

로보스타 로봇  
RCS 시리즈 옵션  
로보스타 프로토콜



- |  옵션 모듈
  - 로보스타 프로토콜

**Robostar**

[www.robostar.co.kr](http://www.robostar.co.kr)

(주) 로보스타

ROBOSTAR ROBOT  
RCS Series Option  
ROBOSTAR Protocol



- |  Option Module  
- Robostar Protocol

**Robostar**

[www.robostar.co.kr](http://www.robostar.co.kr)

---

Copyright © ROBOSTAR Co., Ltd 2016

이 사용 설명서의 저작권은 주식회사 로보스타에 있습니다.  
어떠한 부분도 로보스타의 허락 없이 다른 형식이나 수단으로 사용할 수 없습니다.

사양은 예고 없이 변경 될 수 있습니다.

---

## 제품 보증에 관하여

(주) 로보스타의 제품은 엄격한 품질 관리로 제조되고 있으며, 로보스타의 전 제품의 보증 기간은 제조일로부터 1년간입니다. 이 기간 내에 로보스타 측의 과실로 인한 기계의 고장 또는 정상적인 사용 중의 설계 및 제조상의 문제로 발생하는 고장에 한해서만, 무상으로 서비스를 합니다.

다음과 같은 경우에는 무상 서비스가 불가능합니다.

- (1) 보증 기간이 만료된 이후
- (2) 귀사 또는 제 3 자의 지시에 따른 부적당한 수리, 개조, 이동, 기타 취급 부주의로 인한 고장
- (3) 부품 및 그리스 등 당사의 지정 품 이외의 것의 사용으로 인한 고장
- (4) 화재, 재해, 지진, 풍수해 기타 천재지변에 의한 사고로 발생하는 고장
- (5) 분료 및 침수 등 당사의 제품 사양 외의 환경에서 사용함으로 인한 고장
- (6) 소모 부품의 소모로 인한 고장
- (7) 사용설명서 및 취급 설명서에 기재된 보수 점검 작업 내용대로 실시하지 않음으로 인해 발생하는 고장
- (8) 로봇 수리에 드는 비용 이외의 손해

### (주) 로보스타 주소 및 연락처

- 본사 및 공장  
경기도 안산시 상록구 수인로 700  
700, Suin-ro, Sangnok-gu, Ansan-si,  
Gyeonggi-do, Republic of South Korea  
(426-220)
- 수원 공장  
경기도 수원시 권선구 산업로 155번길 37  
37, Saneop-ro 155beon-gil, Gwonseon-gu,  
Suwon-si, Gyeonggi-do, Korea  
(441-813)
- 서비스요청 및 제품문의  
- 영업문의  
TEL. 031-400-3600  
FAX. 031-419-4249  
- 고객문의  
TEL. 1588-4428



www.robostar.co.kr

# 사용 설명서의 구성

본 제품에 관한 사용 설명서는 다음과 같이 구성되어 있습니다. 본 제품을 처음 사용하는 경우 모든 설명서를 충분히 숙지하신 후 사용하시기 바랍니다.

- **ROBOSTAR Protocol**

RS-232C통신을 사용하여 RCS 제어기의 접속 방법 및 사용법에 대하여 설명합니다.

# 목차

<b>제1장</b>	<b>개요.....</b>	<b>1-1</b>
1.1	시스템 구성.....	1-1
<b>제2장</b>	<b>기능.....</b>	<b>2-1</b>
2.1	RS-232통신에 대하여.....	2-1
2.2	D-SUB 9PIN 커넥터.....	2-1
2.3	RS-422통신에 대하여.....	2-2
2.4	RS-422 커넥터사양.....	2-2
<b>제3장</b>	<b>설치 및 동작 설정.....</b>	<b>3-1</b>
3.1	HARDWARE 설치 방법.....	3-1
3.1.1	RS-232.....	3-1
3.1.2	RS-422.....	3-1
3.2	CABLE 연결 방법.....	3-2
3.2.1	RS-232.....	3-2
3.2.2	RS-422.....	3-2
3.3	CONTROLLER 설정.....	3-3
3.3.1	통신 설정.....	3-3
3.4	컴퓨터와 통신을 위한 규격.....	3-4
3.5	RS-422 ID 설정.....	3-5
3.5.1	MYID 설정.....	3-5
<b>제4장</b>	<b>통신 프로토콜.....</b>	<b>4-1</b>
4.1	통신 규칙.....	4-1
4.2	RS422 통신 규칙.....	4-2
4.3	전체 프로토콜 명령어.....	4-3
4.4	세부 프로토콜 명령어.....	4-5
4.4.1	로봇 상태 정보 읽기(AA).....	4-5
4.4.2	현재 ERROR 상태 읽기(AB).....	4-6
4.4.3	로봇 현재 위치 좌표 읽기(AC).....	4-8
4.4.4	제어기 정보 읽기(펌웨어 버전)(AD).....	4-9
4.4.5	원점 수행(BA).....	4-10
4.4.6	지정한 데이터로 모션 수행(BC).....	4-11
4.4.7	지정된 데이터만큼 증분이동 수행(BD).....	4-12
4.4.8	JOG 이동 수행(BE).....	4-13

4.4.9	연속 JOG이동 수행 (BF)	4-14
4.4.10	JOG 이동 중지(BG)	4-15
4.4.11	로봇 속도 읽기(CA)	4-16
4.4.12	로봇 속도 쓰기(CB)	4-17
4.4.13	로봇 JOB 실행(CC)	4-18
4.4.14	로봇 JOB 정지(CD)	4-19
4.4.15	로봇 JOB 초기화(CE)	4-20
4.4.16	온라인 비상정지(CF)	4-21
4.4.17	온라인 Alarm 초기화 (CG)	4-22
4.4.18	온라인 MOVE STOP (CH)	4-23
4.4.19	온라인 원점 수행 정지(CI)	4-24
4.4.20	SERVO ON/OFF(DB)	4-25
4.4.21	실행 할 JOB 설정(DC)	4-26
4.4.22	JOB의 실행 방식 설정(EA)	4-27
4.4.23	현재 JOB Step 확인 (ED)	4-28
4.4.24	현재 실행 중인 JOB 이름 읽기(EF)	4-29
4.4.25	제어기에서 PC로 파일 전송(FA)	4-30
4.4.26	파일 저장(FB)	4-32
4.4.27	파일 검색(FC)	4-34
4.4.28	파일 정보 요청(FD)	4-35
4.4.29	파일 삭제(FE)	4-36
4.4.30	파일 복사(FF)	4-37
4.4.31	제어기 JOB파일 이름 변경 (FG)	4-38
4.4.32	저장 되어 있는 알람 내용 읽기(FH)	4-39
4.4.33	I/O의 Byte단위 읽기 (GA)	4-42
4.4.34	I/O의 Bit단위 읽기 (GB)	4-43
4.4.35	I/O의 Bit단위 쓰기(GC)	4-44
4.4.36	I/O의 2Byte단위 읽기 (GE)	4-45
4.4.37	I/O의 2Byte단위 쓰기 (GF)	4-46
4.4.38	I/O의 Byte단위 쓰기 (GG)	4-47
4.4.39	통신 속도 설정(KA)	4-48
4.4.40	제어기 Type 확인(KC)	4-49
4.4.41	통신상 에러 확인(KD)	4-50
4.4.42	GAIN을 제외한 파라미터값 변경 (XB)	4-51
4.4.43	변수의 종류 및 개수를 선택하여 값 확인 (XC)	4-55
4.4.44	Back up RAM에 파라미터 저장(XD)	4-57
4.4.45	통신 상 에러확인(XE)	4-58

4.4.46	현재 속도(RPM) 확인(XF)	4-59
4.4.47	전체 I/O 상태 확인(XG)	4-60
4.4.48	전체 위치, 정수형 변수 확인(XH)	4-61
4.4.49	위치, 정수형 변수 값 설정(XI)	4-63
4.4.50	정수, 위치형 변수 값 확인(XL)	4-65
4.4.51	위치변수값으로 이동 명령어 수행(XM)	4-67
4.4.52	파라미터 MY ID 값 확인(XN)	4-69
4.4.53	현재 설정 된 속도 확인(XP)	4-70
4.4.54	현재 가·감속값 설정(XQ)	4-71
4.4.55	현재 가·감속값 확인(XR)	4-73
4.4.56	현재 위치 값 확인(XS)	4-74
4.4.57	속도 값은 0으로 설정(XT)	4-75
4.4.58	현재 위치값을 0.000으로 설정(XU)	4-76
4.4.59	모터의 현재 상태 확인(XV)	4-77
4.4.60	지정한 번호부터 정수형 및 위치형 변수의 개수를 선택하여 값 변경(XW)	4-78

## 제1장 개요

### 1.1 시스템 구성

RCS 제어기는 Robostar 프로토콜을 사용하여 PC와 같은 시스템과 통신을 수행할 수 있습니다. RCS 제어기 Main Board의 Host Port를 이용하여 Robostar 프로토콜 통신이 가능합니다.

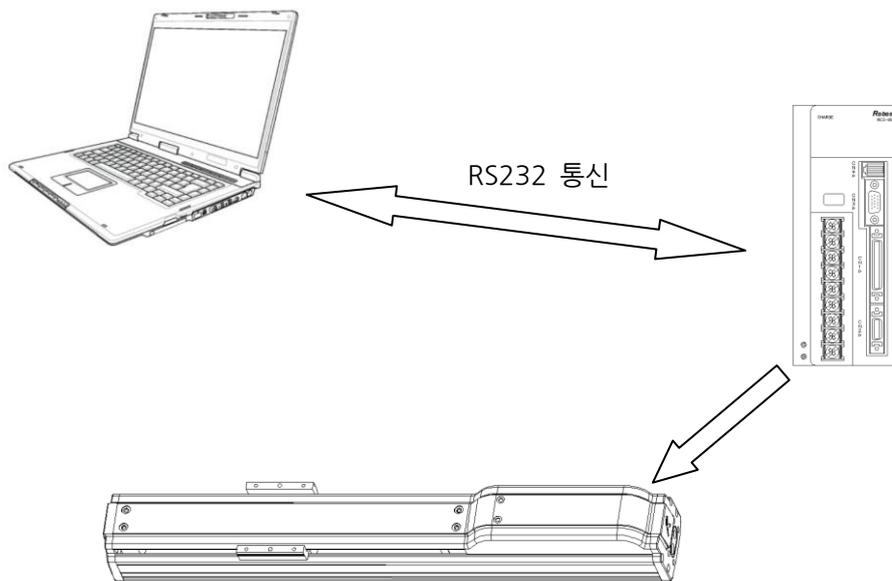


그림 1.1 시스템 구성도

## 제2장 기능

### 2.1 RS-232통신에 대하여

RS232C 통신 거리는 일반적으로 15m 이지만, 통신 속도가 높을수록 거리가 짧아 집니다.

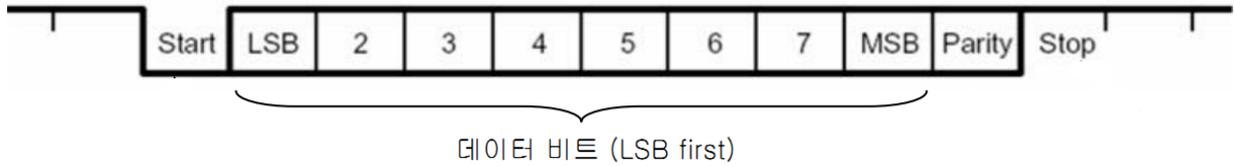
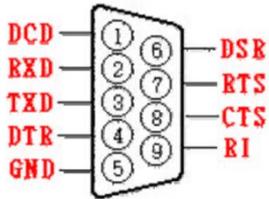


그림 2.1 비동기 통신 Format

- Start 비트 : 통신 Format 시작을 나타냄
- DATA 비트 : 5/6/7/8 비트 크기로 설정이 가능하며, LSB first 로 통신 데이터 값을 나타냄
- Parity 비트 : No/Even/Odd 로 설정이 가능하고, 노이즈로 1 비트가 변경되었을 때 검출이 가능
- Stop 비트: 1/1.5/2 크기로 설정이 가능하고, 데이터의 끝을 나타냄

### 2.2 D-SUB 9Pin 커넥터



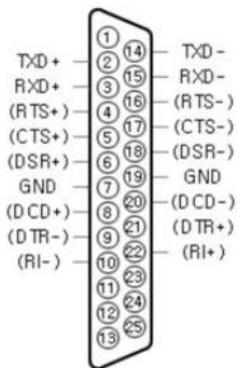
핀	설명
TXD	Transmit Data, 통신 데이터 출력 신호
RXD	Receive Data, 통신 데이터 입력 신호
RTS	Ready To Send 모뎀 통신 등에 사용하며 통신 준비 상태를 표시하며, 범용 출력 포트 사용 가능
CTS	Clear To Send 모뎀 통신 등에 사용하며 통신 준비 상태를 표시하며, 범용 입력 포트 사용가능
DTR	Data Terminal Ready 모뎀 통신 준비 신호로, 출력 포트 사용가능
DSR	Data Set Ready 모뎀 통신 준비 신호로 입력 포트 사용 가능
DCD	Data Carrier Detect, 입력 포트
RI	Ring Indicator 입력 포트
GND	그라운드

표 2.1 커넥터 설명

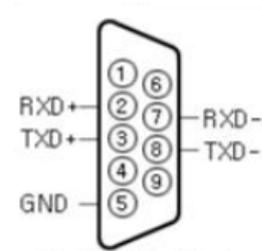
### 2.3 RS-422통신에 대하여

RS-422 는 EIA 에 의해서 전기적인 사양이 규정되어 있으며 RS-422 에서는 Point To Point 모드와 Multi-Drop 모드 두 가지가 있습니다. Point To Point 모드인 경우 RS-232 와 신호선 당 2 개의 라인이 필요한 것만 빼고 사용하는 방법에 있어서 별다른 필요가 없습니다. 하지만 Multi-Drop 모드인 경우는 Stream 에 의한 사용 방법이므로 다소 사용법이 복잡하다. 일반적으로 사용되는 신호선은 TXD+, TXD-, RXD+ 및 RXD- 이고 나머지 신호선은 거의 사용되지 않습니다.

### 2.4 RS-422 커넥터사양



D-SUB 25핀 RS422 커넥터



D-SUB 9핀 RS422 커넥터

물리적으로 하나의 신호선에 두개의 라인이 필요한데 그들의 표현은 신호선명 뒤에 '+' 와 '-' 로써 구분표기 한다.

즉, 예를 들면 RS-232 의 TXD 신호선이 RS-422에서는 TXD+와 TXD-로 나누어 지는 것과같다.

## 제3장 설치 및 동작 설정

### 3.1 Hardware 설치 방법

다음과 같은 과정을 거쳐 제어기의 프로토콜을 사용할 수 있습니다.

#### 3.1.1 RS-232

- 1) 제어기 Main Board 통신포트(HOST PORT)에 Serial 케이블을 연결합니다.

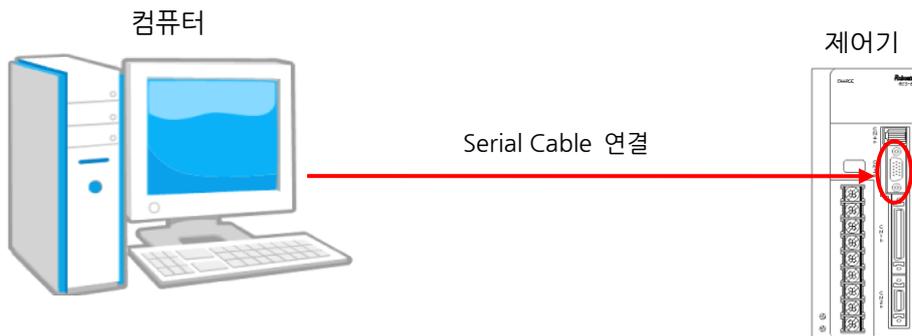


그림 3.1 Serial 케이블 연결방법

#### 3.1.2 RS-422

- 2) 최대 255 대의 제어기 Main Board 통신포트(HOST PORT)에 422 케이블을 PC 에 연결합니다.

