G-STEP (C-Type) MANUAL

통신편

SPG MOTOR

-- 목 차 --

1. 통신 프로토콜

1 - 1 . 통신 기능

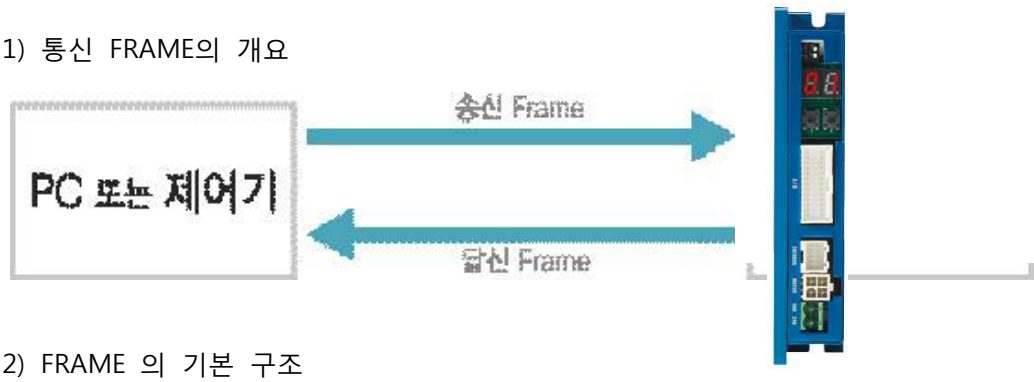
GSTEP-C는 RS-485 (2 선식)에서의 Daisy-Chain 링크에 의해 99축 까 지 의 제어가 가능합니다.

1 - 1 - 1 . 통신사양

1) 통신 방식 : 비동기식, 반이중 통신
2) 보우레이트 : 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600
3) 데이터 형식 : 8bit ASCII코드, HEX
4) 패리티, Stop Bit : None Parity, 1 Stop Bit
5) 오류 검사 : CRC
6) 최대 99축 연결

1 - 1 - 2 . RS-485 통신 프로토콜

1) 통신 FRAME의 개요



2) FRAME 의 기본 구조

Header	Frame Data	Tail
0xBB 0xCC	세부 구성 참조	0xBB 0xEE

① Frame Data 항의 데이터 중 실제 데이터가'0xBB'인 경우에는 바로 뒤에'0xBB'를 추가하여 줍니다. (byte stuffing)
② '0xBB'다음의 데이터가 '0xBB','0xCC'또는'0xEE'이 아니면 에러가 발생한 상황입니다.

위의 기본 구조 중 Frame Data항의 세부 구성은 다음과 같습니다.

Slave ID	Command	Length	Data	CRC	
1 Byte	1 Byte	1 Byte	0 ~ 255 bytes	2 Bytes	
				LOW	HIGH

ⓐ Slave ID : 연결된 드라이버 번호입니다. (1 ~ 99)
ⓑ Command : 아래의 Command 세부 내용 참조
ⓒ Data : Command 에 따라 데이터 구조 및 길이가 달라집니다.
ⓓ CRC : 통신 중 에러를 확인하기 위해 CRC-16-IBM의 생성다항식 'X16+X15+X2+1'을 사용합니다. CRC계산은 CRC항 이전의 모든 항에 대해 수행합니다.

1-2. Command 의 구성

1-2-1. Command 별 송수신 내용

* 모든 함수는 성공 시에 True(1)을 리턴하고 실패 시에는 False(0)을 리턴한다.
  만약, CRC에러일 경우 -1을 리턴한다.

Command	라이브러리 명	내 용
0x01	Gstep_Connect	포트번호와 통신속도를 지정하여 시리얼 통신을 초기화 합니다. 통신 속도 : 19200, 38400, 57600, 115200, 230400, 460800, 921600
0x03	Gstep_ServoAlarmReset	드라이버 알람을 리셋합니다. 송신 : 0 byte 답신 : 1 bytes #0 : 통신상태
0x04	Gstep_SaveAllParameters	파라미터를 드라이버내의 메모리에 저장합니다. 송신 : 0 byte 답신 : 1 bytes #0 : 통신상태
0x10	Gstep_GetParameter	드라이버내의 파라미터 값을 읽어옵니다. 송신 : 1 byte #0 : 표1. 파라미터 번호 참조 답신 : 5 bytes #0 : 통신상태 #1 : DATA-LL #2 : DATA-LH #3 : DATA-HL #4 : DATA-HH
0x11	Gstep_GetIOAssignMap	CN8 단자대의 PIN 설정상태를 읽어들입니다. 송신 : 1 byte #0 : I/O Num 0 ~11 은 입력에 해당 12 ~ 22는 출력에 해당 답신 : 4 bytes #0 : 통신상태 #1 : I/O번호-L (표2. I/O 번호 참조) #2 : I/O번호-H #3 : Level 상태
0x12	Gstep_GetSlaveInfo	드라이버 및 모터 정보를 요청합니다. 송신 : 0 byte 답신 : 6 bytes #0 : 통신상태

