, Pq, p 의 명령이 일단 실행되면 다음의 PP, PQ, Pp, Pq, p 를 계속 기다리는 상태가 된다. 이 상태를 빠져나오려면 PPE; 를 실행하여야함.</p> <p>RJM_VER5/6 에서는 의미없는 명령임.</p>
;PPR; (+#^%)	operation	모터가 stall된 채로 정지된 상태를 해지함.
	return value	PPR; or PQR;
;PQR; (+#^%)	comment	<p>motor1_stall_flg, motor2_stall_flg, motor1_stall_K_flg, motor2_stall_K_flg를 clear 시킴. 다시 모터를 기동시킬 수 있게 됨.</p> <p>RJM_VER5/6 에서는 의미없는 명령임.</p>

명령어	구분	설명
<p>;pHBMBLBHMBLBCS ; (+#^%) (RS485의 경우)</p> <p>;pHBMBLBHMBLBCS (+#^%) (CAN의 경우 8 byte, 1 packet)</p> <p>(2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우로서)</p> <p>;pHBMBLBCS; (+#^%) (RS485의 경우)</p> <p>;pHBMBLBCS (+#^) (CAN의 경우 5 byte, 1 packet)</p>	<p>operation</p> <p>return value</p>	<p>모터제어기의 작동모드에 따라서 10msec 동안에 가야할 위치/속도/전류의 명령값을 설정하고 모터가 움직이도록 함.</p> <p>1st 24-bit long word : 모터1의 목표위치값 [position1_minus_limit ~ position1_plus_limit] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF]</p> <p>2nd 24-bit long word : 모터2의 목표위치값 [position2_minus_limit ~ position2_plus_limit] or 속도명령값 [004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60] or 전류명령값 [004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF]</p> <p>p 와 ; 를 빼고는 binary 값임. HBMBLB 는 binary 형태의 24bit word를 가리키며 상위,중간,하위바이트를 가리킴. CS 는 앞선 6byte의 합의 inverse를 binary 한 바이트로 표현한 것임.</p> <p>2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우는 2nd 24-bit long word 값은 생략됨.</p> <p>RS485의 경우 : pLBHBLBLBHBLBHBLBCS; 또는 pHBLBHBLBHBLBHBLBCS;(2007년 9월 19일 S/W 버전부터 사용가능)</p> <p>CAN의 경우 : LBHBLBLBHBLBHBLB 또는 HBLBHBLBHBLBHBLB(2007년 9월 19일 S/W 버전부터 사용가능) LB 는 binary 형태의 8bit 바이트 한개를 가리킴. HBLB 는 binary 형태의 16bit word의 상위바이트 한개, 하위바이트 한개를 가리킴. CS 는 앞선 8byte의 합의 inverse를 binary 한 바이트로 표현한 것임.</p> <p>p 명령의 리턴값 변수주소를 설정하는 명령어로 SrHLHL,HLHL;을 사용하였을 때는, p 명령에 대한 return 값으로 첫번째 바이트는 LIMIT_SWITCH_IN, 두번째 바이트는 high byte of fault_status, 세번째 바이트는 low byte of fault_status, 네번째 바이트는 low byte of fault1_status, 다섯번째 바이트는 high byte of *PpPq_return_value_ADDRESS1, 여섯번째 바이트는 low byte of *PpPq_return_value_ADDRESS1, 일곱번째 바이트는 high byte of *PpPq_return_value_ADDRESS2, 여덟번째 바이트는 low byte of *PpPq_return_value_ADDRESS2 임.</p>

명령어	구분	설명
	return value	<p>2007년 9월 19일 S/W 버전부터 p 명령의 리턴값 변수주소를 설정하는 명령으로 SrHLHL,HLHL,HLHL,HLHL;을 사용하였을 때는, p 명령에 대한 return 값으로 1st/2nd/3rd/4th word에 의하여 지정된 4개 주소내의 값이 됨.</p> <p>2013년 8월 08일 S/W 버전부터 p 또는 q 명령을 수신할때 CS 값이 ((앞선 6byte 의 합 의 inverse) & 0x00ff) 와 같을 때는 앞페이지에 기술된 형식으로 리턴값이 만들어져서 host로 전송하는데, CS 값이 ((앞선 6byte 의 합 의 inverse에 1을 더한 값) & 0x00ff) 와 같을 때는 리턴값이 다음과 같음. 첫번째 바이트는 motor_status_word 의 하위바이트, 두번째 바이트는 highest byte of position1, 세번째 바이트는 middle byte of position1, 네번째 바이트는 lowest byte of position1, 다섯번째 바이트는 highest byte of position2, 여섯번째 바이트는 middle byte of position2, 일곱번째 바이트는 lowest byte of position2, 여덟번째 바이트는 LIMIT_SWITCH_IN</p> <p>CS 값이 ((앞선 6byte 의 합 의 inverse에 2를 더한 값) & 0x00ff) 와 같을 때는 리턴값이 다음과 같음. 첫번째 바이트는 motor_status_word 의 하위바이트, 두번째 바이트는 high byte of LS1C 포트의 전압값, 세번째 바이트는 low byte of LS1C 포트의 전압값, 네번째 바이트는 high byte of LS2C 포트의 전압값, 다섯번째 바이트는 low byte of LS2C 포트의 전압값, 여섯번째 바이트는 high byte of fault_status, 일곱번째 바이트는 low byte of fault_status, 여덟번째 바이트는 LIMIT_SWITCH_IN</p> <p>2013년 11월 22일 이후 S/W 버전으로서 SX 명령어 세번째 파라미터의 LSB가 0인 경우 CS 값이 ((앞선 3byte 의 합 의 inverse에 1을 더한 값) & 0x00ff) 와 같을 때는 리턴값이 다음과 같음. 첫번째 바이트는 motor_status_word 의 하위바이트, 두번째 바이트는 highest byte of position1, 세번째 바이트는 middle byte of position1, 네번째 바이트는 lowest byte of position1, 다섯번째 바이트는 high byte of current1_AD, 여섯번째 바이트는 low byte of current1_AD, 일곱번째 바이트는 low byte of operation_status, 여덟번째 바이트는 low byte of LIMIT_SWITCH_IN</p> <p>로서 모터1의 값만을 리턴함.</p> <p>LIMIT_SWITCH_IN 은 QS; 명령어를 참조하고, fault_status 는 Q2; 명령어를 참조하며, motor_status_word 의 하위바이트는 Q7 명령어의 리턴값 하위바이트와 같고, operation_status는 Q1; 명령어를 참조.</p>

명령어	구분	설명
	comment	<p>checksum이 일치되지 않으면 수신값을 무시하고 return 값의 마지막은 :임. 이 경우 내부에서 수신에러 카운트 Pp_receive_error_cnt를 하나 증가시킴. 전송데이터 형식을 빼고는 PP/PQ 명령과 같음.</p> <p>SeHLHL; 명령으로 해당모터의 motorx_start_to_move_to_position_flg가 세트 가능토록 설정된 모터 (전원 투입시 기본값은 세트 가능토록 하는 것임) 만 명령이 유효하며, 10msec 이전에 다음 명령이 입력되어야 연속으로 모터가 움직임. 10msec 명령갱신 주기는 St 명령으로 변경이 가능함.</p> <p>position mode 로 설정된 모터는 위치명령값으로 설정되며, 위치명령값의 위치로 이동하되 명령갱신주기 만큼의 시간 후에 도달하도록 위치명령값이 1ms 단위로 쪼개져서 증분됨. 위치명령값은 4C4B40 이 기본값으로서 원점위치를 의미하며 항상 양수이어야함.</p> <p>speed mode 로 설정된 모터는 위치명령값이 아니고 속도명령값으로 설정되며, 설정값을 004C4B40-7D00/EA60 ~ 004C4B40+7D00/EA60 범위에서 주면 속도명령값은 -32000/60000 ~ 32000/60000 범위에서 설정됨.</p> <p>current mode 로 설정된 모터는 설정전류값을 설정하게 되며, 설정값을 004C4B40-3FF/FFF ~ 004C4B40+3FF/FFF 범위에서 주면 전류명령값은 -1023/4095 ~ 1023/4095 범위에서 설정됨.</p> <p>2014년 4월 9일 이전 S/W 버전에서는 p 명령의 증분이 -32768 ~ +32767 범위를 넘어가면 안됨.</p> <p>RJM_VER5/6 에서는 의미없는 명령임.</p>

명령어	구분	설명
;PsA55A; (+#^%) ;PtA55A; (+#^)	operation	모터1,2를 모두 현재의 위치에서 감속을 실시한 후에 정지함.
	return value	PsA55A;
	comment	PM/PP/PQ/Pp/Pq/p에 의한 위치제어 도중이었다면 이를 중지하고 현재의 위치로부터 감속하여 정지함. 속도제어 중이었다면 감속을 실시한 후 정지하며 전류제어중이었다면 전류값을 0으로 강제설정함. 단 감속률은 PsA55A;의 경우 Ss 명령어나 Sa 명령어로 설정된 감속도에 따르며, PtA55A;의 경우는 So 명령어로 설정된 감속도에 따름. 2012년 1월 19일 S/W 버전부터 사용가능한 명령임. PtA55A; 명령어의 경우 2013년 7월 31일 이전 S/W 버전에서는 So 명령어로 설정된 감속도가 작을때 정상적이 작동이 안됨.
;PsA5A5; (+#^%) ;PtA5A5; (+#^)	operation	모터1을 현재의 위치에서 감속을 실시한 후에 정지함.
	return value	PsA5A5;
	comment	PM/PP/PQ/Pp/Pq/p에 의한 위치제어 도중이었다면 이를 중지하고 현재의 위치로부터 감속하여 정지함. 속도제어 중이었다면 감속을 실시한 후 정지하며 전류제어중이었다면 전류값으로 0으로 강제설정함. 단 감속률은 PsA55A;의 경우 Ss 명령어나 Sa 명령어로 설정된 감속도에 따르며, PtA55A;의 경우는 So 명령어로 설정된 감속도에 따름. 2012년 1월 19일 S/W 버전부터 사용가능한 명령임. PtA5A5; 명령어의 경우 2013년 7월 31일 이전 S/W 버전에서는 So 명령어로 설정된 감속도가 작을때 정상적이 작동이 안됨.
;Ps5A5A; (+#^%) ;Pt5A5A; (+#^)	operation	모터2를 현재의 위치에서 감속을 실시한 후에 정지함.
	return value	Ps5A5A;
	comment	PM/PP/PQ/Pp/Pq/p에 의한 위치제어 도중이었다면 이를 중지하고 현재의 위치로부터 감속하여 정지함. 속도제어 중이었다면 감속을 실시한 후 정지하며 전류제어중이었다면 전류값으로 0으로 강제설정함. 단 감속률은 PsA55A;의 경우 Ss 명령어나 Sa 명령어로 설정된 감속도에 따르며, PtA55A;의 경우는 So 명령어로 설정된 감속도에 따름. 2012년 1월 19일 S/W 버전부터 사용가능한 명령임. Pt5A5A; 명령어의 경우 2013년 7월 31일 이전 S/W 버전에서는 So 명령어로 설정된 감속도가 작을때 정상적이 작동이 안됨.

명령어	구분	설명
;PeA55A; (+#^)	operation	digital Hall sensor만을 사용하는 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1,2 모두를 02번 위치 제어모드로 바꾸고, 현재의 위치에서 감속후 정지함.
	return value	PeA55A;
	comment	감속률은 Ss 명령어로 설정된 감속도에 따름. digital Hall 센서만을 사용하는 BLDC 모터제어에서만 유효함.
;PeA5A5; (+#^)	operation	digital Hall sensor만을 사용하는 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1을 02번 위치제어모드로 바꾸고, 현재의 위치에서 감속후 정지함.
	return value	PeA5A5;
	comment	감속률은 Ss 명령어로 설정된 감속도에 따름. digital Hall 센서만을 사용하는 BLDC 모터제어에서만 유효함.
;Pe5A5A; (+#^)	operation	digital Hall sensor만을 사용하는 BLDC 모터에 있어서, 05번 속도제어 상태의 모터2를 02번 위치제어모드로 바꾸고, 현재의 위치에서 감속후 정지함.
	return value	Pe5A5A;
	comment	감속률은 Ss 명령어로 설정된 감속도에 따름. digital Hall 센서만을 사용하는 BLDC 모터제어에서만 유효함.

명령어	구분	설명
;PvA55A; (+#^)	operation	digital Hall sensor만을 사용하는 wafer 회전구동용 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1,2 모두를 감속한 후에, 낮은 속도가 되면 02번 위치제어모드로 바꾸고, 감속후 정지함.
	return value	PvA55A;
	comment	감속률은 Sa, Ss 명령어로 설정된 감속도에 따름. digital Hall 센서만을 사용하는 WAFER 구동용 BLDC 모터제어에서만 유효함.
;PvA5A5; (+#^)	operation	digital Hall sensor만을 사용하는 wafer 회전구동용 BLDC 모터에 있어서, 05번 속도제어 상태의 모터1을 감속한 후에, 낮은 속도가 되면 02번 위치제어모드로 바꾸고, 감속후 정지함.
	return value	PvA5A5;
	comment	감속률은 Sa, Ss 명령어로 설정된 감속도에 따름. digital Hall 센서만을 사용하는 WAFER 구동용 BLDC 모터제어에서만 유효함.
;Pv5A5A; (+#^)	operation	digital Hall sensor만을 사용하는 wafer 회전구동용 BLDC 모터에 있어서, 05번 속도제어 상태의 모터2를 감속한 후에, 낮은 속도가 되면 02번 위치제어모드로 바꾸고, 감속후 정지함.
	return value	Pv5A5A;
	comment	감속률은 Sa, Ss 명령어로 설정된 감속도에 따름. digital Hall 센서만을 사용하는 WAFER 구동용 BLDC 모터제어에서만 유효함.
;Pv 1, 2; (+#^) ;Pv?; (+#^)	operation	digital Hall sensor만을 사용하는 BLDC 모터가 아닌 경우에, 속도제어 모드에서 속도명령값을 설정하되, 리턴값으로 현재 위치의 원점기준 변위값을 host로 전송함. 1st long integer : 모터1 속도명령값 [-32000/60000 ~ 32000/60000] (0 rpm) 2nd long integer : 모터2 속도명령값 [-32000/60000 ~ 32000/60000] (0 rpm)
	return value	Pv 1, 2; 1st long integer : 모터1의 현재 위치의 원점기준 변위값 (4채배 pulse, 32bit) 2nd long integer : 모터2의 현재 위치의 원점기준 변위값 (4채배 pulse, 32bit)
	comment	속도제어 모드에서만 사용이 가능함. 리턴값은 엔코더 원점기준의 변위를 의미하므로 0 근방의 값인데 대하여, QP/PA 등의 명령이 5,000,000 근방의 값을 사용하는 것과 다르므로 주의를 해야 함. digital Hall sensor만을 사용하는 BLDC 모터의 경우는 해당사항 없음.

명령어	구분	설명
;Pc!!!!1,!!!!2; (+#^) ;Pc?: (+#^)	operation	digital Hall sensor만을 사용하는 BLDC 모터가 아닌 경우에, 전류제어 모드에서 전류명령값을 설정하되, 리턴값으로 현재 속도값을 host로 전송함. 1st integer : 모터1 전류명령값 [-4095 ~ 4095] (bit) 2nd integer : 모터2 전류명령값 [-4095 ~ 4095] (bit)
	return value	Pc!!!!1,!!!!2; 1st integer: 모터1의 현재 속도값 (rpm) 2nd integer : 모터2의 현재 속도값 (rpm)
	comment	전류제어 모드에서만 사용이 가능함. digital Hall sensor만을 사용하는 BLDC 모터의 경우는 해당사항 없음.
;PV!!!!1,!!!!2,d ddd3; (+#^) ;Pvr!!!!1,!!!!2, dddd3; (+#^) ;PV?: (+#^)	operation	속도제어 모드에서 속도명령값을 설정하되 설정값의 checksum을 사용함. 1st integer : 모터1 속도명령값 [-60000 ~ 60000] (0 rpm) 2nd integer : 모터2 속도명령값 [-60000 ~ 60000] (0 rpm) 3rd integer : '!!!!1,!!!!2' 의 각 글자code값을 더한 값 (예를들어 PV0,0,140; 에서 140은 '0'의 code 값 48과 ','의 code 값 44, '0'의 code 값 48을 모두 더한 값임)
	return value	PV!!!!1,!!!!2;
	comment	PV 명령의 경우에 스테핑모터가 아닐때 속도오차가 크면 속도명령값을 재설정함. 따라서 같은 속도 명령을 주기적으로 주는 경우에도 속도변화가 발생할수 있음. 2010년 7월 19일 S/W 버전부터 속도오차가 클때 적분오차를 0으로 초기화하는 작동을 내부에서 수행함. 2013년 3월 27일 S/W 버전부터 속도오차가 클때 적분오차를 0으로 초기화하는 속도오차의 크기를 WA 명령어로 설정할수있으며, SX 명령어 두 번째파라미터의 bit20으로 적분오차를 0으로 초기화하는 작동을 enable/disable 할수 있음. Pvr 명령의 경우는 속도오차가 큰 경우에 속도명령값을 재설정하고 적분오차를 0으로 초기화하는 작동을 하지 않음. Pvr 명령의 경우는 2011년 9월 24일 S/W 버전부터 사용가능한 명령임. F2811/F28334에서는 long integer를 입력할수있으며 속도명령값의 설정가능 범위가 [-60000 ~ 60000] (rpm)임.

명령어	구분	설명
;PH0; (+#^) ;PHo; (+#^)	operation	홈위치 또는 가상홈위치로의 이동명령을 수행함.
	return value	PH0; 또는 PHo;
	comment	직선보간운동 중일때는 실행할 수 없음. PH0; 는 PA5000000,5000000; 명령의 수행과 결과가 같음. PHo; 는 PC5000000,5000000; 명령의 수행과 결과가 같음. 2011년 11월 24일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;GHHL,HL,iiii1,i ii2; (+#^%) ;GhHL,HL,IIII1,I II2; (+#^) ;GH?; (+#^) ;Gh?; (+#^) 	operation	2개의 리미트센서와 1개의 홈센서를 사용한 홈 동작을 개시함. (Gh는 0.01도 단위의 offset 변위이며 Gh는 4 체배 엔코더 펄스단위의 offset 변위임) 1st byte : 모터번호 [0이면 선택없음, 1이면 모터1, 2이면 모터2, 3이면 모터1/2] 2nd byte : home 동작 개시 방향 [0이면 음의 방향, 1이면 양의 방향이며 하위 4bit는 모터1, 상위 4bit는 모터2] iiii1 : 모터1이 홈센서를 찾은 후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도] iiii2 : 모터2가 홈센서를 찾은 후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도] IIII1 : 모터1이 홈센서를 찾은 후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준] IIII2 : 모터2가 홈센서를 찾은 후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준]
	return value	GHHL,HL,IIII1,IIII2; or GhHL,HL,IIII1,IIII2;
	comment	GH 와 Gh 명령에서 offset변위의 기본단위는 각각 0.01도(SG 명령어에 의한 gear ratio 바퀴수 만큼 모터 축이 회전할 때가 360도이므로, 예로 gear ratio가 100, offset 변위가 180 이라면 180*100/36000=0.5 바 퀴의 모터축 회전에 해당함) 와 4체배 엔코더 펄스임. SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 home 동작 개시 방향의 리미트센서를 찾은 후 (+방향이면 LSxA 입력이 0, -방향이면 LSxC 입력이 0이 될 때 리미트를 찾은 것으로 봄) 반대방향으로 움직 여서 홈센서를 찾은 후 (LSxB 입력이 0이 될 때 홈을 찾은 것으로 봄) 홈센서가 꺼질 때까지 움직여서 목 표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함. 만약 리 미트센서보다 홈센서를 먼저 발견하게 되면 바로 반대방향으로 홈센서가 꺼질때까지 움직여서 목표위치, 엔 코더값등을 초기값으로 재설정하고 offset 변위만큼 이동함. 이때 홈센서가 꺼지기 전에 home 동작 개 시 반대 방향의 리미트센서가 ON 되어 있으면 홈센서와 리미트센서가 공용인 것으로 판단하여 이동방향을 바꾼 후에 홈센서가 꺼질때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 offset 변위만큼 이동 함. SI 명령어로 설정된 위치 범위내에서만 리미트센서와 홈센서를 찾으며 만약 범위를 벗어나면 즉시 명령의 수 행을 중지하고 정지함. SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함. PM/PP/PQ/Pp/Pq 명령에 의한 연속궤적동작을 안하고 있을 때로서, 홈동작이 한번도 안되었을 때 또는 완료 되었을 때 또는 오류가 발생했을 때 : 즉 홈동작을 실행중이지 않을 때에 적용가능함. 홈동작 진행/성공여부는 Q1; 명령으로 알수있으며 PRHLHL; 명령으로 홈동작을 중지시킬 수 있음.

명령어	구분	설명
	comment	<p>스텝모터에서 2014년 9월 20일 이전의 S/W 버전에서는 흡동작과 offset 설정에 문제가 있었음. 절대각센서를 사용하는 경우, 흡센서만을 사용하는 BLDC의 경우는 명령의 사용이 불가함.</p> <p>2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 흡동작후 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 흡동작 완료후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 흡동작 완료 상태의 위치를 5,000,000 원점으로 하는 위치명령 사용이 가능함. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.</p> <p>2014년 9월 21일 S/W 버전부터는 원점을 검출하면 PsA55A; 등의 명령으로 정지를 하고 정지후 100ms를 기다렸다가 원점을 설정한 다음 offset 만큼 이동함.</p> <p>2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.</p>

명령어	구분	설명
;GIHL,HL,iiii1,iiii2; (+#^)	operation	<p>2개의 리미트센서 만을 사용한 홈 동작을 개시함. (Gi는 0.01도 단위의 offset 변위이며 Gi는 4체배 엔코더 펄스단위의 offset 변위임)</p> <p>1st byte : 모터번호 [0이면 선택없음, 1이면 모터1, 2이면 모터2, 3이면 모터1/2]</p> <p>2nd byte : home 동작 개시 방향 [0이면 음의 방향, 1이면 양의 방향이며 하위 4bit는 모터1, 상위 4bit는 모터2]</p> <p>iiii1 : 모터1이 리미트센서를 찾은후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도]</p> <p>iiii2 : 모터2가 리미트센서를 찾은후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도]</p> <p>IIII1 : 모터1이 리미트센서를 찾은후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준]</p> <p>IIII2 : 모터2가 리미트센서를 찾은후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준]</p>
	return value	GIHL,HL,IIII1,IIII2; or GiHL,HL,IIII1,IIII2;
;GiHL,HL,IIII1,IIII2; (+#^) ;GI?; (+#^) ;Gi?; (+#^)	comment	<p>Gi 와 Gi 명령에서 offset변위의 기본단위는 각각 0.01도(SG 명령어에 의한 gear ratio 바퀴수 만큼 모터 축이 회전할 때가 360도이므로, 예로 gear ratio가 100, offset 변위가 180 이라면 $180 \times 100 / 36000 = 0.5$ 바퀴의 모터축 회전에 해당함) 와 4체배 엔코더 펄스임.</p> <p>SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 home 동작 개시 방향의 리미트센서를 찾은 후 (+방향이면 LSxA 입력이 0, -방향이면 LSxB 입력이 0이 될 때 리미트를 찾은 것으로 봄) 리미트센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함.</p> <p>SI 명령어로 설정된 위치 범위내에서만 리미트센서를 찾으며 만약 범위를 벗어나면 즉시 명령의 수행을 중지하고 정지함.</p> <p>SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함.</p> <p>PM/PP/PQ/Pp/Pq 명령에 의한 연속궤적동작을 안하고 있을 때로서, 홈동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 홈동작을 실행중이지 않을 때에 적용가능함.</p> <p>홈동작 진행/성공여부는 Q1; 명령으로 알수있으며 PRHLHL; 명령으로 홈동작을 중지시킬 수 있음.</p>

명령어	구분	설명
	comment	<p>스텝모터에서 2014년 9월 20일 이전의 S/W 버전에서는 흡동작과 offset 설정에 문제가 있었음. 절대각센서를 사용하는 경우, 흡센서만을 사용하는 BLDC의 경우는 명령의 사용이 불가함.</p> <p>2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 흡동작후 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 흡동작 완료후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 흡동작 완료 상태의 위치를 5,000,000 원점으로 하는 위치명령 사용이 가능함. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.</p> <p>2014년 9월 21일 S/W 버전부터는 원점을 검출하면 PsA55A; 등의 명령으로 정지를 하고 정지후 100ms를 기다렸다가 원점을 설정한 다음 offset 만큼 이동함.</p> <p>2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.</p>

명령어	구분	설명
;GJHL,HL,iiii1,i iii2; (+#^) ;GjHL,HL,IIII1,I III2; (+#^) ;GJ?; (+#^) ;Gj?; (+#^)	operation	<p>한개의 홀센서 만을 사용한 홈 동작을 개시함. (GJ는 0.01도 단위의 offset 변위이며 Gj는 4체배 엔코더 펄스단위의 offset 변위임)</p> <p>1st byte : 모터번호 [0이면 선택없음, 1이면 모터1, 2이면 모터2, 3이면 모터1/2]</p> <p>2nd byte : home 동작 개시 방향 [0이면 음의 방향, 1이면 양의 방향이며 하위 4bit는 모터1, 상위 4bit는 모터2]</p> <p>iiii1 : 모터1이 홀센서의 경계를 찾은 후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도]</p> <p>iiii2 : 모터2가 홀센서의 경계를 찾은 후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도]</p> <p>IIII1 : 모터1이 홀센서의 경계를 찾은 후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준]</p> <p>IIII2 : 모터2가 홀센서의 경계를 찾은 후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준]</p>
	return value	GJHL,HL,IIII1,IIII2; or GjHL,HL,IIII1,IIII2;
	comment	<p>GJ 와 Gj 명령에서 offset변위의 기본단위는 각각 0.01도(SG 명령어에 의한 gear ratio 바퀴수 만큼 모터 축이 회전할 때가 360도이므로, 예로 gear ratio가 100, offset 변위가 180 이라면 $180 \times 100 / 36000 = 0.5$ 바퀴의 모터축 회전에 해당함) 와 4체배 엔코더 펄스임.</p> <p>SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 홀센서를 찾은 후 (LSxB 입력이 0이 될 때 홈을 찾은 것으로 봄) 홀센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함. 만약 처음부터 홀센서가 0N이면 홀센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함.</p> <p>SI 명령어로 설정된 위치 범위내에서만 홀센서를 찾으며 만약 범위를 벗어나면 즉시 명령의 수행을 중지하고 정지함.</p> <p>SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함.</p> <p>PM/PP/PQ/Pp/Pq 명령에 의한 연속궤적동작을 안하고 있을 때로서, 홈동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 홈동작을 실행중이지 않을 때에 적용가능함.</p> <p>홈동작 진행/성공여부는 Q1; 명령으로 알수있으며 PRHLHL; 명령으로 홈동작을 중지시킬 수 있음.</p>

명령어	구분	설명
	comment	<p>스텝모터에서 2014년 9월 20일 이전의 S/W 버전에서는 흡동작과 offset 설정에 문제가 있었음. 절대각센서를 사용하는 경우, 흡센서만을 사용하는 BLDC의 경우는 명령의 사용이 불가함.</p> <p>2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 흡동작후 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 흡동작 완료후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 흡동작 완료 상태의 위치를 5,000,000 원점으로 하는 위치명령 사용이 가능함. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.</p> <p>2014년 9월 21일 S/W 버전부터는 원점을 검출하면 PsA55A; 등의 명령으로 정지를 하고 정지후 100ms를 기다렸다가 원점을 설정한 다음 offset 만큼 이동함.</p> <p>2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.</p>

명령어	구분	설명
;GkHL,HL,iiii1,i iii2; (+#^) ;GkHL,HL,IIII1,I IIII2; (+#^) ;Gk?; (+#^) ;Gk?; (+#^)	operation	한개의 홈센서 만을 사용한 홈 동작을 개시하되 홈센서가 ON 되어 있는 상태에서만 개시함. (Gk는 0.01도 단위의 offset 변위이며 Gk는 4체배 엔코더 펄스단위의 offset 변위임) 1st byte : 모터번호 [0이면 선택없음, 1이면 모터1, 2이면 모터2, 3이면 모터1/2] 2nd byte : home 동작 개시 방향 [0이면 음의 방향, 1이면 양의 방향이며 하위 4bit는 모터1, 상위 4bit는 모터2] iiii1 : 모터1이 홈센서의 경계를 찾은 후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도] iiii2 : 모터2가 홈센서의 경계를 찾은 후 offset 변위 [-32000 ~ +32000, 감속기 출력기준의 0.01도 단위 각도] IIII1 : 모터1이 홈센서의 경계를 찾은 후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준] IIII2 : 모터2가 홈센서의 경계를 찾은 후 offset 변위 [모터에 부착된 엔코더의 4체배 펄스기준]
	return value	GkHL,HL,IIII1,IIII2; or GkHL,HL,IIII1,IIII2;
	comment	GK 와 Gk 명령에서 offset변위의 기본단위는 각각 0.01도(SG 명령어에 의한 gear ratio 바퀴수 만큼 모터 축이 회전할 때가 360도이므로, 예로 gear ratio가 100, offset 변위가 180 이라면 $180 \times 100 / 36000 = 0.5$ 바퀴의 모터축 회전에 해당함) 와 4체배 엔코더 펄스임. 만약 처음부터 홈센서가 OFF(LSxB 입력이 1 일 때)이면 명령을 무시하며, 홈센서가 처음부터 ON이면 SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 홈센서가 꺼질 때까지 움직여서 목표위치, 엔코더 값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함. SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함. PM/PP/PQ/Pp/Pq 명령에 의한 연속계적동작을 안하고 있을 때로서, 홈동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 홈동작을 실행중이지 않을 때에 적용가능함. 홈동작 진행/성공여부는 Q1; 명령으로 알수있으며 PRHLHL; 명령으로 홈동작을 중지시킬 수 있음.

명령어	구분	설명
	comment	<p>스텝모터에서 2014년 9월 20일 이전의 S/W 버전에서는 흡동작과 offset 설정에 문제가 있었음. 절대각센서를 사용하는 경우, 흡센서만을 사용하는 BLDC의 경우는 명령의 사용이 불가함.</p> <p>2014년 9월 21일 S/W 버전부터는 스텝모터이거나 스테핑모터가 아닌 경우로서 엔코더를 사용하지 않는 경우는 흡동작후 위치의 초기값이 5,000,000과 조금 다른 값이 될 수있는데, 흡동작 완료후 PA?; 의 리턴값을 읽어서 이를 원점으로 감안하여 위치명령을 내리는 방법을 쓸수있고, 또는 PC 명령을 사용하면 흡동작 완료 상태의 위치를 5,000,000 원점으로 하는 위치명령 사용이 가능함. 흡동작 후의 위치초기값은 PA?; 명령어로 확인할수 있으며, 위치초기값 - 5,000,000의 값은 QEi; 명령어로 확인할 수 있음.</p> <p>2014년 9월 21일 S/W 버전부터는 원점을 검출하면 PsA55A; 등의 명령으로 정지를 하고 정지후 100ms를 기다렸다가 원점을 설정한 다음 offset 만큼 이동함.</p> <p>2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.</p>

명령어	구분	설명
;GTdddd1,dddd2; (+#^) ;GT?: (+#^)	operation	리미트센서와 홀센서의 ON/OFF를 검출하는 경계전압을 설정함. 1st word : OFF를 검출하는 경계전압 [100 - 280] (225 0.01V) 2nd wprd : ON를 검출하는 경계전압 [20 - 200] (75 0.01V)
	return value	GTdddd1,dddd2;
	comment	1st word 보다 높은 전압일 경우 리미트센서나 홀센서는 OFF인 것으로 처리함. 2nd word 보다 낮은 전압일 경우 리미트센서나 홀센서는 ON인 것으로 처리함. 비서로봇의 경우는 기본값이 2.63V와 1.31V로 설정되어 있음. ⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Gtdddd1,dddd2; (+#^) ;Gt?: (+#^)	operation	RJM_VER7의 Hall 센서입력의 ON/OFF를 검출하는 기준전압과 히스테리시스 진폭을 설정함. 1st word : ON/OFF를 검출하는 기준전압 [25 - 400] (25/200 0.01V) 2nd wprd : 히스테리시스 진폭 [5 - 200] (5/10 0.01V)
	return value	Gtdddd1,dddd2;
	comment	1st word + 2nd word 보다 높은 전압일 경우 Hall 센서는 OFF인 것으로 처리함. 1st word - 2nd word 보다 낮은 전압일 경우 Hall 센서는 ON인 것으로 처리함. open collector 방식의 Hall sensor일때는 신호전압의 상승 속도가 매우 느리기 때문에, ON/OFF를 검출하는 기준전압과 히스테리시스 진폭을 작은 값으로 설정하여야함. F28334 H/W 버전에서는 Gt 명령어로 설정된 값을 저장할 때 F2811에 비하여 16배 더 큰 값으로 저장되나 겉으로 들어나는 차이는 없음. 나노모션의 analog Hall sensor를 사용하는 경우에는 225,10의 값을 사용할 것. ⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

흡동작 - 2개의 리밋센서와 1개의 흡센서를 쓰는 경우

☞ 흡동작을 하려면

- 1) SMHLHL; 명령어로 01/02/03의 위치제어 모드를 선택함.
 상위/하위바이트가 01이면 모터2/모터1이 위치제어모드(inner loop 없고 PWM사용, DC 모터전용)
 상위/하위바이트가 02이면 모터2/모터1이 위치제어모드(inner loop 전류제어, DC/BLDC)
 상위/하위바이트가 03이면 모터2/모터1이 위치제어모드(inner loop 속도제어, BLDC 모터전용)
- 2) PEHLHL; 명령어로 해당 모터제어보드의 출력을 enable 시킴. (HLHL은 보드번호를 써야함)
- 3) SSddd1,ddd2; 명령어로 흡이동속도를 정의함.
- 4) GHHL,HL,iiii1,iiii2; 또는 GhHL,HL,iiii1,iiii2; 명령어로 흡동작을 개시함.
- 5) 흡동작의 상태는 Q1; 명령어로 확인할 수 있음.
- 6) 흡동작도중에 중지하고자하면 PRHLHL; 명령어로서 모든 에러 플래그를 clear 함.

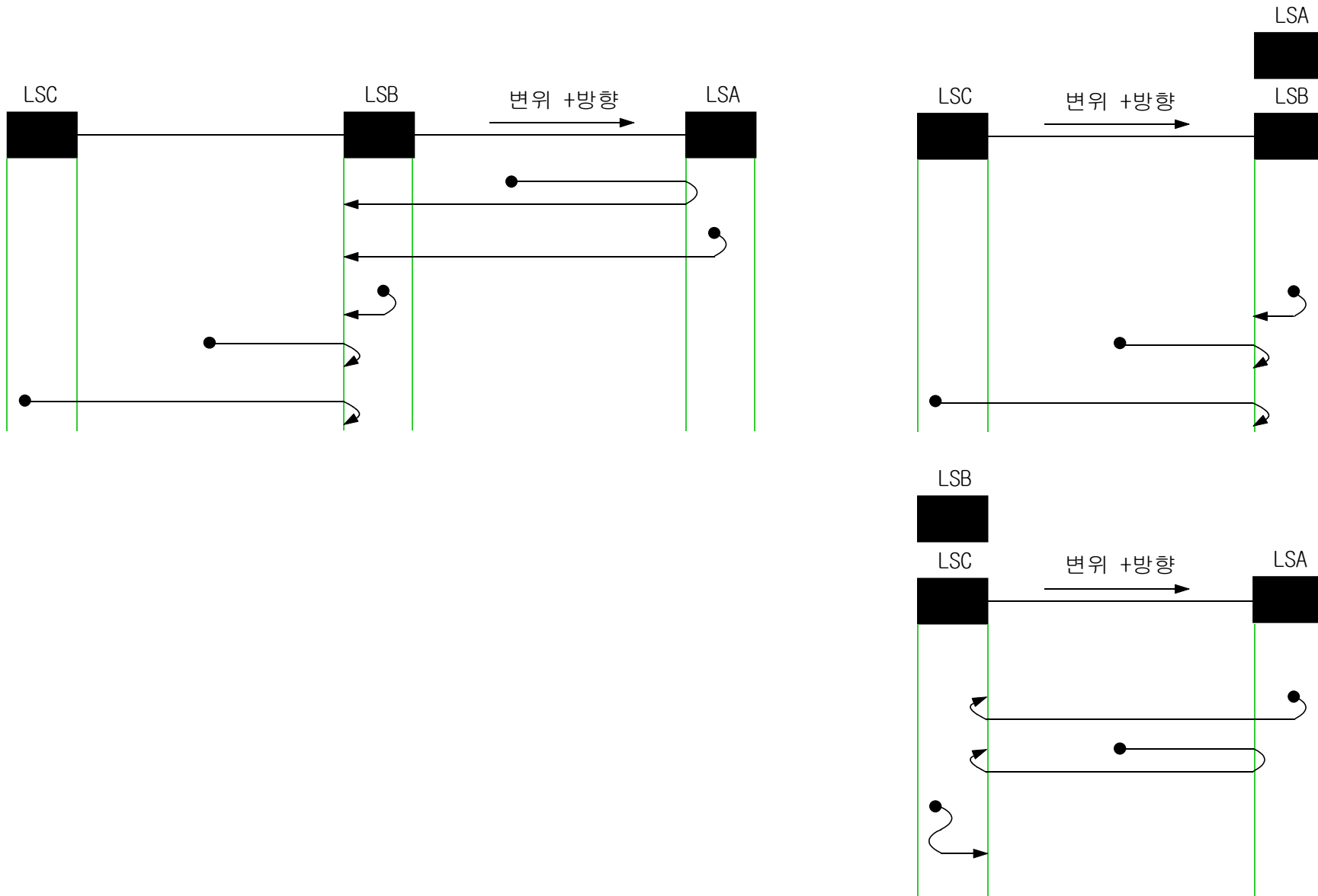
GH 와 Gh 명령에서 offset변위의 기본단위는 각각 0.01도와 4채배 엔코더 펄스임.

SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 home 동작 개시 방향의 리밋센서를 찾은 후 (+방향이면 LSxA 입력이 0, -방향이면 LSxC 입력이 0이 될 때 리밋을 찾은 것으로 봄) 반대방향으로 움직여서 흡센서를 찾은 후 (LSxB 입력이 0이 될 때 흡을 찾은 것으로 봄) 흡센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함. 만약 리밋센서보다 흡센서를 먼저 발견하게 되면 바로 반대방향으로 흡센서가 꺼질때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 offset 변위만큼 이동함. 이때 흡센서가 꺼지기 전에 home 동작 개시 반대 방향의 리밋센서가 ON 되어 있으면 흡센서와 리밋센서가 공용인 것으로 판단하여 이동방향을 바꾼 후에 흡센서가 꺼질때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 offset 변위만큼 이동함. SI 명령어로 설정된 위치 범위내에서만 리밋센서와 흡센서를 찾으며 만약 범위를 벗어나면 즉시 명령의 수행을 중지하고 정지함. SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함.

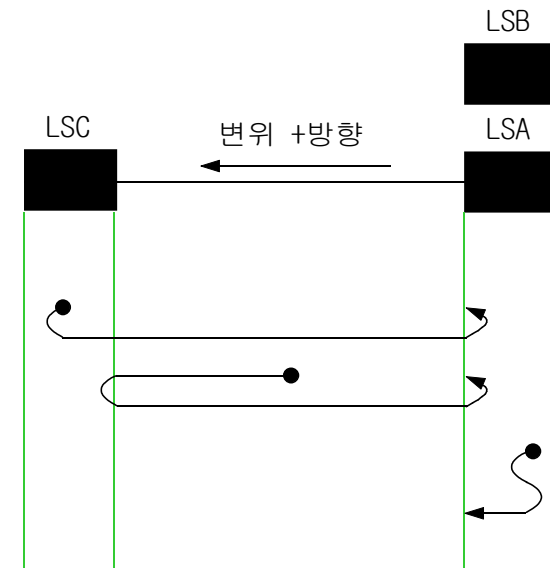
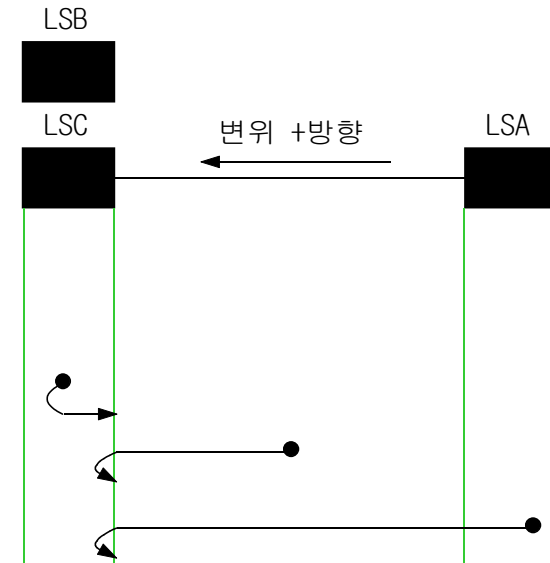
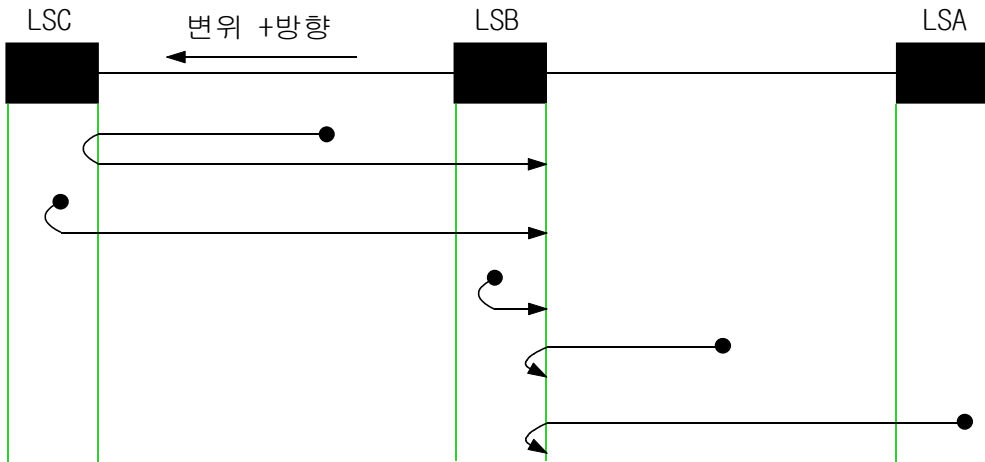
PM/PP/PQ/Pp/Pq 명령에 의한 연속계적동작을 안하고 있을 때로서, 흡동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 흡동작을 실행중이지 않을 때에 적용가능함.

2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.

흡동작 - 2개의 리밋센서와 1개의 흡센서를 쓰는 경우 : +방향으로 개시하였을때



흡동작 - 2개의 리밋센서와 1개의 홈센서를 쓰는 경우 : -방향으로 개시하였을때



흡동작 - 2개의 리밋센서만을 쓰는 경우

☞ 흡동작을 하려면

- 1) SMHLHL; 명령어로 01/02/03의 위치제어 모드를 선택함.
 상위/하위바이트가 01이면 모터2/모터1이 위치제어모드(inner loop 없고 PWM사용, DC 모터전용)
 상위/하위바이트가 02이면 모터2/모터1이 위치제어모드(inner loop 전류제어, DC/BLDC)
 상위/하위바이트가 03이면 모터2/모터1이 위치제어모드(inner loop 속도제어, BLDC 모터전용)
- 2) PEHLHL; 명령어로 해당 모터제어보드의 출력을 enable 시킴. (HLHL은 보드번호를 써야함)
- 3) SSddd1,ddd2; 명령어로 흡이동속도를 정의함.
- 4) GiHL,HL,iiii1,iiii2; 또는 GiHL,HL,iiii1,iiii2; 명령어로 흡동작을 개시함.
- 5) 흡동작의 상태는 Q1; 명령어로 확인할 수 있음.
- 6) 흡동작도중에 중지하고자하면 PRHLHL; 명령어로서 모든 에러 플래그를 clear 함.

Gi 와 Gi 명령에서 offset변위의 기본단위는 각각 0.01도와 4채배 엔코더 펄스임.

SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 home 동작 개시 방향의 리밋센서를 찾은 후 (+방향이면 LSxA 입력이 0, -방향이면 LSxB 입력이 0이 될 때 리밋을 찾은 것으로 봄) 리밋센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함.

SI 명령어로 설정된 위치 범위내에서만 리밋센서와 흡센서를 찾으며 만약 범위를 벗어나면 즉시 명령의 수행을 중지하고 정지함.

SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함.

PM/PP/PQ/Pp/Pq 명령에 의한 연속궤적동작을 안하고 있을 때로서, 흡동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 흡동작을 실행중이지 않을 때에 적용가능함.

2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.

흡동작 - 1개의 흡센서만을 쓰는 경우로서 흡센서의 방향을 알때

☞ 흡동작을 하려면

- 1) SMHLHL; 명령어로 01/02/03의 위치제어 모드를 선택함.
 상위/하위바이트가 01이면 모터2/모터1이 위치제어모드(inner loop 없고 PWM사용, DC 모터전용)
 상위/하위바이트가 02이면 모터2/모터1이 위치제어모드(inner loop 전류제어, DC/BLDC)
 상위/하위바이트가 03이면 모터2/모터1이 위치제어모드(inner loop 속도제어, BLDC 모터전용)
- 2) PEHLHL; 명령어로 해당 모터제어보드의 출력을 enable 시킴. (HLHL은 보드번호를 써야함)
- 3) SSddd1,ddd2; 명령어로 흡이동속도를 정의함.
- 4) GJHL,HL,iiii1,iiii2; 또는 GjHL,HL,iiii1,iiii2; 명령어로 흡동작을 개시함.
- 5) 흡동작의 상태는 Q1; 명령어로 확인할 수 있음.
- 6) 흡동작도중에 중지하고자하면 PRHLHL; 명령어로서 모든 에러 플래그를 clear 함.

GJ 와 Gj 명령에서 offset변위의 기본단위는 각각 0.01도와 4채배 엔코더 펄스임.

SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 흡센서를 찾은 후 (LSxB 입력이 0이 될 때 흡을 찾은 것으로 봄) 흡센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함. 만약 처음부터 흡센서가 ON이면 흡센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함. SI 명령어로 설정된 위치 범위내에서만 흡센서를 찾으며 만약 범위를 벗어나면 즉시 명령의 수행을 중지하고 정지함.

SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함.

PM/PP/PQ/Pp/Pq 명령에 의한 연속궤적동작을 안하고 있을 때로서, 흡동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 흡동작을 실행중이지 않을 때에 적용가능함.

2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.

흡동작 - 1개의 흡센서만을 쓰는 경우로서 흡센서 방향을 모를때

☞ 흡동작을 하려면

- 1) SMHLHL; 명령어로 01/02/03의 위치제어 모드를 선택함.
 상위/하위바이트가 01이면 모터2/모터1이 위치제어모드(inner loop 없고 PWM사용, DC 모터전용)
 상위/하위바이트가 02이면 모터2/모터1이 위치제어모드(inner loop 전류제어, DC/BLDC)
 상위/하위바이트가 03이면 모터2/모터1이 위치제어모드(inner loop 속도제어, BLDC 모터전용)
- 2) PEHLHL; 명령어로 해당 모터제어보드의 출력을 enable 시킴. (HLHL은 보드번호를 써야함)
- 3) SSddd1,ddd2; 명령어로 흡이동속도를 정의함.
- 4) GKHL,HL,iiii1,iiii2; 또는 GkHL,HL,iiii1,iiii2; 명령어로 흡동작을 개시함.
- 5) 흡동작의 상태는 Q1; 명령어로 확인할 수 있음.
- 6) 흡동작도중에 중지하고자하면 PRHLHL; 명령어로서 모든 에러 플래그를 clear함.

GK 와 Gk 명령에서 offset변위의 기본단위는 각각 0.01도와 4채배 엔코더 펄스임.

만약 처음부터 흡센서가 OFF(LSxB 입력이 1 일 때)이면 명령을 무시하며, 흡센서가 처음부터 ON이면 SS 명령어로 정의된 속도로 home 동작 개시방향으로 이동하여 흡센서가 꺼질 때까지 움직여서 목표위치, 엔코더값등을 초기값으로 재설정하고 (초기값은 5000000임) offset 변위만큼 이동함.

SM 명령에 의한 작동모드값이 01/02/03에서만 적용가능함.

PM/PP/PQ/Pp/Pq 명령에 의한 연속계적동작을 안하고 있을 때로서, 흡동작이 한번도 안되었을 때 또는 완료되었을 때 또는 오류가 발생했을 때 : 즉 흡동작을 실행중이지 않을 때에 적용가능함.

2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 흡센서로 ENCxZ를 사용할수 있도록 하였음.

제어모드의 설정 및 제어작동의 실행 (홈동작 - 1개의 홈센서만을 쓰는 Direct Drive 모터의 경우)

- ⚙ 이 경우는 처음부터 홈센서가 ON 이어야 하며, 이 경우 전원을 켜 위치가 홈위치로 될 뿐아니라, 모터의 영구자석 위치도 판별하는 기준이 됨.
- ⚙ 2014년 11월 15일 S/W 버전부터는 SX 명령어 3번째 파라미터의 bit8/9/10 설정에 따라서 홈센서로 ENCxZ를 사용할수 있도록 하였음.

명령어	구분	설명
;J; (+#^%)	operation	모터1의 속도모드 조그 작동을 수행함.
	return value	J;
	comment	J; 명령어를 계속해서 보내어 오는 동안은 조그작동을 수행하며, J;의 입력 여부는 10ms 마다 체크됨. 모터1이 05/06 속도모드로 정의되어 있을때 만 가능함.
;K; (+#^%)	operation	모터2의 속도모드 조그 작동을 수행함.
	return value	K;
	comment	K; 명령어를 계속해서 보내어 오는 동안은 조그작동을 수행하며, K;의 입력 여부는 10ms 마다 체크됨. 모터2가 05/06 속도모드로 정의되어 있을때 만 가능함.
;SJHL, dddd1, dddd2, dddd3; (+#^%) ;SJ?; (+#^%)	operation	속도모드의 조그동작에서 조그방향, 조그이동속도 및 조그동작 지속시간을 정의함. 1st byte : jog_move_direction 변수에 조그동작시에 방향을 저장함. [0이면 음의 방향, 1이면 양의 방향이며 하위 4bit는 모터1, 상위 4bit는 모터2] (00) dddd1 : jog_move_speed1 변수에 조그동작 속도를 설정함 [1 - MOTOR1_SPEED_MAX] (500 rpm) dddd2 : jog_move_speed2 변수에 조그동작 속도를 설정함 [1 - MOTOR2_SPEED_MAX] (500 rpm) dddd3 : jog_move_lasting_time 변수에 조그동작 지속시간 설정치를 저장함. [2 - 50] (10 10ms)
	return value	SJHL, dddd1, dddd2, dddd3;
	comment	SJ 명령을 보낸후에 J;/K; 명령어를 보내면 가속을 거쳐서 jog_move_speed1/jog_move_speed2 의 속도로 이동을 함. 조그동작을 계속하려면 jog_move_lasting_time×10ms 이내에 J;/K; 명령어를 보내기를 반복해야 함. J;/K; 를 마지막으로 보낸 후에 jog_move_lasting_time×10ms 만큼을 기다린 다음, 감속을 하면서 정지하고 조그작동을 멈춤. 가속/감속율은 Sa 명령어로 설정된 값에 따름. 조그동작 도중에 SJ 명령어를 보내면 조그속도를 바꿀 수 있음. 2012년 6월 13일 이전 S/W 버전에서는 +방향 조그이동속도를 크게 설정하지 못하는 문제가 있음.

명령어	구분	설명
;j; (+#^%)	operation	모터1의 위치모드 조그 작동을 수행함.
	return value	j;
	comment	j; 명령어를 계속해서 보내어 오는 동안은 조그작동을 수행하며, j;의 입력 여부는 10ms 마다 체크됨. 모터1이 01/03 위치모드로 정의되어 있을때 만 가능함.
;k; (+#^%)	operation	모터2의 위치모드 조그 작동을 수행함.
	return value	k;
	comment	k; 명령어를 계속해서 보내어 오는 동안은 조그작동을 수행하며, k;의 입력 여부는 10ms 마다 체크됨. 모터2가 01/03 위치모드로 정의되어 있을때 만 가능함.
;SjHL,dddd1,dddd2,dddd3; (+#^%) ;Sj?; (+#^%)	operation	위치모드의 조그동작에서 조그방향, 조그이동속도 및 조그동작 지속거리를 정의함. 1st byte : jog_move_direction 변수에 조그동작시에 방향을 저장함. [0이면 음의 방향, 1이면 양의 방향이며 하위 4bit는 모터1, 상위 4bit는 모터2] (00) dddd1 : jog_move_speed1 변수에 조그동작 속도를 설정함 [1 - MOTOR1_SPEED_MAX] (500 rpm) dddd2 : jog_move_speed2 변수에 조그동작 속도를 설정함 [1 - MOTOR2_SPEED_MAX] (500 rpm) dddd3 : jog_move_step_size 변수에 조그동작 지속거리 설정치를 저장함. [2 - 1000] (200 pulse)
	return value	SjHL,dddd1,dddd2,dddd3;
	comment	Sj 명령을 보낸후에 j; 또는 k; 명령어를 보내면 가속을 거쳐서 jog_move_speed1/jog_move_speed2 의 속도로 이동을 함. 조그동작을 계속하려면 jog_move_step_size 거리를 이동하기 전에 j; 또는 k; 명령어를 보내기를 반복해야 함. j; 또는 k; 를 마지막으로 보낸 후에 jog_move_step_size 거리를 이동하는 동안 감속을 하면서 정지하고 조그작동을 멈춤. 가속/감속시간은 Ss 명령어로 설정된 값에 따름. 조그동작 도중에 Sj 명령어를 보내면 조그속도를 바꿀 수 있음. 조그동작 도중에 비정상적인 가감속을 피하려면 가속/감속시간이 짧거나 jog_move_step_size가 크거나 j; 또는 k; 명령어의 입력주기가 짧아야 함.

JOG 작동

⚙ JOG 동작을 하려면 (속도모드의 JOG)

1) SMHLHL; 명령어로 05/06의 속도제어 모드를 선택함.

상위/하위바이트가 05이면 모터2/모터1이 속도제어모드(inner loop 전류제어)

상위/하위바이트가 06이면 모터2/모터1이 속도제어모드(inner loop 없고 PWM사용(DC 2811), inner loop 속도제어(BLDC))

2) SJHL, dddd1, dddd2, dddd3; 명령어로 조그방향, 조그이동속도1/2 및 조그동작 지속시간을 설정함.

3) Saddd1, dddd2, dddd3, dddd4; 가감속도를 설정함. (단위는 RJM_VER7인 경우에 rpm/ms이고 나머지는 rpm/100ms 임)

4) PEHLHL; 명령어로 해당 모터제어보드의 출력을 enable 시킴. (HLHL은 보드번호를 써야함)

5) 모터1을 JOG 하려면 J; 를 입력하고, 모터2를 JOG 하려면 K; 를 입력함.

이때 조그동작 지속시간이내에 J; 나 K; 가 입력되면 입력 순간으로부터 조그동작 지속시간만큼 조그동작 정지시간이 연장됨.

⚙ JOG 동작을 하려면 (위치모드의 JOG)

1) SMHLHL; 명령어로 01/03의 위치제어 모드를 선택함.

상위/하위바이트가 01이면 모터2/모터1이 위치제어모드(inner loop 없음, PWM사용, DC 모터전용)

상위/하위바이트가 02이면 모터2/모터1이 위치제어모드(inner loop 전류제어, DC/BLDC, 현재 RJM_VER4(2811)/RJM_VER8 DC, BLDC 만 가능)

상위/하위바이트가 03이면 모터2/모터1이 위치제어모드(inner loop 속도제어, BLDC 모터전용)

2) SJHL, dddd1, dddd2, dddd3; 명령어로 조그방향, 조그이동속도1/2 및 조그동작 지속시간을 설정함.

3) SjHL, dddd1, dddd2, dddd3; 명령어로 조그방향, 조그이동속도1/2 및 조그동작 지속거리를 설정함.

SJ 명령어를 이용하여 조그동작 지속시간을 설정한 다음에 Sj 명령어를 사용하여 나머지 파라미터를 설정해야함.

4) Ssddd1, dddd2; 가감속 시간을 설정함. (단위는 ms 임)

5) PEHLHL; 명령어로 해당 모터제어보드의 출력을 enable 시킴. (HLHL은 보드번호를 써야함)

6) 모터1을 JOG 하려면 j; 를 입력하고, 모터2를 JOG 하려면 k; 를 입력함.

이때 조그동작 지속거리 만큼을 이동하기 전에 j; 나 k; 가 입력되면 현 위치로부터 조그동작 지속거리만큼 정지위치가 연장됨.

명령어	구분	설명
;SAHLHL; (+#^%) ;SA?; (+#^%)	operation	<p>모터제어기의 주소를 설정함.</p> <p>1st word 2012년 10월 18일 이전 S/W 버전인 경우 또는 SX 명령어 첫번째파라미터의 bit0 = 1 인 경우 : 주소값 [0000 - 00FF] (807F) 2012년 10월 18일 S/W 버전부터로서 SX 명령어 첫번째파라미터의 bit0 = 0 인 경우 : 주소값 [0000 - 00FF] (807F) - SX 명령어 두번째파라미터의 bit15/14 = 11 일때 : 주소값 [0000 - 07FF] (007F) - SX 명령어 두번째파라미터의 bit15/14 = 10 일때 : 주소값 [0000 - FFFF] (007F) - SX 명령어 두번째파라미터의 bit15/14 = 01 일때 : 주소값 [0000 - 00FF] (807F) - SX 명령어 두번째파라미터의 bit15/14 = 00 일때</p>
	return value	SAHLHL;
	comment	<p>⚡ 2012년 10월 18일 이전 S/W 버전인 경우 또는 SX 명령어 첫번째파라미터의 bit0 = 1 인 경우와 2012년 10월 18일 S/W 버전부터로서 SX 명령어 두번째파라미터의 bit15/14 = 11 또는 00 일때는, 주소값의 상위 바이트는 반드시 하위바이트의 inverse이어야만 baord 주소의 설정이 유효함.</p> <p>⚡ 2012년 10월 18일 S/W 버전부터로서 SX 명령어 첫번째파라미터의 bit0 = 0 일때 SX 명령어 두번째파라미터의 bit15/14 값에 따라서 CAN 통신의 주소사용방법이 달라짐.</p> <p>bit15/14 = 11 일때 11-bit 주소모드로서 주소로 0-255만 사용 = 10 일때 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용 = 01 일때 29-bit 주소모드로서 주소로 0-65535만 사용 (CAN2.0B), 2012년 10월 16일부터 적용 = 00 일때 11-bit 주소모드로서 주소로 0-255만 사용</p> <p>⚡ 주소값으로서 254(0xFE) 및 255(0xFF)는 bradcasting용으로 사용하는 주소이므로 126(0x8E), 127(0x8F), 254(0xFE) 및 255(0xFF)는 제어기의 주소로 사용하면 안됨.</p> <p>⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
	operation	위치, 속도, 전류제어 등의 작동모드를 설정함. 1st word : 작동모드 [0000 - FFFF] (0000)
	return value	SMHLHL;
	comment	<p>☞ 상위/하위바이트가</p> <p>01 : 모터2/모터1이 위치제어모드(inner loop 없고 PWM사용, DC/BLDC)</p> <p>02 : 모터2/모터1이 위치제어모드(inner loop 전류제어, DC/BLDC/STEP(엔코더형))</p> <p>03 : 모터2/모터1이 위치제어모드(inner loop 속도제어, DC/BLDC/STEP(엔코더형))</p> <p>04 : 모터2/모터1이 속도제어모드(위치제어형식으로 속도제어, DC/BLDC/STEP(엔코더형))</p> <p>05 : 모터2/모터1이 속도제어모드(inner loop 전류제어, DC/BLDC/STEP(엔코더형))</p> <p>06 : 모터2/모터1이 속도제어모드(inner loop 없고 PWM사용, DC/BLDC)</p> <p>07 : 모터2/모터1이 전류제어모드(DC/BLDC/STEP(엔코더형))</p> <p>08 : 모터2/모터1이 host electrical angle 속도제어(BLDC/STEP, STEP의 경우 위치명령도 가능)</p> <p>09 : 모터2/모터1이 host electrical angle 속도제어(BLDC/STEP)</p> <p>0A : 모터2/모터1이 속도제어모드(03번 위치제어에서 사용하는 속도제어와 같음, DC/BLDC/STEP(엔코더형))</p> <p>0D : 모터2/모터1이 사인파 위치제어(DC/BLDC/STEP, 03 또는 08번 위치제어모드 사용,</p> <p>0E : 모터2/모터1이 사인파 위치제어(DC/BLDC/STEP, 01 또는 03 또는 08번 위치제어모드 사용</p> <p>0F : 모터2/모터1이 사인파 위치제어(DC/BLDC/STEP, 02 또는 03 또는 08번 위치제어모드 사용</p> <p>00 : 하위바이트가 00이면 모터2 off, 하위바이트가 00이면 모터1 off.</p> <p>☞ 04번 모드는 02번 위치제어모드를 사용하여 속도를 제어함.</p> <p>☞ 모드간 전환을 하는 경우에 이전 모드의 작동을 정지하고 새 모드의 작동을 개시함. 이때 위치/속도/전류/electrical-angle 등의 설정치는 0 또는 기본값으로 설정됨.</p> <p>☞ 380V 제어기의 경우에는 PE 명령어로 서보를 enable 시킨후에 SM 명령어로 작동모드를 선택해야, 0.55초후에 전류측정 offset을 정상적으로 설정하게 되며, 이후에 PAL나 SV같은 구동명령을 사용 수 있음.</p>

명령어	구분	설명
		<ul style="list-style-type: none"> ⚡ 작동모드로 03/0D/0E가 선택되는 순간 IRMCK201의 가감속율을 9323 RPM/100ms 이상으로, RJM_VER7에서는 1000*(10/8) RPM/ms이상으로(2013년 5월 26일 이전 S/W 버전에서만) 설정함. 이는 위치제어의 성능을 높이는데 필요한 사항임. ⚡ SM 01/02/03/04/08/09/0D/0E/0F 모드로 설정할때 가감속시간 MOTORx_ACC_DEC_PERIOD_SET을 설정하도록 하였음. ⚡ 2007년 10월 31일 현재 01,02(RJM_VER4(F2811/F28334)/RJM_VER8 DC, BLDC),03,04,05(RJM_VER4(F2811/F28334)/RJM_VER8 DC, BLDC(F2811/F28334)),06,07,08,0F 만이 구현되어 있음. ⚡ 2010년 7월3일 S/W 버전부터 09번모드는 스테핑모터의 속도모드를 설정하는데 사용함. 스테핑모터의 08번모드의 작동은 예전과 동일하며, BLDC모터의 경우 09번 작동모드는 08번 작동모드로 처리됨. ⚡ 03번 작동모드는 2012년 10월 3일 S/W 버전부터 위치제어루프 내에 속도제어루프가 있는 위치제어모드로 사용할수 있음. 속도제어루프 안에는 전류제어루프가 있음. Digital Hall 센서만을 사용하는 BLDC에는 사용할수 없음. 2012년 10월 3일 S/W 버전부터 2013년 4월 15일 이전 S/W 버전에서는 게인 스케줄링을 사용하는 경우 오류가 있음. ⚡ 0A번 작동모드는 속도제어모드로서 2012년 10월 3일 S/W 버전부터 사용할수 있음. 03번 제어모드에서 속도제어루프의 게인을 설정하는데 쓸수 있음. Digital Hall 센서만을 사용하는 BLDC에는 사용할수 없음. 2011년 11월 20일 S/W 버전부터는 PE 명령어를 수행한 이후에만 SM 명령에 의한 작동모드 설정이 가능하게 하였음. 2011년 11월 27일 S/W 버전부터 2012년 12월 20일 이전의 S/W 버전에서는 위치모드 제어를 사용하던 도중에 완전히 정지하지 않고 SM0000; 명령을 사용하여 감속을 하였다가, 다시 위치모드로 재설정하면 잠깐 재감속하는 위치명령이 발생하는 문제점이 있으므로 주의를 요함. 2013년 1월 10일 S/W 버전부터는 03번 작동모드일 경우에 펄스주기를 측정하는 방식(SX 명령어 두번째 파라미터의 bit8로 설정)을 사용하면서 PWM 주파수(Z8 명령어 하위 8비트로 설정)가 60KHz 보다 크면 03번 모드의 설정을 못하게 하였음.

명령어	구분	설명
		<p>2012년 10월 3일 S/W 버전부터 2013년 4월 15일 S/W 버전은 03번 작동모드일 경우에 속도에 따라서 계인을 스케줄링하는 부분에서 속도명령값의 크기를 잘못 계산하여, 계인스케줄링이 정확히 이루어지지 않는 문제점이 있음.</p> <p>2013년 4월 25일 이전 버전에서는 Z8 명령어로 설정된 PWM 주파수가 60KHz를 넘는 경우, 03번 작동모드로 설정이 안되는 경우가 있는데, 이때는 Z8 명령어로 PWM 주파수를 60KHz 와 같거나 작게 설정하면 됨.</p> <p>리턴값이 SM0000: 인 경우에 작동모드 설정을 제대로 못했다는 뜻이며, 이러한 상황은</p> <p>(1) PWM 형식의 전류센서를 사용하는 경우 PE 명령을 수행할때 상전류/총전류값이 0에 해당하는 offset 값을 측정하게 되는데, 이 전류 offset 값이 비정상적인 값을 갖을 때 발생하며, (2) 그외 AD8205/AD8206을 사용하는 경우는 전원 투입후 초기화 과정에서 상전류/총전류값이 0에 해당하는 offset 값을 측정하게 되는데, 이 전류 offset 값이 비정상적인 값을 갖을 때도 발생함. (3) 사용할 수 없는 작동모드를 선택하였을 때는 해당 채널만 00값으로 리턴됨. (4) 펄스주기를 사용하여 속도계산을 하는 모드를 선택하였을때 (SX 명령어 두번째 파라미터의 BIT8이 0일 때) PWM 주파수가 60KHz 보다 클때 (Z8 명령어로 설정하는 값은 하위 바이트가 06보다 클때)</p> <p>특히 총전류 측정회로에서 AD8205/AD8206의 전류 offset을 설정하는 저항이 0 Ohm이 아닐 경우에는 0 Ohm으로 교체해주어야 함. (모터제어기의 구버전 하드웨어인 경우에 한함)</p> <p>2014년 7월 21일 S/W버전부터는 07번 전류제어모드에서도 제어기의 발열에 따라서 강제로 전류제어리미트의 크기를 줄이는 기능이 작용되게 하였음.</p> <p>2014년 8월 16일 S/W버전부터는 처리할 수 없는 작동모드를 설정하면 강제로 0번모드로 설정되면서 이전의 제어작동을 멈추고 fault2_status 플래그(Q3; 명령어로 알 수 있음)를 설정함.</p> <p>☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

SM 명령어로 설정 가능한 작동 모드 (2013년 6월 03일 S/W 버전부터 적용)

	00	01 위치제어 (PWM 직접 설정)	02 위치제어 (내부전류 제어루프)	03 위치제어 (내부속도 제어루프)	04 위치제어방 식의 속도 제어	05 속도제어 (내부전류 제어루프)	06 속도제어 (PWM 직접 설정)	07 전류제 어	08 (위치)속도 제어 (개루프)	09 속도제어	0A 속도제어 (내부전류 제어루프)	0D 사인파 위 치제어	0E 사인파 위 치제어	0F 사인파 위 치제어
STEP (엔코더 없음)	<input type="radio"/> (PWM 출력 강제설정 가 능)	-	-	-	-	-	-	-	<input type="radio"/> (위치/속도명 령 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	-	<input type="radio"/> (08번 모드 로 실행)	<input type="radio"/> (08번 모드 로 실행)	<input type="radio"/> (08번 모드 로 실행)
STEP (엔코더 있고 lead angle 제어 사 용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	-	<input type="radio"/> (lead angle 제어 실시)	<input type="radio"/> (lead angle 제어 실시)	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/> (lead angle 제어 실시)	-	<input type="radio"/> (lead angle 제 어 실시)	<input type="radio"/> (위치/속도명 령 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (08번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
DC	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	-	-	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
BLDC (IRMCK201 을 사용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
AC induction motor	<input type="radio"/> (PWM 출력 강제설정 가 능)	-	-	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	-	-	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 속도 feedback 제어)	-	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (03번 모 드로 실행)	<input type="radio"/> (03번 모드 로 실행)
BLDC (digital Hall IC 만 사 용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
BLDC (digital Hall IC와 Encoder를 사 용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
BLDC (3상 analog Hall 센서를 사용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
BLDC (2상 resolver를 사 용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
BLDC (PWM 방식 의 절대각센서 를 사용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)
BLDC (통신방식 의 절대각센서 를 사용)	<input type="radio"/> (PWM 출력 강제설정 가 능)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (02번 모드 로 실행)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> (속도명령만 수신)	<input type="radio"/> (속도명령만 수신, 08번 드로 실행)	<input type="radio"/>	<input type="radio"/> (03번 모드 로 실행)	<input type="radio"/> (01번 모드 로 실행)	<input type="radio"/> (02번 모드 로 실행)

2013년 5월 8일 S/W 버전부터 적용되는 ADVANCED_CONTROLLER + OBSERVER 는 STEP 모터와 AC induction 모터를 제외한 모터들로서 02/03/05/0A 번 작동모드에서만 사용할수있으며, AU/AV/Au/Av/Aa/Az/SX 명령등으로 관련파라미터들이 적절하게 설정되어야 사용가능함.

observer+advanced_controller 작동 (2013년 6월 03일 S/W 버전부터 적용)

	02 위치제어 (내부전류제어루프)		03 위치제어 (내부에 0A번 속도제어루프)		04 위치 제어방식 의 속도제어		05 속도제어 (내부전류제어루프)		0A 속도제어 (내부전류제어루프)		0D 사인파 위치 제어		0F 사인파 위치 제어	
	((operation_mode_SWITC H_more & 0x00400000L)=0)&&(mot orx_activate==0xa55a)	그외					((operation_mode_SWITC H_more & 0x00c00000L)!=0x00c00000L)& &(motorx_activate==0xa55a)		03/0A/0D 번 모드로서 ((operation_mode_SWITC H_more & 0x00c00000L)!=0x00c00000L)& &(motorx_activate==0xa55a)					
STEP (엔코더 없음)	-	-												
STEP (엔코더 있고 lead angle 제어 사용)														
DC	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
BLDC (IRMCK201을 사 용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
AC induction motor														
BLDC (digital Hall IC 만 사용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
BLDC (digital Hall IC 와 Encoder를 사용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
BLDC (3상 analog Hall 센서를 사용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
BLDC (2상 resolver를 사용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
BLDC (PWM 방식의 절대 각센서를 사용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				
BLDC (통신방식의 절대 각센서를 사용)	advanced_control (1KHz)	PID	P-control	02번 실행	모드로	advanced_control (1KHz)	advanced_control (1KHz or 5KHz)	03번 실행	모드로	02번 실행				

2013년 5월 8일 S/W 버전부터 적용되는 ADVANCED_CONTROLLER + OBSERVER 는 STEP 모터와 AC induction 모터를 제외한 모터들로서 02/03/05/0A 번 작동모드에서만 사용할수있으며, AU/AV/Au/Av/Aa/Az/SX 명령등으로 관련파라미터들이 적절하게 설정되어야 사용가능함.