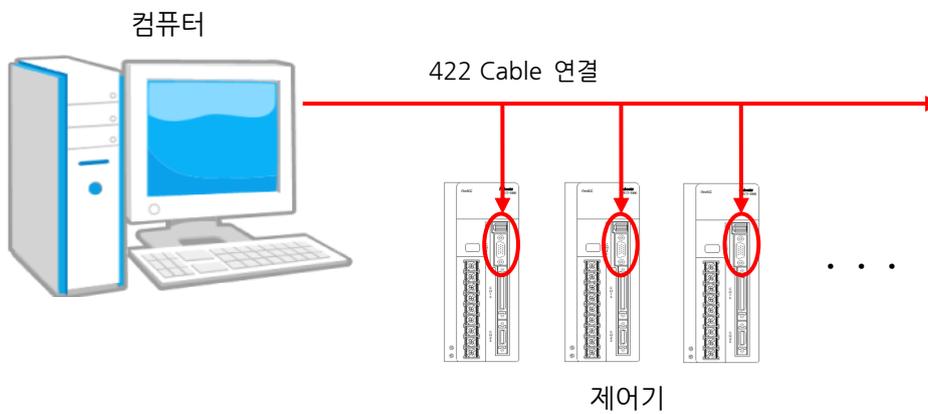


그림 3.2 422 케이블 연결방법

### 3.2 Cable 연결 방법

#### 3.2.1 RS-232

케이블 연결 방법으로 "그림 3.2 케이블 결선도"와 "표3.1 결선핀 번호 설명"를 참조하시기 바랍니다.

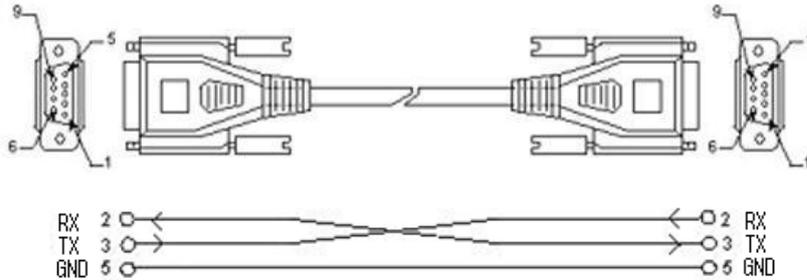
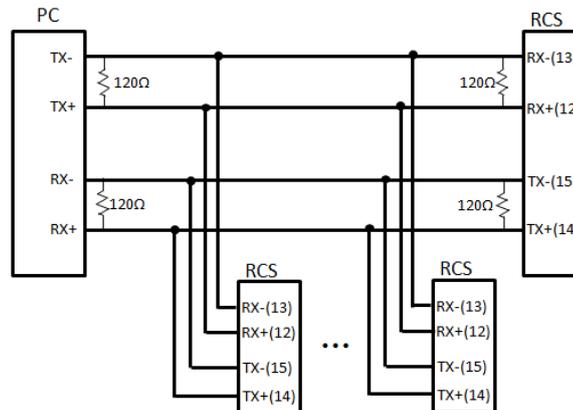


그림 3.3 케이블 결선도

핀 번호	이름	내용	기능 설명
2	RxD	Received Data	COM1 수신 데이터 신호
3	TxD	Transmitted Data	COM1 송신 데이터 신호
9	RxD2	Received Data2	COM2 수신 데이터 신호
10	TxD2	Transmitted Data2	COM2 송신 데이터 신호
5	GND	Ground	그라운드

표3.1 결선핀 번호 설명

#### 3.2.2 RS-422



핀 번호	이름	내용	기능 설명
12	RX+	+Received Data	RS422 +수신 데이터 신호
13	RX-	-Received Data	RS422 -수신 데이터 신호
14	TX+	+Transmitted Data	RS422 +송신 데이터 신호
15	TX-	-Transmitted Data	RS422 -송신 데이터 신호
5	GND	Ground	그라운드

표3.2 결선핀 번호 설명

### 3.3 Controller 설정

RCS 시리즈 컨트롤러에서 프로토콜을 사용하기 위해서는 PC와 컨트롤러의 통신 속도를 동일하게 설정을 하셔야 합니다.

#### 3.3.1 통신 설정

■ 설정순서

Step 1.

**MAIN 화면 이동**

TPS-9000T Ver1.3  
F1: Teach Pendant  
 F2: RS-422 Multipoint  
 F3: Data up/Down Load

**F1**

컨트롤러의 전원을 ON 시킨 후 Teach Pendant를 선택합니다.

RoboStar RCS-8000C  
 Servo Controller  
 PARA V00.22  
PRESS ENT KEY

**ENT**

ENTER를 누릅니다.

Step 2.

**파라미터 화면 이동**

Servo Controller  
 ROBOT PLC PARA VIEW

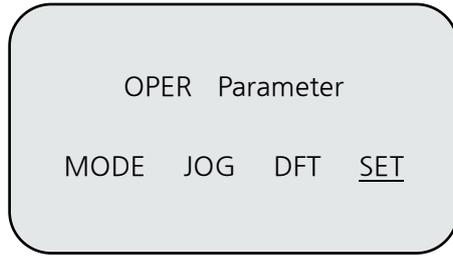
**F3**

PARA를 선택합니다

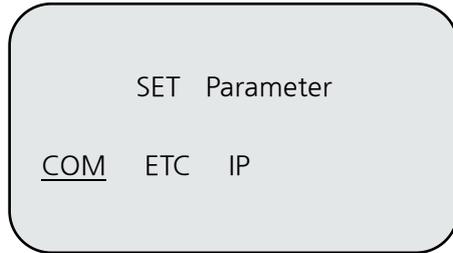
Parameter Setting  
 SERVO MECH OPER I/O

**F3**

OPER를 선택합니다.



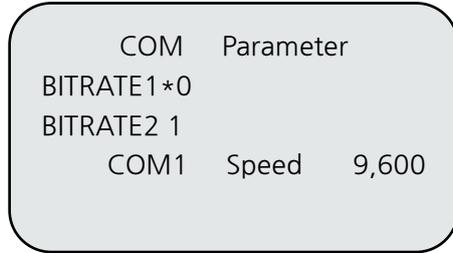
**F4** 컨트롤러의 전원을 ON 시킨 후 Teach Pendant를 선택합니다.



**F1** ENTER를 누릅니다.

Step 3.

통신속도 설정



사용할 포트를 선택합니다.  
 BITRATE1 : COM1  
 BITRATE2 : COM2  
 BITRATE2 : RS422

설정 값  
 0 : 9600bps, 1 : 19,200bps,  
 2 : 38,400bps, 3 : 115,200bps

3.4 컴퓨터와 통신을 위한 규격

항 목		초기값
COM1 사용 시	BAUD RATE1	0 (9,600bps)
COM2 사용 시	BAUD RATE2	1 (19,200bps)
My_ID	BAUD RATE3	1 (19,200bps)



**CAUTION**

- 통신속도 변경 시 반드시 전원을 ON/OFF 바랍니다.
- COM1의 통신 속도를 0 (9,600bps)이 아닌 다른 값으로 변경 시 Teach Pendant가 동작하지 않습니다.

### 3.5 RS-422 ID 설정

RS422통신을 위한 제어기 ID를 설정합니다.

#### 3.5.1 MY ID 설정

##### ■ 설정순서

Step 1.

##### MAIN 화면 이동

TPS-9000T Ver1.3  
F1: Teach Pendant  
 F2: RS-422 Multipoint  
 F3: Data up/Down Load

F1

컨트롤러의 전원을 ON 시킨 후 Teach Pendant를 선택합니다.

RoboStar RCS-8000C  
 Servo Controller  
 PARA V00.22  
PRESS ENT KEY

ENT

ENTER를 누릅니다.

Step 2.

##### Program 화면 이동

Servo Controller  
 ROBOT PLC PARA VIEW

F3

PARA을 선택합니다.

Parameter Setting  
 SERVO MECH OPER I/O

F3

OPER을 선택합니다.

OPER. Parameter  
 MODE JOG DFT SET

F4

SET을 선택합니다.

Step 3.

```
SET Parameter
COM ETC IP
```

COM을 선택합니다.



MY\_ID설정

```
COM Parameter
BITRATE1*0
BITRATE2 1
COM1 Speed 9,600
```

2 : Down 선택  
MY\_ID로 이동합니다.



```
DATAMODE 0
MY_ID *0
<end of parameter>
0,255
```

RS422통신을 위한 ID를 설정합니다.  
0 : 사용 안함.  
1~255 : RS422통신 설정.

## 제4장 통신 프로토콜

### 4.1 통신 규칙

- 최대 Packet 길이는 STX, ETX, LRC를 포함하여 250Bytes까지 송/수신이 가능 합니다.
- Packet에 포함되어 있는 LRC를 확인 하여 LRC가 다른 경우 NAK를 보내고 일정 횟수 이상이면 RST를 보냅니다. NAK를 받으면 다시 이전 Packet를 보내야 합니다.
- RST는 통신 종료를 의미하며 RST를 받으면 즉시 통신을 종료하고 통신 대기점으로 복귀합니다.
- ACK 는 한 개의 통신 Packet을 잘 받았다는 Acknowledge의 의미로 사용됩니다.  
한 개의 통신 Packet이란 다음과 같이 STX, DATA, ETX, LRC 로 이루어진 구조를 나타 냅니다.

STX 0x02	DATA	ETX 0x03	LRC
-------------	------	-------------	-----

STX: 0x02

ETX: 0x03

NAK: 0x15

RST: 0x12

ACK: 0x06

- LRC 계산법 :STX, LRC를 제외한 **exclusive-OR(^)** 입니다.  
 $LRC = DATA[0] \wedge DATA[1] \wedge DATA[2] \wedge \dots \wedge DATA[n] \wedge ETX$   
 (만약 LRC 값이 0 이면 ETX로 합니다.)
- FLAG  
 0x30: 기능 실행 OK  
 요구한 프로토콜의 기능에 대하여 실행이 가능 하거나 완료 되었음을 나타냅니다.  
 0x31: 프로토콜 ERROR  
 PC 에서 보내는 Packet 내용이 제어가 판단 가능한 DATA가 아닌경우 반환 됩니다.  
 EX) “지정한 데이터로 모션 수행(BC)”의 경우 좌표계 입력 값이 ‘0’ or ‘1’인데 그 이상의 값이 입력 된 경우 0x31를 반환 합니다.  
 0x32: 기능 실행 FAIL  
 PC에서 요청한 기능을 실행 하지 못하는 경우 반환됩니다. 기능 실행 FAIL에 대한 자세한 내용은 “통신 에러 원인 읽기(KD)”를 이용하여 확인 가능 합니다.  
 0x33: 현재 제어가 지원하지 않는 기능  
 프로토콜은 존재하지만 제어기 종류에 따라 기능 구현이 어려운 경우에 0x33을 반환 합니다.  
 0x34: 연속된 패킷의 종료  
 Packet이 한번으로 끝나지 않고 반복 될때 Packet의 종료를 나타냅니다.  
 제어기로부터 파일을 읽어올 때에 파일의 크기에 따라 한 패킷을 넘는 경우가 발생 될 경우 제어기는 파일의 내용을 보내주다가 파일 내용을 다 보내게 되면 FLAG에 0x34를 보내어 Packet 종료를 나타냅니다.

## 4.2 RS422 통신 규칙

기본적인 규칙은 4.1. 통신 규칙과 동일 하며 추가 변경 되는 사항은 아래와 같습니다.

- 통신은 제어기 전원 ON 이후 5초 뒤에 해당 기능을 사용 바랍니다.
- 1:N통신을 위해 제어기로 보내는 패킷에 아래와 같이 RS422통신을 구분하는 '#'과 제어기의 파라미터 MY\_ID로 지정하는 ID값이 추가됩니다. ID값은 1~255까지 설정 할 수 있습니다.

STX 0x02	# 0x23	ID 0x01~0xFF	DATA	ETX 0x03	LRC
-------------	-----------	-----------------	------	-------------	-----

- 한대의 제어기 통신이 완료되기 전까지 다른 제어기는 동작하지 않습니다.

4.3 전체 프로토콜 명령어

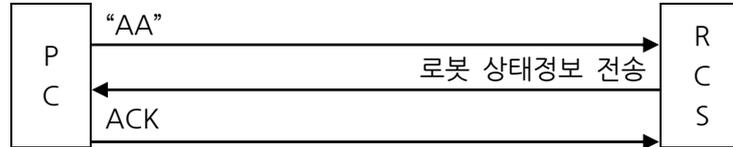
GROUP1	GROUP1	기능설명
A	A	로봇 상태 정보 읽기
	B	현재 알람 정보 확인
	C	로봇 현재 위치 좌표 읽기
	D	제어기 정보 읽기(펌웨어 버전)
B	A	원점 수행
	C	지정된 데이터로 지정한 모션 수행
	D	지정된 데이터만큼 증분이동 수행
	E	JOG 이동 수행
	F	연속 JOG 이동 수행
	G	JOG 이동 중지
C	A	현재 속도 확인
	B	로봇 속도 쓰기
	C	로봇 JOB 실행
	D	로봇 JOB 정지
	E	로봇 JOB 초기화
	F	온라인 비상정지
	G	온라인 Alarm 초기화
	H	온라인 MOVE STOP
	I	온라인 원점 수행 정지
D	B	서보 ON/OFF
	C	실행 할 JOB 선택
E	A	JOB 실행 방식 설정
	D	현재 JOB Step 확인
	F	현재 설정 된 JOB 번호 확인
F	A	제어기 JOB 을 PC 로 전송
	B	PC 의 JOB 을 제어기로 전송
	C	파일 존재여부 확인
	D	파일 정보 확인
	E	제어기 JOB 파일 삭제
	F	제어기 JOB 파일 복사
	G	제어기 JOB 파일 이름 변경
	H	현재 알람 확인

GROUP1	GROUP1	기능설명
G	A	I/O의 Byte 단위 읽기
	B	I/O의 Bit 단위 읽기
	C	I/O Byte 단위 쓰기
	E	I/O의 2Byte 단위 읽기
	F	I/O의 2Byte 단위 쓰기
	G	I/O의 Byte 단위 쓰기
K	A	통신속도 설정
	C	제어기 Type 확인
	D	통신 상 에러확인
X	B	GAIN을 제외한 파라미터값 변경
	C	변수의 종류 및 개수를 선택하여 값 확인
	D	Back up RAM에 파라미터 저장
	E	현재 메인 알람코드 확인
	F	현재 속도(RPM) 확인
	G	전체 I/O 상태 확인
	H	전체 위치, 정수형 변수 확인
	I	위치, 정수형 변수 값 설정
	L	위치, 정수형 변수 값 확인
	M	위치변수 값으로 이동명령어 수행
	N	MY ID 확인
	P	현재 설정된 속도 확인
	Q	현재 가·감속값 설정
	R	현재 가·감속 설정값 확인
	S	현재 위치값 확인
	T	속도 값 0으로 설정
	U	현재 위치값을 0.000으로 설정
	V	모터의 현재상태확인
W	원하는 번호부터 정수형 및 위치형변수의 개수를 선택하여 값 변경	

### 4.4 세부 프로토콜 명령어

#### 4.4.1 로봇 상태 정보 읽기 (AA)

현재 로봇 상태 확인 명령어입니다.