		#1 : Driver Info #2 : Main version #3 : Sub Version1 #4 : Sub Version2 #5 : Motor
0x13	Gstep_GetPosTableItem	송신 : 2 byte #0 : 번호-L #1 : 번호-H 답신 : 65 bytes #0 : 통신상태 #1~64 : 표3. 패턴 테이블 항목 참조
0x14	Gstep_GetActualPos	모터의 엔코더 위치값을 요청합니다. 송신 : 0 byte 답신 : 6 bytes #0 : 통신상태 #1 : 위치값 LL #2 : 위치값 LH #3 : 위치값 HL #4 : 위치값 HH #5 : 에러 번호
0x15	Gstep_GetPosError	모터의 위치 에러값을 요청합니다. 송신 : 0 byte 답신 : 6 bytes #0 : 통신상태 #1 : 에러값 LL #2 : 에러값 LH #3 : 에러값 HL #4 : 에러값 HH #5 : 에러 번호
0x16	Gstep_GetCommandPos	모터의 지령 위치값을 요청합니다. 송신 : 0 byte 답신 : 6 bytes #0 : 통신상태 #1 : 위치값 LL #2 : 위치값 LH #3 : 위치값 HL #4 : 위치값 HH #5 : 에러 번호
0x17	Gstep_GetActualVel	모터의 실제 속도값을 요청합니다. 송신 : 0 byte 답신 : 6 bytes #0 : 통신상태 #1 : 위치값 LL #2 : 위치값 LH #3 : 위치값 HL #4 : 위치값 HH #5 : 에러 번호
0x18	Gstep_GetAxisStatus	모터의 운전상태를 요청합니다. 송신 : 0 byte 답신 : 6 bytes #0 : 통신상태

		#1 : 운전상태-LL #2 : 운전상태-LH #3 : 운전상태-HL #4 : 운전상태-HH #5 : 에러 번호 표4. 모터 운전상태 참조
0x19	Gstep_GetAllStatus	송신 : 0 byte 답신 : 32 bytes #0 : 통신상태 #1 ~ 4 (4 Byte) : 입력 상태값 #5 ~ 8 (4 Byte) : 출력 상태값 #9 ~ 12 (4 Byte) : 운전 상태값 #13 ~ 16 (4 Byte) : 지령위치 값 #17 ~ 20 (4 Byte) : 실제위치 값 #21 ~ 24 (4 Byte) : 위치 편차 값 #25 ~ 28 (4 Byte) : 속도 값 #29 ~ 30 (2 Byte) : 현재 Pos Table값 #31 : 에러 번호
0x1a	Gstep_GetIO	I/O 상태 값을 읽어 들임 송신 : 0 byte 답신 : 10 bytes #0 : 통신상태 #1 ~ 4 (4 Byte) : 입력 상태값 #5 ~ 8 (4 Byte) : 출력 상태값 #9 : 에러 번호
0x1b	Gstep_SetIOOutput	Ouput 출력값을 지정함. 송신 : 2 byte #0 : I/O Number #1 : ON(1), OFF(0) 답신 : 1 byte #0 : 통신 상태
0x1c	Gstep_GetIsPatternTable	Pattern 테이블이 존재하는 위치를 읽어들임 송신 : 0 byte 답신 : 33 bytes #0 : 통신상태 #1 : bit0 - 0 번호 Pattern bit1 - 1 번호 Pattern bit2 - 2 번호 Pattern bit3 - 3 번호 Pattern bit4 - 4 번호 Pattern bit5 - 5 번호 Pattern bit6 - 6 번호 Pattern bit7 - 7 번호 Pattern ~ #33 : bit0 - 248 번호 Pattern bit1 - 249 번호 Pattern bit2 - 250 번호 Pattern bit3 - 251 번호 Pattern