명령어	구분	설명
;STHLHL; (+#^%) ;ST?; (+#^%)	operation	모터의 기본파라미터를 쉽게 설정하기 위하여 미리 준비되어있는 motor type을 선정함. 하위 byte : 모터1의 tpye [00 - FF] 상위 byte : 모터2의 tpye [00 - FF]
	return value	STHLHL;
	comment	<p>⚙ EDA55A;가 후속으로 실행되어야 그 효과가 반영되는데, 이때 MOTOR1_PULSE_PER_REV, MOTOR1_GEAR_RATIO, MOTOR2_PULSE_PER_REV, MOTOR2_GEAR_RATIO 가 설정되고 제어 파라미터도 설정됨.</p> <p>⚙ motor 상위바이트/하위 바이트가</p> <p>00 : 4체배된 1024-PPR MAXON RE-max 29/24V/22W DC + 100:1 감속기 + max 8790 rpm, 01 : 4체배된 2048-PPR FAULHARBER 3257CR 24V/83W DC + 100:1 감속기 + max 5900 rpm, 02 : 4체배된 4096-PPR MAXON EC 45 flat 24V/50W(△) BLDC(8 pole pair) + 100:1 + max 6700 rpm, 03 : 4체배된 4000-PPR MAXON EC-powermax 30 24V/100W(△) BLDC(2 pole pair) + 100:1 + max 17800 rpm, 04 : 4체배된 2000-PPR MAXON EC 45 24V/250W(△) BLDC(1 pole pair) + 100:1 + max 9090 rpm, 05 : 4체배된 2000-PPR FAULHARBER 4490 24V/200W(△) BLDC(1 pole pair) + 100:1 + max 9550 rpm, 06 : analog Hall sensor FAULHARBER 3564 24V/100W(△) BLDC(1 pole pair) + 100:1 + max 11300 rpm, 07 : analog Hall sensor FAULHARBER 2057 24V/60W(△) BLDC(1 pole pair) + 100:1 + max 26500 rpm, 08 : analog Hall sensor FAULHARBER 4490 24V/200W(△) BLDC(1 pole pair) + 100:1 + max 9550 rpm, 09 : 4체배된 2000-PPR MAXON EC 90 flat 24V/90W BLDC(12 pole pair) + 100:1 + max 3190 rpm, 0A : 4체배된 4096-PPR KOLLMORGEN RBEH 01212 24V/181W BLDC(4 pole pair) + 100:1 + max 3820 rpm, 0B : 4체배된 10000-PPR KOMOTEK DANQ-01EF1N2 24V/100W BLDC(4 pole pair) + 100:1 + max 5000 rpm, 또는 KOMOTEK 380V/3KW 0C : 4체배된 8192-PPR KOLLMORGEN RBEH 01211/13 24V/152W/203W BLDC(4 pole pair) + 100:1 + max 5600/3240 rpm, 0D : 4체배된 8192-PPR KOLLMORGEN RBEH 02110 24V/241W BLDC(6 pole pair) + 100:1 + max 1470 rpm, 0E : 4체배된 8192-PPR KOLLMORGEN RBEH 00511 24V/52W BLDC(3 pole pair) + 100:1 + max 12800 rpm, 0F : 4체배된 4000-PPR MAXON RE35/40 48V/90W/150W DC + 100:1 감속기 + max 3810/7580 rpm,</p>

명령어	구분	설명
	comment	<p>10 : digital Hall-IC type AMT용 BLDC(7 pole pair) + 100:1 감속기 + max 6700 rpm, 11 : 4체배된 2048-PPR MAXON EC 45 flat 24V/50W(△) BLDC(8 pole pair) + 100:1 + max 6700 rpm, 12 : 4체배된 60000-PPR NANO MOTION linear 모터(2 pole pair) + 100:1 감속기 + max 3000 rpm, 13 : 4체배된 4000-PPR 퍼스텍 BLDC(3 pole pair) + 100:1 감속기 + max 12800 rpm, 14 : 4체배된 72000-PPR MAXON EC 90 flat 24V/90W BLDC(12 pole pair) + 100:1 + max 3190 rpm, 15 : 4체배된 8000-PPR MITSUBISHI 380V/3KW BLDC(4 pole pair) + 100:1 + max 5000(?) rpm, 16 : 4체배된 3276800-PPR YASKAWA DD-motor(10 pole pair) + 1:1 + max 120 rpm,</p> <p>그외 모터 type이 FFFF/FFFE 값이 아닌 한 2048-PPR FAULHARBER 3257CR 24V/83W + 100:1 + max 5900 rpm으로 기본값을 설정함.</p> <p>⚡ 모터 type이 FFFE 보다 작으면, PWM_PERIOD 값이 10/μs으로 설정되며 (PWM 출력주파수는 100KHz), 상기된 펄스수/감속비 설정에 따라 제어파라미터가 재설정됨.</p> <p>⚡ 모터 type이 FFFF 이면, SEA55A 명령어로 1회전당 모터펄스값과 SG 명령어로 PWM_PERIOD/감속비의 변경이 가능하고, 이어서 EDA55A; 명령을 수행하여 제어파라미터를 재설정하고, SA 명령어로 주소를 재설정하고, STFFFF; 명령을 수행한 다음, EsA55A; 명령으로 파라미터를 저장할 수 있음.</p> <p>⚡ 모터 type이 FFFF 인 상태에서 1회전당 모터펄스값과 PWM_PERIOD/감속비를 변경치 않으면 직전에 EEPROM에 저장되었던 파라미터가 사용되나, 이때 실수로 SG 명령이 수행되면 PWM_PERIOD 값이 변경될 수 있음. 따라서 파라미터의 저장 직전에 반드시 motor의 type을 FFFE 로 변경해야 함.</p> <p>⚡ 모터 type이 FFFE 이면, 1회전당 모터펄스값과 PWM_PERIOD/감속비의 변경이 안됨.</p> <p>⚡ 모터 type 06/07/08/09은 F2811/F28334에서만 설정이 가능함.</p> <p>⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

ST 명령을 이용한 모터 기본파라미터의 설정과 저장

- ☞ 모터의 기본파라미터를 설정하고 저장하려면 (상기 테이블에 존재하는 모터타입이 있는 경우)
 - 1) STHLHL; 명령을 실행하여 모터의 type를 지정함. (하위 8bit:모터1, 상위 8bit:모터2)
 - 2) EDA55A; 명령을 실행하며 파라미터를 초기화함. (이때 모터의 주소가 7F로 바뀜)
 - 3) SAHLHL; 명령을 실행하여 모터에 주소를 설정함.
 - 4) SB115200; 명령어를 사용하여 제어기 RS232C 통신의 기본 Baudrate를 바꾸어줌.
 - 5) ES0A00,0100; 명령 또는 EsA55A; 을 실행하여 EEPROM에 파라미터를 저장함.
 - 6) 전원을 껐다가 켜.
 - 7) ST?;, SA?;, SB?; 등을 사용하여 바뀐 파라미터가 반영되었는지를 확인함.

- ☞ 모터의 기본파라미터를 설정하고 저장하려면 (상기 테이블에 존재하는 모터타입이 없는 경우)
 - 1) STFFFF; 명령을 실행하고 나서 SE 명령과 SG 명령을 사용하여 모터1회전당 엔코더펄스수, PWM_PERIOD, 감속비, 최대속도 등을 설정함.
 - 2) EDA55A; 명령을 실행하며 파라미터를 초기화함. (이때 모터의 주소가 7F로 바뀜)
 - 3) SAHLHL; 명령을 실행하여 모터에 주소를 설정함.
 - 4) SB115200; 명령어를 사용하여 제어기 RS232C 통신의 기본 Baudrate를 바꾸어줌.
 - 5) STFFFE; 명령을 실행하여 모터타입을 FFFE로 설정함.
 - 6) ES0A00,0100; 명령 또는 EsA55A; 을 실행하여 EEPROM에 파라미터를 저장함.
 - 7) 전원을 껐다가 켜.
 - 8) ST?;, SA?;, SB?; 등을 사용하여 바뀐 파라미터가 반영되었는지를 확인함.

모터 기본파라미터의 설정과 저장

☞ 모터의 기본파라미터를 설정하고 저장하려면 (상기 테이블에 존재하는 모터타입을 사용하면서 엔코더 펄스수나 무부하 최고속도, 기어비 등을 바꾸고자 하면)

- 1) STHLHL; 명령을 실행하여 모터의 type를 지정함. (하위 8bit:모터1, 상위 8bit:모터2)
- 2) EDA55A; 명령을 실행하며 파라미터를 초기화함. (모든 파라미터를 factory default 값으로 설정함)
- 3) SEA55A,HLHL,HLHL,HLHL,HLHL; 명령을 실행하며 4체배된 엔코더 펄스수와 pole pair 수를 HEX ASCII로 입력함.
(DC/STEP 모터의 경우에 pole pair 수는 1이며, BLDC의 경우는 모터의 카타로그를 통하여 실제값을 입력하여야함.)
- 4) 만약 무부하 최고 속도나 감속비를 바꾸고자하면 SGA55A,dddd1,dddd2,dddd3,dddd4; 명령어로 바꾸고자 하는 값을 입력함.
- 5) EDA55A; 명령을 실행하여 파라미터를 다시 초기화함. (이 과정에서 바뀐 파라미터에 따른 초기화를 실시함)
- 6) SAHLHL; 명령어를 사용하여 제어기의 주소를 설정함.
- 7) SB115200; 명령어를 사용하여 제어기 RS232C 통신의 기본 Baudrate를 바꾸어줌.
- 8) ES0A00,0100; 명령 또는 EsA55A; 을 실행하여 EEPROM에 파라미터를 저장함.
- 9) 전원을 꺾다가 켜.
- 10) ST?;, SE?;, SG?;, SA?;, SB?; 등을 사용하여 바뀐 파라미터가 반영되었는지를 확인함.

※ CUBE-BLxxxx-SixR 모델의 경우는 엔코더 펄스수가 바뀌거나, pole pair가 바뀌는 경우는 RJMC 프로그램을 사용하여 파라미터의 갱신을 추가로 해주어야 함.

모터 기본파라미터의 설정과 저장

- ☞ analog Hall sensor 타입 BLDC 모터의 홀센서 파라미터를 설정하고 저장하려면
 - 1) STHLHL; 명령을 실행하여 모터의 type를 지정함. (하위 8bit:모터1, 상위 8bit:모터2)
 - 2) EDA55A; 명령을 실행하며 파라미터를 초기화함. (모든 파라미터를 factory default 값으로 설정함)
 - 3) SEA55A,4000,4000,HLHL,HLHL; 명령을 실행하며 pole pair 수를 HEX ASCII로 입력함.
(BLDC 모터의 카타로그를 통하여 실제값을 입력하여야함.)
 - 4) 만약 무부하 최고 속도나 감속비를 바꾸고자하면 SGA55A,dddd1,dddd2,dddd3,dddd4; 명령어로 바꾸고자 하는 값을 입력함.
 - 5) EDA55A; 명령을 실행하여 파라미터를 다시 초기화함. (이 과정에서 바뀐 파라미터에 따른 초기화를 실시함)
 - 6) SAHLHL; 명령어를 사용하여 제어기의 주소를 설정함.
 - 7) SB115200; 명령어를 사용하여 제어기 RS232C 통신의 기본 Baudrate를 바꾸어줌.
 - 8) ESOA00,0100; 명령 또는 EsA55A; 을 실행하여 EEPROM에 파라미터를 저장함.
 - 9) 전원을 껐다가 켜.
 - 10) ST?;, SE?;, SG?;, SA?;, SB?; 등을 사용하여 바뀐 파라미터가 반영되었는지를 확인함.
 - 11) WJ1,10,20,08,12; (여기서 8,10은 8000 RPM을 정역방향 전류치를 같게 하는 튜닝을 실시하는 RPM으로 정한다는 뜻이며, 파라미터 설정 완료후에 12000 RPM을 정역 구동을 최종시험하는 RPM으로 정한다는 뜻임.)
 - 13) WJ?;의 리턴값이 WJ7,xx,xx,xx,xx,xx; 이면 성공적으로 파라미터 튜닝과 저장이 이루어 졌다는 뜻임.

ST 명령을 이용한 모터 기본파라미터의 설정

STHLHL;에서 HL의 값	모터종류		엔코더 펄스수 (4채배된 값)	엔코더 안쓰는 경우	pole pair	Gear ratio	maximum speed [RPM]
00	DC	1024 pulse MAXON RE-max	1024	-	1	100	8790
01	DC	2048 pulse FAULHARBER	2048	-	1	100	5900
02	BLDC	4096 pulse MAXON EC 45 flat	4096	16384 (digital)	8	100	6700
03	BLDC	4000 pulse MAXON EC powermax	4000	16384 (digital)	2	100	17800
04	BLDC	2000 pulse MAXON EC 45	2000	16384 (digital)	1	100	9090
05	BLDC	2000 pulse FAULHARBER 4490 (△)	2000	16384 (digital)	1	100	9550
06	BLDC	analog Hall sensor FAULHARBER 3564 100W	-	16384 (analog)	1	100	11300
07	BLDC	analog Hall sensor FAULHARBER 2057 60W	-	16384 (analog)	1	100	26500
08	BLDC	analog Hall sensor FAULHARBER 4490 200W	-	16384 (analog)	1	100	9550
09	BLDC	2000 pulse MAXON EC 90 flat	2000	16384 (digital)	12	100	3190
0a	BLDC	4096 pulse KOLLMORGEN RBEH 01212 181W	4096	16384 (digital)	4	100	3820
0b	BLDC	10000 pulse KOMOTEK DANQ-01EF1N2 100W	10000	16384 (digital)	4	100	5000
0c	BLDC	8192 pulse KOLLMORGEN RBEH 01211/13 152/203W	8192	16384 (digital)	4	100	5600
0d	BLDC	8192 pulse KOLLMORGEN RBEH 02110 241W	8192	16384 (digital)	6	100	1470
0e	BLDC	8192 pulse KOLLMORGEN RBEH 00511 52W	8192	16384 (digital)	3	100	12800
0f	DC	4000 pulse MAXON RE35/40 48V 90/150W	4000	-	1	100	3810
10	BLDC	digital type AMT용	-	16384 (digital)	7	100	6700
11	BLDC	2048 pulse MAXON EC 45 flat 24V/50W(△)	2048	16384 (digital)	8	100	6700
12	LM	NANO MOTION linear 모터	60000 (15000 * 4)	16384 (analog)	2 (1 analog)	100	3000
13	BLDC	ENC type 퍼스텍	4000	16384 (digital)	3	100	12800
14	BLDC	2000 pulse MAXON EC 90 flat + linear scale	72000 (18000 * 4)	16384 (digital)	12	100	3190

ST 명령을 이용한 모터 기본파라미터의 설정

STHLHL;에서 HL의 값	모터종류		엔코더 펄스수 (4체배된 값)	엔코더 안쓰는 경우	pole pair	Gear ratio	maximum speed [RPM]
15	BLDC	8000 pulse MITSUBISHI 3KW (TMS28334/5 H/W 버전 만 적용가능)	8000	16384 (digital)	4	100	5000
16	DD	3276800 pulse YASKAWA DD-motor	3276800	-	10	1	120
17	AC induction	8000 pulse 100W SPG induction motor	8000	-	2	100	1500
18 ~ FD	DC	default	2048	-	1	100	5900
any value	STEP	50 pole (200 step or 1.8도) 2상 스텝모터	-	16384	50	1	3000
FE		사용자지정 파라미터인 경우	사용자지정	사용자지정	사용자 지정	사용자지정	사용자지정
FF		SE와 SG명령에 의한 파라미터 변경 가능 상태	SE 명령어로 설정	SE 명령어로 16384 값을 설정 해야함	SE 명 령어로 설정	SG 명령어 로 설정	SG 명령어로 설정

명령어	구분	설명
;SXHLHL; (+#^%) ;SX?; (+#^%)	operation	<p>통신과 제어의 option을 선택하는 DIP switch의 값을 설정함.</p> <p>bit0 : DIP switch 1 : 통신모드의 설정 [CAN(0)/RS485(1)]</p> <p>bit1 : 직선보간 기능의 사용여부 [안함(0)/실시함(1)]</p> <p>bit2 : 모터권선의 온도보호 기능의 사용여부 [안함(0)/실시함(1)]</p> <p>bit3 : 모터의 입력/출력 power, 출력토크, 효율, 모터권선/하우징의 온도계산의 실시여부 [안함(0)/실시함(1)]</p> <p>bit4 : DIP switch 4 : 통신테스트 모드의 설정</p> <p>bit5 : DIP switch 3 : 통신테스트 모드의 설정</p> <p>bit4/5 : 외부의 20X4 text display에 선택된 4개의 변수값을 실시간으로 표시하는 기능의 사용여부 (화면 update는 500ms 마다 실시됨) (SX 명령어 두번째파라미터의 bit16 ~ bit19 참조)</p> <p>bit6 : DIP switch 2 : ETHERCAT등의 통신모드의 설정. 2015년 3월 17일 S/W 버전부터 ETHERCAT H/W option이 장착되고 firmware도 ETHERCAT version 탑재되었을 때만 적용됨.</p> <p>bit7 : 홈 찾기를 위하여 사용하는 리미트센서를 선택함.</p> <p>bit8 : 모터2가 모터1을 추종하도록 제어하는 dummy모드의 사용여부 [안함(0)/실시함(1)]</p> <p>bit9 : 속도제어중 정지시에 적분항을 0으로 처리하는지의 여부 [안함(0)/실시함(1)]</p> <p>bit10 : 위치제어시에 (가상)위치리미트센서를 만나면 정지하고, 반대방향으로만 위치값 설정이 가능하게함. [적용(0)/비적용(1)]</p> <p>bit11 : analog Hall sensor type의 RJM_VER7에서 Hall sensor parameter를 매 출발시마다 갱신을 하는지의 여부 [안함(0)/실시함(1)]</p> <p>bit12 : 속도제어시에 (가상)위치리미트센서를 만나면 정지하고, 반대방향으로만 속도값명령이 설정가능토록 함. [안함(0)/실시함(1)]</p> <p>bit13 : LSxC입력단자의 전압에 따라서 속도명령 또는 전류명령을 설정 하는지의 여부 [안함(0)/실시함(1)] 엔코더가 있는 스텝모터의 경우는 위치명령도 엔코더에 의하여 설정되는지의 여부도 결정함.</p> <p>bit14 : sequence control mode에서 스텝동작 여부 [연속작동(0)/스텝작동(1)]</p> <p>bit15 : sequence control mode의 사용여부 [안함(0)/실시함(1)]</p>

명령어	구분	설명
	return value	SXHLHL; SXHLHLHLHL;(TMS320F2810/2811/2812/28334/28335 로서 2009년 8월 16일 S/W 버전부터 적용됨)
	comment	<ul style="list-style-type: none"> ☞ 2012년 5월 28일 S/W 버전부터는 SXHLHLHLHL,HLHLHLHL; 도 사용 가능함. ☞ bit0/4/5/6의 경우는 DIP switch가 실제로 존재할 때 현재 DIP switch의 상태를 나타내며, bit1/2/3/7/8-15의 경우는 소프트웨어적으로 설정된 DIP switch의 상태를 나타냄. 따라서 DIP switch가 실제로 존재하는 경우에는 bit0/4/5/6의 소프트웨어적 설정이 불가능함. ☞ DIP switch 1의 ON(0)/OFF(1)에 따라서 통신모드가 선택되며, bit0(DIP switch 1)가 0이면 CAN 통신 모드로 작동하며, bit0(DIP switch 1)가 1이면 RS485 통신 모드로 작동함. RS232C 통신은 항상 사용이 가능함. DIP switch 2에 따라서 ETHERCAT등의 통신모드의 설정이 가능한데, ETHERCAT H/W와 ETHERCAT firmware가 장착된 경우에 해당함. ETHERCAT이 선택되면 RS485는 사용할수 없음. ☞ DIP switch 10이 0 일 때 CAN 통신 모드로 작동하며, LAN 이 장치된 제어기에서는 LAN 통신이 가능함. DIP switch 10이 1 일 때는 LAN 이 장치되었더라도 RS485 통신만 가능함. ☞ DIP switch 3/4의 ON(0)/OFF(1)에 따라서 통신테스트모드 설정되며, DIP switch 3/4가 1/0일때, 통신테스트모드가 설정됨. 이 경우는 받은 값을 2msec 후에 바로 되돌려주며, 수신에러가 발생하였을 때는 수신에러 상태값(SCI_lRXST_{reg})을 되돌려줌. ☞ 통신 모드의 경우는 SXHLHL; 로 값을 설정하고 ES0A00,0100; 명령어로 EEPROM에 저장한 다음, 전원을 재 투입하고 나면 바뀐 설정에 따른 동작을 실행하나, bit6의 경우는 SXHLHL; 로 값을 설정하는 순간부터 동작에 반영됨. (이 설정은 F2811/F28334에서만 설정이 가능함) ☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. ☞ bit10이 1이면 직선보간 기능을 사용함. (모터작동모드가 02/05/07번 일때로서 2축제어기의 경우만 해당됨.) (이 설정은 F2811/F28334에서만 설정이 가능함) ☞ bit2가 1이면 모터권선의 온도보호 기능을 작동 시킴. (모터작동모드가 02/05/07번 일때만 사용가능) (이 설정은 F2811/F28334에서만 설정이 가능함) ☞ bit3가 1이면 모터의 입력/출력 power, 출력토크, 효율, 모터권선/하우징의 온도계산을 실시함. (모터작동모드가 02/05/07번 일때만 사용가능) (이 설정은 F2811/F28334에서만 설정이 가능함)

명령어	구분	설명
	comment	<ul style="list-style-type: none"> ☞ bit4/5가 1/0이면 외부의 20X4 text display에 선택된 4개의 변수값을 실시간으로 표시하는 기능을 실시함. 기능설정에 사용하는 명령은 DZ/DF/DS/DT 임. (RS485_MODE, EtherNetIP_or_USB_MODE, COM_TEST_MODE가 아닐때만 사용가능) (이 설정은 F2811/F28334에서만 설정이 가능함) ☞ bit6가 0일때 ETHERCAT등의 통신모드의 설정. 2015년 3월 17일 S/W 버전부터 ETHERCAT H/W option이 장착되고 firmware도 ETHERCAT version 탑재되었을 때만 적용됨. ☞ bit7이 1이면 리미트센서로 LSxA(+리미트), LSxB(-리미트)를 사용하며, 0이면 리미트센서로 LSxA(+리미트), LSxC(-리미트)를 사용함. bit10이 0이어서 positionx_limit을 넘어설 때 강제 정지하는 기능에만 적용되며 홀동작에는 적용되지 않음. (이 설정은 F2811/F28334에서만 설정이 가능함) ☞ bit8이 1이면 두번째 채널의 PWM은 첫번째 채널의 PWM 출력을 copy 하여 실행되며, 두번째 채널의 엔코더 입력에 의한 제어는 무시됨. ☞ bit9이 1이면 속도제어중 속도명령이 0일때 적분제어항을 0으로 처리함으로써, 정지시에 적분제어항 때문에 원위치를 유지하려는 작동을 계속함으로써 모터에 과부하를 주는 일을 막을수 있음. 이렇게 하면 속도명령이 0일때는 비례항 이외에는 반력을 만들지 않으나, 05번 속도제어 모드에서는 전류 offset에 따라서 모터가 한 방향으로 흐르는 문제가 발생할 수 있음. 제일메디컬의 경우는 무조건 0으로 처리함. ☞ bit10이 0이면 위치모드에서 구동중에 위치리미트센서 또는 가상위치리미트를 만나면 강제정지하도록 함. 2011년 6월 01일, 2011년 9월 13일, 2012년 10월 19일 삼성메디컬 수정버전과, 버전 2013년 11월 26일 S/W 버전부터는 작동모드 00에서 PW 명령을 사용하여 PWM duty를 설정하는 경우에도 적용됨. 2014년 10월 07일 이전의 S/W 버전에서는 BLDC에서 엔코더를 안쓰는 경우 정지위치의 설정값에 오류가 있음. ☞ bit11이 1이면 analog Hall sensor type의 RJM_VER7 경우 Hall sensor parameter를 속도 0에서 출발할 때 마다 매번 계산하여 반영하는데, 단 가속이 10과 같거나 작아야 하고, 06번 작동 모드여야 함. (이 설정은 F2811/F28334에서만 설정이 가능함). ☞ bit12가 1이면 속도모드에서 구동중에 위치리미트센서 또는 가상위치리미트를 만나면 강제정지하도록 함. (이 설정은 F2811/F28334에서만 설정이 가능함).

명령어	구분	설명
	comment	<p>⚡ bit13가 1이면 LSxC입력단자의 전압에 따라서 속도명령값(속도모드의 경우) 또는 전류명령값(전류모드의 경우)을 설정함. 속도 설정 범위는 SG 명령어로 설정되는 최대 속도 까지 임. 전류설정범위는 Sw 명령어로 설정되는 전류제어리밋까지 임. 엔코더가 있는 스텝모터의 경우는 위치명령도 엔코더에 의하여 설정됨. 속도명령값을 설정하는 것인지, 전류명령값을 설정하는 것인지, 위치명령값을 설정하는 것 인지는 SM 명령어의 의한 작동모드 설정에 따름. (이 설정은 F2811/F28334에서만 설정이 가능함)</p> <p>⚡ bit14가 1이면 sequence control mode에서 스텝동작을 실시하며, 0이면 연속작동을 실시함. (이 설정은 F2811/F28334에서만 설정이 가능함)</p> <p>⚡ bit15가 1이면 sequence 제어가 실시되며, 0이면 sequence 제어를 실시하지 않음. (이 설정은 F2811/F28334에서만 설정이 가능함)</p> <p>⚡ 32비트의 리턴값은 2009년 8월 16일 S/W 버전부터 적용되며, 2009년 8월 16일 이전 S/W 버전은 16비트의 리턴값이 적용됨.</p>

명령어	구분	설명
;SXHLHLHLHL; (+#^) ;SX?; (+#^)	operation	통신과 제어의 option을 선택하는 DIP switch의 값을 설정함. bit0-15 : SXHLHL; 의 경우와 같음. bit16 : 모터1의 엔코더에서 A/B 상의 신호를 서로 바꾸는 기능의 사용여부 [안함(0)/실시함(1)] (이 설정은 F2809/F28334에서만 설정이 가능하나, 2010년 7월 17일 S/W 버전부터는 F2811에서도 가능하며, 2011년 10월 16일 S/W 버전부터는 절대각센서에도 적용됨) bit17 : 모터2의 엔코더에서 A/B 상의 신호를 서로 바꾸는 기능의 사용여부 [안함(0)/실시함(1)] (이 설정은 F2809/F28334에서만 설정이 가능함, 2010년 7월 17일 S/W 버전부터는 F2811에서도 가능하며, 2011년 10월 16일 S/W 버전부터는 절대각센서에도 적용됨) bit18 : 스텝핑모터에서 50V 승압회로의 사용여부 [안함(0)/실시함(1)] (이 설정은 F28334에서만 설정이 가능함) 속도모드에서 전압의 크기에 따라 가속/감속도를 조정하는 기능의 사용여부 [실시함(0)/안함(1)] (이 설정은 F28334에서만 설정이 가능함) bit19 : 스텝핑모터1에서 엔코더가 있는 경우에 lead angle control의 적용여부 [안함(0)/실시함(1)] (이 설정은 F28334에서만 설정이 가능함) bit20 : 스텝핑모터2에서 엔코더가 있는 경우에 lead angle control의 적용여부 [안함(0)/실시함(1)] (이 설정은 F28334에서만 설정이 가능함) bit21 : 스텝핑모터에서 저속에서의 전류스위칭 소음을 줄이기 위하여 stepping motor의 전류제어게인을 모터속도에 따라서 가변하는 기능을 사용하는 여부 [안함(1)/실시함(0)] (이 설정은 F28334에서만 설정이 가능함) bit22 : 위치리미트초과 오류검출에 위치리미트센서와 가상위치리미트중 어느것을 사용하는지를 설정함 [가상위치리미트(1)/위치리미트센서(0)], 이 기능은 2011년 10월 01일 S/W 버전부터 적용되며, 이전 S/W 버전은 위치리미트센서만 사용함. bit23 : digital Hall 만을 사용하는 BLDC 모터에서 05번 속도제어중 정지후 출발할때 위치값을 초기화하는지를 설정함 [사용안함(1)/사용함(0)], 이 기능은 2015년 1월 10일 S/W 버전부터 적용되며, 2010년 11월 30일 이전 S/W 버전에서는 위치값 초기화 기능이 없고, 2010년 11월 30일 S/W 버전부터는 위치값 초기화 기능을 사용하는 것이 기본임. bit24 : 위치제어에서 설정속도가 과도할 때 발생하는 문제를 처리 하는 여부 [안함(1)/실시함(0)]

명령어	구분	설명
	operation	<p>bit25 : 1ms main loop의 동기화 및 위치/전류 명령값의 수신을 외부로부터의 CAN 데이터에 의하여 실시하는 여부 (2012년 12월 26일 S/W 버전부터 정상 작동되며 CAN 통신에서만 가능) [안함(0)/실시함(1)] 2013년 1월 10일 S/W 버전부터 전송되어온 위치/전류 명령값이 모터제어기에 설정된 작동모드와 부합되지 않으면 전송되어온 값은 무시됨. 2013년 1월 24일 이전의 S/W 버전에서는 전송 bit25가 세트되어 있는 경우에, FF번 주소를 사용한 1ms main loop의 동기화 명령과 RS232를 통한 통신이외에는 처리하지 않음. 2013년 11월 10일 S/W 버전부터는 1ms main loop의 동기화 및 위치/전류 명령값의 수신이 실시되고 있는 중에 모터제어기의 상태값을 상위제어기로 전송하며, 주소는 해당 모터제어기의 주소와 같음. 전송되는 값은 SX 명령어 다음의 표와 같음.</p> <p>bit26 : 속도제어에서 감속에 따른 역기전력을 감소시키는 기능을 사용하는 여부 [안함(1)/실시함(0)] bit27 : 위치제어에서 PA/PB/Pa 명령수행시에 감속에 따른 역기전력을 감소시키는 기능을 사용하는 여부 [안함(0)/실시함(1)] bit29:28 : 00 일때 BLDC 모터제어에서 space vector PWM을 사용함. 01 일때 BLDC 모터제어에서 modified-SVPWM을 사용함. 10 일때 BLDC 모터제어에서 사인파 PWM을 사용함. 11 일때 BLOCK COMMUTATION 방식을 사용함.</p> <p>bit30 : 모터작동 불능 fault가 발생하였을때 모터제어기를 자동으로 위치제어모드로 재부팅하는 여부 [안함(0)/실시함(1)] (이 기능은 analog 센서타입의 BLDC 모터에는 적용되지 않음) bit31 : 일정변위를 연속으로 움직이게 하는 데모동작의 사용여부 [안함(0)/실시함(1)]</p>
	return value	<p>SXHLHLHLHL; SXHLHLHLHL,HLHLHLHL; (2012년 5월 28일 S/W 버전부터) SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터는)</p>
	comment	<p>엔코더에서 A/B 상의 신호를 서로 바꾸는 것은 엔코더 카운트값이 증가하는 회전방향이 바뀐다는 것으로서, 물리적인 연결이 바뀌는 것은 아님.</p> <p>bit25가 1인 경우에 모터제어기는 상위로부터 총 8바이트를 수신하며 처음 7바이트와 8번째 바이트의 하위 4비트는 위치 또는 전류명령값 또는 자이로 각속도이며 8번째 바이트의 상위 4비트가 1/2/3/4 이면 위치명령/전류명령/각속도/사인파전류명령 임을 나타냄.</p> <p>bit26, bit27 과 관련한 파라미터는 SD 명령과 Sd 명령으로 설정됨. bit28, bit29로 BLDC 모터제어에서 사인파 PWM 제어 방식이 선택되었을때는 PWM 주파수를 80KHz 이내에서만 사용할 것. (Z8 명령어 참조)</p>

명령어	구분	설명
	comment	<p>SX 명령어에 의한 32비트의 설정은 TMS320F2810/2811/2812/28334/28335 로서 2009년 8월 16일 S/W 버전부터 적용됨.</p> <p>bit30이 1 이더라도 모터작동 불능 fault의 사유가 아니고 정상적으로 전원을 투입한 경우라면 위치제어모드로 자동 부팅되지 않으며, analog 센서타입의 BLDC 모터의 경우는 원점을 감지하는 초기동작이 필요하므로 안전상 재부팅을 안하도록 하였음.</p> <p>☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;SXHLHLHLHL,HLHL HLHL; (+#^) ;SX?; (+#^) 	operation	1st long word : SXHLHLHLHL; 의 경우와 같음. 2nd long word bit0/1/2/6 : 1 일때 모터1의 리밋센서 A/B/C/D 입력전압이 low 이면 active : 0 일때 모터1의 리밋센서 A/B/C/D 입력전압이 high 이면 active bit3/4/5/7 : 1 일때 모터2의 리밋센서 A/B/C/D 입력전압이 low 이면 active : 0 일때 모터2의 리밋센서 A/B/C/D 입력전압이 high 이면 active bit8 : 1 일때 03/05/06/0A번 제어모드에서 속도제어시에 speedx_RPM_10을 사용함. (기존 방식) : 0 일때 03/05/06/0A번 제어모드에서 속도제어시에 speedx_RPM_combined_10을 사용함. (TMS320F28334/5 H/W 버전에서만 펄스주기 방식에 의한 속도값이 적용되며 TMS320F2810/11/12에서는 펄스카운트 방식의 속도값이 적용됨) bit10/9 : 0 일때 03번 위치제어 sample time을 10ms 로 설정함. : 1 일때 03번 위치제어 sample time을 5ms 로 설정함. : 2 일때 03번 위치제어 sample time을 2ms 로 설정함. : 3 일때 03번 위치제어 sample time을 1ms 로 설정함. bit11 : digital Hall IC만을 사용하는 BLDC 모터를 제외한 경우에만 해당되며, bit11이 0인 경우 05/06번 속도제어에서 속도제어루프에 저속에서 속도제어게인을 낮추는 기능을 적용함. (Wd 명령어 참조) bit12 : 0 일때 1ms당의 펄스를 카운트하면서 1ms unit timer를 사용함. (TMS320F28334/5 H/W 버전인 경우로서 엔코더를 사용하는 경우에만 적용 가능함) : 1 일때 1ms당의 펄스를 카운트하면서 1ms unit timer를 사용하지 않음. bit13 : 0 일때 CAN이나 RS485 통신 포트를 통하여 입력된 명령을 RS232C 포트로 전송함. 1일때 상기 작동을 하지 않음. bit15/14 : 3 일때 CAN 11-bit 주소모드로서 주소로 0-255만 사용 : 2 일때 CAN 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용 : 1 일때 CAN 29-bit 주소모드로서 주소로 0-65535만 사용, 2012년 10월 16일부터 적용 : 0 일때 CAN 11-bit 주소모드로서 주소로 0-255만 사용

명령어	구분	설명
;SXHLHLHLHL,HLHL HLHL; (+#^) ;SX?; (+#^)	operation	<p>bit19/18/17/16</p> <p>F : 외부표시모드는 SX 명령어 첫번째 파라미터의 bit4/5 의 설정에 따름. (2013년 2월 14일 이전 S/W 버전과 호환됨)</p> <p>0 : 외부표시모드가 작동되지 않음.</p> <p>1 : host로부터 DT 명령을 사용하여 외부표시내용을 받아서 표시하는 모드이며 표시내용이 자주 바뀌지 않는 경우에 사용함. (화면 update는 250ms 마다 실시됨)</p> <p>2 : host로부터 DT 명령을 사용하여 외부표시내용을 받아서 표시하는 모드이며 표시내용이 자주 바뀌는 경우에 사용함. (화면 update는 250ms 마다 실시됨)</p> <p>bit20 : 1 : 속도모드에서 오차가 커서 순간속도명령값을 수정할때 속도오차의 적분값을 초기화함. : 0 : 속도모드에서 오차가 커서 순간속도명령값을 수정할때 속도오차의 적분값을 초기화 안함.</p> <p>bit21 : 1 : Qx monitoring에서 전류치를 0.01A 단위로 보여주지 않음. : 0 : Qx monitoring에서 전류치를 0.01A 단위로 보여줌.</p> <p>bit23/22 : 01 : 02/03/05/0A번 작동모드에서 위치/속도 오차를 feedback하는 advanced controller를 적용함. (단 bit25에 따라서 03/0A번 모드에서 advanced controller가 1KHz/5KHz로 호출되며, AU/Au/AV/Av/Aa/Az 명령을 사용하여 파라미터가 적절히 설정되어야함)</p> <p>00 : 02/03/05/0A번 작동모드에서 위치/속도 오차를 feedback하는 advanced controller를 적용함. (단 bit25에 관계없이 advanced controller가 1KHz로 호출되며, AU/Au/AV/Av/Aa/Az 명령을 사용하여 파라미터가 적절히 설정되어야함)</p> <p>10 : 03/05/0A번 작동모드에서 속도오차대신 속도를 feedback 하는 advanced controller를 적용함. (단 bit25에 따라서 03/0A번 모드에서 advanced controller가 1KHz/5KHz로 호출되며, AU/Au/AV/Av/Aa/Az 명령을 사용하여 파라미터가 적절히 설정되어야함)</p> <p>11 : advanced controller를 적용하지 않음.</p> <p>bit24 : 1 : 모터1과 모터2에 대하여 At 명령으로 설정한 모터의 제어출력값계산 및 갱신시점을 맞추는 동작을 하지 않음. : 0 : 모터1과 모터2에 대하여 At 명령으로 설정한 모터의 제어출력값계산 및 갱신시점을 맞추는 동작을 실시함.</p> <p>bit25 : 1 : observer 및 03/0A 작동모드의 advanced 속도제어를 1KHz로 호출함. (sample time은 1ms) 0 : observer 및 03/0A 작동모드의 advanced 속도제어를 5KHz로 호출함. 이 경우에는 observer와 속도제어의 계산을 계산하고 설정할때 sample time을 0.2ms로 설정한 값이 사용되어야하며, 위치제어계산은 항상 sample time을 1ms로 설정한 값이 사용되어야함. (단 bit23/22이 00 이면 advanced 속도제어의 호출은 1KHz로 실시함.)</p>

명령어	구분	설명
	operation	<p>bit26 : 1 : 위치제어에서 위치오차의 변화량을 미분제어에 사용함. (통상값) 0 : 위치제어에서 위치오차의 변화량 대신에 속도값을 미분제어에 사용함. 5KHz observer를 사용하는 것이 좋으며, 1KHz observer를 사용하거나 observer를 사용하지 않는 경우는 위치제어게인을 1/2 ~ 1/5가량 낮추어야 안정하게 동작함. 위치명령의 미분값이 적용되지 않음으로 p 명령에 의한 연속 궤적구동시에 통신주기의 jitter에 의한 jirk를 줄일수 있음. 2013년 10월 04일 S/W 버전부터 적용됨.</p> <p>bit27 : 1 : CAN 통신을 사용할 때, 일정주기로 모터1의 상태값을 전송하지 않음. (통상값) 0 : CAN 통신을 사용할 때, 일정주기로 모터1의 상태값을 전송함. 전송되는 주소는 모터1의 경우 기본값 0xfd 임. 주기값과 주소는 EU 명령을 참조하고, 전송되는 데이터는 SX 명령어 다음에 있는 내용을 참조. SX 명령어 첫번째 파라미터의 bit25가 1 이거나 (1ms main loop의 동기화 경우), 국방로봇용 제어기이거나, CAN 통신모드가 아닌 경우는 적용되지 않음. 설정후 재부팅하여야 반영됨. 2013년 11월 22일 S/W 버전부터 적용됨. 전송되는 데이터의 내용은 Eu 명령어로 바꿀 수 있음 (2014년 5월 25일 S/W 버전부터).</p> <p>bit28 : 1 : 02번 위치제어모드에서 정지시에는 01번 모드로 전환하는 기능을 사용안함. (통상값) 0 : 02번 위치제어모드에서 정지시에는 01번 모드로 전환하여 정지시에 진동과 노이즈를 저감함. 2013년 11월 22일 S/W 버전부터 적용됨.</p> <p>bit29 : 1 : SX 명령어 첫번째 파라미터의 비트 25가 1 일때 (CAN 통신을 이용한 1ms 동기제어모드) 모터제어기가 위치제어모드 상태인 경우 모터제어기가 수신한 값이 순시 위치명령값임. 0 : SX 명령어 첫번째 파라미터의 비트 25가 1 일때 (CAN 통신을 이용한 1ms 동기제어모드) 모터제어기가 위치제어모드 상태인 경우 모터제어기가 수신한 값이 목표위치명령값이며, 설정된 가감속도에 따라서 순시 위치명령값이 변화됨. 2013년 11월 02일 S/W 버전부터 적용됨.</p> <p>bit30 : 1 : RS485/CAN 통신이 일정시간 이상 두절될 때 강제정지를 하지 않음. 0 : 강제정지함. 이때 강제정시 감속도는 So 명령값에 따름. RC 명령어 참조. 2013년 10월 23일 S/W 버전부터 적용됨. 전원투입시에 이 값이 0이면 idle 모드로 진입하며 RC/Rc/SX 명령어만 처리함. Rc 명령어 참조. 2013년 11월 05일 S/W 버전부터 적용됨.</p>

명령어	구분	설명
	operation	<p>bit31 : 1 : RS485통신에서 통신 패킷 마지막바이트에 checksum을 추가하지 않음. (통상값) 0 : RS485통신에서 통신 패킷 마지막바이트에 checksum을 추가함. 이전의 통신패킷에다 마지막에 checksum 한바이트를 추가하여야하며, 리턴값도 checksum 한바이트가 추가되어 호스트로 전송됨. checksum은 주소바이트를 포함한 모든 데이터의 합을 logical inverse한 값임. 기존 명령 중에서 p/Pa/Pb/Pp/Pq 명령에 이미 checksum 바이트가 있지만, checksum 한바이트가 더 추가됨. 2013년 10월 17일 S/W 버전부터 적용됨.</p>
	return value	<p>SXHLHLHLHL,HLHLHLHL; SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (2013년 11월 22일 S/W 버전부터는)</p>
	comment	<p>bit 0~7의 기본값은 모든 비트가 1인 경우이며, 이때 해당 리밋센서의 입력전압이 low이면 리밋센서가 ON 된 것으로 판단함. 이러한 설정의 영향을 받는 경우는 리밋센서를 위치리밋센서로 사용하는 경우와 리밋센서입력으로 비상정지를 하는 경우등 리밋센서를 사용하는 모든 경우가 영향을 받음. 2012년 5월 28일 S/W 버전부터 적용됨.</p> <p>bit 8의 기본값은 1이며 TMS320F28334/5를 사용한 H/W 버전으로서 2012년 6월 12일 S/W 버전부터, 이 비트가 0으로 설정되면 펄스주기에 의한 속도 계산값이 03/05번 제어모드에서 속도제어시에 사용되며 이 비트가 1이거나 TMS320F2810/11/12에서는 03/05번 제어모드에서 속도제어시에 무조건 speedx_RPM_10을 사용함. 펄스주기에 의한 속도계산을 사용하려면 엔코더가 광학식인 것을 추천하며, Hall IC를 사용한 경우에는 펄스주기값의 fluctuation이 매우 심할수있음.</p> <p>bit10/9의 기본값은 3이며 2012년 10월 06일 S/W 버전부터 유효함. bit11의 기본값은 1이며 2012년 10월 08일 S/W 버전부터 유효함. bit12의 기본값은 1이며 2012년 10월 11일 S/W 버전부터 유효함. bit13의 기본값은 1이며 2012년 10월 16일 S/W 버전부터 유효함. CAN이나 RS485를 통하여 과도한 명령어 데이터가 전송되어 오는 경우는 사용하지 말고, 통신 테스트용으로만 사용할 것.</p> <p>bit15/14의 기본값은 3이며 2012년 10월 16일 S/W 버전부터 유효함. 이 값은 EsA55A;에 의한 저장후 전원을 재투입하였을때 적용되므로, 이 값을 바꾸는 경우는 다른 작업을 하지말고 EsA55A;를 실행한 다음 RrA55A;를 사용하여 재부트 한 후에 나머지 설정을 진행해야함.</p> <p>bit19/18/17/16의 기본값은 F이며 1 ~ E 의 경우는 2013년 2월 14일 S/W 버전부터 적용됨. bit20의 기본값은 1이며 0인 경우 2013년 3월 27일 S/W 버전부터 유효함. bit21의 기본값은 1이며 0인 경우 2013년 4월 03일 S/W 버전부터 유효함.</p>

명령어	구분	설명
	comment	<p>bit22의 기본값은 1이며 0인 경우 2013년 4월 11일 S/W 버전부터 유효함. bit23의 기본값은 1이며 0인 경우 2013년 4월 14일 S/W 버전부터 유효함. (기동전류 적음) bit22를 0으로 설정하여 advanced control을 적용하는 경우는 bit23을 0으로 설정하여 advanced control을 적용하는 경우보다 기동전류가 크며 제어오차가 0으로 빨리 수렴하는 특성이 있으나, p 명령등에 의한 연속위치이동을 실시할 때 위치명령의 갱신주기가 불균일할 경우는 좀더 심한 저크를 발생시킬수 있음. bit22/23이 모두 0인 경우는 bit22가 0인 경우의 작동을 실시함. 03번 제어모드(위치제어 루프안에 속도제어루프가 있는 제어모드)에서는 bit23을 1로 세트하여야 정지위치 근방에서 진동이 없음 (기동전류가 적음으로서 제어기능이 약간 손해보는 때문으로 보임). bit24의 기본값은 1이며 0인 경우 2013년 5월 8일 S/W 버전부터 유효함. bit25의 기본값은 1이며 0인 경우 2013년 5월 25일 S/W 버전부터 유효함. 단 bit23/22의 값과 AU/Au/AV/Av/Aa/Az 명령을 사용한 적절한 파라미터 설정이 필요함. bit22/23/25의 값이 바뀌는 경우에 작동모드를 00으로 초기화하여 모터의 구동을 못하게 함. bit22/23 중에서 한비트라도 0이거나, bit25가 0일 때는 bit8을 강제로 1로 설정하여 펄스주기에 의한 속도계산을 하지 않도록 하였음. bit26의 기본값은 1이며 0인 경우 2013년 10월 02일 S/W 버전부터 유효함. bit26 값이 0인 경우에 bit25/23/22 값이 111이면 약 시정수 3ms의 속도값을 사용하며, bit25=1 이면서 bit23/22 둘중 한개라도 0이면 1kHz observer로부터의 속도값을 사용하며, bit25=0이면 5kHz observer로부터의 속도값을 사용함. 이때 AU 명령에 의하여 observer 파라미터가 적절하게 설정되어 있어야함. bit27의 기본값은 1이며 0인 경우 2013년 11월 22일 S/W 버전부터 유효함. bit28의 기본값은 1이며 0인 경우 2013년 11월 22일 S/W 버전부터 유효함. (기존의 자동화버전 제어기는 이 비트를 0으로 설정하여야만 진동/노이즈를 저감하기 위한 01번 모드로의 전환이 실시됨) bit29의 기본값은 1이며 0인 경우 2013년 11월 02일 S/W 버전부터 유효함. bit30의 기본값은 1이며 0인 경우 2013년 10월 23일 S/W 버전부터 유효함. 모터가 정지중일때만 0으로 설정해야함. bit31의 기본값은 1이며 0인 경우 2013년 10월 17일 S/W 버전부터 유효함. ☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
<p>(2013년 11월 22일 S/W 버전부터 적용됨.)</p> <p>;SXHLHLHLHL,HLHLHLHL,HLHLHLHL; (+#^)</p> <p>;SX?: (+#^)</p>	operation	<p>1st long word : SXHLHLHLHL; 의 경우와 같음. 2nd long word : SXHLHLHLHL,HLHLHLHL; 의 경우와 같음. 3rd long word</p> <p>bit0 : 1 일때 Pa/Pb/Pp/Pq/p/Qb9 명령에서 모터2에 대한 데이터를 생략하지 않음. 0 일때 Pa/Pb/Pp/Pq/p/Qb9 명령에서 모터2에 대한 데이터를 생략함. 2013년 11월 22일 S/W 버전부터 적용됨. 2014년 2월 01일 S/W 버전부터는 모든 명령에 적용됨.</p> <p>bit1 : 1 일때 2채널 제어기에서 명령어 사용자 매뉴얼에 기술된 그대로 적용함. 0 일때 2채널 제어기에서 모터2에 대한 데이터를 모터1에 대한 데이터 다음에 모아서 위치하도록 파라미터 순서를 바꾸어 적용하고, 리턴값도 바꾼 순서대로 리턴됨. 단, 리턴값으로 되돌아 오는 값에서 추가로 붙여지는 파라미터에 한해서는 사용자 매뉴얼에 기술된 순서대로 리턴됨. 2014년 2월 02일 S/W 버전부터 적용됨.</p> <p>bit2 : 1 : CAN 통신을 사용할 때, 일정주기로 모터2의 상태값을 전송하지 않음. (통상값) 0 : CAN 통신을 사용할 때, 일정주기로 모터2의 상태값을 전송함. 전송되는 주소는 모터2의 경우 기본값 0xfc 임. 주기값과 주소는 EU 명령을 참조하고, 전송되는 데이터는 SX 명령어 다음에 부분에 있는 내용을 참조. SX 명령어 첫번째 파라미터의 bit25가 1 이거나 (1ms main loop의 동기화 경우), 국방로봇용 제어기이거나, CAN 통신모드가 아닌 경우는 적용되지 않음. 설정후 재부팅하여야 반영됨. 2013년 11월 25일 S/W 버전부터 적용됨. 전송되는 데이터의 내용은 Ev 명령어로 바꿀 수 있음 (2014년 5월 25일 S/W 버전부터).</p> <p>bit3 : 1 : CAN 통신으로 p 명령을 사용할 때, 리턴값을 보냄. (통상값) 0 : 보내지 않음. 2014년 1월 29일 S/W 버전부터 적용됨.</p> <p>bit4 : 1 : CAN 통신을 통하여 주소값 0xff로 broadcasting 명령어를 받았을 때, 명령어의 끝이 ';' 이라도 리턴값을 보내지 않음. (통상값) 0 : CAN 통신을 통하여 주소값 0xff로 broadcasting 명령어를 받았을 때, 명령어의 끝이 ';' 이면 리턴값을 보냄. 2014년 3월 09일 S/W 버전부터 적용됨.</p> <p>bit5 : 1 : 자동화버전에서 로타리스위치의 값이 모터제어기 주소가 되도록 하지 않음. 0 : 자동화버전에서 로타리스위치의 값이 모터제어기 주소가 됨. 2014년 3월 21일 S/W 버전부터 적용됨</p>

명령어	구분	설명
(2013년 11월 22 일 S/W 버전부터 적용됨.) ;SXHLHLHLHL,HLHL HLHL,HLHLHLHL; (+#^) ;SX?: (+#^) 	operation	bit7/6 : 11 : RS232C 통신에서 time over를 판단하는 시간을 길게 함. (기본값) 00 : RS232C 통신에서 time over를 판단하는 시간을 1/32로 짧게 함. 01 : RS232C 통신에서 time over를 판단하는 시간을 1/16로 짧게 함. 10 : RS232C 통신에서 time over를 판단하는 시간을 1/8로 짧게 함. 00/01/10의 경우는 2014년 6월 02일 S/W 버전부터 적용됨 bit8 : 1 : 원점센서로 ENCxZ 신호를 사용하지 않음. 0 : 원점센서로 ENCxZ 신호를 사용함. bit9 : 1 일때 ENC1Z 입력전압이 low 이면 active, 0 일때 high 이면active bit10 : 1 일때 ENC2Z 입력전압이 low 이면 active, 0 일때 high 이면active bit8/9/10은 2014년 11월 15일 S/W 버전부터 적용됨
	return value	SXHLHLHLHL,HLHLHLHL,HLHLHLHL;
	comment	bit7/6 의 값이 00/01/10의 경우는 RS232C 통신에서 time over를 검출하는데 시간이 많이 걸린다고 판단하 는 경우에 사용할 것. 기본값은 9600 baud rate에서 13초이상이면 time over를 판단함. baud rate가 높을수록 time over를 판단하는 시간이 짧음. bit7/6을 00으로 설정하여 1/32을 선택하였다면, 대략 391바이트 정도를 연속으로 입력 또는 출력하는 시 간이 경과하였을때 RS232C 통신에서 time over로 검출하게 됨.

SX 명령에 의한 작동모드의 설정

☞ TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335의 작동모드는 16/32비트의 operation_mode_SWITCH 라는 변수명으로 설정되는데, SX 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할 수 있음. 단, DIP 스위치가 있는 일부 초기 H/W 버전의 경우에 bit0/6/5/4는 DIP 스위치의 상태를 반영하고 있으며 나머지 비트는 EEPROM에 저장할 수 있음.

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335 (SX 명령어 첫번째 파라미터)								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	1=sequence 제어를 실행함. 0=sequence 제어를 실행안함.	1=한스텝씩 sequence 제어를 실행함. 0=연속 sequence 제어를 실행함.	1=LSxC입력단자의 전압에 따라서 속도명령 또는 전류명령을 설정함. 엔코더가 있는 스텝 모터의 경우는 위치명령도 엔코더에 의하여 설정됨.	1=속도제어시에 (가상)위치리미트센서를 만나면 정지하고, 반대방향으로만 속도값명령이 설정가능토록 함. 0=속도제어시에 (가상)위치리미트센서를 사용안함.	1=analog 센서 타입의 RJM_VER7에서 Hall sensor 파라미터를 매 출발시마다 계산함.	1=위치제어시의 (가상)위치리미트센서를 적용안함. 0=위치제어시에 (가상)위치리미트센서를 만나면 정지하고, 반대방향으로만 위치값 설정이 가능하게함.	1=속도제어중 정지시에 적분항을 0으로 처리함. (제일메디컬은 무조건 0으로 처리) digital Hall IC 만 사용하는 경우는 속도/위치제어에 모두 적용되며 필수 선택사항임.	1=dummy 작동모드 실행함. 0=dummy 작동모드 실행안함. (DC/BLDC 모터만 해당됨)
bit7 - bit0	1=LSA, LSB를 리미트센서로 사용함. 0=LSA, LSC를 리미트센서로 사용함.	2번 스위치 OFF(1)=RS485 또는 CAN ON(0)=ETHERCAT	3번 스위치 통신테스트모드 또는 외부표시모드	4번 스위치 통신테스트모드 또는 외부표시모드	1=모터 power를 계산함. 0=모터 power를 계산하지 않음.	1=모터권선온도 리미트기능 사용함. 0=모터권선온도 리미트기능 사용안함.	1=모터1과 모터2가 직선보간운동을 실시함. 0=모터1과 모터2가 직선보간운동을 실시하지 않음.	1번 스위치 OFF(1)=RS485 ON(0)=CAN

* LSxC 포트를 analog 입력으로 사용할때, bit10에 의한 위치리미트의 적용에서 발생하는 문제를 해결하는데 bit7을 사용함.

SX 명령에 의한 작동모드의 설정

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335 (SX 명령어 첫번째 파라미터)								
	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
bit31 - bit24	1=데모작동을 실시함. 0=데모작동을 멈춤.	1=모터작동 불가능 fault가 발생하였을때 모터제어를 자동으로 위치제어모드로 재부팅함. (analog 센서타입의 BLDC 모터에는 적용되지 않음) 0=재부팅 안함.	11=BLDC 모터제어에서 BLOCK COMMUTATION 방식을 사용함. 10=BLDC 모터제어에서 pure sine PWM 방식을 사용함. (수행시간 때문에 PWM 주파수를 80KHz 이내에서만 사용할 것) 01=BLDC 모터제어에서 modified-SVPWM 방식을 사용함. 00=BLDC 모터제어에서 space vector PWM 방식을 사용함.		1 = 위치제어에서 PA/PB/Pa 명령수행시에 감속에 따른 역기전력을 감소시키기 위하여, 감속도를 제한하는 기능을 사용함. 0=사용하지 않음.	1 = 속도제어에서 감속에 따른 역기전력을 감소시키기 위하여, 감속도를 제한하는 기능을 사용하지 않음. 0=사용함.	1=1ms main loop의 기동을 CAN 데이터가 수신될때 실시함. (DC 모터로서 CAN 통신에서만 가능) 0=1ms main loop의 기동내부타이머로 실시함.	1=위치제어에서 설정속도가 과도하여 위치제어 오차가 매우 커질 때 발생하는 문제를 처리하지 않음. 0=처리함.
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
bit23 - bit16	1=digital Hall만을 사용하는 BLDC 모터에서 05번 속도제어 중 정지후 출발할때 위치값을 초기화하는 기능을 사용안함. 2015년 1월 10일 S/W 버전부터 적용됨. 0=사용함.	1=위치리미트초과 오류검출에 상위치리미만을 사용함. 0=위치리미트초과 오류검출에 위치리미트센서만을 사용함.	1=저속에서 전류스위칭 소음을 줄이기 위하여, stepping motor의 전류제어게인을 모터 속도에 따라서 가변하는 기능을 사용하지 않음. 0=사용함.	1=stepping 모터2의 lead angle 제어를 실시함. (반드시 엔코더가 부착되어 있어야 함) 0=stepping 모터2의 lead angle 제어를 실시안함.	1=stepping 모터1의 lead angle 제어를 실시함. (반드시 엔코더가 부착되어 있어야 함) 0=stepping 모터1의 lead angle 제어를 실시안함.	1=stepping 모터의 승압회로를 사용함. 0=stepping 모터의 승압회로를 사용안함.	1=엔코더2의 A/B상 신호를 서로 바꿈. 0=엔코더2의 A/B상 신호를 서로 안바꿈.	1=엔코더1의 A/B상 신호를 서로 바꿈. 0=엔코더1의 A/B상 신호를 서로 안바꿈.

SX 명령에 의한 작동모드의 설정

☞ 2012년 5월 28일 S/W 버전부터 사용이 가능한 operation_mode_SWITCH_more 라는 변수명으로 설정되는데, SX 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할 수 있음. (SX 명령어로 설정하는 파라미터가 기존의 1개에서 2개로 늘었음)

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335 (SX 명령어 두번째 파라미터)								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8	11=CAN 11-bit 주소모드로서 주소로 0-255만 사용. 10=CAN 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용. 01=CAN 29-bit 주소모드로서 주소로 0-65535만 사용, 2012년 10월 16일부터 적용. 00=CAN 11-bit 주소모드로서 주소로 0-255만 사용.		0일때 CAN이나 RS485 통신 포트를 통하여 입력된 명령을 RS232C 포트로 전송함. 1일때 상기 작동을 하지 않음.	0 일때 1ms당의 펄스를 카운트하면서 1ms unit timer를 사용함 (TMS320F28334/5 H/W 버전인 경우로서 엔코더를 사용하는 경우에만 적용 가능함) 1 일때 1ms당의 펄스를 카운트하면서 1ms unit timer를 사용하지 않음.	digital Hall IC 만을 사용하는 BLDC 모터를 제외한 경우에만 해당되며, bit11이 0인 경우 05/06번 속도제어에서 속도제어루프에 저속에서 속도제어게인을 낮추는 기능을 적용함. (Wd 명령 참조)	bit10:bit9의 값이 0/1/2/3 일때 03번 위치제어에서 위치제어 루프의 sample time을 10/5/2/1ms로 설정함.		1 일때 모터의 속도제어시에 speedx_RPM_10을 사용하며, 0 일때 speedx_RPM_combined_10(TMS320F28334/5 H/W 버전에서만 펄스주기 이용 방식의 속도값이고 그외는 카운트방식의 속도값)을 사용함
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	1 일때 모터2의 리미트센서 D 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터1의 리미트센서 D 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터2의 리미트센서 C 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터2의 리미트센서 B 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터2의 리미트센서 A 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터1의 리미트센서 C 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터1의 리미트센서 B 입력전압이 low 이면 active, 0 일때 high active	1 일때 모터1의 리미트센서 A 입력전압이 low 이면 active, 0 일때 high active

SX 명령에 의한 작동모드의 설정

☞ 2012년 5월 28일 S/W 버전부터 사용이 가능한 operation_mode_SWITCH_more 라는 변수명으로 설정되는데, SX 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할 수 있음. (SX 명령어로 설정하는 파라미터가 기존의 1개에서 2개로 늘었음)

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335 (SX 명령어 두번째 파라미터)								
	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
bit31 - bit24	1 : RS485통신에서 통신 패킷 마지막 바이트에 checksum을 추가하지 않음. 0 : 추가함. 2013년 10월 17일 S/W 버전부터 적용됨.	1 : RS485/CAN 통신을 하던 중에 일정 시간 이상 통신이 두절되면 강제정지를 하지 않음. 0 : 정지함. 2013년 10월 23일 S/W 버전부터 적용됨	1 : SX 명령어 첫번째 파라미터의 비트 25가 1 일때 모터제어기에서 수신된 값이 순시 위치명령값임. 0 : 목표 위치명령값임. 2013년 11월 02일 S/W 버전부터 적용됨	1 : 02번 위치제어모드에서 정지시에는 01번 모드로 전환하는 기능을 사용안함. 0 : 사용하여 정지에 진동을 저감함. 2013년 11월 22일 S/W 버전부터 적용됨	1 : CAN 통신을 사용할 때, 일정주기로 모터의 상태값을 전송하지 않음. 0 : 전송함. 2013년 11월 22일 S/W 버전부터 적용됨	1 : advanced controller를 사용하지 않는 위치제어에서 위치오차 미분값 대신에 속도값을 feedback 하는 기능을 사용 안함. 0 : 사용함. (제한 조건 있음) 2013년 10월 04일 S/W 버전부터 적용됨.	1 : 1KHz로 observer 및 03/0A 작동모드의 속도제어를 실행함. 0: 5KHz로 실행함. 2013년 5월 25일 S/W 버전부터 적용됨.	1 : 모터의 제어 출력값계산 및 갱신시점을 맞추는 동작을 하지 않음. 0 : 시점을 맞춤. 2013년 5월 8일 S/W 버전부터 적용됨.
bit23 - bit16	1 : 03/05/0A번 작동모드에서 속도오차대신 속도를 feedback 하는 advanced controller를 적용하지 않음. 0 : 적용함. 2013년 4월 14일 S/W 버전부터 적용됨. (기동전류 적용)	1 : 02/03/05/0A번 작동모드에서 위치/속도 오차를 feedback하는 advanced controller를 적용하지 않음. 0 : 적용함. 2013년 4월 11일 S/W 버전부터 적용됨. (기동전류 큼)	1 : Qx monitoring에서 전류치를 0.01A 단위로 보여 주지 않음. 0 : Qx monitoring에서 전류치를 0.01A 단위로 보여 줌. 2013년 4월 03일 S/W 버전부터 적용됨.	1 : 속도제어에서 가속시 속도오차가 크면 속도 명령값을 수정함. 0 : 속도 명령값을 수정하지 않음. 2013년 3월 27일 S/W 버전부터 적용됨.	F : 외부표시모드는 SX 명령어 첫번째 파라미터의 bit4/5 의 설정에 따름. (2013년 2월 14일 이전 S/W 버전과 같음) 0 : 외부표시모드가 작동되지 않음. 1 : host로부터 DT 명령을 사용하여 외부표시내용을 받아서 표시하는 모드이며 표시내용이 자주 바뀌지 않는 경우에 사용함. 2 : host로부터 DT 명령을 사용하여 외부표시내용을 받아서 표시하는 모드이며 표시내용이 자주 바뀌는 경우에 사용함. 1 ~ E 의 경우는 2013년 2월 14일 S/W 버전부터 적용됨.			

bit25가 0일지라도 bit23/22이 00 이면 advanced 속도제어의 호출은 1KHz로 실시함. 단 bit25가 00이면 속도제어게인은 bit23/22의 값에 관계없이 5KHz에서의 값을 입력하여야함. bit23/22이 11이 아니더라도 Aa 명령어에 의하여 advanced controller가 activate 되어 있지 않으면 advanced controller가 적용되지 않음. bit25/23/22이 111 이면 observer를 계산하지 않음.

SX 명령에 의한 작동모드의 설정

☞ 2013년 11월 22일 S/W 버전부터 사용이 가능한 operation_mode_SWITCH_more2 라는 변수명으로 설정되는데, SX 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할 수 있음. (SX 명령어로 설정하는 파라미터가 기존의 2개에서 3개로 늘었음)

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335 (SX 명령어 세번째 파라미터)								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
bit15 - bit8						1 일때 ENC2Z 입력전압이 low 이면 active, 0 일때 high 이면 active 2014년 11월 15일 S/W 버전부터 적용됨	1 일때 ENC1Z 입력전압이 low 이면 active, 0 일때 high 이면 active 2014년 11월 15일 S/W 버전부터 적용됨	1 : 원점센서로 ENCxZ 신호를 사용하지 않음. 0 : 원점센서로 ENCxZ 신호를 사용함. 2014년 11월 15일 S/W 버전부터 적용됨
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7 - bit0	11 : RS232C 통신에서 time over를 판단하는 시간을 길게 함. 00,01,10 : RS232C 통신에서 time over를 판단하는 시간을 짧게 함. (00은 1/32, 01은 1/16, 10은 1/8) 2014년 6월 02일 S/W 버전부터 적용됨		1 : 자동화버전에서 로타리스위치의 값이 모터제어기 주소가 되도록 하지 않음. 0 : 자동화버전에서 로타리스위치의 값이 모터제어기 주소가 됨. 2014년 3월 21일 S/W 버전부터 적용됨	1 : CAN 통신을 통하여 주소값 0xff로 broadcasting 명령어를 받았을 때, 명령어의 끝이 ';'이라도 리턴값을 보내지 않음. 0 : 보냄. 2014년 3월 09일 S/W 버전부터 적용됨	1 : CAN 통신으로 p 명령을 사용할 때 리턴값을 보냄. 0 : 보내지 않음. 2014년 1월 29일 S/W 버전부터 적용됨	1 : CAN 통신을 사용할 때, 일정 주기로 모터2의 상태값을 전송하지 않음. 0 : 전송함. 2013년 11월 25일 S/W 버전부터 적용됨	1 : 2채널 제어기에서 명령어 매뉴얼에 기술된 그대로 적용함. 0 : 2채널 제어기에서 모터2에 대한 데이터를 모아서 위치하도록 파라미터 순서를 바꿈. 2014년 2월 02일 S/W 버전부터 적용됨.	1 : Pa/Pb/Pp/Pq/p/Qb9 명령에서 모터2에 대한 데이터를 생략하지 않음. 0 : 생략함. 2013년 11월 22일 S/W 버전부터 적용됨. 2014년 1월 31일 S/W 버전부터는 모든 명령에 적용됨.