- Protocol DATA

PC	0	1	2	3	4	
↓	STX	A	A	ETX	LRC	
제어기	0x02	0x41	0x41	0x03		
제어기	0	1	2	3	4	5
↓	STX	상태	상태정보	상태정보	ETX	LRC
PC	0x02	Flag	1	2	0x03	
PC	0					
↓	ACK					
제어기	0x06					

- 상태정보 1

로봇의 상태 정보 (8Bits)								
BIT	7	6	5	4	3	2	1	0
설명	없음	없음	1	1	Alarm	1	INPOS	RUN

- 상태정보 2

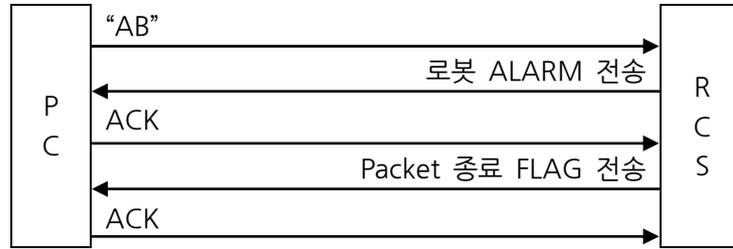
로봇의 상태정보 (8Bits)								
BIT	7	6	5	4	3	2	1	0
설명	없음	없음	1	1	없음	없음	Servo on	Origin

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.
- 상태정보 1의 2번 BIT의 경우 이전 RCS7000시리즈와 호환을 위해 1로 표시 합니다.
- 상태정보 1, 2의 0~3번 BIT의 값이 ETX(0x30)이 되는 경우를 배제하기 위해 5, 6번 BIT를 1로 표시 합니다.

4.4.2 현재 ERROR 상태 읽기(AB)

제어기 Alarm 정보 읽기 명령어 입니다.

패킷은 일반 Alarm과 "E14.XX"두 가지 형태로 송·수신 받습니다.



- Protocol DATA

1) 일반 Alarm

Alarm 메시지를 전송합니다.

PC	0	1	2	3	4
↓	STX	A	B	ETX	LRC
제어기	0x02	0x41	0x42	0x03	

제어기	0	1	2 ~ 21	22	23
↓	STX	상태	Alarm	ETX	LRC
PC	0x02	Flag		0x03	

PC	0
↓	ACK
제어기	0x06

제어기	0	1	2	3
↓	STX	Flag	ETX	LRC
PC	0x02	0x34	0x03	

PC	0
↓	ACK
제어기	0x06

- Alarm : 알람코드의 내용 또는 소프트웨어 알람의 번호를 표시합니다.

Alarm (20Bytes)

패킷 번호	2~21
내용	알람코드 내용

알람이 소프트웨어 알람(E15.XX)인 경우

Alarm (20Bytes)

패킷 번호	2	3~16	17	18	19	20	21
내용	0x20	"Software Alarm"	알람코드			0x20	0x20

2) E14.XX (Parameter Error)

알람 번호가 E14.XX인 경우 Parameter Error관련 Alarm 메시지와 Limit값을 전송합니다.

PC	0	1	2	3	4
↓	STX	A	B	ETX	LRC
제어기	0x02	0x41	0x42	0x03	

제어기	0	1	2 ~ 21	22 ~ 41	22	23
↓	STX	상태	Alarm	Limit	ETX	LRC
PC	0x02	Flag			0x03	

PC	0
↓	ACK
제어기	0x06

제어기	0	1	2	3
↓	STX	Flag	ETX	LRC
PC	0x02	0x34	0x03	

PC	0
↓	ACK
제어기	0x06

- Alarm : 소프트웨어 알람 코드번호를 표시 합니다.

Alarm (20Bytes)

패킷 번호	2	3~16	17	18	19	20	21
내용	0x20	Parameter Error	알람코드			0x20	0x20

- Limit : Limit를 표시합니다.

Limit (19Bytes)

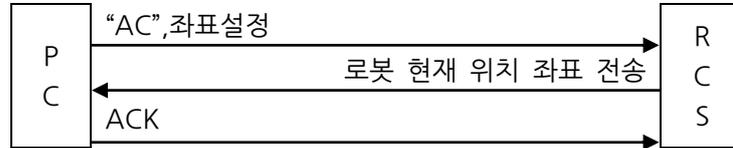
패킷 번호	22	23	24~28	29	30~39	40
내용	0x20	0x20	"Limit"	0x20	Limit 값	0x20

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

 CAUTION

- 제어기가 알람 상태에 있는 경우에만 알람 내용에 대하여 반환됩니다.  
(현재 발생된 알람 내용을 알고 싶을 경우 사용합니다.)

4.4.3 로봇 현재 위치 좌표 읽기(AC)  
지정한 채널의 로봇 현재 위치 좌표 읽기 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4	5
↓	STX	A	C	좌표	ETX	LRC
제어기	0x02	0x41	0x43	설정	0x03	
제어기	0	1	2 ~ 11	12	13	
↓	STX	상태	위치 값	ETX	LRC	
PC	0x02	Flag		0x03		
PC	0					
↓	ACK					
제어기	0x06					

● 좌표 설정 : 로봇 좌표 종류 설정.

좌표 설정 (1bytes)

값	좌표 종류
0	Pulse
1	Joint

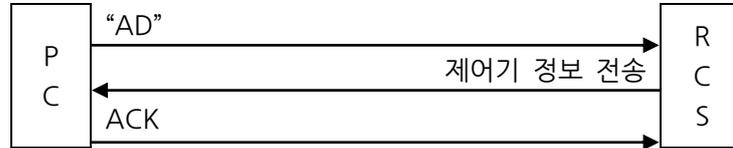
● 위치 값 : 현재 위치값 확인.

위치 값 (10bytes)

패킷 번호	2	3	4	5	6	7	8	9	10	11
내용	0x20	정수형 자릿수					'.'	소수점 자릿수		

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.4 제어기 정보 읽기(펌웨어 버전) (AD)  
 제어기 INFO 읽기 명령어 입니다.



● Protocol DATA

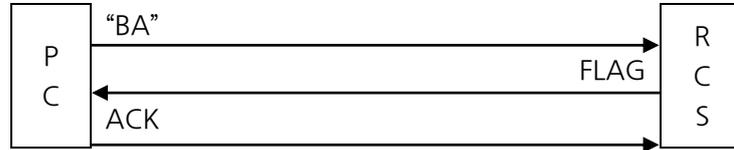
PC	0	1	2	3	4
↓	STX	A	D	ETX	LRC
제어기	0x02	0x41	0x44	0x03	
제어기	0	1	2~41	42	43
↓	STX	상태	제어기	ETX	LRC
PC	0x02	Flag	정보	0x03	
PC	0				
↓	ACK				
제어기	0x06				

- 제어기 정보 : 제어기 종류 및 메인 및 파라미터 펌웨어 버전입니다.

제어기 정보 (40bytes)								
패킷 번호	2	3	...	21	22	23	...	41
내용	RCS80C 펌웨어 버전				펌웨어 날짜 파라미터버전			

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하시기 바랍니다.

4.4.5 원점 수행(BA)  
 로봇 원점(Origin)수행 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	B	A	ETX	LRC
제어기	0x02	0x42	0x41	0x03	
제어기	0	1	2	3	
↓	STX	상태	ETX	LRC	
PC	0x02	Flag	0x03		
PC	0				
↓	ACK				
제어기	0x06				

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하시기 바랍니다.

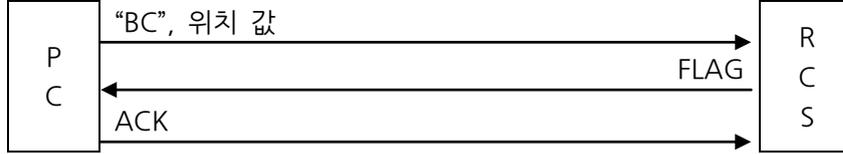
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.



**CAUTION**

- “원점수행(BA)”시 자동으로 Servo ON이 됩니다
- ENCTYPE 파라미터가 ABS이거나 ORG\_RULE이 0인경우 RUNFAIL(0x32)이 발생합니다.

4.4.6 지정한 데이터로 모션 수행(BC)  
로봇의 목표 좌표를 입력하여 모션을 수행하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4	5~14	15	16
↓	STX	B	C	1	1	위치 값	ETX	LRC
제어기	0x02	0x42	0x43	0x31	0x31		0x03	
제어기	0	1	2	3				
↓	STX	상태	ETX	LRC				
PC	0x02	Flag	0x03					
PC	0							
↓	ACK							
제어기	0x06							

- 위치 값 : 이동 할 위치값입니다.  
제어기에서 인식되는 위치값은 0x20를 제외한 숫자에서 0.001을 곱한 값 입니다.

위치 값(10bytes)

예 1) 위치값	패킷 번호	5	6	7	8	9	10	11	12	13	14
12345.678	내용	0x20	0x20	1	2	3	4	5	6	7	8

예 2) 위치값	패킷 번호	5	6	7	8	9	10	11	12	13	14
123.456	내용	0x20	1	2	3	4	5	6	0x20	0x20	0x20

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

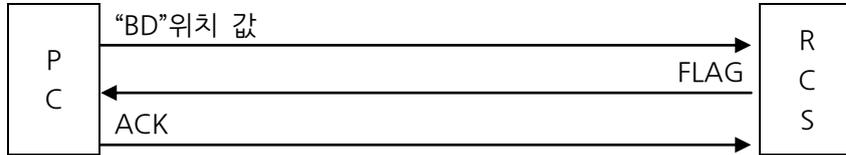
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

CAUTION

➤ “지정한 데이터로 모션 수행(BC)”시 자동으로 Servo ON이 됩니다

4.4.7 지정된 데이터만큼 증분이동 수행 (BD)

현재 위치에서 입력한 위치 값 만큼 로봇을 증분 이동 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4	5~14	15	16
↓	STX	B	D	1	1	위치 값	ETX	LRC
제어기	0x02	0x42	0x44	0x31	0x31		0x03	
제어기	0	1	1	2				
↓	STX	상태	ETX	LRC				
PC	0x02	Flag	0x03					
PC	0							
↓	ACK							
제어기	0x06							

- 위치 값 : 증분 이동 할 위치값입니다.  
제어기에서 인식되는 위치값은 0x20를 제외한 숫자에서 0.001을 곱한 값 입니다.

위치 값 : 이동 할 위치값 (10bytes)

예 1) 위치값	패킷 번호	5	6	7	8	9	10	11	12	13	14
12345.678	내용	0x20	0x20	1	2	3	4	5	6	7	8

예 2) 위치값	패킷 번호	6	7	8	9	10	11	12	13	14
123.456	내용	1	2	3	4	5	6	0x20	0x20	0x20

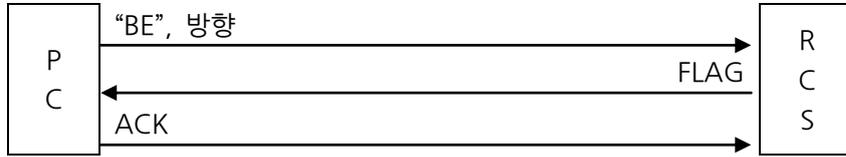
- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

CAUTION

➤ “지정된 데이터만큼 증분이동 수행(BD)”시 자동으로 Servo ON이 됩니다

4.4.8 JOG이동 수행 (BE)  
JOG동작 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4	5	6	7
↓	STX	B	E	0x20	방향	0x20	ETX	LRC
제어기	0x02	0x42	0x45	0x20	Data	0x20	0x03	
제어기	0	1	1	2				
↓	STX	상태	ETX	LRC				
PC	0x02	Flag	0x03					
PC	0							
↓	ACK							
제어기	0x06							

● 방향 Data : JOG 이동 방향을 설정합니다.

방향 Data	
값	Data
0	JOG +
1	JOG -

※ 이 외의 값을 입력하면 "0x31 프로토콜 ERROR"의 FLAG가 발생합니다.

● STX, ETX, LRC, FLAG: "4.1통신 규칙"을 참조 하십시오.

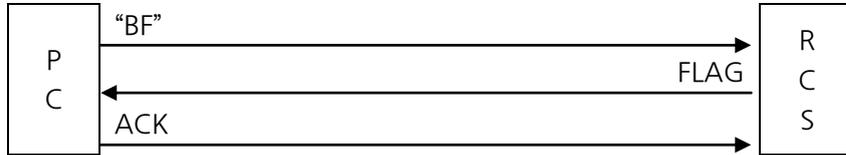
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 "통신에러 원인읽기(KD)"프로토콜을 이용하여 읽기를 권장합니다.

 CAUTION

- "JOG이동 수행(BE)"시 자동으로 Servo ON이 됩니다
- 연속적인 Jog동작을 유지하기 위해서는 370ms 이내로 "연속 JOG이동 수행"(BF)프로토콜을 전송합니다.

4.4.9 연속 JOG이동 수행 (BF)

JOG 동작이 계속 유지 되도록 제어기에 요구하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	B	F	ETX	LRC
제어기	0x02	0x42	0x46	0x03	
제어기	0	1	2	3	
↓	STX	상태	ETX	LRC	
PC	0x02	Flag	0x03		
PC	0				
↓	ACK				
제어기	0x06				

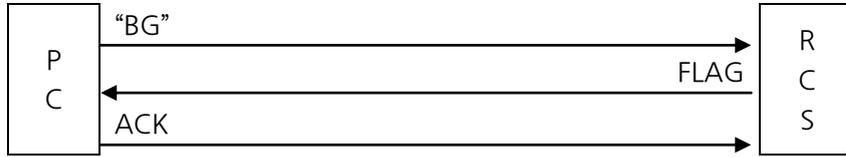
- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.



**CAUTION**

- 연속적인 Jog동작을 유지하기 위해서는 370ms 이내로 "연속 JOG이동 수행"(BF)프로토콜을 전송합니다.
- 370ms 초과 시 로봇은 정지합니다.

4.4.10 JOG 이동 중지(BG)  
 JOG 동작 정지 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	B	G	ETX	LRC
제어기	0x02	0x42	0x47	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

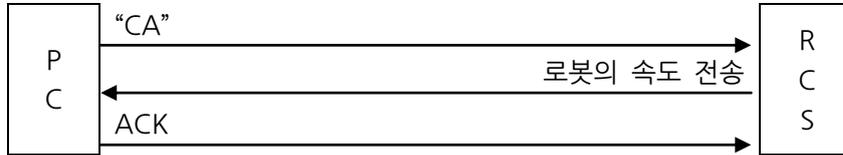
  

PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

### 4.4.11 로봇 속도 읽기(CA)

로봇 채널의 로봇 현재 속도 읽기 명령어 입니다.



- Protocol DATA

PC	0	1	2	3	4
↓	STX	C	A	ETX	LRC
제어기	0x02	0x43	0x41	0x03	
제어기	0	1	2~4	5	6
↓	STX	상태	현재	ETX	LRC
PC	0x02	Flag	속도값	0x03	
PC	0				
↓	ACK				
제어기	0x06				

- 현재 속도값 : 현재 구동중인 로봇의 속도값(%)입니다.

현재 속도 값(3Bytes)

패킷 번호	2	3	4
내용	백의 자리	십의자리	일의자리

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

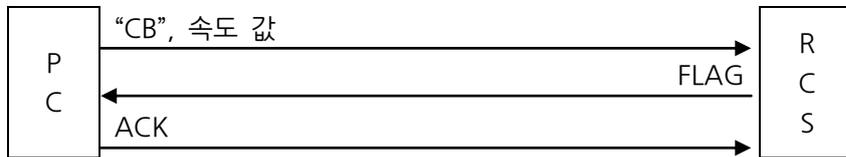


#### CAUTION

- 로봇 이동 속도의 최대값은 100(%)입니다.