		bit4 - 252 번호 Pattern bit5 - 253 번호 Pattern bit6 - 254 번호 Pattern bit7 - 255 번호 Pattern
0x1d	Gstep_GetMultiActualPos	10개단위로 모터 엔코더 위치값을 요청합니다. 송신 : 0 byte 답신 : 41 bytes #0 : 통신상태 #01 ~ #04: (시작번호+0)의 위치값 #05 ~ #08: (시작번호+1)의 위치값 #09 ~ #12: (시작번호+2)의 위치값 #13 ~ #16: (시작번호+3)의 위치값 #17 ~ #20: (시작번호+4)의 위치값 #21 ~ #24: (시작번호+5)의 위치값 #25 ~ #28: (시작번호+6)의 위치값 #29 ~ #32: (시작번호+7)의 위치값 #33 ~ #36: (시작번호+8)의 위치값 #37 ~ #40: (시작번호+9)의 위치값 * 시작번호는 Slave ID임
0x20	Gstep_SetParameter	파라미터 값을 전송합니다. 송신 : 5 byte #0 : 표1. 파라미터 번호 참조 #1 : DATA-LL #2 : DATA-LH #3 : DATA-HL #4 : DATA-HH 답신 : 1 bytes #0 : 통신상태
0x21	Gstep_SetIOAssignMap	I/O를 설정합니다. 송신 : 0 byte #0 : I/O Num 0 ~11 은 입력에 해당 12 ~ 22는 출력에 해당 #1 : I/O번호-L (표2. I/O 번호 참조) #2 : I/O번호-H #3 : I/O LEVEL (H(1). L(0)) 답신 : 1 bytes #0 : 통신상태
0x22	Gstep_SetPosTableItem	패턴 테이블 값을 설정함 송신 : 66 byte #0 : 테이블 No의 Low BYTE #1 : 테이블 No의 High BYTE #2~65 : 표3. 패턴 테이블 항목 참조 답신 : 1 bytes #0 : 통신상태
0x30	Gstep_MoveOriginSingleAxis	현재 설정된 파라미터 조건으로 원점 복귀를 합니다. 송신 : 0 byte

		답신 : 1 bytes #0 : 통신상태
0x31	Gstep_MoveSingleAxisAbsPos	절대위치로 이동 또는 설정합니다. 송신 : 9 byte #0 : 절대위치값_LL #1 : 절대위치값_LH #2 : 절대위치값_HL #3 : 절대위치값_HH #4 : 운전속도PPS_LL #5 : 운전속도PPS_LH #6 : 운전속도PPS_HL #7 : 운전속도PPS_HH #8 : 이동(1), 단지설정(0) 답신 : 1 bytes #0 : 통신상태
0x32	Gstep_MoveSingleAxisIncPos	상대값[Pulse]위치 만큼 이동합니다. 송신 : 8 byte #0 : 상대위치값_LL #1 : 상대위치값_LH #2 : 상대위치값_HL #3 : 상대위치값_HH #4 : 운전속도PPS_LL #5 : 운전속도PPS_LH #6 : 운전속도PPS_HL #7 : 운전속도PPS_HH 답신 : 0 bytes #0 : 통신상태
0x33	Gstep_MoveVelocity	JOG이송을 합니다. 송신 : 5 byte #0 : 운전방향, CW(1), CCW(0) #1 : 운전속도PPS_LL #2 : 운전속도PPS_LH #3 : 운전속도PPS_HL #4 : 운전속도PPS_HH 답신 : 0 bytes #0 : 통신상태
0x40	Gstep_ClearPosition	드라이버의 목표 위치값과 실제 위치값을 0으로 초기화 합니다. 송신 : 0 byte 답신 : 1 bytes #0 : 통신상태
0x41	Gstep_ServoEnable	Servo On/Off를 지령합니다. 송신 : 1 byte #0 : On(1), Off(0) 답신 : 1 bytes #0 : 통신상태
0x42	Gstep_SlowStop	모터 운전을 정지합니다. 송신 : 0 byte 답신 : 1 bytes