SX 명령에 의한 작동모드의 설정

☞ 2013년 11월 22일 S/W 버전부터 사용이 가능한 operation_mode_SWITCH_more2 라는 변수명으로 설정되는데, SX 명령어로 설정하고 EsA55A; 명령어로 EEPROM에 저장할 수 있음. (SX 명령어로 설정하는 파라미터가 기존의 2개에서 3개로 늘었음)

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335 (SX 명령어 세번째 파라미터)								
	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
bit31 - bit24	BLDC_INDUCTION_DC 0=BLDC, 1=INDUCTION, 2=DC, 3=STEP				ENC_dHALL_aHALL_RESOLVER_POT 0=ENC, 1=dHALL, 2=aHALL, 3= RESOLVER, 4=POT, 5=PWM_TYPE_POT, 6=SPI_TYPE_POT1, 7=SPI_TYPE_POT2, 8=LOW_RESOLUTION_ENCODER(DC), 9=absolute ENC(Netzer 19bit), 10=absolute ENC(Heidenheim 26bit)			
	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
bit23 - bit16	NO_HALL_BUT_HOME_SENSOR 0 = 통상값, 1 = 상대각 엔코더는 있고, Hall IC는 없고, 대신하여 홀센서를 사용하는 경우, 2 = 상대각 엔코더는 있고, Hall IC와 홀센서 모두 사용하지 않는 경우 (특정방향으로 회전자를 위치시키고, 엔코더를 초기화함), 3 = 상대각 엔코더는 있고, Hall IC와 홀센서 모두 사용하지 않으나 모터 한바퀴내의 절대각 센서가 있는 경우				0=encoder A/B 상 입력 1=resolver 입력 (모터제어기내에서 엔코더신호로 변환함)			

SX 명령어 두번째 파라미터의 bit27이 0일 경우 상위제어기로의 전송값

- ☞ 상위제어기로의 전송 주소 : 기본값 0xfd (모터1의 상태값), 상위제어기로의 전송 데이터수 : 8 byte, 1 packet
 - byte1 : 상위 4비트 : 모터1의 작동모드 (SM 명령어로 설정하는 작동모드의 하위 바이트)
 - 비트3 : 모터1의 MOTOR1_MAX_CURRENT_fault_flg | 모터1의 current1_over_flg
 - 비트2 : MAX_VOLTAGE_fault_flg, 비트1 : LOW_VOLTAGE_fault_flg
 - 비트0 : 모터1의 MOTOR1_gate_kill_fault_flg
 - byte2 : 모터1의 원점으로부터의 위치변화값 하위 8비트
 - byte3 : 모터1의 원점으로부터의 위치변화값 중간 8비트
 - byte4 : 모터1의 원점으로부터의 위치변화값 상위 8비트 (최상위비트가 1이면 음수임)
 - byte5 : 모터1의 상전류크기 하위 8비트
 - byte6 : 모터1의 상전류크기 상위 8비트
 - byte7 : 모터1의 LS1C 포트의 전압값을 필터링한 값의 하위 8비트
 - byte8 : 모터1의 LS1C 포트의 전압값을 필터링한 값의 상위 8비트
- 전송되는 데이터의 내용은 Eu 명령어로 바꿀 수 있음 (2014년 5월 25일 S/W 버전부터).

SX 명령어 두번째 파라미터의 bit27 또는 세번째 파라미터의 bit2가 0일 경우 상위제어기로의 전송값

☞ 상위제어기로의 전송 주소 : 기본값 0xfc (모터2의 상태값), 상위제어기로의 전송 데이터수 : 8 byte, 1 packet

byte1 : 상위 4비트 : 모터2의 작동모드 (SM 명령어로 설정하는 작동모드의 상위 바이트)

비트3 : 모터2의 MOTOR2_MAX_CURRENT_fault_flg | 모터2의 current2_over_flg

비트2 : MAX_VOLTAGE_fault_flg, 비트1 : LOW_VOLTAGE_fault_flg

비트0 : 모터2의 MOTOR2_gate_kill_fault_flg

byte2 : 모터2의 원점으로부터의 위치변화값 하위 8비트

byte3 : 모터2의 원점으로부터의 위치변화값 중간 8비트

byte4 : 모터2의 원점으로부터의 위치변화값 상위 8비트 (최상위비트가 1이면 음수임)

byte5 : 모터2의 상전류크기 하위 8비트

byte6 : 모터2의 상전류크기 상위 8비트

byte7 : 모터2의 LS2C 포트의 전압값 하위 8비트

byte8 : 모터2의 LS2C 포트의 전압값 상위 8비트

전송되는 데이터의 내용은 Ev 명령어로 바꿀 수 있음 (2014년 5월 25일 S/W 버전부터).

☞ 만일 1채널 제어기라면

byte1 : fault_status 변수의 상위 8비트

byte2 : 모터1의 원점으로부터의 순간위치명령값 하위 8비트

byte3 : 모터1의 원점으로부터의 순간위치명령값 중간 8비트

byte4 : 모터1의 원점으로부터의 순간위치명령값 상위 8비트 (최상위비트가 1이면 음수임)

byte5 : 모터1의 상전류명령크기 하위 8비트

byte6 : 모터1의 상전류명령크기 상위 8비트

byte7 : 모터1의 속도값(RPM) 하위 8비트

byte8 : 모터1의 속도값(RPM) 상위 8비트

전송되는 데이터의 내용은 Ev 명령어로 바꿀 수 있음 (2014년 5월 25일 S/W 버전부터).

SX 명령어 첫번째 파라미터의 bit25가 1일 경우 1ms당 상위제어기로의 전송값

☞ SX 명령 첫번째 파라미터의 bit25가 1일 경우

상위제어기로의 전송 주소 : SA 명령어로 설정하는 모터제어기의 주소의 하위바이트와 같음.

상위제어기로의 전송 데이터수 : 8 byte, 1 packet

byte1 : 상위 4비트 : 모터1의 작동모드 (SM 명령어로 설정하는 작동모드의 하위 바이트)

비트3 : 모터1의 MOTOR1_MAX_CURRENT_fault_flg | 모터1의 current1_over_flg

비트2 : 모터1의 MAX_VOLTAGE_fault_flg

비트1 : 모터1의 LOW_VOLTAGE_fault_flg

비트0 : 모터1의 MOTOR1_gate_kill_fault_flg

byte2 : 모터1의 원점으로부터의 위치변화값 하위 8비트 (작동모드가 위치제어모드일 경우)

: 모터1의 속도값X10 하위 8비트 (작동모드가 속도제어모드일 경우)

byte3 : 모터1의 원점으로부터의 위치변화값 중간 8비트 (작동모드가 위치제어모드일 경우)

: 모터1의 속도값X10 중간 8비트 (작동모드가 속도제어모드일 경우)

byte4 : 모터1의 원점으로부터의 위치변화값 상위 8비트 (작동모드가 위치제어모드일 경우)

: 모터1의 속도값X10 상위 8비트 (작동모드가 속도제어모드일 경우)

: 최상위비트가 1이면 음수임

byte5 : 모터1의 상전류크기 하위 8비트

byte6 : 모터1의 상전류크기 상위 8비트

byte7 : 모터1의 LS1C 포트의 전압값을 필터링한 값의 하위 8비트

byte8 : 모터1의 LS1C 포트의 전압값을 필터링한 값의 상위 8비트

SX 명령에 의한 두번째 파라미터 bit25/23/22 설정과 advanced controller의 작동의 상관 관계

	SX 명령에 의한 두번째 파라미터의 bit25	SX 명령에 의한 두번째 파라미터의 bit23	SX 명령에 의한 두번째 파라미터의 bit22	Aa 명령에 의한 파라미터 설정값	observer	적용 제어알고리즘
case 1	1	1	1	XXXX	계산 안함	기존의 PID 제어방식 (01/02/03/04/05/06/0A/0D/0E/0F)
case 2	1	bit23/bit22 중 한개라도 0		!=A55A	1KHz로 계산함	기존의 PID 제어방식 (01/02/03/04/05/06/0A/0D/0E/0F)
case 3	1	1	0	=A55A	1KHz로 계산함	02/03/04/05/0A/0D/0F 번 작동모드인 경우, 위치/속도 오차를 feedback하는 1KHz advanced controller를 적용함. (기동전류가 크나 제어성능은 더 좋음) 01/06/0E 번 작동모드에서는 기존의 PID 제어방식을 적용함.
case 4	1	0	1	=A55A	1KHz로 계산함	case 3과 같으나 속도제어 루프를 사용하는 경우, 속도 오차대신 속도값을 feedback함. (기동전류 적음)
case 5	1	0	0	=A55A	1KHz로 계산함	case 3과 같음. (1KHz advanced controller)
case 6	0	1	1	XXXX	5KHz로 계산함	기존의 PID 제어방식 (01/02/03/04/05/06/0A/0D/0E/0F)
case 7	0	bit23/bit22 중 한개라도 0		!=A55A	5KHz로 계산함	기존의 PID 제어방식 (01/02/03/04/05/06/0A/0D/0E/0F)
case 8	0	1	0	=A55A	5KHz로 계산함	02/03/04/05/0A/0D/0F 번 작동모드인 경우, 위치/속도 오차를 feedback하는 advanced controller를 적용하되, 위치 제어에서 inner 속도제어를 사용하는 모드에서는 속도제어를 5KHz로 실시함. (기동전류가 크나 제어성능은 더 좋음) 01/06/0E 번 작동모드에서는 기존의 PID 제어방식을 적용함.
case 9	0	0	1	=A55A	5KHz로 계산함	case 8과 같으나 속도제어 루프를 사용하는 경우, 속도 오차대신 속도값을 feedback함. (기동전류 적음)
case 10	0	0	0	=A55A	5KHz로 계산함	case 3과 같음. (1KHz advanced controller)

명령어	구분	설명
(저가형 1채널 버전이 아닌 경우) ;SEA55A,HLHL,HLHL,HLHL,HLHL; (+#^%) ;SE?; (+#^%)	operation	<p>모터1회전당 펄스수와 POLE PAIR 갯수를 설정함.</p> <p>1st word : 모터1의 1회전당 펄스수 [1-32767] (MOTOR_TYPE에 따름)</p> <p>2nd word : 모터2의 1회전당 펄스수 [1-32767] (MOTOR_TYPE에 따름)</p> <p>3rd word : MOTOR1_NUMBER_of_POLE_PAIR [1-20] (MOTOR_TYPE에 따름)</p> <p>4th word : MOTOR2_NUMBER_of_POLE_PAIR [1-20] (MOTOR_TYPE에 따름)</p>
	return value	<p>SEHLHL,HLHL,HLHL,HLHL; (2012년 1월 24일 이전 S/W 버전)</p> <p>SEHLHL,HLHL,HLHL,HLHL,HLHL; (2012년 1월 24일 S/W 버전부터)</p> <p>SEHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (2013년 5월 19일 S/W 버전부터)</p> <p>5th word : PWM_PERIOD (단위 크기는 1/120 uS, 2012년 1월 24일 S/W 버전부터 적용됨)</p> <p>6th word : SPEED_SCALE_FACTOR (2013년 5월 19일 S/W 버전부터 적용됨)</p>
	comment	<p>⚡ 4채배된 엔코더 펄스수로 정의 됨.</p> <p>⚡ 기본값으로 MAXON RE-max 29/24V/22W DC 모터는 1024, FAULHARBER 3257CR 24V/83W DC 모터는 2048, MAXON EC 45 flat 24V/50W(△) BLDC 모터는 4096, MAXON EC-powermax 30 24V/100W(△) BLDC 모터는 4000, MAXON EC 45 24V/250W(△) BLDC 모터는 2000, FAULHARBER 4490 24V/200W(△) BLDC 모터는 2000, MAXON EC 90 flat 24V/90W BLDC 모터는 2000임.</p> <p>⚡ MOTORx_NUMBER_of_POLE_PAIR는 DC 모터의 경우는 무조건 10이며, BLDC 모터의 경우 pole pair의 수를 의미하며, STEP 모터의 경우는 90° /step각 을 의미함.</p> <p>⚡ 이 명령은 EDA55A;를 실행한 이후이거나, ST 명령에 의한 MOTOR_TYPE 설정이 0xffff 이거나, 또는 모터를 enable 하지 않고 작동모드도 0일때에만 적용이 가능함.</p> <p>⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
(저가형 1채널 버전인 경우) ;SEA55A,HLHL,HLHLHLHL; (+#^) ;SE?; (+#^)	operation	모터1회전당 펄스수와 POLE PAIR 갯수를 설정함. (저가형 1채널 버전인 경우) 1st word : 모터1의 1회전당 펄스수 [1-32767*65536] (MOTOR_TYPE에 따름) 2nd long word : MOTOR1_NUMBER_of_POLE_PAIR [1-20] (MOTOR_TYPE에 따름)
	return value	SEHLHL,HLHLHLHL,HLHL,HLHL; 3rd word : PWM_PERIOD (단위 크기는 1/90 uS) 4th word : SPEED_SCALE_FACTOR
	comment	<ul style="list-style-type: none"> ⚙ 4채배된 엔코더 펄스수로 정의 됨. ⚙ 기본값으로 MAXON RE-max 29/24V/22W DC 모터는 1024, FAULHARBER 3257CR 24V/83W DC 모터는 2048, MAXON EC 45 flat 24V/50W(△) BLDC 모터는 4096, MAXON EC-powermax 30 24V/100W(△) BLDC 모터는 4000, MAXON EC 45 24V/250W(△) BLDC 모터는 2000, FAULHARBER 4490 24V/200W(△) BLDC 모터는 2000, MAXON EC 90 flat 24V/90W BLDC 모터는 2000임. ⚙ MOTORx_NUMBER_of_POLE_PAIR는 DC 모터의 경우는 무조건 1이며, BLDC 모터의 경우 pole pair의 수를 의미하며, STEP 모터의 경우는 90° /step각 을 의미함. ⚙ 이 명령은 EDA55A;를 실행한 이후이거나, ST 명령에 의한 MOTOR_TYPE 설정이 0xffff 이거나, 또는 모터를 enable 하지 않고 작동모드도 0일때에만 적용이 가능함. ⚙ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;SGA55A, dddd1, dd dd2, dddd3, dddd4; (+#^%) ;SG?; (+#^%)	operation	<p>감속비와 최대회전속도를 설정함.</p> <p>1st word : 모터1의 Gear Ratio [1 - 2000] (MOTOR_TYPE에 따름)</p> <p>2nd word : 모터2의 Gear Ratio [1 - 2000] (MOTOR_TYPE에 따름)</p> <p>3rd word : 모터1의 최대속도 [10 - 32000/60000] (rpm, MOTOR_TYPE에 따름)</p> <p>4th word : 모터2의 최대속도 [10 - 32000/60000] (rpm, MOTOR_TYPE에 따름)</p>
	return value	SGdddd1, dddd2, dddd3, dddd4;
	comment	<p>⚡ 이 명령은 EDA55A;를 실행한 이후 또는 모터 type이 0xfff 일때에만 적용이 가능함. 그러나 2011년 10월 16일 S/W 버전부터는 PE 명령에 의한 작동개시를 하지 않고, 작동모드도 0일때에는 실행이 가능함.</p> <p>⚡ Gear Ratio는 모터에 연결된 감속기의 감속비를 의미하는데, 모터 출력축에 변위센서가 설치된 경우에는 그 값이 부정확하여도 큰 문제가 되지는 않으나, 감속기 출력단에 변위센서가 장착된 경우에는 정확한 값을 설정하여야함. linear 모터인 경우에는 감속기가 없는 구조이기는 하나, 직선 구동부 레일에 장착된 영구자석 N-S 극 pair의 총개수의 1/2 값으로 Gear Ratio 값을 설정하면 됨.</p> <p>⚡ 최대속도값은 2012년 12월 29일 이전 S/W 버전에서는 1000부터, 그때 버전 부터는 10부터 설정가능.</p> <p>⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;Sgiiii1,iiii2,iiii3,iiii4; (+#^) ;Sg?: (+#^)	operation	<p>위치제어 또는 속도제어에서 phase lead 또는 phase lag compensator의 파라미터를 설정함. (이 경우는 미분제어 게인이 무시됨)</p> <p>1st integer : COMPENSATOR_CENTER_FREQ1 [1 - 63] (50 Hz) 2nd integer : COMPENSATOR_ALPHA1 [-100 - 100] (0) 3rd integer : COMPENSATOR_CENTER_FREQ2 [1 - 63] (50 Hz) 4th integer : COMPENSATOR_ALPHA2 [-100 - 100] (0)</p>
	return value	Sgiiii1,iiii2,iiii3,iiii4;
	comment	<p>☞ COMPENSATOR_CENTER_FREQx는 대략 Hz 단위의 compensator 중심주파수를 의미함.</p> <p>☞ COMPENSATOR_ALPHAx가 양수일 때는 phase lead compensator이며, 음수일 때는 phase lag compensator이고 0 일 때는 compensator를 적용하지 않음.</p> <p>☞ COMPENSATOR_ALPHAx가 양수일 때(phase lead compensator 일때)는 PI 만을 적용하고 PID를 적용하지 않음.</p> <p>☞ COMPENSATOR_ALPHAx는 gain_increase(COMPENSATOR_ALPHAx가 양수일 때) 또는 gain_reduction(COMPENSATOR_ALPHAx가 음수일 때)을 설정하는 값임.</p> <p>☞ COMPENSATOR_ALPHAx에 따라서 phase_lead/phase_lag 각의 크기도 값이 변화되며 COMPENSATOR_ALPHAx가 10/8/6/4/2일 때 약 55도/52도/46도/37도/20도의 phase_lead/phase_lag 각이 설정됨.</p> <p>☞ F2811/F28334 에서만 가능한 명령임.</p> <p>☞ 2008년 7월 23일 S/W 버전부터 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p> <p>☞ 2008년 10월 5일 S/W 버전부터 COMPENSATOR_CENTER_FREQx 대신 COMPENSATOR_ALPHAx 값이 양수 또는 음수로 설정이 되도록 바꾸었음.</p>

명령어	구분	설명
;SL#####; (+#%) ;SL?; (+#%)	operation	<p>위치제어모드에서 위치의 한계값을 설정함.</p> <p>1st long integer : 모터1 설정위치의 상한값 (감속기출력 기준으로 +350도)</p> <p>2nd long integer : 모터1 설정위치의 하한값 (감속기출력 기준으로 -350도)</p> <p>3rd long integer : 모터2 설정위치의 상한값 (감속기출력 기준으로 +350도)</p> <p>4th long integer : 모터2 설정위치의 하한값 (감속기출력 기준으로 -350도)</p>
	return value	SL#####;
	comment	<ul style="list-style-type: none"> ☞ 상하한설정위치값은 모터1의 경우는 모터1의 현재위치값인 position1을 포함하는 범위이어야만 유효하며, 모터2의 경우 모터2의 현재위치값인 position2을 포함하는 범위에서이어야만 유효. ☞ 상한값이 하한값보다는 커야만 유효함. 원점위치는 5,000,000임. ☞ 각종 위치명령값을 설정할때에만 크기를 제한하는 용도로 쓰임. ☞ F2811/F28334의 경우 SX 명령어로 설정하는 bit10 값이 0이면 위치설정값의 제한이 실제로 가동됨. ☞ 2014년 9월 20일 이전의 S/W 버전에서 위치리밋값이 0보다 작은 경우에는 위치리밋값 설정과 흡동작 처리에 문제가 있었음. ☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;SI#####; (+#^) ;SI?; (+#^)	operation	<p>흡찾기 동작에서 위치의 한계값을 설정함.</p> <p>1st long integer : 모터1 흡찾기 동작의 상한값 (감속기출력 기준으로 +350도)</p> <p>2nd long integer : 모터1 흡찾기 동작의 하한값 (감속기출력 기준으로 -350도)</p> <p>3rd long integer : 모터2 흡찾기 동작의 상한값 (감속기출력 기준으로 +350도)</p> <p>4th long integer : 모터2 흡찾기 동작의 하한값 (감속기출력 기준으로 -350도)</p>
	return value	SI#####;
	comment	<ul style="list-style-type: none"> ☞ 상하한설정위치값은 모터1의 경우는 모터1의 현재위치값인 position1을 포함하는 범위이어야만 유효하며, 모터2의 경우 모터2의 현재위치값인 position2을 포함하는 범위에서이어야만 유효. ☞ 상한값이 하한값보다는 커야만 유효함. 원점위치는 5,000,000임. ☞ 흡찾기 동작을 할 때에만 크기를 제한하는 용도로 쓰임. ☞ 2014년 9월 20일 이전의 S/W 버전에서 위치리밋값이 0보다 작은 경우에는 위치리밋값 설정과 흡동작 처리에 문제가 있었음. ☞ EEPROM에 저장되지 않으며 기본값은 SL 명령에 의하여 설정된 값임.

명령어	구분	설명
;SSdddd1,dddd2; (+#^%) ;SS?; (+#^%)	operation	위치제어모드에서 Point-to-Point 이동시 이동속도값을 설정함. 1st word : 모터1 속도명령값 [1 ~ 32000/60000] (1000/60 rpm) 2nd word : 모터2 속도명령값 [1 ~ 32000/60000] (1000/60 rpm)
	return value	SSdddd1,dddd2;
	comment	<ul style="list-style-type: none"> ☞ 단위는 rpm이며 RJM_VER6의 기본값은 60rpm, 나머지는 1000rpm임. ☞ 2014년 2월 24일 이전 S/W 버전에서는 자동부팅 또는 데모 작동시에 속도를 바꾸고, EEPROM에 저장하는 경우 속도명령값이 조금씩 변하는 문제가 있음. ☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Ssdddd1,dddd2; (+#^%) ;Ss?; (+#^%)	operation	PA, PB, Pa 명령에 의한 위치구동시의 가속/감속 기간을 설정함. 1st word : 모터1의 가속/감속기간 [1 ~ 10000] (250 or 1000 ms) 2nd word : 모터2의 가속/감속기간 [1 ~ 10000] (250 or 1000 ms)
	return value	Ssdddd1,dddd2;
	comment	<ul style="list-style-type: none"> ☞ 단위는 ms임. ☞ 기본값은 STEP RJM_VER5의 경우 250ms이며 나머지의 경우는 1000ms임. ☞ SS 명령어로 정속구간에서의 속도를 먼저 설정해야만 Ss 명령어에 의한 가감속도 설정이 정상적으로 이루어짐. Sa 명령어로도 가감속도 설정이 가능한데 이때는 SS 명령어에 의한 정속구간에서의 속도와 무관하게 설정이 가능함. ☞ 엔코더의 펄스수가 적거나 가속/감속 기간이 큰 경우 Ss 명령어로 설정된 값과 실제의 가속/감속기간의 오차가 있을수 있음. 이 경우에는 펄스수가 작은 경우에도 적용할 수 있는 firmware를 사용하여야함. ☞ RJM_VER5의 경우는 500 이 최대 설정값임. ☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Sdddd1,dddd2,ddddd3,dddd4; (+#^) ;Sd?; (+#^)	operation	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 위하여 가감속률을 제한하는 파라미터를 설정함. 1st word : 모터1의 가속도 저감률 [0 ~ 65535] (100 rpm/100ms/V or 1 rpm/ms/V) 2nd word : 모터1의 감속도 저감률 [0 ~ 65535] (400 rpm/100ms/V or 4 rpm/ms/V) 3rd word : 모터2의 가속도 저감률 [0 ~ 65535] (100 rpm/100ms/V or 1 rpm/ms/V) 4th word : 모터2의 감속도 저감률 [0 ~ 65535] (400 rpm/100ms/V or 4 rpm/ms/V)
	return value	Sdddd1,dddd2,dddd3,dddd4;
	comment	<ul style="list-style-type: none"> ⚙️ RJM_VER7에서는 단위가 rpm/ms/V이며, 그외에는 rpm/100ms/V임. (확인요) ⚙️ 설정값이 0이면 가감속도를 조정하는 일을 하지 않음. ⚙️ (operation_mode_SWITCH & 0x04000000L)==0L 일때 속도제어모드에서 가감속도 조정을 실시함. ⚙️ (operation_mode_SWITCH & 0x08000000L)!=0L 일때 위치제어모드에서 가감속도 조정을 실시함. ⚙️ 파라미터는 각각 BLDC1_PrmBuffer[36]/BLDC1_PrmBuffer[37]/BLDC2_PrmBuffer[36]/BLDC2_PrmBuffer[37]에 저장됨. ⚙️ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. ⚙️ F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
(2013년 3월 3일 이전 S/W 버전) :SDddd1,ddd2; (+#^) :SD?; (+#^)	operation	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 시작하는 전압변동량을 설정함. 1st word : 모터1의 전압변동량 [0 ~ 최고사용전압/4] (4 V, 제일메디컬은 1 V) 2nd word : 모터2의 전압변동량 [0 ~ 최고사용전압/4] (4 V, 제일메디컬은 1 V)
	return value	SDddd1,ddd2;
	comment	⚡ 최고사용전압(Z6 명령어로 설정함)이 12V 보다 작거나 600V보다 크면 30V 로 취급함. ⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. ⚡ F2811/F28334 에서만 가능한 명령임.
(2013년 3월 3일 부터 TMS320F2811 용, 2014년 2월 1 일부터 TMS320F28334용 S/W 버전) :SDddd1,ddd2; 또는 :SDddd1,ddd2,d ddd3,ddd4; (+#^) :SD?; (+#^)	operation	가속/감속시 저전압/과전압이 발생하는 것을 억제하기 시작하는 전압변동량과 전압제어게인을 설정함. 1st word : 모터1의 전압변동량 [0 ~ 최고사용전압/4] (4 V, 제일메디컬은 1 V) 2nd word : 모터2의 전압변동량 [0 ~ 최고사용전압/4] (4 V, 제일메디컬은 1 V) 3rd word : 비례제어게인 [0 ~ 100] (0 bit) 4th word : 미분제어게인 [0 ~ 500] (0 bit)
	return value	SDddd1,ddd2,ddd3,ddd4;
	comment	⚡ 최고사용전압(Z6 명령어로 설정함)이 12V 보다 작거나 600V보다 크면 30V 로 취급함. ⚡ 비례제어게인이 0 이면 ON/OFF 형식으로 브레이크저항을 작동시키는 기존방식으로 작동됨. ⚡ 제어게인을 사용할때 비례게인은 8, 미분게인은 64에서 시작하며, Qx23/24/25; 를 이용하여 제어상황을 모니터할수 있음. ⚡ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. ⚡ F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;S1dddd1,dddd2; (+#^)	operation	OE/OF 위치제어모드에서 모터1의 sine파 변위 주기 및 진폭을 설정함. 1st word : 주파수 [100 - 30000] (1000 0.001Hz) 2nd word : 진폭 [0 - 32767] (0 pulse)
	return value	S1dddd1,dddd2;
;S1?; (+#^)	comment	⚡ SM mode 0f에서 sine파 위치제어가 실시됨. ⚡ 스텝모터에서는 주기가 최대 1310ms 까지만 가능함. ⚡ F2811/F28334 에서만 가능한 명령임.
;S2dddd1,dddd2; (+#^)	operation	OE/OF 위치제어모드에서 모터2의 sine파 변위 주기 및 진폭을 설정함. 1st word : 주파수 [100 - 30000] (1000 0.001Hz) 2nd word : 진폭 [0 - 32767] (0 pulse)
	return value	S2dddd1,dddd2;
;S2?; (+#^)	comment	⚡ SM mode 0f에서 sine파 위치제어가 실시됨. ⚡ 스텝모터에서는 주기가 최대 1310ms 까지만 가능함. ⚡ F2811/F28334 에서만 가능한 명령임.
;Sodddd1,dddd2; (+#^)	operation	속도제어 및 위치제어 모드에서 리밋센서, 비상정지 명령 또는 비상정지 상황에 의한 강제 감속시의 감속율을 설정함. 1st word : 모터1 감속도 설정값 [1 ~ 60000] (9523 rpm/100ms or 10 rpm/ms) 2nd word : 모터2 감속도 설정값 [1 ~ 60000] (9523 rpm/100ms or 10 rpm/ms)
	return value	Sodddd1,dddd2;
	comment	단위는 RJM_VER7/RJM_VER7에서는 rpm/ms, 나머지 H/W 버전에서는 rpm/100ms임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
(Force/Moment 센서가 아닌 경우) ;SI dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8; (+#^%) ;SI?; (+#^%) 2013년 4월 4일 S/W 버전부터는 Si 명령을 사용하여 0.01A 단위로 전류값을 설정할 수 있음.	operation	구동회로와 기구를 보호하기 위한 logic에서 사용하는 전류리미트값과 time threshold를 설정함. 1st word : 모터1 전류리미트 [0 - 1023/1638 or 1800(스텝용) or 8190(380V용)] (1638) 2nd word : 모터2 전류리미트 [0 - 1023/1638 or 1800(스텝용) or 8190(380V용)] (1638) 3rd word : 모터1의 전류초과를 허용하는 시간 [1 - 50000] (1000 ms) 4th word : 모터2의 전류초과를 허용하는 시간 [1 - 50000] (1000 ms) 5th word : 모터1의 전류초과를 허용하는 시간 대비 전류를 끄는 기간의 비율 [1 - 10] (5) 6th word : 모터2의 전류초과를 허용하는 시간 대비 전류를 끄는 기간의 비율 [1 - 10] (5) 7th word : 모터1의 전류초과시에 위치명령을 hold하기 까지의 시간 [1 - 10000] (500 ms) 8th word : 모터2의 전류초과시에 위치명령을 hold하기 까지의 시간 [1 - 10000] (500 ms)
	return value	SI dddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8;
	comment	2008년 6월 이전의 S/W 버전에서는 $1A \times 25m\Omega \times 11(\text{앰프 증폭률}) = 0.275V$ 이므로 3.3V/3.0V는 $3.3V \div 0.275V = 12A/10.9A$ 에 해당하며 85.25/375.375 bit/A. (DC ver1.1) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$ 이므로 3.3V/3.0V는 $3.3V \div 0.25V = 13.2A/12A$ 에 해당하며 77.5/341.25 bit/A. (DC V1.2부터) $1A \times 2m\Omega \times 50(\text{앰프 증폭률}) = 0.1V$ 이므로 3.3V/3.0V는 $3.3V \div 0.1V = 33A/30A$ 에 해당하며 31/136.5 bit/A. (Piggyback-DC/RJM_VER8부터) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) \times 3.3/5 = 0.165V$ 이므로 3.3V/3.0V는 $3.3V \div 0.165V = 20A/18.18A$ 에 해당하며 51.15/225.225 bit/A. (Piggyback-BLDC) $1A \times 10m\Omega \times 50(\text{앰프 증폭률}) = 0.5V$ 이므로 3.3V/3.0V는 $3.3V \div 0.5V = 6.6/6A$ 에 해당하며 155/682.5 bit/A. (mini-DC) $1A \times 20(10)m\Omega \times 50(\text{앰프 증폭률}) = 1.00V/0.5V$ 이므로 3.3V/3.0V는 $3.3V \div 1.00V = 3.3A/6A$ 에 해당하며 310/682.5 bit/A. (STEP형) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$ 이므로 5.0V는 $5.0V \div 0.25V = 20A$ 에 해당하며 204.75 bit/A. (Sensor-BLDC)

명령어	구분	설명
	comment	<p>따라서 DC V1.1 인 경우에 0 - 1023/4095의 값은 0 - 12/10.9A을 전류명령값을 의미함. 1[A]=85/375, 2[A]=170/751, 3[A]=256/1126, 4[A]=341/1502, 5[A]=426/1877, 6[A]=512/2252, 7[A]=597/2628, 8[A]=682/3003, 9[A]=768/3378, 10[A]=853/3754, 11[A]=938 임.</p> <p>DC V1.2부터는 1[A]=78/341, 2[A]=155/683, 3[A]=233/1024, 4[A]=310/1365, 5[A]=388/1706, 6[A]=465/2048, 7[A]=543/2389, 8[A]=620/2730, 9[A]=698/3071, 10[A]=775/3413, 11[A]=853/3754, 12[A]=930, 13[A]=1008 임.</p> <p>Piggyback-DC/RJM_VER8부터는 1[A]=31/137, 2[A]=62/273, 3[A]=93/410, 4[A]=124/546, 5[A]=155/683, 6[A]=186/819, 7[A]=217/956, 8[A]=248/1092, 9[A]=279/1229, 10[A]=310/1365, 11[A]=341/1502, 12[A]=372/1638, 13[A]=403/1775, 14[A]=434/1911, 15[A]=465, 16[A]=496 임.</p> <p>Piggyback-BLDC인 경우는 1[A]=51/225, 2[A]=102/450, 3[A]=153/676, 4[A]=205/901, 5[A]=256/1126, 6[A]=307/1351, 7[A]=358/1577, 8[A]=409/1802, 9[A]=460/2027, 10[A]=512/2252, 11[A]=563/2477, 12[A]=614/2703, 13[A]=665/2928, 14[A]=716/3153, 15[A]=767/3378 임.</p> <p>mini-DC 에서는 1[A]=155/683, 2[A]=310/1365 임.</p> <p>STEP 에서는 0.25[A]=78/171, 0.5[A]=155/341, 0.75[A]=233/512, 1[A]=310/683, 1.25[A]=388/853, 1.5[A]=465/1024, 1.75[A]=1194, 2[A]=1365, 2.25[A]=1536, 2.5[A]=1706, 2.75[A]=1878 임.</p> <p>Sensor-BLDC는 1[A]=205, 2[A]=410, 3[A]=614, 4[A]=819, 5[A]=1024, 6[A]=1229, 7[A]=1443, 8[A]=1638, 9[A]=1843 임.</p> <p>2008년 6월 이후의 모든 F2811/F28334 H/W 버전에서는 다음과 같이 통일 되었음.</p> <p>1mΩ인 경우는 40.95/40.95 bit/A (RJM_VER7/RJM_VER8 : 400W DC제어기)</p> <p>2mΩ인 경우는 81.9/81.9 bit/A (RJM_VER3 : IRMCK201을 사용한 BLDC제어기, RJM_VER8 : 200W DC제어기)</p> <p>4mΩ인 경우는 163.8/163.8 bit/A (RJM_VER4/RJM_VER7 : 100W DC/BLDC제어기)</p> <p>5mΩ인 경우는 204.75/204.75 bit/A (RJM_VER7 : IRMCK201을 사용하지 않은 BLDC제어기)</p> <p>8mΩ인 경우는 327.6/327.6 bit/A (RJM_VER5 : 100W STEP제어기)</p> <p>10mΩ인 경우는 409.5/409.5 bit/A (RJM_VER4 : mini DC, RJM_VER5 : 50W STEP제어기)</p> <p>Zr 명령을 사용하면 설치된 전류감지 저항값을 확인할수 있음.</p>

명령어	구분	설명
	comment	<p>2012년 4월 12일 S/W 이전 버전에서는 RJM_VER3/7/8 의 경우는 각각의 모터구동 BRIDGE 회로의 high side 쪽으로 총전류감지저항을 통해서 들어가는 전류를 감지하고 이를 사용하여 전류리미를 적용하나, RJM_VER1/2/4/5/6 의 경우는 코일에 흐르는 상전류를 사용하여 전류 리미를 적용함.</p> <p>2012년 4월 12일 S/W 버전부터는 SI 명령어로 설정되는 전류 리미트는 상전류의 크기이며, 총전류는 그 값의 2/3가 리미트로 적용됨. SI 명령어로 설정되는 전류 리미트값은 Z7 명령어로 설정되는 전류최대치보다 크지 않게 제한되며, Sw 명령어로 설정되는 제어전류 최대치보다 클수 없음. SI 명령어로 설정되는 전류 리미트의 125%에 해당하는 전류값을 단 한번이라도 초과하면 과전류 오류를 발생시키는데, 이 값이 Sw 명령어로 설정하는 전류치의 125%보다 작다면 Sw 명령어로 설정하는 전류치의 125%로 강제 설정됨.</p> <p>2012년 5월 28일 S/W 버전부터는 모터x의 전류초과를 허용하는 시간값(SI 명령어의 세번째/네번째 값)의 3/4 크기보다 Si 명령어로 설정하는 값이 작아지지 않게 (전원투입하여 마이컴 초기화 할때) 하였음.</p> <p>2012년 12월 19일 이전의 S/W 버전에서는 1채널제어기에서 SI 명령으로 두번째값을 작게 설정하면 오류를 발생시키므로 첫번째 값과 같은 값을 설정해야함.</p> <p>2013년 3월 5일 S/W 버전부터는 스텝모터의 경우 전류상한치를 1800 까지 설정할수 있게 하였음. 2013년 4월 4일 S/W 버전부터는 Si 명령을 사용하여 0.01A 단위로 전류값을 설정할수있음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
(Force/Moment 센서인 경우만 해당됨) ;SIA55A, dddd1, dddd2; (+#^)	operation	F/T 센서에서 AD channel 8/9/10/11(내장형의 경우는 0/1/2/3) 전압의 평균값을 설정하기 위한 filtering 을 실시하고, 온도값과 함께 force_data_offset 변수값을 설정함. 1st word : 필터링 시간 [10 - 10000] (xxxx ms) 2nd word : 필터 시정수 [4 - 1024] (xxxx)
	return value	SI dddd1, dddd2; 1st word : filtering 타이머의 남은 시간 (ms) 2nd word : 필터 시정수 (ms)
	comment	필터시정수로 필터링시간동안 필터링을 행하여 force_data_offset 변수를 설정하며, 설정된 값은 Qb2; 명령어로 읽어낼수있음. F/T 센서를 조립하였다가 풀었을때는 SIA5A5,0,0; 명령어로 F/T센서의 앰프 offset을 2048로 맞추어준 이후에 SIA55A, dddd1, dddd2; 명령을 사용할 것. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
(Force/Moment 센서인 경우만 해당됨) ;SI5AA5, dddd1, dddd2; (+#^)	operation	F/T 센서에서 AD channel 8/9/10/11(내장형의 경우는 0/1/2/3) 전압의 평균값을 설정하기 위한 filtering 을 실시하고, 온도증가에 따른 전압상승의 기울기를 설정함. 1st word : 필터링 시간 [10 - 10000] (xxxx ms) 2nd word : 필터 시정수 [4 - 1024] (xxxx)
	return value	SI dddd1, dddd2; 1st word : filtering 타이머의 남은 시간 + 15000 (ms) 2nd word : 필터 시정수 (ms)
	comment	필터시정수로 필터링시간동안 필터링을 행하여 전압상승의 기울기를 설정하며, 설정된 값은 MS?; 명령어로 읽어낼수있음. SI5AA5 를 수행하기 전에 F/T 센서의 온도를 일정하게 유지한 상태로 SIA55A 명령을 먼저 수행하여야함. SI5AA5 를 수행할때도 F/T 센서의 온도를 일정하게 유지하되, 이전의 SIA55A 명령을 수행때와는 다른 온도에서 실시해야함. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
(Force/Moment 센서인 경우만 해당됨) ;SIA5A5,0,0; (+#^) ;SI?; (+#^)	operation	F/T 센서에서 no load 상태일때 AD channel 8/9/10/11(내장형의 경우는 0/1/2/3)의 변환값이 2048이 되도록 D/A 변환기를 조절하여 만듦. (F/T센서의 앰프 offset을 2048로 맞추어주는 작동이며 총 5초가 걸림) 1st word : 의미 없음. 2nd word : 의미 없음.
	return value	SI dddd1, dddd2; 1st word : filtering 타이머의 남은 시간 + 30000 (ms) 2nd word : 필터 시정수 (ms)
	comment	Qx3; 명령어로 모니터링을 하면 SIA5A5,0,0; 명령을 보낸후 5초이내에 4채널의 값이 모두 2048 근방의 값을 갖게 됨. F/T 센서를 조립하였다가 풀었을때는 반드시 이 명령어를 한번 수행하고 그 결과값을 EsA55A; 명령어로 저장해주어야함. 이때 F/T 센서에는 외력이 없는 상태이어야함. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
	operation	과전류시에 HW적으로 회로를 차단하기 위한 과전류 설정값을 설정함. 1st word : 모터1 전류리미트 : DC의 경우 [0 - PWM_PERIOD] (PWM_PERIOD) 2nd word : 모터2 전류리미트 : DC의 경우 [0 - PWM_PERIOD] (PWM_PERIOD)
	return value	SWdddd1, dddd2;
;SWdddd1, dddd2; (+##^%) ;SW?; (+##^%)	comment	<p>항상 설정이 가능하며 전원투입시의 기본값은 PWM_PERIOD(DC의 경우)이며 EEPROM에 저장되지 않음. 최대값을 설정하면 3.3V/3.0V의 전압이 HW적인 과전류 차단회로로 공급되는데, 이때 다음과 같이 적용됨.</p> <ul style="list-style-type: none"> - $1A \times 25m\Omega \times 11(\text{앰프 증폭률}) = 0.275V$이므로 3.3V/3.0V는 $3.3V \div 0.275V = 12A/10.9A$의 과전류 설정치에 해당함. (DC ver1.1) - $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$이므로 3.3V/3.0V는 $3.3V \div 0.25V = 13.2A/12A$의 과전류 설정치에 해당함. (DC V1.2부터) - $1A \times 2m\Omega \times 50(\text{앰프 증폭률}) = 0.1V$이므로 3.3V/3.0V는 $3.3V \div 0.1V = 33A/30A$의 과전류 설정치에 해당함. (Piggyback-DC) <p>따라서 DC V1.1 인 경우에 0 - PWM_PERIOD의 값은 0 - 12A/10.9A의 전류제한값을 의미함. DC V1.2에서는 0 - PWM_PERIOD의 값은 0 - 13.2A/12A의 전류제한값을 의미함. Piggyback-DC에서는 PWM_PERIOD/2 - PWM_PERIOD의 값은 0 - 16.5A/15A의 전류제한값을 의미함. Piggyback-DC에서는 PWM_PERIOD/2의 offset을 더한 값을 과전류설정값으로 설정하여야함. BLDC V1.1에서는 PWM_PERIOD/3.96 - PWM_PERIOD의 값은 0 - 14.95A/13.6A의 전류제한값을 의미함. BLDC V1.1에서는 PWM_PERIOD/3.96의 offset을 더한 값을 과전류설정값으로 설정하여야함.</p>

명령어	구분	설명
	operation	제어시 사용하는 전류의 리밋을 설정함. 1st word : 모터1 제어전류리밋 : (8190 or 6142 or 1800 or 1638, 2009년 7월 15일 S/W 버전부터는 comment 마지막부분 참조) 2nd word : 모터2 제어전류리밋 : (8190 or 6142 or 1800 or 1638, 2009년 7월 15일 S/W 버전부터는 comment 마지막부분 참조)
	return value	Swdddd1,dddd2;
	comment	전류를 사용하여 전류나 위치나 속도제어를 하는 경우에 사용하는 전류의 최대치를 설정함. IRMCK201을 사용하는 RJM_VER3인 경우는 4095 값은 정격전류를 의미함. 2009년 7월 15일 S/W 버전부터 모든 F2811/F28334 H/W 버전에서는 다음과 같은 적용을 받음. KOMOTEK 380V모터의 경우 설정가능한 값의 범위는 -8190 ~ +8190 이며, 전류감지저항으로 10m0hm을 사용하는 경우 -20A ~ +20A 의 범위에 해당함. IRMCK201을 사용한 경우 설정가능한 값의 범위는 -12288 ~ +12288 이며, Zc 명령어로 설정되고 IRMCK201의 파라미터설정에 사용한 정격전류의 3배에 해당하는 범위임. 그 외의 경우 설정가능한 값의 범위는 -1638 ~ +1638 이며, 이는 Z3 명령어에 의하여 설정되고 실제 장착된 전류감지저항에 따라서 다른데, 2m0hm은 -20A ~ +20A 범위를 가리키며 5m0hm은 -8A ~ +8A 범위를 가리키고 10m0hm은 -4A ~ +4A 범위를 가리킴. 2012년 4월 10일 S/W 버전부터는 Sw 명령어로 설정되는 제어전류 리밋값은 Z7 명령어로 설정되는 전류최대치보다 크지 않게 제한됨. 2013년 3월 5일 S/W 버전부터는 스텝모터의 경우 Sw 명령어로 설정되는 제어전류 리밋값은 1800 까지로 늘어남. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Szdddd1,dddd2; (+#^) ;Sz?; (+#^)	operation	제어시 사용하는 전류의 리밋을 0.01A 단위로 설정함. 1st word : 모터1 제어전류리밋 : (comment 마지막부분 참조) 2nd word : 모터2 제어전류리밋 : (comment 마지막부분 참조)
	return value	Szdddd1,dddd2;
	comment	2009년 7월 15일 S/W 버전부터 적용되는 명령어임. 전류를 사용하여 전류나 위치나 속도제어를 하는 경우에 사용하는 전류의 최대치를 설정함. Z3 명령어로 전류감지저항값이 정상적으로 설정되어 있어야함. IRMCK201을 사용한 경우는 Zc 명령어로 nominal current가 정상적으로 설정되어 있어야함. KOMOTEK 380V모터의 경우 설정가능한 값의 범위는 전류감지저항으로 10m0hm을 사용하는 경우 -20A ~ +20A의 범위임. 그 외의 경우 설정가능한 값의 범위는 -1638 ~ +1638 이며, 이는 Z3 명령어에 의하여 설정되고 실제 장착된 전류감지저항에 따라서 다른데, 2m0hm은 -20A ~ +20A 범위이고 5m0hm은 -8A ~ +8A 범위이며 10m0hm은 -4A ~ +4A 범위임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Svddd1,ddd2,d ddd3; (+#^%) ;Sv?: (+#^%)	operation	<p>MOSFET의 gate를 off 시키는 전압 강하량과 과전압 및 검출시간의 크기를 설정함.</p> <p>1st word : 이 설정전압 이상의 전압강하가 발생하기를 3rd word 로 설정된 시간동안 연속되면 게이트를 off 시킴 [100 - 3000/22500(380V 모터구동기의 경우)] (800/10000 bit/0.01V)</p> <p>2nd word : 이 설정전압 이상의 전압이 공급되기를 3rd word 로 설정된 시간동안 연속되면 게이트를 off 시킴 [800 - 9000/45000(380V 모터구동기의 경우)] (3400/4000(제일메디컬)/45000 bit/0.01V)</p> <p>3rd word : 설정시간 [1 - 100] (10 ms)</p>
	return value	<p>Svddd1,ddd2,ddd3,ddd4,ddd5; (2011년 9월 11일 이전 S/W 버전)</p> <p>Svddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7; (2011년 9월 11일 S/W 버전부터)</p> <p>4th word : 전원투입시의 filtering된 모터용 공급전압의 크기 (1 0.01V/bit)</p> <p>5th word : 현재의 filtering된 모터용 공급전압의 크기 (1 0.01V/bit)</p> <p>6th word : 현재의 slow filtering된 모터용 공급전압의 크기 (1 0.01V/bit)</p> <p>7th word : 현재의 filtering된 모터용 또는 제어용(모터용 전원에서 제어용 전원을 내부에서 만드는 경우) 공급전압의 크기 (1 0.01V/bit)</p>
	comment	<p>이 설정값은 DSP의 SARAM에 저장되며 만약 EEPROM에 저장하려면 ESxxxx:명령어를 사용하여야함.</p> <p>기본값은 800(8V)/10000(100V,380V 모터구동기의 경우), 3400(34V)/4000(40V, 제일메디컬)/45000(450V,380V 모터구동기의 경우), 10(10ms) 임.</p> <p>DC motor에서는 AD7 입력전압을 사용하는데, 전압이상으로 gate가 off 되었을때, 복구하려면 PR 명령어를 사용하여야 함.</p> <p>BLDC motor에서도 적용되는데, 가변저항으로 설정된 전압을 사용하여 IRMCK201 자체에서 과전압 또는 저전압일때 gate를 자동 off 시키는 기능도 병행됨. (46.33V 이상이면 과전압, 13.56V 이하이면 저전압)</p> <p>이러한 과정을 통하여 과전류시에 발생하는 전압강하를 이용하여 DC와 BLDC의 구동회로를 보호할수있음.</p> <p>supply 전압의 허용 강하량은 통상 supply 전압의 1/3에 해당하는 값을 설정하는 것이 좋음.</p> <p>supply 전압의 최대허용값은 통상 supply 전압의 1 1/3에 해당하는 값을 설정하는 것이 좋음.</p> <p>2nd word 값의 최고치가 2012년 2월 6일 S/W 버전부터 6000에서 9000으로 상향조정되었음.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;SCiiii1,iiii2; (%) ;SC?; (%)	operation	전류제어 모드에서 전류명령값을 설정함. 1st integer : 모터1 전류명령값 [-1023/4095 ~ 1023/4095] (bit) 2nd integer : 모터2 전류명령값 [-1023/4095 ~ 1023/4095] (bit)
	return value	SCiiii1,iiii2; or SCiiii1,iiii2,iiii3,iiii4;(RJM_VER5 STEP형 만 해당됨) or SCiiii1,iiii2,iiii3,iiii4,iiii5,iiii6;(RJM_VER6 3상STEP형 만 해당됨) 1st integer : 모터1 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil1) 2nd integer : 모터2 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil2) 3rd integer : coil3 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil3) 4th integer : coil4 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil4) 5rd integer : coil5 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER6 3상STEP형 만 해당, coil5) 6th integer : coil6 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER6 3상STEP형 만 해당, coil6)
	comment	1A×25mΩ×11(앰프 증폭률) = 0.275V이므로 3.3V/3.0V는 3.3V÷0.275V = 12A/10.9A에 해당하며 85.25/375.375 bit/A. (DC ver1.1) 1A×5mΩ×50(앰프 증폭률) = 0.25V이므로 3.3V/3.0V는 3.3V÷0.25V = 13.2A/12A에 해당하며 77.5/341.25 bit/A. (DC V1.2부터) 1A×2mΩ×50(앰프 증폭률) = 0.1V이므로 3.3V/3.0V는 3.3V÷0.1V = 33A/30A에 해당하며 31/136.5 bit/A. (Piggyback-DC/RJM_VER8부터) BLDC 모터제어 chip인 IRMCK201에서 전류를 측정하고 스케일링하기 때문에, 전류감지저항값과 IR2175S 및 전류스케일과 연관성이 있음. (Piggyback-BLDC형) 1A×10mΩ×50(앰프 증폭률) = 0.5V이므로 3.3V/3.0V는 3.3V÷0.5V = 6.6/6A에 해당하며 155/682.5 bit/A. (mini-DC) 1A×20(10)mΩ×50(앰프 증폭률) = 1.00V/0.5V이므로 3.3V/5.0V는 3.3V÷1.00V = 3.3A/10A에 해당하며 310/409.5 bit/A. (STEP형) 다음페이지 참조!

명령어	구분	설명
	comment	<p> $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$이므로 $5.0V$는 $5.0V \div 0.25V = 20A$에 해당하며 204.75 bit/A. (Sensor-BLDC) </p> <p> 따라서 DC V1.1 인 경우에 $0 - 1023/4095$의 값은 $0 - 12/10.9A$을 전류명령값을 의미함. $1[A]=85/375$, $2[A]=170/751$, $3[A]=256/1126$, $4[A]=341/1502$, $5[A]=426/1877$, $6[A]=512/2252$, $7[A]=597/2628$, $8[A]=682/3003$, $9[A]=768/3378$, $10[A]=853/3754$, $11[A]=938$ 이며 $10A$ 이상으로 설정하지 말것. DC V1.2부터는 $1[A]=78/341$, $2[A]=155/683$, $3[A]=233/1024$, $4[A]=310/1365$, $5[A]=388/1706$, $6[A]=465/2048$, $7[A]=543/2389$, $8[A]=620/2730$, $9[A]=698/3071$, $10[A]=775/3413$, $11[A]=853/3754$, $12[A]=930$, $13[A]=1008$ 이름 $11A$ 이상으로 설정하지 말것. </p> <p> Piggyback-DC/RJM_VER8부터는 $1[A]=31/137$, $2[A]=62/273$, $3[A]=93/410$, $4[A]=124/546$, $5[A]=155/683$, $6[A]=186/819$, $7[A]=217/956$, $8[A]=248/1092$, $9[A]=279/1229$, $10[A]=310/1365$, $11[A]=341/1502$, $12[A]=372/1638$, $13[A]=403/1775$, $14[A]=434/1911$, $15[A]=465$, $16[A]=496$ 이며 $14A$ 이상으로 설정하지 말 것. </p> <p> Piggyback-BLDC인 경우는 BLDC 모터파라미터 설정시에 주어야하는 정격전류값의 전류가 실제로 흐를 때, 측정된 전류값의 최종치가 4096이 되도록 하기 때문에, 전류명령값으로 $4096/16=256$를 설정하면 정격전류 를 명령하는 것이 됨. </p> <p> mini-DC 에서는 $1[A]=155/683$, $2[A]=310/1365$ 이며 $2.5A$ 이상으로 설정하지 말것. </p> <p> STEP 에서는 $0.25[A]=78/102$, $0.5[A]=155/205$, $0.75[A]=233/307$, $1[A]=310/410$, $1.25[A]=388/512$, $1.5[A]=465/614$, $1.75[A]=717$, $2[A]=819$, $2.25[A]=921$, $2.5[A]=1024$, $2.75[A]=1126$ 이며 $1.5/2.75A$ 이상으 로 설정하지 말것. </p> <p> Sensor-BLDC는 $1[A]=205$, $2[A]=410$, $3[A]=614$, $4[A]=819$, $5[A]=1024$, $6[A]=1229$, $7[A]=1443$, $8[A]=1638$, $9[A]=1843$ 이며 $9A$ 이상으로 설정하지 말것. </p> <p> 다음페이지 참조! </p>

명령어	구분	설명
	comment	RJM_VER6 3상STEP형은 모터1(coil1,coil2,coil3)/모터2(coil4,coil5,coil6) 으로 코일이 사용됨. iiii1,iiii2,iiii3,iiii4,iiii5,iiii6은 부호가 있는 십진수임.

명령어	구분	설명
;SCiiii1,iiii2; (+#^) ;SC?; (+#^)	operation	전류제어 모드에서 전류명령값을 설정함. 1st integer : 모터1 전류명령값 [-1023/4095 ~ 1023/4095, 2009년 7월 15일 S/W 버전부터는 comment 마지막부분 참조, 2013년 6월 18일 S/W 버전부터는 comment 마지막부분 참조] (bit) 2nd integer : 모터2 전류명령값 [-1023/4095 ~ 1023/4095, 2009년 7월 15일 S/W 버전부터는 comment 마지막부분 참조, 2013년 6월 18일 S/W 버전부터는 comment 마지막부분 참조] (bit)
	return value	SCiiii1,iiii2; or SCiiii1,iiii2,iiii3,iiii4;(RJM_VER5 STEP형 만 해당됨) or SCiiii1,iiii2,iiii3,iiii4,iiii5,iiii6;(RJM_VER6 3상STEP형 만 해당됨) 1st integer : 모터1 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil1) 2nd integer : 모터2 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil2) 3rd integer : coil3 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil3) 4th integer : coil4 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER5/6 STEP형은 coil4) 5rd integer : coil5 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER6 3상STEP형 만 해당, coil5) 6th integer : coil6 전류명령값 [-1023/4095 ~ 1023/4095] (bit) (RJM_VER6 3상STEP형 만 해당, coil6)
	comment	1A×25mΩ×11(앰프 증폭률) = 0.275V이므로 3.3V/3.0V는 3.3V÷0.275V = 12A/10.9A에 해당하며 85.25/375.375 bit/A. (DC ver1.1) 1A×5mΩ×50(앰프 증폭률) = 0.25V이므로 3.3V/3.0V는 3.3V÷0.25V = 13.2A/12A에 해당하며 77.5/341.25 bit/A. (DC V1.2부터) 1A×2mΩ×50(앰프 증폭률) = 0.1V이므로 3.3V/3.0V는 3.3V÷0.1V = 33A/30A에 해당하며 31/136.5 bit/A. (Piggyback-DC/RJM_VER8부터) BLDC 모터제어 chip인 IRMCK201에서 전류를 측정하고 스케일링하기 때문에, 전류감지저항값과 IR2175S 및 전류스케일과 연관성이 있음. (Piggyback-BLDC형) 1A×10mΩ×50(앰프 증폭률) = 0.5V이므로 3.3V/3.0V는 3.3V÷0.5V = 6.6/6A에 해당하며 155/682.5 bit/A. (mini-DC) 다음페이지 참조!

명령어	구분	설명
	comment	<p> $1A \times 20(10)m\Omega \times 50(\text{앰프 증폭률}) = 1.00V/0.5V$이므로 $3.3V/5.0V$는 $3.3V \div 1.00V = 3.3A/10A$에 해당하며 $310/409.5$ bit/A. (STEP형) $1A \times 5m\Omega \times 50(\text{앰프 증폭률}) = 0.25V$이므로 $5.0V$는 $5.0V \div 0.25V = 20A$에 해당하며 204.75 bit/A. (Sensor-BLDC) </p> <p> 따라서 DC V1.1 인 경우에 $0 - 1023/4095$의 값은 $0 - 12/10.9A$을 전류명령값을 의미함. 1[A]=85/375, 2[A]=170/751, 3[A]=256/1126, 4[A]=341/1502, 5[A]=426/1877, 6[A]=512/2252, 7[A]=597/2628, 8[A]=682/3003, 9[A]=768/3378, 10[A]=853/3754, 11[A]=938 이며 10A 이상으로 설정하지 말것. DC V1.2부터는 1[A]=78/341, 2[A]=155/683, 3[A]=233/1024, 4[A]=310/1365, 5[A]=388/1706, 6[A]=465/2048, 7[A]=543/2389, 8[A]=620/2730, 9[A]=698/3071, 10[A]=775/3413, 11[A]=853/3754, 12[A]=930, 13[A]=1008 이며 11A 이상으로 설정하지 말것. Piggyback-DC/RJM_VER8부터는 1[A]=31/137, 2[A]=62/273, 3[A]=93/410, 4[A]=124/546, 5[A]=155/683, 6[A]=186/819, 7[A]=217/956, 8[A]=248/1092, 9[A]=279/1229, 10[A]=310/1365, 11[A]=341/1502, 12[A]=372/1638, 13[A]=403/1775, 14[A]=434/1911, 15[A]=465, 16[A]=496 이며 14A 이상으로 설정하지 말것. </p> <p> Piggyback-BLDC인 경우는 BLDC 모터파라미터 설정시에 주어야하는 정격전류값의 전류가 실제로 흐를 때, 측정된 전류값의 최종치가 4096이 되도록 하기 때문에, 전류명령값으로 $4096/16=256$를 설정하면 정격전류를 명령하는 것이 됨. </p> <p> mini-DC 에서는 1[A]=155/683, 2[A]=310/1365 이며 2.5A 이상으로 설정하지 말것. STEP 에서는 0.25[A]=78/102, 0.5[A]=155/205, 0.75[A]=233/307, 1[A]=310/410, 1.25[A]=388/512, 1.5[A]=465/614, 1.75[A]=717, 2[A]=819, 2.25[A]=921, 2.5[A]=1024, 2.75[A]=1126 이며 1.5/2.75A 이상으로 설정하지 말것. </p> <p> 다음페이지 참조! </p>

명령어	구분	설명
	comment	<p>Sensor-BLDC는 1[A]=205, 2[A]=410, 3[A]=614, 4[A]=819, 5[A]=1024, 6[A]=1229, 7[A]=1443, 8[A]=1638, 9[A]=1843 이며 9A 이상으로 설정하지 말것.</p> <p>RJM_VER6 3상STEP형은 모터1(coil1,coil2,coil3)/모터2(coil4,coil5,coil6) 으로 코일이 사용됨. iiii1,iiii2,iiii3,iiii4,iiii5,iiii6은 부호가 있는 십진수임.</p> <p>2008년 6월 이후의 모든 F2811/F28334 H/W 버전에서는 다음과 같이 통일 되었음. 2mΩ인 경우는 81.9/81.9 bit/A (RJM_VER3 : IRMCK201을 사용한 BLDC제어기, RJM_VER8 : 고출력 DC) 5mΩ인 경우는 204.75/204.75 bit/A (RJM_VER7 : IRMCK201을 사용하지 않은 BLDC제어기) 10mΩ인 경우는 409.5/409.5 bit/A (RJM_VER4 : mini DC, RJM_VER5 : step motor)</p> <p>2009년 7월 15일 S/W 버전부터 모든 F2811/F28334 H/W 버전에서는 다음과 같은 적용을 받음. KOMOTEK 380V모터의 경우 설정가능한 값의 범위는 -8190 ~ +8190 이며, 전류감지저항으로 10mΩ을 사용하는 경우 -20A ~ +20A 의 범위에 해당함. IRMCK201을 사용한 경우 설정가능한 값의 범위는 -12288 ~ +12288 이며, Zc 명령어로 설정되고 IRMCK201의 파라미터설정에 사용한 정격전류의 3배에 해당하는 범위임. 그 외의 경우 설정가능한 값의 범위는 -1638 ~ +1638 이며, 이는 Z3 명령어에 의하여 설정되고 실제 장착된 전류감지저항에 따라서 다른데, 2mΩ은 -20A ~ +20A 범위를 가리키며 5mΩ은 -8A ~ +8A 범위를 가리키고 10mΩ은 -4A ~ +4A 범위를 가리킴.</p> <p>2013년 6월 18일 S/W 버전부터 lead angle제어 모드일때 SC 명령에 의한 설정값은 전류명령값이 아니라 lead_anglex_set 값임.</p>

명령어	구분	설명
	operation	전류치를 암페어로 환산하는 scale factor를 설정함. 1st word : 모터1의 전류치 환산 factor [1 ~ 32767] (8192*0.01A/bit) (32768*0.01A/bit) 2nd word : 모터2의 전류치 환산 factor [1 ~ 32767] (8192*0.01A/bit) (32768*0.01A/bit)
	return value	Scddd1,ddd2;
	comment	<p>1A × 25mΩ × 11(앰프 증폭률) = 0.275V이므로 3.3V/3.0V는 3.3V ÷ 0.275V = 12A/10.9A에 해당하며 8192*1200/1023 = 9609/8729. (DC ver1.1)</p> <p>1A × 5mΩ × 50(앰프 증폭률) = 0.25V이므로 3.3V/3.0V는 3.3V ÷ 0.25V = 13.2A/12A에 해당하며 8192*1320/1023 = 10570/9602. (DC V1.2부터)</p> <p>1A × 2mΩ × 50(앰프 증폭률) = 0.1V이므로 3.3V/3.0V는 3.3V ÷ 0.1V = 33A/30A에 해당하며 8192*3300/1023 = 26426/24006. (Piggyback-DC/RJM_VER8부터)</p> <p>1A × 5mΩ × 50(앰프 증폭률) × 3.3/5 = 0.165V이므로 3.3V/3.0V는 3.3V ÷ 0.165V = 20A/18.18A에 해당하며 8192*2000/1023 = 16016/14549. (Piggyback-BLDC)</p> <p>1A × 10mΩ × 50(앰프 증폭률) = 0.5V이므로 3.3V/3.0V는 3.3V ÷ 0.5V = 6.6/6A에 해당하며 8192*660/1023 = 5285/4801. (mini-DC)</p> <p>1A × 20(10)mΩ × 50(앰프 증폭률) = 1.00V/0.5V이므로 3.3V/5.0V는 3.3V ÷ 1.00V = 3.3A/10A에 해당하며 8192*330/1023 = 2643/8002 bit/A. (STEP형)</p> <p>1A × 5mΩ × 50(앰프 증폭률) = 0.25V이므로 5.0V는 5.0V ÷ 0.25V = 20A에 해당하며 32768*2000/4095 = 16004. (Sensor-BLDC)</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;Sciiii1,iiii2; (+#^) ;Sc?; (+#^)	operation	전류제어 모드에서 0.01A 단위로 전류명령값을 설정함. 1st integer : 모터1 전류명령값 [-2000/2000 ~ 2000/2000, 2009년 7월 15일 S/W 버전부터는 comment 마지막부분 참조] (0.01A) 2nd integer : 모터2 전류명령값 [-2000/2000 ~ 2000/2000, 2009년 7월 15일 S/W 버전부터는 comment 마지막부분 참조] (0.01A)
	return value	Sciiii1,iiii2;
	comment	2009년 7월 15일 S/W 버전부터 Z3 명령어로 전류감지저항값이 정상적으로 설정되어 있어야함. 2009년 7월 15일 S/W 버전부터 IRMCK201을 사용한 경우는 Zc 명령어로 nominal current가 정상적으로 설정되어 있어야함. 2009년 7월 15일 S/W 버전부터 모든 F2811/F28334 H/W 버전에서는 다음과 같은 적용을 받음. KOMOTEK 380V모터의 경우 설정가능한 값의 범위는 전류감지저항으로 10m0hm을 사용하는 경우 -20A ~ +20A의 범위임. 그 외의 경우 설정가능한 값의 범위는 Z3 명령어에 의하여 설정되고 실제 장착된 전류감지저항에 따라서 다른데, 2m0hm은 -20A ~ +20A 범위이고 5m0hm은 -8A ~ +8A 범위를 이며 10m0hm은 -4A ~ +4A 범위임.

명령어	구분	설명
;SViiii1,iiii2; (%) ;SVIIII1,IIII2; (F2811/F28334인 경우) (+#^) ;SVrIIII1,IIII2; (F2811/F28334인 경우) (+#^) ;SV?: (+#^%)	operation	<p>속도제어 모드에서 속도명령값을 설정함.</p> <p>1st integer : 모터1 속도명령값 [-32000/60000 ~ 32000/60000] (0 rpm)</p> <p>2nd integer : 모터2 속도명령값 [-32000/60000 ~ 32000/60000] (0 rpm)</p>
	return value	SViiii1,iiii2;, SVIIII1,IIII2; (F2811/F28334인 경우)
	comment	<p>SV 명령의 경우에 스테핑모터가 아닐때 속도오차가 크면 속도명령값을 재설정함. 따라서 같은 속도 명령을 주기적으로 주는 경우에도 속도변화가 발생할수 있음.</p> <p>2010년 7월 19일 S/W 버전부터 속도오차가 클때 적분오차를 0으로 초기화하는 작동을 내부에서 수행함.</p> <p>2010년 11월 30일 S/W 버전부터 digital Hall IC 만을 사용하는 BLDC 모터에서 모터1가 정지상태에서 출발할때 위치값을 초기화하는 기능을 추가하였음.</p> <p>2012년 11월 04일 S/W 버전부터 digital Hall IC 만을 사용하는 BLDC 모터에서 모터2가 정지상태에서 출발할때 위치값을 초기화하는 기능을 추가하였음.</p> <p>2013년 3월 27일 S/W 버전부터 속도오차가 클때 적분오차를 0으로 초기화하는 속도오차의 크기를 WA 명령어로 설정할수있으며, SX 명령어 두 번째파라미터의 bit20으로 적분오차를 0으로 초기화하는 작동을 enable/disable 할수 있음.</p> <p>SVr 명령의 경우는 속도오차가 큰 경우에 속도명령값을 재설정하고 적분오차를 0으로 초기화하는 작동을 하지 않으므로, 속도 명령을 주기적으로 주는 경우에는 SVr 명령을 사용하여야함.</p> <p>SVr 명령의 경우는 2011년 9월 24일 S/W 버전부터 사용가능한 명령임.</p> <p>F2811/F28334에서는 long integer를 입력할수있으며 속도명령값의 설정가능 범위가 [-60000 ~ 60000] (rpm)임.</p> <p>2015년 1월 10일 S/W 버전부터 digital Hall 만을 사용하는 BLDC 모터에서 05번 속도제어중 정지후 출발할 때 위치값을 초기화하는지를 설정할수 있으며, SX 명령어 bit 23이 0이면 초기화를 하며 1이면 초기화하지 않음. 이전의 S/W 버전으로서 2010년 11월 30일 S/W 버전부터는 기능이 구현된 채널에 대하여 초기화함 기본임.</p>

명령어	구분	설명
;Saaaa1,aaaa2,d aaaa3,aaaa4; (+#^%) ;Sa?; (+#^%)	operation	<p>속도제어 및 위치제어 모드에서 가속 및 감속율을 설정함.</p> <p>1st word : 모터1 가속도 설정값 [1 ~ 60000] (9523 or 100 or 400 rpm/100ms or 1 rpm/ms)</p> <p>2nd word : 모터1 감속도 설정값 [1 ~ 60000] (9523 or 100 or 400 rpm/100ms or 1 rpm/ms)</p> <p>3rd word : 모터2 가속도 설정값 [1 ~ 60000] (9523 or 100 or 400 rpm/100ms or 1 rpm/ms)</p> <p>4th word : 모터2 감속도 설정값 [1 ~ 60000] (9523 or 100 or 400 rpm/100ms or 1 rpm/ms)</p>
	return value	Saaaa1,aaaa2,aaaa3,aaaa4;
	comment	<p>단위는 RJM_VER7/RJM_VER7에서는 rpm/ms, 나머지 H/W 버전에서는 rpm/100ms임.</p> <p>단 wafer 회전 제어기의 경우는 0.01rpm/ms 단위임.</p> <p>기본값은 BLDC RJM_VER3에서는 9523rpm/100ms, Wafer 회전제어기에서는 1rpm/100ms, 나머지 BLDC에서는 1 rpm/ms, STEP RJM_VER5는 400 rpm/100ms, 그외의 경우는 100 rpm/100ms임.</p> <p>RJM_VER3 BLDC의 경우는 RMCK201의 가감속율을 9523 RPM/100ms 이상, RJM_VER7의 03번모드에서는 1000 RPM/ms이상으로 충분히 커야 위치제어에서의 불안정성을 없앨 수 있음. 이러한 이유로 BLDC의 경우에 SM0303; 또는 SM0E0E; 명령을 수행할 때 일정값 이상이 되도록 강제로 바꿈.</p> <p>RJM_VER7에서 03/0E 작동모드에서는 Sa 명령에 의하여 가속 및 감속률을 설정하지는 않으나 위치제어 모드에서의 가감속에 필요한 펄스수는 계산함.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
(RJM_VER5/6 STEP형인 경우) ;SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (+#^%) ;SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7,iiii8; (+#^%) ;SH?; (+#^%) 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함.	operation	RJM_VER5/6 STEP형의 host direct electrical angle 제어 모드에서 구동전류의 진폭과 전류 OFFSET을 설정함. 1st integer : 모터1 구동전류 진폭 [1 ~ 2000] (200) 2nd integer : 모터2 구동전류 진폭 [1 ~ 2000] (200) 3rd integer : coil1 구동전류 offset [-800 ~ 800] 4th integer : coil2 구동전류 offset [-800 ~ 800] 5th integer : coil3 구동전류 offset [-800 ~ 800] 6th integer : coil4 구동전류 offset [-800 ~ 800] 7th integer : coil5 구동전류 offset [-800 ~ 800] 8th integer : coil6 구동전류 offset [-800 ~ 800]
	return value	SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (RJM_VER5 STEP형 만 해당됨) 또는 SHiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7,iiii8; (RJM_VER6 3상STEP형 만 해당됨)
	comment	RJM_VER5/6 STEP형에서 전류의 진폭과 전류 OFFSET은 310일때 1A에 해당함. RJM_VER5 STEP형을 2상스텝핑 모터의 구동기로 사용하는 경우는 coil1/2 가 한개의 모터가 되고 coil3/4 가 또 한개의 모터가 되어 총 2개의 스텝핑모터를 구동할 수 있음. RJM_VER5 STEP형을 3상스텝핑 모터의 구동기로 사용하는 경우는 coil1/2/3 가 해당코일이 되며, coil4는 단독 코일로서 구동전류 offset에 해당하는 전류가 흐르도록 제어됨. RJM_VER6 3상STEP형인 경우는 첫번째 모터는 coil1/2/3 가 해당코일이 되며 두번째 모터는 coil4/5/6 이 해당코일이 됨. 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함. F2811/F28334를 사용한 step motor 제어기(RJM_VER5)의 경우는 10mΩ으로 409.5 bit/A의 스케일을 갖음. step motor 제어기의 경우(RJM_VER5/RJM_VER6)는 1st interger 와 2nd interger는 ES 명령어로 EEPROM에 저장될 수 있음.