4.4.12 로봇 속도 쓰기(CB)

로봇 채널의 로봇 구동 속도 값을 입력 하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3~5	6	7
↓	STX	C	B	속도 값	ETX	LRC
제어기	0x02	0x43	0x42	지정(%)	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

- 속도 값 지정 : 이동 할 로봇의 속도값(%)을 설정 합니다.

속도 값 지정(3Bytes)			
패킷 번호	3	4	5
내용	백의 자리	십의자리	일의자리

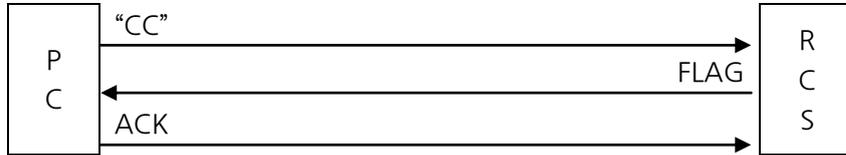
- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.



CAUTION

- 로봇 이동 속도의 최대값은 100(%)입니다.
- 기구부의 허용 최대 RPM을 초과하여 사용 시 소음 및 파손의 위험이 있습니다. 반드시 기구부에 부착된 라벨을 확인 후 사용 하시길 바랍니다.

4.4.13 로봇 JOB 실행(CC)  
JOB을 실행하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	C	C	ETX	LRC
제어기	0x02	0x43	0x43	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

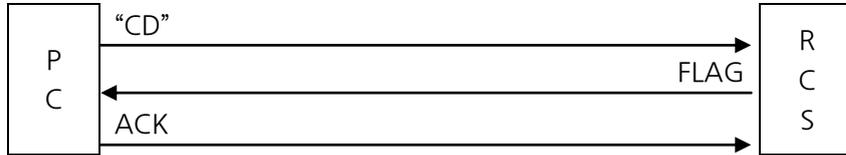
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

 CAUTION

- 반드시 “로봇 JOB 실행(CC)” 하기 전에 “실행 할 JOB 설정(DC)”으로 실행 JOB을 설정한 상태에서 실행을 하여야 합니다.
- Alarm이 발생하거나 원점수행 중이거나 완료되지 않는경우, Servo ON 및 JOB이 구동중인 상태에서는 수행하지 않습니다.

4.4.14 로봇 JOB 정지(CD)

로봇의 JOB 실행을 정지하는 명령어 입니다.



- Protocol DATA

PC	0	1	2	3	4
↓	STX	C	D	ETX	LRC
제어기	0x02	0x43	0x44	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

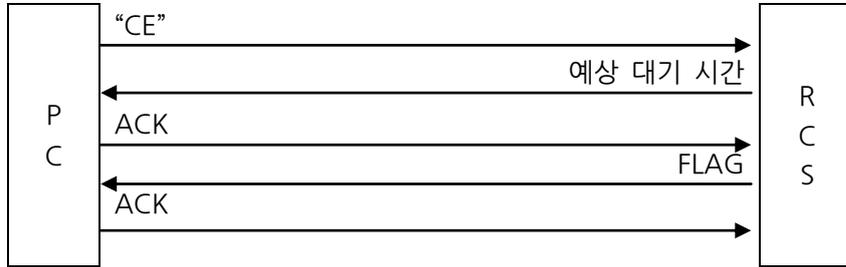
- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

 CAUTION

➤ "로봇 JOB 정지"(CD) 명령 수행시 로봇 정지후 Servo ON 상태를 유지 합니다.

#### 4.4.15 로봇 JOB 초기화(CE)

로봇 JOB 실행 STEP을 처음(Step 0)으로 설정하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	C	E	ETX	LRC
제어기	0x02	0x43	0x45	0x03	
제어기	0	1	2~4	5	6
↓	STX	상태	실행시간	ETX	LRC
PC	0x02	Flag	10	0x03	
PC	0				
↓	ACK				
제어기	0x06				
제어기	0	1	2	3	
↓	STX	Flag	ETX	LRC	
PC	0x02	0x34	0x03		
PC	0				
↓	ACK				
제어기	0x06				

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.
- 실행시간 : 예상 대기 시간:실행에 걸리는 예측 시간.10(초)

실행시간 (3Bytes)			
패킷 번호	2	3	4
내용	'0'	'1'	'0'

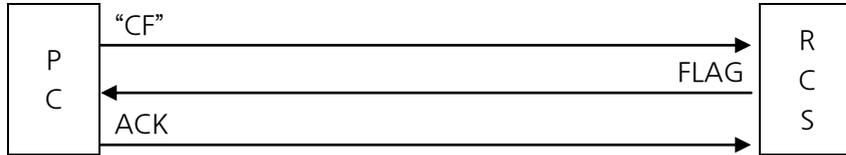
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.



**CAUTION**

- “로봇 JOB 초기화(CE)”은 JOB이 구동 중인 상태에서 동작하지 않습니다.

4.4.16 온라인 비상정지(CF)  
 로봇 온라인 비상정지 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	C	F	ETX	LRC
제어기	0x02	0x43	0x46	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

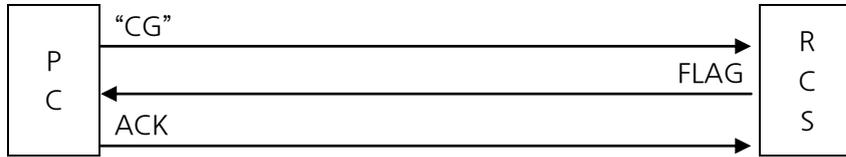
- 티치 펜던트에 E12.02 “Host Emergency”를 띄우며 알람이 발생합니다.
- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.



**CAUTION**

- 로봇 운전 중 긴급 정지 시에만 사용하시기 바랍니다.
- "온라인 Alarm 초기화" (CG)를 통해 비상 정지를 해제 할 수 있습니다.

4.4.17 온라인 Alarm 초기화 (CG)  
 제어기에 발생된 알람을 초기화 하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	C	G	ETX	LRC
제어기	0x02	0x43	0x47	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

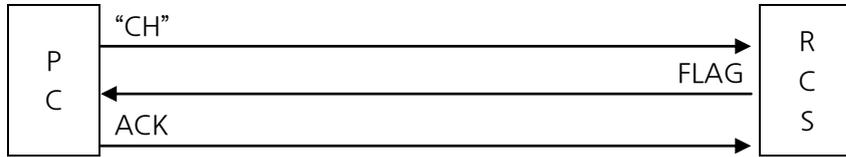
  

PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.18 온라인 MOVE STOP (CH)

동작중인 로봇의 속도를 0%로 만들어 정지시키는 명령어 입니다..



● Protocol DATA

PC	0	1	2	3	4
↓	STX	C	H	ETX	LRC
제어기	0x02	0x43	0x48	0x03	

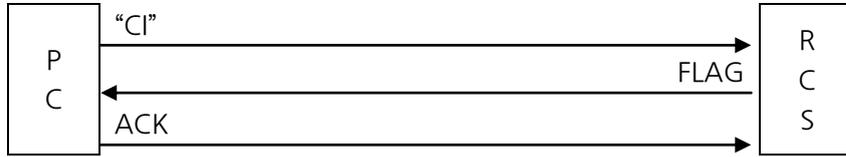
제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.19 온라인 원점 수행 정지(CI)  
 해당 채널의 원점수행 정지 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	C	I	ETX	LRC
제어기	0x02	0x43	0x49	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

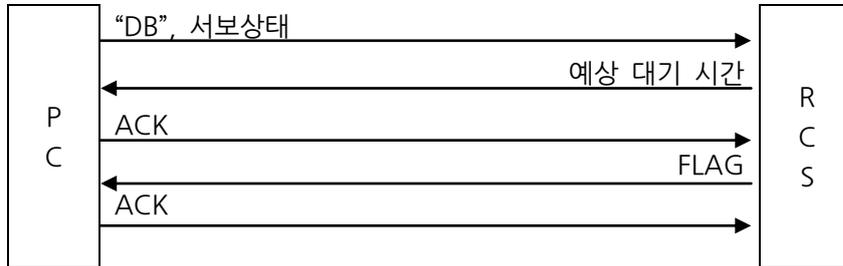
- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

 CAUTION

➤ SERVO ON 상태에서 "온라인 원점 수행 정지"(CI) 시 SERVO OFF 상태가 됩니다..

### 4.4.20 SERVO ON/OFF(DB)

해당 로봇 채널의 SERVO를 ON/OFF시키는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4	5
↓	STX	D	B	서보	ETX	LRC
제어기	0x02	0x44	0x42	상태	0x03	
제어기	0	1	2~4	5	6	
↓	STX	상태	실행시간	ETX	LRC	
PC	0x02	Flag	10	0x03		
PC	0					
↓	ACK					
제어기	0x06					

● 서보 상태 : SERVO의 ON/OFF설정(1Byte)

서보상태 (1Byte)	
값	Data Type
0	OFF
1	ON

※ 이외의 값 입력시 "0x31 프로토콜ERROR" FLAG가 발생합니다.

● 실행시간 : 예상 대기 시간:실행에 걸리는 예측 시간.10(초)

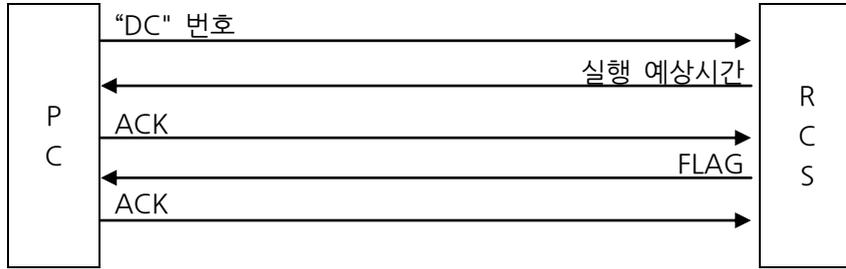
실행시간 (3Bytes)			
패킷 번호	2	3	4
내용	'0'	'1'	'0'

● STX, ETX, LRC, FLAG: "4.1통신 규칙"을 참조 하십시오.

CAUTION

➢ Alarm이 발생하거나 서보가 ON상태인경우 수행하지 않습니다.

4.4.21 실행 할 JOB설정(DC)  
 실행할 JOB파일을 설정하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3~9	10	11~14	15	16
↓	STX	D	C	0x20	실행 할 JOB 번호	0x20	ETX	LRC
제어기	0x02	0x44	0x43				0x03	
제어기	0	1	2~4	5	6			
↓	STX	상태	수행시간	ETX	LRC			
PC	0x02	Flag	10	0x03				
PC	0							
↓	ACK							
제어기	0x06							
제어기	0	1	2	3				
↓	STX	Flag	ETX	LRC				
PC	0x02	0x34	0x03					
PC	0							
↓	ACK							
제어기	0x06							

- 실행 할 JOB번호 : (0~7)설정

실행 할 JOB 번호 (1Byte)	
패킷 번호	10
내용	'번호'

- 수행시간 10 :  
실행에 걸리는 예측 시간.10(초)

수행시간 10 (3Bytes)			
패킷 번호	2	3	4
내용	'0'	'1'	'0'

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

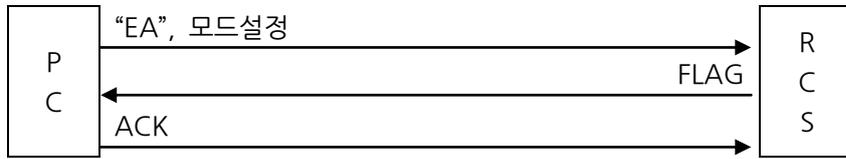
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

**CAUTION**

- 원점 수행이 되지 않은 상태에서 실행 Job(DC) 설정 시 Run fail 에러가 발생합니다.
- Servo ON 상태에서 “DC” 명령어 실행 시 Run fail 에러가 발생합니다.

4.4.22 JOB의 실행 방식 설정(EA)

JOB 동작 모드를 설정하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4	5
↓	STX	E	A	Mode	ETX	LRC
제어기	0x02	0x45	0x41	설정	0x03	
제어기	0	1	2	3		
↓	STX	상태	ETX	LRC		
PC	0x02	Flag	0x03			
PC	0					
↓	ACK					
제어기	0x06					

● Mode 설정: JOB구동 방식 설정

Mode 설정 (1Byte)

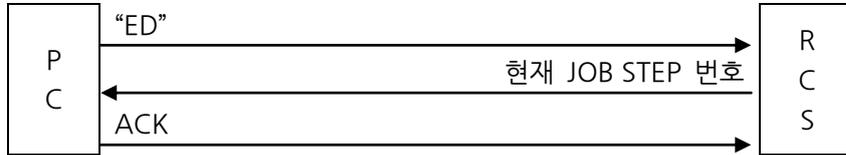
값	내용
0	Auto Run
1	Step Run

※ 이외의 값 입력시 상태 FLAG에서 "0x31 프로토콜ERROR"가 발생합니다.

● STX, ETX, LRC, FLAG: "4.1통신 규칙"을 참조 하십시오.

### 4.4.23 현재 JOB Step 확인 (ED)

현재 실행중인 JOB STEP번호 읽는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	E	D	ETX	LRC
제어기	0x02	0x45	0x44	0x03	
제어기	0	1	2~4	5	6
↓	STX	상태	현재	ETX	LRC
PC	0x02	Flag	Step 번호	0x03	
PC	0				
↓	ACK				
제어기	0x06				

● 현재 Step번호: 실행중인 STEP 번호

실행중인 STEP 번호(3Bytes)			
패킷 번호	2	3	4
내용	백의 자리	십의자리	일의자리

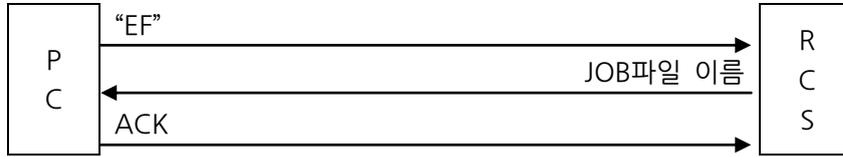
예1)현재 JOB의 27번 STEP이 실행중인 경우

패킷 번호	2	3	4
내용	0x20	'2'	'7'

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.24 현재 실행 중인 JOB이름 읽기(EF)

지정한 로봇 채널에서 현재 실행 중인 JOB 이름 읽기 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	E	F	ETX	LRC
제어기	0x02	0x45	0x46	0x03	
제어기	0	1	2~13	14	15
↓	STX	상태	설정 된	ETX	LRC
PC	0x02	Flag	JOB 번호	0x03	
PC	0				
↓	ACK				
제어기	0x06				

- 설정 된 JOB 번호: 현재 수행중인 JOB 파일 이름  
 - JOB 번호 + '.' + 'J' + 'O' + 'B'의 형태로 전송합니다.  
 - JOB 번호 외 나머지 부분에는 0x20 ' '이 들어가게 됩니다.