		#0 : 통신상태
0x43	Gstep_QuickStop	비상 정지 합니다. 송신 : 0 byte 답신 : 1 bytes #0 : 통신상태
0x44	Gstep_RunPosTableItem	패턴 테이블이 원하는 번호부터 시작합니다. 송신 : 2 byte #0 : 번호 Low Byte #1 : 번호 High Byte #2 : Auto Flag (0일 경우 해당 테이블 만 실행함) 답신 : 1 bytes #0 : 통신상태
0x45	Gstep_Compareout	Compare Out 신호를 발생시키기 위한 명령 송신 : 14 byte #0 : 출력 시작/종료 명령(1:시작,0:종료) #1 ~ 5 (4Byte) : 출력 시작위치[pulse] #6 ~ 9 (4Byte) : Pulse주기[pulse] #10 ~ 13 (4Byte) : Pulse폭[msec] 답신 : 2 bytes #0 : 통신상태 #1 : 명령 수행 여부 (0: 완료, 이외값: 에러)
0x46	Gstep_CompareoutStatus	Compare Out 신호출력 기능이 작동중인지 여부를 확인하는 명령 송신 : 0 byte 답신 : 2 bytes #0 : 통신상태 #1 : 현재 상태(1:출력 중, 0:종료)
0x70	Gstep_PositionAbsOverride	운전 중인 상태에서 목표 위치 (절대 위치값)을 변경하는 명령 송신 : 4 byte #0 ~ 3 (4Byte): 변경된 목표 위치값 답신 : 1 bytes #0 : 통신상태
0x71	Gstep_PositionIncOverride	운전 중인 상태에서 목표 위치 (상대위치값)을 변경하는 명령 송신 : 4 byte #0 ~ 3 (4Byte): 변경된 목표 위치값 답신 : 1 bytes #0 : 통신상태
0x72	Gstep_VelocityOverride	운전 중인 상태에서 운전 속도값[pps]을 변경하는 명령 송신 : 4 byte #0 ~ 3 (4Byte): 변경된 운전 속도 답신 : 1 bytes #0 : 통신상태
0x80	Gstep_AllMoveOrigin	연결된 모든 드라이버의 현재 설정된 파라미터 조건으로 원점 복귀 운전 명령 송신 : 0 byte 답신 : 없음

0x82	Gstep_AllIncPos	연결된 모든 드라이버에 대해 상대위치만큼 이동을 명령 송신 : 8 byte #0 ~ 3 (4Byte): 이송 펄스[pps] #4 ~ 7 (4Byte): 이송 속도[pps] 답신 : 없음
0x83	Gstep_AllAbsPosSet	연결된 모든 드라이버에 대해 절대위치만큼 이동을 명령 송신 : 8 byte #0 ~ 3 (4Byte): 이송 펄스[pps] #4 ~ 7 (4Byte): 이송 속도[pps] 답신 : 없음
0x90	Gstep_SingleAxisAbsPosEx	가감속을 포함한 절대값 이송 명령 송신 : 12 byte #0 ~ 3 (4Byte): 절대 위치값 #4 ~ 7 (4Byte): 이송 속도[pps] #8 ~ 9 : Accel 가속도 [msec] #10 ~ 11 : Decel 가속도 [msec] 답신 : 1 byte #0 : 통신상태
0x91	Gstep_SingleAxisIncPosEx	가감속을 포함한 상대 이송 명령 송신 : 12 byte #0 ~ 3 (4Byte): 상대 위치값 #4 ~ 7 (4Byte): 이송 속도[pps] #8 ~ 9 : Accel 가속도 [msec] #10 ~ 11 : Decel 가속도 [msec] 답신 : 1 byte #0 : 통신상태
0x92	Gstep_MoveLinearAbsPos	두 개의 드라이버에 대해 절대위치로의 직선 보간 운전을 명령 송신 : 12 byte #0 ~ 3 (4Byte): 이송 속도[pps] #4 ~ 7 (4Byte): 절대 위치값 #8 ~ 9 : Accel 가속도 [msec] #10 ~ 11 : Decel 가속도 [msec] #12 ~ 15 (4Byte): 최대이송 펄스 답신 : 1 byte #0 : 통신상태
0x93	Gstep_MoveLinearIncPos	두 개의 드라이버에 대해 상대위치로의 직선 보간 운전을 명령 송신 : 12 byte #0 ~ 3 (4Byte): 이송 속도[pps] #4 ~ 7 (4Byte): 상대 위치값 #8 ~ 9 : Accel 가속도 [msec] #10 ~ 11 : Decel 가속도 [msec] #12 ~ 15 (4Byte): 최대이송 펄스 답신 : 1 byte #0 : 통신상태
0x94	Gstep_MovePause	현재의 운전 상태를 일시정지 및 해제하는 명령