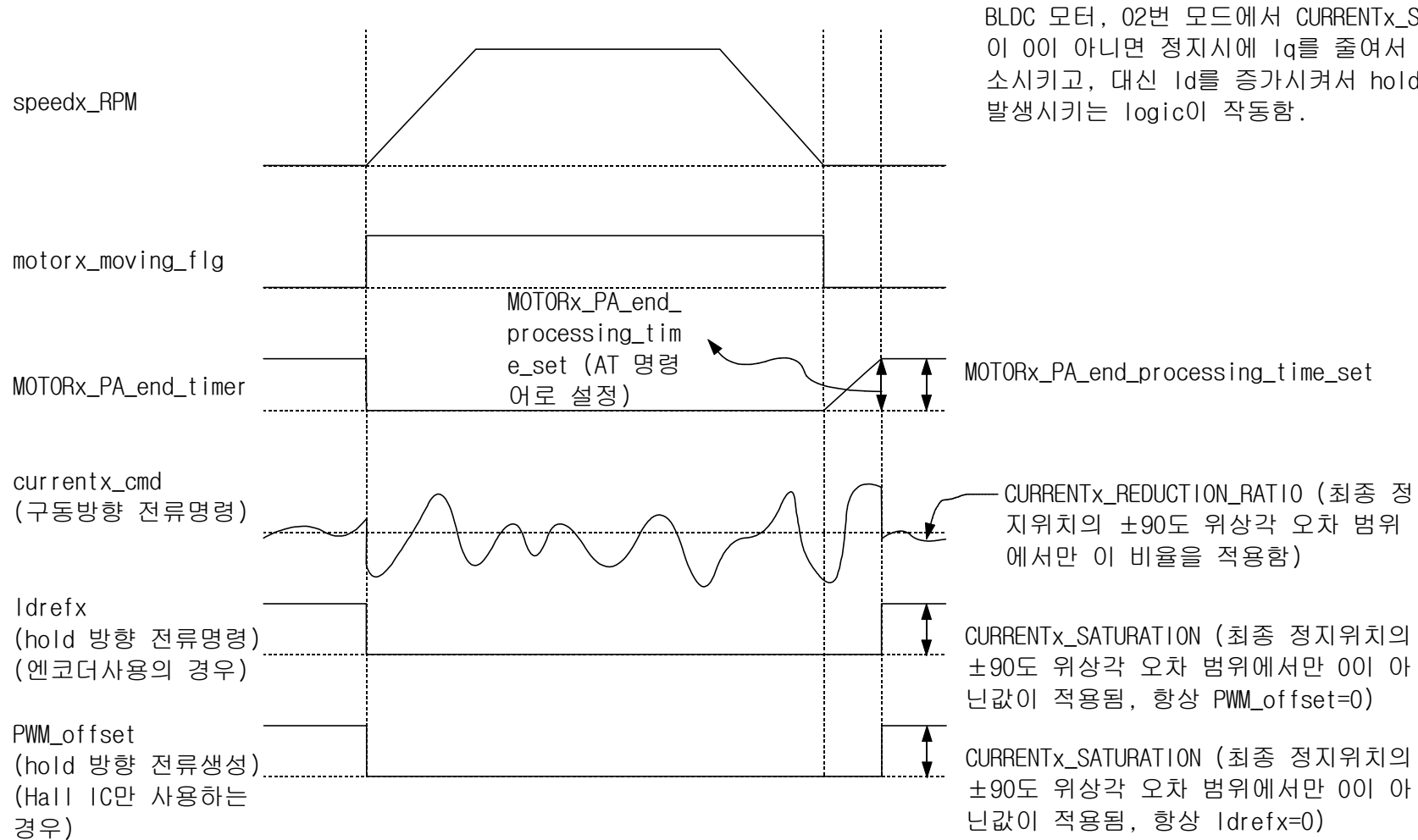
명령어	구분	설명
(RJM_VER5/6 STEP 형이 아닌 경우) ;SHiiii1,iiii2; (+#^%) ;SH?; (+#^%) 2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함.	operation	host direct electrical angle 제어 모드에서 구동전류 (RJM_VER7형의 3상/5상 STEP의 경우로서 작동모드 09번이 아닐때) 또는 구동전압의 진폭 (BLDC 구동 chip IRMCK201 경우) 또는 PWM duty (DC 모터를 제외한 기타의 경우)를 설정함. 1st integer : 모터1의 구동전류 [1 ~ 2000] 또는 구동전압 진폭 [-2048 ~ 2047] 또는 PWM duty [-PWM_PERIOD ~ +PWM_PERIOD] 2nd integer : 모터2의 구동전류 [1 ~ 2000] 또는 구동전압 진폭 [-2048 ~ 2047] 또는 PWM duty [-PWM_PERIOD ~ +PWM_PERIOD]
	return value	SHiiii1,iiii2;
	comment	2013년 4월 4일 S/W 버전부터 Sh 명령을 사용하여 0.01A 단위로 전류값설정이 가능함. RJM_VER7형의 3상/5상 step motor 제어기의 경우는 1st interger 와 2nd interger는 ES 명령어로 EEPROM에 저장될 수 있음.
(스테핑모터의 경우) ;Shdddd1,dddd2,dddd3,dddd4; (+#^%) ;Sh?; (+#^%) 2013년 4월 4일 S/W 버전부터 SK/Sk 명령으로 바뀜.	operation	스테핑모터에서 전류명령값의 최대치와 정지시의 전류치 비율을 설정함. dddd1 : CURRENT1_SATURATION [0 - 1000/2000] (300/400) dddd2 : CURRENT2_SATURATION [0 - 1000/2000] (300/400) dddd3 : CURRENT1_REDUCTION_RATIO [0 - 128] (8) dddd4 : CURRENT2_REDUCTION_RATIO [0 - 128] (8)
	return value	Shdddd1,dddd2,dddd3,dddd4;
	comment	CURRENT1/2_REDUCTION_RATIO는 정지시의 전류감소를 설정하는 파라미터로서, 128이면 정지시전류는 구동시전류의 100%, 64이면 정지시전류는 구동시전류의 50%, 8이면 정지시전류는 구동시전류의 6.25% 임. 전류진폭이 CURRENT1/2_SATURATION 값을 초과하는 전류진폭을 SH 명령어로 설정하더라도 실제 전류제어과정에서 전류의 진폭은 최대치로 제한됨. 따라서 진폭이 제한되는 경우에는 전류의 명령값이 sine파가 아니고 saturation 된 sine파가 됨. 2013년 4월 4일 S/W 버전부터 SK/Sk 명령으로 바뀜. (SK 명령은 2013년 4월 4일 이전의 S/W 버전에서 Sh 명령과 같고, Sk 명령은 0.01A단위로 CURRENT1_SATURATION과 CURRENT2_SATURATION 값을 설정하는 명령임.) 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
(Wafer control의 경우로서 2012년 2월 18일 S/W 버전부터)	operation	BLDC모터에서 holding torque를 만드는 파라미터를 설정함. dddd1 : CURRENT1_SATURATION [0 - 2000] (0) dddd2 : CURRENT2_SATURATION [0 - 2000] (0) dddd3 : CURRENT1_REDUCTION_RATIO [0 - 128] (32) dddd4 : CURRENT2_REDUCTION_RATIO [0 - 32767] (100)
	return value	Shddd1, dddd2, dddd3, dddd4;
;Shddd1, dddd2, dddd3, dddd4; (+#^) ;Sh?; (+#^) 2013년 4월 4일 S/W 버전부터 SK/Sk 명령어로 바뀜.	comment	CURRENT1_SATURATION는 정지(position1_cmd==position1_set)후 MOTORx_PA_end_processing_time_set[ms] (AT 명령어로 설정하는 시간값) 만큼의 시간이 경과한 이후에 적용하는 holding torque의 크기임. CURRENT2_SATURATION는 정지(position1_cmd==position1_set)후 CURRENT2_REDUCTION_RATIO[ms] 만큼의 시간이 경과한 이후에 적용하는 holding torque의 크기임. CURRENT1_REDUCTION_RATIO는 정지(position1_cmd==position1_set)후 CURRENT2_REDUCTION_RATIO[ms] 만큼의 시간이 경과한 후에 회전방향으로의 회전력을 감소시키는 비율임. (예로 64이면 1/2, 32이면 1/4임) WE 명령어 두번째 파라미터로 설정하는 값은 정지위치에서 필요한 holding 토크를 만드는 vector의 위상각 보정량임. 2013년 4월 4일 S/W 버전부터 SK/Sk 명령어로 바뀜. (SK 명령은 2013년 4월 4일 이전의 S/W 버전에서 Sh 명령과 같고, Sk 명령은 0.01A단위로 CURRENT1_SATURATION과 CURRENT2_SATURATION 값을 설정하는 명령임.) 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
(Wafer control이 아닌 BLDC 모터의 경우로서 2012년 2월 18일 S/W 버전부터)	operation	BLDC모터에서 holding torque를 만드는 파라미터를 설정함. dddd1 : CURRENT1_SATURATION [0 - 2000] (0) dddd2 : CURRENT2_SATURATION [0 - 2000] (0) dddd3 : CURRENT1_REDUCTION_RATIO [0 - 128] (32) dddd4 : CURRENT2_REDUCTION_RATIO [0 - 128] (32)
	return value	Shddd1, dddd2, dddd3, dddd4;
;Shddd1, dddd2, dddd3, dddd4; (+#^) ;Sh?; (+#^) 2013년 4월 4일 S/W 버전부터 SK/Sk 명령어로 바뀜.	comment	CURRENTx_SATURATION는 정지(position1_cmd==position1_set)후 MOTORx_PA_end_processing_time_set[ms] 만큼의 시간이 경과한 후에 적용하는 holding torque의 크기임. CURRENTx_REDUCTION_RATIO는 정지(position1_cmd==position1_set)후 MOTORx_PA_end_processing_time_set[ms] 만큼의 시간이 경과한 후에 회전방향으로의 회전력을 감소시키는 비율임. (예로 64이면 1/2, 32이면 1/4임) 2013년 4월 4일 S/W 버전부터 SK/Sk 명령어로 바뀜. (SK 명령은 2013년 4월 4일 이전의 S/W 버전에서 Sh 명령과 같고, Sk 명령은 0.01A단위로 CURRENT1_SATURATION과 CURRENT2_SATURATION 값을 설정하는 명령임.) 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

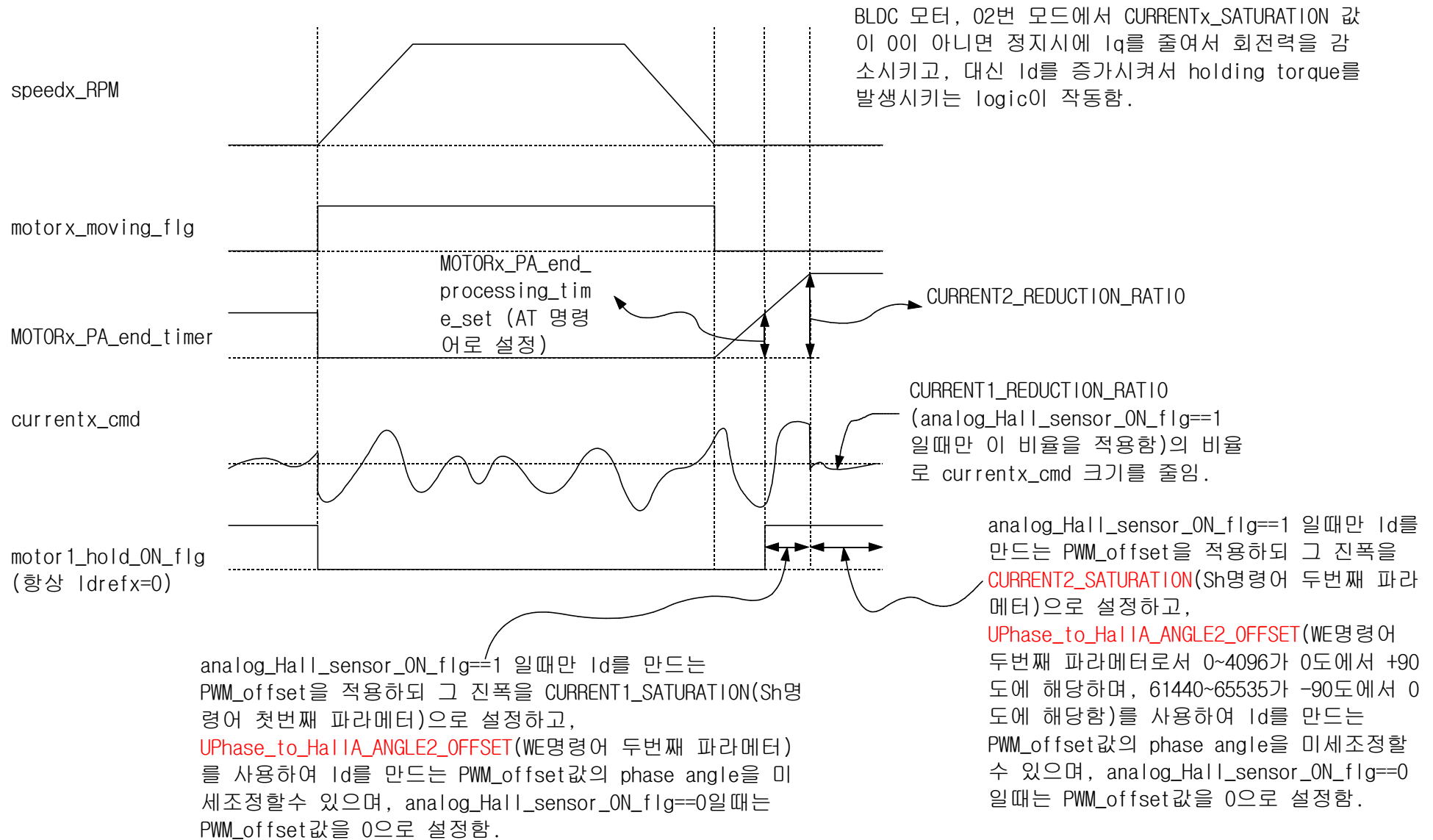
BLDC 모터에서 정지시 정지토크의 부여 (wafer control이 아닐때 - 2012년 2월 18일 S/W 버전부터

적용)



BLDC 모터, 02번 모드에서 CURRENTx_SATURATION 값이 0이 아니면 정지시에 Iq를 줄여서 회전력을 감소시키고, 대신 Id를 증가시켜서 holding torque를 발생시키는 logic이 작동함.

BLDC 모터에서 정지시 정지토크의 부여 (wafer control일때 - 2012년 2월 18일 S/W 버전부터 적용)



명령어	구분	설명
;SPdddd1, llll2, llll3; (+#^%) ;SP?dddd1; (+#^%)	operation	좌표점을 정의함. dddd1 : point number [0 - 151] llll2 : 모터1의 위치값 [position1_minus_limit ~ position1_plus_limit] llll3 : 모터2의 위치값 [position2_minus_limit ~ position2_plus_limit]
	return value	SPdddd1, llll2, llll3;
	comment	이렇게 설정된 좌표점은 DSP의 SARAM에 저장되며 만약 EEPROM에 저장하려면 ESxxxx;명령어를 사용하여야 함. 위치값은 상하한리밋범위 내에 있어야 유효함. 2012년 12월 03일 S/W 버전부터 좌표점의 개수가 192개에서 168개로 줄었음. 2013년 4월 14일 S/W 버전부터 좌표점의 개수가 168개에서 152개로 줄었음. 자석가동자의 경우 사용할수 없음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Sfdddd1, dddd2; (+#^) ;Sf?: (+#^)	operation	위치제어의 feedforward 게인을 설정함. 1st word : 모터1의 feedforward 게인 [0 - 10] (0) 2nd word : 모터2의 feedforward 게인 [0 - 10] (0)
	return value	Sfdddd1, dddd2;
	comment	매 1ms마다 (positionx_set - position1_set_old)에 feedforward 게인을 곱한 값을 위치명령값으로 설정함. 이렇게 함으로써 동적 상태에서 위치오차를 줄일수있음. 기본값은 0이며 게인이 크면이 크면 진동이 발생할 우려가 큼. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;SeHLHL; (+#^%)	operation	모터1 과 모터2 에 대하여 PM, PP, PQ, Pp, Pq command를 적용가능토록 설정함. 0th bit : motor1_start_to_move_trajectory_flg 가 set (by PM) 가능하도록 설정함. (1) 1st bit : motor2_start_to_move_trajectory_flg 가 set (by PM) 가능하도록 설정함. (1) 4th bit : motor1_start_to_move_to_position_flg 가 set (by PP, PQ, Pp, Pq) 가능하도록 설정함. (1) 5th bit : motor2_start_to_move_to_position_flg 가 set (by PP, PQ, Pp, Pq) 가능하도록 설정함. (1)
	return value	SeHLHL;
;Se?; (+#^%)	comment	이 설정값은 DSP의 SARAM에 저장되며 만약 EEPROM에 저장하려면 ESxxxx;명령어를 사용하여야함. 기본값은 0x0033임. 해당 bit가 세트되지 않으면 해당 모터는 PM 또는 PP, PQ, Pp, Pq 명령에 의한 작동이 되지 않음. 해당 bit가 세트되어 있어도 해당모터의 모드가 속도 모드이면 motorx_start_to_move_trajectory_flg, motorx_start_to_move_to_position_flg 는 세트되지 않음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
	operation	위치제어에서 정지위치 근방에서 강성을 높이기 위하여 제어 게인을 높게 하는 비율을 설정함. 1st word : 모터1의 게인 증가비율 [1 - 10] (1) 2nd word : 모터2의 게인 증가비율 [1 - 10] (1)
;Sudddd1,dddd2; (+#^)	return value	Sudddd1,dddd2;
	comment	설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.
;Su?; (+#^)		

명령어	구분	설명
;SQHLHLHLHL,HLHL HLHL,HLHLHLHL,HL HL,HLHLHLHL,HLH L,HLHL,HLHLHLHL, HLHL,HLHLHLHL; (%) ;SQ 또는 ;SqHLHLHLHL,HLHL HLHL,HLHLHLHL,HL HL,HLHLHLHL,HLH L,HLHL,HLHLHLHL, HLHL,HLHL,HLHLHL HL; (F2811/F28334인 경우) (+#^) ;SQ?; (+#^%) 또 는 ;Sq?; (F2811/F28334인 경우만) (+#^) 	operation	<p>모터1의 파라미터 리밋과 위치/속도/전류 제어파라미터를 설정함. (WQ 명령어로 속도 및 전류제어 파라미터를 설정하는 경우는 제외)</p> <p>1st long word : MOTOR1_ERROR_SIZE_LIMIT (MOTOR1_PULSE_PER_REV * 10)</p> <p>2nd long word : MOTOR1_INTEGRAL_START_ERROR_SIZE (MOTOR1_PULSE_PER_REV * 4)</p> <p>3rd long word : MOTOR1_INTEGRAL_LIMIT (MOTOR1_PULSE_PER_REV * 100)</p> <p>4th word : MOTOR1_INVERSE_OF_P_GAIN (MOTOR1_PULSE_PER_REV*1000L)/(5L*PWM_PERIOD)</p> <p>5th long word : MOTOR1_INVERSE_OF_I_GAIN (MOTOR1_INTEGRAL_LIMIT*100L)/(50L*PWM_PERIOD)</p> <p>6th word : MOTOR1_INVERSE_OF_D_GAIN (MOTOR1_PULSE_PER_REV*(MOTOR1_SPEED_MAX/60)*1000)/(250*PWM_PERIOD)</p> <p>7th word : 속도제어시 MOTORv1_INVERSE_OF_P_GAIN</p> <p>8th long word : 속도제어시 MOTORv1_INVERSE_OF_I_GAIN</p> <p>9th word : 속도제어시 MOTORv1_INVERSE_OF_D_GAIN (F2811/F28334인 경우만)</p> <p>10th word : 전류제어시 MOTORi1_INVERSE_OF_P_GAIN : 자동화용 H/W 버전에서는 외부 analog 입력명령의 영점 offset 값.</p> <p>11th long word : 전류제어시 MOTORi1_INVERSE_OF_I_GAIN : 자동화용 H/W 버전에서는 외부 analog 입력명령의 게인 값.</p>
	return value	SQHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL,HLHLHLHL; SQ 또는 SqHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL; (F2811/F28334인 경우만)
	comment	<p>Sq 명령일때는 게인값들은 50000/inverse_gain의 값들을 의미함.</p> <p>((RJM_VER7_ENCODER RJM_VER7_ANALOG)==1)&&(TPC_FOCUSING==1) 인 경우에는 SQ 명령어로 설정하는 마지막 2개의 파라미터가 각각 외부 입력전압으로부터 전류명령값으로 변환하는 offset과 게인의 의미를 가짐.</p> <p>10/11th word의 경우 자동화용 H/W 버전의 BLDC 모터 제어기에서는 (이 경우 전류게인은 WQ/WR 명령어로 설정함) 외부에서 인가되는 속도명령 또는 전류명령 전압값을 내부의 속도명령 및 전류명령값으로 변환하는데 있어서 각각 offset과 게인설정용 파라미터로 사용함.</p> <p>2013년 3월 7일 S/W 버전부터 5th-8th/6th-9th 값이 0이면 위치-속도제어에서 적분/미분제어를 사용않음. 저가형 1채널 모터제어기에는 해당없으며, 대신 AJ 명령과 WQ 명령을 참조할것.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
<pre> ;SRHLHLHLHL,HLHL HLHL,HLHLHLHL,HL HL,HLHLHLHL,HLH L,HLHL,HLHLHLHL, HLHL,HLHLHLHL; (%) ;SR 또는 ;SpHLHLHLHL,HLHL HLHL,HLHLHLHL,HL HL,HLHLHLHL,HLH L,HLHL,HLHLHLHL, HLHL,HLHL,HLHLHL HL; (F2811/F28334인 경우) (+#^) ;SR?; (+#^%) 또 는 ;Sp?; (F2811/F28334인 경우만) (+#^) </pre>	<p>operation</p> <p>return value</p> <p>comment</p>	<p>모터2의 파라미터 리밋과 위치/속도/전류 제어파라미터를 설정함. (WR 명령어로 속도 및 전류제어 파라미터를 설정하는 경우는 제외)</p> <p>1st long word : MOTOR2_ERROR_SIZE_LIMIT (MOTOR2_PULSE_PER_REV * 10)</p> <p>2nd long word : MOTOR2_INTEGRAL_START_ERROR_SIZE (MOTOR2_PULSE_PER_REV * 4)</p> <p>3rd long word : MOTOR2_INTEGRAL_LIMIT (MOTOR2_PULSE_PER_REV * 100)</p> <p>4th word : MOTOR2_INVERSE_OF_P_GAIN (MOTOR2_PULSE_PER_REV*1000L)/(5L*PWM_PERIOD)</p> <p>5th long word : MOTOR2_INVERSE_OF_I_GAIN (MOTOR2_INTEGRAL_LIMIT*100L)/(50L*PWM_PERIOD)</p> <p>6th word : MOTOR2_INVERSE_OF_D_GAIN (MOTOR2_PULSE_PER_REV*(MOTOR2_SPEED_MAX/60)*1000)/(250*PWM_PERIOD)</p> <p>7th word : 속도제어시 MOTORv2_INVERSE_OF_P_GAIN</p> <p>8th long word : 속도제어시 MOTORv2_INVERSE_OF_I_GAIN</p> <p>9th word : 속도제어시 MOTORv2_INVERSE_OF_D_GAIN (F2811/F28334인 경우만)</p> <p>10th word : 전류제어시 MOTORi2_INVERSE_OF_P_GAIN</p> <p>11th long word : 전류제어시 MOTORi2_INVERSE_OF_I_GAIN</p> <p>SRHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL,HLHLHLHL; SR 또는 SpHLHLHLHL,HLHLHLHL,HLHLHLHL,HLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL; (F2811/F28334인 경우만)</p> <p>Sp 명령일때는 게인값들은 50000/inverse_gain의 값을 의미함. ((RJM_VER7_ENCODER RJM_VER7_ANALOG)==1)&&(TPC_FOCUSING==1) 인 경우에는 SR 명령어로 설정하는 마지막 2개의 파라미터가 각각 외부 입력전압으로부터 전류명령값으로 변환하는 offset과 게인의 의미를 가짐. (KOMOTEK_380V==1)&&(MOVING_MAGNET==1) 일때 SR 명령어로 설정하는 마지막에서 두번째 파라미터가 left analog Hall 센서에 대한 right analog Hall 센서의 위상차를 설정하는 값이며, 마지막 파라미터가 순위가 높은 제어기의 right analog Hall 센서에 대한 현 제어기의 left analog Hall 센서의 위상차를 설정하는 값임.</p> <p>10/11th word의 경우 자동화용 H/W 버전의 BLDC 모터 제어기에서는 (이 경우 전류게인은 WQ/WR 명령어로 설정함) 외부에서 인가되는 속도명령 또는 전류명령 전압값을 내부의 속도명령 및 전류명령값으로 변환하는데 있어서 각각 offset과 게인설정용 파라미터로 사용함.</p> <p>2013년 3월 7일 S/W 버전부터 5th-8th/6th-9th 값이 0이면 위치-속도제어에서 적분/미분제어를 사용하지 않음. 저가형 1채널 모터제어기에는 해당없음.</p>
		<p>설정된 값은 SS 명령어로 EEPROM에 저장될 수 있음.</p> <p>Copyright © 2008-2010 RoboCubeTech. All Rights Reserved.</p> <p>367</p>

명령어	구분	설명
;AQ 또는 ;AqHLHL,HLHLHLHL, HLHL,HLHL,HLHL, HLHL,HLHL; (+#^)	operation	모터1의 위치/속도 제어파라미터를 설정함. (WQ 명령어로 속도제어 파라미터를 설정하는 경우는 제외) 1st word : MOTOR1_INVERSE_OF_P_GAIN (MOTOR1_PULSE_PER_REV*1000L)/(5L*PWM_PERIOD) 2nd long word : MOTOR1_INVERSE_OF_I_GAIN (MOTOR1_INTEGRAL_LIMIT*100L)/(50L*PWM_PERIOD) 3rd word : MOTOR1_INVERSE_OF_D_GAIN (MOTOR1_PULSE_PER_REV*(MOTOR1_SPEED_MAX/60)*1000)/(250*PWM_PERIOD) 4th word : 속도제어시 MOTORv1_INVERSE_OF_P_GAIN 5th long word : 속도제어시 MOTORv1_INVERSE_OF_I_GAIN 6th word : 속도제어시 MOTORv1_INVERSE_OF_D_GAIN
	return value	AQ 또는 AqHLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL;
	comment	Aq 명령일때는 게인값들은 50000/inverse_gain의 값들을 의미함. 2013년 3월 7일 S/W 버전부터 2nd-5th/3rd-6th 값이 0이면 위치-속도제어에서 적분/미분제어를 사용않음. 저가형 1채널 모터제어기에는 해당없으며, 대신 AJ 명령과 WQ 명령을 참조할것. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;AR 또는 ArHLHL,HLHLHLHL, HLHL,HLHL,HLHLHL HL,HLHL; (+#^)	operation	모터2의 위치/속도 제어파라미터를 설정함. (WR 명령어로 속도제어 파라미터를 설정하는 경우는 제외) 1st word : MOTOR2_INVERSE_OF_P_GAIN (MOTOR2_PULSE_PER_REV*1000L)/(5L*PWM_PERIOD) 2nd long word : MOTOR2_INVERSE_OF_I_GAIN (MOTOR2_INTEGRAL_LIMIT*100L)/(50L*PWM_PERIOD) 3rd word : MOTOR2_INVERSE_OF_D_GAIN (MOTOR2_PULSE_PER_REV*(MOTOR2_SPEED_MAX/60)*1000)/(250*PWM_PERIOD) 4th word : 속도제어시 MOTORv2_INVERSE_OF_P_GAIN 5th long word : 속도제어시 MOTORv2_INVERSE_OF_I_GAIN 6th word : 속도제어시 MOTORv2_INVERSE_OF_D_GAIN
	return value	AR 또는 ArHLHL,HLHLHLHL,HLHL,HLHL,HLHLHLHL,HLHL;
	comment	Ar 명령일때는 게인값들은 50000/inverse_gain의 값들을 의미함. 2013년 3월 7일 S/W 버전부터 2nd-5th/3rd-6th 값이 0이면 위치-속도제어에서 적분/미분제어를 사용않음. 저가형 1채널 모터제어기에는 해당없음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;AS 또는 ;AsHLHL,HLHLHLHL L,HLHL,HLHLHLHL; (+#^) ;AS?; (+#^) 또는 ;As?; (+#^)	operation	모터1,2의 전류 제어파라미터를 설정함. (WQ, WR 명령어로 전류제어 파라미터를 설정하는 경우는 제외) 1st word : 전류제어시 MOTORi1_INVERSE_OF_P_GAIN 2nd long word : 전류제어시 MOTORi1_INVERSE_OF_I_GAIN 3rd word : 전류제어시 MOTORi2_INVERSE_OF_P_GAIN 4th long word : 전류제어시 MOTORi2_INVERSE_OF_I_GAIN
	return value	AS 또는 AsHLHL,HLHLHLHL,HLHL,HLHLHLHL;
	comment	As 명령일때는 게인값들은 50000/inverse_gain의 값들을 의미함. ((RJM_VER7_ENCODER RJM_VER7_ANALOG)==1)&&(TPC_FOCUSING==1) 인 경우에는 AS 명령어로 설정하는 파라 미터가 각각 외부 입력전압으로부터 전류명령값으로 변환하는 offset과 게인의 의미를 가짐. 자동화용 H/W 버전의 BLDC 모터 제어기에서는 (이 경우 전류게인은 WQ/WR 명령어로 설정함) 외부에서 인가 되는 속도명령 또는 전류명령 전압값을 내부의 속도명령 및 전류명령값으로 변환하는데 있어서 각각 offset과 게인설정용 파라미터로 사용함. (SQ/SR 명령어 참조) 저가형 1채널 모터제어기에는 해당없음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;AJddd1,ddd2,d ddd3,ddd4,ddd 5,ddd6; (+#^) ;AJ?; (+#^)	operation	저가형 1채널 모터제어기에서 모터1의 위치/속도 제어파라미터를 설정함. 1st word : POSITION_P_gain1 2nd word : POSITION_I_gain1 3rd word : POSITION_D_gain1 4th word : SPEED_P_gain1 5th word : SPEED_I_gain1 6th word : SPEED_D_gain1
	return value	AJddd1,ddd2,ddd3,ddd4,ddd5,ddd6;
	comment	저가형 1채널 모터제어기에서만 적용됨. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

모터 제어파라미터의 설정과 저장

☞ 모터의 제어파라미터를 설정하고 저장하려면

- 1) SQ, Sq, SR, Sp, WQ, WR 명령어를 사용하여 제어게인을 바꾸어줌. (이때 SQ?;, Sq?; SR?;, Sp?; WQ?; WR?; 를 사용하여 현재값을 읽어내어 조금씩 바꾸는 형식으로 실시할 것)
- 2) ESOA00,0100; 명령 또는 EsA55A; 을 실행하여 EEPROM에 파라미터를 저장함.
- 3) 전원을 껐다가 켜.
- 4) SQ?;, Sq?; SR?;, Sp?; WQ?; WR?; 등을 사용하여 바뀐 파라미터가 반영되었는지를 확인함.

제어기 H/W 버전	모터 종류	위치제어 PID 게인 (01번, 02번, 03번 위치제어시에 사용)	속도제어 PID 게인 (05번 속도제어시에 사용)	속도제어 PID 게인 (06번 속도제어시에 사용)	전류제어 PI 게인 (07/08번 전류제어시에 사용)	비고		
RJM_VER 1,2,3(2406a)	DC	SQ/Sq 또는 SR/Sp 명령어의 4/5/6번째 파라미터		SQ/Sq 또는 SR/Sp 명령어의 7/8/9번째 파라미터	SQ/Sq 또는 SR/Sp 명령어의 마지막 2개 파라미터 (1KHz 제어주기)			
RJM_VER 3(2811),4,8					SQ/Sq 또는 SR/Sp 명령어의 마지막 2개 파라미터 (02번 위치제어시에 사용) (05번 속도제어시에도 사용)			
RJM_VER 3	BLDC			SQ/Sq 또는 SR/Sp 명령어의 7/8/9번째 파라미터	WQ 또는 WR 명령어의 앞쪽 2개 파라미터 (03번 위치제어시에도 사용)	WQ 또는 WR 명령어의 마지막 2개 파라미터 (02번 위치제어시에도 사용) (05번 속도제어시에도 사용)		
RJM_VER 7					WQ 또는 WR 명령어의 앞쪽 3개 파라미터 (03번 위치제어시에도 사용)			
RJM_VER 5,6	STEP						SQ/Sq 또는 SR/Sp 명령어의 마지막 2개 파라미터	

명령어	구분	설명
;AI dddd1, dddd2, d ddd3, dddd4; (+ # ^) ;AI?; (+ # ^)	operation	PA 또는 SV 명령에 의한 가감속시에 전류명령값의 feedforward 값을 계산하는 scale factor를 설정함. 1st word : 모터1의 가속시 전류명령값 offset 계산용 scale factor [0 - 10000] (0) 2nd word : 모터1의 감속시 전류명령값 offset 계산용 scale factor [0 - 10000] (0) 3rd word : 모터2의 가속시 전류명령값 offset 계산용 scale factor [0 - 10000] (0) 4th word : 모터2의 감속시 전류명령값 offset 계산용 scale factor [0 - 10000] (0)
	return value	AI dddd1, dddd2, dddd3, dddd4;
	comment	(motorx의 전류명령값 offset 계산용 scale factor) * 가감속도값 / 4096 의 크기를 가감속시의 전류 feedforward 값으로 설정함. Qx114/115; 를 사용하여 가감속시에 4번째값(전류명령 offset)이 3번째값(전류값)의 평균값이 되도록 AI 명령에 의한 파라미터를 수정함. Qx113; 으로 가감속을 확인할수 있음. 02/03번 위치제어나 05번 속도제어에서만 사용이 가능함. 2011년 11월 22일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.
;Ai dddd1, dddd2; (+ # ^) ;Ai?; (+ # ^)	operation	PA 명령에 의한 가속시에 사다리꼴 모양의 가속 프로파일을 만들기 위한 가속 변화율값을 설정함. 1st word : 모터1의 가속시 1ms당 가속변화량 [1 - 65535] (65535) 2nd word : 모터2의 가속시 1ms당 가속변화량 [1 - 65535] (65535)
	return value	Ai dddd1, dddd2;
	comment	단위크기는 (1000 * 1000 * 60) / (SPEED_SCALE_FACTOR * MOTOR1_PULSE_PER_REV) RPM/sec에 해당함. 1보다 작은값의 설정이 불가하며, 큰값을 주는 경우는 사각모양의 가속 프로파일에 해당하는 가속을 실시함. Qx114/115; 의 네번째값으로 가감속 변화를 확인할수 있음. 2012년 1월 30일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;ANddd1,ddd2; (+#^) ;AN?; (+#^)	operation	모터1에 적용되는 Notch filter의 중심주파수와 Q factor를 설정함. 1st word : 모터1의 Notch filter의 중심주파수 [0 - 5000] (500 0.1Hz) 2nd word : 모터1의 Notch filter의 Q factor [0 - 100] (0 0.1)
	return value	ANddd1,ddd2;
	comment	전류명령값을 계산하고나서 MOTOR1_NOTCH_FILTER1_Q!=0 이면 notch filter를 적용하여 전류명령값을 최종 결정함. 2011년 11월 24일 S/W 버전부터 사용가능한 명령임. 2014년 3월 9일 이전 버전에서는 노치필터가 제어부 결과값 (PWM duty 값)에 적용하였으며, 2014년 3월 9일 버전부터는 노치필터가 제어오차에 적용되는 것으로 바뀜. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.
;Anddd1,ddd2; (+#^) ;An?; (+#^)	operation	모터2에 적용되는 Notch filter의 중심주파수와 Q factor를 설정함. (저가형 1채널 제어기에서는 모터1에 적용됨) 1st word : 모터2의 Notch filter의 중심주파수 [0 - 5000] (500 0.1Hz) 2nd word : 모터2의 Notch filter의 Q factor [0 - 100] (0 0.1)
	return value	Anddd1,ddd2;
	comment	전류명령값을 계산하고나서 MOTOR2_NOTCH_FILTER1_Q!=0 이면 notch filter를 적용하여 전류명령값을 최종 결정함. 2011년 11월 24일 S/W 버전부터 사용가능한 명령임. 2014년 3월 9일 이전 버전에서는 노치필터가 제어부 결과값 (PWM duty 값)에 적용하였으며, 2014년 3월 9일 버전부터는 노치필터가 제어오차에 적용되는 것으로 바뀜. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;ALddd1,ddd2,ddd3,ddd4; (#)	operation	28~31번 butterworth filterA의 시정수를 설정함. 1st word : 28번 butterworth filterA의 시정수. [0 - 4096] (0 ms) 2nd word : 29번 butterworth filterA의 시정수. [0 - 4096] (0 ms) 3rd word : 30번 butterworth filterA의 시정수. [0 - 4096] (0 ms) 4th word : 31번 butterworth filterA의 시정수. [0 - 4096] (0 ms)
	return value	ALddd1,ddd2,ddd3,ddd4;
;AL?; (#)	comment	butterworth filterA는 2차 필터로서 0-31 까지 총 32개가 있으며 1ms 마다 filter가 동작함. 시정수 설정값을 2로 나눈 값이 ms 단위의 실제 시정수에 해당함. 0인 경우는 필터링하지 않음. 28/30번 butterworth filterA는 02번 제어모드에서 모터1/2의 전류명령값을 생성할때 low pass filter로 사용함. 2012년 8월 14일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;ATdddd1; (+#^) ;AT?; (+#^)	operation	PA 명령의 완료 시점에서 추가로 실행하는 기능을 위한 시간파라미터를 설정함. 1st word : PA 명령에 의한 위치이동 종료후 계인을 유지하는 기간 [1 - 30000] (100 ms) (RJM_VER7_ENCODER==1)&&(ELECTRONIC_POWER_STEER==1) 일때는 가감속 완료후 오차의 적분값을 강제로 0이 되게하기 까지의 경과시간임.
	return value	ATdddd1;
	comment	이동중의 PID 계인이 PA 명령에 의한 위치이동 종료후에도 일정시간 유지되며, 설정시간이 경과된 이후에는 비례, 적분 및 미분계인을 ABA55A 명령으로 설정된 비율로 낮추거나 높임. 새로운 PA 명령어로 이동을 시작하면 원래의 계인으로 자동 전환됨. 이렇게 함으로서 정지시의 흔들림이나 강성을 조절할수 있음. 2011년 12월 10일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.
;ABA55A,dddd1; (+#^) ;AB?; (+#^)	operation	위치제어에서 정지하고 있을때, jitter와 소음을 줄이거나 또는 강성을 높이기 위하여, 위치제어 계인을 낮추거나 높이기 위한 파라미터를 설정함. 2012년 1월 11일 S/W 버전부터 사용가능한 명령임. 1st word : 정지시 계인을 낮추거나 높이기 위한 파라미터값 [0 - 800] (0)
	return value	ABdddd1;
	comment	이동중의 PID 계인이 PA 명령에 의한 위치이동 종료후에도 일정시간 유지되며, 설정시간이 경과된 이후에는 비례, 적분 및 미분계인을 ABA55A 명령으로 설정된 비율로 낮추거나 높여짐. 101~200, 201~300, 301~400, 401~500, 501~600, 601~700, 701~800 의 값은 1~100과 같게 취급되며, 0이면 정지시 계인의 조정이 없고, 1/10/100이면 0.1/1/10 배의 계인이 되도록 하는 것임. 201~300, 301~400, 601~700, 701~800 인 경우는 정지시 추가로 전류계인을 반으로 줄이며, 401~500, 501~600, 601~700, 701~800인 경우는 정지시 추가로 미분계인을 반으로 줄임. 새로운 PA 명령어로 이동을 시작하면 원래의 계인으로 자동 전환됨. 이렇게 함으로서 정지시의 흔들림이나 강성을 조절할수 있음. (KOMOTEK_380V & AUTOMATION)==1 이거나 ((RJM_VER7_ENCODER RJM_VER7_ANALOG) & AUTOMATION)==1 일때는 ABA55A 명령에 의한 값이 101~200, 301~400, 501~600, 701~800 범위에 있을때는 정지시에 01번 제어모드로 전환하며 (이때 KOMOTEK_380V==1이면 공급전압에 따라서 ABA55A 명령에 의한 값을 줄임), 이동중에는 02번 보드로 이동함. 01번 제어모드일때의 PWM 크기 최대치는 계인과 같은 비율로 크거나 작게 되며 단 25% 이하이거나 75%이상의 duty로 설정되지는 않음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;AWdddd1,dddd2; (+#^) ;AW?; (+#^)	operation	BLDC 모터에서 최대속도를 높이기 위한 field weakening 비례제어 게인을 설정함. 1st word : 모터1의 field weakening 비례제어게인 [0 - 100] (0 bit) 2nd word : 모터2의 field weakening 비례제어게인 [0 - 100] (0 bit)
	return value	AWdddd1,dddd2;
	comment	field weakening 비례제어게인 값이 0이면 field weakening 비례제어를 하지 않는 기존방식과 같음. 이 기능을 사용한다면 비례게인값을 8정도에서 시작하는 것이 좋으며, Qx78/79;를 사용하며 제어상황을 모니터링할 수 있음. Qx78/79; 그래프에서 field weakening이 적용되는 때는 두번째값이 증가하고, 세번째값은 감소하며, 네번째값은 세번째값에 의하여 제한됨. 위치제어에서 field weakening 제어를 사용하면 위치명령 추종에 문제가 생길 수 있음. SX 명령어 리턴값의 첫번째파라미터 상위에서 3번째/4번째 비트가 1이면 (block commutation 사용) 적용되지 않으며, SG 명령어에 의한 MOTORx_SPEED_MAX 값의 1/4보다 큰 속도에서만 field weakening을 실시함. 2013년 2월 25일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.
;AwA55A,dddd1,dd dd2; (+#^) ;Aw?; (+#^)	operation	사용가능한 최대크기의 PWM duty 비를 설정함. 1st word : 모터1의 사용가능한 최대크기의 PWM duty 비 [10 - 100] (100 %) 2nd word : 모터2의 사용가능한 최대크기의 PWM duty 비 [10 - 100] (100 %)
	return value	Awdddd1,dddd2;
	comment	2012년 7월 10일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;ADA55A,iiii1,ii ii2; (+#^) ;AD?; (+#^)	operation	MOTOR 구동 Half-bridge의 dead-time을 보상하되, Z6 명령어로 설정하는 dead-time에 추가하여 보상하려는 보상량을 설정함. 1st word : 모터1의 dead-time을 추가보상하는 보상량 $[-\alpha \sim PWM_PERIOD/8]$ ($-\alpha$ 8.33ns) 2nd word : 모터2의 dead-time을 추가보상하는 보상량 $[-\alpha \sim PWM_PERIOD/8]$ ($-\alpha$ 8.33ns)
	return value	ADiiii1,iiii2; (2013년 5월 20일 이전 S/W 버전) ADiiii1,iiii2,iiii3,iiii4; (2013년 5월 20일 S/W 버전부터) 3rd word : 모터1의 PWM 구동회로에서 dead_time generator에 설정된 값, 단위크기는 8.33ns 4th word : 모터2의 PWM 구동회로에서 dead_time generator에 설정된 값, 단위크기는 8.33ns
	comment	추가보상하는 보상량이 0인 경우는 Z6 명령어로 설정하는 dead-time 만큼의 보상을 하는 경우에 해당함. 추가보상하는 보상량의 하한값 $-\alpha$ 는 Z6 명령어로 설정하는 dead-time의 음의 값임. 이 경우에는 실질적인 dead-time 보상은 0임. Z6 명령어로 설정하는 dead-time 값이 0인 경우(모터드라이브 IC 또는 gate drive IC에서 dead-time을 부여하는 경우)는 추가보상하는 보상량의 범위가 $-PWM_PERIOD/16 \sim PWM_PERIOD/8$ 이며 기본값은 0임. 2012년 8월 14일 S/W 버전부터 사용가능한 명령어나 컴파일시에 option을 선택한 경우에 한하며, 통상의 경우에는 사용할수 없음. dead-time 보상에 CPU time을 많이 소모하므로 PWM 주파수를 60KHz 이내에서 사용하여야함. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 에서만 가능한 명령임.

명령어	구분	설명
;Acddd1,ddd2; (+)	operation	엔코더펄스 카운트 방식으로 속도를 계산할 때 moving average 구간의 크기를 설정함. 1st word : 모터1의 속도를 계산할 때 moving average 구간의 크기 [1 ~ 10] (10 ms) 2nd word : 모터2의 속도를 계산할 때 moving average 구간의 크기 [1 ~ 10] (10 ms)
	return value	Acddd1,ddd2;
;Ac?; (+)	comment	엔코더펄스를 카운트하여 속도를 계산하는데, moving average를 이용하여 속도계산값을 필터링함. 첫번째와 두번째 파라미터값을 작은 값으로 지정할수록 속도계산값의 분해능은 나빠지나 시간지연은 감소함. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811 에서만 가능한 명령임.
;Acddd1,ddd2,ddd3,ddd4; (#)	operation	엔코더펄스 카운트 방식으로 속도를 계산할 때 moving average 구간의 크기와, 엔코더펄스의 주기를 사용하여 속도를 계산할 때 주기계산에 사용되는 엔코더펄스개수의 2에 대한 지수를 설정함. 1st word : 모터1의 속도를 계산할 때 moving average 구간의 크기 [1 ~ 10] (10 ms) 2nd word : 모터2의 속도를 계산할 때 moving average 구간의 크기 [1 ~ 10] (10 ms) 3rd word : 모터1의 주기계산에 사용되는 엔코더펄스 개수의 2에 대한 지수 [2 ~ 11] (2 multiple_of_2) 4th word : 모터2의 주기계산에 사용되는 엔코더펄스 개수의 2에 대한 지수 [2 ~ 11] (2 multiple_of_2)
	return value	Acddd1,ddd2,ddd3,ddd4;
;Ac?; (#)	comment	엔코더펄스를 카운트하여 속도를 계산하는데, moving average를 이용하여 속도계산값을 필터링함. 첫번째와 두번째 파라미터값을 작은 값으로 지정할수록 속도계산값의 분해능은 나빠지나 시간지연은 감소함. 세번째와 네번째 파라미터값에 있어서 엔코더펄스 개수의 2에 대한 지수가 2인 경우는 체배하지 않은 엔코더 펄스 한주기를 사용하는 것에 해당하고, 3인 경우는 체배하지 않은 엔코더 펄스 두주기를 사용하는 것에 해당하고, 4인 경우는 체배하지 않은 엔코더 펄스 네주기를 사용하는 것에 해당함. 따라서 11인 경우는 512주기를 사용하는 것에 해당함. 세번째와 네번째 파라미터값을 작은 값으로 지정할수록 속도계산값의 분해능은 나빠지나 시간지연은 감소함. 세번째와 네번째 파라미터값이 최대속도에서 1ms당 full wave pulse 수가 1~15개 범위이면 2, 16~31개 범위이면 3, 32~63개 범위이면 4, 64~127개 범위이면 5, 128~255개 범위이면 6이 되는 것이 기본값임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. F28334/5 에서만 가능한 명령임.

명령어	구분	설명
;AXA55A, dddd1, dd dd2, dddd3, dddd4, dddd5, dddd6, dddd 7, dddd8, dddd9, dd dd10, dddd11, dddd 12, dddd13, dddd14 ; (+#^)	operation	<p>모터1의 상위 속도에서 제어 게인을 높이거나 낮추는 파라미터를 설정함.</p> <p>1st word : 모터1의 상위 속도값 [~ 60000] (- RPM)</p> <p>2nd/3rd/4th word : 상위 속도에서의 P/I/D 위치게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>5th/6th/7th word : 01/02/04/05/06/0E/0F번 제어모드에서 상위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>8th/9th word : DC 또는 STEP 모터에서 상위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>10th/11th/12th word : 03/0A번 제어모드에서 상위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>13th/14th word : BLDC 모터에서 상위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p>
	return value	AXdddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14;
	comment	<p>2nd ~ 14th word의 값은 1000일때 게인을 조정하지 않는 것이며, 2000 이면 2배를 크게 하며, 500 이면 1/2배로 줄이는 것임. 2012년 12월 20일 S/W 버전부터 문제 없이 적용됨.</p> <p>1st word의 값을 65535로 설정하면 상위속도구간에서의 게인 조정이 실시되지 않음.</p> <p>설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.</p>
;AxA55A, dddd1, dd dd2, dddd3, dddd4, dddd5, dddd6, dddd 7, dddd8, dddd9, dd dd10, dddd11, dddd 12, dddd13, dddd14 ; (+#^)	operation	<p>모터1의 하위 속도에서 제어 게인을 높이거나 낮추는 파라미터를 설정함.</p> <p>1st word : 모터1의 하위 속도값 [~ 60000] (- ms)</p> <p>2nd/3rd/4th word : 하위 속도에서의 P/I/D 위치게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>5th/6th/7th word : 01/02/04/05/06/0E/0F번 제어모드에서 하위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>8th/9th word : DC 또는 STEP 모터에서 하위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>10th/11th/12th word : 03/0A번 제어모드에서 하위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>13th/14th word : BLDC 모터에서 하위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p>
	return value	Axdddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14;
	comment	<p>2nd ~ 14th word의 값은 1000일때 게인을 조정하지 않는 것이며, 2000 이면 2배를 크게 하며, 500 이면 1/2배로 줄이는 것임. 2012년 12월 20일 S/W 버전부터 문제 없이 적용됨.</p> <p>1st word의 값을 65535로 설정하면 하위속도구간에서의 게인 조정이 실시되지 않음.</p> <p>설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;AYA55A, dddd1, dd dd2, dddd3, dddd4, dddd5, dddd6, dddd 7, dddd8, dddd9, dd dd10, dddd11, dddd 12, dddd13, dddd14 ; (+#^)	operation	<p>모터2의 상위 속도에서 제어 게인을 높이거나 낮추는 파라미터를 설정함.</p> <p>1st word : 모터2의 상위 속도값 [~ 60000] (- ms)</p> <p>2nd/3rd/4th word : 상위 속도에서의 P/I/D 위치게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>5th/6th/7th word : 01/02/04/05/06/0E/0F번 제어모드에서 상위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>8th/9th word : DC 또는 STEP 모터에서 상위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>10th/11th/12th word : 03/0A번 제어모드에서 상위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>13th/14th word : BLDC 모터에서 상위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p>
	return value	AYdddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14;
	comment	<p>2nd ~ 14th word의 값은 1000일때 게인을 조정하지 않는 것이며, 2000 이면 2배를 크게 하며, 500 이면 1/2배로 줄이는 것임. 2012년 12월 20일 S/W 버전부터 문제 없이 적용됨.</p> <p>1st word의 값을 65535로 설정하면 상위속도구간에서의 게인 조정이 실시되지 않음.</p> <p>설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.</p>
;AyA55A, dddd1, dd dd2, dddd3, dddd4, dddd5, dddd6, dddd 7, dddd8, dddd9, dd dd10, dddd11, dddd 12, dddd13, dddd14 ; (+#^)	operation	<p>모터2의 하위 속도에서 제어 게인을 높이거나 낮추는 파라미터를 설정함.</p> <p>1st word : 모터2의 하위 속도값 [~ 60000] (- RPM)</p> <p>2nd/3rd/4th word : 하위 속도에서의 P/I/D 위치게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>5th/6th/7th word : 01/02/04/05/06/0E/0F번 제어모드에서 하위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>8th/9th word : DC 또는 STEP 모터에서 하위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>10th/11th/12th word : 03/0A번 제어모드에서 하위 속도에서의 P/I/D 속도게인 조정값 [50 ~ 20000] (1000 bit)</p> <p>13th/14th word : BLDC 모터에서 하위 속도에서의 P/I 전류게인 조정값 [50 ~ 20000] (1000 bit)</p>
	return value	Aydddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9, dddd10, dddd11, dddd12, dddd13, dddd14;
	comment	<p>2nd ~ 14th word의 값은 1000일때 게인을 조정하지 않는 것이며, 2000 이면 2배를 크게 하며, 500 이면 1/2배로 줄이는 것임. 2012년 12월 20일 S/W 버전부터 문제 없이 적용됨.</p> <p>1st word의 값을 65535로 설정하면 하위속도구간에서의 게인 조정이 실시되지 않음.</p> <p>설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;AZA55A, dddd1, dddd2; (+#^) ;AZ?; (+#^)	operation	SQ/SR/Sq/Sr/AQ/AR/Aq/Ar/WQ/WR 등의 명령어로 설정하는 게인이 적용되는 중간 속도값을 설정함. 1st word : 모터1의 중간 속도값 [~ 60000] (- RPM) 2nd word : 모터2의 중간 속도값 [~ 60000] (- RPM)
	return value	AZdddd1, dddd2, dddd3, dddd4, dddd5, dddd6; 1st word : 모터1의 중간 속도값 [~ 60000] (- RPM) 2nd word : 모터2의 중간 속도값 [~ 60000] (- RPM) 3rd word : 모터1의 하위 속도값 [~ 60000] (- RPM) 4th word : 모터1의 상위 속도값 [~ 60000] (- RPM) 5rd word : 모터2의 하위 속도값 [~ 60000] (- RPM) 6th word : 모터2의 상위 속도값 [~ 60000] (- RPM)
	comment	1st word의 값을 65535로 설정하면 모터1에 대하여 게인 조정이 실시되지 않음. 2nd word의 값을 65535로 설정하면 모터2에 대하여 게인 조정이 실시되지 않음. 중간속도값은 상위속도보다 작아야 하고, 하위속도보다 커야함. 2012년 12월 30일 S/W 버전부터 문제 없이 적용됨. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;AUeeee1,eeee2,eeee3,eeee4,eeee5,eeee6; (+#^) ;AU?; (+#^)	operation	Observer에서 사용하는 모터1의 파라미터를 설정함. 1st double : 위치값을 갱신하는 식에서 속도값에 곱하는 상수 2nd double : 속도값을 갱신하는 식에서 속도값에 곱하는 상수 3rd double : 위치값을 갱신하는 식에서 전류값에 곱하는 상수 4th double : 속도값을 갱신하는 식에서 전류값에 곱하는 상수 5rd double : 위치값을 갱신하는 식에서 위치 추정오차값에 곱하는 상수 6th double : 속도값을 갱신하는 식에서 위치 추정오차값에 곱하는 상수
	return value	AUgggg1,gggg2,gggg3,gggg4,gggg5,gggg6;
	comment	eeee 값은 실수로서 예를 들면 +10.2, -0.034, 100.0, -123.E+3, 0.2E-4 등의 형식을 사용할수 있으며, 지수는 -9 ~ +9까지만 허용됨. 파라미터는 별도의 튜닝용 S/W를 사용하여 설정하여야하며, AU/Au/AV/Av 명령을 사용하여 정상적인 값이 저장되었을 경우에 한하여, SX 명령어 두번째파라미터의 bit22/23을 사용하여 advanced control algorithm을 사용할수 있음. (정상적인 값이 저장되지 않았을때 SX 명령어 두번째파라미터의 bit22/23을 0으로 설정하면 안됨.) AU 명령으로 observer 파라미터를 수정하는 경우에, 모터1의 advanced controller 적용이 차단되며, 모터1의 advanced controller를 적용하려면 Aa 명령어로 모터1의 advanced controller 적용을 가능하게 설정하여야함. 2013년 4월 15일 S/W 버전부터 적용됨. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Au <code>eeee1,eeee2,eeee3,eeee4,eeee5,eeee6</code> ; (<code>+#^</code>) ;Au?; (<code>+#^</code>)	operation	Advanced controller에서 사용하는 모터1의 파라미터를 설정함. 1st double : 위치제어에서의 비례게인 2nd double : 위치제어에서의 적분게인 3rd double : 위치제어에서의 미분게인 4th double : 속도제어에서의 비례게인 5rd double : 속도제어에서의 적분게인 6th double : 속도제어에서 속도명령값의 적분값을 scaling 하는데 사용하는 상수값
	return value	Au <code>gggg1,gggg2,gggg3,gggg4,gggg5,gggg6</code> ;
	comment	<code>eeee</code> 값은 실수로서 예를 들면 +10.2, -0.034, 100.0, -123.E+3, 0.2E-4 등의 형식을 사용할수 있으며, 지수는 -9 ~ +9까지만 허용됨. 파라미터는 별도의 튜닝용 S/W를 사용하여 설정하여야하며, AU/Au/AV/Av 명령을 사용하여 정상적인 값이 저장되었을 경우에 한하여, SX 명령어 두번째파라미터의 bit22/23을 사용하여 advanced control algorithm을 사용할수 있음. (정상적인 값이 저장되지 않았을때 SX 명령어 두번째파라미터의 bit22/23을 0으로 설정하면 안됨.) Au 명령으로 advanced controller의 제어파라미터를 수정하는 경우에, 모터1의 advanced controller 적용이 차단되며, 모터1의 advanced controller를 적용하려면 Aa 명령어로 모터1의 advanced controller 적용을 가능하게 설정하여야함. 2013년 4월 15일 S/W 버전부터 적용됨. AC induction motor, STEP motor는 해당없음. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;AV <code>eeee1,eeee2,eeee3,eeee4,eeee5,eeee6</code> ; (+#^) ;AV?; (+#^)	operation	Observer에서 사용하는 모터2의 파라미터를 설정함. 1st double : 위치값을 갱신하는 식에서 속도값에 곱하는 상수 2nd double : 속도값을 갱신하는 식에서 속도값에 곱하는 상수 3rd double : 위치값을 갱신하는 식에서 전류값에 곱하는 상수 4th double : 속도값을 갱신하는 식에서 전류값에 곱하는 상수 5rd double : 위치값을 갱신하는 식에서 위치 추정오차값에 곱하는 상수 6th double : 속도값을 갱신하는 식에서 위치 추정오차값에 곱하는 상수
	return value	AV <code>gggg1,gggg2,gggg3,gggg4,gggg5,gggg6</code> ;
	comment	eeee 값은 실수로서 예를 들면 +10.2, -0.034, 100.0, -123.E+3, 0.2E-4 등의 형식을 사용할수 있으며, 지수는 -9 ~ +9까지만 허용됨. 파라미터는 별도의 튜닝용 S/W를 사용하여 설정하여야하며, AU/Au/AV/Av 명령을 사용하여 정상적인 값이 저장되었을 경우에 한하여, SX 명령어 두번째파라미터의 bit22/23을 사용하여 advanced control algorithm을 사용할수 있음. (정상적인 값이 저장되지 않았을때 SX 명령어 두번째파라미터의 bit22/23을 0으로 설정하면 안됨.) AV 명령으로 observer 파라미터를 수정하는 경우에, 모터2의 advanced controller 적용이 차단되며, 모터2의 advanced controller를 적용하려면 Aa 명령어로 모터2의 advanced controller 적용을 가능하게 설정하여야함. 2013년 4월 15일 S/W 버전부터 적용됨. AC induction motor, STEP motor는 해당없음. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Aveeee1,eeee2,eeee3,eeee4,eeee5,eeee6; (+#^) ;Av?; (+#^)	operation	Advanced controller에서 사용하는 모터2의 파라미터를 설정함. 1st double : 위치제어에서의 비례게인 2nd double : 위치제어에서의 적분게인 3rd double : 위치제어에서의 미분게인 4th double : 속도제어에서의 비례게인 5rd double : 속도제어에서의 적분게인 6th double : 속도제어에서 속도명령값의 적분값을 scaling 하는데 사용하는 상수값
	return value	Avgggg1,gggg2,gggg3,gggg4,gggg5,gggg6;
	comment	eeee 값은 실수로서 예를 들면 +10.2, -0.034, 100.0, -123.E+3, 0.2E-4 등의 형식을 사용할수 있으며, 지수는 -9 ~ +9까지만 허용됨. 파라미터는 별도의 튜닝용 S/W를 사용하여 설정하여야하며, AU/Au/AV/Av 명령을 사용하여 정상적인 값이 저장되었을 경우에 한하여, SX 명령어 두번째파라미터의 bit22/23을 사용하여 advanced control algorithm을 사용할수 있음. (정상적인 값이 저장되지 않았을때 SX 명령어 두번째파라미터의 bit22/23을 0으로 설정하면 안됨.) Av 명령으로 advanced controller의 제어파라미터를 수정하는 경우에, 모터2의 advanced controller 적용이 차단되며, 모터2의 advanced controller를 적용하려면 Aa 명령어로 모터2의 advanced controller 적용을 가능하게 설정하여야함. 2013년 4월 15일 S/W 버전부터 적용됨. AC induction motor, STEP motor는 해당없음. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;AaHLHL,HLHL; (+#^) ;Aa?; (+#^)	operation	Advanced controller 를 activate 시키는 파라미터를 설정함. 1st word : A55A 이면 모터1의 Advanced controller 를 activate 시키며, 그외는 deactivate 2nd word : A55A 이면 모터2의 Advanced controller 를 activate 시키며, 그외는 deactivate
	return value	AaHLHL,HLHL;
	comment	AU/Au/AV/Av 명령을 사용하는 경우 자동으로 deactivate 되며, Aa 명령으로 activate 하고자 할때는 반드시 observer 파라미터와 advanced controller 파라미터값이 정상적인 값으로 저장되었어야 하며, Advanced controller의 실질적인 작동은 SX 명령어 두번째파라미터의 bit22와 bit23이 0일때 각각 모터1과 모터2에 대하여 개시됨. 기본값은 deactivate 임. 2013년 4월 15일 S/W 버전부터 적용됨. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.
;AzA55A; (+#^)	operation	AX/Ax/AY/Ay/AZ 명령에 의한 설정값과 advanced control algorithm의 파라미터를 EEPROM에 저장함. (단 저가형 1채널 제어기에서는 EsA55A; 명령으로 저장됨)
	return value	AzA55A;
	comment	2013년 4월 15일 S/W 버전부터 advanced control algorithm의 파라미터의 저장기능이 구현되었음.
;AtA55A,dddd1,dd dd2; (+#^) ;At?; (+#^)	operation	모터1과 모터2의 제어알고리즘 적용시점을 설정함. 1st word : 모터1의 제어알고리즘 적용시점 [0 ~ 90] (0 %) 2nd word : 모터2의 제어알고리즘 적용시점 [0 ~ 90] (0 %)
	return value	Atdddd1,dddd2;
	comment	1ms의 main loop 처리 시간에서 모터1과 모터2의 제어알고리즘 적용시점을 1ms에 대하여 몇 %의 시점으로 할지를 설정함. 설정된 값은 SX 명령어 두번째파라미터의 bit24가 0일때 각각 모터1/2에 대하여 실제로 적용됨. SX 명령어로 적용하기 전에 위치제어모드에서 Qx20; 모니터에서 첫번째 값을 이용하여 첫번째 모터의 제어계산시점(아래두자리값)과 두번째모터의 제어계산시점(상위두자리)을 알아낸후에 여유있는 값을 At 명령어로 설정하고 이를 사용하여야함. 2013년 5월 8일 S/W 버전부터 적용됨. 2013년 5월 23일 이전 S/W 버전에서는 리턴값오류가 있음. 적절한 값이 설정되면 모터1과 모터2의 제어알고리즘이 일정한 시간간격을 갖게 되어, 제어출력의 fluctuation이 현저히 감소하고 구동이 정속해짐. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;A0ddd1,ddd2,d ddd3,ddd4,ddd 5,ddd6,ddd7; (+#^)	operation	<p>모터1에 사용되는 resolver (cos-sin analog type) 또는 삼상 analog Hall 센서의 offset/gain/phase를 보정하는 파라미터를 설정함.</p> <p>1st word : COS 신호 또는 삼상 analog Hall 센서 A상 신호의 offset. [0 ~ 16383] (8192 bit) 2nd word : SIN 신호 또는 삼상 analog Hall 센서 B상 신호의 offset. [0 ~ 16383] (8192 bit) 3rd word : 삼상 analog Hall 센서 C상 신호의 offset. [0 ~ 16384] (8192 bit) 4th word : SIN 신호 또는 삼상 analog Hall 센서 B상 신호의 gain 보정값. [4096 ~ 32768] (16384 bit) 5th word : 삼상 analog Hall 센서 C상 신호의 gain 보정값. [4096 ~ 32768] (16384 bit) 6th word : SIN 신호 또는 삼상 analog Hall 센서 B상 신호의 phase 보정값. [-4096 ~ 4096] (0 bit) 7th word : 삼상 analog Hall 센서 C상 신호의 phase 보정값. [-4096 ~ 4096] (0 bit)</p>
	return value	A0ddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;
	comment	<p>2013년 5월 17일 S/W 버전부터 적용됨. 단 2013년 5월 17일 현재 6th ~ 7th word의 적용은 실제로 되고 있지 않으며, 삼상 analog Hall 센서에 대해서는 1st ~ 7th word의 파라미터는 저장은 되나 실제로 적용되고 있지 않음. 단 2014년 2월 17일부터는 TMS320F28334 버전의 경우 삼상 analog Hall 센서에 대해서 ANGLE_RESOLVING_METHOD=1 일때 1st ~ 7th word의 파라미터가 적용됨. 2014년 4월 7일부터는 TMS320F2811 버전에서도 경우 삼상 analog Hall 센서에 대해서 ANGLE_RESOLVING_METHOD=1 일때 1st ~ 7th word의 파라미터가 적용됨.</p> <p>설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.</p>
;Aoddd1,ddd2,d ddd3,ddd4,ddd 5,ddd6,ddd7; (+#^)	operation	<p>모터2에 사용되는 resolver (cos-sin analog type) 또는 삼상 analog Hall 센서의 offset/gain/phase를 보정하는 파라미터를 설정함.</p> <p>1st ~ 7th word의 파라미터는 모터2에 대하여 적용되는 것 빼고는 A0 명령과 기능이 같음.</p>
	return value	Aoddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;
	comment	<p>2013년 5월 17일 S/W 버전부터 적용됨. 단 2013년 5월 17일 6th ~ 7th word의 적용은 실제로 되고 있지 않으며, 삼상 analog Hall 센서에 대해서는 1st ~ 7th word의 파라미터는 저장은 되나 실제로 적용되고 있지 않음. 단 2014년 2월 17일부터는 TMS320F28334 버전의 경우 삼상 analog Hall 센서에 대해서 ANGLE_RESOLVING_METHOD=1 일때 1st ~ 7th word의 파라미터가 적용됨. 2014년 4월 7일부터는 TMS320F2811 버전에서도 경우 삼상 analog Hall 센서에 대해서 ANGLE_RESOLVING_METHOD=1 일때 1st ~ 7th word의 파라미터가 적용됨.</p> <p>설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.</p>
;Ao?; (+#^)		

명령어	구분	설명
;Adddd1,dddd2; (+#^)	operation	모터1과 모터2의 마찰력을 보상하는 전류크기를 설정함. 1st word : 모터1의 마찰력을 보상하는 전류크기 [0 ~ 1000] (0 bit) 2nd word : 모터2의 마찰력을 보상하는 전류크기 [0 ~ 1000] (0 bit)
	return value	Adddd1,dddd2;
;Ad?; (+#^)	comment	1st/2nd word로 설정하는 전류크기가 0.01A 단위로 얼마인지를 알려면 MS?; 명령을 사용하면 됨. 2013년 5월 17일 S/W 버전부터 observer + advanced_control에서만 적용됨. 설정된 값은 AzA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Stddd1,ddd2,d ddd3; (+#^%) ;St?; (+#^%)	operation	PP/Pp/PQ/Pq/p 명령어의 update 주기와 PP/Pp/PQ/Pq/p 명령어 미수신의 연속허용횟수를 설정함. 1st word : PP/Pp/p 명령어의 update 주기 [2-1000, 2002-3000] (10 ms) 2nd word : PQ/Pq 명령어의 update 주기 [2-1000, 2002-3000] (30 ms) 3rd word : PP/Pp/PQ/Pq/p 명령어 미수신의 연속허용횟수 [2-10] (3 회)
	return value	Stddd1,ddd2,ddd3;
	comment	기본값은 10ms, 30ms, 3회 이며 기본단위는 1ms 임. 2002-3000 범위의 값을 설정하고 EEPROM에 저장한 다음 재부팅하면 메인루프의 실행시간이 1.25ms가 되며, 이에 따라서 update 주기는 2.5ms부터 1250ms 까지 1.25ms 단위로 설정됨. 2002-3000 범위의 값을 설정하는 경우, 1st word 또는 2nd word 둘중에 한개만 2002-3000 범위일지라도 EEPROM에 저장하고 재부팅한다면 1.25ms가 기본단위가 됨. 1st word 와 2nd word 둘 모두가 2-1000 범위일 때, EEPROM에 저장하고 재부팅한다면 1ms가 기본단위가 됨. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
	operation	Pp/Pq/p 명령어에 대한 return값의 주소를 설정함. 1st word : Pp/Pq/p 명령어에 대하여 return 받을 첫번째 변수의 주소값. [범위는 comment 참조] 2nd word : Pp/Pq/p 명령어에 대하여 return 받을 두번째 변수의 주소값. [범위는 comment 참조] 3rd word : p 명령어에 대하여 return 받을 세번째 변수의 주소값. [범위는 comment 참조] 4th word : p 명령어에 대하여 return 받을 네번째 변수의 주소값. [범위는 comment 참조]
	return value	SrHLHL,HLHL; or SrHLHL,HLHL,HLHL,HLHL;
;SrHLHL,HLHL; (+##^%) ;SrHLHL,HLHL,HLHL,HLHL; (+##^%) ;Sr?; (+##^%)	comment	SrHLHL,HLHL; 명령을 보내면 p 명령에 대한 return 값은 1st/2nd word로 지정된 2개 주소내의 값이 되며, SrHLHL,HLHL,HLHL,HLHL; 명령을 보내면 p 명령에 대한 return 값은 1st/2nd/3rd/4th word로 지정된 4개 주소내의 값이 됨. 이때 1st/2nd word는 QZ 명령어에 의해 설정되는 1st/2nd 주소값을 변경시킴. Pp/Pq 명령에 대한 return 값은 항상 1st/2nd word로 지정된 2개 주소내의 값이 됨. TMS320F810 [0400 - 05FF] : M1 RAM block (.stack) [0600 - 07FF] : M1 RAM block (.bss) [0880 - 783F] : peripheral memory [8000 - 9FFF] : L0/L1 RAM block (.ebss) [E000 - FFFF] : H0 RAM block (.esystem) 70A8(7108) - 70B7(7117) : A/D channel 0 - 15 (AD0 CURRENT1_SEN+, AD1 CURRENT1_SEN-, AD2 CURRENT1_SET, AD3 LS1A, AD4 LS1B, AD5 LS1C, AD6 BLDC_CURR_SEN1, AD7 DC_BUS_VOLT, AD8 CURRENT2_SEN+, AD9 CURRENT2_SEN-, AD10 CURRENT2_SET, AD11 LS2A, AD12 LS2B, AD13 LS2C, AD14 BLDC_CURR_SEN2, AD15 FET_TEMP) 기본값은 70AB(710B) : LS1A, 70B3(7113) : LS2A 임. AD 변환값을 읽는 경우는 MSB에서 부터 10bit/12bit가 유효값임.