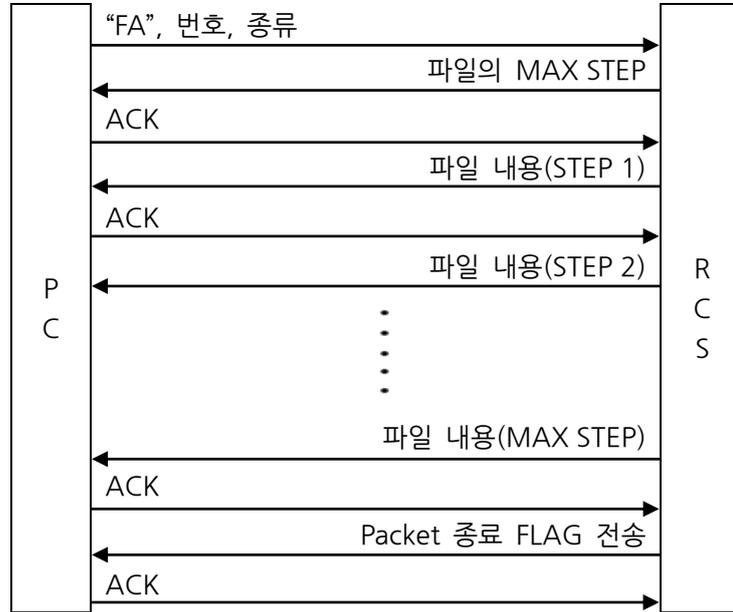
설정 된 JOB 번호 (12Bytes)						
패킷 번호	0~7	8	9	10	11	12
내용	0x20	설정된 번호	'.'	'J'	'O'	'B'

예 1) 3 번 JOB 이 설정 된 경우

패킷 번호	0~7	8	9	10	11	12
내용	0x20	'3'	'.'	'J'	'O'	'B'

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.25 제어기에서 PC로 파일 전송(FA)  
 제어기에서 PC로 로봇 JOB, PLC JOB 파일을 전송하는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3~10	11	12	13	14~16	17	18
↓	STX	F	A	0x20	보낼 JOB 번호	0x20	JOB 종류	0x20	ETX	LRC
제어기	0x02	0x46	0x41						0x03	

제어기	0	1	2~n	n+1	n+2
↓	STX	상태	JOB 정보	ETX	LRC
PC	0x02	Flag	한 Step	0x03	

PC	0
↓	ACK
제어기	0x06

제어기	마지막 Step 까지
PC	0
↓	ACK
제어기	0x06

제어기	0	1	2	3
↓	STX	Flag	ETX	LRC
PC	0x02	0x34	0x03	

PC	0
↓	ACK
제어기	0x06

- 보낼 JOB 번호 : 상위로 보낼 JOB번호(0~7)

보낼 JOB 번호 (1Byte)	
패킷 번호	11
내용	JOB 번호

- JOB 종류 : JOB 종류 선택

JOB 종류 (1Byte)	
값	내용
'J'(0x4A)	Robot JOB
'P'(0x50)	PLC JOB

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

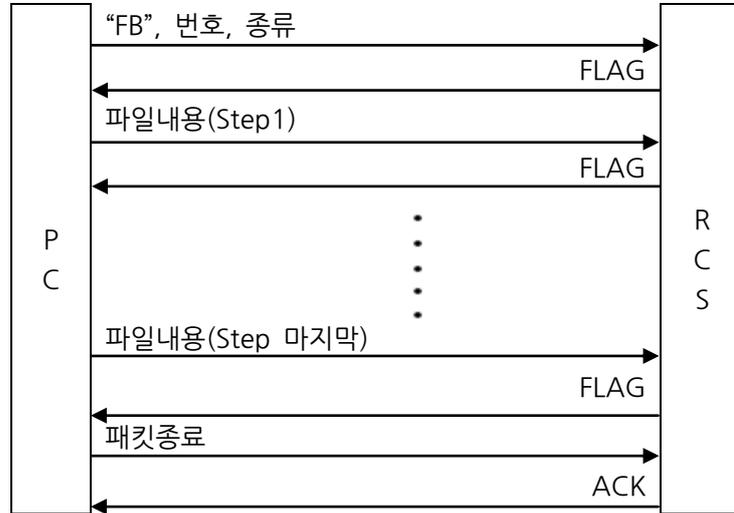


**CAUTION**

- 보낼 JOB번호가 허용범위를 벗어난 경우 Run fail 에러가 발생합니다.  
(로봇 JOB : 0~7, PLC JOB : 0~4)
- JOB 종류 선택이 잘못된 경우 Run Fail에러가 발생합니다.

4.4.26 파일 저장(FB)

PC에서 제어기로 파일 전송 명령어 입니다.



● Protocol DATA

PC	0	1	2	3~10	11	12	13	14~16	17	18
↓	STX	F	B		저장 될				ETX	
제어기	0x02	0x46	0x42	0x20	번호 설정	0x20	JOB 종류	0x20	0x03	LRC

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0	1	2~n	n+1	n+2
↓	STX	상태	JOB 정보	ETX	LRC
제어기	0x02	Flag	한 Step	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

⋮

마지막 Step까지
------------

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0	1	2	3
↓	STX	상태	ETX	LRC
제어기	0x02	Flag	0x03	

제어기	0
↓	ACK
PC	0x06

- 저장 될 번호 설정 : 저장 할 위치의 JOB번호(0~7)

저장 될 번호 설정 (1Byte)	
패킷 번호	11
내용	JOB 번호

- JOB 종류 : JOB 종류 선택

JOB 종류 (1Byte)	
값	내용
'J'(0x4A)	Robot JOB
'P'(0x50)	PLC JOB

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

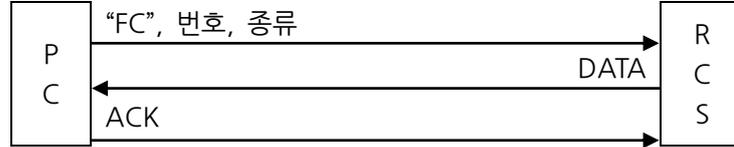
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

 CAUTION

- 저장 될 JOB번호가 허용범위를 벗어난 경우 Run fail 에러가 발생합니다.  
(로봇 JOB : 0~7, PLC JOB : 0~4)
- JOB 종류 선택이 잘못된 경우 Run Fail에러가 발생합니다.

4.4.27 파일 검색(FC)

제어기내 파일을 검색하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3~10	11	12	13	14~15	16	17
↓	STX	F	C	0x20	검색 할	0x20	JOB 종류	0x20	ETX	LRC
제어기	0x02	0x46	0x43		번호 설정				0x03	
제어기	0	1	2	3	4					
↓	STX	상태	유/무	ETX	LRC					
PC	0x02	Flag	확인	0x03						
PC	0									
↓	ACK									
제어기	0x06									

- 검색 할 번호 설정 검색 할 JOB번호(0~7) 또는 PLC JOB번호(0~3)

검색 할 번호 설정 (1Byte)

패킷 번호	11
내용	JOB 번호

- JOB 종류 : JOB 종류 선택

JOB 종류 (1Byte)

값	내용
'J'(0x4A)	Robot JOB
'P'(0x50)	PLC JOB

- 유/무 확인 : JOB의 유/무 확인

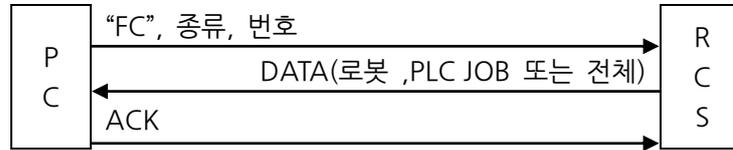
유/무 확인 (1Byte)

값	내용
'0'	없음
'1'	있음

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.28 파일 정보 요청(FD)

제어기내 파일 정보를 요청하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4~14	15	16	17		
↓	STX	F	D	0	0x20	JOB	ETX	LRC		
제어기	0x02	0x46	0x44	0x30		종류	0x03			
제어기	0	1	2~13	14~29	30~41	42~57	58~n	n+1	n+2	
↓	STX	상태	JOB 번호	DUMMY	JOB 번호	DUMMY	...	ETX	LRC	
PC	0x02	Flag						0x03		
PC	0									
↓	ACK									
제어기	0x06									

● JOB 종류 : JOB 종류 선택

JOB 종류 (1Byte)

값	내용
'B'(0x42)	Robot JOB
'C'(0x43)	PLC JOB
'*(0x2A)	전체

● JOB 번호 : JOB 번호(0~7)

JOB 번호 (13bytes)

패킷 번호	2~8	9	10	11	12	13
내용	0x20	번호	.	J	O	B

● Dummy : 패킷 번호 14~29 과 42~57 은 동일한 값을 전송합니다.

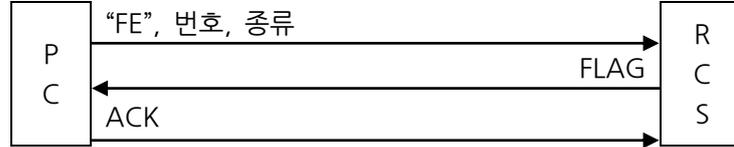
Dummy (16Bytes)

패킷 번호	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
내용	1	2	3	4	5	6	9	7	0	5	1	7	1	4	2	0

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.29 파일 삭제(FE)

제어기에 저장되어 있는 파일을 삭제하는 명령어 입니다.



● Protocol Data

PC	0	1	2	3~10	11	12	13	14~16	17	18
↓	STX	F	E	0x20	삭제 할 번호 설정	0x20	JOB 종류	0x20	ETX	LRC
제어기	0x02	0x46	0x45						0x03	
제어기	0	1	3	4						
↓	STX	상태	ETX	LRC						
PC	0x02	Flag	0x03							
PC	0									
↓	ACK									
제어기	0x06									

- 저장 할 번호 설정 : 저장 할 위치의 JOB번호(0~7)

저장 할 번호 설정 (1Byte)

패킷 번호	11
내용	JOB 번호

- JOB 종류 : JOB 종류 선택

JOB 종류 (1Byte)

값	내용
'J'(0x4A)	Robot JOB
'P'(0x50)	PLC JOB

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

### 4.4.30 파일 복사(FF)

동일 채널에서 JOB 파일을 복사하는 명령어 입니다.



● Protocol Data

PC	0	1	2	3~10	11	12	13	14~23	24	25~28	29	30
↓	STX	F	F	0x20	복사 할 JOB 번호	0x20	JOB 종류	0x20	복사 될 JOB 번호	0x20	ETX	LRC
제어기	0x02	0x46	0x46								0x03	
제어기	0	1	3	4								
↓	STX	상태	ETX	LRC								
PC	0x02	Flag	0x03									
PC	0											
↓	ACK											
제어기	0x06											

● 복사 할 JOB번호 : 복사 할 위치의 JOB번호

복사 할 JOB번호 (1Byte)	
패킷 번호	11
내용	JOB 번호

● JOB 종류 : JOB 종류 선택

JOB 종류 (1Byte)	
값	내용
'J'(0x4A)	Robot JOB
'P'(0x50)	PLC JOB

● 복사 될 JOB번호 : 복사 될 위치의 JOB번호

복사 될 JOB번호 (1Byte)	
패킷 번호	24
내용	JOB 번호

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

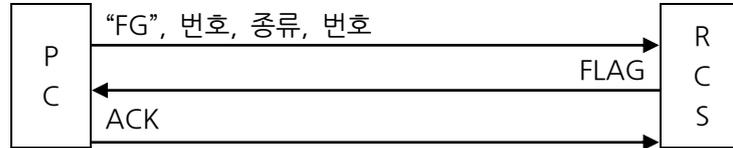
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.



**CAUTION**

- PLC JOB이 동작중인 경우 Run Fail 에러를 발생합니다.
- 존재하고 있는 로봇 또는 PLC JOB번호에 복사 할 경우 Run Fail 에러를 발생합니다.

4.4.31 제어기 JOB파일 이름 변경 (FG)  
JOB 파일 이름을 변경하는 명령어 입니다.



● Protocol Data

PC	0	1	2	3~10	11	12	13	14~23	24	25~28	29	30
↓	STX	F	G	0x20	변경 전 JOB 번호	0x20	JOB 종류	0x20	변경 후 JOB 번호	0x20	ETX	LRC
제어기	0x02	0x46	0x47								0x03	
제어기	0	1	3	4								
↓	STX	상태	ETX	LRC								
PC	0x02	Flag	0x03									
PC	0											
↓	ACK											
제어기	0x06											

- 변경 전 JOB 번호 : 변경 할 JOB번호.

변경 전 JOB 번호 1 (1Byte)	
패킷 번호	11
내용	JOB 번호

- JOB 종류 : JOB 종류

JOB 종류 (1Byte)	
값	내용
'J'(0x4A)	Robot JOB
'P'(0x50)	PLC JOB

- 변경 후 JOB 번호 : 변경 후 JOB번호

변경 후 JOB 번호 (1Byte)	
패킷 번호	24
내용	JOB 번호

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

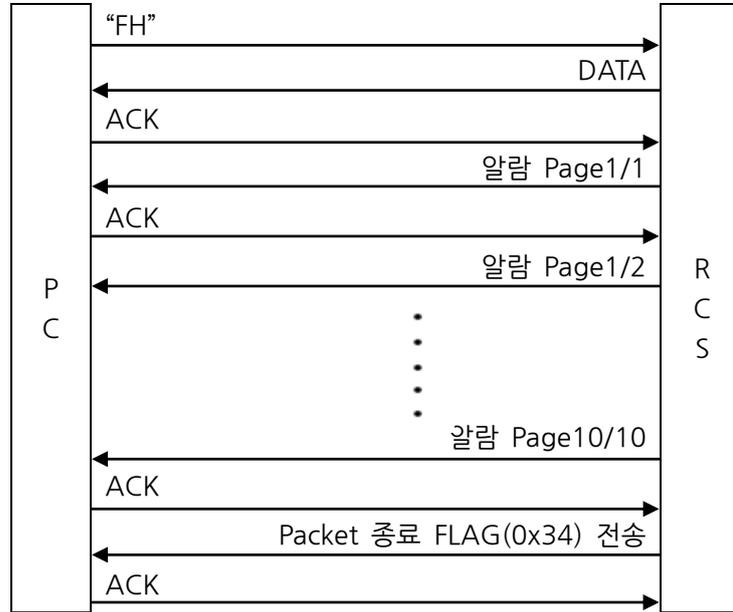
※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.



CAUTION

- PLC JOB이 동작중인 경우 Run Fail 에러를 발생합니다.
- 존재하고 있는 로봇 또는 PLC JOB번호로 변경 할 경우 Run Fail 에러를 발생합니다.

4.4.32 저장 되어 있는 알람 내용 읽기(FH)  
RCS 제어기에 저장되어 있는 알람 내용 읽기 명령어 입니다.



● Protocol Data

PC	0	1	2	3	4~33	34	35
↓	STX	F	H	''	Data1	ETX	LRC
제어기	0x02	0x46	0x48	0x20		0x03	

제어기	0	1	2~31	32	33
↓	STX	상태	Data 2	ETX	LRC
PC	0x02	Flag		0x03	

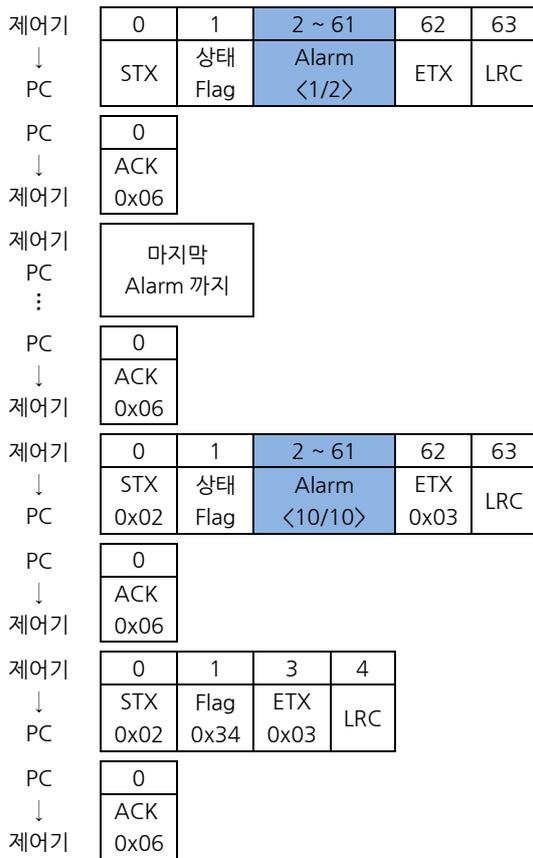
PC	0
↓	ACK
제어기	0x06

제어기	0	1	2 ~ 61	62	63
↓	STX	상태	Alarm	ETX	LRC
PC	0x02	Flag	<1/1>	0x03	

PC	0
↓	ACK
제어기	0x06



- Data 1 : ERR message 파일 요청

Data 1 (30Bytes)	
패킷 번호	2~31
내용	"alarm_history.txt"

※ 30Byte중 "alarm\_history.txt"부분을 제외한 빈(13Byte)공간은 SPACE(0x20) 문자를 삽입하여 주십시오.