		송신 : 1 byte #0 : 0-일시정지 해제, 1-일시정지 답신 : 1 byte #0 : 통신상태
0x95	Gstep_MovePush	정해진 힘을 유지하기 위한 Push 운전 명령 송신 : 28 byte #0 ~ 3 : 위치이동 시작속도[pps] #4 ~ 7 : 위치이동 운전속도[pps] #8 ~ 11 : 위치이동 절대위치 #12 ~ 13 : 위치이동 가속시간[msec] #14 ~ 15 : 위치이동 감속시간[msec] #16 ~ 17 : Push이동 토크 비율 (20 ~ 90 %) #18 ~ 21 : Push이동 운전 속도값 (1 ~ 100,000[pps]) #22 ~ 25 : Push이동 절대위치 #26 ~ 27 : Push mode (0:stop mode, 1:non-stop mode) 답신 : 1 byte #0 : 통신 상태
0x96	Gstep_GetPushStatus	현재의 Push 운전 상태에 대한 확인 명령 송신 : 0 byte 답신 : 2 byte # 0 : 통신 상태 # 1 : 0 - 일반 위치 이동 대기 상태 1 - Push 운전 중이며 접촉하지 않은 상태 2 - 접촉되었고 힘이 유지되고 있는 상태
0x97	Gstep_LinearStart	직선 보간 운전을 명령 송신 : 0 byte 답신 : 1 byte # 0 : 통신 상태
0x98	Gstep_LinearStop	직선 보간 운전 중 정지를 명령 송신 : 0 byte 답신 : 1 byte # 0 : 통신 상태

표1. 파라미터 번호 참조

번호	이름	단위	하한	상한	출하치
0	Pulse per Revolution		0	11	11
1	Axis Max Speed	[pps]	1	500,000	500,000
2	Axis Start Speed	[pps]	1	35,000	1
3	Axis Acc Time	[msec]	1	9,999	100
4	Axis Dec Time	[msec]	1	9,999	100
5	Speed Override	[%]	1	500	100
6	Jog Speed	[pps]	1	500,000	5,000
7	Jog Start Speed	[pps]	1	35,000	1
8	Jog Acc Dec Time	[msec]	1	9,999	100
9	Servo Alarm Logic		0	1	0
10	Servo On Logic		0	1	0
11	Servo Alarm Reset Logic		0	1	0
12	S/W Limit Plus Value	[pulse]	-134,217,727	+134,217,727	+134,217,727
13	S/W Limit Minus Value	[pulse]	-134,217,727	+134,217,727	-134,217,727
14	S/W Limit Stop Method		0	1	1
15	H/W Limit Stop Method		0	1	1
16	Limit Sensor Logic		0	1	0
17	Org Speed	[pps]	1	500,000	5,000
18	Org Search Speed	[pps]	1	500,000	1,000
19	Org Acc Dec Time	[msec]	1	9,999	50
20	Org Method		0	2	0
21	Org Dir		0	1	0
22	Org Offset	[pulse]	-134,217,727	+134,217,727	0
23	Org Position Set	[pulse]	-134,217,727	+134,217,727	0
24	Org Sensor Logic		0	1	0
25	Position Loop Gain		0	15	4
26	Inpos Value		0	15	0
27	Pos Tracking Limit	[pulse]	1	+134,217,727	5,000
28	Motion Dir		0	1	0
29	Limit Sensor Dir		0	1	0
30	Org Torque Ratio	[%]	10	100	50
31	Pos. Error Overflow Limit	[pulse]	1	+134,217,727	5,000
32	Pos.Value Counting Method		0	1	0

표2. I/O 번호 참조

No.	출력 이름
0	NONE
1	COMP
2	INPOS
3	ALARM
4	MOVING
5	ACCDEC
6	PT ACK
7	PT END
8	ALARMBLK
9	ORGOK
10	SERVO_READY
11	TEMP_OUT0(예비신호)
12	PTOUT0
13	PTOUT1
14	PTOUT2
15	USEROUT0
16	USEROUT1
17	USEROUT2
18	USEROUT3
19	USEROUT4
20	USEROUT5
21	USEROUT6
22	USEROUT7
23	USEROUT8