명령어	구분	설명
;SZddd1,ddd2; (+#^%) ;SZ?; (+#^%)	operation	Hall sensor pulse의 지정된 edge를 기준으로 하여 엔코더 Z펄스의 위치를 엔코더펄스 갯수로 읽어 냄. (BLDC 구동 chip IRMCK201 전용) 1st word : BLDC 모터1 에서 출력되는 Hall sensor pulse의 지정된 edge 값 [0 - 6] 2nd word : BLDC 모터2 에서 출력되는 Hall sensor pulse의 지정된 edge 값 [0 - 6] 0 이면 Z펄스의 위치를 엔코더펄스 갯수로 읽어 내는 작동을 하지 않음.
	return value	SZddd1,ddd2; 1st word : BLDC 모터1 - Hall sensor pulse의 지정된 edge 값 (SZddd1,ddd2; 명령시) : BLDC 모터1 - Hall sensor pulse의 지정된 edge를 기준으로 한 엔코더 Z펄스의 위치값 (SZ?; 명령시, 단위는 4체배된 엔코더펄스수) 2nd word : BLDC 모터2 - Hall sensor pulse의 지정된 edge 값 (SZddd1,ddd2; 명령시) : BLDC 모터2 - Hall sensor pulse의 지정된 edge를 기준으로 한 엔코더 Z펄스의 위치값 (SZ?; 명령시, 단위는 4체배된 엔코더펄스수)
	comment	BLDC 모터제어기에 전원을 투입한후, SM 과 PE 명령어로 모터의 작동개시를 하지 않은 상태에서 명령을 수행토록 해야하며, 한번에 한개의 모터에 대해서만 실시하여야 함. SZddd1,ddd2; 명령을 입력하면 작동 LED가 빠르게 깜빡거리기 시작함. 이어서 모터축을 들어다 볼 때 반시계방향으로 모터축을 천천히 돌리면서 작동 LED의 초당 10회 깜빡임이 정상주기 (500ms, 초당 2회)로 돌아올 때까지 계속함. 이후에 SZ?; 명령어로 엔코더 Z펄스의 위치값을 읽음. SZ?; 에 대한 return 값은 SZddd1,ddd2; 명령을 수행이 끝나서 얻어진 엔코더 Z펄스의 위치값임. 이러한 동작을 몇번 반복하여 읽은 엔코더 Z펄스의 위치값들이 거의 같은 상태가 되도록 만들고 최종값을 적절한 연산을 거쳐 변환한 다음, Qt06,00,HLHL; 또는 Qt06,01,HLHL; 명령어를 사용하여 모터제어기에 전송하고, ESOA00,0100; 명령어로 EEPROM에 저장하면 됨. BLDC인 경우에 한하여만 사용이 가능한 명령임.

명령어	구분	설명
;SB#####; (+#^%) ;SB?; (+#^%)	operation	RS232C 직렬통신속도값을 설정함. ##### : 속도설정값 [9600,19200,38400,57600,115200, 625000,1250000,2500000] (115200 bps)
	return value	SB#####; (SB?; 인 경우에만 echo 있음)
	comment	DIP 스위치 1번이 ON 되어 있는 상태에서 전원이 투입되었을 때(RS232C 모드)는 baud rate 설정 레지스터 값을 바꾸어서 실제 통신속도를 바꾸나, RS485모드인 상태에서는 EEPROM 저장용 parameter 만 설정함. 설정값이 적당한 baud rate가 아니면 설정값이 무시됨. 전원투입시에 DIP 스위치 4번을 ON 하고 전원을 투입하면 default 값인 115200 으로 설정되며, 그 이외의 경우는 EEPROM에 저장된 값으로 설정되나, 적당한 baud rate가 아니면 default 값인 115200으로 설정됨. ####x 는 unsigned long integer 임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Sb#####; (+^%) ;Sb?; (+^%)	operation	RS485 직렬통신속도값을 설정함. ##### : 속도설정값 [9600,19200,38400,57600,115200, 250000, 500000, 625000,1250000,2500000] (1250000 bps)
	return value	Sb#####; (Sb?; 인 경우에만 echo 있음)
	comment	DIP 스위치 1번이 OFF 되어 있는 상태에서 전원이 투입되었을 때(RS485 모드)는 baud rate 설정 레지스터 값을 바꾸어서 실제 통신속도를 바꾸나, RS232C모드인 상태에서는 EEPROM 저장용 parameter 만 설정함. 설정값이 적당한 baud rate가 아니면 설정값이 무시됨. 전원투입시에 DIP 스위치 4번을 ON 하고 전원을 투입하면 default 값인 1250000 으로 설정되며, 그 이외의 경우는 EEPROM에 저장된 값으로 설정되나, 적당한 baud rate가 아니면 default 값인 1250000으로 설정됨. ####x 는 unsigned long integer 임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;SbIIII1,IIII2; (#) ;Sb?; (#)	operation	RS485 와 RS232C extra 직렬통신속도값을 설정함. IIII1 : 속도설정값 [9600,19200,38400,57600,115200, 625000,1250000,2500000] (1250000 bps) IIII2 : 속도설정값 [9600,19200,38400,57600,115200, 625000,1250000,2500000] (115200 bps)
	return value	SbIIII1,IIII2; (Sb?: 인 경우에만 echo 있음)
	comment	RS485의 경우 DIP 스위치 1번이 OFF 되어 있는 상태에서 전원이 투입되었을 때(RS485 모드)는 baud rate 설정 레지스터값을 바꾸어서 실제 통신속도를 바꾸나, RS232C모드인 상태에서는 EEPROM 저장용 parameter 만 설정함. 설정값이 정당한 baud rate가 아니면 설정값이 무시됨. RS485의 경우 전원투입시에 DIP 스위치 4번을 ON 하고 전원을 투입하면 default 값인 1250000 으로 설정되며, 그 이외의 경우는 EEPROM에 저장된 값으로 설정되나, 정당한 baud rate가 아니면 default 값인 1250000으로 설정됨. RS232C extra는 설정값이 정당한 baud rate가 아니면 115200 으로 설정됨. IIIIx 는 unsigned long integer 임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;SFdddd1,dddd2,dddd3; (+#^%) ;SF?dddd1; (+#^%)	operation	butterworth filter의 시정수와 변화율 상한값을 설정함. 1st word : filter 번호. [0 - 7/8/9] (번) 2nd word : filter의 시정수. [1 - 32767] (bit) 3rd word : filter의 출력변화율 상한값. [1 - 32767] (bit)
	return value	SFdddd1,dddd2,dddd3;
	comment	butterworth filter는 2차 필터로서 0-7/15 까지 총 8/16개가 있으며 1ms 마다 filter가 동작함. 2nd word를 2로 나눈 값이 ms 단위의 실제 시정수에 해당함. 3rd word는 출력이 1ms 당 변할 수 있는 최대값을 지정함. 6번 필터는 모터구동용 전압의 AD 변환값에 사용되며 시정수/변화율상한치의 기본값은 16/64 임. 7번 필터는 MOTOR1의 전류 AD 변환값에 사용되며 시정수/변화율상한치의 기본값은 256/4 임. 2010년 12월 7일 S/W 버전부터는 filter의 시정수로 1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192 값만 유효한데, 계산시간을 줄이기 위하여 shift operator를 사용함으로써 새로이 생겨난 제한임. 2010년 4월 1일 S/W 버전부터는 고전압 전류앰프인 경우에 2번 필터를 외부전류명령용 analog입력의 필터링에 사용함. 2014년 4월 5일 S/W 버전부터는 filter의 시정수로 0을 사용할 수 있으며, 이때는 필터링을 하지 않는 것과 같음. 또 Butterworth 필터의 스텝입력에 대한 응답에서 시정수가 작을 때 발생하는 약간의 overshoot가 생기지 않도록 하였음. 0-7번, 9번 필터의 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음. 8번 필터의 설정값은 F2811/F28334에서만 유효하며 내부에 있는 8-15번까지의 필터에 대하여 시정수 및 출력변화율 상한값은 같은 모두 값을 사용함. 8-15번 필터의 시정수 및 출력변화율 상한값의 기본값은 8, 64이며 EEPROM에 저장이 안되는 값임. 2012년 3월 9일 S/W 버전부터 필터번호가 9인 경우는 F2811/F28334 H/W 버전에서만 유효하며 F/T센서와 외부 analog 명령의 필터일에 사용하는 추가필터의 시정수를 설정하기 위한 것임.

명령어	구분	설명
(2012년 10월 03일 이전 S/W 버전) ;Si dddd1, dddd2; (+#^%) ;Si?; (+#^%)	operation	<p>RJM_VER1과 RJM_VER2의 경우에는 과전류에러 fault 진단을 위한 1.024초 동안의 과전류검출 횟수를 설정하며, 그외의 경우는 SI 명령어로 설정한 전류치보다 실제전류가 연속적으로 초과하기를 허용하는 최대시간을 설정함.</p> <p>RJM_VER1 또는 RJM_VER2의 경우 :</p> <p>1st word : 이 횟수 이상으로 모터1의 과전류를 검출하였을때 MOTOR1_MAX_CURRENT_fault_flg를 세트함. [1 - 1024] (50 회)</p> <p>2nd word : 이 횟수 이상으로 모터2의 과전류를 검출하였을때 MOTOR2_MAX_CURRENT_fault_flg를 세트함. [1 - 1024] (50 회)</p> <p>그외의 경우 :</p> <p>1st word : 이 횟수 이상으로 연속해서 모터1에서 SI 명령어로 설정한 전류치를 초과하는 경우 MOTOR1_MAX_CURRENT_fault_flg를 세트함. [1 - 10000] (500 ms, 380V 제어기의 경우는 50)</p> <p>2nd word : 이 횟수 이상으로 연속해서 모터2에서 SI 명령어로 설정한 전류치를 초과하는 경우 MOTOR2_MAX_CURRENT_fault_flg를 세트함. [1 - 10000] (500 ms, 380V 제어기의 경우는 50)</p>
	return value	Si dddd1, dddd2;
	comment	<p>이 설정값은 DSP의 SARAM에 저장되며 만약 EEPROM에 저장하려면 ESxxxx;명령어를 사용하여야함.</p> <p>RJM_VER1 또는 RJM_VER2의 경우에 기본값은 50(1024 번 중에 50번에 해당), 50(1024 번 중에 50번에 해당)임. 그외의 경우는 500 또는 50ms로서 500 또는 50ms동안 SI 명령어로 설정한 전류치를 초과하는 경우 MOTORx_MAX_CURRENT_fault_flg를 세트하고 출력을 차단하며, 한번이라도 SI 명령어로 설정한 전류치를 초과하지 않으면 타이머를 0으로 clear함. 총전류에 대하여서는 SI 명령어로 설정한 전류치의 3/4 크기를 상한선으로 사용함. 380V 제어기가 아닌 경우에는 Si 명령에 의한 시간값을 SI 명령어로 설정하는 TIME_LIMIT_OF_CURRENT1_OVER 값의 3/4 보다 작지 않게 제한함.</p> <p>2012년 5월 28일 이전 S/W 버전에서는 RJM_VER1 또는 RJM_VER2의 DC motor에만 해당되었는데, 그 이후부터는 모든 제어기에 대하여 적용됨.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
(2012년 10월 03일 S/W 버전부터) ;Sidd1,ddd2; (+#^%) 또는 ;Sidd1,ddd2,ddd3,ddd4; (+#^%)	operation	RJM_VER1과 RJM_VER2의 경우에는 과전류에러 fault 진단을 위한 1.024초 동안의 과전류검출 횟수를 설정하며, 그외의 경우는 SI 명령어로 설정한 전류치보다 실제전류가 연속적으로 초과하기를 허용하는 최대시간을 설정함. RJM_VER1 또는 RJM_VER2의 경우 : 1st word : 이 횟수 이상으로 모터1의 과전류를 검출하였을때 MOTOR1_MAX_CURRENT_fault_flg를 세트함. [1 - 1024] (50 회) 2nd word : 이 횟수 이상으로 모터2의 과전류를 검출하였을때 MOTOR2_MAX_CURRENT_fault_flg를 세트함. [1 - 1024] (50 회) 그외의 경우 : 1st word : 이 횟수 이상으로 연속해서 모터1에서 SI/Sw 명령어로 설정한 전류치중 작은 값의 125%를 초과하는 경우 MOTOR1_MAX_CURRENT_fault_flg를 세트함. [1 - 10000] (500 ms, 380V 제어기의 경우는 50) 2nd word : 이 횟수 이상으로 연속해서 모터2에서 SI/Sw 명령어로 설정한 전류치중 작은 값의 125%를 초과하는 경우 MOTOR2_MAX_CURRENT_fault_flg를 세트함. [1 - 10000] (500 ms, 380V 제어기의 경우는 50) 3rd word : 이 횟수 이상으로 연속해서 모터1에서 SI/Sw 명령어로 설정한 전류치중 큰 값의 125%를 초과하는 경우 MOTOR1_MAX_CURRENT_fault_flg를 세트함. [1 - 10000] (3 ms) 4th word : 이 횟수 이상으로 연속해서 모터2에서 SI/Sw 명령어로 설정한 전류치중 큰 값의 125%를 초과하는 경우 MOTOR2_MAX_CURRENT_fault_flg를 세트함. [1 - 10000] (3 ms)
	return value	Sidd1,ddd2,ddd3,ddd4;
;Si?; (+#^%) 2013년 4월 4일 S/W 버전부터 Wi 명령어로 바뀜.	comment	이 설정값은 DSP의 SARAM에 저장되며 만약 EEPROM에 저장하려면 ESxxx;명령어를 사용하여야함. RJM_VER1 또는 RJM_VER2의 경우에 기본값은 50(1024 번 중에 50번에 해당), 50(1024 번 중에 50번에 해당)임. 그외의 경우는 500 또는 50ms로서 500 또는 50ms동안 SI 명령어로 설정한 전류치를 초과하는 경우 MOTORx_MAX_CURRENT_fault_flg를 세트하고 출력을 차단하며, 한번이라도 SI 명령어로 설정한 전류치를 초과하지 않으면 타이머를 0으로 clear함. 총전류에 대하여서는 SI 명령어로 설정한 전류치의 3/4 크기를 상한선으로 사용함. 380V 제어기가 아닌 경우에는 Si 명령에 의한 시간값을 SI 명령어로 설정하는 TIME_LIMIT_OF_CURRENT1_OVER 값의 3/4 보다 작지 않게 제한함. 2012년 5월 28일 이전 S/W 버전에서는 RJM_VER1 또는 RJM_VER2의 DC motor에만 해당되었는데, 그 이후부터는 모든 제어기에 대하여 적용됨. 2013년 4월 4일 S/W 버전부터 Wi 명령어로 바뀜. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Sxddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8; (+#^%) ;Sx?; (+#^%) 2013년 4월 4일 S/W 버전부터 MX/Mx 명령으로 바뀜.	operation	제어오차가 과도하게 큰 상태를 진단하기 위한 위치오차, 속도오차, 전류오차, 허용시간을 설정함. (RJM_VER5에는 해당없음) 1st word : 모터1의 감속기 출력 기준의 허용 오차각. [1 - 1000] (10 도) 2nd word : 모터2의 감속기 출력 기준의 허용 오차각. [1 - 1000] (10 도) 3rd word : 모터1의 모터 출력축의 허용 속도오차. [1 - 10000] (1000 또는 SG 명령어로 설정되는 MOTOR1_SPEED_MAX/5 rpm) 4th word : 모터2의 모터 출력축의 허용 속도오차. [1 - 10000] (1000 또는 SG 명령어로 설정되는 MOTOR2_SPEED_MAX/5 rpm) 5th word : 모터1의 허용 전류오차 [1 - 1000] (200 또는 SI 명령어로 설정되는 current1_limit/2 bit) 6th word : 모터2의 허용 전류오차 [1 - 1000] (200 또는 SI 명령어로 설정되는 current2_limit/2 bit) 7th word : 모터1에서 이 시간 이상으로 연속으로 오류를 검출하였을 때 에러를 설정함. [1 - 50000] (10000 ms) 8th word : 모터1에서 이 시간 이상으로 연속으로 오류를 검출하였을 때 에러를 설정함. [1 - 50000] (10000 ms)
	return value	Sxddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8;
	comment	이 설정값은 DSP의 SARAM에 저장되며 만약 EEPROM에 저장하려면 ESxxxx:명령어를 사용하여야함. 기본값은 10도, 1000 또는 SG 명령어로 설정되는 MOTOR1_SPEED_MAX/5 rpm, 200 또는 SI 명령어로 설정되는 current1_limit/2 bit, 10000ms 임. 2013년 4월 4일 S/W 버전부터 MX/Mx 명령으로 바뀜. (MX 명령은 2013년 4월 4일 이전 S/W 버전의 Sx 명령과 같으며, 2013년 4월 4일 S/W 버전부터는 Mx 명령을 사용하여 0.01A 단위로 모터1/2의 허용 전류오차를 설정할수있음.) 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;SmHLHL; (+#^%) ;Sm?; (+#^%)	operation	<p>명령어에 대한 return값을 보낼 때 사용하는 host의 주소를 설정함.</p> <p>1st word 2012년 10월 18일 이전 S/W 버전인 경우 또는 SX 명령어 첫번째파라미터의 bit0 = 1 인 경우 : 주소값 [0000 - 00FF] (FF00)</p> <p>2012년 10월 18일 S/W 버전부터로서 SX 명령어 첫번째파라미터의 bit0 = 0 인 경우 : 주소값 [0000 - 00FF] (FF00) - SX 명령어 두번째파라미터의 bit15/14 = 11 일때 : 주소값 [0000 - 07FF] (0000) - SX 명령어 두번째파라미터의 bit15/14 = 10 일때 : 주소값 [0000 - FFFF] (0000) - SX 명령어 두번째파라미터의 bit15/14 = 01 일때 : 주소값 [0000 - 00FF] (FF00) - SX 명령어 두번째파라미터의 bit15/14 = 00 일때</p>
	return value	SmHLHL;
	comment	<p>☞ 2012년 10월 18일 이전 S/W 버전인 경우 또는 SX 명령어 첫번째파라미터의 bit0 = 1 인 경우와 2012년 10월 18일 S/W 버전부터로서 SX 명령어 두번째파라미터의 bit15/14 = 11 또는 00 일때는, 주소값의 상위 바이트는 반드시 하위바이트의 inverse이어야만 baord 주소의 설정이 유효함.</p> <p>☞ 전원 투입시에 기본값은 FF00 또는 0000으로서 0번지를 return 값의 기본 host 주소로 지정함.</p> <p>☞ 이 명령어에 대한 리턴값은 바뀐 주소의 host로 전송함.</p> <p>☞ 전원을 껐다 켜지 않는 한 RS485 모드일 때 설정된 호스트 주소는 CAN에서는 적용되지 않으나, CAN 모드일 때 설정된 호스트 주소는 RS485 모드일때도 바로 적용됨.</p> <p>☞ 2012년 10월 18일 S/W 버전부터로서 SX 명령어 첫번째파라미터의 bit0 = 0 일때 SX 명령어 두번째파라미터의 bit15/14 값에 따라서 CAN 통신의 주소사용방법이 달라짐.</p> <p>bit15/14 = 11 일때 11-bit 주소모드로서 주소로 0-255만 사용 = 10 일때 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용 = 01 일때 29-bit 주소모드로서 주소로 0-65535만 사용 (CAN2.0B), 2012년 10월 16일부터 적용 = 00 일때 11-bit 주소모드로서 주소로 0-255만 사용</p> <p>☞ 주소값으로서 254(0xFE) 및 255(0xFF)는 bradcasting용으로 사용하는 주소이므로 126(0x8E), 127(0x8F), 254(0xFE) 및 255(0xFF)는 host의 주소로 사용하면 안됨.</p> <p>☞ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;WQdddd1,dddd2,dddd3,dddd4,dddd5; ;(+#) ;WQ?; (+#)	operation	<p>모터1의 03/06/0A번 위치/속도제어모드에서 속도게인과 , RJM_VER7형의 3상/5상 STEP모터와 BLDC 모터1의 전류제어파라미터를 설정함.</p> <p>1st word : MOTOR1_SPEED_P_GAIN (32/20) 2nd word : MOTOR1_SPEED_I_GAIN (16/4/2) 3rd word : MOTOR1_SPEED_D_GAIN (8/2/1) 4th word : MOTOR1_CURRENT_P_GAIN (40) [0 - 1000] 5th word : MOTOR1_CURRENT_I_GAIN (180) [0 - 8000/32000]</p>
	return value	WQdddd1,dddd2,dddd3,dddd4,dddd5;
	comment	<p>⚙ 처음 3개의 파라미터는 모터1의 03/06/0A번 위치/속도/속도제어모드에서의 속도제어게인을 설정함. 03번 위치제어모드에서는 위치제어루프 내부의 1KHz 속도제어루프에서 사용하는 속도제어게인을 설정하며, 06번 속도제어모드에서는 BLDC에 한하여 T1UFINT_ISR에서 처리되는 10KHz 속도제어루프에서 속도제어게인을 설정하며, 0A번 속도제어모드에서는 1KHz 속도제어루프에서 사용하는 속도제어게인을 설정하는데 0A번 속도제어모드는 03번 위치제어루프 내부의 1KHz 속도제어루프와 같은 알고리즘이고 같은 게인을 사용함. 마지막 2개의 파라미터는 모터1로서 BLDC 모터와 RJM_VER7형의 3상 STEP모터인 경우에 07번 전류제어모드에서 전류제어게인을 설정함. 전류제어주기는 PWM 주파수와 같으며 전류제어 파라미터는 전류제어주파수(=PWM 주파수)와 무관하게 같은 값을 사용할수 있음. 07번 전류제어모드에서의 전류제어게인은 속도나 위치제어에서 내부 전류제어루프를 사용하는 경우에도 같은 값이 적용됨. 전류제어의 I-게인은 TMS320F2810/11/12의 경우는 8000까지, TMS320F28334/5의 경우는 32000까지 설정 가능함.</p> <p>⚙ MOTOR1_SPEED_I_GAIN 은 2012년 12월 19일 S/W 버전부터 0으로 설정하지 못하도록 하였음.</p> <p>⚙ 4th/5th word의 경우 DC 또는 STEP 모터제어기로서 자동화용 H/W 버전의 제어기에서는 (이 경우 전류게인은 SQ/SR/AS 명령어로 설정함) 외부에서 인가되는 속도명령 또는 전류명령 전압값을 내부의 속도명령 및 전류명령값으로 변환하는데 파라미터로 사용함.</p> <p>⚙ BLDC1_PrmBuffer [25]/BLDC1_PrmBuffer [26]/BLDC1_PrmBuffer [27]/BLDC1_PrmBuffer [13]/BLDC1_PrmBuffer [14] 에도 설정된 값이 저장되며 이를 통하여 ES 명령어에 의해 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
	comment	<ul style="list-style-type: none"> ☞ F2811/F28334 RJM_VER7에서만 가능한 명령임. ☞ 2014년 1월 27일 S/W 버전부터 (RJM_VER8==1)&&(RJM_VER8_from_VER7==1)&&(DRV8432==1)&&(CURRENT_DRIVER==1)일때는 첫번째 파라미터와 두번째 파라미터를 외부전류명령에 대한 offset과 gain으로 사용하도록 하였음. 이때 offset은 8192가 기본값이며, 게인은 16384일때 실제게인 1에 해당함.

명령어	구분	설명
;WRddd1,ddd2,ddd3,ddd4,ddd5; ;(+#) ;WR?;(+#)	operation	<p>모터2의 03/06/0A번 위치/속도제어모드에서 속도게인과 , RJM_VER7형의 3상/5상 STEP모터와 BLDC 모터2의 전류제어파라미터를 설정함.</p> <p>1st word : MOTOR2_SPEED_P_GAIN (32/20) 2nd word : MOTOR2_SPEED_I_GAIN (16/4/2) 3rd word : MOTOR2_SPEED_D_GAIN (8/2/1) 4th word : MOTOR2_CURRENT_P_GAIN (40) [0 - 1000] 5th word : MOTOR2_CURRENT_I_GAIN (180) [0 - 8000/32000]</p>
	return value	WRddd1,ddd2,ddd3,ddd4,ddd5;
	comment	<p>☞ 처음 3개의 파라미터는 모터2의 03/06/0A번 위치/속도/속도제어모드에서의 속도제어게인을 설정함. 03번 위치제어모드에서는 위치제어루프 내부의 1KHz 속도제어루프에서 사용하는 속도제어게인을 설정하며, 06번 속도제어모드에서는 BLDC에 한하여 T1UFINT_ISR에서 처리되는 10KHz 속도제어루프에서 속도제어게인을 설정하며, 0A번 속도제어모드에서는 1KHz 속도제어루프에서 사용하는 속도제어게인을 설정하는데 0A번 속도제어모드는 03번 위치제어루프 내부의 1KHz 속도제어루프와 같은 알고리즘이고 같은 게인을 사용함. 마지막 2개의 파라미터는 모터2로서 BLDC 모터와 RJM_VER7형의 3상 STEP모터인 경우에 07번 전류제어모드에서 전류제어게인을 설정함. 전류제어주기는 PWM 주파수와 같으며 전류제어 파라미터는 전류제어주파수(=PWM 주파수)와 무관하게 같은 값을 사용할수 있음. 07번 전류제어모드에서의 전류제어게인은 속도나 위치제어에서 내부 전류제어루프를 사용하는 경우에도 같은 값이 적용됨. 전류제어의 I-게인은 TMS320F2810/11/12의 경우는 8000까지, TMS320F28334/5의 경우는 32000까지 설정 가능함.</p> <p>☞ MOTOR2_SPEED_I_GAIN 은 2012년 12월 19일 S/W 버전부터 0으로 설정하지 못하도록 하였음.</p> <p>☞ 4th/5th word의 경우 DC 또는 STEP 모터제어기로서 자동화용 H/W 버전의 제어기에서는 (이 경우 전류게인은 SQ/SR/AS 명령어로 설정함) 외부에서 인가되는 속도명령 또는 전류명령 전압값을 내부의 속도명령 및 전류명령값으로 변환하는데 파라미터로 사용함.</p> <p>☞ BLDC2_PrmBuffer [25]/BLDC2_PrmBuffer [26]/BLDC2_PrmBuffer [27]/BLDC2_PrmBuffer [13]/BLDC2_PrmBuffer [14] 에도 설정된 값이 저장되며 이를 통하여 ES 명령어에 의해 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
	comment	<ul style="list-style-type: none"> ☞ F2811/F28334 RJM_VER7에서만 가능한 명령임. ☞ 2014년 1월 27일 S/W 버전부터 (RJM_VER8==1)&&(RJM_VER8_from_VER7==1)&&(DRV8432==1)&&(CURRENT_DRIVER==1)일때는 첫번째 파라미터와 두번째 파라미터를 외부전류명령에 대한 offset과 gain으로 사용하도록 하였음. 이때 offset은 8192가 기본값이며, 게인은 16384일때 실제게인 1에 해당함.

명령어	구분	설명
;WBddd1,ddd2; (+#^) ;WB?; (+#^)	operation	RJM_VER7 제어시에 100kHz interrupt routine에서 사용하는 Butterworth filter의 시정수를 설정함. 1st word : BUTTERWORTH_TIME_CONSTANT_for_MODE5 [0 - 4] (1) 2nd word : BUTTERWORTH_TIME_CONSTANT_for_MODE6 [0 - 4] (3)
	return value	WBddd1,ddd2; 1st word : BUTTERWORTH_TIME_CONSTANT_for_MODE5 2nd word : BUTTERWORTH_TIME_CONSTANT_for_MODE6
	comment	RJM_VER7에 한하여 적용됨. F2811/F28334 RJM_VER7에서만 가능한 명령임.
(Wafer control의 경우로서 2012년 2월 18일 S/W 버전부터 적용) ;WEddd1,ddd2; (+#^) ;WE?; (+#^)	operation	BLDC 모터1,2에 있어서 Hall_A상 상승에지에 대한 U상 전압 positive zero crossing 포인트의 위상지연 값을 설정하며, 모터1,2의 POLE_PAIR 수, 모터1,2의 MOTORx_POSITION_SCALE_FACTOR를 읽어서 host로 전송함. 1st word : UPhase_to_HallA_ANGLE1_OFFSET (525/480/610/600 0.1도) 2nd word : UPhase_to_HallA_ANGLE2_OFFSET (525/480/610/600 0.1도)
	return value	WEddd1,ddd2,ddd3,ddd4,ddd5,ddd6; 1st word : UPhase_to_HallA_ANGLE1_OFFSET 2nd word : UPhase_to_HallA_ANGLE2_OFFSET 3rd word : MOTOR1_NUMBER_of_POLE_PAIR 4th word : MOTOR2_NUMBER_of_POLE_PAIR 5th word : MOTOR1_POSITION_SCALE_FACTOR 6th word : MOTOR2_POSITION_SCALE_FACTOR
	comment	RJM_VER7에 한하여 적용됨. Wafer control의 경우 WE명령어 두번째 파라미터로서 0~4096가 0도에서 +90도에 해당하며, 61440~65535가 -90도에서 0도에 해당함)를 사용하여 Id를 만드는 PWM_offset값의 phase angle을 미세조정할수 있음. BLDC1_PrmBuffer[3]/BLDC1_PrmBuffer[4]/BLDC2_PrmBuffer[3]/BLDC2_PrmBuffer[4]에도 설정된 값이 저장되며 이를 통하여 ES 명령어에 의해 EEPROM에 저장될 수 있음. F2811/F28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
(Wafer control이 아닌 경우) ;WEdddd1,dddd2; (+#^) ;WE?: (+#^)	operation	BLDC 모터1,2에 있어서 Hall_A상 상승에지에 대한 U상 전압 positive zero crossing 포인트의 위상지연 값을 설정하며, 모터1,2의 POLE_PAIR 수, 모터1,2의 MOTORx_POSITION_SCALE_FACTOR를 읽어서 host로 전송함. 1st word : UPhase_to_HallA_ANGLE1_OFFSET (525/480/610/600 0.1도) 2nd word : UPhase_to_HallA_ANGLE2_OFFSET (525/480/610/600 0.1도)
	return value	WEdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; 1st word : UPhase_to_HallA_ANGLE1_OFFSET 2nd word : UPhase_to_HallA_ANGLE2_OFFSET 3rd word : MOTOR1_NUMBER_of_POLE_PAIR 4th word : MOTOR2_NUMBER_of_POLE_PAIR 5th word : MOTOR1_POSITION_SCALE_FACTOR 6th word : MOTOR2_POSITION_SCALE_FACTOR
	comment	RJM_VER7에 한하여 적용됨. Wafer control의 경우 WE명령어 두번째 파라미터로서 0~4096가 0도에서 +90도에 해당하며, 61440~65535가 -90도에서 0도에 해당함)를 사용하여 Id를 만드는 PWM_offset값의 phase angle을 미세조정할수 있음. BLDC1_PrmBuffer[3]/BLDC1_PrmBuffer[4]/BLDC2_PrmBuffer[3]/BLDC2_PrmBuffer[4]에도 설정된 값이 저장되며 이를 통하여 ES 명령어에 의해 EEPROM에 저장될 수 있음. F2811/F28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
;WGdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; (+#^)	operation	analog Hall sensor type의 RJM_VER7 모터1의 센서 scale과 offset을 설정함. 1st word : HALL_Sensor1_A_Scale (45621/47410/54142) 2nd word : HALL_Sensor1_B_Scale (45792/46362/53092) 3rd word : HALL_Sensor1_C_Scale (46377/47077/55370) 4th word : HALL_Sensor1_A_Offset (6518/7076/8038) 5th word : HALL_Sensor1_B_Offset (6558/6926/7852) 6th word : HALL_Sensor1_C_Offset (6652/7070/8317)
	return value	WGdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;
	comment	RJM_VER7에 한하여 적용됨. BLDC1_PrmBuffer[57]/BLDC1_PrmBuffer[58]/BLDC1_PrmBuffer[59]/BLDC1_PrmBuffer[60]/BLDC1_PrmBuffer[61]/BLDC1_PrmBuffer[62]에도 설정된 값이 저장되며 이를 통하여 ES 명령어에 의해 EEPROM에 저장될 수 있음. F2811/F28334 RJM_VER7에서만 가능한 명령임.
;WHdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; (+#^)	operation	analog Hall sensor type의 RJM_VER7 모터2의 센서 scale과 offset을 설정함. 1st word : HALL_Sensor2_A_Scale (45621/47410/54142) 2nd word : HALL_Sensor2_B_Scale (45792/46362/53092) 3rd word : HALL_Sensor2_C_Scale (46377/47077/55370) 4th word : HALL_Sensor2_A_Offset (6518/7076/8038) 5th word : HALL_Sensor2_B_Offset (6558/6926/7852) 6th word : HALL_Sensor2_C_Offset (6652/7070/8317)
	return value	WHdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;
	comment	RJM_VER7에 한하여 적용됨. BLDC2_PrmBuffer[57]/BLDC2_PrmBuffer[58]/BLDC2_PrmBuffer[59]/BLDC2_PrmBuffer[60]/BLDC2_PrmBuffer[61]/BLDC2_PrmBuffer[62]에도 설정된 값이 저장되며 이를 통하여 ES 명령어에 의해 EEPROM에 저장될 수 있음. F2811/F28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
;WCddd1,ddd2; (+#^) ;WC?; (+#^)	operation	analog Hall sensor type의 RJM_VER7 모터에 대하여 위상각 측정용 파라미터의 설정을 개시함. 1st word : 모터1의 파라미터 설정을 개시하지 않음(0)/개시함(1)/중지함(2) 2nd word : 모터2의 파라미터 설정을 개시하지 않음(0)/개시함(1)/중지함(2)
	return value	WCddd1,ddd2; 1st word : analog_Hall_sensor1_calibration_cnt 2nd word : analog_Hall_sensor2_calibration_cnt
	comment	RJM_VER7에 한하여 적용됨. 작동모드 0일 상태에서만 설정의 개시가 가능하며, 일단 개시되면 강제로 중지 시키지 않는 한 약 1+3/4바 퀴정도의 회전이 있어야, 파라미터 설정에 필요한 측정을 수행하고 파라미터 설정을 완료함. 파라미터 설정과정이 수행중일 때는 램프가 초당 10회 깜빡거리며, 설정이 완료되거나 취소되면 정상주기 (500ms, 초당 2회)로 돌아옴. 실행결과로 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음. F2811/F28334 RJM_VER7에서만 가능한 명령임.
;WXiiii1,iiii2; (+#^) ;WX?; (+#^)	operation	RJM_VER7 모터의 electrical angle을 강제로 설정함. 1st integer : 모터1의 electrical angle [-360 - +360] (기본값 없음, 1도 단위) 2nd integer : 모터2의 electrical angle [-360 - +360] (기본값 없음, 1도 단위)
	return value	WXiiii1,iiii2;
	comment	RJM_VER7에 한하여 적용됨. 작동모드 8이면서 속도 0일 때에만 설정의 의미가 있음. F2811/F28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
;WNdddd1,dddd2; (#) ;WN?; (#)		RJM_VER7 analog Hall sensor type에서 구동가능한 영역에서의 analog Hall sensor wave 총갯수. 1st word : 모터1의 analog Hall sensor wave 총갯수 [3 - 63] (7) 2nd word : 모터2의 analog Hall sensor wave 총갯수 [3 - 63] (7)
	return value	WNdddd1,dddd2;
	comment	RJM_VER7 analog Hall sensor에 한하여 적용됨. 2012년 5월 1일 S/W 버전부터 wave 총갯수가 128에서 64개로 줄었음. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음. F28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
(제일메티컬의 경우) ;WJdddd1,dddd2,dddd3,dddd4,dddd5 ; (+#^) ;WJ?; (+#^)	operation	<p>analog Hall sensor type BLDC 모터의 linear interpolation 방식 자동 파라미터 설정을 개시함.</p> <p>1st word : 1 : [STEP 1] analog Hall sensor의 파라미터를 측정하여 저장하기를 개시함. 2 : [STEP 2] U상과 Hall sensor A상의 위상각차이를 측정하여 저장하기를 개시함. 3 : [STEP 3] 정역회전을 하면서 U상과 Hall sensor A상의 위상각 차이를 세밀하게 조정하여 저장하기를 개시함. 4 : [STEP 4] 설정 RPM으로 정역회전을 하면서 속도값이 미달되는 지를 검사함.</p> <p>2nd word : STEP 3의 세밀한 조정시에 Hall sensor A상의 위상각을 업데이트하는 게인 [1 - 20] 3rd word : STEP 3의 세밀한 조정시에 정역운전의 최대 허용 횟수 [1-20] 4th word : STEP 3의 세밀한 조정시에 사용하는 RPM [0-40] 5th word : STEP 1/2/3의 성공적인 완료후 최종 시험을 실시하는 RPM [0-40]</p>
	return value	<p>WJdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;</p> <p>1st word : 1 : analog Hall sensor의 파라미터를 측정중임. 2 : U상과 Hall sensor A상의 위상각차이를 측정중임. 3 : 정역회전을 하면서 U상과 Hall sensor A상의 위상각차이를 세밀하게 조정중임. 4 : 설정 RPM으로 정역회전을 하면서 속도값이 미달되는 지를 검사하는 중임. 0 : 측정동작을 개시하지 않았음. 7 : 측정동작을 완료하였음. 8 : U상과 Hall sensor A상의 위상각차이가 50 (5도) 이상 업데이트되는 오류가 있었음 9 : 측정동작에서 오류가 있었음.</p> <p>2nd word : STEP 3의 세밀한 조정시에 Hall sensor A상의 위상각을 업데이트하는 게인 3rd word : STEP 3의 세밀한 조정시에 정역운전의 최대 허용 횟수로부터 down count된 현재값 4th word : STEP 3의 세밀한 조정시에 사용하는 RPM [0-40] 5th word : STEP 1/2/3의 성공적인 완료후 최종 시험을 실시하는 RPM [0-40] 6th word : 파라미터 측정 및 저장의 진척도 count</p>

명령어	구분	설명
	comment	<p>작동모드가 0000일때만 개시가 가능함.</p> <p>STEP 1 측정이 정상적으로 이루어 졌으면 STEP 2 측정을 자동으로 실행하며, STEP 2 측정이 정상적으로 이루어 졌으면 STEP 3 측정을, STEP 3 측정이 정상적으로 이루어 졌으면 STEP 4 측정을 자동으로 실행함.</p> <p>매 스텝마다 측정이 정상이었으면 해당 파라미터를 매 스텝마다 EEPROM에 저장함.</p> <p>4th word, 5th word에서 0~40은 0~40000 RPM 을 의미함.</p> <p>리턴된 값의 첫번째 워드가 8일 때 U상과 Hall sensor A상의 위상각차이가 5도 이상 업데이트되는 오류가 있었음.</p> <p>리턴된 값의 첫번째 워드가 9일 때 (측정동작에서 오류가 있었을 때), 리턴된 마지막 워드값이</p> <ul style="list-style-type: none"> 300 이면 아날로그 홀센서의 진폭이 적절하지 못한 경우임. 1050 이면 U상과 Hall sensor A상의 위상각 차이가 30도에서 90도 범위를 벗어 난 경우임. 2400 이면 4th word로 지정된 +RPM을 200RPM 이상 미달 한 경우임. 2700 이면 4th word로 지정된 -RPM을 200RPM 이상 미달 한 경우임. 2701 이면 허용된 횟수 이내에서 STEP 3의 세밀한 조정을 실패하였음. 3200 이면 5th word로 지정된 +RPM을 200RPM 이상 미달 한 경우임. 3500 이면 5th word로 지정된 -RPM을 200RPM 이상 미달 한 경우임. 3000 이상이면 측정된 파라미터의 저장은 모두 이루어진 경우임. <p>측정된 analog Hall sensor의 파라미터값은 모터1/2에 대하여 각각 WG?; WH?; 의 리턴값임.</p> <p>측정된 U상과 Hall sensor A상의 위상각차이는 WE?; 명령어의 리턴값중에서 앞쪽의 2개임.</p> <p>설정된 파라미터 값은 측정시 오류가 없었다면 EEPROM에 저장되었음.</p> <p>F2811/F28334 RJM_VER7에서만 가능한 명령임.</p>

명령어	구분	설명
(나노모션의 linear Hall sensor의 경우) ;WJddd1; (#^) ;WJ?; (#^)	operation	analog Hall sensor type BLDC 모터의 linear interpolation 방식 자동 파라미터 설정을 개시함. 1st word : 1 : [STEP 1] analog Hall sensor1의 파라미터를 측정하여 저장하기를 개시함. 2 : [STEP 2] U상과 Hall sensor1 A상의 위상각차이를 측정하여 저장하기를 개시함.
	return value	WJddd1,ddd2; 1st word : 1 : analog Hall sensor1의 파라미터를 측정중임. 2 : U상과 Hall sensor1 A상의 위상각차이를 측정중임. 0 : 측정동작을 개시하지 않았음. 11 : 측정동작을 완료하였음. 2nd word : 파라미터 측정 및 저장의 진척도 count
	comment	작동모드가 0000일때만 개시가 가능함. STEP 1 측정이 정상적으로 이루어 졌으면 STEP 2 측정을 자동으로 실행함. 매 스텝마다 측정이 정상이었으면 해당 파라미터를 매 스텝마다 EEPROM에 저장함. 28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
(그 외의 경우) ;WJdddd1,dddd2,d ddd3,dddd4,dddd5 ; (+#^) ;WJ?; (+#^)	operation	analog Hall sensor type BLDC 모터의 linear interpolation 방식 자동 파라미터 설정을 개시함. 1st word : 1 : [STEP 1] analog Hall sensor1의 파라미터를 측정하여 저장하기를 개시함. 2 : [STEP 2] U상과 Hall sensor1 A상의 위상각차이를 측정하여 저장하기를 개시함. 3 : [STEP 3] 정역회전을 하면서 모터1의 U상과 Hall sensor A상의 위상각 차이를 세밀하게 조정하여 저장하기를 개시함. 4 : [STEP 4] 모터1을 설정 RPM으로 정역회전을 하면서 속도값이 미달되는 지를 검사함. 6 : [STEP 1] analog Hall sensor2의 파라미터를 측정하여 저장하기를 개시함. 7 : [STEP 2] U상과 Hall sensor2 A상의 위상각차이를 측정하여 저장하기를 개시함. 8 : [STEP 3] 정역회전을 하면서 모터2의 U상과 Hall sensor A상의 위상각 차이를 세밀하게 조정하여 저장하기를 개시함. 9 : [STEP 4] 모터2를 설정 RPM으로 정역회전을 하면서 속도값이 미달되는 지를 검사함. 2nd word : STEP 3의 세밀한 조정시에 Hall sensor A상의 위상각을 업데이트하는 게인 [1 - 20] 3rd word : STEP 3의 세밀한 조정시에 정역운전의 최대 허용 횟수 [1-20] 4th word : STEP 3의 세밀한 조정시에 사용하는 RPM [0-40] 5th word : STEP 1/2/3의 성공적인 완료후 최종 시험을 실시하는 RPM [0-40]
	return value	WJdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; 1st word : 1 : analog Hall sensor1의 파라미터를 측정중임. 2 : U상과 Hall sensor1 A상의 위상각차이를 측정중임. 3 : 정역회전을 하면서 모터1의 U상과 Hall sensor A상의 위상각차이를 세밀하게 조정중임. 4 : 모터1을 설정 RPM으로 정역회전을 하면서 속도값이 미달되는 지를 검사하는 중임. 6 : analog Hall sensor2의 파라미터를 측정중임. 7 : U상과 Hall sensor2 A상의 위상각차이를 측정중임. 8 : 정역회전을 하면서 모터2의 U상과 Hall sensor A상의 위상각차이를 세밀하게 조정중임. 9 : 모터2을 설정 RPM으로 정역회전을 하면서 속도값이 미달되는 지를 검사하는 중임. 0 : 측정동작을 개시하지 않았음.

명령어	구분	설명
		<p>11 : 측정동작을 완료하였음. 12 : U상과 Hall sensor A상의 위상각차이가 50 (5도) 이상 업데이트되는 오류가 있었음 13 : 측정동작에서 오류가 있었음.</p> <p>2nd word : STEP 3의 세밀한 조정시에 Hall sensor A상의 위상각을 업데이트하는 게인 3rd word : STEP 3의 세밀한 조정시에 정역운전의 최대 허용 횟수로부터 down count된 현재값 4th word : STEP 3의 세밀한 조정시에 사용하는 RPM [0-40] 5th word : STEP 1/2/3의 성공적인 완료후 최종 시험을 실시하는 RPM [0-40] 6th word : 파라미터 측정 및 저장의 진척도 count</p>
	comment	<p>작동모드가 0000일때만 개시가 가능함.</p> <p>STEP 1 측정이 정상적으로 이루어 졌으면 STEP 2 측정을 자동으로 실행하며, STEP 2 측정이 정상적으로 이루어 졌으면 STEP 3 측정을, STEP 3 측정이 정상적으로 이루어 졌으면 STEP 4 측정을 자동으로 실행함. 매 스텝마다 측정이 정상이었으면 해당 파라미터를 매 스텝마다 EEPROM에 저장함.</p> <p>4th word, 5th word에서 0~40은 0~40000 RPM 을 의미함.</p> <p>리턴된 값의 첫번째 워드가 12일 때 U상과 Hall sensor A상의 위상각차이가 5도 이상 업데이트되는 오류가 있었음.</p> <p>리턴된 값의 첫번째 워드가 13일 때 (측정동작에서 오류가 있었을 때), 리턴된 마지막 워드값이</p> <ul style="list-style-type: none"> 300 이면 아날로그 출센서의 진폭이 적절하지 못한 경우임. 1050 이면 U상과 Hall sensor A상의 위상각 차이가 30도에서 90도 범위를 벗어 난 경우임. 2400 이면 4th word로 지정된 +RPM을 200RPM 이상 미달 한 경우임. 2700 이면 4th word로 지정된 -RPM을 200RPM 이상 미달 한 경우임. 2701 이면 허용된 횟수 이내에서 STEP 3의 세밀한 조정을 실패하였음. 3200 이면 5th word로 지정된 +RPM을 200RPM 이상 미달 한 경우임. 3500 이면 5th word로 지정된 -RPM을 200RPM 이상 미달 한 경우임. 3000 이상이면 측정된 파라미터의 저장은 모두 이루어진 경우임.

명령어	구분	설명
	comment	측정된 analog Hall sensor의 파라미터값은 모터1/2에 대하여 각각 WG?; WH?; 의 리턴값임. 측정된 U상과 Hall sensor A상의 위상각차이는 WE?; 명령어의 리턴값중에서 앞쪽의 2개임. 설정된 파라미터 값은 측정시 오류가 없었다면 EEPROM에 저장되었음. F2811/F28334 RJM_VER7에서만 가능한 명령임.

명령어	구분	설명
;WKdddd1; (+#^) ;WK?; (+#^)	operation	encoder/digital_Hall_IC_only type BLDC 모터의 자동 파라미터 설정을 개시함. 1st word : 1 : [STEP 1] encoder type BLDC 모터1의 pole pair 수를 측정하여 저장하기를 개시함. 2 : [STEP 2] 모터1의 U상과 Hall sensor A상의 위상각차이를 측정하여 저장하기를 개시함. 6 : [STEP 1] encoder type BLDC 모터2의 pole pair 수를 측정하여 저장하기를 개시함. 7 : [STEP 2] 모터2의 U상과 Hall sensor A상의 위상각차이를 측정하여 저장하기를 개시함.
	return value	WKdddd1, dddd2; 1st word : 1 : encoder type BLDC 모터1의 pole pair 수를 측정중임. 2 : 모터1의 U상과 Hall sensor A상의 위상각차이를 측정중임. 6 : encoder type BLDC 모터2의 pole pair 수를 측정중임. 7 : 모터2의 U상과 Hall sensor A상의 위상각차이를 측정중임. 0 : 측정동작을 개시하지 않았음. 11 : 측정동작을 완료하였음. 12 : Hall sensor A/B/C 상이 음의 방향으로 회전함. 13 : 엔코더 값이 음의 방향으로 증가함. 14 : 계산된 pole pair 수가 0임. 15 : 계산된 pole pair 수가 50 보다 큼. 16 : Hall sensor의 low level 전위가 너무 높음. 17 : Hall sensor의 high level 전위가 너무 낮음. 18 : digital_Hall_IC_only type 이므로 STEP 2 측정을 하지 않았음. 2nd word : 파라미터 측정 및 저장의 진척도 count
	comment	작동모드가 0000일때만 개시가 가능함. pole pair 수를 측정하기 전에 digital Hall sensor의 threshold를 자동으로 설정함. STEP 1 측정이 정상적으로 이루어 졌으면 STEP 2 측정을 자동으로 실행함. 매 스텝마다 측정이 정상이었으면 해당 파라미터를 매 스텝마다 EEPROM에 저장함.

명령어	구분	설명
		리턴된 값의 첫번째 워드가 11이면 측정과 저장이 완료된 것임. digital_Hall_IC_only type BLDC 모터의 경우는 STEP 1의 Hall sensor threshold 측정만 실시함. 측정된 pole pair 수는 SE?;의 리턴값중 마지막 2개의 값임. 측정된 U상과 Hall sensor A상의 위상각차이는 WE?; 명령어의 리턴값중에서 앞쪽의 2개임.
;Wqdddd1,dddd2; (+#^) ;Wq?; (+#^)	operation	모터1의 MAX_PERCENT_SLIP1, ELECTRIC_TIME_CONSTANT1 값을 설정함. 1st word : MAX_PERCENT_SLIP1 (25 %) 2nd word : ELECTRIC_TIME_CONSTANT1 (4000 us)
	return value	Wqdddd1,dddd2;
	comment	2012년 11월 30일 S/W 버전부터 브레이크제어기/조향제어기/자석가동자에서는 적용안됨. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Wrdddd1,dddd2; (+#^) ;Wr?; (+#^)	operation	모터2의 MAX_PERCENT_SLIP2, ELECTRIC_TIME_CONSTANT2 값을 설정함. 1st word : MAX_PERCENT_SLIP2 (25 %) 2nd word : ELECTRIC_TIME_CONSTANT2 (4000 us)
	return value	Wrdddd1,dddd2;
	comment	2012년 11월 30일 S/W 버전부터 브레이크제어기/조향제어기/자석가동자에서는 적용안됨. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;WLdddd1,dddd2, d dddd3,dddd4, dddd 5,dddd6,dddd7; (+#^) ;WL?dddd1; (+#^)	operation	3상 analog Hall 센서를 사용하는 경우에 A/B/C 상신호의 교차점값을 설정함. 1st word : 3상 analog Hall 신호의 삼상 wave sector 번호 (0 ~ 63) 2nd word : C과 A상의 교차점중 큰 값 (0 ~ 4095) 3rd word : A과 B상의 교차점중 큰 값 (0 ~ 4095) 4th word : B과 C상의 교차점중 큰 값 (0 ~ 4095) 5th word : B과 C상의 교차점중 작은 값 (0 ~ 4095) 6th word : C과 A상의 교차점중 작은 값 (0 ~ 4095) 7th word : A과 B상의 교차점중 작은 값 (0 ~ 4095)
	return value	WLdddd1,dddd2, dddd3, dddd4, dddd5, dddd6, dddd7;
	comment	설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;WLPdddd1; (+#^)	operation	3상 analog Hall 센서를 사용하는 경우에 지정된 삼상 wave sector 번호부터 A/B/C 상신호의 교차점값을 자동 설정함. 1st word : 3상 analog Hall 신호의 삼상 wave sector 번호 (0 ~ 63)
	return value	WLP;
	comment	자석가동자를 -방향쪽 센서보다 더 -쪽으로 위치시킨 상태에서는 RrA55AWLP0;를 차례로 입력하고 교차점 자동설정을 개시하면 되며, -방향쪽의 위치리밋에 자석가동자가 닿은 상태에서 이미 -방향쪽 센서를 자석가동자가 덮고 있다면 그때의 삼상 wave sector 번호를 알아내어 RrA55A; WLPdddd1; 을 차례로 입력하고 교차점 자동설정을 개시해야함. -쪽 위치리밋에 닿아 있을때의 삼상 wave sector 번호를 알아내는 방법은 자석가동자를 +방향쪽 센서보다 더 +쪽으로 위치시킨 상태에서 RrA55A;를 입력한 다음 -방향쪽으로 -방향 위치리밋까지 움직여서 moving_magnet_sensor_cnt 변수값을 읽으면 됨. 설정된 파라미터 값은 ESOC00,0300; 명령어로 EEPROM에 저장될 수 있음.
;WLQ; (+#^)	operation	3상 analog Hall 센서를 사용하는 자석가동자의 경우에 -방향쪽 센서와 +방향쪽 센서의 위상차를 자동 설정함.
	return value	WLQ;
	comment	좌측 센서의 에지를 감지할수 있는 위치에서 RrA55A;를 실행한 후, 자석가동자를 움직여서 좌측 센서의 에지만 검출한 상태를 만들고 나서, WLQ; 명령을 주고 자석가동자를 + 방향으로 서서히 움직여서 우측센서 에지가 감지되도록 함. 설정된 값은 SR?; 의 리턴값에서 마지막 두번째 값임. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;WLR; (+#^)	operation	3상 analog Hall 센서를 사용하는 자석가동자의 경우에 순위가 높은 제어기의 +방향쪽 센서와 순위가 낮은 현재 제어기의 -방향쪽 센서 사이의 위상차를 자동 설정함.
	return value	WLR;
	comment	순위가 높은 제어기의 -방향쪽 또는 +방향쪽 센서의 에지를 검출하도록 자석가동자를 움직인 상태에서 (단 이때 순위가 낮은 현재 제어기의 -방향쪽 또는 +방향쪽 센서의 에지가 검출되면 안됨) 순위가 낮은 현재 제어기에 RrA55A; WLR; 명령을 차례로 주고 가동자를 손으로 + 방향으로 서서히 움직여서 현재 제어기의 우측센서 에지가 감지되지 위치까지 이동시키면 됨. 설정된 값은 SR?; 의 리턴값에서 마지막 값임. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;WSddd1,ddd2; (+#^)	operation	위치제어모드에서 속도에 비례하는 damping 회전력을 발생시키는 계인을 설정함. 1st word : SPEED1_FEEDBACK_GAIN [0 ~ 30000] (bit) 2nd word : SPEED2_FEEDBACK_GAIN [0 ~ 30000] (bit)
	return value	WSddd1,ddd2;
;WS?; (+#^)	comment	(RPM * SPEEDx_FEEDBACK_GAIN) / 4096 의 크기 만큼 전류명령값을 감소시킴. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Wsddd1,ddd2; (+#^)	operation	DD motor 등에서 속도에 비례하는 damping current를 발생시키는 계인을 설정함. 1st word : SPEED1_DAMPING_GAIN [0 ~ 30000] (bit) 2nd word : SPEED2_DAMPING_GAIN [0 ~ 30000] (bit)
	return value	Wsddd1,ddd2;
;Ws?; (+#^)	comment	(PWM 주파수의 기간동안 변화된 위치 펄스수 * SPEEDx_DAMPING_GAIN) / 4096 의 크기 만큼 전류명령값을 감소시킴. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;Wciii1,iii2; (+#^)	operation	유도모터의 경우에 Id 전류의 기준값을 설정함. 1st word : 모터1의 Id 전류의 기준값 (0 bit) 2nd word : 모터2의 Id 전류의 기준값 (0 bit)
	return value	Wciii1,iii2;
;Wc?; (+#^)	comment	유도모터의 경우에만 사용할수 있음.

명령어	구분	설명
;WTdddd1,dddd2,dddd3,dddd4; (+#^) ;WT?; (+#^)	operation	BLDC 모터에 있어서 cogging torque를 보상하는 파라미터를 설정함. 1st word : 모터1의 cogging torque 보상 진폭 [0 ~ 1023] (0 bit) 2nd word : 모터1의 cogging torque 위상 [0 ~ 1023] (0 bit) 3rd word : 모터2의 cogging torque 보상 진폭 [0 ~ 1023] (0 bit) 4th word : 모터2의 cogging torque 위상 [0 ~ 1023] (0 bit)
	return value	WTdddd1,dddd2,dddd3,dddd4;
	comment	cogging torque 보상 진폭이 0이면 cogging torque 보상을 하지 않는 것이며, cogging torque 보상 진폭이 0이 아니면, cogging torque 보상 진폭의 사인파 전류값을 feedback에 의한 전류 명령값에 더하여 전류제어 명령값이 되게함. cogging torque 보상 진폭의 단위는 SC명령에서 사용하는 전류크기의 단위와 같음. cogging torque 위상은 0~1023일때 0~360도의 위상각을 의미하며, 회전자기준으로 반시계방향으로 90도 앞선 위치를 기준으로한 위상각임. 2012년 6월 12일 이전 S/W 버전에서는 cogging torque 보상 진폭설정에 문제가 있어서 제대로 작동하지 않음. 2012년 12월 27일 S/W 버전부터는 TMS320F2811을 사용하는 H/W 버전으로서 analog Hall 센서를 사용하는 경우에 WT 명령 적용이 되지 않음. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;WAdddd1,dddd2; (+#^) ;WA?; (+#^)		속도모드에서 속도오차가 클때 속도명령값을 수정하는 속도오차의 최소크기를 설정함. 1st word : 모터1의 속도오차 최소크기 [1 - 10000] (200 or 50 RPM) 2nd word : 모터2의 속도오차 최소크기 [1 - 10000] (200 or 50 RPM)
	return value	WAdddd1,dddd2;
	comment	2013년 3월 27일 S/W 버전부터 사용가능함. 엔코더 분해능이 매우 적은 경우에 사용하는 S/W 버전과 digital Hall IC 만을 사용하는 경우는 기본값이 200rpm 이며, 그외는 50 rpm이 기본값임. 설정된 파라미터 값은 ES 명령어로 EEPROM에 저장될 수 있음.

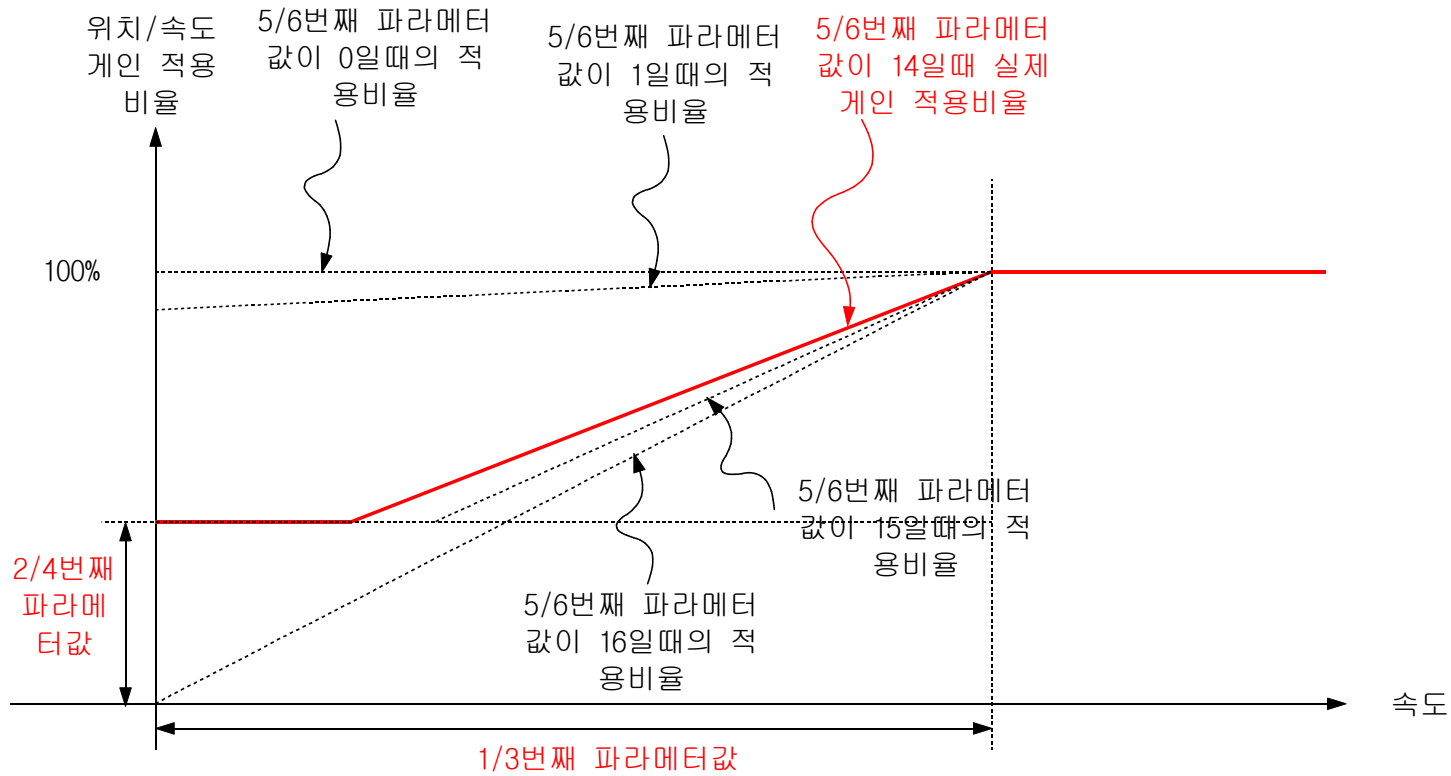
명령어	구분	설명
;Wkdddd1,dddd2; (+#^) (2011년 6월 14일 이전 S/W 버전)	operation	RJM_VER7 BLDC 모터제어기에서, 모드8의 host direct angle control을 사용하여 자동튜닝 등의 구동을 할 때, 사용하는 PWM duty 값을 설정함. 1st word : 모터1의 PWM duty 값 [0 ~ PWM_PERIOD/2] (bit) 2nd word : 모터2의 PWM duty 값 [0 ~ PWM_PERIOD/2] (bit) 3rd word : analog Hall 센서의 파라미터 측정을 위한 시험속도 [20 ~ 1000] (rpm) 4th word : analog Hall 센서의 상전압의 교차점 전압(높은 전압쪽) 크기 [100 ~ 4000] (bit) 5th word : analog Hall 센서의 상전압의 교차점 전압(낮은 전압쪽) 크기 [100 ~ 4000] (bit)
	return value	Wkdddd1,dddd2; (2011년 6월 14일 이전 S/W 버전) Wkdddd1,dddd2,dddd3,dddd4,dddd5; (2011년 6월 14일 S/W 버전부터)
	comment	모터제어기가 TMS320F2810/11/12을 사용하였을 때, 기본값은 analog Hall 센서를 사용하는 경우로서 mini VER7인 경우는 45이며 나머지는 55이고, digital Hall 센서 또는 엔코더를 사용하는 경우로서 mini VER7인 경우는 50이며 나머지는 60임. 모터제어기가 TMS320F28334/5를 사용한 경우는 TMS320F2810/11/12을 사용하였을 때의 2배 값임. 과도한 값을 사용하면 모터가 과열되니 주의할 것. 2013년 1월 02일 S/W 버전부터는 3rd word 값의 하한치가 200에서 20으로 바뀌었음. 2012년 2월 11일 S/W 버전부터는 1st/2nd word 의 경우 EEPROM에 저장되는 변수로 바뀌었음.
;WPdddd1,dddd2; (+#^) ;WP?; (+#^)	operation	Digital Hall sensor type의 RJM_VER7 모터에 대하여 정지희망 edge 번호를 지정함. 1st word : 모터1의 정지희망 edge 번호 [0 ~ 모터1의_pole_pair_수*6-1] 2nd word : 모터2의 정지희망 edge 번호 [0 ~ 모터2의_pole_pair_수*6-1]
	return value	WPdddd1,dddd2;
	comment	Digital Hall sensor type의 모터에서 05번 모드로 회전을 하다가 PeA5A5;/Pe5A5A;/PeA55A; 명령어로 정지할 때 정지를 희망하는 에지의 번호를 지정함. 최초 전원 투입후에 모터를 enable 및 02번 모드 설정을 실시하고, PA5000000,5000000;을 수행하여 정지된 위치에서 WP?;로 읽은 에지번호가 기준위치이며, 05번 모드로 회전을 하다가 PeA5A5;/Pe5A5A;/PeA55A; 명령어로 정지하면 같은 위치에 정지함. 만약 정지 위치를 바꾸고자 하면 WP 명령어로 정지희망 edge 번호를 바꾸어 주고 재기동 및 정지를 실시하면 됨.

명령어	구분	설명
;Wpdddd1, dddd2, d ddd3, dddd4, dddd5 ; (+#^)	operation	Digital Hall sensor type의 RJM_VER7 모터에 대하여 감속완료후 안정화를 개시하는 지연시간과 안정화용 analog Hall sensor의 offset, 및 analog Hall 센서가 사용되는 위치오차 범위등을 설정함. 1st word : 감속완료후 안정화를 개시하는 지연시간 [0 ~ 9999] (100 10ms) 2nd word : 안정화용 analog Hall sensor의 offset [0 ~ 4095] (2000 bit) 3rd word : analog Hall sensor를 적용하는 위치오차 범위 [0 ~ 8000] (1500 bit) 4th word : analog Hall sensor가 적용되는 기간에 사용하는 게인 factor [0 ~ 30000] (16384) 5th word : PvA55A;로 감속할때, 처음에는 속도모드로 감속하다가, 낮은 속도가 되면 위치모드로 정지하게 되는데, 그때의 경계속도값 [1 ~] (10)
	return value	Wpdddd1, dddd2, dddd3, dddd4, dddd5;
	comment	Digital Hall sensor type의 RJM_VER7 BLDC 모터에서 WAFER_ROTATION==1인 경우에만 적용됨. 안정화용 analog Hall sensor는 모터2의 Hall sensor A입력 단자에 연결되어야하며 모터1의 Hall sensor A 입력과 상승하강 에지가 일치하여야함. 네번째 워드의 값이 16384이면 고속에서 적용하는 게인을 적용하는 것이며, 작을수록 게인을 적게하는 것임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;WMHLHL; (+#^)	operation	Wafer 회전위치제어시에 위치모드로 정지한 한 상태에서 속도모드로 작동모드를 변경하며, 이에 따른 변수 초기화 작동을 실시함. 1st word : 작동모드 [0000 - FFFF] (0000)
	return value	WMHLHL;
	comment	⚙ 1st word의 값이 0005 이어야만 초기화작동이 개시되며, 작동모드가 이미 0005 이면 초기화작동이 개시되지 않음. 1st word의 값이 0005 이외의 값이면 작동모드를 설정하지 않고 현재의 작동모드값을 보며 주기만 함.