- DATA 2 : 메시지 창 분류

Data 2 (30Bytes)	
패킷 번호	2~31
내용	"NO.₩tERROR TIME₩tERROR MSG₩t(CODE)"

- Alarm : 제어기에 저장되어 있는 알람 내용을 반환 합니다.

알람 페이지

패킷 번호	0	1 2	3	4 5	6	7	8
내용	< 0x3C	현재 페이지	/ 0x2F	번호	> 0x3E	₩t 0x09	'' 0x20

- 현재 페이지 : Teach Pendant의 VIEW-> ALARM->HISTORY의 페이지
- 번호 : 페이지에 저장 되어있는 번호

동작 시간

패킷 번호	9	15 19	20	21	22 23	24	25	26 27	28	29	30
내용	W- Time:	시간	H 0x48	'' 0x20	분	M 0x4D	'' 0x20	초	S 0x53	₩t 0x09	'' 0x20

- 시간 : 제어기 동작 시간의 시단위.
- 분 : 제어기 동작 시간의 분단위.
- 초 : 제어기 동작 시간의 초단위.

알람 내용

패킷 번호	31~50	51	52	53	54	55 56	57	58 59	60	61
내용	알람 이름	₩t 0x09	'' 0x20	( 0x28	E 0x45	메인 번호	'' 0x2E	세부번호	) 0x29	'' 0x20

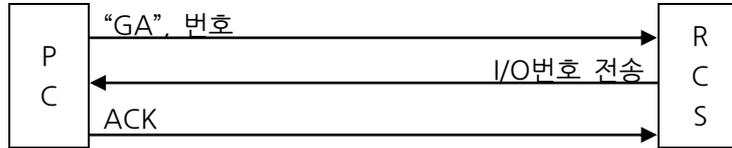
- 알람 이름 : 저장 되어있는 알람 이름.
- 메인 번호 : 알람번호의 메인 번호.
- 세부 번호 : 알람번호 메인의 세부번호.

※ 알람 내용 저장에 대한 자세한 내용 및 해결방법은 “RCS8000메뉴얼 13장 ALARM” 를 참고 바랍니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하시기 바랍니다.

4.4.33 I/O의 Byte단위 읽기 (GA)

I/O카드 입력 접점 상태 정보 읽기 명령어 입니다.



● Protocol Data

PC	0	1	2	3	4	5	6	7		
↓	STX	G	A	0	I/O 번호	ETX	LRC			
제어기	0x02	0x47	0x41	0x30		0x03				
제어기	0	1	2	3	4	5	6	7	8	9
↓	STX	상태	B	Y	T	E	접점 상태	ETX	LRC	
PC	0x02	Flag						0x03		
PC	0									
↓	ACK									
제어기	0x06									

- I/O번호: 입력 PORT를 선택합니다.

I/O 번호 (2Byte)		
패킷 번호	4	5
내용	백의 자리	십의자리

- I/O접점은 B000 ~ B410입니다.

※ 비어있는 패킷은 '(0x20) 또는 '0'(0x30)으로 작성합니다.

- 접점 상태 : 선택한 입력 PORT 접점의 상태를 2의 배수로 반환합니다.

접점 상태 (2Bytes) 단위 : HEX		
패킷 번호	6	7
내용	하위 비트	상위 비트
	설정된 I/O번호의 0~3접점	설정된 I/O번호의 4~7접점

- IN : 선택한 I/O접점에 대한 값

※ 접점 데이터는 ON:1 , OFF:0 입니다.

예1) Bit B001, B006 두 접점이 On인 경우 제어기에서 패킷 6번 "2" 7번 "4"를 송신합니다.

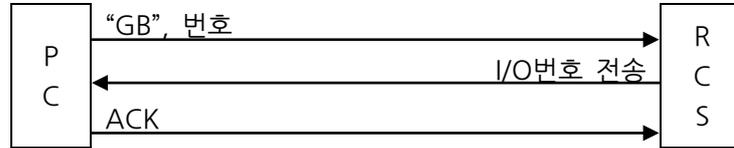
B00접점 상태	7	6	5	4	3	2	1	0
(HEX : 0x42)	0	1	0	0	0	0	1	0

송신 패킷 표시 방법	패킷 번호	6(접점 상태 하위 비트) 2						7(접점 상태 상위 비트) 4									
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.34 I/O의 Bit단위 읽기 (GB)

I/O카드 출력 접점 상태 정보 읽기 명령어 입니다.



● Protocol Data

PC	0	1	2	3	4	5	6	7	8
↓	STX	G	B	0	I/O 번호			ETX	LRC
제어기	0x02	0x47	0x42	0x30				0x03	

제어기	0	1	2	3	4	5	6	7	8	9
↓	STX	상태	B	I	T	B	접점 상태		ETX	LRC
PC	0x02	Flag							0x03	

PC	0
↓	ACK
제어기	0x06

- I/O 번호: 입력 PORT를 선택합니다.

I/O 번호 (3Bytes)			
패킷 번호	4	5	6
내용	백의 자리	십의자리	일의 자리

- I/O접점은 B000 ~ B410입니다.

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- 접점 상태 : 선택한 접점에 대해서 2의 배수로 반환합니다. 이외의 접점은 확인하지 않습니다.

접점 상태 (2Bytes) 단위 : HEX		
패킷 번호	6	7
내용	하위 비트	상위 비트
	설정된 I/O번호의 0~3접점	설정된 I/O번호의 4~7접점

- IN : 선택한 I/O접점에 대한 값

예1) Bit B006접점 확인 BITB04 (예: B006의 접점 상태 HEX 0x42)

B00접점 상태	7	6	5	4	3	2	1	0
(HEX : 0x42)	0	1	0	0	0	0	1	0

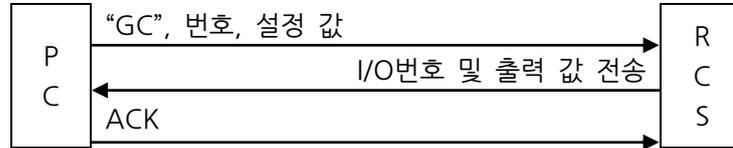
송신 패킷 표시 방법	패킷 번호	6(접점 상태 하위 비트) 0							7(접점 상태 상위 비트) 4								
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

4.4.35 I/O의 Bit단위 쓰기(GC)

I/O 접점 Bit단위 접점 쓰기 명령어 입니다.



● Protocol Data

PC	0	1	2	3	4	5	6	7	8	9
↓	STX	G	C	I/O		I/O		접점상태	ETX	LRC
제어기	0x02	0x47	0x43	백,십의자리		일의 자리		0x31	0x03	
제어기	0	1	2	3						
↓	STX	상태	ETX	LRC						
PC	0x02	Flag	0x03							
PC	0									
↓	ACK									
제어기	0x06									

● I/O 백, 십의 자리.

I/O 번호 (2Bytes)		
패킷 번호	3	4
내용	백의자리	십의 자리

● I/O 일의 자리.

I/O 번호 (2Bytes)		
패킷 번호	5	6
내용	' '또는 '0'	일의자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

● 접점 상태 : ON/OFF설정

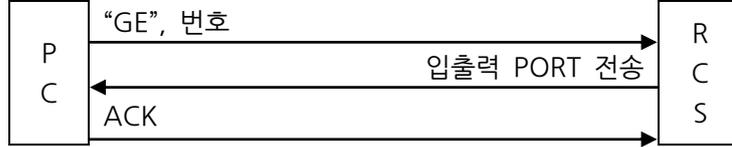
접점상태 (1Byte)	
값	Data Type
0	OFF
1	ON

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하시기 바랍니다.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

4.4.36 I/O의 2Byte단위 읽기 (GE)

제어기에 설정 되어 있는 입/출력 BOARD 설정 값을 2Byte단위로 읽는 명령어 입니다.



● Protocol Data

PC	0	1	2	3	4	5	6	7				
↓	STX	G	E	0	I/O 번호		ETX	LRC				
제어기	0x02	0x47	0x45	0x30			0x03					
제어기	0	1	2	3	4	5	6	7	8	9	10	11
↓	STX	상태	W	O	R	D	I/O 번호	I/O 번호+10	ETX	LRC		
PC	0x02	Flag					접점 상태	접점 상태	0x03			
PC	0											
↓	ACK											
제어기	0x06											

- I/O 번호: 입력 PORT를 선택합니다.( I/O접점은 B000 ~ B410입니다.)

I/O 번호(2Byte)		
패킷 번호	4	5
내용	백의 자리	십의자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- I/O번호 접점 상태 : 선택 한 접점의 상태 값을 2의배수로 반환합니다.

I/O 번호 접점 상태 (2Bytes x 2)				
패킷 번호	6	7	8	9
내용	설정된 I/O번호의 하위 비트	설정된 I/O번호의 상위 비트	설정된 I/O번호+10의 하위 비트	설정된 I/O번호+10의 상위 비트
	설정된 I/O번호의 0~3접점	설정된 I/O번호의 4~7접점	설정된 I/O번호+10의 0~3접점	설정된 I/O번호+10의 4~7접점

예1) I/O번호를 10번 선택한 경우. 접점 6,7, 8, 9에 "2", "6", "1", "9"를 반환합니다.

B10접점 상태	7	6	5	4	3	2	1	0	B11접점 상태	7	6	5	4	3	2	1	0
(HEX : 0x62)	0	1	1	0	0	0	1	0	(HEX : 0x19)	0	0	0	1	1	0	0	1

송신 패킷 표시 방법	패킷 번호	6(접점 상태 하위 비트) 2							7(접점 상태 상위 비트) 6								
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0

송신 패킷 표시 방법	패킷 번호	8(접점 상태 하위 비트) 1							9(접점 상태 상위 비트) 9								
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.37 I/O의 2Byte단위 쓰기 (GF)

제어기에 설정 되어 있는 입/출력 BOARD 설정 값을 2Byte단위로 쓰는 명령어 입니다.



● Protocol Data

PC	0	1	2	3	4	5	6	7	8	9	10
↓	STX	G	F	I/O 번호		출력 값 1		출력 값 2		ETX	LRC
제어기	0x02	0x47	0x46							0x03	
제어기	0	1	2	3							
↓	STX	상태	ETX	LRC							
PC	0x02	Flag	0x03								
PC	0										
↓	ACK										
제어기	0x06										

- I/O 번호 : 출력 할 I/O PORT를 설정합니다.

I/O 번호 (2Bytes)		
패킷 번호	3	4
내용	백의자리	십의 자리

※ 비어있는 패킷은 '(0x20) 또는 '0'(0x30)으로 작성합니다.

- 출력 값1, 2: 선택 한 접점에 출력 할 값을 설정합니다.

출력 값 (2Bytes x 2)				
패킷 번호	5	6	7	8
내용	하위 비트	상위 비트	하위 비트	상위 비트
	설정 할 I/O상태 의 0~3접점	설정 할 I/O상태 의 4~7접점	설정 할 I/O 번호 +10의 0~3접점	설정 할 I/O 번호 +10의 4~7접점

- 예1) I/O번호 : 4, 출력 값1 : 2, 6 출력 값2 : 1, 9인 경우

송신 패킷 표시 방법	패킷 번호	5(접점 상태 하위 비트) 2							6(접점 상태 상위 비트) 6								
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0

송신 패킷 표시 방법	패킷 번호	7(접점 상태 하위 비트) 1							8(접점 상태 상위 비트) 9								
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1

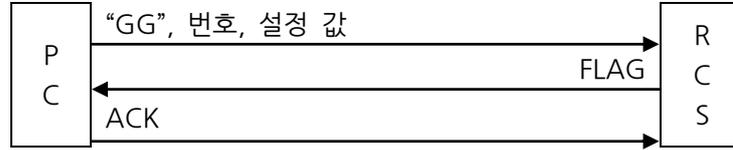
B04접점 상태	7	6	5	4	3	2	1	0
(HEX : 0x62)	0	1	1	0	0	0	1	0

B05접점 상태	7	6	5	4	3	2	1	0
(HEX : 0x19)	0	0	0	1	1	0	0	1

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하시기 바랍니다.

4.4.38 I/O의 Byte단위 쓰기 (GG)

I/O 접점을 Byte단위로 변경하는 명령어입니다.



● Protocol Data

PC	0	1	2	3	4	5	6	7	8
↓	STX	G	G	I/O 번호		값 설정		ETX	LRC
제어기	0x02	0x47	0x47					0x03	
제어기	0	1	2	3					
↓	STX	상태	ETX	LRC					
PC	0x02	Flag	0x03						
PC	0								
↓	ACK								
제어기	0x06								

- I/O 설정 : 출력 할 I/O PORT를 설정합니다.

I/O 설정 (2Bytes)		
패킷 번호	3	4
내용	백의자리	십의 자리

- 값 설정 : 선택 한 접점에 출력 할 값을 설정합니다

값 설정 (2Bytes) 단위 : HEX		
패킷 번호	5	6
내용	하위 비트	상위 비트
	설정 할 I/O상태의 0~3접점	설정 할 I/O상태의 4~7접점

예1) I/O 번호 : 4(B04), 접점 값 설정 : 4, 1인 경우

송신 패킷 표시 방법	패킷 번호	5(접점 상태 하위 비트) 4								6(접점 상태 상위 비트) 1							
	접점 번호	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	접점 상태	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1

B04접점 상태 (HEX : 0x14)	7	6	5	4	3	2	1	0
	0	0	0	1	0	1	0	0

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하시기 바랍니다.

### 4.4.39 통신 속도 설정(KA)

통신 속도를 설정 합니다.



● Protocol Data

PC	0	1	2	3	4	5	6
↓	STX	K	A	COM Port	속도 설정	ETX	LRC
제어기	0x02	0x4B	0x41	설정		0x03	
제어기	0	1	8	9			
↓	STX	상태	ETX	LRC			
PC	0x02	Flag	0x03				
PC	0						
↓	ACK						
제어기	0x06						

- COM Port : 설정: 변경 할 통신 포트 설정..

COM Port 설정 (1Byte)	
값	설명
0	COM1
1	COM2
2	COM3

- 속도 설정 : 통신 속도 설정..

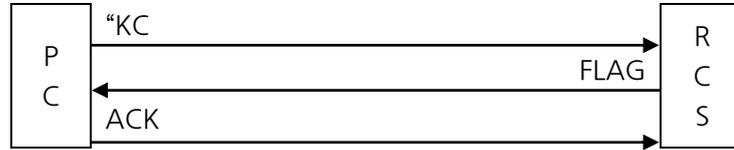
속도 설정 (1Byte)	
값	설명
0	9,600 bps
1	19,200 bps
2	38,400 bps
3	115,200 bps

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

 CAUTION

- 통신속도 변경 시 반드시 전원을 ON/OFF 바랍니다.
- COM1은 통신 속도를 0 (9,600bps)에서 1~3로 변경 시 Teach Pendant가 동작하지 않습니다.