No.	입력 이름
0	NONE
1	LIMIT +
2	LIMIT -
3	ORG_IN
4	CLEARPOS
5	PT_A0
6	PT_A1
7	PT_A2
8	PT_A3
9	PT_A4
10	PT_A5
11	PT_A6
12	PT_A7
13	PT_START
14	STOP
15	JOG +
16	JOG -
17	ALARM RESET
18	SERVO ON
19	PAUSE
20	ORG SEARCH
21	TEMP_IN0(예비신호)
22	ESTOP
23	JPT_IN0
24	JPT_IN1
25	JPT_IN2
26	JPT_START
27	USER_IN0
28	USER_IN1
29	USER_IN2
30	USER_IN3
31	USER_IN4
32	USER_IN5

표3. 포지션 테이블 항목

No.	명 칭	구조체 변수명칭	Offset 위치	크기	단위	하한	상한
1	Position	l_Position	0	4	pulse	-134217728	+134217728
2	Low Speed	dw_StartSpd	4	4	pps	0	500000
3	High Speed	dw_MoveSpd	8	4	pps	0	500000
4	Accel Time	w_AccelRate	12	2	msec	1	9999
5	Decel Time	w_DecelRate	14	2	msec	1	9999
6	Command	w_Command	16	2		0	10
7	Wait Time	w_WaitTime	18	2	msec	0	600000
8	Continuous Action	w_Continuous	20	2		0	1
9	Jump Table No.	w_Branch	22	2		0 10000	255 10255
10	Jump PT0	w_Cond_branch0	24	2		0 10000	255 10255
11	Jump PT1	w_Cond_branch1	26	2		0 10000	255 10255
12	Jump PT2	w_Cond_branch2	28	2		0 10000	255 10255
13	Loop Count	w_LoopCount	30	2		0	100
14	Loop Jump Table No	w_BranchAfterLoop	32	2		0 10000	255 10255
15	Position Table	w_PosTableSet	34	2		0	15
16	Loop Counter Clear	w_LoopCountCLR	36	2		0	255
17	Check Inposition	b_CheckInpos	38	2		0	1
18	Compare Position	l_TriggerPos	40	4	pulse	-134217728	+134217728
19	Compare Width	w_TriggerOnTime	44	2	msec	1	9999
20	Push Ratio	w_PushRatio	46	2	%	10	100
21	Push Speed	dw_PushSpeed	48	4	pps	0	50000
22	Push Position	l_PushPosition	52	4	pulse	-134217728	+134217728
23	Push Mode	w_PushMode	56	2		0	1

표4. 모터 운전상태

No.	STATUS Define	내용	Bit
1	FFLAG_ERRORALL	하나 이상의 에러가 발생함.	0x00000001
2	FFLAG_HWLIMITP	+방향 리미트 센서가 ON 이 된 경우	0x00000002
3	FFLAG_HWLIMITM	-방향 리미트 센서가 ON 이 된 경우	0x00000004
4	FFLAG_SWLIMITP	+방향 소프트웨어 리미트를 초과한 경우	0x00000008
5	FFLAG_SWLIMITM	-방향 소프트웨어 리미트를 초과한 경우	0x00000010
6	FFLAG_POSCNTOVER	Inposition 오차가 'Pos Tracking Limit'값보다 크게 오차가 발생한 경우	0x00000020
7	FFLAG_ERRUNDERLVOLT	저전압 에러가 발생한 경우	0x00000040
8	FFLAG_ERROVERSPEED	모터의 속도가 3000[rpm]을 초과한 경우	0x00000080
9	FFLAG_ERROVERLOAD	과부하 에러가 발생한 경우	0x00000100
10	FFLAG_ERROVERHEAT	과열 에러가 발생한 경우	0x00000200
11	FFLAG_ERRINPOSITION	Inposition이상 에러 발생	0x00000400
12	FFLAG_ERRINPULSE	입력 펄스가 과다한 경우	0x00000800
13	FFLAG_ERRINPUT_SVON	서버온 시 입력펄스가 들어온 경우	0x00001000
14	NOT USE		0x00002000
15	NOT USE		0x00004000
16	FFLAG_EMGSTOP	비상정지 상태임	0x00008000
17	FFLAG_SLOWSTOP	일반정지 상태임	0x00010000
18	FFLAG_ORIGINRETURNING	원점복귀 운전 중임	0x00020000
19	FFLAG_INPOSITION	Inposition 상태임	0x00040000
20	FFLAG_SERVOON	서보온 상태임	0x00080000
21	FFLAG_ALRALRESET	알람 리셋한 상태임	0x00100000
22	FFLAG_PTSTOPED	포지션 테이블 운전이 완료된 상태임	0x00200000
23	FFLAG_ORIGINSENSOR	원점 복귀 센서 입력이 들어온 상태임	0x00400000
24	FFLAG_ZPULSE	엔코더 Z상이 들어온 상태임	0x00800000
25	FFLAG_ORIGINRETOK	원점 복귀가 완료된 상태임	0x01000000
26	FFLAG_MOTIONCW	CW운전 중임	0x02000000
27	FFLAG_MOTIONING	모터가 현재 운전 중임	0x04000000
28	FFLAG_MOTIONPAUSE	모터가 일시 정지중임	0x08000000
29	FFLAG_MOTIONACCEL	가속구간의 운전 중인 상태	0x10000000
30	FFLAG_MOTIONDECEL	감속구간의 운전 중인 상태	0x20000000
31	FFLAG_MOTIONCONST	정속도 운전 중인 상태	0x40000000