명령어	구분	설명
;WDA55A, dddd1, dd dd2; (+#^) ;WD?; (+#^)	operation	미분값 계산을 실시하는 시간차를 설정함. 1st word : 모터1의 제어에서 미분값 계산을 실시하는 시간차 [1/2/4/8] (2 ms) 2nd word : 모터2의 제어에서 미분값 계산을 실시하는 시간차 [1/2/4/8] (2 ms)
	return value	WDddd1, dddd2;
	comment	<ul style="list-style-type: none"> ⚡ 1,2,4,8 만 유효한 값이며 그 이외의 값을 주면 2로 처리됨. ⚡ 2012년 10월 6일 S/W 버전부터 WD 파라미터의 값이 바뀌더라도 미분계인은 변하지 않도록 조치하였음. <p>따라서 2012년 10월 6일 이전 S/W 버전을 사용하다 이후 S/W 버전의 firmware로 upgrade 하면서 파라미터는 이전의 것을 사용하는 경우로서 WD1,1; 또는 WD4,4; 또는 WD8,8;을 설정하여 사용하던 경우는 SQ/SR 명령어에서 위치-inverse-D-gain을 작거나(*2 WD1,1;인경우) 크게(/2 WD4,4;인경우, /4 WD4,4;인경우) 바꾸어주어야함.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;Wdddd1,dddd2,dddd3,dddd4,dddd5,dddd6; (+#^) ;Wd?; (+#^)	operation	<p>inner 속도제어루프가 있는 03번 위치제어모드의 속도제어 또는 Digital Hall sensor만을 사용하는 BLDC모터의 위치/속도 제어 또는 0A번 속도제어모드 또는 속도제어로서 SX 명령어로 설정하는 두번째 파라미터의 bit11이 0인 경우에, 낮은 속도에서 위치/속도 제어게인(3번 위치제어모드의 경우는 속도게인만)을 낮게하는 파라미터를 설정함.</p> <p>1st word : 모터1의 위치/속도게인을 정상값으로 적용하는 모터속도의 최소값 [1 ~ 2000] (1000 rpm) 2nd word : 모터1의 위치/속도게인을 스케일링하는 스케일의 최소값 [1 ~ 100] (100 %) 3rd word : 모터2의 위치/속도게인을 정상값으로 적용하는 모터속도의 최소값 [1 ~ 2000] (1000 rpm) 4th word : 모터2의 위치/속도게인을 스케일링하는 스케일의 최소값 [1 ~ 100] (100 %) 5th word : 모터1의 위치/속도게인을 낮게하는 비율 [1 ~ 16] (16) 6th word : 모터2의 위치/속도게인을 낮게하는 비율 [1 ~ 16] (16)</p>
	return value	Wdddd1,dddd2,dddd3,dddd4,dddd5,dddd6;
	comment	<p>inner 속도제어루프가 있는 03번 위치제어모드의 속도제어 또는 Digital Hall sensor만을 사용하는 BLDC모터의 위치/속도 제어 또는 0A번 속도제어모드 또는 속도제어로서 SX 명령어로 설정하는 두번째 파라미터의 bit11이 0인 경우에, 속도에 비례하여 위치/속도게인(3번 위치제어모드의 경우는 속도게인만)을 스케일링하는데, 1st/3rd word에 해당하는 속도 이상에서는 scale factor는 1이며, 그 이하의 속도에서는 속도에 비례하여 scale factor 값이 낮아지는데 따라서 1보다 작은 값이 됨.</p> <p>scale factor는 2nd/4th word 로 지정한 값 이상으로 그 크기가 제한되며, 이 값이 100%이면 게인을 낮추는 작동을 하지 않는 것임. Qx77; 을 사용하면 적용되는 게인의 변화를 볼수 있음. (Qx77; 에서의 값이 16384일때 게인으로는 1.0에 해당함)</p> <p>Wafer 제어인 경우에는 기본값이 200rpm, 20%임.</p> <p>모터x의 위치/속도게인을 낮게하는 비율이 16이면 속도에 비례하여 게인을 낮추는 것이고, 15이면 속도의 93.75% 비율로만 낮추는 것이며 (덜 낮추는 것임), 1이면 속도의 6.25% 비율로만 낮추는 것임 (가장 적게 낮추는 것임). 03/0A번 제어모드에서는 낮추는 비율을 1/2만큼 작게 적용함.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

Wd 명령에 의한 위치/속도 게인 적용 비율



명령어	구분	설명
;RQq; (+#^)	operation	1ms 단위로 CAN 통신을 통하여 동기위치제어를 할때 모터1의 현재의 위치값을 position1_ref 변수에 저장함.
	return value	RQHHLHLHL; 1st long word : position1_ref
;RQ?; (+#^)	comment	1ms 단위로 CAN 통신을 통하여 동기위치제어를 할때, FF번 주소(broadcasting용 주소)로 상대위치명령값의 전송을 개시하기 전에 반드시 한번을 실행하여야만 함. 이렇게 함으로서 FF번 주소(broadcasting용 주소)로 상대위치명령값 0을 전송하였을때 현재의 위치를 고수하는 위치명령을 인가한것과 같게 됨.
;RRr; (+#^)	operation	1ms 단위로 CAN 통신을 통하여 동기위치제어를 할때 모터2의 현재의 위치값을 position2_ref 변수에 저장함.
	return value	RRHHLHLHL; 1st long word : position2_ref
;RR?; (+#^)	comment	1ms 단위로 CAN 통신을 통하여 동기위치제어를 할때, FF번 주소(broadcasting용 주소)로 상대위치명령값의 전송을 개시하기 전에 반드시 한번을 실행하여야만 함. 이렇게 함으로서 FF번 주소(broadcasting용 주소)로 상대위치명령값 0을 전송하였을때 현재의 위치를 고수하는 위치명령을 인가한것과 같게 됨.
;RIA55A; (+#^)	operation	모터1/2의 적분오차값을 강제로 0으로 설정함.
	return value	RIA55A;
	comment	
;RIA5A5; (+#^)	operation	모터1의 적분오차값을 강제로 0으로 설정함.
	return value	RIA5A5;
	comment	
;RI5A5A; (+#^)	operation	모터2의 적분오차값을 강제로 0으로 설정함.
	return value	RI5A5A;
	comment	

명령어	구분	설명
;RrA55A; (+#^)	operation	모터제어기를 리셋함.
	return value	N.A.
	comment	<p>전원투입시의 초기화과정이 다시 시작됨. 2011년 12월 4일 S/W 버전부터 사용가능한 명령임. precharge 릴레이가 있는 72V 이상의 고전압 H/W 버전에서는 2012년 6월 30일 S/W 버전부터 PWM출력을 먼저 disable 한 다음 100ms(단 자동화용과 BRAKE_ACTUATOR는 200ms, ELECTRONIC_POWER_STEER는 10ms) 후에 precharge 릴레이를 끄면서 전원투입시의 초기화과정이 다시 시작됨.</p> <p>상위제어기가 RrA55A; 명령을 보내지 않더라도, 모터제어기가 스스로 재부팅을 하는 경우는 다음과 같음. PWM 출력이 disable되어 있고, 작동모드 00인 경우로서 (BRAKE_ACTUATOR ELECTRONIC_POWER_STEER)==1 일때는 ((fault_status&0x041d)!=0) ((fault1_status&0x0410)!=0) 이거나 상위제어기에서 CAN message를 사용하여 리셋 코드를 보내어 왔을때 전류 offset값을 저장하고, CANRX0_Buf[253]를 증가시키고 RrA55A; 실행. (BRAKE_ACTUATOR ELECTRONIC_POWER_STEER)==0 일때는 ((fault_status&0x041d)!=0) ((fault1_status&0x0400)!=0) 이면서 (operation_mode_SWITCH & 0x40000000L)!=0)&&(RJM_VER7_Hall_IC_analog==0) 이면 전류 offset값을 저장하고, CANRX0_Buf[253]를 증가시키고 RrA55A; 실행. 자동 재부팅후에 PE 명령을 자동으로 실행한 다음, CANRX0_Buf[253]!=0 이면 저장된 전류 offset 값으로 적용 전류offset 값을 설정하고 ZB 명령어로 설정하는 하위바이트의 값에 따라서 작동모드를 설정함.</p>

명령어	구분	설명
;RSHLHL; (+#^) ;RS?; (+#^)	operation	비상정지시의 정지 형태를 정의함. bit0~3 : 1이면 모터1이 비상정지시에 작동모드를 0으로 설정하여 정지함. (PWM은 켜있음) 2이면 모터1이 So 명령어로 설정하는 감속률로 비상정지를 실시함. 3이면 모터1이 PWM 출력을 끄는 방식으로 비상정지를 실시함. bit4~7 : 1/2/4이면 리미트스위치 LS1A/B/C 가 low 일때 모터1이 비상정지를 실시함. bit8~11 : 1이면 모터2가 비상정지시에 작동모드를 0으로 설정하여 정지함. (PWM은 켜있음) 2이면 모터2가 So 명령어로 설정하는 감속률로 비상정지를 실시함. 3이면 모터2가 PWM 출력을 끄는 방식으로 비상정지를 실시함. bit12~15 : 1/2/4이면 리미트스위치 LS2A/B/C 가 low 일때 모터2가 비상정지를 실시함.
	return value	RSHLHL;
	comment	비상정지는 Rs 명령에 의하여서도 실시될수 있으며 리미트스위치 입력에 의하여서도 실시될수 있음. 2012년 1월 19일 S/W 버전부터 사용가능한 명령임.
;RsA55A; (+#^)	operation	모터1과 모터2가 명령에 의한 비상정지를 실행함.
	return value	RsA55A;
	comment	비상정지의 형태는 RSHLHL; 에 따름. 2011년 12월 4일 S/W 버전부터 사용가능한 명령임.
;RsA5A5; (+#^)	operation	모터1이 명령에 의한 비상정지를 실행함.
	return value	RsA5A5;
	comment	비상정지의 형태는 RSHLHL; 에 따름. 2011년 12월 4일 S/W 버전부터 사용가능한 명령임.
;Rs5A5A; (+#^)	operation	모터2가 명령에 의한 비상정지를 실행함.
	return value	Rs5A5A;
	comment	비상정지의 형태는 RSHLHL; 에 따름. 2011년 12월 4일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;RBHLHL; (+#^) ;RB?; (+#^)	operation	<p>feedback 제어를 개시하도록 작동모드 설정에 필요한 조치를 자동으로 처리하게 함.</p> <p>1st word : A55A or 0202: 전류제어기반의 위치제어모드(02번 제어모드) 작동에 필요한 과정을 처리함. : 5AA5 or 0303 : 속도제어기반의 위치제어모드(03번 제어모드) 작동에 필요한 과정을 처리함. : A5A5 or 0505 : 전류제어기반의 속도제어모드(05번 제어모드) 작동에 필요한 과정을 처리함. : 5A5A or 0606 : PWM제어기반의 속도제어모드(06번 제어모드) 작동에 필요한 과정을 처리함. : AA55 or 0707 : 전류제어모드(07번 제어모드) 작동에 필요한 과정을 진행함. : 55AA or 0A0A : 전류제어와 펄스주기계산법 기반의 속도제어모드(0A번 제어모드) 작동에 필요한 과정을 진행함. (2012년 12월 30일 S/W 버전부터 사용가능함.) : 0101 : 위치제어모드(01번 제어모드) 작동에 필요한 과정을 진행함. : 0404 : 속도제어모드(04번 제어모드) 작동에 필요한 과정을 진행함. : 0808 : host direct angle control 제어모드(08번 제어모드) 작동에 필요한 과정을 진행함. : 0909 : host direct angle control 제어모드(09번 제어모드) 작동에 필요한 과정을 진행함. : 0E0E : 사인파 제어모드(03번 제어모드 사용) 작동에 필요한 과정을 진행함. : 0F0F : 사인파 제어모드(02번 제어모드 사용) 작동에 필요한 과정을 진행함.</p>
	return value	<p>RBHLHL; 1st word : 처리과정을 보여주는 카운터, 카운터가 0이면 처리가 끝났음을 의미함.</p>
	comment	<p>전원투입후로서 모터의 PWM이 켜지지 않았고 (PE 명령이 수행되지 않은 것과 같음), 작동모드도 00일때만 적용이 가능한 명령임.</p> <p>처리과정에는 약간의 시간이 소요되며, 모터제어기에 따라서는 전류 offset 설정 및 초기 위치 설정등의 과정을 실시하기도 함.</p> <p>위상각을 알수 없는 analog형 위치센서를 가지고있는 경우는 모터를 움직이는 초기화 동작이 실시될수 있음.</p> <p>2011년 12월 15일 S/W 버전부터 사용가능한 명령임.</p> <p>2012년 4월 10일 S/W 버전부터 부팅을 완료할때 상전류의 크기가 1A 이상이면 작동모드를 00으로 강제로 바꾸어서 모터의 작동을 차단하며, RB0101/0202/0303/0404/0505/0606/0707/0808/0909/0A0A/0E0E/0F0F;를 사용할수 있음.(단 0A0A는 2012년 12월 30일 S/W 버전부터 사용가능함.) 일부 작동모드는 정상작동하지 않는 것이 있을수 있음.</p>

명령어	구분	설명
;RJI; (+#^)	operation	fault code를 저장한 RAM buffer를 읽어서 fault의 내용을 문자로 보여주는 index를 가장 최근의 fault code를 지정하도록 초기화함.
	return value	RJn<AAAAAAAAAAAA>; n : fault code가 저장된 RAM buffer의 index로서 가장 최근에 발생한 오류코드의 index. (0 - 7) AAAAAAAAAAAA : n의 index에 저장된 fault code의 내용
	comment	RJ; 명령어로 한개를 읽으면 1씩 감소하여 다음번에 RJ; 명령을 보내면 이전의 fault의 내용을 문자로 보여줌. 전원을 켜다음 처음으로 RJ;명령을 보내면 가장 최근의 fault 내용을 보여줌. 오류코드가 전혀 없으면 <No fault> 가 보여짐. 전원을 투입할때 오류코드저장용 RAM 버퍼는 0으로 clear 됨. 따라서 RJ 명령어로 보여지는 오류코드는 전원투입 이후에 발생한 것임. 2012년 8월 18일 S/W 버전부터 사용가능한 명령임.
;RJ; (+#^)	operation	fault code를 저장한 RAM buffer를 읽어서 fault의 내용을 문자로 보여줌.
	return value	RJn<AAAAAAAAAAAA>; n : fault code가 저장된 RAM buffer의 index. (0 - 7) AAAAAAAAAAAA : n의 index에 저장된 fault code의 내용
	comment	RJ; 명령어로 한개를 읽으면 1씩 감소하여 다음번에 RJ; 명령을 보내면 이전의 fault의 내용을 문자로 보여줌. 전원을 켜다음 처음으로 RJ;명령을 보내면 가장 최근의 fault 내용을 보여줌. 오류코드가 전혀 없으면 <No fault> 가 보여짐. 전원을 투입할때 오류코드저장용 RAM 버퍼는 0으로 clear 됨. 따라서 RJ 명령어로 보여지는 오류코드는 전원투입 이후에 발생한 것임. 2012년 8월 18일 S/W 버전부터 사용가능한 명령임.
;RJR; (+#^)	operation	fault code를 저장한 RAM buffer를 지움.
	return value	RJR;
	comment	2012년 11월 03일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;RjI; (+#^)	operation	EEPROM에 저장된 fault code를 읽어서 fault의 내용을 문자로 보여주는 index를 가장 최근의 fault code를 지정하도록 초기화함.
	return value	Rj n old<AAAAAAAAAAAA>; 또는 Rj n new<AAAAAAAAAAAA>; n : fault code가 저장된 EEPROM buffer의 index로서 가장 최근에 발생한 오류코드의 index (0~14). _old : 현재의 전원투입 이전에 발생한 오류코드임을 의미함. _new : 현재의 전원투입 이후에 발생한 오류코드임을 의미함. AAAAAAAAAAAA : n의 index에 저장된 fault code의 내용
	comment	Rj; 명령어로 한개를 읽으면 1씩 감소하여 다음번에 Rj; 명령을 보내면 그 이전의 fault의 내용을 문자로 보여줌. 전원을 켜다음 처음으로 Rj;명령을 보내면 가장 최근의 fault 내용을 보여줌. 오류코드가 전혀 없으면 <No fault> 가 보여짐. 전원을 투입할때 오류코드저장용 EEPROM 버퍼는 0으로 clear 되지 않음. 따라서 Rj 명령어로 보여지는 오류코드는 전원투입 이후와 이전을 포함하여 가장 최근의 오류코드임. 2012년 8월 18일 S/W 버전부터 사용가능한 명령임. 설정된 값은 EEPROM에 저장된 값임.

명령어	구분	설명
;Rj; (+#^)	operation	EEPROM에 저장된 fault code를 읽어서 fault의 내용을 문자로 보여줌.
	return value	Rj _n old<AAAAAAAAAAAA>; 또는 Rj _n new<AAAAAAAAAAAA>; n : fault code가 저장된 EEPROM buffer의 index (0~14). _old : 현재의 전원투입 이전에 발생한 오류코드임을 의미함. _new : 현재의 전원투입 이후에 발생한 오류코드임을 의미함. AAAAAAAAAAAA : n의 index에 저장된 fault code의 내용
	comment	Rj; 명령어로 한개를 읽으면 1씩 감소하여 다음번에 Rj; 명령을 보내면 그 이전의 fault의 내용을 문자로 보여줌. 전원을 켜다음 처음으로 Rj;명령을 보내면 가장 최근의 fault 내용을 보여줌. 오류코드가 전혀 없으면 <No fault> 가 보여짐. 전원을 투입할때 오류코드저장용 EEPROM 버퍼는 0으로 clear 되지 않음. 따라서 Rj 명령어로 보여지는 오류코드는 전원투입 이후와 이전을 포함하여 가장 최근의 오류코드임. 2012년 8월 18일 S/W 버전부터 사용가능한 명령임. 2013년 1월 16일 이전 S/W 버전에서는 정상적인 전원 off 시에도 저전압오류를 검출하고 오류코드가 저장되는 경우가 있으며, 각종 오류코드의 저장 자체가 제대로 안되는 경우도 있음. 설정된 값은 EEPROM에 저장된 값임.
;RjR; (+#^)	operation	EEPROM에 저장된 fault code를 지움. (단 EsA55A;를 실행하여야 지운값이 저장됨)
	return value	RjR;
	comment	2012년 11월 03일 S/W 버전부터 사용가능한 명령임. 설정된 값은 EEPROM에 저장된 값임.

명령어	구분	설명
;RCddd1,ddd2; (+#^) ;RC?; (+#^)	operation	CAN 이나 RS485 통신이 일정시간 두절되었을 때를 감지하는 time threshold 파라미터를 설정함. ddd1 : TIME_threshold_of_BROKEN_COMM_FAILL [2 - 60000] (1000 ms) ddd2 : BROKEN_COMM_FAIL_flg [0 1] (0 bit)
	return value	RCddd1,ddd2,ddd3,ddd4; ddd1 : TIME_threshold_of_BROKEN_COMM_FAILL [2 - 60000] (1000 ms) ddd2 : BROKEN_COMM_FAIL_flg [0 1] (0 bit) ddd3 : BROKEN_COMM_FAIL_timer ddd4 : SHUTDOWN_by_BROKEN_COMM_FAIL_timer
	comment	SX 명령어 두번째 파라미터의 bit30 이 0이면 CAN 이나 RS485 통신이 일정시간 두절되었을 때 강제 정지가 실행됨. 이때 강제정지 감속도는 So 명령에 따름. CAN 이나 RS485 통신이 일정시간 두절된 오류가 발생되었을 때, 오류상태를 지우려면 RCddd1,0; 를 보내면 되나, 이전에 강제정지를 수행하고 있었다면 강제정지가 완료된 후에만 오류상태를 지울수 있음. 2013년 10월 23일 S/W 버전부터 사용가능한 명령임. 첫번째 파라미터 설정된 값은 EEPROM에 저장된 값임.

명령어	구분	설명
;Rcddd1; (+#^) ;Rc?; (+#^)	operation	CAN 이나 RS485 통신에서 idle mode로 진입하거나 또는 정상통신모드로 진입하는 것을 설정함. 2013년 11월 05일 S/W 버전부터 사용가능한 명령임.
	return value	Rcddd1; ddd1 : idle_mode_flg [0 1] (0 bit)
	comment	<p>☞ SX 명령어로 설정하는 두번째 파라미터의 bit30 이 1 이면 (2013년 10월 23일 이전의 S/W 버전 에서 사용하던 통신기능과 호환됨)</p> <ul style="list-style-type: none"> - 전원 투입 직후 RS485/CAN/RS232C 통신에 관계없이 모든 명령어를 처리하는 상태가 됨. (engineering mode) - Rc 명령어로 idle mode로 진입하는 flag를 세트하면 idle mode 상태로 바뀜. - idle mode에서는 RC/Rc/SX 명령만 처리하며 (단 RS232C 통신인 경우는 모든 명령어를 처리함), Rc 명령어로 idle mode로 진입하는 flag를 클리어하면 engineering mode 상태로 돌아감. <p>☞ SX 명령어로 설정하는 두번째 파라미터의 bit30 이 0 이면 (safe mode 기능이 추가로 적용됨)</p> <ul style="list-style-type: none"> - 전원 투입 직후에는 RC/Rc/SX 명령만 처리함. (idle mode) 단 RS232C 포트를 통하여 입력되는 명령어는 모두 처리함. - 이후에 Rc 명령어로 idle mode로 진입하는 flag를 클리어하면 safe mode 상태로 바뀜. - safe mode 상태에서는 RS485/CAN 통신을 지속적으로 하여야 하며, 모든 명령어를 사용할 수 있으며, 일정시간(RC 명령어로 설정함) 이상 RS485/CAN 통신이 두절되면 자동으로 강제 정지하고, idle mode로 진입하는 flag를 자동으로 세트하여 idle mode로 진입함. - idle mode에서는 RC/Rc/SX 명령만 처리하며, 정지상태에서 Rc 명령어로 idle mode로 진입하는 flag를 다시 클리어하여야만 safe mode 상태로 바뀜.

명령어	구분	설명
;XSHL; (+#^)	operation	순서제어에서 command를 설정하거나 제어상태를 읽어냄. (프로아텍 전용)
	return value	XSHLHLHLHL;
;XS?; (+#^)	comment	XS로 설정하는 값의 bit0/1/2/3은 리턴값의 bit8/9/10/11에 반영됨. 프로아텍 초기 S/W 버전에서만 유효한 명령임.

☞ XS?; 명령어로 리턴되는 32bit 파라미터의 비트별 의미는 다음과 같음.

32bit 리턴값	단자	기능	설정 주체
LWORD.bit0	LS1A	forward limit sensor (0 if turned ON)	forward limit sensor
LWORD.bit1	LS1B	backward limit sensor (0 if turned ON)	backward limit sensor
LWORD.bit2	LS1C	move command switch (forward if 0, backward if 1)	move command switch
LWORD.bit3	LS2A	jam sensor (0 if turned ON)	jam sensor
LWORD.bit4	LS2B	speed select switch (low speed if 1, high speed if 0)	speed select switch
LWORD.bit5	LS2C	recover command switch (0 if turned ON)	recover command switch
LWORD.bit8		move forward command bit (1 if forward move)	move command switch
LWORD.bit9		move backward command bit (1 if backward move)	move command switch
LWORD.bit12		recovering status (1 if recovering status)	motor controller
LWORD.bit13		error status (1 if error status)	motor controller
LWORD.bit14		move forward status bit (1 if forward move status)	motor controller
LWORD.bit15		move backward status bit (1 if backward move status)	motor controller
LWORD.bit29	LS1D/FAULT1	red lamp	motor controller
LWORD.bit30	LS2D/FAULT2	buzzer	motor controller

명령어	구분	설명
;XViiii1,iiii2,iiii3; (+#^)	operation	순서제어에서 사용하는 X_speed1_low, X_speed1_mid, X_speed1_high 를 설정함. (프로아텍 전용) 1st integer : X_speed1_low [-32768 ~ 32767] (1000 rpm or 50 rpm) 2nd integer : X_speed1_medium [-32768 ~ 32767] (1000 rpm or 50 rpm) 3rd integer : X_speed1_high [-32768 ~ 32767] (1000 rpm or 50 rpm)
	return value	XViiii1,iiii2,iiii3;
;XV?; (+#^)	comment	이 값은 자동화 시퀀스제어에서 사용하는 값임. 프로아텍 초기 S/W 버전에서만 유효한 명령임. RJM_VER/6의 경우에 기본값이 50임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;X0HL,HL; (+#^)	operation	EXT_FAULT1_port와 EXT_FAULT2_port를 강제로 set 또는 clear 함. 1st hexcode : EXT_FAULT1_port의 출력값 [0(OFF) ~ 1(ON)] (0) 2nd hexcode : EXT_FAULT2_port의 출력값 [0(OFF) ~ 1(ON)] (0)
	return value	X0HL,HL;
;X0?; (+#^)	comment	자동화용 제어기를 제외하고는 ZC 파라미터로 설정하는 값의 bit0/bit1이 1일 때만 X0 명령에 의한 강제설정이 가능함 (2014년 1월 27일 S/W 버전부터는 ZC로 설정된 값에서 bit2/3이 0 이어야하는 조건도 추가 됨). ZC 파라미터로 설정하는 값의 bit0/bit1이 0일 때는, 주문자의 요청에 의하여 해당되는 EXT_FAULTx 단자에 +5V가 연결되어 있는 상태이며, 이때 해당 단자는 50mA 이내의 전류를 제어기 밖에서 사용할수 있는 전원단자의 역할을 함. ZC 파라미터로 설정하는 값의 bit0/bit2는 EXT_FAULT1_port의 출력에 적용되는 값이며, bit1/bit3은 EXT_FAULT2_port의 출력에 적용되는 값임.
(스텝모터인 경우) ;Xodddd1; (#^)	operation	스텝모터에서 50V 승압회로와 역기전력 해소 pass를 ON/OFF 함. 1st word : 0 : 50V 승압회로 OFF, 역기전력 해소 pass OFF 1 : 50V 승압회로 ON, 역기전력 해소 pass OFF 2 : 50V 승압회로 OFF, 역기전력 해소 pass ON others : 50V 승압회로 OFF, 역기전력 해소 pass OFF
	return value	Xodddd1;
;Xo?; (#^)	comment	RJM_VER5 스텝모터 H/W 버전에서만 설정이 가능함.

명령어	구분	설명
(고전압 제어기인 경우) ;Xodddd1; (#^) ;Xo?; (#^)	operation	<p>380V 용 모터제어기에서 PRECHARGE_RELAY, MAIN_RELAY, BRAKE_RELAY를 강제 설정함. IRAM136을 사용한 제어기의 경우</p> <p>1st word : 1 : PRECHARGE_RELAY on, MAIN_RELAY off, BRAKE_RELAY off 2 : PRECHARGE_RELAY off, MAIN_RELAY on, BRAKE_RELAY off 3 : PRECHARGE_RELAY off, MAIN_RELAY off, BRAKE_RELAY on others : PRECHARGE_RELAY off, MAIN_RELAY off, BRAKE_RELAY off</p> <p>IXYS사의 CIB 모듈을 사용한 경우</p> <p>1st word : 2 : MAIN_RELAY on, 나머지 off 3 : BRAKE_RELAY on, 나머지 off 4 : Phase_BRAKE_RELAY on, 나머지 off 5 : Cool_FAN on, 나머지 off others : MAIN_RELAY off, BRAKE_RELAY off, Phase_BRAKE_RELAY off, Cool_FAN off</p>
	return value	<p>Xodddd1; IRAM136을 사용한 제어기의 경우</p> <p>1st word : 0 : PRECHARGE_RELAY off, MAIN_RELAY off, BRAKE_RELAY off 1 : PRECHARGE_RELAY on, MAIN_RELAY xxx, BRAKE_RELAY xxx 2 : PRECHARGE_RELAY off, MAIN_RELAY on, BRAKE_RELAY xxx 3 : PRECHARGE_RELAY off, MAIN_RELAY off, BRAKE_RELAY on 4 : 기타의 on/off 조합인 경우</p> <p>IXYS사의 CIB 모듈을 사용한 경우</p> <p>1st word : 0 : MAIN_RELAY off, BRAKE_RELAY off, Phase_BRAKE_RELAY off, Cool_FAN off 2 : MAIN_RELAY on, BRAKE_RELAY xxx, Phase_BRAKE_RELAY xxx, Cool_FAN xxx 3 : MAIN_RELAY off, BRAKE_RELAY on, Phase_BRAKE_RELAY xxx, Cool_FAN xxx 4 : MAIN_RELAY off, BRAKE_RELAY off, Phase_BRAKE_RELAY on, Cool_FAN xxx 5 : MAIN_RELAY off, BRAKE_RELAY off, Phase_BRAKE_RELAY off, Cool_FAN on 6 : 기타의 on/off 조합인 경우</p>
	comment	380V 용 BLDC 모터제어기에서만 설정이 가능함.

명령어	구분	설명
;Xedddd1,dddd2; (+#^)	operation	스텝핑모터에서 DC-DC converter의 승압전압의 크기와 turn-off에 소요되는 시간을 설정함. 1st word : DC-DC converter의 승압전압의 크기 [24 - 60] (48 V) 2nd word : DC-DC converter의 off time [1-50] (5 10ms)
	return value	Xedddd1,dddd2;
	comment	2012년 4월 10일 S/W 버전부터는 DC-DC converter의 승압전압 크기 기본값이 49V로 바뀌었음. ⚙ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;XEdddd1,dddd2; or (+#^) ;XEIIII1,IIII2; (+#^) ;XE?; (+#^)	operation	스텝핑모터 또는 절대각엔코더를 사용하는 경우 엔코더의 4체배된 펄스수를 설정함. 1st word : 모터1의 1회전당 펄스수 [1-65535] or [1-2147483647] 2nd word : 모터2의 1회전당 펄스수 [1-65535] or [1-2147483647]
	return value	XEdddd1,dddd2; or XEIIII1,IIII2;
	comment	28334/5 H/W 버전에서는 long 변수가 할당되기 때문에 1-2147483647 범위의 설정이 가능함. 2810/2811 H/W 버전에서는 16비트 변수가 할당되기 때문에 1-65535 범위의 설정이 가능함. LOW_RESOLUTION_ENCODER를 사용하도록 하는 S/W 버전에서는 부팅시에 XE 명령에 의한 파라미터값을 SE 명령에 의한 파라미터값으로 강제 설정함. 2013년 6월 08일 S/W 버전부터는 28334/5 H/W 버전에서는 음수값도 설정이 가능함. ⚙ 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;XPXAAAAA.....A AAA; (+#^)	operation	시퀀스제어 프로그램의 현재 포인터 위치에 시퀀스제어 프로그램을 입력함. AAAAA.....AAAA : 현재 포인터 위치에 저장할 시퀀스제어 프로그램 한 스텝.
	return value	XPXAAAAA.....AAAA;
	comment	SX 로 설정되는 값의 최상위 비트가 0 일 때만 입력이 가능함. 포인터의 위치는 입력된 시퀀스제어 프로그램 바로 다음으로 변경됨. 시퀀스제어 프로그램은 최대 640 워드, 1280자 까지 입력이 가능함. 2014년 2월 13일 S/W 버전 부터는 고전압 2채널 버전의 H/W에서는 프로그램 메모리 부족으로 시퀀스제어 기능을 삭제하였음. 설정된 프로그램은 XPsA55A; 명령어로 EEPROM에 저장될 수 있음.
;XP?; (+#^)	operation	시퀀스제어 프로그램의 현재 포인터 위치에 있는 시퀀스제어 프로그램 한 스텝을 보여주고, 포인터 위치는 변경하지 않음.
	return value	XPAAAAA.....AAAA; AAAAA.....AAAA : 현재 포인터 위치에 있는 시퀀스제어 프로그램 한 스텝.
	comment	시퀀스제어 프로그램 한 스텝이란 마지막이 ; 로 끝나는 범위 까지를 한 스텝으로 봄. 포인터의 값은 변하지 않음.
;XP+; (+#^)	operation	시퀀스제어 프로그램의 현재 포인터 위치에 있는 시퀀스제어 프로그램 한 스텝을 보여주고, 포인터 위치를 다음 스텝위치로 이동함.
	return value	XPAAAAA.....AAAA; AAAAA.....AAAA : 현재 포인터 위치에 있는 시퀀스제어 프로그램 한 스텝.
	comment	포인터의 값은 다음 스텝의 시퀀스제어 프로그램 위치로 이동함.
;XPddd1; (+#^)	operation	시퀀스제어 프로그램의 포인터 위치를 강제 설정하고, 설정된 포인터 위치의 시퀀스제어 프로그램 한 스텝을 보여주고, 포인터 위치는 변경하지 않음. ddd1 : 시퀀스제어 프로그램 영역의 글자 위치 [0 - 1278] (0 th)
	return value	XPAAAAA.....AAAA; AAAAA.....AAAA : dddd1 번째 글자부터 시작하는 시퀀스제어 프로그램 한 스텝.
	comment	ddd1은 짝수만 의미가 있음.

명령어	구분	설명
;XPP; (+#^)	operation	현재의 시퀀스제어 프로그램 포인터 위치를 보여줌.
	return value	XPpdddd1; 1st word : X_character_pointer
	comment	시퀀스제어 프로그램의 입력이나 보기에 사용하는 포인터로서 한 글자가 1의 값에 해당하며 항상 짝수임.
;XPsA55A; (+#^)	operation	메모리상의 시퀀스제어 프로그램을 EEPROM으로 저장함.
	return value	XPsA55A;
	comment	총 640 word의 사용자 프로그램 저장영역에 저장되며 저장위치는 TMS320F2810/11/12에서는 0x3f8700 ~ 0x3f897f 이며 TMS320F28334/5에서는 0xE700 ~ 0xE97f 임.
;XPt; (+#^)	operation	현재의 X_timer0, X_timer0_period, X_counter0의 값을 보여줌.
	return value	XPtHLHL,HLHL,HLHL; 1st word : X_timer0 2nd word : X_timer0_period 3rd word : X_counter0
	comment	타이머는 단위값이 10ms 임.
;XPP; (+#^)	operation	현재와 직전의 시퀀스제어 프로세스 카운트를 보여줌.
	return value	XPPHLHL,HLHL; 1st word : X_process_cnt 2nd word : X_process_cnt_old
	comment	시퀀스제어 프로세스 카운트는 어디까지 작동이 진행되었는지와 직전의 작동상태를 보여주는 값임. 직전의 시퀀스제어 프로세스 카운트는 한스텝씩 프로그램을 실행시키는 모드에서만 유효함.

명령어	구분	설명
;XPT; (+#^)	operation	현재의 PORT_STATUS_FOR_X_CONTROL와 X_PORT_STATUS를 보여줌.
	return value	<p>XPTHHLHL,HLHL; (2013년 8월 28일 이전 S/W 버전인 경우나 자동화버전/저가형버전이 아닌 경우) XPTHHLHLHL,HLHLHLHL; (저가형 1채널 버전인 경우) XPTHHLHLHLHL,HLHLHLHL; (자동화버전으로서 2013년 8월 28일 S/W 버전부터인 경우)</p> <p>bit0 : LS1A 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit1 : LS1B 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit2 : LS1C 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit3 : LS2A 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit4 : LS2B 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit5 : LS2C 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit6 : MOT_EN_N_port 0 이면 모터가 enable 된 상태이고 1이면 disable된 상태임. bit7 : OFF12V_port 0 이면 charge pump가 작동중이며 1이면 off된 상태임. bit8 : DISABLE1_IN_port 0 이면 L62070이 정상이며 1이면 fault 상태임. bit9 : DISABLE2_IN_port 0 이면 L62070이 정상이며 1이면 fault 상태임. bit10 : LS1D_port 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit11 : LS2D_port 0 이면 GND 단자와 short 상태이고 1이면 open 상태임. bit12 : EXT_FAULT1_port 0 이면 출력이 OFF 된 상태이며 1이면 ON 상태임. bit13 : EXT_FAULT2_port 0 이면 출력이 OFF 된 상태이며 1이면 ON 상태임.</p> <p>자동화버전으로서 2013년 8월 28일 S/W 버전부터인 경우</p> <p>bit16 : DATA_IN 의 bit 0 0 이면 절연 입력 bit0가 OFF 상태임 (optocoupler가 off 된 상태). bit17 : DATA_IN 의 bit 1 0 이면 절연 입력 bit1가 OFF 상태임 (optocoupler가 off 된 상태). bit18 : DATA_IN 의 bit 2 0 이면 절연 입력 bit2가 OFF 상태임 (optocoupler가 off 된 상태). bit19 : DATA_IN 의 bit 3 0 이면 절연 입력 bit3가 OFF 상태임 (optocoupler가 off 된 상태).</p>

명령어	구분	설명
		<p>자동화버전으로서 2013년 8월 28일 S/W 버전부터인 경우</p> <p>bit20 : DATA_IN 의 bit 4 0 이면 절연 입력 bit4가 OFF 상태임 (optocoupler가 off 된 상태). bit21 : DATA_IN 의 bit 5 0 이면 절연 입력 bit5가 OFF 상태임 (optocoupler가 off 된 상태). bit22 : DATA_IN 의 bit 6 0 이면 절연 입력 bit6가 OFF 상태임 (optocoupler가 off 된 상태). bit23 : DATA_IN 의 bit 7 0 이면 절연 입력 bit7가 OFF 상태임 (optocoupler가 off 된 상태). bit24 : DATA_OUT 의 bit 0 0 이면 절연 출력 bit0가 OFF 상태임 (optocoupler가 off 된 상태). bit25 : DATA_OUT 의 bit 1 0 이면 절연 출력 bit1 OFF 상태임 (optocoupler가 off 된 상태). bit26 : DATA_OUT 의 bit 2 0 이면 절연 출력 bit2가 OFF 상태임 (optocoupler가 off 된 상태). bit27 : DATA_OUT 의 bit 3 0 이면 절연 출력 bit3가 OFF 상태임 (optocoupler가 off 된 상태). bit28 : DATA_OUT 의 bit 4 0 이면 절연 출력 bit4가 OFF 상태임 (optocoupler가 off 된 상태). bit29 : DATA_OUT 의 bit 5 0 이면 절연 출력 bit5가 OFF 상태임 (optocoupler가 off 된 상태). bit30 : DATA_OUT 의 bit 6 0 이면 절연 출력 bit6가 OFF 상태임 (optocoupler가 off 된 상태). bit31 : DATA_OUT 의 bit 7 0 이면 절연 출력 bit7가 OFF 상태임 (optocoupler가 off 된 상태).</p> <p>2013년 8월 28일자 S/W 버전 부터는 자동화 버전인 경우에 XPTHHLHLHL,HLHLHLHL; 로 바뀌었으며, 상위 16 비트중에서 최상위 8비트는 절연 출력의 ON(1)/OFF(0) 상태를, 그 아래 8비트는 절연 입력의 ON(1)/OFF(0) 상태를 나타냄. 자동화버전으로서 2013년 8월 28일 이전 S/W 버전으로 시퀀스제어 프로그램을 작성한 후에, firmware를 upgrade 하였다면 시퀀스제어 프로그램에서 PORT_STATUS_FOR_X_CONTROL와 X_PORT_STATUS 를 사용하는 부분은 8자리 16진수로 프로그램을 바꾸어야함.</p>
	return value	
	comment	X_PORT_STATUS은 시퀀스제어 프로그램으로 조작한 PORT_STATUS_FOR_X_CONTROL 값에 해당함.

명령어	구분	설명
;XPF; (+#^)	operation	현재의 X_STATUS_FLAG를 보여줌.
	return value	XPFHLHL; 1st word : X_STATUS_FLAG
	comment	시퀀스제어 프로그램으로 조작한 X_STATUS_FLAG 값을 보여줌.
;XPS; (+#^)	operation	한스텝씩 시퀀스제어 프로그램을 실행시키는 모드일 때 한 스텝 씩 시퀀스제어 프로그램을 실행하게 함.
	return value	XPSHLHL,HLHL; 1st word : X_process_cnt 2nd word : X_process_cnt_old
	comment	SX 명령어로 설정되는 작동모드 스위치의 MSB-1 비트가 1 일 때 한 스텝씩 프로그램을 실행시키는 모드임. 여기서 한 스텝이란 시퀀스제어 프로세스 카운트값이 변화되는 구간을 한 스텝이라고 정의함.

명령어	구분	설명
;XTAAAAA.....AA AA; (+#^)	operation	text의 현재 포인터 위치에 text를 저장하며 저장된 text를 보여줌.
	return value	XTAAAAA.....AAAA; AAAAA.....AAAA : 현재 포인터 위치에 저장할 text 한 스텝.
	comment	총 240 character가 저장 가능함. ! 글자와 한글은 사용하면 안됨. (SPACE"#\$%&'()*~!~ 및 숫자, 대소문자 가능함) ?,+,p,s 는 첫글자로 사용하면 안됨. 설정된 text은 XTsA55A; 명령어로 EEPROM에 저장될 수 있음.
;XT?dddd1; (+#^)	operation	text의 포인터 위치를 강제 설정하고 해당 위치에 있는 text 한 줄을 보여주고, 포인터 위치는 강제 설정된 값을 유지함. dddd1 : text 영역의 글자 위치 [0 - 222(쓰기할때)/254(읽기할때)] (0 th)
	return value	XTdddd1,AAAAA.....AAAA; AAAAA.....AAAA : dddd1 번째 글자부터의 text 한 스텝.
	comment	dddd1은 짝수만 의미가 있음. 한 줄이란 마지막이 ; 로 끝나는 범위 까지를 한 줄로 봄.
;XT?; (+#^)	operation	text의 현재 포인터 위치에 있는 text 한 줄을 보여주고, 포인터 위치는 변경하지 않음.
	return value	XTAAAAA.....AAAA; AAAAA.....AAAA : text의 현재 포인터 위치에 있는 text 한 스텝.
	comment	포인터의 값은 변하지 않음.
;XT+; (+#^)	operation	text의 현재 포인터 위치에 있는 text 한 줄을 보여주고, 포인터 위치를 다음 줄 위치로 이동함.
	return value	XTAAAAA.....AAAA; AAAAA.....AAAA : text의 현재 포인터 위치에 있는 text 한 스텝.
	comment	포인터의 값은 다음 줄의 text 위치로 이동함.
;XTp; (+#^)	operation	현재의 text 포인터 위치값을 보여줌.
	return value	XTpdddd1; dddd1 : 현재의 text 포인터 위치.
	comment	text의 입력이나 보기에 사용하는 포인터로서 한 글자가 1의 값에 해당하며 항상 짝수임.

명령어	구분	설명
;XTsA55A; (+#^)	operation	메모리상의 text 를 EEPROM으로 저장함.
	return value	XTsA55A;
	comment	총 112 word의 TEXT 저장영역에 총 224글자가 저장되며 저장위치는 TMS320F2810/11/12에서는 0x3f8980 ~ 0x3f89ef 이며 TMS320F28334/5에서는 0xE980 ~ 0xE9ef 임.
;Xpdddd1,dddd2; (+#^)	operation	변위값이 일정량 증가 또는 감소할 때 fault 출력 단자를 통하여 일정기간의 low 펄스를 내보내는 기능에서 변위값의 증분을 설정함. dddd1 : 모터1의 변위 증분값 [PA 명령의 단위와 같음] (0). dddd2 : 모터2의 변위 증분값 [PA 명령의 단위와 같음] (0).
	return value	Xpdddd1,dddd2;
	comment	ZC로 설정된 값에서 bit2/3가 0/0 (기본값은 1/1) 이면 모터1/모터2에 해당하는 low pulse출력을 발생시키지 않으며, 증분값이 0이어도 low pulse출력을 발생시키지 않음. 증분값은 5 이하를 설정할 수 없음. 모터1/2의 펄스출력은 fault1/2 출력 단자를 통하여 출력됨. 2014년 10월 03일 S/W 버전부터 사용가능한 명령임.
;Xqdddd1,dddd2; (+#^)	operation	변위값이 일정량 증가 또는 감소할 때 fault 출력 단자를 통하여 일정기간의 low 펄스를 내보내는 기능에서 low 펄스의 기간을 설정함. dddd1 : fault1 출력 단자를 통한 low 펄스의 지속기간 [1us] (100 us). dddd2 : fault2 출력 단자를 통한 low 펄스의 지속기간 [1us] (100 us).
	return value	Xqdddd1,dddd2;
	comment	low 펄스의 기간은 기본값이 100us 이며, 너무 작은 값은 설정할 수 없으며, 리턴값은 실제 적용 가능한 펄스폭을 의미하므로 Xq 명령으로 설정한 값과 조금 다를 수 있음. 펄스를 내보내는 시점의 분해능은 PWM 주파수와 같으며, 위치값을 판단하는 주파수는 전류제어 주파수와 같은데, Zp?; 명령을 사용하면 PWM 주파수와 전류제어 주파수를 읽어낼수 있음. 2014년 10월 03일 S/W 버전부터 사용가능한 명령임.
;Xr; 또는 ;Xr?;	operation	Xp 명령으로 펄스개시 출력을 개시하는 순간의 위치값을 읽어서 호스트로 전송함.
	return value	Xr 1, 2;
	comment	2014년 11월 24일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;Z0iiii1; ~ ;ZFiiii1; (단 ;Z3HLHL; ;Z4HLHL; ;Z5HLHL; ;Z6HLHL; ;Z7HLHL; ;Z8HLHL; ;Z9HLHL; ;ZAHLHL; ;ZBHLHL; ;ZCHLHL; 는 예외) (+#^)(2012년 12 월 19일 이전 버전 인 경우)	operation	0 ~ F 까지의 숫자로 지정된 위치의 FACTORY SETTING parameter RAM에 값을 저장함. iiii1 : 저장하고자하는 십진수의 값 [0 - 65535] (단 Z3A55A,HLHL; Z4A55A,HLHL; Z5A55A,HLHL; Z6A55A,HLHL; Z7A55A,HLHL; Z8A55A,HLHL; Z9A55A,HLHL; ZAA55A,HLHL; ZBA55A,HLHL; ZCA55A,HLHL; 는 16진수 의 값임, A55A는 2012년 12월 19일 S/W 버전부터 적용됨.)
	return value	Z0(~F)HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; HLHL,HLHL,....,HLHL : 16개의 FACTORY SETTING parameter 값
;Z0A55A,iiii1; ~ ;ZFA55A,iiii1; (단 ;Z3A55A,HLHL; ;Z4A55A,HLHL; ;Z5A55A,HLHL; ;Z6A55A,HLHL; ;Z7A55A,HLHL; ;Z8A55A,HLHL; ;Z9A55A,HLHL; ;ZAA55A,HLHL; ;ZBA55A,HLHL; ;ZCA55A,HLHL; 는 예외) (+#^)(2012 년 12월 19일 S/W 버전부터)	comment	<p>⚡ Z0 로 설정되는 값은 전압측정의 offset calibration 값임. 예를 들어서 30V의 전압이 Sv 명령어로 2990로 읽혀졌다면 Z0-10;으로 offset 값을 설정하고 ZsA55A; 명령어로 저장하면 됨. 이 값은 -400~+400 까지만 유효하며 기본값은 0임.</p> <p>⚡ Z1으로 설정되는 값은 제일메디컬인 경우 속도명령값을 강제 수정하는 경계전압값임. 예를 들어서 31V의 전압이하에서 속도명령값을 강제로 줄이고자 한다면 Z13100;으로 경계전압값을 설정하고 ZsA55A; 명 령어로 저장하면 됨. 이 값은 2700~3300 까지만 유효하며 기본값은 3000임.</p> <p>Z1으로 설정되는 값은 제일메디컬이 아닌 경우 AD 변환기의 scale factor이며, 16384 값일 때 실질적인 scale factor는 1로 적용됨. 14746 ~ 18022 범위의 값만 유효하며, 이 범위를 벗어나면 16384로 처리 됨. 이러한 AD 변환기의 scale factor 조정은 TMS320F28334/28335를 사용한 H/W 버전에서는 실시하지 않음. 제일메디컬의 경우는 16384 값이 적용됨.</p> <p>⚡ Z2로 설정되는 값은 제일메디컬인 경우 속도명령값을 강제 수정하는 수정비율임. 예를 들어서 1V의 전압이 강하될 때마다 1600 rpm을 감소시킨다면 Z216;으로 수정비율을 설정하고 ZsA55A; 명령어로 저장하 면 됨. 이 값은 10~22 까지만 유효하며 기본값은 16임.</p> <p>⚡ Z3로 설정되는 값은 모터제어기에 설치된 전류감지 저항값임. 예를 들어서 2mΩ의 총/상전류감지저 항이 모터1에, 10mΩ의 총/상전류감지저항이 모터2에 설치되어 있다면 Z30A02; 명령어로 설정하고 ZsA55A; 명령어로 저장하면 됨. 저항값은 1~20, 21~50이 유효한 값이며, 이 범위를 벗어난 경우에는 2 의 값으 로 처리됨. 값이 1~20 범위 일때는 mΩ의 단위이며, 21~50 범위 일때는 0.01mΩ 단위임. 이 값은 임의 로 바꾸면 안됨. 2012년 11월 30일 S/W 버전부터 상전류 측정에 IR21771을 사용하면서 총전류 측정에 AD8205를 사용하는 1채널 제어기 (IRF7779를 사용하는 72V 제어기와 DC 전압을 사용하는 72V 자동화용 제 어기) 에서는 상위 8비트로서 총전류 측정용 저항값의 크기를 설정하도록 하여야함. 2013년 2월 18일 S/W 버전부터는 모터1개만 제어하는 H/W 버전의 경우 총전류저항값을 의미함. (총전류저항이 없으면 00)</p>

명령어	구분	설명
<p>2012년 12월 19일 S/W 버전부터 2013년 2월 11일 S/W 버전까지는 Z0, Z1에 한하여 ;Z0A55Aiiii1; ;Z1A55Aiiii1; 형식만 허용됨 (, 생략됨)</p>	comment	<p>⚡ Z4로 설정된 값은 하드웨어 버전 정보를 나타냄. ⚡ Z5로 설정된 값은 serial number임. ⚡ Z6로 설정된 값의 least significant 4bit는 PWM 출력의 deadband 크기를 설정하는 것으로서, 예를 들어서 Z60002; 명령을 수행하면 PWM deadband 크기가 $0.1333 \times 2 \mu\text{s}$ (IGBT를 사용하는 고전압 H/W 버전의 경우는 $0.2666 \times 2 \mu\text{s}$) 가 되도록 설정하는 것이며, ZsA55A;로 저장한다음 전원 재투입 이후부터 반영됨. 저장시에 그 값이 F 이면 2로 대치하여 저장함. 톰슨사 또는 TI사의 intelligent power IC를 사용한 경우는 0, IR사의 모듈 MOSFET를 사용한 경우는 4, IGBT를 사용하는 경우는 4 이상, 기타의 경우에 기본값은 2임. ⚡ Z6의 상위 12비트는 사용가능한 최대전압값(단위는 V)을 의미함. 24V 모터의 경우는 30(01E), 36V 모터의 경우는 45(02D), 48V 모터의 경우는 60(03C)임. 12(00C)보다 작거나 500(1F4) 보다 크다면 기본값 30(01E)으로 설정됨. MOSFET의 gate를 off하는 전압강하량의 설정값(Sv 명령의 첫번째값)은 사용 가능 최대전압의 1/2 보다 크지 않게 제한됨. ⚡ Z7으로 설정되는 값은 모터구동 최대전류값 (단위는 A)를 의미함. 예를 들어서 모터1은 최대 14A, 모터2는 최대 8A 라면 Z7080E; 명령어로 설정하고 ZsA55A; 명령어로 저장하면 됨. 제품 출하시에 Z6/Z7 으로 저장된 값은 사용자가 임의로 바꾸면 안됨. ⚡ Z8으로 설정되는 하위바이트는 모터제어용 PWM 주파수를 설정하는 값으로서 10KHz의 배수를 의미함. 예를 들어서 100KHz의 PWM을 사용한다면 0A 값이어야 하며, 20KHz라면 02 값이어야 함. 02에서 0A의 범위값만 허용함. RJM_VER7으로서 digital Hall 센서만을 사용하는 경우에는 80KHz 까지만 설정이 가능함. 제어로직을 수행하는데 CPU time이 부족하면 부팅시에 PWM 주파수를 낮추어서 적용하므로 실제로 적용되는 PWM 주파수가 얼마인지를 확인하려면 SE 명령의 리턴값으로서 제일 마지막값을 보면 됨. 예를 들어서 1770 이 리턴되었다면 $120000 / 6000 = 20\text{KHz}$ 인데 6000은 1770의 십진수 값임. 2013년 4월 25일 이전 버전에서는 Z8 명령어로 설정된 PWM 주파수가 60KHz를 넘는 경우, 03번 작동모드로 설정이 안되는 경우가 있는데, 이때는 Z8 명령어로 PWM 주파수를 60KHz 와 같거나 작게 설정하면 됨. ⚡ Z8 명령어로 모터제어용 PWM 주파수를 설정한 값은 전원을 재투입하여야 반영됨. 또 재부팅시에 모터타입이나 센서타입에 따라서 추가로 PWM 주파수가 제한되므로 실제 적용되는 PWM 주파수를 알려면, SE?; 명령어에 대한 세번째 리턴값을 10진수로 바꾼다음 120000(통상값) 또는 90000(TMS320F28069를 사용하는 저가형 버전) 또는 60000(TMS320F28035를 사용하는 RC 버전) 을 나누면 KHz 단위의 PWM 주파수임.</p>

명령어	구분	설명
	comment	<p>⚡ Z8 명령으로 모터제어용 PWM 주파수를 설정할때 : PID제어가 아니고 AU/Au/AV/Av/Aa/SX 명령어로 설정하는 고급제어를 사용하는 경우로서 PWM 주파수를 바꾸고 저장하는 경우에는 재부트하여 제어파라미터도 다시 설정하여야함. 또 고급제어를 사용하는 경우에, 계산시간이 PWM 주파수가 높아서 PWM 신호 한 주기내에 모든 계산을 처리하지 못하는 경우, PWM 주파수를 낮추어야하며, 2~6범위의 짝수값으로 설정하여야함. (모니터링프로그램과 Qx20; 을 사용하여 CPU 사용률을 확인할수 있음)</p> <p>⚡ Z8으로 설정되는 상위바이트는 모터제어기의 발열량계산에서 모터 stall시의 전류 ripple에 의하여 증가되는 발열량의 배율을 설정함. 예를 들어서 0(00)/16(10)/32(20)/48(30)/64(40)이면 ripple이 없을때의 발열량의 1/2/3/4/5배가 모터 stall시에 발열한다고 설정하는 것임. (2010년 9월 22일 S/W 버전부터 적용) (2014년 7월 21일 S/W 버전부터는 07번 전류제어모드에서도 적용됨)</p> <p>⚡ Z9로 설정되는 최상위 4bit는 모터제어기의 발열이 허용치를 넘는 허용시간(sec)를 가리키며, 나머지 12bit는 방열판에서 허용하는 연속 발열량(W)을 가리킴. 기본 방열판의 허용 연속 발열량은 초소형의 경우(25W급 제어기) 3W, 소출력형의 경우(50W급 제어기) 6W, 중출력형의 경우(100W급 제어기) 9W, 고출력형의 경우(200W급 제어기) 12W 임. (2010년 9월 22일 S/W 버전부터 적용)</p> <p>⚡ ZA로 설정된 값의 하위 8비트는 brake 저항의 저항값 (0hm)을 의미하며 0x00이나 0xFF이면 설치되어 있지 않음을 의미함.</p> <p>⚡ ZA로 설정된 값의 상위 8비트는 제어기에 공급되는 전압의 크기를 측정하는 분할저항의 저항값 (K0hm)을 의미하며, 공급전압이 48V 보다 낮은 경우는 통상 20 이지만, 48V인 경우에는 30 K0hm을 사용하여야하며 H/W 에도 30K0hm이 조립되어야함. 공급전압이 48V보다 높은 제어기의 경우는 50K0hm(100V 제어기) 또는 200K0hm(380V 제어기)이 사용되지만, 20 으로 설정하여도 기본값이 50 또는 200의 값이 적용됨. 또 공급전압이 48V보다 높은 제어기의 경우는 제어용 전원을 외부에서 공급받거나 내부의 전원회로를 사용하여 제어전원을 만드는데, 이때 전압분할에는 H/W 상에서 반드시 20K0hm을 사용하여야함. 스테핑모터의 경우 2가지 전압을 감지하는 회로가 적용되는데 분할저항값을 똑같이 조립하여야함.</p>

명령어	구분	설명
	comment	<p>⚡ ZB로 설정된 값의 하위 8비트가 0xE1 또는 0xB4이면 상위로부터의 펄스 입력에 따라서 위치명령값이 설정되는 모드로서 0xE1 일때는 01번/02번 위치제어모드를 사용하며, 0xB4 일때는 03번 위치제어모드를 사용함. 0xD2이면 상위로부터의 analog 입력에 따라서 속도명령값이 설정되는 모드(제어모드는 05)이며, 0xA5이면 상위로부터의 analog 입력에 따라서 속도명령값이 설정되는 모드(제어모드는 0A)이며, 0xC3이면 상위로부터의 analog 입력에 따라서 전류명령값이 설정되는 모드이며, 그 외는 RS232C/RS485/CAN/USB/LAN 등의 통신에 의한 명령으로 작동되는 모드임. (이 기능은 자동화용 H/W 버전으로서 2012년 3월 15일 S/W 버전부터 적용됨, 0xA5 인 경우는 2012년 12월 30일 S/W 버전부터 적용됨. 그러나 2013년 11월 02일 S/W 버전 부터는 모든 제어기에 적용됨)</p> <p>⚡ ZB로 설정된 값에서 하위 8비트의 바로위 4비트 값이 A0이면 servo ON 입력 또는 RB 명령어로 부팅할때 08번 모드로 가장 가까운 초기위치를 설정하고 부팅하며, ZB로 설정된 값에서 하위 8비트의 바로위 4비트 값이 00이면 servo ON 입력 또는 RB 명령어로 부팅할때 08번 모드로 -방향 끝에 가장 가까운 초기위치를 설정하고 부팅함. 또 PE/SM 명령으로만 부팅하는 경우는 특정위치에서 해야함.</p> <p>⚡ ZC로 설정된 값에서 하위 비트(bit0)의 값이 00이면, 모터제어기 외부입력 터미널중에서 FAULT1 상태를 출력하는 단자에 5V 전원 출력이 연결되어 있음을 의미하며 이 경우에 FAULT1 출력 기능을 사용할수 없음. 또 ZC로 설정된 값에서 bit1의 값이 00이면, 모터제어기 외부입력 터미널중에서 FAULT2 상태를 출력하는 단자에 5V 전원 출력이 연결되어 있음을 의미하며 이 경우에 FAULT2 출력 기능을 사용할수 없음. 모터제어기 출하시에 ZC로 설정된 값에서 bit0/bit1 값이 0으로 설정되어 있다면 절대로 바꾸지 말것!!!.</p> <p>⚡ ZC로 설정된 값에서 bit2의 값이 00이면, 모터제어기 외부입력 터미널중에서 FAULT1 상태를 출력하는 단자를 모터1의 fault 상태를 출력하는 용도로 사용하지 않고 X0 명령에 의하여 on/off 출력을 내보내는 용도로 사용하게하며, bit0가 00이면 bit0의 기능이 우선함. 또 bit3의 값이 00이면, 모터제어기 외부입력 터미널중에서 FAULT2 상태를 출력하는 단자를 모터2의 fault 상태를 출력하는 용도로 사용하지 않고 X0 명령에 의하여 on/off 출력을 내보내는 용도로 사용하게하며, bit1가 00이면 bit1의 기능이 우선함. (ZC로 설정된 값에서 bit0/1/2/3 의 기능은 2014년 1월 27일 S/W 버전부터 사용방법이 확립됨)</p> <p>제품 출하시에 Z3/Z4/Z5/Z6/Z7/Z8/Z9/ZA/ZB/ZC 으로 저장된 값은 사용자가 임의로 바꾸면 안됨. 설정된 값은 ZsA55A; 명령어로 EEPROM에 저장될 수 있음. (주의! EsA55A 아님)</p>

Zx 명령에 의한 factory setting 의 설정

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335				
	Z0 (십진수)	Z1 (십진수)	Z2 (십진수)	Z3 (16진수)
기능	전압측정의 offset calibration 값	속도명령값을 강제 수정하는 경계전압값 (제일메디컬인 경우만 해당됨)	속도명령값을 강제 수정하는 수 정비율 (제일메디컬인 경우만 해당됨)	모터제어기에 설치된 전류감지 저항값
세부 내역				<p>하위 8bit : 모터1의 상전류 감지 저항값 (단위는 mΩ 또는 값이 20보다 클 경우는 0.01mΩ)</p> <p>상위 8bit : 모터2의 상전류 감지 저항값 (단위는 mΩ 또는 값이 20보다 클 경우는 0.01mΩ), 단 2012년 11월 30일 S/W 버전부터 상전류 측정에 IR21771을 사용하면서 총전류 측정에 AD8205를 사용하는 1채널 제어기의 경우는 총전류감지 저항의 크기</p> <p>2013년 2월 18일 S/W 버전부터는 모터1개만 제어하는 H/W 버전의 경우 총전류저항값을 의미함</p>
	Z4 (16진수)	Z5 (16진수)	Z6 (16진수)	Z7 (16진수)
기능	하드웨어 버전 정보	serial number	deadband 및 사용 최대 전압값	모터구동 최대전류값 (단위는 A)
세부 내역			<p>최하위 4bit : PWM 출력의 deadband 크기 (단위 크기는 0.1333uS 또는 IGBT를 사용하는 고전압 H/W 버전의 경우 0.2666uS)</p> <p>상위 12비트 : 사용가능한 최대 전압값 (단위는 V)</p>	<p>하위 8bit : 모터1의 최대 허용 상전류값 (단위는 A)</p> <p>상위 8bit : 모터2의 최대 허용 상전류값 (단위는 A)</p>

Zx 명령에 의한 factory setting 의 설정

TMS320F2810/TMS320F2811/TMS320F2812/TMS320F28334/TMS320F28335					
	Z8 (16진수)	Z9 (16진수)	ZA (16진수)	ZB (16진수)	
기능	모터제어용 PWM 주파수와 모터 stall시 발열 파라미터	연속 발열량과 허용 시간	brake 저항값과 공급전압 측정용 전압분할 저항값	전원 투입시의 자동부팅 모드	
세부 내역	<p>하위 8bit : 모터제어용 PWM 주파수를 설정값 (PID제어가 아니고 AU/Au/AV/Av/Aa/SX 명령어로 설정하는 고급제어를 사용하는 경우로서, PWM 주파수를 바꾸고 저장하는 경우는 재부트하여 제어파라미터도 다시 설정하여야함)</p> <p>상위 8bit : 모터 stall시의 전류 ripple에 의하여 증가되는 발열량의 배율</p>	<p>하위 12bit : 방열판에서 허용하는 연속 발열량(단위는 W)</p> <p>상위 4bit : 방열판에서 발열이 허용치를 넘는 허용시간 (단위는 sec)</p>	<p>하위 8bit : brake 저항의 저항값 (단위는 Ohm)을 의미하며 0x00이나 0xFF이면 설치되어 있지 않음을 의미함.</p> <p>상위 8bit : 제어기에 공급되는 전압의 크기를 측정하는 분할저항의 저항값 (단위는 KOhm)</p>	<p>하위 8bit : 0xE1/0xB4/0xD2/0xA5/0xC3 일때 각각 (01/02번_위치제어)/03번_위치제어/05번_속도제어/0A번_속도제어/07번_전류제어 모드로 전원 투입시 자동 부팅됨. 이때 위치명령은 상위제어기로 부터의 펄스입력이 사용되며, 속도 및 전류명령은 상위제어기로 부터의 analog입력이 사용됨.</p> <p>상위 4bit : A이면 servo ON 입력 또는 RB 명령어로 부팅할때 08번 모드로 가장 가까운 초기위치를 설정하고 부팅하며, 0이면 servo ON 입력 또는 RB 명령어로 부팅할때 08번 모드로 -방향 끝에 가장 가까운 초기위치를 설정하고 부팅함.</p>	
	ZC (16진수)	ZD (십진수)		ZE (십진수)	ZF (십진수)
기능	fault 출력 단자를 통한 +5V 전원 출력설정				
세부 내역	<p>bit0 : 0이면 모터제어기 외부입력 터미널중에서 FAULT1 상태를 출력하는 단자에 5V 전원 출력이 연결되어 있음을 의미하며 이 경우에 FAULT1 출력 기능을 사용할수 없음.</p> <p>bit1 : 0이면 모터제어기 외부입력 터미널중에서 FAULT2 상태를 출력하는 단자에 5V 전원 출력이 연결되어 있음을 의미하며 이 경우에 FAULT2 출력 기능을 사용할수 없음.</p> <p>bit2 : 0이면 모터제어기 외부입력 터미널중에서 FAULT1 상태를 출력하는 단자를 FAULT1 출력 기능이 아니고 X0 명령에 의하여 on/off 출력을 내보내는 용도로 사용함.</p> <p>bit3 : 0이면 모터제어기 외부입력 터미널중에서 FAULT2 상태를 출력하는 단자를 FAULT2 출력 기능이 아니고 X0 명령에 의하여 on/off 출력을 내보내는 용도로 사용함.</p>				

FACTORY SETTING parameter의 설정

⚙ FACTORY SETTING parameter는 EEPROM의 제일마지막 16 word에 저장되며 Z0~ZF 명령어로 개발 설정할수있고, Zr 명령어로 읽어 낼수있으며, ZsA55A; 명령어로 EEPROM에 저장할수 있음.

⚙ 전압측정의 offset calibration 값의 설정 (제일메디컬의 경우)

Z0 로 설정되는 값은 전압측정의 offset calibration 값임. 예를 들어서 30V의 전압이 Sv 명령어로 2990로 읽혀졌다면 Z0-10;으로 offset 값을 설정하고 ZsA55A; 명령어로 저장하면 됨. 이 값은 -200~+200 까지만 유효하며 기본값은 0임.

⚙ 속도명령값을 강제 수정하는 경계전압값의 설정

Z1으로 설정되는 값은 제일메디컬인 경우 속도명령값을 강제 수정하는 경계전압값임. 예를 들어서 31V의 전압이하에서 속도명령값을 강제로 줄이고자 한다면 Z13100;으로 경계전압값을 설정하고 ZsA55A; 명령어로 저장하면 됨. 이 값은 2700~3300 까지만 유효하며 기본값은 3000임.

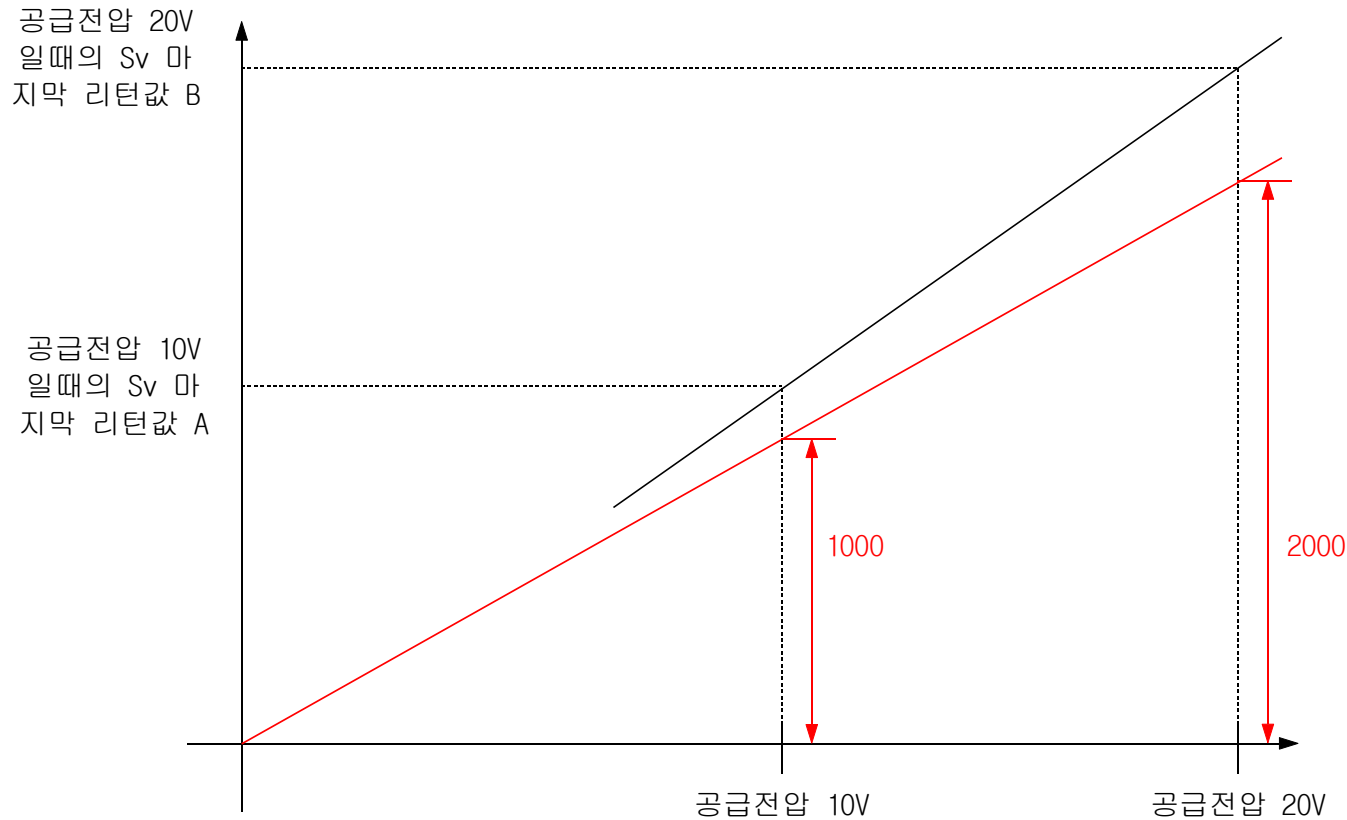
⚙ 속도명령값을 강제 수정하는 수정비율의 설정

Z2로 설정되는 값은 제일메디컬인 경우 속도명령값을 강제 수정하는 수정비율임. 예를 들어서 1V의 전압이 강하될 때마다 1600 rpm을 감소시킨다면 Z216;으로 수정비율을 설정하고 ZsA55A; 명령어로 저장하면 됨. 이 값은 10~22 까지만 유효하며 기본값은 16임.