4.4.40 제어기 Type 확인(KC)  
 제어기 종류를 확인 합니다.

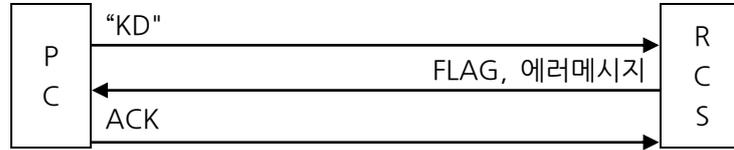


● Protocol Data

PC	0	1	2	3	4		
↓	STX	K	C	ETX	LRC		
제어기	0x02	0x4B	0x43	0x03			
제어기	0	1	2	3	4	5	6
↓	STX	상태	0	8	0	1	ETX
PC	0x02	Flag					0x03
PC	0						
↓	ACK						
제어기	0x06						

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.41 통신상 에러 확인(KD)  
 프로토콜 통신 중 발생한 에러를 확인합니다.

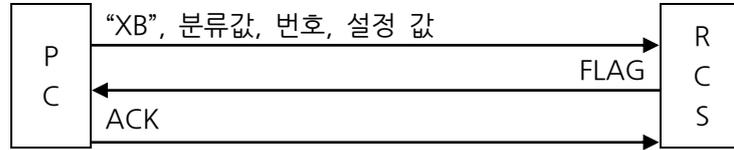


● Protocol Data

PC	0	1	2	3	4
↓	STX	K	D	ETX	LRC
제어기	0x02	0x4B	0x44	0x03	
제어기	0	1	n	n+1	n+2
↓	STX	상태	에러	ETX	LRC
PC	0x02	Flag	메시지	0x03	
PC	0				
↓	ACK				
제어기	0x06				

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.42 GAIN을 제외한 파라미터값 변경 (XB)  
 GAIN을 제외한 파라미터의 값을 변경합니다..



● Protocol Data

PC	0	1	2	3	4	5	6~14	15	16
↓	STX	X	B	분류 값	파라미터		위치 값	ETX	LRC
제어기	0x02	0x58	0x42		번호			0x03	
제어기	0	1	8	9					
↓	STX	상태	ETX	LRC					
PC	0x02	Flag	0x03						
PC	0								
↓	ACK								
제어기	0x06								

● 분류 값 : 번호에 따라 구성된 파라미터 분류 설정.

분류 값 (1Bytes)

번호	내용	
	1 차분류	2 차 분류
0	SERVO	AMP
1	SERVO	-
2	SERVO	PROTECT
3	MECH	-
4	OPER	MODE
5	OPER	JOG
6	OPER	DEF
7	OPER	COM
8	OPER	ETC
9	OPER	IP
A	I/O	INPUT
B	I/O	OUTPUT

- 파라미터 번호 : 분류 값에 해당되는 파라미터의 번호.

파라미터 번호(2bytes)

패킷 번호	4	5
내용	파라미터 번호	
	십의 자리	일의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

분류 값	번호	이름
AMP (0)	0	AMP
	1	MOTOR ID
	2	ENC TYPE
	3	ENC PLS
	4	MOT_TYPE
	5	ENC_DIR
	6	R
	7	L
	8	R_I
	9	MAX_I
	10	BACK_EMF
	11	Jm
	12	Kt
	13	MAX_RPM
	14	Kcp
	15	Kci
	16	POLE
	17	Z_OFFSET
	18	H_OFFSET
19	ELC_PIT	

분류 값	번호	이름
PROTECT (2)	0	OVS
	1	OVT
	2	OVL
	3	BRK_S
	4	BRK_R
	5	MAIN_PWR
	6	ALM_SEQ
	7	SVO_SEQ
	8	OPEN_CHK

분류 값	번호	이름
MECH (3)	0	MIN_LMT
	1	MAX_LMT
	2	LMT_RPM
	3	LMT_TRQ
	4	ORG_OFS
	5	ABS_OFS
	6	CAL_OFS
	7	END_POS
	8	MOV_MOT
	9	MOV_MECH
	10	MOV_POL
	11	MPG_PLS0
	12	MPG_MOV0
	13	MPG_PLS1
	14	MPG_MOV1
15	T_CYCLE	

분류 값	번호	이름
MODE (4)	0	AUTO_PLC
	1	AUTO_ORG
	2	S_MODE
	3	MOV_AVG
	4	ORG_RULE
	5	MON_MODE

분류 값	번호	이름
JOG (5)	0	JOG_SPD0
	1	JOG_SPD1
	2	JOG_SPD2
	3	JOG_SPD3
	4	JOG_RES0
	5	JOG_RES1
	6	JOG_RES2
	7	JOG_RES3

분류 값	번호	이름
DEF (6)	0	DFT_SPD
	1	DFT_ACC
	2	DFT_DEC
	3	ORG_SPD0
	4	ORG_SPD1
	5	IO_SPD0
	6	IO_SPD1
	7	IO_SPD2
	8	IO_SPD3

분류 값	번호	이름
COM (7)	0	BITRATE1
	1	BITRATE2
	2	BITRATE3
	3	DATAMODE
	4	MY_ID

분류 값	번호	이름
IP (9)	0	PNETSIZE
	1	IP_ADD1
	2	IP_ADD1
	3	IP_ADD1
	4	IP_ADD1
	5	G/W_ADD1
	6	G/W_ADD1
	7	G/W_ADD1
	8	G/W_ADD1

분류 값	번호	이름
OUTPUT (B)	0	ALARM
	1	READY
	2	ORIGIN
	3	IN_POS
	4	ALARM0
	5	ALARM1
	6	ALARM2
	7	ALARM3
	8	BRAKE
	9	PGMRUN
	10	PGM_NUM0
	11	PGM_NUM1
	12	PGM_NUM2
	13	SVON OUT
	14	WARN
	15	WARNO
	16	WARN1
17	WARN2	

분류 값	번호	이름
ETC (8)	0	FLO_ERR
	1	INPOS
	2	ROB_PGM
	3	PLC_PGM
	4	INI_TRQ
	5	JOG_SV
	6	ORG_SV
	7	ORG_TRQ
	8	BCD_READ
	9	BACKLASH
	10	USERMODE
	11	SENSOR
	12	HW_LIMIT
	13	TP_TYPE
	14	TP_LINE
15	JOG_DMAN	

분류 값	번호	이름
INPUT (A)	0	ROB_RUN
	1	PLC_RUN
	2	STOP
	3	RESET
	4	SVON
	5	SVOFF
	6	ORIGIN
	7	STEP_RUN
	8	PGM_SEL
	9	PGM_SELO
	10	PGM_SEL1
	11	PGM_SEL2
	12	JOG+
	13	JOG-
	14	JOG_SET0
	15	JOG_SET1
	16	JOG_MODE
	17	IOPOS0
	18	IOPOS1
	19	IOPOS2
	20	IOPOS3
	21	IOPOS4
	22	IOPOS5
	23	IOPOS6
	24	IOPOS7
	25	IOSPD0
	26	IOSPD1
	27	CW S/W
	28	CCW S/W
	29	ORG S/W
	30	MPG_RATE
31	MOVT_ST	

- 설정 값 : 분류 값에서 설정 한 1, 2차 분류 이후 3차 분류..

위치 값 (9Bytes)

예 1) 설정 값	패킷 번호	6	7	8	9	10	11	12	13	14
1234567	내용	0x20	0x20	1	2	3	4	5	6	7

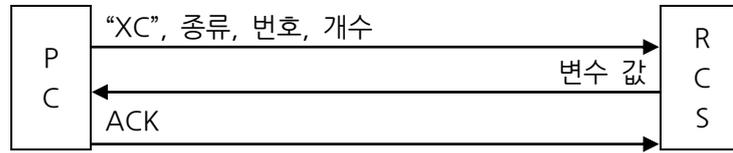
※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

4.4.43 변수의 종류 및 개수를 선택하여 값 확인 (XC)

원하는 번호부터 정수형 및 위치형변수의 개수를 선택하여 값 확인하는 명령어 입니다.



● Protocol Data

변수 번호가 3자리인 경우.

PC	0	1	2	3	4	5	6	7	8	9
↓	STX	X	C	변수	확인 할 변수의		확인 할	ETX	LRC	
제어기	0x02	0x58	0x43	종류	번호 설정		변수 개수	0x03		

제어기	0	1	2~11	4	5~n*	n+1	n+2	n+3
↓	STX	상태	위치, 정수	;	...	;	ETX	LRC
PC	0x02	Flag	변수 값	0x3B		0x3B	0x03	

PC	0
↓	ACK
제어기	0x06

변수 번호가 4자리인 경우.

PC	0	1	2	3	4	5	6	7	8	9	10
↓	STX	X	C	변수	확인 할 변수의 번호 설정			확인 할	ETX	LRC	
제어기	0x02	0x58	0x43	종류				변수 개수	0x03		

제어기	0	1	2~11	4	5~n*	n+1	n+2	n+3
↓	STX	상태	위치, 정수	;	...	;	ETX	LRC
PC	0x02	Flag	변수 값	0x3B		0x3B	0x03	

PC	0
↓	ACK
제어기	0x06

● 확인 할 변수 종류 : 확인 할 변수 종류를 설정합니다.

확인 할 변수 종류(1Byte)

값	설정
0	정수형 변수
1	위치형 변수

- 확인 할 변수의 번호 설정: 확인 할 변수 번호를 설정합니다.  
3자리, 4자리수 자리수 설정 없이 번호를 설정 할 수 있습니다.

확인 할 변수의 번호 설정(3Bytes)			
패킷 번호	4	5	6
내용	위치, 정수형 번호 설정		
	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

확인 할 변수의 번호 설정(4Bytes)				
패킷 번호	4	5	6	7
내용	위치, 정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- 확인 할 변수 개수 : 확인 할 변수의 개수를 설정합니다.

확인 할 변수 개수(1Byte)	
패킷 번호	7 또는 8
내용	일의 자리

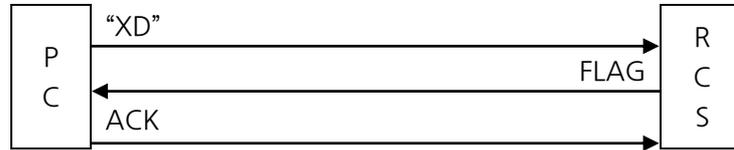
※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

4.4.44 Back up RAM에 파라미터 저장(XD)

백업 RAM에 파라미터를 저장 시키는 명령어 입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	D	ETX	LRC
제어기	0x02	0x58	0x44	0x03	

제어기	0	1	2~4	5	6
↓	STX	상태	실행시간	ETX	LRC
PC	0x02	Flag	10	0x03	

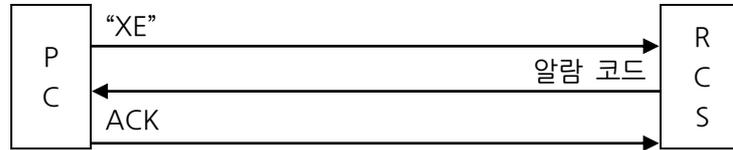
PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

4.4.45 통신 상 에러확인(XE)

제어기에 발생한 알람코드를 확인하는 명령어입니다.



● Protocol DATA

PC ↓ 제어기	0	1	2	3	4
	STX 0x02	X 0x58	E 0x45	ETX 0x03	LRC

제어기 ↓ PC	0	1	2	3	4	5	6	7	8	9	10
	STX 0x02	상태 Flag	E	R	R	O	R	알람 코드		ETX 0x03	LRC

PC ↓ 제어기	0
	ACK 0x06

● 알람 코드 : 현재 발생한 알람코드번호.

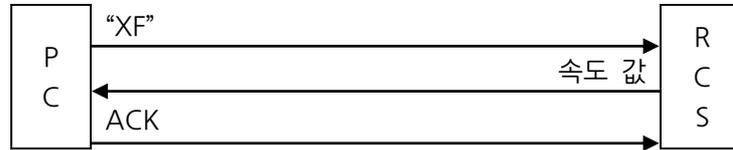
알람코드 (2Bytes)		
패킷 번호	7	8
내용	백의자리	십의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

#### 4.4.46 현재 속도(RPM) 확인(XF)

제어기의 현재 속도(RPM)를 확인하는 명령어 입니다.



- Protocol DATA

PC	0	1	2	3	4
↓	STX	X	F	ETX	LRC
제어기	0x02	0x58	0x46	0x03	

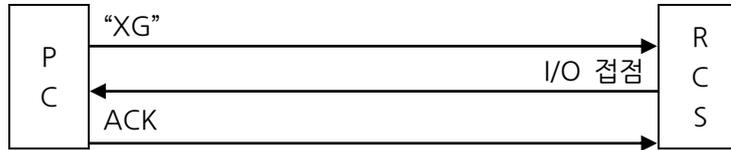
제어기	0	1	2~11	12	13
↓	STX	상태	속도값	ETX	LRC
PC	0x02	Flag	(RPM)	0x03	

PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

#### 4.4.47 전체 I/O 상태 확인(XG)

I/O접점 B000 부터 B417까지의 접점 상태를 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4	5
↓	STX	X	G	0	ETX	LRC
제어기	0x02	0x58	0x47	0x30	0x03	

제어기	0	1	2~85	86	87
↓	STX	상태	I/O 접점	ETX	LRC
PC	0x02	Flag		0x03	

PC	0
↓	ACK
제어기	0x06

● I/O 접점 : I/O접점 전체를 전송 합니다.

I/O 접점 (2Bytes x 84)					
	B00		...	B41	
순서	2	3	...	84	85
내용	설정한 I/O 번호		...	설정한 I/O 번호	
	하위 비트	상위 비트	...	하위 비트	상위 비트

예1)

패킷 번호	2								3							
Bit(B00)	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
⋮	⋮								⋮							
패킷 번호	84								85							
Bit(B41)	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B00 (HEX : 0x41)	7	6	5	4	3	2	1	0
	0	0	0	1	0	1	0	0
⋮	⋮							
B41 (HEX : 0x00)	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

#### 4.4.48 전체 위치, 정수형 변수 확인(XH)

위치형 변수 0 부터 1023번 및 정수형 변수 0 부터 255까지 값을 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4		
↓	STX	X	H	ETX	LRC		
제어기	0x02	0x58	0x48	0x03			
제어기	0	1	2~21	22	23	24	25
↓	STX	상태	위치형 변수	0x20	'\n'	ETX	LRC
PC	0x02	Flag	P0000			0x03	
PC	0						
↓	ACK						
제어기	0x06						
제어기	0	1	2~21	22	23	24	25
↓	STX	상태	위치형 변수	0x20	'\n'	ETX	LRC
PC	0x02	Flag	P0001			0x03	
PC	0						
↓	ACK						
제어기	0x06						
⋮							
제어기	0	1	2~n	n+1	n+2	n+3	n+4
↓	STX	상태	위치형 변수	0x20	'\n'	ETX	LRC
PC	0x02	Flag	P1023			0x03	
PC	0						
↓	ACK						
제어기	0x06						
제어기	0	1	2~n	n+1	n+2	n+3	n+4
↓	STX	상태	정수형 변수	0x20	'\n'	ETX	LRC
PC	0x02	Flag	I000			0x03	
PC	0						
↓	ACK						
제어기	0x06						

제어기	0	1	2~n	n+1	n+2	n+3	n+4
↓	STX	상태	정수형 변수				
PC	0x02	Flag	I001	0x20	'\wn'	ETX	LRC
						0x03	

PC	0
↓	ACK
제어기	0x06

⋮

제어기	0	1	2~n	n+1	n+2	n+3	n+4
↓	STX	상태	정수형 변수				
PC	0x02	Flag	I255	0x20	'\wn'	ETX	LRC
						0x03	

PC	0
↓	ACK
제어기	0x06

- 위치형 변수 : P0000 ~ P1023번까지 총 1024개의 위치형 변수 값 확인.