1 - 1 - 3 . CRC 계산 예제

1)CRC16 의 '0xA001'사용방법

```
const unsigned short TABLE_CRCVALUE[] = {
 0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280,  0XC241,
 0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481,  0X0440,
 0XCC01, 0X0CC0, 0X0D8 0, 0XCD41, 0X0F00, 0XCFC1, 0XCE81,  0X0E40,
 0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880,  0XC841,
 0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81,  0X1A40,
 0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80,  0XDC41,
 0X1400, 0XD4C1, 0 XD581, 0X1540, 0XD701, 0X17C0, 0X1680,  0XD641,
 0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081,  0X1040,
0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281,  0X3240,
0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480,  0XF441,
0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80,  0XFE41,
0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881,  0X3840,
0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80,  0XEA41,
0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81,  0X2C40,
0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681,  0X2640,
0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080,  0XE041,
0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281,  0X6240,
0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480,  0XA441,
0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80,  0XAE41,
0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881,  0X6840,
0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80,  0XBA41,
0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81,  0X7C40,
0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681,  0X7640,
0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080,  0XB041,
0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280,  0X9241,
0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481,  0X5440 ,
0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81,  0X5E40,
0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880,  0X9841,
0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81,  0X4A40,
0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80,  0X 8C41,
0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680,  0X8641,
0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081,  0X4040
};
```

```
unsigned short CalcCRC(unsigned char* pDataBuffer, unsigned long usDataLen)
{
        unsigned char nTemp;
        unsigned short wCRCWord = 0xFFFF;

        while (usDataLen --)
        {
                nTemp = wCRCWord ^ *(pDataBuffer++);
                wCRCWord >>= 8;
                wCRCWord ^= TABLE_CRCVALUE[nTemp];
        }
        return wCRCWord;
}
```

1 - 1 - 4 . 답신 Frame 구조와 통신 에러

송신측의 명령에 대한 답신 측의 Frame 기본 구조는 동일합니다. 다만 아래의 Frame Data항에서의 차이점은 DATA의 첫 번째에 통신 상태가 추가되어 있습니다.

Slave ID	Command	Length	Data	CRC	
1 Byte	1 Byte	1 Byte	0 ~ 255 bytes	2 Bytes	
				LOW	HIGH

Data[0]	내 용
0x00	정상 상태입니다
0x80	수신한 Frame Type명령을 인식할 수 없습니다.
0x81	수신한 데이터의 값이 정해진 범위외의 데이터입니다.
0x82	수신된 Frame이 규격에 맞지 않은 데이터입니다.
0x83	운전 명령 실패 다음과 같은 상태에서 새로운 운전을 실행하려고 했습니다. 1) 현재 모터가 운전 중 2) 정지 명령 중 3) Servo OFF 상태 4) 기타 잘못된 운전 명령
0x84	RESET 실패 1) Servo On상태 2) 외부입력신호에 의해 이미 RESET상태임
0x85	Servo On 실패 : 알람 발생 중 Servo On명령을 실행하려고 함
0x86	Servo On 실패 : 비상 정지 중 Servo On명령을 실행하려고 함
0x87	Servo On 실패 : 외부 I/O로만 Servo On/Off할 수 있습니다.
0x88	CRC 에러 : 주변 노이즈에 의해 CRC에러가 발생합니다. 이 경우 DLL라이브러리에서 자동으로 1회 더 통신을 시도합니다.
0x89	패턴 테이블이 존재하지 않습니다.