⚙ 모터제어기에 설치된 전류감지 저항값의 설정

Z3로 설정되는 값은 모터제어기에 설치된 전류감지 저항값임. 예를 들어서 5mΩ의 전류감지저항이 모터1에, 10mΩ의 전류감지저항이 모터2에 설치되어 있다면 Z30A05; 명령어로 설정하고 ZsA55A; 명령어로 저장하면 됨. 이 값은 02,05,0A 만 유효함. 임의로 바꾸면 안됨. 같은 모터제어 채널이 있어서 총전류감지 저항과 상전류 감지 저항은 일치하여야함.

Z0, Z1 parameter의 설정



$$\begin{aligned} \left(A \times \frac{65}{100} \right) \times \frac{Z_1}{16384} - \frac{65}{100} \times Z_0 &= 1000 \times \frac{65}{100} \\ \left(B \times \frac{65}{100} \right) \times \frac{Z_1}{16384} - \frac{65}{100} \times Z_0 &= 2000 \times \frac{65}{100} \\ \Rightarrow Z_1 &= \frac{1000}{B - A} \times 16384, Z_0 = A \times \frac{Z_1}{16384} - 1000 \end{aligned}$$

- ① Z0-1; Z1-1; ZsA55A;를 실행한 후, 전원을 껐다켜고
 - ② Sv?;를 실행한 상태에서 10V 전압을 공급한 다음 (정확한 멀티미터로 제어기 전원공급단자에서 확인) Sv 마지막 리턴값 A를 읽을것
 - ③ Sv?;를 실행한 상태에서 20V 전압을 공급한 다음 (정확한 멀티미터로 제어기 전원공급단자에서 확인) Sv 마지막 리턴값 B를 읽을것
 - ④ Z1 parameter = 16384 * 1000 / (B - A)
 - ⑤ Z0 parameter = (Z1 parameter * A / 16384) - 1000
- 을 계산하여 Z0, Z1명령어를 사용하여 10진수로 입력한 다음
- ⑥ ZsA55A; 를 실행한 후, 전원을 껐다 켜고 Sv?; 명령어로 보정된 전압값 확인.

명령어	구분	설명
;Zr; (+#^)	operation	16개의 FACTORY SETTING parameter 값을 읽어서 보여줌 (RAM 영역의 값).
	return value	ZrHLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL,HLHL; HLHL,HLHL,....,HLHL : 16개의 FACTORY SETTING parameter 값
	comment	일곱번째값(Z6 명령어로 저장된 값)의 least significant 4bit의 값이 F 이면 2로 대치하여 저장함.
;ZsA55A; (+#^)	operation	메모리상의 FACTORY SETTING parameter 를 EEPROM으로 저장함.
	return value	ZsA55A;
	comment	이 명령을 사용하면 FACTORY SETTING parameter 값이 변경되므로, 사용자는 특별한 사유가 없는 한 이 명령의 사용을 금함.
;Zp?; (+#^)	operation	PWM 주파수와 모터1/2의 전류제어 주파수, 속도 및 위치제어 주파수를 읽어서 호스트로 전송함.
	return value	Zpdddd1,dddd2; (2015년 3월 23일 이전 버전인 경우) Zpdddd1,dddd2,dddd3,dddd4,dddd5,dddd6,dddd7,dddd8; (2015년 3월 23일 버전부터) dddd1 : PWM 주파수 (1KHz 단위, 2015년 3월 23일 버전부터는 0.1KHz 단위) dddd2 : 전류제어 주파수 (1KHz 단위, 2015년 3월 23일 버전부터는 0.1KHz 단위) dddd3 : 위치제어의 inner loop가 아닌 속도제어 주파수 (0.1KHz 단위) dddd4 : 위치제어의 inner loop인 모터1의 속도제어 주파수 (0.1KHz 단위) dddd5 : 위치제어의 inner loop인 모터2의 속도제어 주파수 (0.1KHz 단위) dddd6 : inner 속도제어가 없는 위치제어의 주파수 (0.1KHz 단위) dddd7 : inner 속도제어가 있는 위치제어의 주파수 (0.1KHz 단위) dddd8 : observer의 계산 주파수 (0.1KHz 단위)
	comment	PWM 주파수의 단위는 KHz 이며 (2015년 3월 23일 버전부터는 0.1KHz 단위), Z8 명령으로 설정하는 PWM 주파수 설정값 * 10 (2015년 3월 23일 버전부터는 PWM 주파수 설정값 * 100)과 다른 값일 수 있는데, 이는 CPU time의 부족으로 전원 투입시에 강제로 PWM 주기를 늘이는 경우가 있기 때문이며, 실제로 적용되는 PWM 주파수는 Zp?; 의 리턴값과 같음. 2014년 10월 03일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;ZQdddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8; (+#^)	operation	<p>모터1의 특성파라미터(저항, 인덕턴스, Kt, Kb, Rth1, Rth2, Tau1, Tau2)를 설정함.</p> <p>dddd1 : 권선저항(terminal resistance) (mOhm 단위)</p> <p>dddd2 : 권선인덕턴스(terminal inductance) (micro Henry 단위)</p> <p>dddd3 : 토크상수 (mNm/100A 단위, KOMOTEK 380V 경우는 mNm/10A)</p> <p>dddd4 : 역기전력상수 (micro Volt/RPM)</p> <p>dddd5 : 권선과 모터케이스 사이의 열전달저항 (0.001deg/W 단위)</p> <p>dddd6 : 모터케이스와 주변공기 사이의 열전달저항 (0.01deg/W 단위)</p> <p>dddd7 : 권선과 모터케이스 사이의 열전달 시정수 (0.1sec 단위)</p> <p>dddd8 : 모터케이스와 주변공기 사이의 열전달 시정수 (sec 단위)</p>
	return value	ZQdddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8;
	comment	<p>powermax 30 200W 24V : 102, 16, 1350, 1408, 79, 530, 7, 848</p> <p>4490 BS 200W 24V (Y) : 690, 220, 4186, 4384, 1350, 394, 290, 1756</p> <p>4490 B 200W 24V (△) : 237, 76, 2383, 2495, 1350, 394, 290, 1756</p> <p>3257 CR 24V: 1630, 270, 3770, 3950, 2000, 800, 170, 810</p> <p>3863 C 24V: 620, 130, 3330, 3490, 1500, 600, 330, 843</p> <p>EC45 FLAT 24V 50W : 978, 573, 3350, 3508, 4500, 425, 166, 212</p> <p>KOMOTEK 380V용의 모터경우 토크상수의 단위는 mNm/10A 이므로 주의할 것.</p> <p>설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.</p>
;ZQ?; (+#^)		

명령어	구분	설명
;ZRddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8; (+#^)	operation	<p>모터2의 특성파라미터(저항, 인덕턴스, Kt, Kb, Rth1, Rth2, Tau1, Tau2)를 설정함.</p> <p>ddd1 : 권선저항(terminal resistance) (mOhm 단위)</p> <p>ddd2 : 권선인덕턴스(terminal inductance) (micro Henry 단위)</p> <p>ddd3 : 토크상수 (mNm/100A 단위, KOMOTEK 380V 경우는 mNm/10A)</p> <p>ddd4 : 역기전력상수 (micro Volt/RPM)</p> <p>ddd5 : 권선과 모터케이스 사이의 열전달저항 (0.001deg/W 단위)</p> <p>ddd6 : 모터케이스와 주변공기 사이의 열전달저항 (0.01deg/W 단위)</p> <p>ddd7 : 권선과 모터케이스 사이의 열전달 시정수 (0.1sec 단위)</p> <p>ddd8 : 모터케이스와 주변공기 사이의 열전달 시정수 (sec 단위)</p>
	return value	ZRddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7,ddd8;
	comment	<p>powermax 30 200W 24V : 102, 16, 1350, 1408, 79, 530, 7, 848</p> <p>4490 BS 200W 24V (Y) : 690, 220, 4186, 4384, 1350, 394, 290, 1756</p> <p>4490 B 200W 24V (△) : 237, 76, 2383, 2495, 1350, 394, 290, 1756</p> <p>3257 CR 24V: 1630, 270, 3770, 3950, 2000, 800, 170, 810</p> <p>3863 C 24V: 620, 130, 3330, 3490, 1500, 600, 330, 843</p> <p>EC45 FLAT 24V 50W : 978, 573, 3350, 3508, 4500, 425, 166, 212</p> <p>KOMOTEK 380V용의 모터경우 토크상수의 단위는 mNm/10A 이므로 주의할 것.</p> <p>설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.</p>
;ZR?; (+#^)		

명령어	구분	설명
;ZTdddd1,dddd2; (+#^)	operation	모터1과 모터2의 최대허용 권선온도상승값을 설정함. dddd1 : 모터1의 최대허용 권선온도상승값 (0.1도 단위) dddd2 : 모터2의 최대허용 권선온도상승값 (0.1도 단위)
	return value	ZTdddd1,dddd2;
	comment	온도 상승값은 주변온도를 기준으로 한 값인데, 주변온도는 0도로 고정되어 있음. 02/05/07번 작동모드에서만 정상적인 작동이 이루어짐. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
;Zcdddd1,dddd2; (+#^)	operation	IRMCK201을 사용한 RJM_VER3에 있어서 모터1과 모터2의 nominal current(IRMCK201 파라미터 설정에 사용한 값)를 설정함. dddd1 : 모터1의 nominal current (0.01A 단위) dddd2 : 모터2의 nominal current (0.01A 단위)
	return value	Zcdddd1,dddd2;
	comment	nominal current는 IRMCK201의 전류 scale factor를 정할때 사용하는 파라미터로서, 모터 카타로그상 연속 정격전류(RMS 값)인데, 100배의 값을 Zc 명령어로 설정하여야함. ※ IRMCK201의 작동파라미터를 다시 설정하기 어려울때 (설정 utility 프로그램이 없을때) : QRE248,01; 의 리턴값 QRE248,01,efgh,ijkl; 을 읽어서 0xIkef 값을 뽑아내고 27886044.6661633 / (상전류감지저항*0xIkef) 값을 계산하여 이 값을 Zc 명령어로 설정하면 됨. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
(2014년 2월 13일 이전 S/W 버전) ;ZPdddd1; (+#^) ;ZP?; (+#^)	operation	역기전력 해소용 전력저항의 최대 ON time을 설정함. dddd1 : 역기전력 해소용 전력저항의 최대 ON time (10ms 단위)
	return value	ZPdddd1;
	comment	설정된 첫번째 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
(2014년 2월 13일 S/W 버전부터) ;ZPdddd1; (+#^) 또는 ;ZPdddd1,dddd2,d dddd3; (+#^) ;ZP?; (+#^)	operation	역기전력 해소용 전력저항의 최대 ON time을 설정함. 2014년 2월 13일 S/W 버전 부터는 brake 저항 작 동을 테스트하는 기능도 겸함. dddd1 : 역기전력 해소용 전력저항의 최대 ON time (10ms 단위) dddd2 : brake_resistor_test_voltage (1V 단위) dddd3 : brake_resistor_test_timer_set (1ms 단위)
	return value	ZPdddd1; or ZPdddd1,dddd2,dddd3; (2014년 2월 13일 S/W 버전 부터) dddd3 : brake_resistor_test_timer (1ms 단위, 현재값)
	comment	테스트 전압은 삼각파 형식으로 brake_resistor_test_voltage 크기만큼 전압을 올렸다 내리는 방식으로 인 가되며, Qx23; 모니터링으로 확인할수 있으며, 실제 전압이 아니고 가상적으로만 올리는 것인데, 브레이크 저항은 실제로 작동됨. 설정된 첫번째 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Zbdddd1; (+#^) ;Zb?; (+#^)	operation	CAN 통신속도를 설정함. dddd1 : 통신속도설정값 [125,250,500,1000] (1000 Kbps)
	return value	Zbdddd1;
	comment	<p>전원 투입시에 DIP 스위치가 있는 경우에는 DIP 스위치 1번이 ON 되어 있을 때, DIP 스위치가 없는 경우에는 SX 명령어로 설정되는 파라미터의 LSB가 0 일때 CAN 통신모드가 선택되며, 이 경우에 baud rate 설정 레지스터값을 바꾸어서 실제 통신속도를 바꿈. 설정값이 정당한 baud rate가 아니면 기본값 1000 Kbps 로 설정됨.</p> <p>2014년 6월 14일 이전 S/W 버전에서는 Zbdddd1; 명령어에 의하여 곧바로 CAN 통신속도가 바뀌는 것은 아니며, EsA55A; 명령어로 EEPROM에 저장한 다음, 전원을 껐다 켜야지 바뀐 CAN 통신속도가 반영됨. 2014년 6월 14일 ~ 2014년 9월 12일 S/W 버전에서는 CAN 모드로 설정되어 있는 상태에서 CAN 통신라인에 120 Ohm pull-up 저항이 연결된 경우에는 Zb 명령으로 baudrate를 설정한 것이 바로 적용되나, CAN 모드로 설정되어 있는 상태에서 120 Ohm pull-up 저항이 연결되지 않은 경우에 Zb 명령을 사용하면 제어기가 리셋됨.</p> <p>2014년 9월 13일 S/W 버전부터는 CAN 모드로 설정되어 있는 상태에서 CAN 통신라인에 120 Ohm pull-up 저항이 연결된 경우에는 Zb 명령으로 baudrate를 설정한 것이 바로 적용되며, 그 외의 경우에 Zb 명령을 사용하면 재부팅 이후에 설정이 반영됨.</p> <p>설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명															
;DZHLHL,HLHL,HLHL,HLHL; L,HLHL; (+#^) ;DZ?; (+#^)	operation	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 값의 주소를 설정함. 1st word : 첫번째 변수의 주소 2nd word : 두번째 변수의 주소 3rd word : 세번째 변수의 주소 4th word : 네번째 변수의 주소															
	return value	DZHLHL,HLHL,HLHL,HLHL;															
	comment	<p>표시될 값이 2-word 라도 첫번 word의 주소만 설정하면 됨. QRHLHL,HL; 명령어로 설정가능한 변수의 주소범위를 벗어나면 reset됨. 설정가능한 주소범위는 다음과 같음.</p> <p>1st/2nd/3rd/4th : RAM address (TMS320F810/11/12)</p> <table border="0"> <tr> <td>[0000 - 017F] : M0 RAM block (.stack)</td> <td>[0180 - 01BF] : M0 RAM block (.bss)</td> </tr> <tr> <td>[01C0 - 06FF] : M0/1 RAM block (.ebss)</td> <td>[0700 - 07FF] : M1 RAM block (asm에서 사용)</td> </tr> <tr> <td>[0880 - 783F] : peripheral memory</td> <td>[8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용)</td> </tr> <tr> <td>[E000 - FFFF] : H0 RAM block (.esystem)</td> <td>[E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨.</td> </tr> </table> <p>1st/2nd/3rd/4th : RAM address (TMS320F28334/28335)</p> <table border="0"> <tr> <td>[0000 - 004F] : M0 RAM block (stack for boot)</td> <td>[0050 - 03FF] : M0 RAM block (no use)</td> </tr> <tr> <td>[0400 - 05FF] : M1 RAM block (.stack)</td> <td>[0600 - 07FF] : M1 RAM block (.bss)</td> </tr> <tr> <td>[0880 - 793F] : peripheral memory</td> <td>[8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용)</td> </tr> <tr> <td>[C000 - DFFF] : L4/L5 RAM block (.ebss)</td> <td>[E000 - FFFF] : L6/L7 RAM block (.esystem)</td> </tr> </table> <p>사용 가능한 주소영역과 각 주소의 변수값의 의미는 S/W 버전마다 다를수 있음. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.</p>	[0000 - 017F] : M0 RAM block (.stack)	[0180 - 01BF] : M0 RAM block (.bss)	[01C0 - 06FF] : M0/1 RAM block (.ebss)	[0700 - 07FF] : M1 RAM block (asm에서 사용)	[0880 - 783F] : peripheral memory	[8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용)	[E000 - FFFF] : H0 RAM block (.esystem)	[E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨.	[0000 - 004F] : M0 RAM block (stack for boot)	[0050 - 03FF] : M0 RAM block (no use)	[0400 - 05FF] : M1 RAM block (.stack)	[0600 - 07FF] : M1 RAM block (.bss)	[0880 - 793F] : peripheral memory	[8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용)	[C000 - DFFF] : L4/L5 RAM block (.ebss)
[0000 - 017F] : M0 RAM block (.stack)	[0180 - 01BF] : M0 RAM block (.bss)																
[01C0 - 06FF] : M0/1 RAM block (.ebss)	[0700 - 07FF] : M1 RAM block (asm에서 사용)																
[0880 - 783F] : peripheral memory	[8000 - 9FFF] : L0/L1 RAM block (RAM 실행코드용)																
[E000 - FFFF] : H0 RAM block (.esystem)	[E000 - FFFF]의 주소를 설정하면 실제로는 [3F8000 ~ 3F9FFF]의 주소에 있는 값을 지칭하도록 내부에서 처리됨.																
[0000 - 004F] : M0 RAM block (stack for boot)	[0050 - 03FF] : M0 RAM block (no use)																
[0400 - 05FF] : M1 RAM block (.stack)	[0600 - 07FF] : M1 RAM block (.bss)																
[0880 - 793F] : peripheral memory	[8000 - BFFF] : L0/1/2/3 RAM block (RAM 실행코드용)																
[C000 - DFFF] : L4/L5 RAM block (.ebss)	[E000 - FFFF] : L6/L7 RAM block (.esystem)																

명령어	구분	설명
;DFHLHL,HLHL,HLHL,HLHL; L,HLHL; (+#^) ;DF?; (+#^)	operation	<p>컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 값의 type과 표시형식을 설정함. 1st word : 첫번째 변수의 type과 표시형식 2nd word : 두번째 변수의 type과 표시형식 3rd word : 세번째 변수의 type과 표시형식 4th word : 네번째 변수의 type과 표시형식</p> <p>각 word의 상위바이트가 0이면 4자리 HexASCII로 값이 표시됨. 1이면 8자리 HexASCII로 값이 표시됨. 2이면 5자리 unsigned integer로 값이 표시됨. 3이면 6자리 signed integer로 값이 표시됨. 4이면 7자리 unsigned long integer로 값이 표시됨. 5이면 8자리 signed long integer로 값이 표시됨. 6이면 9자리 double로 값이 표시됨.</p> <p>하위바이트가 0이면 표시대상 변수가 16bit 정수임. 1이면 표시대상 변수가 32bit 정수임. 2이면 표시대상 변수가 32bit 실수임.</p>
	return value	DFHLHL,HLHL,HLHL,HLHL;
	comment	<p>표시되는 값은 4x20 영문 디스플레이의 11번부터 20번 column에 표시됨. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;DSffff1,ffff2,ffff3,ffff4; (+#^)	operation	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 값의 scale factor를 설정함. 1st double : 첫번째 변수의 scale factor 2nd double : 두번째 변수의 scale factor 3rd double : 세번째 변수의 scale factor 4th double : 네번째 변수의 scale factor 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	return value	DSffff1,ffff2,ffff3,ffff4;
	comment	DZ 명령어로 설정된 값에다 DS 명령어에 의한 scale factor를 곱한 값이 디스플레이에 표시됨. 원래의 값을 그대로 표시하려면 scale factor의 값으로 1.0을 설정하면 됨. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
;DTdddd1,AAAAAAA AAA; (+#^)	operation	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시되는 값의 타이틀을 설정함. 1st word : 타이틀의 번호 [0 - 3] (2013년 2월 14일 S/W 버전부터는 외부표시모드가 2일때 0 - 7 범위이며, 0 - 3 은 페이지 1이 되고, 4 - 7 은 페이지 2가 됨) next character : 최대 10자의 타이틀 (2013년 2월 14일 S/W 버전부터는 외부표시모드가 1또는 2일때는 최대 20자)
	return value	DTdddd1,AAAAAAA; DTdddd1,AAAAAAAAAAAAAAAAAAAA; (2013년 2월 14일 S/W 버전부터로서 외부표시모드가 1또는 2일때)
	comment	타이틀은 4x20 영문 디스플레이의 1번부터 10번 column (2013년 2월 14일 S/W 버전부터는 외부표시모드가 1또는 2일때는 1번부터 20번 column)에 표시됨. 외부표시모드 1또는 2는 SX 명령어의 두번째파라미터 bit16-bit19에 의하여 설정됨. (2013년 2월 14일 S/W 버전부터 사용가능) 설정된 값은 TEXT 저장영역의 마지막 40byte/20word에 저장됨. (TMS320F2810/11/12에서는 0x3f89dc ~ 0x3f89ef 에 저장되며 TMS320F28334/5에서는 0xE9dc ~ 0xE9ef 에 저장됨) (2013년 2월 14일 S/W 버전부터는 TEXT 저장영역의 마지막 80byte/40word에 저장됨) 설정된 값은 XTsA55A; 명령어로 EEPROM에 저장될 수 있음. (주의! EsA55A; 아님)
;DT?dddd1; (+#^)		

명령어	구분	설명
;DPddd1; (+#^) ;DP?; (+#^)	operation	컴파일 테크놀로지사의 직렬통신 4x20 영문 디스플레이에 표시될 페이지를 지정함. 1st word : page 번호 [0 - 1]
	return value	DPddd1;
	comment	SX 명령어로 설정하는 외부표시모드 2번에서만 사용 가능함. 페이지 0은 DT 명령어를 사용하여 타이틀 번호 0 ~ 3 로 저장된 데이터가 표시되는 페이지이며, 페이지 1은 DT 명령어를 사용하여 타이틀 번호 4 ~ 7 로 저장된 데이터가 표시되는 페이지임. 2013년 2월 14일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;DDddd1,ddd2,d ddd3; (+#^) or ;DDddd1,ddd2,d ddd3,ddd4; (+#^) ;DD?; (+#^)	operation	<p>일정스텝으로 이동을 반복하는 데모작동의 파라미터를 설정함.</p> <p>1st word : set number_of_steps [1 - 20] (10 step)</p> <p>2nd word : demo_step_size [1 - 60000] (40000/10000 pulse, 2012년 2월 4일 S/W 버전부터는 100pulse가 기본단위임)</p> <p>3rd word : demo_delay_time [1 - 100] (20 10ms)</p> <p>4th word : demo_offset (2 pulse) - 2011년 4월 15일 S/W 버전부터만 적용됨.</p>
	return value	DDddd1,ddd2,ddd3; (2011년 4월 15일 이전 S/W 버전) or DDddd1,ddd2,ddd3,ddd4; (2011년 4월 15일 S/W 버전부터) 2012년 2월 4일 S/W 버전부터는 demo_step_size의 기본단위가 100 pulse 이므로 주의를 요함. 특히 이전에 데모작동파라미터를 설정하여 사용하고 있었다면 에 100을 곱하여 스텝크기로 설정하므로 이전에 설정된 값을 100으로 나누어서 재설정하여야함.
	comment	설정된 값은 XTsA55A; 명령어로 EEPROM에 저장될 수 있음. (주의! EsA55A; 아님)
;DEHLHL; (+#^) ;DE?; (+#^)	operation	<p>일정스텝으로 이동을 반복하는 데모작동의 작동모드를 설정함.</p> <p>1st word : MSB(bit15) : 1이면 후진시에도 스텝동작을 실시함.</p> <p>bit14 : 전진중(0)인지 후진중(1)인지를 알려주는 상태비트로서 강제 설정이 되지 않음.</p> <p>bit13 : 1이면 LSxC 포트의 입력이 0일때 한 사이클의 데모작동이 개시됨.</p> <p>bit12 : 1이면 데모작동이 개시될 때 bit7-bit0의 크기에 관계없이 저속/고속/스텝전후진 한 사이클만 수행하고 데모작동을 멈추며, 0이면 스텝전후진 한 사이클만 수행하고 데모작동을 멈춤. 단, SX 명령어로 MSB를 세트한 경우에는 데모작동을 계속함.</p> <p>bit7-bit0 : 00이 아니면 데모작동을 개시하며 이 값의 횟수만큼 반복한 후에 0으로 자동 크리어됨. 단 bit12가 1일 경우에는 저속/고속/스텝전후진 한 사이클 데모만 실행하고 멈춤. 이 값을 0이 아닌값으로 다시 설정하면 데모작동을 다시 개시함.</p>
	return value	DEHLHL; 1st word : bit14 : 전진중(0)인지 후진중(1)인지를 알려주는 상태비트임.
	comment	DE 명령어는 2012년 2월 4일 S/W 버전부터만 적용됨.

명령어	구분	설명
	comment	<p>SX 명령어로 MSB를 세트하지 않더라도 DE 명령어로 bit13 또는 bit7-bit0 를 설정하여 데모작동을 개시할 수 있음. 단 DE 명령어로 bit13 또는 bit7-bit0 를 사용한 경우는 정해진 횟수 또는 한사이클만 데모작동을 실시하고, 다시 트리거 되기를 기다리며, 데모작동이 개시될때 EXT_FAULT1_port(자동화용 H/W 버전은 isolated 출력 0)를 HIGH가 되게 출력하며, 데모작동을 멈출때 EXT_FAULT1_port(자동화용 H/W 버전은 isolated 출력 0)에 0V 가 약 100ms 출력되게 함. 이를 이용하면 두대의 모터제어기가 데모동작을 번갈아 할수있음.</p> <p>SX 명령어로 MSB를 세트할때 데모동작이 개시되며 SX 명령어로 MSB를 크리어할때까지는 데모작동을 계속 함. 이때 DE0000; 이면 스텝전진/고속후진을 반복하며, DE8000; 이면 스텝전진/스텝후진을 반복하며, DE9000; 이면 저속/고속/스텝전후진을 반복함.</p> <p>SX 명령어로 MSB가 크리어 되어 있을때는 DE0002; 이면 스텝전진/고속후진을 2번 반복하고 정지하며, DE8002; 이면 스텝전진/스텝후진을 2번 반복하고 정지하며, DE9002; 이면 저속/고속/스텝전후진을 1번하고 정지함 (이 경우 마지막숫자는 0만 아니면 1인 경우와 같은 결과를 가짐). DE2000; 이면 LS1C 입력이 low가 될때 스텝전진/고속후진을 1번하고 정지하며, DEA000; 이면 LS1C 입력이 low가 될때 스텝전진/스텝후진을 1번하고 정지하며, DEB000; 이면 LS1C 입력이 low가 될때 저속/고속/스텝전후진을 1번하고 정지함.</p> <p>SX 명령어로 MSB를 세트하고 EsA55A;로 저장을 하고난 다음, 재부팅하여 위치제어모드로 설정하여도 데모작동이 개시되지 않음. (demo_cnt가 202로 고정됨) 따라서 SX 명령어로 데모를 개시하려고 하면 위치제어모드로 설정하고난 다음 SX 명령어로 MSB를 세트하여야함. 중지하려면 SX 명령어로 MSB를 크리어하면 됨.</p> <p>두 모터제어기의 EXT_FAULT1_port(자동화용 H/W 버전은 isolated 출력 0)와 LS1C 포트를 서로 cross 연결하고, 각 모터제어기에 DEB000; 을 설정한 다음 한쪽제어기의 LS1C 포트를 버튼스위치로 low로 강제 세트하면 두 모터제어기가 번갈아서 저속/고속/스텝전후진 작동을 실시함. 단 두 제어기의 전위차가 크면 오류가 발생할수 있으므로 에러가 발생할때는 GT 명령어로 LS1C 포트의 on/off를 구별하는 threshold를 조정해볼 것.</p>

명령어	구분	설명
;Dddddd1; (+#^)	operation	포텐시오메터로 속도명령 또는 전류명령을 받아서 움직이는 모드에서 속도명령값 또는 전류명령값의 deadband 크기를 설정함. 1st word : LSxC 포트의 A/D 변환값의 deadband 크기 [1 - 500] (16 bit)
	return value	Dddddd1;
	comment	설정된 값은 XTsA55A; 명령어로 EEPROM에 저장될 수 있음. (주의! EsA55A; 아님)
;DOA55A, dddd1, dddd2; (+#^)	operation	포텐시오메터로 속도명령 또는 전류명령을 받아서 움직이는 모드에서 속도명령값 또는 전류명령값이 0일 때의 LSxC 포트 A/D 변환값을 설정함. 1st word : 모터1의 속도명령값 또는 전류명령값이 0이 되는 LS1C 포트의 A/D 변환값 [0 - 4095] (기본값은 전원투입시의 LS1C 포트의 A/D 변환값) 2nd word : 모터2의 속도명령값 또는 전류명령값이 0이 되는 LS2C 포트의 A/D 변환값 [0 - 4095] (기본값은 전원투입시의 LS2C 포트의 A/D 변환값)
	return value	D0dddd1, dddd2;
	comment	기본값은 전원투입시의 LSxC 포트 A/D 변환값임. 이 값을 강제로 바꾸고자한다면 D0dddd1, dddd2; 명령어를 사용하여 바꿀수 있음. EEPROM에 저장되지는 않음. 포텐시오메터로 속도명령 또는 전류명령을 받아서 움직이는 모드의 설정은 SX 명령어의 bit13을 참조.
;Do?; (+#^)	operation	LSxC 포트 A/D 변환값의 필터링값을 읽어서 호스트로 전송함.
	return value	D0dddd1, dddd2; 1st word : LS1C 포트 A/D 변환값을 필터링한 값 2nd word : LS2C 포트 A/D 변환값을 필터링한 값
	comment	필터는 butterworth_filter 12번과 13번을 사용함.

명령어	구분	설명
;DXHLHL,HLHLHL; (+#^) ;DX?; (+#^)	operation	ON/OFF 출력과 7-seg 표기값을 설정하고 ON/OFF 입력과 switch 입력을 읽어서 host로 전송함. 1st word : 8-bit의 ON/OFF 출력 설정값 2nd 24bit-word : 7-seg에 표시할 값 (이중에서 최상위 4-bit 1 character는 밝기 및 dot ON/OFF 값임)
	return value	DXHLHL,HLHL,HLHL,HLHL,HLHL; 1st word : 8-bit의 ON/OFF 출력값 2nd word : 7-seg의 표시값의 상위 3 character (이중에서 최상위 1 character는 밝기 및 dot ON/OFF 값임) 3rd word : 7-seg의 표시값의 하위 2 character 4th word : 8-bit의 ON/OFF 입력값 5th word : DIP 스위치 값
	comment	자동화용 모터제어기에서만 쓸수 있는 명령어임.

명령어	구분	설명
;Tsddd1,ddd2; (+#^) ;Ts?; (+#^)	operation	모터1의 step 구동을 위한 주기와 진폭을 설정하고 step 구동을 실행하도록 함. 1st word : motor1_period_of_step_response [0 - 2000] (0 ms) 2nd word : motor1_amplitude_of_step_response [0 - 30000] (0 bit)
	return value	Tsddd1,ddd2; 1st word : 8의 배수로 재설정됨.
	comment	위치/속도/전류제어모드에서 양방향으로 2nd word의 크기만큼 교대로, 반복적으로 step구동을 실시함. step구동을 중지하려면 1st word 값을 0으로 설정하면 됨. 1st word 값의 처음 1/8 동안은 구동명령이 0, 다음 1/4 동안은 구동명령이 +2nd word 값, 다음 1/4 동안은 구동명령이 0, 다음 1/4 동안은 구동명령이 -2nd word 값, 마지막 1/8 동안은 구동명령이 0임. 처음부터 과도한 크기의 2nd word 값을 설정하면 fault가 발생하거나, 제어전류의 크기가 전류리밋에 걸려서 step 응답이 왜곡되므로 작은값 부터 시도하여야함. 위치제어모드에서는 Qx93/97; 속도제어모드에서는 Qx92/96; 전류제어모드에서는 Qx91/95; 을 사용하여 1ms 단위로 응답을 관찰할수 있음. 전류제어모드에서는 Qx2; 또는 Qx55/56; 를 사용하여 1ms단위의 전류제어 응답을 볼수 있으며, Qx508/509; 와 Qx9;을 사용하여 전류제어 sample time 단위로 응답을 관찰할수도 있음. 2012년 8월 19일 S/W 버전부터 사용가능한 명령임. 2013년 4월 14일 이전 S/W 버전에서는 속도모드인 경우 Ts0,0; 명령어를 사용하여 step구동을 중지시킬때 중지후 일정속도로 회전하는 현상이 있음. 2013년 10월 09일 S/W 버전부터는 Ts 명령수행중에 Ts 명령으로 주기와 진폭을 바꾸어도 문제 없음.

명령어	구분	설명
;Ttdddd1,dddd2; (+#^) ;Tt?; (+#^)	operation	모터2의 step 구동을 위한 주기와 진폭을 설정하고 step 구동을 실행하도록 함. 1st word : motor2_period_of_step_response [0 - 2000] (0 ms) 2nd word : motor2_amplitude_of_step_response [0 - 30000] (0 bit)
	return value	Ttdddd1,dddd2; 1st word : 8의 배수로 재설정됨.
	comment	Ts 명령과 comment가 같음. 2012년 8월 19일 S/W 버전부터 사용가능한 명령임. 2013년 4월 14일 이전 S/W 버전에서는 속도모드인 경우 Tt0,0; 명령어를 사용하여 step구동을 중지시킬 때 중지후 일정속도로 회전하는 현상이 있음. 2013년 10월 09일 S/W 버전부터는 Tt 명령수행중에 Tt 명령으로 주기와 진폭을 바꾸어도 문제 없음.
;Tudddd1,dddd2; (+#^) ;Tu?; (+#^)	operation	모터1의 사인파 구동을 위한 주기와 진폭을 설정하고 사인파 구동을 실행하도록 함. 1st word : motor1_period_of_freq_response [0 - 2000] (0 ms) 2nd word : motor1_amplitude_of_freq_response [0 - 30000] (0 bit)
	return value	Tudddd1,dddd2;
	comment	위치/속도/전류제어모드에서 2nd word의 크기에 해당하는 진폭으로 사인파 구동을 실시함. 사인파 구동을 중지하려면 1st word 값을 0으로 설정하면 됨. 처음부터 과도한 크기의 2nd word 값을 설정하면 fault가 발생하거나, 제어전류의 크기가 전류리미트에 걸려 서 사인파 응답이 왜곡되므로 작은값 부터 시도하여야함. 위치제어모드에서는 Qx93/97; 속도제어모드에서는 Qx92/96; 전류제어모드에서는 Qx91/95; 을 사용하여 1ms 단위로 응답을 관찰할수 있음. 전류제어모드에서는 Qx2; 또는 Qx55/56; 를 사용하여 1ms단위의 전류제 어 응답을 볼수 있으며, Qx508/509; 와 Qx9;을 사용하여 전류제어 sample time 단위로 응답을 관찰할수도 있음. 2012년 8월 19일 S/W 버전부터 사용가능한 명령임. 2013년 4월 14일 이전 S/W 버전에서는 속도모드인 경우 Tu0,0; 명령어를 사용하여 사인파구동을 중지시킬 때 중지후 일정속도로 회전하는 현상이 있음. 2013년 10월 09일 S/W 버전부터는 Tu 명령수행중에 Tu 명령으로 주기와 진폭을 바꾸어도 문제 없음.

명령어	구분	설명
;Tvdddd1,dddd2; (+#^) ;Tv?: (+#^)	operation	모터2의 사인파 구동을 위한 주기와 진폭을 설정하고 사인파 구동을 실행하도록 함. 1st word : motor2_period_of_freq_response [0 - 2000] (0 ms) 2nd word : motor2_amplitude_of_freq_response [0 - 30000] (0 bit)
	return value	Tvdddd1,dddd2;
	comment	Tu 명령과 comment가 같음. 2012년 8월 19일 S/W 버전부터 사용가능한 명령임. 2013년 4월 14일 이전 S/W 버전에서는 속도모드인 경우 Tv0,0; 명령어를 사용하여 사인파구동을 중지시킬 때 중지후 일정속도로 회전하는 현상이 있음. 2013년 10월 09일 S/W 버전부터는 Tv 명령수행중에 Tv 명령으로 주기와 진폭을 바꾸어도 문제 없음.
;T0dddd1,dddd2; (+#^) ;T0?: (+#^)	operation	07번 전류제어모드의 Ts/Tt/Tu/Tv 명령에 의한 test 구동시에 변위의 중심위치가 한쪽으로 흐르는 것을 방지하기 위한 전류 offset을 설정함. 1st word : motor1_current_offset [-1000 ~ 1000] (0 bit) 2nd word : motor2_current_offset [-1000 ~ 1000] (0 bit)
	return value	T0dddd1,dddd2;
	comment	Ts/Tt/Tu/Tv 명령에 의한 test 구동시에서만 적용됨.

명령어	구분	설명
	operation	온도측정용 센서 파라미터를 설정함. 1st word : constant (temp_C1) 2nd word : linear coefficient (temp_C2) 3rd word : quadratic coefficient (temp_C3) 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	return value	MTffff1,ffff2,ffff3;
;MTffff1,ffff2,ffff3; (+#^) ;MT?; (+#^)	comment	<p>☞ 온도센서로는 NTC 온도센서를 사용하는 경우가 일반적이거나, 특별한 경우 linear temprature 센서 (TC1047)를 사용하는 경우가 있으므로 주의를 요함.</p> <p>☞ NTC 온도센서를 사용하는 경우 : 온도값은 Ohm 단위의 저항값 R을 사용하여 $r = \ln(R)$을 계산한 다음, $100000000.0 / \{temp_C1 + temp_C2 * r + temp_C3 * r * r * r\} - 23750$ 식을 이용하여 계산함. 계산된 온도값의 단위는 섭씨 0.01도 임.</p> <p>NTC 파라미터의 기본값은 temp_C1 = 1377.113100035604; temp_C2 = 234.029059587146; temp_C3 = 0.354431764337; 이며, 이는 -10도 55480 Ohm, 25도 10000 Ohm, 60도 2365 Ohm 인 103k5(D20k) 모델의 파라미터에 해당함.</p> <p>☞ linear temprature 센서(TC1047)를 사용하는 경우 : (온도센서 A/D 변환값 - temp_C1) * temp_C2 식을 이용하여 온도값을 계산함. 계산된 온도값의 단위는 섭씨 0.01도 임. temp_C1의 기본값은 682.5로서 0.5V에 해당하는 A/D 변환값이며, temp_C2의 기본값은 7.326으로서 1V당 100도 (0.01도 단위로는 10000)의 값을 갖도록 하는 scale factor 값임.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;Mtddd1; (+#^) ;Mt?; (+#^)	operation	온도측정용 센서를 사용하여 측정된 저항치 및 온도값을 읽어내어 host로 전송함. 1st word : 온도측정을 실시하는 지의 여부 [0 ~ 1] (0 or 1)
	return value	Mtddd1,ddd2,iiii3; 1st word : 온도측정을 실시하는 지의 여부 (0 or 1) 2nd word : NTC의 저항값 (0hm 단위) : NTC 온도센서를 사용하는 경우에 한함. 3rd integer : 온도값 (0.01도 단위)
	comment	⚡ NTC 온도센서를 사용하는 경우 : NTC의 저항값은 전원투입후부터 계산을 실시함. 온도값 계산은 TMS320F2809/TMS320F2810/11/12을 사용한 제어기로서 F/T 센서가 아닌 경우는 Mt1; 명령을 입력하여야만 개시되나, F/T 센서이거나 TMS320F28334/5를사용한 제어기의 경우는 전원투입후부터 자동 개시됨. 온도 측정을 멈추려면 Mt0; 명령을 전송하면 됨. NTC를 high-side에 위치시키고 low-side에 4.7K0hm 장착한 상태에서 3.3VA를 인가하여, NTC와 4.7K0hm에 의하여 분할된 전압을 A/D 변환한 다음, NTC의 저항값을 계산함. 측정된 NTC의 저항값과 MT 명령어로 설정된 파라미터를 사용하여, Steinhart-Hart 방정식으로부터 온도를 계산함. ⚡ linear temprature 센서(TC1047)를 사용하는 경우 : 온도값계산을 항상 실시함.

명령어	구분	설명
;MUffff1,ffff2,ffff3; (+^) ;MU?; (+^)	operation	온도측정용 두번째 센서의 파라미터를 설정함. 1st word : constant (temp2_C1) 2nd word : linear coefficient (temp2_C2) 3rd word : quadratic coefficient (temp2_C3) 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	return value	MUffff1,ffff2,ffff3;
	comment	<ul style="list-style-type: none"> ☞ 온도센서가 2개 장착되는 모델 (FT센서로서 외장형)에 한함. 이때 두번째 온도센서란 센서 연결용 screw terminal에서 가장 바깥에 위치한 온도 센서를 가리킴. 이 온도는 센서 연결용 screw terminal에서 가장 바깥에 위치한 strain-gauge bridge의 온도 보상을 위하여 사용됨. ☞ 온도센서로는 NTC 온도센서를 사용하는 경우가 일반적이거나, 특별한 경우 linear temprature 센서 (TC1047)를 사용하는 경우가 있으므로 주의를 요함. ☞ NTC 온도센서를 사용하는 경우 : 온도값은 Ohm 단위의 저항값 R을 사용하여 $r = \ln(R)$을 계산한 다음, $100000000.0 / \{temp2_C1 + temp2_C2 * r + temp2_C3 * r * r * r\} - 23750$ 식을 이용하여 계산함. 계산된 온도값의 단위는 섭씨 0.01도 임. NTC 파라미터의 기본값은 temp2_C1 = 1377.113100035604; temp2_C2 = 234.029059587146; temp2_C3 = 0.354431764337; 이며, 이는 -10도 55480 Ohm, 25도 10000 Ohm, 60도 2365 Ohm 인 103k5(D20k) 모델의 파라미터에 해당함. ☞ linear temprature 센서(TC1047)를 사용하는 경우 : (온도센서 A/D 변환값 - temp2_C1) * temp2_C2 식을 이용하여 온도값을 계산함. 계산된 온도값의 단위는 섭씨 0.01도 임. temp2_C1의 기본값은 682.5로서 0.5V에 해당하는 A/D 변환값이며, temp2_C2의 기본값은 7.326으로서 1V당 100도 (0.01도 단위로는 10000)의 값을 갖도록 하는 scale factor 값임. <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

명령어	구분	설명
;Mudddd1; (+^) ;Mu?; (+^)	operation	온도측정용 두번째 센서를 사용하여 측정된 저항치 및 온도값을 읽어내어 host로 전송함. 1st word : 온도측정을 실시하는 지의 여부 [0 ~ 1] (0 or 1)
	return value	Mudddd1, dddd2, iii i3; 1st word : 온도측정을 실시하는 지의 여부 (0 or 1) 2nd word : NTC의 저항값 (0hm 단위) : NTC 온도센서를 사용하는 경우에 한함. 3rd integer : 온도값 (0.01도 단위)
	comment	<p>☞ 온도센서가 2개 장착되는 모델 (FT센서로서 외장형)에 한함. 이때 두번째 온도센서란 센서 연결용 screw terminal에서 가장 바깥에 위치한 온도 센서를 가리킴. 이 온도는 센서 연결용 screw terminal에서 가장 바깥에 위치한 strain-guage bridge의 온도 보상을 위하여 사용됨.</p> <p>☞ NTC 온도센서를 사용하는 경우 : NTC의 저항값은 전원투입후부터 계산을 실시함. 온도값 계산은 TMS320F2809/TMS320F2810/11/12을 사용한 제어기로서 F/T 센서가 아닌 경우는 Mt1; 명령을 입력하여야만 개시되나, F/T 센서이거나 TMS320F28334/5를 사용한 제어기의 경우는 전원투입후부터 자동 개시됨. 온도 측정을 멈추려면 Mt0; 명령을 전송하면 됨.</p> <p>NTC를 high-side에 위치시키고 low-side에 4.7K0hm 장착한 상태에서 3.3VA를 인가하여, NTC와 4.7K0hm에 의하여 분할된 전압을 A/D 변환한 다음, NTC의 저항값을 계산함.</p> <p>측정된 NTC의 저항값과 MT 명령어로 설정된 파라미터를 사용하여, Steinhart-Hart 방정식으로부터 온도를 계산함.</p> <p>☞ linear temprature 센서(TC1047)를 사용하는 경우 : 온도값계산을 항상 실시함.</p>

온도센서 설정 과정

1. 전압 측정값의 calibration 실시
 - Z0-1; Z1-1; ZsA55A;를 실행한 후, 전원을 껏다켜고
 - Sv?;를 실행한 상태에서 10V 전압을 공급한 다음 (정확한 멀티미터로 제어기 전원공급단자에서 확인) Sv 마지막 리턴값 A를 읽을것
 - Sv?;를 실행한 상태에서 20V 전압을 공급한 다음 (정확한 멀티미터로 제어기 전원공급단자에서 확인) Sv 마지막 리턴값 B를 읽을것
 - Z1 parameter값 = $16384 * 1000 / (B - A)$
 Z0 parameter값 = $(Z1 \text{ parameter값} * A / 16384) - 1000$
 을 계산하여 Z0, Z1명령어를 사용하여 부호있는 10진수로 입력한 다음
 - ZsA55A; 를 실행한 후
 - 전원을 껏다 켜고 Sv?; 명령어로 보정된 전압값 확인
2. 온도센서의 calibration 실시 (linear temperature sensor의 경우)
 - 첫번째 온도센서의 경우 Qx8;를 사용하여 네번째값인 온도값(0.01도 단위)을 보면서 MT 명령어의 첫번째 파라미터를 조정하여 현재 온도값과 같게 되게 맞추고
 - 두번째 온도센서의 경우 Qx22;를 사용하여 네번째값인 온도값(0.01도 단위)을 보면서 MU 명령어의 첫번째 파라미터를 조정하여 현재 온도값과 같게 되게 맞추고
 - EsA55A;로 저장
2. 온도센서의 calibration 실시 (NTC의 경우)
 - 다음 Matlab 코드를 이용하여 온도계산 파라미터를 계산함.
 예를 들어서 -10°C에서 55480Ω, 25°C에서 10000Ω, 60°C에서 2365Ω 의 NTC라면

```

r1=log(55480);
T1=-10;
r2=log(10000);
T2=25;
r3=log(2365);
T3=60;
a=[1 r1 r1*r1*r1;
  1 r2 r2*r2*r2;
  1 r3 r3*r3*r3]
b=[1/(T1+237.5);1/(T2+237.5);1/(T3+237.5)]
c=1000000*inv(a)*b
          
```
 - 앞서 계산된 c 값 3개를 MT(첫번째 온도센서)/MU(두번째 온도센서) 명령어를 사용하여 입력하고
 - EsA55A;로 저장

온도센서 설정 과정

3. 온도값의 확인

- Q8;(첫번째 온도센서) 또는 Q22;(두번째 온도센서)를 사용하여 온도값의 정확도 확인

계산된 온도 값은 Mt(첫번째 온도센서), Mu(두번째 온도센서) 명령을 사용하여 읽어낼수 있음.

⚡ 온도센서가 2개 연결가능한 외장형 F/T 센서인 경우에는, 단자대 마지막 단자에 연결된 온도센서가 두번째 온도 센서이며, 이보다 2개 단자 앞에 연결된 온도센서가 첫번째 온도센서임.

명령어	구분	설명
;MPffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12; (+#^)	operation	4개의 strain guage 신호를 증폭하고 AD 변환된 값을 이용하여 X-moment, Y-moment, Z-force를 계산하는 변환행렬 값을 설정함. ffff1 - ffff4 : AD2FT11 - AD2FT14 (X-moment를 계산하기 위한 계수) ffff5 - ffff8 : AD2FT21 - AD2FT24 (Y-moment를 계산하기 위한 계수) ffff9 - ffff12 : AD2FT31 - AD2FT34 (Z-force를 계산하기 위한 계수) 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	return value	MPffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12;
	comment	F/T 센서 회로에서만 유효한 명령임. 4개의 계수는 순서대로 각각 bridge amplifier #3/#4/#1/#2의 전압값에 곱해짐. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
;Mpffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12; (+#^)	operation	6개의 strain guage 신호를 증폭하고 AD 변환된 값을 이용하여 X-moment, Y-moment, Z-force, X-force를 계산하는 변환행렬 값을 설정함. 2011년 11월 16일 S/W 버전부터 사용가능한 명령임. ffff1 - ffff2 : AD2FT15, AD2FT16 (X-moment를 계산하기 위한 계수) ffff3 - ffff4 : AD2FT25, AD2FT26 (Y-moment를 계산하기 위한 계수) ffff5 - ffff6 : AD2FT35, AD2FT36 (Z-force를 계산하기 위한 계수) ffff7 - ffff12 : AD2FT41 - AD2FT46 (X-force를 계산하기 위한 계수) 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	return value	Mpffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12;
	comment	6축 F/T 센서 회로에서만 유효한 명령임. (단 삼성메디컬의 경우 첫 4개의 파라메터가 보조모멘트 측정용) 계수는 순서대로 각각 bridge amplifier #5/#6 또는 #3/#4/#1/#2/#5/#6의 전압값에 곱해짐. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
;Mp? (+#^)		

명령어	구분	설명
;Mqffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12; (+#^) ;Mq? (+#^)	operation	6개의 strain guage 신호를 증폭하고 AD 변환된 값을 이용하여 Y-force, Z-moment를 계산하는 변환행렬 값을 설정함. 2011년 11월 16일 S/W 버전부터 사용가능한 명령임. ffff1 - ffff6 : AD2FT51 - AD2FT56 (Y-force를 계산하기 위한 계수) ffff7 - ffff12 : AD2FT61 - AD2FT66 (Z-moment를 계산하기 위한 계수) 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	return value	Mqffff1,ffff2,ffff3,ffff4,ffff5,ffff6,ffff7,ffff8,ffff9,ffff10,ffff11,ffff12;
	comment	6축 F/T 센서 회로에서만 유효한 명령임. 계수는 순서대로 각각 bridge amplifier #3/#4/#1/#2/#5/#6의 전압값에 곱해짐. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
(F/T센서의 경우) ;M0dddd1,dddd2,dddd3,dddd4; 또는 ;M0dddd1,dddd2,dddd3,dddd4,dddd5,dddd6; (+#^)	operation	F/T 센서에서 TLV5630 또는 DAC124S085 D/A converter의 출력값을 설정함. 1st word : channel #1의 D/A 출력값을 설정함. [0-4095] (1536 bit) 2nd word : channel #2의 D/A 출력값을 설정함. [0-4095] (1536 bit) 3rd word : channel #3의 D/A 출력값을 설정함. [0-4095] (1536 bit) 4th word : channel #4의 D/A 출력값을 설정함. [0-4095] (1536 bit) 5th word : channel #5의 D/A 출력값을 설정함. [0-4095] (1536 bit) - 6축 F/T센서인 경우 6th word : channel #6의 D/A 출력값을 설정함. [0-4095] (1536 bit) - 6축 F/T센서인 경우
	return value	M0dddd1,dddd2,dddd3,dddd4; 또는 M0dddd1,dddd2,dddd3,dddd4,dddd5,dddd6;
	comment	F2809 내장형 또는 F2811 외장형으로 D/A 변환기가 장착된 F/T 센서 회로에서만 유효한 명령임. 5번/6번째 값은 6축 F/T센서인 경우에만 해당됨. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
(절대각센서를 사용하는 경우) ;M0iiii1,iiii2; (#) ;M0?; (#)	operation	절대각센서를 사용하는 경우에 BLDC 모터의 Hall A/B/C상 신호와 모터위상각신호의 위상차를 없애기 위한 위상 offset을 설정함. 1st word : 모터1의 위상 offset [-32768 ~ 32767] (0 bit) 2nd word : 모터2의 위상 offset [-32768 ~ 32767] (0 bit)
	return value	M0iiii1,iiii2;
	comment	F28334를 사용한 절대각센서를 사용한 모델에만 적용이 가능함. 모터를 손으로 서서히 돌리거나, 08번 모드(BLDC의 경우) 또는 07번 제어모드(DC 모터의 경우)로 낮은 속도로 모터를 구동하면서, Qx5; 명령어로 Hall A/B/C상 신호와 모터위상각신호를 관찰하되, Hall A상 신호가 상승할때 모터위상각신호가 0이 되도록 M0 명령을 사용하여 조정함. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;MSiiii1,iiii2,iiii3,iiii4; 또는 ;MSiiii1,iiii2,iiii3,iiii4,iiii5,iiii6; (+#^)	operation	F/T 센서에서 증폭후 A/D 변환된 스트레인 값의 온도증가에 따른 변화율을 설정함. 1st integer : channel #1의 변화율. [-32768 ~ 32767] 2nd integer : channel #2의 변화율. [-32768 ~ 32767] 3rd integer : channel #3의 변화율. [-32768 ~ 32767] 4th integer : channel #4의 변화율. [-32768 ~ 32767] 5th integer : channel #5의 변화율. [-32768 ~ 32767] - 6축 F/T센서인 경우 6th integer : channel #6의 변화율. [-32768 ~ 32767] - 6축 F/T센서인 경우
	return value	MSiiii1,iiii2,iiii3,iiii4; 또는 MSiiii1,iiii2,iiii3,iiii4,iiii5,iiii6;
	comment	변화율은 (A/D 변환된 스트레인 값의 변화량) * 16384 / (0.01도 단위의 온도변화) 로 계산된 값이어야 함. 온도증가량의 정의는 SI 명령어로 A/D 변환기 offset을 설정할때의 온도값을 기준으로한 변화량을 의미함. 5번/6번째 값은 6축 F/T센서인 경우에만 해당됨. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.
;MViiii1,iiii2; (+#^)	operation	F/T 센서에서 온도보상의 기준온도와 온도보상 기울기를 자동 계산할 때 사용한 또 다른 온도값을 설정함. 1st integer : 온도보상의 기준온도 (0.01도 단위) 2nd integer : 온도보상 기울기를 자동 계산할 때 사용한 또 다른 온도값 (0.01도 단위)
	return value	MViiii1,iiii2;
	comment	온도보상의 기준온도는 SIA55A; 명령에 의하여 앰프와 스트레인 게이지의 영점을 설정할때 자동으로 설정됨. 온도보상 기울기를 자동 계산할 때 사용한 또 다른 온도값은 SI5AA5; 명령에 의하여 자동으로 설정됨. MV 명령은 특별한 이유가 없는 한, 수동으로 값을 설정하는데 사용하지 마시오. 2012년 3월 13일 S/W 버전부터 사용가능한 명령임. 설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.

명령어	구분	설명
;Moiiii1,iiii2,iiii3,iiii4,iiii5 ;(+#^) ;Mo?; (+#^)	operation	F/T 센서에서 수직력의 auto zero rate, maximum allowed drift, X 모멘트 coupling, Y 모멘트 coupling , FORCE_SENSOR_ERROR_DETECTION_TIME 파라미터를 설정함. 1st integer : FT_RETURN_HOME_RATE [0 ~ 32767] (10 0.1수직력단위크기/sec) 2nd integer : FT_DRIFT_ERROR_BOUND [0 ~ 32767] (200 수직력단위크기) 3rd integer : X_MOMENT_COUPLING [-32768 ~ 32767] (0 0.1% of 수직력) 4th integer : Y_MOMENT_COUPLING [-32768 ~ 32767] (0 0.1% of 수직력) 5th integer : FORCE_SENSOR_ERROR_DETECTION_TIME [0 ~ 65535] (2000 ms)
	return value	Moiiii1,iiii2,iiii3,iiii4,iiii5;
	comment	<p>☞ FT_RETURN_HOME_RATE는 기본값이 10 이며, 이는 1초에 수직력의 단위크기만큼 수직력의 크기를 줄인다는 뜻과 같음. 이렇게 함으로서 수직력에 drift 존재할 때 drift를 없애는 작용을 하도록 함. 이 작용은 drift 보상된 수직력의 크기가 두번째 파라미터인 FT_DRIFT_ERROR_BOUND 보다 작을 때만 실시됨. 단점으로는 수직력의 크기가 두번째 파라미터인 FT_DRIFT_ERROR_BOUND 보다 작은 상태로 시간이 지속되면, 수직력의 크기가 FT_RETURN_HOME_RATE의 비율로 줄어든다는 단점이 있음. 따라서 수직력이 return-to-zero의 특성을 가지고 있지 않을 때는 0이 아닌 값을 부여하는데 주의가 필요함.</p> <p>☞ FT_DRIFT_ERROR_BOUND 는 FT_RETURN_HOME_RATE에 의하여 보정된 drift의 총량이 이 값을 초과할때 오류를 발생시키기 위한 값임. 단위는 수직력의 단위크기와 같음. drift 총량이 허용치를 초과하기를 FORCE_SENSOR_ERROR_DETECTION_TIME이상 경과하게 되면 수직력 및 모멘트의 값은 0으로 처리됨. 이 오류가 발생하면 23번 숫자를 램프로 flashing하며, Q2; 명령어의 리턴값에서 첫번째 워드의 bit6가 1로 세트됨. 오류에서 벗어나려면 SI 명령어로 offset 설정을 다시하면 됨.</p> <p>☞ X/Y_MOMENT_COUPLING 값은 단위가 0.1% 이며, 수직력의 크기에 따라서 모멘트의 값이 변하는 경우 모멘트의 변화를 없애는데 사용하는 값임. 수직력크기 * (X/Y_MOMENT_COUPLING * 0.001) 만큼 모멘트의 크기를 뺀.</p> <p>☞ 33번 숫자가 램프로 flashing하면 F/T센서의 4개 bridge amp 중 한개 이상이 비정상적인 값을 갖기를 FORCE_SENSOR_ERROR_DETECTION_TIME이상 경과한 것이며, 이 경우에도 수직력 및 모멘트의 값은 0으로 처리됨. 또한 Q2; 명령어의 리턴값에서 첫번째 워드의 bit7이 1로 세트됨. 정상범위로 값이 복귀되면 에러상태는 정상작동상태로 자동복구됨.</p> <p>설정된 값은 ES 명령어로 EEPROM에 저장될 수 있음.</p>

F/T 센서 설정 과정

1. 전압 측정값의 calibration 실시
 - Z0-1; Z1-1; ZsA55A;를 실행한 후, 전원을 껐다켜고
 - Sv?;를 실행한 상태에서 10V 전압을 공급한 다음 (정확한 멀티미터로 제어기 전원공급단자에서 확인) Sv 마지막 리턴값 A를 읽을것
 - Sv?;를 실행한 상태에서 20V 전압을 공급한 다음 (정확한 멀티미터로 제어기 전원공급단자에서 확인) Sv 마지막 리턴값 B를 읽을것
 - Z1 parameter값 = $16384 * 1000 / (B - A)$
 Z0 parameter값 = $(Z1 \text{ parameter값} * A / 16384) - 1000$
 을 계산하여 Z0, Z1명령어를 사용하여 부호있는 10진수로 입력한 다음
 - ZsA55A; 를 실행한 후
 - 전원을 껐다 켜고 Sv?; 명령어로 보정된 전압값 확인
2. 온도센서의 calibration 실시 (linear temperature sensor의 경우)
 - Qx8;를 사용하여 네번째값인 온도값(0.01도 단위)을 보면서 MT 명령어의 첫번째 파라미터를 조정하여 현재 온도값과 같게 되게 맞추고 (온도센서가 2개 장착 가능한 외장형 모델인 경우는 단자대 마지막 단자 옆/옆의 단자에 온도센서를 연결해야함)
 - EsA55A;로 저장
2. 온도센서의 calibration 실시 (NTC의 경우)
 - 다음 Matlab 코드를 이용하여 온도계산 파라미터를 계산함.
 예를 들어서 -10℃에서 55480Ω, 25℃에서 10000Ω, 60℃에서 2365Ω 의 NTC라면


```

r1=log(55480);
T1=-10;
r2=log(10000);
T2=25;
r3=log(2365);
T3=60;
a=[1 r1 r1*r1*r1;
1 r2 r2*r2*r2;
1 r3 r3*r3*r3]
b=[1/(T1+237.5);1/(T2+237.5);1/(T3+237.5)]
c=1000000*inv(a)*b

```
 - 앞서 계산된 c 값 3개를 MT 명령어를 사용하여 입력하고
 - EsA55A;로 저장

F/T 센서 설정 과정

3. 온도값의 확인
 - Q8; 을 사용하여 온도값의 정확도 확인 (두번째값이 NTC 저항값(Ω 단위)이며 마지막 값이 온도값(0.01도 단위)임)
4. F/T센서의 calibration 실시
 - Qx3;를 사용하여 증폭된 전압값이 2000근방의 값이 되도록 M0 명령을 사용하여 offset을 조정함. (증폭된 전압값이 작으면 M0 명령어로 기본값 보다 큰 값을 입력하면 됨)
 - SI5000,256; 명령어를 사용하여 offset을 설정하고 (이때 F/T센서에는 5초간 힘이 인가되면 안되며, 온도는 일정하게 유지되어야함)
 - MS0,0,0,0;
 - 같은 온도를 유지하면서 아는 하중을 인가하여 Qx4;로 offset이 제거된 값을 기록하였다가 maxtix을 만든 다음 (이때 1번 채널을 다른 용도로 사용하는 H/W 버전이라면 1번채널의 측정값(수직력1)은 2번채널의 측정값(수직력2)을 copy하여 사용할 것.)
 - MP 명령어로 matrix 값을 설정한 다음
 - EsA55A;로 저장
5. F/T 센서값의 확인
 - Qx5; 이용하여 아는 하중을 인가하면서 계산된 모멘트/force 값을 확인
6. F/T센서의 온도보상 실시
 - 좀더 높은 온도 (예 : 40도)를 유지하면서 Qx3;를 사용하여 증폭된 전압값을 기록
 - 이 값을 이용하여 MS 명령어로 보상값을 설정함
 - EsA55A;로 저장
7. F/T 센서값의 온도에 따른 확인
 - Qx5; 이용하여 계산된 모멘트/force 값을 온도를 변화시켜가면서 확인

계산된 온도 값은 Mt 명령을 사용하여 읽어낼수 있음.

계산된 F/T 값은 Qf; 명령을 사용하여 읽어내거나, Qb3; 명령을 사용하여 읽어 낼수 있음.

명령어	구분	설명
(F/T센서가 아닌 경우로서 2013년 11월 22일 S/W 버전부터)	operation	resolver를 위치센서로 사용하는 경우, U/V/W상 전류가 한 사이클 변하면서 움직이는 동안 발생하는 resolver 센서 신호의 사이클 수를 설정함. 2013년 11월 22일 S/W 버전부터 사용가능한 명령임. 1st word : 모터1의 resolver 센서 신호의 사이클 수. [1-32767] (16 개) 2nd word : 모터2의 resolver 센서 신호의 사이클 수. [1-32767] (16 개)
	return value	Msddd1,ddd2;
	comment	U/V/W상 전류 한 사이클에 대하여 51.2 mm가 움직이는 리니어모터에 대하여 200um 피치의 리졸버를 위치센서로 사용하는 경우 : (리니어모터는 SE 명령에 의한 pole pair 값을 1로 설정해야함) 리니어모터 51.2 mm 1 피치당 리졸버 사인파 갯수는 $(51.2\text{mm} / 0.2\text{mm}) = 256$ 개 U/V/W상 전류 한 사이클에 대하여 18 mm가 움직이는 리니어모터에 대하여 20um 피치의 리졸버를 센서로 사용하는 경우 : (리니어모터는 SE 명령에 의한 pole pair 값을 1로 설정해야함) 리니어모터 18 mm 1 피치당 리졸버 사인파 갯수는 $(18\text{mm} / 0.02\text{mm}) = 900$ 개 리졸버를 사용하지 않는 경우는 16의 기본값이 사용됨. 2013년 11월 22일 이전 S/W를 2013년 11월 22일 이후 S/W 버전으로 upgrade 하는 경우에, 리졸버를 사용하는 경우라면 Ms 명령어를 사용하여 사이클수를 설정하여야함. 파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.
(F/T센서인 경우)	operation	삼성메디컬용 F/T 센서의 X-moment, Z-force coupling을 처리하는 파라미터를 설정함. 1st word : Z-force를 강제설정하는 X-moment의 크기 [0 - 30000] (1500 0.01Nm) 2nd word : X-moment가 클때 Z-force를 강제설정하는 Z-force 크기 [0 - 30000] (500 0.1N)
	return value	Msddd1,ddd2;
	comment	삼성메디컬에 적용되는 4-bridge F/T 센서에 한함. X-moment가 1st word로 설정하는 모멘트보다 크면 Z-force는 크기에 관계없이 2nd word로 설정되는 force로 강제 설정됨. 2011년 12월 19일 S/W 버전부터 사용가능한 명령임. 파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.

명령어	구분	설명
;MS?; (+#^)	operation	내부에서 사용하는 전류값을 0.01A 단위로 변환하는 scale factor을 읽어서 호스트로 전송함.
	return value	MSffff1,ffff2; 1st 실수 : 모터1의 bit 단위의 전류값을 0.01A 단위로 바꾸어주는 scale factor임. 2nd 실수 : 모터2의 bit 단위의 전류값을 0.01A 단위로 바꾸어주는 scale factor임. 2013년 4월 15일 S/W 버전부터 eeee 형식, 즉 -9 ~ +9 까지의 지수가 있는 실수를 사용할수 있음.
	comment	F/T센서가 아닌 경우로서 2013년 4월 07일 S/W 버전부터 사용가능한 명령임. 1/2/4/8/10/20 mOhm의 전류감지 저항을 사용하는 경우 40.95/81.9/163.8/327.6/409.5/819 bit가 1A에 해당하는데, 이때 scale factor는 2.442/1.221/0.6105/0.30525/0.2442/0.1221 가 됨.
;Mfdddd1; (+#^)	operation	butterworth filterA의 시정수를 읽어서 호스트로 전송함. 1st word : 필터번호 [0 - 31] (0 번호)
	return value	Mfdddd1,dddd2; 1st word : 필터번호 [0 - 31] 2nd word : 필터시정수
	comment	2^(필터시정수-1) [ms] 의 값이 실제 시정수값[ms]에 해당함. 2013년 5월 17일 S/W 버전부터 사용가능한 명령임.
;Mgffff1; (+#^)	operation	엔코더값을 analog 센서에 의한 위치값으로 바꿀때의 scale factor를 설정함.
	return value	Mgffff1; 1st 실수 : 엔코더값을 analog 센서에 의한 위치값으로 바꿀때의 scale factor (1.0)
	comment	

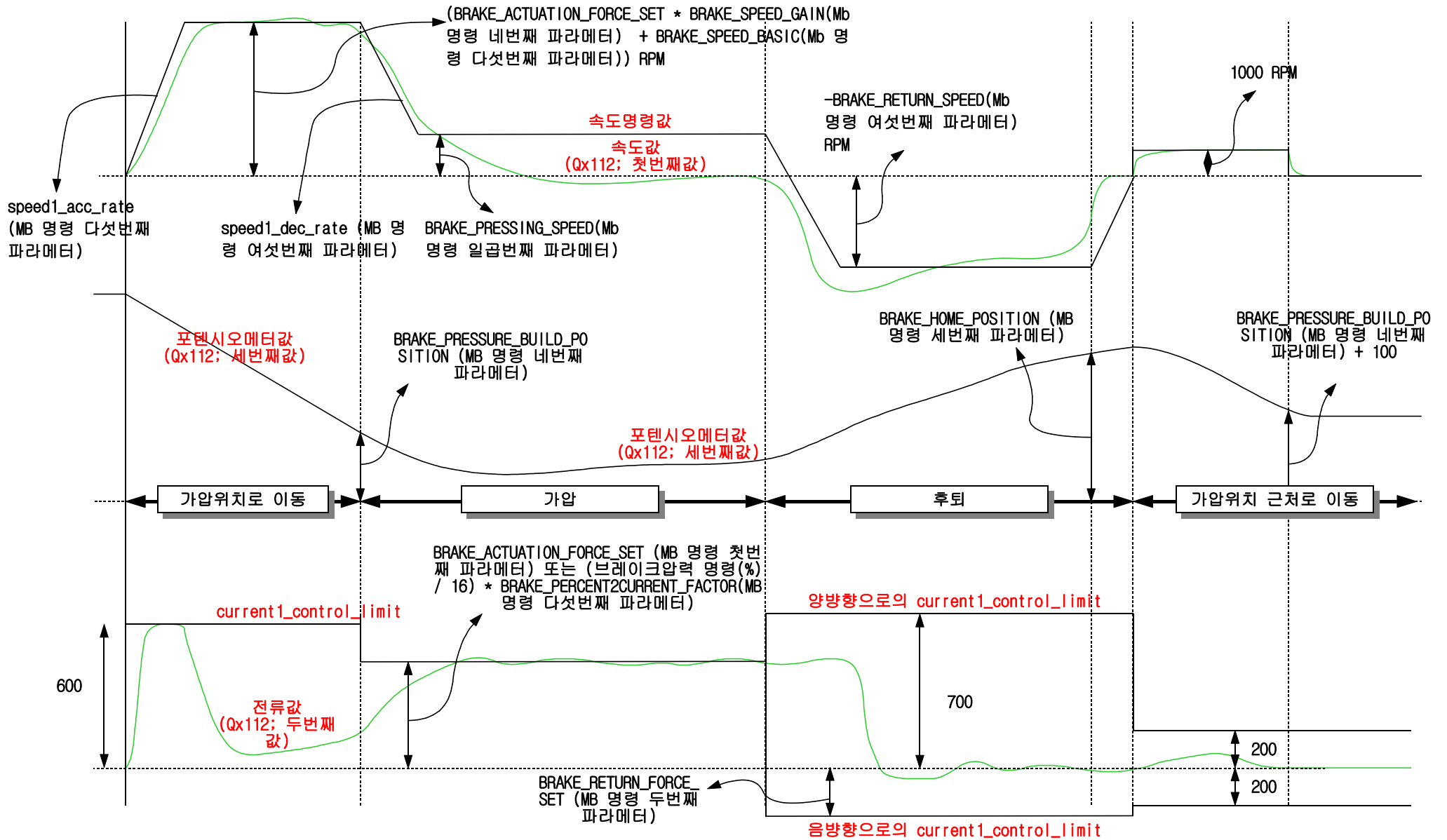
명령어	구분	설명
;MAddd1,ddd2,ddd3; (+#^)	operation	자동화용 H/W 버전에서 상위명령값과 실제값의 차이가 설정값이내에 들어왔는지를 판단하고 mission 완료 를 결정하는 경계값(threshold)을 설정함. 1st word : 위치제어시의 threshold [1 ~ 32767] (100 pulse) 2nd word : 속도제어시의 threshold [1 ~ 32767] (10 rpm) 3rd word : 전류제어시의 threshold [1 ~ 32767] (50 0.01A)
	return value	MAddd1,ddd2,ddd3;
;MA?; (+#^)	comment	상위명령값과 실제값의 차이가 설정값 이내에 들어오면 mission complete 출력을 ON 하며, 설정값의 2배 이상으로 커지면 mission complete 출력을 OFF 함. 속도나 전류제어의 경우는 명령값의 절대크기가 설정값의 2배 이내인 경우에 mission complete 출력을 OFF 함. 2012년 3월 22일 S/W 버전부터 자동화용 H/W 버전에서만 사용가능한 명령임.
;MC; (+#^)	operation	BLDC 또는 AC servo 모터의 Iq, Id 값을 읽어서 host로 전송함.
	return value	MCiiii1,iiii2,iiii3,iiii4; 1st integer : 모터1의 Iq (bit) 2nd integer : 모터1의 Id (bit) 3rd integer : 모터2의 Iq (bit) 4th integer : 모터2의 Id (bit)
	comment	단위크기의 의미는 QC 명령에 대한 리턴값의 의미와 같음. BLDC 또는 AC servo 모터에 한함. 2011년 8월 19일 S/W 버전부터 사용가능한 명령임.
;Mc; (+#^)	operation	BLDC 또는 AC servo 모터의 Iq, Id 값을 읽어서 0.01A 단위의 암페어값으로 환산하여 host로 전송함.
	return value	Mciiii1,iiii2,iiii3,iiii4; 1st integer : 모터1의 Iq (0.01A) 2nd integer : 모터1의 Id (0.01A) 3rd integer : 모터2의 Iq (0.01A) 4th integer : 모터2의 Id (0.01A)
	comment	BLDC 또는 AC servo 모터에 한함. 2011년 8월 19일 S/W 버전부터 사용가능한 명령임. 2012년 4월 10일 이전 S/W 버전에서는 모터2의 전류스케일이 모터1의 전류스케일로 바뀌어져서 계산되는 잘못이 있음.

명령어	구분	설명
;MRddd1,ddd2,ddd3,ddd4; (+#^) ;MR?; (#^) 2013년 4월 4일 S/W 버전부터는 Mr 명령을 사용하여 0.01A 단위로 전류값을 설정할 수 있음.	operation	전류상승률을 제한하는 파라미터를 설정함. 1st word : MOTOR1_CURRENT_RISING_TRIP [0 ~ 4914] (1638, 2m0hm을 전류감지저항으로 사용하는 경우는 20A) 2nd word : MOTOR1_CURRENT_RISING_INCREMENT [0 ~ 10000] (200, 2m0hm을 전류감지저항으로 사용하는 경우는 2442 A/s) 3rd word : MOTOR2_CURRENT_RISING_TRIP [0 ~ 4914] (1638, 2m0hm을 전류감지저항으로 사용하는 경우는 20A) 4th word : MOTOR2_CURRENT_RISING_INCREMENT [0 ~ 10000] (200, 2m0hm을 전류감지저항으로 사용하는 경우는 2442 A/s)
	return value	MRddd1,ddd2,ddd3,ddd4;
	comment	EPS가 아닌 경우로서 BLDC 모터인 경우에 한함. (EPS용 BLDC 모터인 경우에는 MR/Mr 명령이 아니고 MB 명령을 사용함) 2012년 10월 19일 S/W 버전부터 적용됨. 2013년 10월 99일 S/W 버전부터는 DC 모터제 어기에도 적용됨. 1st/3rd 의 값이 1638이면 20A에 해당함. (2m0hm을 전류감지저항으로 사용하는 경우) 이 값보다 더 큰 전류명령값이 계산되는 경우 전류상승률제한 로직을 적용함. 2nd/4th word 의 값이 200이면 1ms당 2.442A의 상승을 허용한다는 말임. (2m0hm을 전류감지저항으로 사용하는 경우) 2013년 4월 4일 S/W 버전부터는 Mr 명령을 사용하여 0.01A 단위로 전류값을 설정할수있음. 파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.
Mrddd1,ddd2,ddd3,ddd4; (#^) Mr?; (#^) 2013년 4월 4일 S/W 버전부터 사용 가능함. 파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.	operation	전류상승률을 제한하는 파라미터를 설정함.
	return value	MRddd1,ddd2,ddd3,ddd4; 값의 단위는 0.01A 임.
	comment	2013년 4월 4일 S/W 버전부터 사용 가능함. 파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.

명령어	구분	설명
;MBdddd1, dddd2, d ddd3, dddd4, dddd 5, dddd6, dddd7, dd dd8, dddd9; (#)	operation	brake actuator의 동작관련 파라미터를 설정함. 1st word : 가압시의 전류설정값 [0 - 1000] (200 bit) 2nd word : 복귀시의 전류설정값 [0 - 100] (50 bit) 3rd word : 복귀위치의 포텐시오미터 값 [0 - 4095] (3200 bit) 4th word : 가압개시위치의 포텐시오미터 값 [0 - 4095] (3050 bit) 5th word : 가압시의 가속률 [1 - 60000] (6000 rpm/0.1sec) 6th word : 가압시의 감속률 [1 - 60000] (3000 rpm/0.1sec) 7th word : %단위의 brake 압력 명령을 전류명령값으로 환산하는 scale factor [2*16 - 12*16] (7*16 bit, 2012년 7월 09일 S/W 버전부터 적용) 8th word : 마스터실린더 피스톤의 과도한 전진을 판단하는 위치값 [0 - 4095] (1300 bit, 2012년 7월 09일 S/W 버전부터 적용) 9th word : 마스터실린더 피스톤의 과도한 전진을 판단하는 위치직전의 구간크기 [10 - 500] (100 bit, 2012년 7월 17일 S/W 버전부터 적용)
	return value	MBdddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7, dddd8, dddd9;
	comment	brake actuator에 한함. SV 또는 SVr 명령어로 +속도를 주면 가압이 개시됨. 가압이 개시되면 정상 속도(가압시의 전류설정치에 따라서 다르게 설정됨)까지 가속하면서 전진하다가, 포텐시오미터의 가압개시 위치에 도달하면 1000 RPM 까지 감속하면서, 가압시의 전류설정치를 전류상한치로 유지하면서 전진을 계속 함. 이때 가압시의 전류설정치를 바꾸면 가압력이 그에 따라서 바뀜. SV 또는 SVr 명령어로 +속도를 주기전에 가압시의 전류설정값을 바꾸면 가압된 결과압력값이 다르게 발생하는데, 100인 경우에 약 100bar, 200인 경우에 약 200bar, 250인 경우에 약 250bar가 발생함. 단 감속기의 정지마찰력 때문에 정지상태의 최종 압력은 오차가 있음. SV 또는 SVr 명령어로 -속도를 주면 복귀가 개시됨. 복귀시의 속도는 3000rpm이며 포텐시오미터의 복귀위치에 도달하면 정지함. 작동상태는 Qx2;로 전류치를, Qx1;으로 속도값을, Qx36;으로 포텐시오미터 변위를, Qx111/112;로 압력제어상태를 볼수 있음. 피스톤의 위치가 8번째값에서 9번째값을 더한 위치에 도달하면 명령 압력값의 크기를 줄이기 시작하며, 8 번째값에 도달하면 0이 됨. 마지막 7개의 파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.
;MB?; (#)		

명령어	구분	설명
;Mbddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7; (#) ;Mb?; (#)	operation	brake actuator의 동작관련 파라미터를 설정함. 1st word : CAN 통신이 끊어지고 비상정지가 ON 되어 있을때 비상정지 압력을 가하는 것을 enable(1), disable(0) 하는 파라미터값. 전원투입시의 기본값은 enable(1) 임. 2nd word : 브레이크 압력이 낮음을 판단하는 압력값 [0 - 200] (200 bar) 3rd word : 브레이크 압력이 낮음을 판단하는 피스톤 위치의 포텐시오메터 값 [0 - 4095] (1400 bit) 4th word : 가압시 압력설정값에 따른 속도증가비율 [0 - 50] (15) 5th word : 가압시 피스톤속도 기본값 [1 - 10000] (5000 rpm) 6th word : 리턴시 피스톤속도 [1 - 10000] (3000 rpm) 7th word : 브레이크 압력생성시기에서의 속도 [1 - 8000] (1000 rpm)
	return value	Mbddd1, dddd2, dddd3, dddd4, dddd5, dddd6, dddd7;
	comment	brake actuator에 한함. CAN 통신이 끊어진 상태에서 비상정지 압력을 가하는 것이 disable 되어 있을 때는, MB 명령어로 가압시의 전류설정값을 설정하고 SV 또는 SVr 명령어로 가압을 개시할수 있음. 이 명령어는 2012년 7월 23일 S/W 버전부터 적용됨.
;MEddd1, dddd2, dddd3, dddd4; (#) ;ME?; (#)	operation	brake actuator의 동작관련 파라미터를 설정함. 1st word : CAN 통신이 끊어지고 비상정지가 ON 되기 까지의 지연시간 [10 - 65000] (5000 ms) 2nd word : 주전원이 차단되었다고 판단하는 경계전압 [1000 - 9000] (6500 0.01V) 3rd word : 주전원이 확실히 복구되었다고 판단하는 경계전압 [1000 - 10000] (7200 0.01V) 4th word : 주전원이 차단되었다고 판단한 후에 주전원 차단으로 최종 판정하기까지의 지연시간 [10 - 2000] (200 ms)
	return value	MEddd1, dddd2, dddd3, dddd4;
	comment	brake actuator에 한함. CAN 통신이 끊어진 상태에서 비상정지 압력을 가하는 것이 disable 되어 있을 때는, MB 명령어로 가압시의 전류설정값을 설정하고 SV 또는 SVr 명령어로 가압을 개시할수 있음. 이 명령어는 2012년 8월 27일 S/W 버전부터 적용됨.

Brake Actuator 작동 파형 (정상 작동시)



제어기 명령어 LIST

Qxddd1 에 의하여 전송되는 변수값 (브레이크제어기의 경우)

브레이크 제어기인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
111	speed1_RPM	current1_AD	speed1_cmd_old / SPEED_SCALE_FACTOR	current1_control_limit
112	speed1_RPM	current1_AD	ADC_RESULT13_reg >> 4 (potentiometer)	ADC_RESULT12_reg >> 4 (pressure sensor)
115	(UINT)(vfilTA[10] * 1.538461538) (공급전압)	brake_release_timer	ADC_RESULT13_reg >> 4 (potentiometer)	ADC_RESULT12_reg >> 4 (pressure sensor)
117	ECU_fault_status	brake_engage_timer	ADC_RESULT13_reg >> 4 (potentiometer)	ADC_RESULT12_reg >> 4 (pressure sensor)

Brake Actuator 오류코드

오류코드	오류코드의 의미	발생원인	조치
0x8000	모터제어기에 구동 불가능한 fault가 발생하였거나 정상 작동모드로 진입을 하지 못하고 있다는 fault code	모터제어기에 구동 불가능한 fault가 발생했을 경우	재부팅
		모터제어기에 구동 불가능한 fault는 없으나 정상작동 모드로 진입하지 못했을 경우	기다림
0x4000	MB명령어 마지막에서 두번째 파라미터로 설정한 피스톤 마스터실린더 전진 위치보다 더 많이 전진했다는 fault code, 이 경우는 압력값 명령이 0으로 강제설정됨.	브레이크 액이 부족할 경우	브레이크액 보충
		MB명령어 마지막에서 두번째 파라미터값이 너무 클 경우	MB명령어 마지막에서 두번째 파라미터값을 작게 설정
		가압을 위한 전류 명령값이 너무 큰 경우	MB명령어 마지막에서 세번째 파라미터값을 작게 설정
0x2000	MB명령어 마지막에서 두번째 파라미터에 마지막 파라미터를 더한 마스터실린더 피스톤 전진 위치보다 더 많이 전진했다는 fault code, 이 경우는 압력값 명령이 전진변위량에 비례적으로 줄어듦.	브레이크 액이 부족할 경우	브레이크액 보충
		MB명령어 마지막 파라미터값이 너무 클 경우	MB명령어 마지막 파라미터값을 작게 설정
0x1000	정상 작동모드로 진입을 하지 못하고 있다는 fault code	부팅중이어서 정상작동 모드로 진입하지 못했을 경우	기다림
		모터제어기에 구동 불가능한 fault가 발생했을 경우	재부팅
0x0800	과도하게 낮은 전압 상태가 되었음을 알리는 fault code	부하전류가 과도하게 흘러서 저전압 상태가 된 경우	Sw 명령어로 사용가능한 전류최대치를 줄임
0x0400	모터의 코일 온도를 추정한 값이 과도하게 큰 값을 알리는 fault code	부하전류가 과도하게 흘러서 모터의 온도가 과도하게 상승함으로 추정될 때	Sw 명령어로 사용가능한 전류최대치를 줄임
		모터 코일온도 추정관련 파라미터가 적절하지 않을 때	코일 온도추정 관련 ZQ 파라미터 수정
0x0200	모터제어기의 구동소자 온도를 추정한 값이 과도하게 큰 값을 알리는 fault code	부하전류가 과도하게 흘러서 모터제어기의 구동소자 온도가 과도하게 상승함으로 추정될 때	Sw 명령어로 사용가능한 전류최대치를 줄임
		모터 구동소자 온도 추정관련 파라미터가 적절하지 않을 때	관련 Z8/Z9 파라미터 수정
0x0080	피스톤이 정해진 위치까지 전진하였으나 설정된 압력보다 낮은 압력이 형성되었음	브레이크오일 부족이나 배관/마스터실린더 손상	배관/마스터실린더 점검
		Mb명령어 두번째와 세번째 파라미터가 적절치 않을 때	Mb명령어 두번째와 세번째 파라미터 재설정

Brake Actuator 오류코드

오류코드	오류코드의 의미	발생원인	조치
0x0010	정상작동을 하던 중에 CAN 통신이 차단되어 브레이크를 비상정지하였다는 fault code	CAN 통신이 장시간 차단된 경우	너무 예민하다고 판단되면 ME 명령어의 첫번째 파라미터를 크게함.
0x0020	브레이크가 가압되어 있는 상태에서, 정상작동을 하던 중에, 공급전압이 현저히 떨어져서 주전원이 차단되었다고 판단하고, 브레이크를 후퇴시켰다는 fault code (0x0010 오류코드와 함께 0x0020 오류코드가 발생한 경우는 0x0020 오류코드에 의한 브레이크작동 해지가 우선함)	브레이크가 가압되어 있는 상태에서 후퇴시키지 않고 주전원을 차단한 경우	제어기가 손상이 되지 않은 경우는 무시하되, 손상이 되었다면 ME 명령어의 2~4번 파라미터를 예민하게 보호하도록 조정함.
		주전원 차단이 안된 상태에서 너무 예민하게 브레이크를 하지	ME 명령어의 2~4번 파라미터를 덜 예민하게 보호되도록 조정함.
0x0100	0x34번 메시지 브레이크압 설정값의 한도 초과	CAN 통신 수신 데이터 오류	
0x0001	0x34번 메시지(braking command) 수신 time over	CAN 통신이 장시간 차단된 경우	
0x0008	0x36번 메시지(driving command) 수신 time over	CAN 통신이 장시간 차단된 경우	
0x0002	0x57번 메시지(steering status) 수신 time over	CAN 통신이 장시간 차단된 경우	
0x0004	0x58번 메시지(driving status) 수신 time over	CAN 통신이 장시간 차단된 경우	

명령어	구분	설명
;MBiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7; (+#) ;MB?; (+#)	operation	<p>EPS의 동작관련 파라미터를 설정함.</p> <p>1st word : EPS_POSITION_SCALE_FACTOR [1 - 200] (85 pulse/0.1%)</p> <p>2nd word : EPS_POSITION_REDUCTION_FACTOR [0 - 3000] (2730 0.00006105%/RPM)</p> <p>3rd word : EPS_POSITION_REDUCTION_MIN [100 - 1000] (200 0.1%)</p> <p>4th word : EPS_CENTER_OFFSET [-200 ~ 200] (0 0.1%)</p> <p>5th word : EPS_CURRENT_CONTROL_LIMIT_UP_DOWN_RATE [0 ~ 100] (16)</p> <p>6th word : MOTOR1_CURRENT_RISING_TRIP [0 ~ 4914] (2457, 0.5mOhm을 전류감지저항으로 사용하는 경우는 409)</p> <p>7th word : MOTOR1_CURRENT_RISING_INCREMENT [0 ~ 10000] (10, 0.5mOhm. 전류감지저항으로 사용하는 경우는 2)</p>
	return value	MBiiii1,iiii2,iiii3,iiii4,iiii5,iiii6,iiii7;
	comment	<p>EPS에 한함. 속도가 증가할수록 위치명령값의 최대치를 제한하는데, EPS_POSITION_REDUCTION_FACTOR 가 2730 이면 6000rpm 일때 위치명령값을 0%까지 제한하는 것에 해당하며, EPS_POSITION_REDUCTION_MIN 이 최소 제한값임. 작동상태는 Qx2;로 전류치를, Qx1;으로 속도값을, Qx112;로 위치값을 볼수 있음.</p> <p>5th word 의 값이 16이면 모터구동전압이 10V 로부터 1V씩 떨어질때마다 10ms당 4(약 0.2A 에 해당함) 만큼의 크기로 전류리밋 값을 감소시킴.</p> <p>6th word 의 값이 2457이면 20A에 해당함. (3mOhm을 전류감지저항으로 사용하는 경우) 이 값보다 더큰 전류명령값이 계산되는 경우 전류상승을제한 로직을 적용함. 만약 0.5mOhm을 전류감지저항으로 사용하는 경우는 409 값이 20A에 해당함.</p> <p>7th word 의 값이 10이면 1ms당 81.4mA의 상승을 허용한다는 말임. (3mOhm을 전류감지저항으로 사용하는 경우) 만약 0.5mOhm. 전류감지저항으로 사용하는 경우는 2 값이 1ms당 97.7mA의 상승을 허용한다는 말임.</p> <p>파라미터는 EsA55A; 명령으로 EEPROM에 저장이 가능함.</p>

Qxddd1 에 의하여 전송되는 변수값 (EPS의 경우)

RJM_VER7 내장형 F/T 센서인 경우				
ddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
112	$(\text{position1_cmd} - \text{POSITION1_MID}) / (\text{MOTOR1_POSITION_SCALE_FACTOR} * 4)$	$(\text{position1_set} - \text{POSITION1_MID}) / (\text{MOTOR1_POSITION_SCALE_FACTOR} * 4)$	$(\text{position1} - \text{POSITION1_MID}) / (\text{MOTOR1_POSITION_SCALE_FACTOR} * 4)$	vehicle_driving_motor_speed

EPS 위치명령 조작 및 보호 작동

위치 명령값의 설정 및 제한 :

MSG_ID = 0x058 번으로 수신된 구동모터속도값 (0.1 rpm 단위, 수신된 세번째 바이트가 high-byte, 네번째 바이트가 low-byte),
MSG_ID = 0x035 번으로 수신된 조향각 명령값 (-1000 ~ +1000, 수신된 세번째 바이트가 high-byte, 네번째 바이트가 low-byte)에 대하여

$1000 - (\text{모터구동속도}(\text{rpm}) * \text{EPS_POSITION_REDUCTION_FACTOR}) / 16384$ 값에 대하여 EPS_POSITION_REDUCTION_MIN 보다 작지 않게 제한한 값을 상한선으로 설정한 다음, 조향각 명령값이 이 상한선보다 크지 않게 제한함.

그런 다음 EPS_CENTER_OFFSET을 더한 값을 최종적인 조향각 명령값으로 설정함. (EPS_CENTER_OFFSET의 단위와 0x035 번으로 수신된 조향각 명령값의 단위는 같으며, -1000 ~ +1000의 범위는 조향중립위치로부터 -100% ~ +100%의 변위를 나타냄)

그런 다음 이 변위값에 EPS_POSITION_SCALE_FACTOR 을 곱하고 5,000,000을 더하여 제어기 내부의 위치명령으로 설정함.
EPS_POSITION_SCALE_FACTOR 값은 85를 넘을수 없으며, EPS_CENTER_OFFSET 값이 큰 경우에는 EPS_POSITION_SCALE_FACTOR 값은 더 작아야 조향의 기구리밋에 닿지 않게됨.

전류상승률이 높을때의 전류명령값 제한 :

전류명령값 (제어기 내부에서 위치제어시에 발생시킨 상전류진폭의 명령값, 0.5mOhm을 사용하는 12V 제어기의 경우에 409 가 20A에 해당하며, 3mOhm을 사용하는 72V 제어기의 경우는 2457 이 20A에 해당함) 이 EPS_CURRENT_RISING_TRIP 보다 크면, 1ms 마다 EPS_CURRENT_RISING_INCREMENT 이상의 전류명령값 상승을 억제함.

3mOhm을 전류감지저항으로 사용하는 경우에 EPS_CURRENT_RISING_INCREMENT값이 10 이면 1s당 81.4A의 상승을 허용한다는 말이며, 0.5mOhm. 전류감지저항으로 사용하는 경우는 EPS_CURRENT_RISING_INCREMENT값이 2 이면 1s당 97.7A의 상승을 허용한다는 말임.

EPS 보호 작동

전압강하가 클때의 제어전류 상한값 제한 :

0.5mOhm을 사용하는 12V 제어기의 경우에만 적용되는 로직이며, EPS_CURRENT_CONTROL_LIMIT_UP_DOWN_RATE 값이 16이면 모터구동전압이 10V로부터 1V씩 떨어질때마다 10ms당 4(약 0.2A 에 해당함) 만큼의 크기로 전류리미트 값을 감소시킴.

```

if((UINT)EPS_CURRENT_CONTROL_LIMIT_UP_DOWN_RATE > 100) EPS_CURRENT_CONTROL_LIMIT_UP_DOWN_RATE = 16;
if((tempW7 = ((DC_BUS_VOLT_1V * 10) - vfilTA[10])) > 0) {
    if((fault1_status&0x2900)==0) {
        if(current1_cmd>=0) current1_control_limit_cnt = current1_control_limit - current1_cmd;
        else current1_control_limit_cnt = current1_control_limit + current1_cmd;
        current1_control_limit_cnt -= current1_control_limit >> 2;
        if(current1_control_limit_cnt < 0) current1_control_limit_cnt = 0;
    }
    if((fault1_status&0x2000)!=0) {
        tempU3 = ((UINT)tempW7 * EPS_CURRENT_CONTROL_LIMIT_UP_DOWN_RATE) >> 8;
        if(current1_control_limit_cnt < current1_control_limit) current1_control_limit_cnt += tempU3;
        if(current1_control_limit_cnt > current1_control_limit) current1_control_limit_cnt = current1_control_limit;
    }
    fault1_status |= 0x2000;
}
else if((fault1_status&0x2000)!=0) {
    tempU3 = ((UINT)(-tempW7) * EPS_CURRENT_CONTROL_LIMIT_UP_DOWN_RATE) >> 8;
    if(tempU2==1) tempU3 = 0;
    else if(tempU2==2) tempU3 = 0;
    if(current1_control_limit_cnt > tempU3) current1_control_limit_cnt -= tempU3;
    else {
        current1_control_limit_cnt = 0;
        fault1_status &= 0xdfff;
    }
}
}

```

EPS 오류코드

오류코드	오류코드의 의미	발생원인	조치
0x8000	모터제어기에 구동 불가능한 fault가 발생하였거나 정상 작동모드로 진입을 하지 못하고 있다는 fault code	모터제어기에 구동 불가능한 fault가 발생했을 경우	재부팅
		모터제어기에 구동 불가능한 fault는 없으나 정상작동모드로 진입하지 못했을 경우	기다림
0x4000	모터의 절대각센서에 의한 위치값과 incremental encoder에 의한 위치값의 차이가 2048이상임을 알리는 fault code	절대각 센서의 노이즈나 파손 등으로 인한 count 오류	system shutdown 하고 케이블 및 절대각 센서점검
		절대각 센서의 읽기 오류	단발성이면 무시
		기어의 파손이나 절대각 센서의 비선형 오차	기구 및 절대각센서 점검
0x2000	모터의 incremental encoder의 펄스 카운트 오차가 발생하여 Hall A/B/C 상 신호와 60도 이상의 위상차가 발생한 경우 (이 경우는 0x8000 오류 코드도 발생함)	incremental encoder의 노이즈 등으로 인한 count 오류	system shutdown 하고 케이블 및 incremental encoder 점검
		Hall A/B/C 센서의 읽기 오류	단발성이면 무시
0x1000	정상 작동모드로 진입을 하지 못하고 있다는 fault code	부팅중이어서 정상작동 모드로 진입하지 못했을 경우	기다림
		모터제어기에 구동 불가능한 fault가 발생했을 경우	재부팅
0x0800	과도하게 낮은 전압 상태가 되었음을 알리는 fault code	부하전류가 과도하게 흘러서 저전압 상태가 된 경우	Sw 명령어로 사용가능한 전류최대치를 줄임
		저전압발생 억제 관련 파라미터가 적절하지 않을 때	MB 명령어 마지막 두 파라미터를 작게 설정함
		외부 72V->12V DC-DC 변환기에 오류가 발생한 경우	DC-DC 변환기 점검 및 교체
0x0400	모터의 코일 온도를 추정한 값이 과도하게 큰 값을 알리는 fault code	부하전류가 과도하게 흘러서 모터의 온도가 과도하게 상승함으로 추정될 때	Sw 명령어로 사용가능한 전류최대치를 줄임
		모터 코일온도 추정관련 파라미터가 적절하지 않을 때	코일 온도추정 관련 ZQ 파라미터 수정
0x0200	모터제어기의 구동소자 온도를 추정한 값이 과도하게 큰 값을 알리는 fault code	부하전류가 과도하게 흘러서 모터제어기의 구동소자 온도가 과도하게 상승함으로 추정될 때	Sw 명령어로 사용가능한 전류최대치를 줄임
		모터 구동소자 온도 추정관련 파라미터가 적절하지 않을 때	관련 Z8/Z9 파라미터 수정

명령어	구분	설명
;MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7; (#) ;MB?; (#)	operation	자석가동자 제어기의 동작관련 파라미터를 설정함. 1st word : NUMBER_OF_S_POLE_MAGNETS [1 - 63] (19 개) 2nd word : START_CONTROL_MAGNET_SENSOR_CNT [1 - 384] (19 번) 3rd word : STOP_CONTROL_MAGNET_SENSOR_CNT [1 - 384] (17 번) 4th word : SWITCHOVER_MAGNET_SENSOR2_CNT [1 - 384] (4 번) 5th word : NUMBER_OF_POLE_PAIRS_BETWEEN_SENSOR12_A_PHASE [1 - 63] (18 pairs) 6th word : NUMBER_OF_POLE_PAIRS_BETWEEN_SENSOR11_A_PHASE [1 - 63] (19 pairs) 7th word : NS_MAGNET_OFFSET [0 - 600] (200 bit)
	return value	MBddd1,ddd2,ddd3,ddd4,ddd5,ddd6,ddd7;
	comment	자석가동자 제어기에 한함. START_CONTROL_MAGNET_SENSOR_CNT 는 삼상 wave sector의 번호로서 전류제어가 개시되는 위치의 번호임. START_CONTROL_MAGNET_SENSOR_CNT 는 삼상 wave sector의 번호로서 전류제어가 정지되는 위치의 번호임. SWITCHOVER_MAGNET_SENSOR2_CNT 는 -방향쪽 센서와 +방향쪽 센서의 사용이 교대되는 삼상 wave sector의 번호임. NUMBER_OF_POLE_PAIRS_BETWEEN_SENSOR12_A_PHASE 는 -방향쪽 센서와 +방향쪽 센서의 A상 신호검출 센서사이의 사잇거리를 NS 자석 one-pair 폭으로 나눈 값임. NUMBER_OF_POLE_PAIRS_BETWEEN_SENSOR11_A_PHASE 는 순위가 높은 제어기의 -방향쪽 센서와 순위가 낮은 제어기의 -방향쪽 센서의 A상 신호검출 센서 사잇거리를 NS 자석 one-pair 폭으로 나눈 값임. NS_MAGNET_OFFSET은 Hall A/B/C 센서를 A/D 변환하였을때 자석을 검출하지 않은 위치에서의 A/D 변환 값이 2200 + NS_MAGNET_OFFSET 보다 확실히 작다고 보장할수 있는 최소크기의 값임. EsA55A; 명령으로 EEPROM에 저장이 가능함.
;Mbddd1,11112; (#) ;Mb?; (#)	operation	자석가동자 제어기의 동작관련 파라미터를 설정함. 1st word : 삼상 아날로그 Hall 센서의 상전압 cross point 근방에서 smoothing하는 기능을 enable (1 일때), disable (0 일때). 전원투입시의 기본값은 enable(1) 임. 2nd long word : 제어가능범위내로서 전류명령값의 부드러운 천이를 적용하는 범위의 크기 [1 - 655360] (16384 단위위치값)
	return value	Mbddd1,11112;
	comment	자석가동자 제어기에 한함.

명령어	구분	설명
;MBddd1,ddd2; (+^) MB?; (+^)	operation	F/T 센서 drift offset 제거 동작관련 파라미터를 설정함. 1st word : FTdataBufSize [1 - 512] (300 word) 2nd word : FTcompensationThresh1 [1 - 10] (2 bit)
	return value	MBddd1,ddd2;
	comment	F/T 센서에 한함. FTdataBufSize 만큼의 A/D 변환 데이터를 ring buffering 하면서 FTcompensationThresh1 값을 넘는 변화가 있는지를 사용하여 drift offset 제거 동작을 실시함. EsA55A; 명령으로 EEPROM에 저장이 가능함.

명령어	구분	설명
(BLDC 모터의 경우) ;MBdddd1, dddd2, dddd3, dddd4, dddd5, dddd6; (+#^) ;Mb?; (+#^)	operation	<p>엔코더를 사용하지 않고 digital Hall IC만을 사용하는 BLDC 모터1의 경우, Hall 센서 A/B/C 상신호의 에지위치를 설정함.</p> <p>1st word : Hall A상 상승에지의 위치값 [00000] (00000 bit) 2nd word : Hall A상 상승에지로부터 C상 하강에지까지의 위치값 [1 - 65535] (2731 bit) 3rd word : Hall A상 상승에지로부터 B상 상승에지까지의 위치값 [1 - 65535] (5462 bit) 4th word : Hall A상 상승에지로부터 A상 하강에지까지의 위치값 [1 - 65535] (8193 bit) 5th word : Hall A상 상승에지로부터 C상 상승에지까지의 위치값 [1 - 65535] (10924 bit) 6th word : Hall A상 상승에지로부터 B상 하강에지까지의 위치값 [1 - 65535] (13655 bit)</p>
	return value	MBdddd1, dddd2, dddd3, dddd4, dddd5, dddd6;
	comment	<p>에지간 간격이 모두 같은 이상적인 경우, 1st~6th word의 값은 00000, 02731, 05462, 08193, 10924, 13655이며, 이때 1st~6th word의 값의 사이간격은 2731임. 모터가 +방향으로 회전할때, 1st~6th word의 값은 Hall A상신호의 상승에지를 기준으로하여 이어진 Hall A/B/C 상신호 에지들의 위치를 숫자화한 것으로서, A상신호의 상승에지를 기준으로하여 A상신호의 다음 상승에지 위치가 16384 값에 해당하도록 스케일링한 것임. 구동을 하지 않고 +방향으로 무부하상태로 일정속도로 돌렸을때, Mb?; 명령에 대한 리턴값을 사용하여, 덧셈 및 스케일링을 한 값으로 설정하면 됨. 단 속도값이 일정하지 않다면 Qx5; 모니터하여 A상 상승에지 후 첫번째 펄스폭과 곧이은 A상 상승에지후의 첫번째 펄스폭 크기의 비율을 사용하여 감속된 속도 계산하고 이를 감안하여 다시 스케일링하여야함. EsA55A; 명령으로 EEPROM에 저장이 가능함.</p>
(BLDC 모터의 경우) ;Mb?; (+#^)	operation	analog Hall 센서를 사용하지 않는 BLDC 모터1의 경우 Hall 센서신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
	return value	<p>Mbdddd1, dddd2, dddd3, dddd4, dddd5, dddd6;</p> <p>1st word : Hall A상 상승에지로부터 C상 하강에지까지의 시간폭 (bit) 2nd word : Hall C상 하강에지로부터 B상 상승에지까지의 시간폭 (bit) 3rd word : Hall B상 상승에지로부터 A상 하강에지까지의 시간폭 (bit) 4th word : Hall A상 하강에지로부터 C상 상승에지까지의 시간폭 (bit) 5th word : Hall C상 상승에지로부터 B상 하강에지까지의 시간폭 (bit) 6th word : Hall B상 하강에지로부터 A상 상승에지까지의 시간폭 (bit)</p>
	comment	PWM 신호의 한펄스 주기가 시간폭의 1 bit 에 해당함. 단 2채널 H/W 버전의 경우는 PWM 신호의 두펄스 주기가 시간폭의 1 bit 에 해당함.

명령어	구분	설명
(BLDC 모터의 경우) ;MEdddd1, dddd2, d ddd3, dddd4, dddd 5, dddd6; (+#^) ;ME?; (+#^)	operation	<p>엔코더를 사용하지 않고 digital Hall IC만을 사용하는 BLDC 모터2의 경우, Hall 센서 A/B/C 상신호의 에지위치를 설정함.</p> <p>1st word : Hall A상 상승에지의 위치값 [00000] (00000 bit) 2nd word : Hall A상 상승에지로부터 C상 하강에지까지의 위치값 [1 - 65535] (2731 bit) 3rd word : Hall A상 상승에지로부터 B상 상승에지까지의 위치값 [1 - 65535] (5462 bit) 4th word : Hall A상 상승에지로부터 A상 하강에지까지의 위치값 [1 - 65535] (8193 bit) 5th word : Hall A상 상승에지로부터 C상 상승에지까지의 위치값 [1 - 65535] (10924 bit) 6th word : Hall A상 상승에지로부터 B상 하강에지까지의 위치값 [1 - 65535] (13655 bit)</p>
	return value	MEdddd1, dddd2, dddd3, dddd4, dddd5, dddd6;
	comment	<p>에지간 간격이 모두 같은 이상적인 경우, 1st~6th word의 값은 00000, 02731, 05462, 08193, 10924, 13655이며, 이때 1st~6th word의 값의 사이간격은 2731임. 모터가 +방향으로 회전할때, 1st~6th word의 값은 Hall A상신호의 상승에지를 기준으로하여 이어진 Hall A/B/C 상신호 에지들의 위치를 숫자화한 것으로서, A상신호의 상승에지를 기준으로하여 A상신호의 다음 상승에지 위치가 16384 값에 해당하도록 스케일링한 것임. 구동을 하지 않고 +방향으로 무부하상태로 일정속도로 돌렸을때, Mb?; 명령에 대한 리턴값을 사용하여, 덧셈 및 스케일링을 한 값으로 설정하면 됨. 단 속도값이 일정하지 않다면 Qx5; 모니터하여 A상 상승에지 후 첫번째 펄스폭과 곧이은 A상 상승에지후의 첫번째 펄스폭 크기의 비율을 사용하여 감속된 속도 계산하고 이를 감안하여 다시 스케일링하여야함. EsA55A; 명령으로 EEPROM에 저장이 가능함.</p>
(BLDC 모터의 경우) ;Me?; (+#^)	operation	analog Hall 센서를 사용하지 않는 BLDC 모터2의 경우 Hall 센서신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
	return value	<p>Medddd1, dddd2, dddd3, dddd4, dddd5, dddd6;</p> <p>1st word : Hall A상 상승에지로부터 C상 하강에지까지의 시간폭 (bit) 2nd word : Hall C상 하강에지로부터 B상 상승에지까지의 시간폭 (bit) 3rd word : Hall B상 상승에지로부터 A상 하강에지까지의 시간폭 (bit) 4th word : Hall A상 하강에지로부터 C상 상승에지까지의 시간폭 (bit) 5th word : Hall C상 상승에지로부터 B상 하강에지까지의 시간폭 (bit) 6th word : Hall B상 하강에지로부터 A상 상승에지까지의 시간폭 (bit)</p>
	comment	PWM 신호의 한펄스 주기가 시간폭의 1 bit 에 해당함. 단 2채널 H/W 버전의 경우는 PWM 신호의 두펄스 주기가 시간폭의 1 bit 에 해당함.

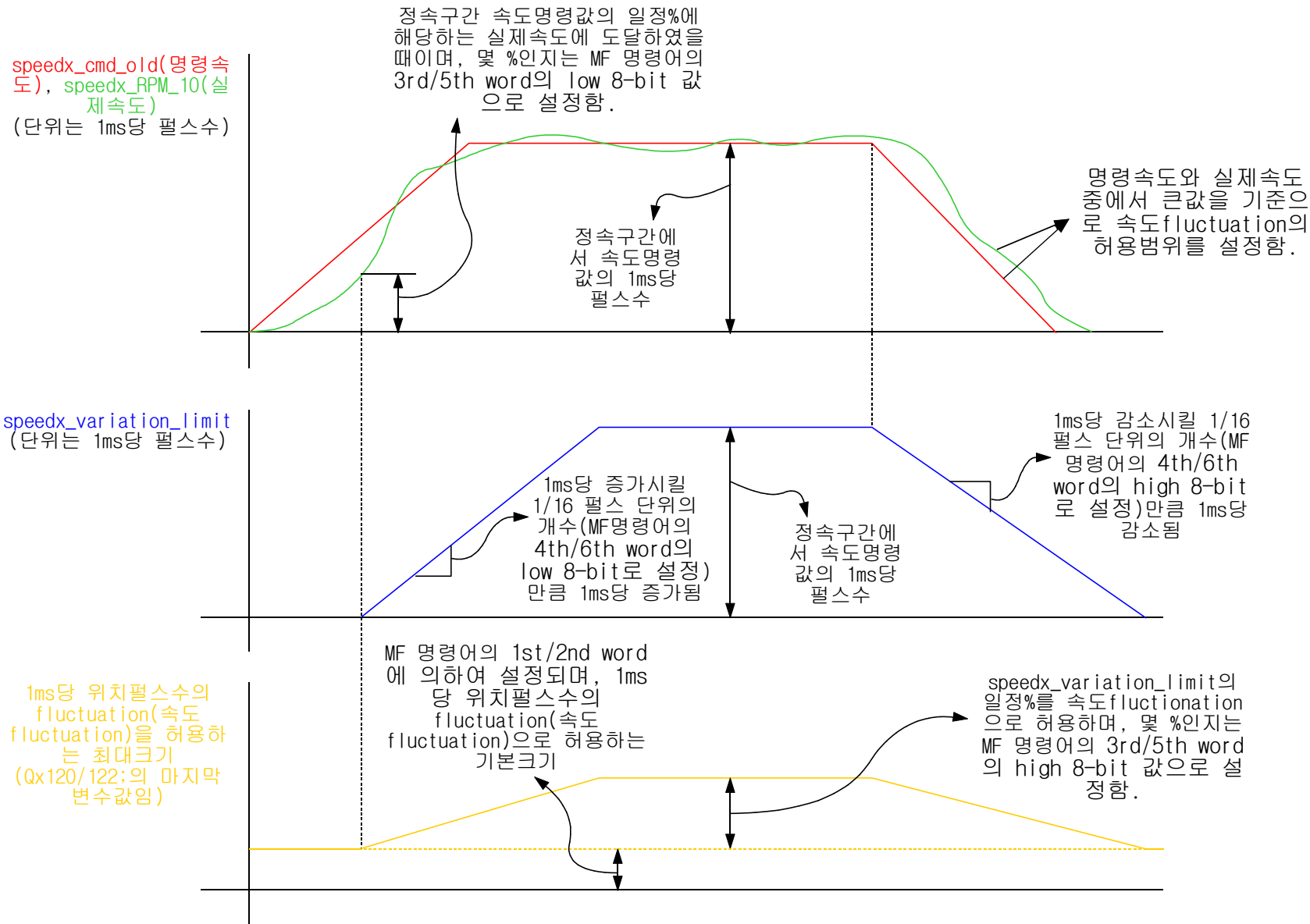
명령어	구분	설명
(low resolution encoder의 DC 모터인 경우) ;MBdddd1,dddd2,dddd3,dddd4; (+#^) ;Mb?; (+#^)	operation	low resolution encoder를 사용하는 DC 모터1의 경우, 엔코더신호 A/B 상신호의 에지위치를 설정함. 1st word : Hall A상 상승에지의 위치값 [00000] (00000 bit) 2nd word : Hall A상 상승에지로부터 B상 상승에지까지의 위치값 [1 - 65535] (4096 bit) 3rd word : Hall A상 상승에지로부터 A상 하강에지까지의 위치값 [1 - 65535] (8192 bit) 4th word : Hall A상 상승에지로부터 B상 하강에지까지의 위치값 [1 - 65535] (12288 bit)
	return value	MBdddd1,dddd2,dddd3,dddd4;
	comment	에지간 간격이 모두 같은 이상적인 경우, 1st~4th word의 값은 00000,4096,8192,12288이며, 이때 1st~4th word의 값의 사이간격은 4096임. 모터가 +방향으로 회전할때, 1st~4th word의 값은 엔코더 A상신호의 상승에지를 기준으로하여 이어진 엔코더 A/B 상신호 에지들의 위치를 숫자화한 것으로서, A상신호의 상승에지를 기준으로하여 A상신호의 다음 상승에지 위치가 16384 값에 해당하도록 스케일링한 것임. 구동을 하지 않고 +방향으로 무부하상태로 일정속도로 돌렸을때, Mb?; 명령에 대한 리턴값을 사용하여, 덧셈 및 스케일링을 한 값으로 설정하면 됨. 단 속도값이 일정하지 않다면 Qx5;를 모니터하여 A상 상승에지후 첫번째 펄스폭과 끝이은 A상 상승에지후의 첫번째 펄스폭 크기의 비율을 사용하여 감속된 속도 계산하고 이를 감안하여 다시 스케일링하여야함. EsA55A; 명령으로 EEPROM에 저장이 가능함.
(low resolution encoder의 DC 모터인 경우) ;Mb?; (+#^)	operation	low resolution encoder를 사용하는 DC 모터1의 경우 엔코더신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
	return value	Mbdddd1,dddd2,dddd3,dddd4; 1st word : Hall A상 상승에지로부터 B상 상승에지까지의 시간폭 (bit) 2nd word : Hall B상 상승에지로부터 A상 하강에지까지의 시간폭 (bit) 3rd word : Hall A상 하강에지로부터 B상 하강에지까지의 시간폭 (bit) 4th word : Hall B상 하강에지로부터 A상 상승에지까지의 시간폭 (bit)
	comment	PWM 신호의 한펄스 주기가 시간폭의 1 bit 에 해당함.

명령어	구분	설명
(low resolution encoder의 DC 모터인 경우) ;MEdddd1,dddd2,dddd3,dddd4; (+#^) ;ME?; (+#^)	operation	low resolution encoder를 사용하는 DC 모터2의 경우, 엔코더신호 A/B 상신호의 에지위치를 설정함. 1st word : Hall A상 상승에지의 위치값 [00000] (00000 bit) 2nd word : Hall A상 상승에지로부터 B상 상승에지까지의 위치값 [1 - 65535] (4096 bit) 3rd word : Hall A상 상승에지로부터 A상 하강에지까지의 위치값 [1 - 65535] (8192 bit) 4th word : Hall A상 상승에지로부터 B상 하강에지까지의 위치값 [1 - 65535] (12288 bit)
	return value	MEdddd1,dddd2,dddd3,dddd4;
	comment	에지간 간격이 모두 같은 이상적인 경우, 1st~4th word의 값은 00000,4096,8192,12288이며, 이때 1st~4th word의 값의 사이간격은 4096임. 모터가 +방향으로 회전할때, 1st~4th word의 값은 엔코더 A상신호의 상승에지를 기준으로하여 이어진 엔코더 A/B 상신호 에지들의 위치를 숫자화한 것으로서, A상신호의 상승에지를 기준으로하여 A상신호의 다음 상승에지 위치가 16384 값에 해당하도록 스케일링한 것임. 구동을 하지 않고 +방향으로 무부하상태로 일정속도로 돌렸을때, Me?; 명령에 대한 리턴값을 사용하여, 덧셈 및 스케일링을 한 값으로 설정하면 됨. 단 속도값이 일정하지 않다면 Qx6;를 모니터하여 A상 상승에지후 첫번째 펄스폭과 끝이은 A상 상승에지후의 첫번째 펄스폭 크기의 비율을 사용하여 감속된 속도 계산하고 이를 감안하여 다시 스케일링하여야함. EsA55A; 명령으로 EEPROM에 저장이 가능함.
(low resolution encoder의 DC 모터인 경우) ;Me?; (+#^)	operation	low resolution encoder를 사용하는 DC 모터2의 경우 엔코더신호의 에지간 간격값을 읽어서 상위제어기로 전송함.
	return value	Medddd1,dddd2,dddd3,dddd4; 1st word : Hall A상 상승에지로부터 B상 상승에지까지의 시간폭 (bit) 2nd word : Hall B상 상승에지로부터 A상 하강에지까지의 시간폭 (bit) 3rd word : Hall A상 하강에지로부터 B상 하강에지까지의 시간폭 (bit) 4th word : Hall B상 하강에지로부터 A상 상승에지까지의 시간폭 (bit)
	comment	PWM 신호의 한펄스 주기가 시간폭의 1 bit 에 해당함.

명령어	구분	설명
(low resolution encoder의 DC 모터 또는 BLDC 모터인 경우) ;MFddd1,ddd2; (2013년 2월 9일 이전 S/W 버전) (+#^) ;MFHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (2013년 2월 9일 S/W 버전부터) ;MF?; (+#^)	operation	low resolution encoder를 사용하는 DC 모터 또는 BLDC 모터인 경우에 속도신호의 변화를 제한하는 범위를 설정함. 1st word : 모터1의 Ss 또는 Sa 명령에 따른 가속도 설정치에 대한 실제 가속도의 제한 배율 [2 - 2000] (100 배) 2013년 2월 9일 S/W 버전부터는 [0002 - 07D0] (0064 배) 2nd word : 모터2의 Ss 또는 Sa 명령에 따른 가속도 설정치에 대한 실제 가속도의 제한 배율 [2 - 2000] (100 배) 2013년 2월 9일 S/W 버전부터는 [0002 - 07D0] (0064 배) 2013년 2월 9일 S/W 버전부터는 3rd word의 low 8-bit : 모터1의 속도변화 제한을 완화하기 시작하는 속도의 명령속도 크기에 대한 %값 [00 - 64] (00 %) 3rd word의 high 8-bit : 모터1의 속도변화 제한을 완화하는 속도변화량 크기의 명령속도 크기에 대한 상대값 [00 - FF] (00 100/256%) 4th word의 low 8-bit : 모터1의 가속/정속시의 속도변화 제한을 완화하는 시간 기울기 [00 - FF] (00 1/16배) 4th word의 high 8-bit : 모터1의 감속/정지시의 속도변화 제한을 완화하는 시간 기울기 [00 - FF] (00 1/16배) 5th word의 low 8-bit : 모터2의 속도변화 제한을 완화하기 시작하는 속도의 명령속도에 대한 %값 [00 - 64] (00 %) 5th word의 high 8-bit : 모터2의 속도변화 제한을 완화하는 크기의 명령속도에 대한 상대값 [00 - FF] (00 100/256%) 6th word의 low 8-bit : 모터2의 가속/정속시의 속도변화 제한을 완화하는 시간 기울기 [00 - FF] (00 1/16배) 6th word의 high 8-bit : 모터2의 감속/정지시의 속도변화 제한을 완화하는 시간 기울기 [00 - FF] (00 1/16배)
	return value	MFddd1,ddd2; (2013년 2월 9일 이전 S/W 버전) MFHLHL,HLHL,HLHL,HLHL,HLHL,HLHL; (2013년 2월 9일 S/W 버전부터)
	comment	

명령어	구분	설명
	operation	<p>low resolution encoder를 사용하는 DC 모터 또는 digital Hall IC만을 사용하는 BLDC 모터인 경우에 실제 속도를 계산할때 일정 크기 이상의 속도 변화를 허용하지 않는데, Ss 또는 Sa 명령에 따른 가속도 설정치의 몇배까지만 허용하는지를 설정하는 명령임.</p> <p>위치제어일 경우에는 Ss 명령어로 설정된 가속도를 기준으로 하며, 속도제어일 경우에는 Sa 명령어로 설정된 가속도를 기준으로하여 제한 배율을 적용함.</p> <p>p 명령을 사용하여 실시간 제어를 하는 경우에는 제어중 있을수 있는 최대 가속도 값을 Ss 또는 Sa 명령으로 설정하여야함.</p> <p>가속도의 제한 배율값이 너무 적으면, 엔코더 또는 Hall 센서신호의 에지간 간격이 균일하지 않은 경우에 속도 측정 오차가 커짐. 이를 방지하려면 MB/ME 명령어를 사용하여 보전을 해주어야함.</p> <p>가속도의 제한 배율값을 모터제어기에서 적용할때는 Ss 또는 Sa 명령에 따른 가속도 설정치가 클수록 가속도의 제한 배율값이 작게 적용되도록 하고 있음. 이 때 최소값은 2이며, 정지시에는 4가 넘지 않게 제한함.</p> <p>2013년 2월 9일 S/W 버전부터는 예를 들어서</p> <p>3rd/5th word의 low 8-bit 값이 1E 이면 명령속도의 30%(16진수 1E는 10진수로 30임)에서부터 속도변화 제한을 완화하기 시작함.</p> <p>3rd/5th word의 high 8-bit 값이 40 이면 명령속도의 25%(16진수 40는 10진수로 64이어서 $64 \cdot 100 / 256 = 25\%$ 임) 만큼의 속도변화의 완화를 허용함.</p> <p>4th/6th word의 low 8-bit 값이 04 이면 가속/정속시에 4/16배로 느리게 속도변화 제한의 완화가 이루어지게 함.</p> <p>4th/6th word의 high 8-bit 값이 10 이면 감속/정지시 16/16배(16진수 10는 10진수로 16임)로 느리게 속도변화 제한의 완화가 이루어지게 함.</p> <p>Qx120/121/122/123; 을 이용하여 속도신호변화가 제한되는 상황을 볼수 있음.</p> <p>EsA55A; 명령으로 EEPROM에 저장이 가능함.</p>

low resolution encoder 또는 digital Hall 센서만을 사용하는 경우의 속도 계산



low resolution encoder 또는 digital Hall 센서만을 사용하는 경우의 속도 계산

RJML_VER7인 경우				
dddd1 값	첫번째 변수	두번째 변수	세번째 변수	네번째 변수
120	모터1의 순간속도명령값 단위크기는 RPM임	모터1의 실제속도값 단위크기는 RPM임	모터1의 속도에 따른 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터1의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임
121	모터1의 순간속도명령값 단위크기는 RPM임	모터1의 속도계산에 사용되는 1ms동안의 위치변화량 단위크기는 1ms당 펄스수임	모터1의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터1의 속도계산에 사용되는 1ms동안의 위치변화량을 제한한 값 (이값으로 속도계산을 실시함) 단위크기는 1ms당 펄스수임
122	모터2의 순간속도명령값 단위크기는 RPM임	모터2의 실제속도값 단위크기는 RPM임	모터2의 속도에 따른 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터2의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임
123	모터2의 순간속도명령값 단위크기는 RPM임	모터2의 속도계산에 사용되는 1ms동안의 위치변화량 단위크기는 1ms당 펄스수임	모터2의 총 속도변화 허용크기 단위크기는 1ms당 펄스수임	모터2의 속도계산에 사용되는 1ms동안의 위치변화량을 제한한 값 (이값으로 속도계산을 실시함) 단위크기는 1ms당 펄스수임

명령어	구분	설명
;VH; (+#^)	operation	현재의 각위치를 홈위치로 설정함.
	return value	VHiiii1,iiii2; iiii1 : panning 방향의 홈위치에서의 각위치값 iiii2 : tilting 방향의 홈위치에서의 각위치값
	comment	한바퀴의 각위치값은 0~16383 범위의 값을 가짐. analog 포텐시오메터나 digital 포텐시오메터를 사용하는 경우와 인공안구에서만 적용가능한 명령임. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
;VOiiii1,iiii2; (+#^)	operation	홈위치의 각위치값을 설정함. iiii1 : panning 방향의 홈위치에서의 각위치값 [0 - 16383] iiii2 : tilting 방향의 홈위치에서의 각위치값 [0 - 16383]
	return value	VOiiii1,iiii2;
	comment	analog 포텐시오메터나 digital 포텐시오메터를 사용하는 경우와 인공안구에서만 적용가능한 명령임. 설정된 값은 EsA55A; 명령어로 EEPROM에 저장될 수 있음.
;VP; (+#^)	operation	현재의 각위치값을 읽어내어 host로 전송함.
	return value	VPiiii1,iiii2; iiii1 : panning 방향의 현재 각위치값 iiii2 : tilting 방향의 현재 각위치값
	comment	analog 포텐시오메터나 digital 포텐시오메터를 사용하는 경우와 인공안구에서만 적용가능한 명령임.
;VAiiii1,iiii2; (+#^)	operation	각위치값을 설정하여 설정 각위치 위치로 이동시킴. iiii1 : panning 방향의 각위치 설정값 [-1365 ~ 1365] iiii2 : tilting 방향의 각위치 설정값 [-1365 ~ 1365]
	return value	VAiiii1,iiii2;
	comment	인공안구에서만 적용가능한 명령임.

명령어	구분	설명
;VTAAAAA.....AA AA; (+#^)	operation	영상안정화 PCB로 명령어를 전송하고 결과값을 수신함. AAAAA.....AAAA : 영상안정화 PCB로 전송될 명령어
	return value	VTAAAAA.....AAAA; AAAAA.....AAAA : 영상안정화 PCB에서 리턴된 값
	comment	인공안구에서만 적용가능한 명령임.
;1; (+#^) ;2; (+#^) ;3; (+#^) ;4; (+#^) ;6; (+#^) ;7; (+#^) ;8; (+#^) ;9; (+#^)	operation	일정시간동안 panning 및 tilting 방향으로 각위치를 일으킴. 1,2,3 : 안구를 아래로 숙임. 7,8,9 : 안구를 위로 올림. 1,4,7 : 안구를 좌로 움직임. 3,6,9 : 안구를 우로 움직임.
	return value	1; 2; 3; 4; 6; 7; 8; 9;
	comment	인공안구에서만 적용가능한 명령임.

명령어	구분	설명
;ERHLHL,HLHL; (+#^%)	operation	<p>파라미터 값을 EEPROM으로부터 읽어내어 DSP SARAM에 저장함. (F2811의 경우는 0x3F8200 - 0x3F89FF, F28334의 경우는 0xE200 - 0xE9FF 번지에 저장함.)</p> <p>1st word : 파라미터를 EEPROM으로부터 읽어서 저장하는 SARAM의 시작주소 [0x0A00 - 0x0FFF/0x11FF] 2nd word : word 수 [0x0001 - 0x0600/0x0800]</p>
	return value	ERHLHL,HLHL;
	comment	<p>EEPROM의 0번지(low byte)/1번지(high byte) 데이터가 SARAM의 0x0A00/0x3F8200/0xE200 번지로 읽혀져 저장됨.</p> <p>마지막에 세미콜론 대신 콜론이 echo되면 시작주소나 word수가 not valid인 경우임.</p> <p>F2811 또는 F28334를 사용할때 25LC320/320A인 경우 2K word까지 사용이 가능하며, 시작주소/word 수가 0x11FF/0x0800 으로 늘어남.</p> <p>2014년 3월 20일 S/W 버전부터 Sleep_10micro(1)을 Sleep_1micro(1)로 바꿈으로서 처리속도를 높였음.</p>
;ESHLHL,HLHL; (+#^%)	operation	<p>DSP SRAM에 있는 파라미터 값을 page단위(8 word 또는 16 byte)로 EEPROM에 저장함. (F2811의 경우는 0x3F8200 - 0x3F89FF, F28334의 경우는 0xE200 - 0xE9FF 번지에 저장함)</p> <p>1st word : 파라미터를 EEPROM으로 저장하는 SARAM의 시작주소 [0x0A00 - 0x0FF8/0x11FF] 2nd word : word 수 [0x0001 - 0x0600/0x0800]</p>
	return value	ESHLHL,HLHL;
	comment	<p>파라미터를 읽어내는 시작주소는 8 또는 16의 배수이어야함. (25LC320/25LC320A인 경우 16)</p> <p>word수도 8또는 16의 배수이어야함. (25LC320/25LC320A인 경우 16)</p> <p>SARAM의 0x0A00/0x3F8200/0xE200 번지 데이터가 EEPROM의 0번지(low byte)/1번지(high byte)에 저장됨.</p> <p>마지막에 세미콜론 대신 콜론이 echo되면 시작주소나 word수가 not valid인 경우이거나, 저장하고 나서 다시 읽어낸 값이 다른 경우임.</p> <p>F2811 또는 F28334를 사용할때 25LC320/320A인 경우 2K word까지 사용이 가능하며, 시작주소/word 수가 0x11FF/0x0800 으로 늘어남.</p> <p>2014년 3월 20일 S/W 버전부터 Sleep_10micro(1)을 Sleep_1micro(1)로 바꿈으로서 처리속도를 높였음.</p>

명령어	구분	설명
;EsA55A; (+#^%)	operation	DSP SRAM에 있는 작동 파라미터 값을 한꺼번에 EEPROM에 저장함.
	return value	EsA55A;
	comment	F2811의 경우는 ESOA00,01C0; 을 수행한 것과 같으며, F28334의 경우는 ESOA00,0200; 을 수행한 것과 같음. 2014년 3월 20일 S/W 버전부터 Sleep_10micro(1)을 Sleep_1micro(1)로 바꿈으로서 처리속도를 높였음.
;EDA55A; (+#^%)	operation	DSP SRAM에 있는 파라미터 값을 default 값으로 바꿈. 이때 ST 명령어로 선정된 motor type에 따라서 파라미터의 기본값이 설정됨.
	return value	EDA55A;
	comment	EEPROM에 저장을 하지는 않음. F2811 또는 F28334의 경우는 EDA55A;수행시에 제어기주소가 정상이면 제어기주소/호스트주소 /operation_mode_SWITCH/DUMMY_MODE/SEQUENCE_CONTROL_MODE 값을 바꾸지 아니함. 2014년 3월 9일 이전 S/W 버전에서는 파라미터 초기화를 하면 MOTORx_BIT2AMP_SCALE_FACTOR가 잘못되는 설정되는 문제가 있었음. 2014년 6월 14일 ~ 2014년 9월 12일 S/W 버전에서는 CAN 모드로 설정되어 있는 경우 120 Ohm pull-up 저항이 연결되지 않은 상태에서 EDA55A; 명령을 사용하면 제어기가 리셋됨.

명령어	구분	설명
;EC?; (+#^%)	operation	RS232, RS485, CAN 통신과정에서 어떤 에러가 발생하였는지를 읽어서 host로 전송함.
	return value	<p>ECHLHL, HLHL, HLHL;</p> <p>1st word : SCIA RS232 통신에러 bit2 : SCIA parity-error flag. bit3 : SCIA overrun-error flag. bit4 : SCIA framing-error flag. (an expected stop bit is not found.) bit5 : SCIA break-detect flag. (a break condition(ten continuous low bits) occurs.) bit7 : SCIA receiver-error flag.</p> <p>2nd word : CAN 통신에러 bit0 : At least one of the error counters reached the warning level of 96. bit1 : The CAN module is in error-passive mode. bit2 : There is an abnormal rate of error occurrences on the CAN bus. bit3 : The CAN module did not receive an acknowledge. bit4 : The stuff bit rule was violated. bit5 : The CAN module received a wrong CRC. bit6 : The CAN module did not detect a recessive bit. bit7 : The received bit does not match the transmitted bit outside of the arbitration field; or during transmission of the arbitration field, a dominant bit was sent but a recessive bit was received. bit8 : A Form Error occurred on the bus.</p>

명령어	구분	설명
	return value	3rd word : SCIB RS232 통신에러 (F2811/F28334의 경우에만 해당됨) bit2 : SCIB parity-error flag. bit3 : SCIB overrun-error flag. bit4 : SCIB framing-error flag. (an expected stop bit is not found.) bit5 : SCIB break-detect flag. (a break condition(ten continuous low bits) occurs.) bit7 : SCIB receiver-error flag.
	comment	가장 최근에 있었던 RS232, RS485, CAN 통신에러상태를 읽어냄. 통신에러가 발생하면 자동 복구되며, 이때의 에러상태를 저장하고 있다가, EC?의 명령에 따라 저장되었던 값을 보내줌. 통신 장치를 무조건 초기화하려면 ECR; 명령어를 사용함.
;ECR; (+#^%)	operation	RS232, RS485, CAN 통신에러 플래그를 없애고 에러를 복구함.
	return value	ECR;
	comment	
;EcS; (+#^)	operation	RS485 또는 CAN을 통하여 모터제어기 주소 254(0xFE)로 전송되어오는 broadcasting 데이터를 수신하도록 설정함.
	return value	EcS;
	comment	전원을 켜올때의 기본상태는 수신하지 않은 것임. 2012년 6월 03일 S/W 버전부터 사용가능한 명령임.
;EcR; (+#^)	operation	RS485 또는 CAN을 통하여 모터제어기 주소 254(0xFE)로 전송되어오는 broadcasting 데이터를 수신하지 못하도록 설정함.
	return value	EcR;
	comment	전원을 켜올때의 기본상태는 수신하지 않은 것임. 브레이크제어기나 조향제어기에는 CAN 통신일때 선택적 broadcasting 기능이 해당 없음. 2012년 6월 03일 S/W 버전부터 사용가능한 명령임.

명령어	구분	설명
;EUHLHL,HLHL,HLHL; (+#^) ;EU?: (+#^)	operation	<p>CAN 통신을 사용할 때, 일정주기로 상위제어기에 제어기 상태값을 전송하는 기능의 파라미터를 설정함.</p> <p>1st word : 하위 8bit : 모터1의 상태값을 전송하는 전송시점 또는 주기 [0 - 10] (0 번) 상위 8bit : 모터2의 상태값을 전송하는 전송시점 또는 주기 [0 - 10] (0 번) 값이 0 - 9 범위 일때는 전송주기가 10ms 이며, 숫자는 제어기내의 1ms 단위의 순서를 의미함. 값이 10이상이면 전송주기는 1ms 임.</p> <p>2nd word : 모터1의 상태값을 전송하는 주소 (기본값 0x00FD) 3rd word : 모터2의 상태값을 전송하는 주소 (기본값 0x00FC)</p> <p>: 주소값 [0000 - 00FF] (FF00) - SX 명령어 두번째파라미터의 bit15/14 = 11 일때 : 주소값 [0000 - 07FF] (0000) - SX 명령어 두번째파라미터의 bit15/14 = 10 일때 : 주소값 [0000 - FFFF] (0000) - SX 명령어 두번째파라미터의 bit15/14 = 01 일때 : 주소값 [0000 - 00FF] (FF00) - SX 명령어 두번째파라미터의 bit15/14 = 00 일때</p>
	return value	EUHLHL,HLHL,HLHL;
	comment	<p>⚡ SX 명령어 두번째파라미터의 bit27 = 0 일때는 모터1의 상태값이 상위제어기로 전송되며, SX 명령어 세번째파라미터의 bit2 = 0 일때는 모터2의 상태값이 상위제어기로 전송됨. 두값 모두 같은 시점에서 전송이 가능함. 데이터의 내용은 SX 명령어의 설명 바로 다음에 제시되어 있음. 전송되는 데이터의 내용은 Eu, Ev 명령어로 바꿀 수 있음 (2014년 5월 25일 S/W 버전부터).</p> <p>⚡ SX 명령어 첫번째파라미터의 bit0 = 0 일때, 즉 CAN 통신모드로 설정되었을 때만 사용가능하며, SX 명령어 두번째파라미터의 bit15/14 값에 따라서 CAN 통신의 주소사용방법이 달라짐.</p> <p>bit15/14 = 11 일때 11-bit 주소모드로서 주소로 0-255만 사용 = 10 일때 11-bit 주소모드로서 주소로 0-2047 사용, 2012년 10월 16일부터 적용 = 01 일때 29-bit 주소모드로서 주소로 0-65535만 사용 (CAN2.0B), 2012년 10월 16일부터 적용 = 00 일때 11-bit 주소모드로서 주소로 0-255만 사용</p> <p>⚡ 주소값으로서 254(0xFE) 및 255(0xFF)는 broadcasting용으로 사용하는 주소이므로 126(0x8E), 127(0x8F), 254(0xFE) 및 255(0xFF)는 host의 주소로 사용하면 안됨.</p> <p>EsA55A; 명령으로 EEPROM에 저장이 가능함.</p> <p>2013년 11월 26일 S/W 버전부터 사용가능한 명령임.</p>

명령어	구분	설명
;EuHLHL,HLHL,HLHL,HLHL; (+#^) ;Eu?; (+#^)	operation	CAN 통신을 사용하여 일정주기로 상위제어기에 모터1의 상태값을 전송할 때, 전송할 모터1의 상태값 주소를 설정함. 1st word : 모터1의 상태값1의 주소 (기본값 0x0000) 2nd word : 모터1의 상태값2의 주소 (기본값 0x0000) 3rd word : 모터1의 상태값3의 주소 (기본값 0x0000) 4th word : 모터1의 상태값4의 주소 (기본값 0x0000)
	return value	EuHLHL,HLHL,HLHL,HLHL;
	comment	☞ Eu 명령에 의한 주소값이 모두 0x0000 이면, CAN 통신을 사용하여 일정주기로 상위제어기에 전송되는 값의 형식은 SX 명령어의 설명 바로 다음에 기술되어 있음. ☞ Eu 명령에 의한 주소값 중에서 0x0000 이 아니고 정당한 주소값이 있으면, 해당 주소에 있는 값 2바이트로 대체되어 상위제어기로 전송됨. ☞ Eu 명령에 의한 주소값이 정당한 주소값이 아니면 설정 주소값은 0x0000 으로 강제 설정됨. 2014년 5월 25일 S/W 버전부터 사용가능한 명령임.
;EvHLHL,HLHL,HLHL,HLHL; (+#^) ;Ev?; (+#^)	operation	CAN 통신을 사용하여 일정주기로 상위제어기에 모터2의 상태값을 전송할 때, 전송할 모터2의 상태값 주소를 설정함. 1st word : 모터2의 상태값1의 주소 (기본값 0x0000) 2nd word : 모터2의 상태값2의 주소 (기본값 0x0000) 3rd word : 모터2의 상태값3의 주소 (기본값 0x0000) 4th word : 모터2의 상태값4의 주소 (기본값 0x0000)
	return value	EvHLHL,HLHL,HLHL,HLHL;
	comment	☞ Ev 명령에 의한 주소값이 모두 0x0000 이면, CAN 통신을 사용하여 일정주기로 상위제어기에 전송되는 값의 형식은 SX 명령어의 설명 바로 다음에 기술되어 있음. ☞ Ev 명령에 의한 주소값 중에서 0x0000 이 아니고 정당한 주소값이 있으면, 해당 주소에 있는 값 2바이트로 대체되어 상위제어기로 전송됨. ☞ Ev 명령에 의한 주소값이 정당한 주소값이 아니면 설정 주소값은 0x0000 으로 강제 설정됨. ☞ 1채널 모터 제어기에서도 유효하며, 이 경우 모터2의 데이터가 아니어도 전송이 가능함. 2014년 5월 25일 S/W 버전부터 사용가능한 명령임.