위치형 변수(21Bytes)

2	3~6	7	8~21	22
P	위치형 변수 번호	0x20	위치값	0x20
0x50				

- 위치형 변수 번호(4Bytes)

패킷 번호	3	4	5	6
내용	위치, 정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

- 위치 값 (14bytes)

패킷 번호	8~16	17	18~21
내용	정수형 자릿수	'.'	소수점 자릿수

- 정수형 변수 : I000~I255까지 총 256개의 정수형 변수의 값 확인.

정수형 변수(n+1Bytes)

2	3	4~7	8	n	n+1
I	0x20	정수형 변수 번호	0x20	정수값	0x20
0x49					

- 정수형 변수의 번호(4Bytes)

패킷 번호	4	5	6	7
내용	정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

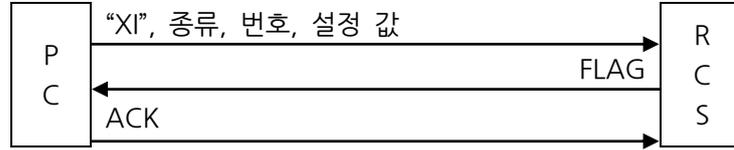
- 위치 값 (n bytes) : 값의 자릿수 n개 만큼 패킷 번호가 늘어납니다.

패킷 번호	n
내용	정수형 자릿수

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.49 위치, 정수형 변수 값 설정(XI)

위치형변수, 정수형변수 값 설정하는 명령어입니다.



● Protocol DATA

변수 번호가 3자리인 경우

PC	0	1	2	3	4	5	6	7~16	17	18
↓	STX	X	I	변수	변경 할 변수의 번호			변수 값	ETX	LRC
제어기	0x02	0x58	0x49	종류	설정				0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

변수 번호가 4자리인 경우.

PC	0	1	2	3	4	5	6	7	8~17	18	19
↓	STX	X	I	변수	변경 할 변수의 번호				변수 값	ETX	LRC
제어기	0x02	0x58	0x49	종류	설정					0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

● 변수 종류 : 확인 할 변수 종류를 설정합니다.

확인 할 변수 종류(1Byte)

값	설정
0	정수형 변수
1	위치형 변수

- 변경 할 변수의 번호 설정: 변경 할 변수 번호를 설정합니다.  
3자리, 4자리수 자리수 설정 없이 번호를 설정 할 수 있습니다.

변경 할 변수의 번호 설정(3Bytes)			
순서	4	5	6
내용	위치, 정수형 번호 설정		
	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ''(0x20) 또는 '0'(0x30)으로 작성합니다.

변경 할 변수의 번호 설정(4Bytes)				
순서	4	5	6	7
내용	위치, 정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ''(0x20) 또는 '0'(0x30)으로 작성합니다.

- 변수 값 : 설정 할 변수의 값입니다.

위치 값(10bytes)										
예 1) 12345.678	5	6	7	8	9	10	11	12	13	14
위치 값	0x20	0x20	1	2	3	4	5	6	7	8

위치값 : 12345.678

※ 비어있는 패킷은 ''(0x20) 또는 '0'(0x30)으로 작성합니다.

예 2) 456	5	6	7	8	9	10	11	12	13	14
정수 값	0x20	0x20	0x20	0x20	4	5	6	0x20	0x20	0x20

위치값 : 456

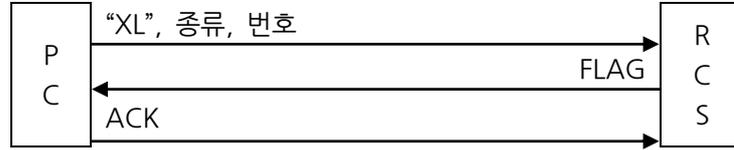
※ 비어있는 패킷은 ''(0x20) 또는 '0'(0x30)으로 작성합니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다.

#### 4.4.50 정수, 위치형 변수 값 확인(XL)

정수형 및 위치형 변수 값을 확인하는 명령어입니다.



● Protocol DATA

변수 번호가 3자리인 경우

PC	0	1	2	3	4	5	6	7	8
↓	STX	X	L	변수	확인 할 변수의			ETX	LRC
제어기	0x02	0x58	0x4C	종류	번호 설정			0x03	

제어기	0	1	2~11	12	13
↓	STX	상태	설정 된	ETX	LRC
PC	0x02	Flag	값	0x03	

PC	0
↓	ACK
제어기	0x06

변수 번호가 4자리인 경우

PC	0	1	2	3	4	5	6	7	8	9
↓	STX	X	L	변수	확인 할 변수의 번호 설정				ETX	LRC
제어기	0x02	0x58	0x4C	종류					0x03	

제어기	0	1	2~11	12	13
↓	STX	상태	설정 된	ETX	LRC
PC	0x02	Flag	값	0x03	

PC	0
↓	ACK
제어기	0x06

● 변수 종류 : 확인 할 변수 종류를 설정합니다.

확인 할 변수 종류(1Byte)

값	설정
0	정수형 변수
1	위치형 변수

- 확인 할 변수의 번호 설정: 확인 할 변수 번호를 설정합니다.  
3자리, 4자리수 자리수 설정 없이 번호를 설정 할 수 있습니다.

확인 할 변수의 번호 설정(3Bytes)			
순서	4	5	6
내용	위치, 정수형 번호 설정		
	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

확인 할 변수의 번호 설정(4Bytes)				
순서	4	5	6	7
내용	위치, 정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.



**CAUTION**

- 위치형 변수는 0~1023까지 설정을 하나 1020~1023까지 기능이 설정이 되어 사용을 하지 않기 바랍니다.(RCS8000 매뉴얼 "4.5 POSITION Teaching" 확인)

### 4.4.51 위치변수값으로 이동 명령어 수행(XM)

위치 변수의 번호를 지정하여 이동 명령어를 수행하는 명령어입니다.



● Protocol DATA

변수 번호가 3자리인 경우

PC	0	1	2	3	4	5	6	7	8
↓	STX	X	M	변수	이동 할 변수의			ETX	LRC
제어기	0x02	0x58	0x4D	종류	번호 설정			0x03	

제어기	0	1	2~11	12	13
↓	STX	상태	설정 된 값	ETX	LRC
PC	0x02	Flag		0x03	

PC	0
↓	ACK
제어기	0x06

변수 번호가 4자리인 경우

PC	0	1	2	3	4	5	6	7	8	9
↓	STX	X	M	변수	이동 할 변수의 번호 설정				ETX	LRC
제어기	0x02	0x58	0x4D	종류					0x03	

제어기	0	1	2~11	12	13
↓	STX	상태	설정 된 값	ETX	LRC
PC	0x02	Flag		0x03	

PC	0
↓	ACK
제어기	0x06

● 변수 종류 : 확인 할 변수 종류를 설정합니다.

확인 할 변수 종류(1Byte)

값	설정
0	정수형 변수
1	위치형 변수

- 이동 할 변수의 번호 설정: 이동 할 변수 번호를 설정합니다.  
3자리, 4자리수 자리수 설정 없이 번호를 설정 할 수 있습니다.

이동 할 변수의 번호 설정(3Bytes)			
순서	4	5	6
내용	위치, 정수형 번호 설정		
	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

이동 할 변수의 번호 설정(4Bytes)				
순서	4	5	6	7
내용	위치, 정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

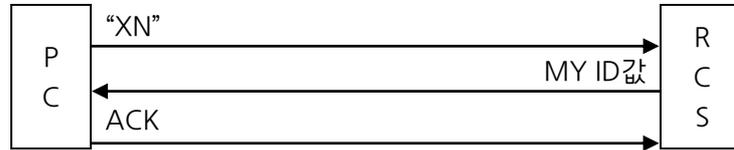
※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용 하여 읽기를 권장합니다.

4.4.52 파라미터 MY ID값 확인(XN)

파라미터 중 MY\_ID값을 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	N	ETX	LRC
제어기	0x02	0x58	0x4E	0x03	LRC

제어기	0	1	2	3	4	5	6	7	8	9	10
↓	STX	상태	M	Y	I	D	설정값			ETX	LRC
PC	0x02	Flag								0x03	

PC	0
↓	ACK
제어기	0x06

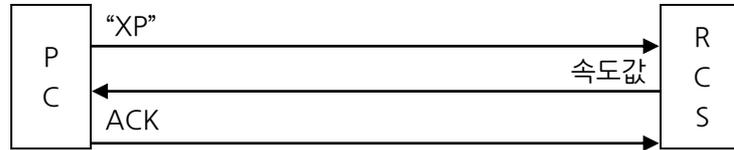
● 설정 값: MY\_ID 번호입니다.

MY_ID 번호 확인 (3Bytes)			
순서	6	7	8
내용	위치, 정수형 번호 설정		
	백의 자리	십의 자리	일의 자리

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

### 4.4.53 현재 설정 된 속도 확인(XP)

현재 RPM값을 최대값에 대해 백분율x100의 값으로 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	P	ETX	LRC
제어기	0x02	0x58	0x50	0x03	

제어기	0	1	2~11	12	13
↓	STX	상태	속도 값 확인	ETX	LRC
PC	0x02	Flag	(RPM)	0x03	

PC	0
↓	ACK
제어기	0x06

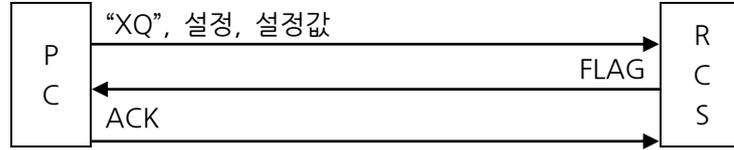
● 속도 값 확인(RPM)

현재 위치 값 (10bytes)										
예 1) 52%	2	3	4	5	6	7	8	9	10	11
위치 값	0x20	0x20	0x20	0x20	0x20	0x20	5	2	0	0

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

### 4.4.54 현재 가·감속값 설정(XQ)

현재 가·감속값을 설정하는 명령어입니다.



● Protocol DATA

PC ↓ 제어기	0	1	2	3	4	5	6	7	8	9	10	11	12
	STX 0x02	X 0x58	Q 0x51	가·감속 설정 1	설정값 Data 2		가·감속 설정 2	설정값 Data 3		ETX 0x03	LRC		

제어기 ↓ PC	0	1	2	3
	STX 0x02	상태 Flag	ETX 0x03	LRC

PC ↓ 제어기	0
	ACK 0x06

- 가·감속 설정 : 가속 또는 감속 설정 입니다.

가·감속 설정	
값	내용
A 0x41	가속 ACC
D 0x44	감속 DEC

- 설정 값: 설정 할 가·감속값입니다.

설정 값 2, 3			
순서	4 또는 8	5 또는 9	6 또는 10
내용	백의 자리	십의자리	일의자리

※ 비어있는 패킷은 ' '(0x20) 또는 '0'(0x30)으로 작성합니다.

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

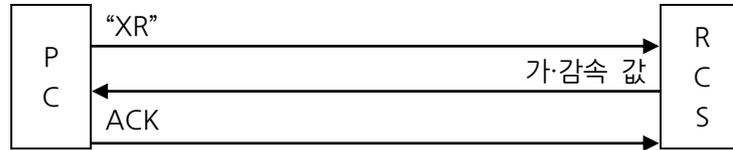


CAUTION

- 가감속 설정1, 2에 같은 설정을 한 경우 2의 설정값이 설정됩니다.

#### 4.4.55 현재 가·감속값 확인(XR)

현재 가·감속값을 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	R	ETX	LRC
제어기	0x02	0x58	0x52	0x03	LRC

제어기	0	1	2	3	4	5	6	7	8	9	10	11
↓	STX	상태	A	가속 설정값		D	감속 설정값			ETX	LRC	
PC	0x02	Flag	가속			감속				0x03	LRC	

PC	0
↓	ACK
제어기	0x06

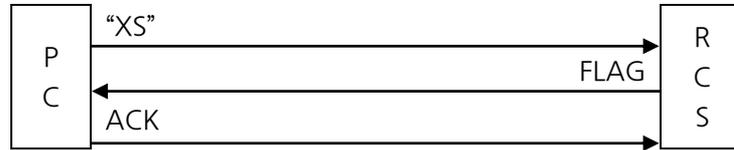
- 가·감속 설정 값: 현재 설정 된 가·감속값입니다.

가·감속 설정 값 (3Bytes)			
순서	3 또는 7	4 또는 8	5 또는 9
내용	백의 자리	십의자리	일의자리

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

### 4.4.56 현재 위치 값 확인(XS)

현재 위치값을 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	S	ETX	LRC
제어기	0x02	0x58	0x53	0x03	LRC

제어기	0	1	2~11	12	13
↓	STX	상태	현재 위치 값	ETX	LRC
PC	0x02	Flag		0x03	LRC

PC	0
↓	ACK
제어기	0x06

● 현재 위치 값

현재 위치 값의 뒤에서 3자리는 소수점으로 설정 됩니다.

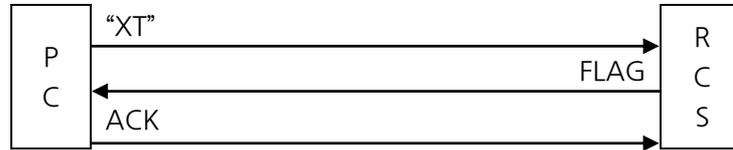
현재 위치 값 (10bytes)

예 1) 12345.678	2	3	4	5	6	7	8	9	10	11
위치 값	0x20	0x20	1	2	3	4	5	6	7	8

● STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

### 4.4.57 속도 값은 0으로 설정(XT)

현재 속도값을 0으로 설정하는 명령어입니다.



- Protocol DATA

PC	0	1	2	3	4
↓	STX	X	T	ETX	LRC
제어기	0x02	0x58	0x54	0x03	

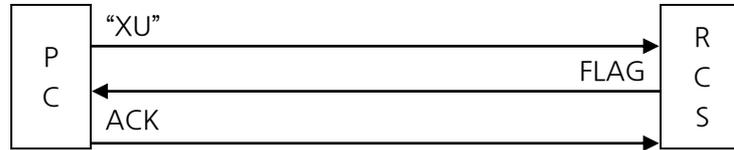
제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.58 현재 위치값을 0.000으로 설정(XU)

설정 위치값을 0.000으로 설정하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	U	ETX	LRC
제어기	0x02	0x58	0x55	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

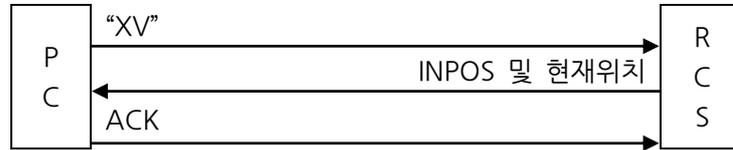
PC	0
↓	ACK
제어기	0x06

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

※ Protocol 명령 후 통신에러 발생시 이에 대한 원인을 “통신에러 원인읽기(KD)”프로토콜을 이용하여 읽기를 권장합니다

#### 4.4.59 모터의 현재 상태 확인(XV)

목적 위치까지 도달여부 및 현재 위치 확인하는 명령어입니다.



● Protocol DATA

PC	0	1	2	3	4
↓	STX	X	V	ETX	LRC
제어기	0x02	0x58	0x56	0x03	

제어기	0	1	2~11	12	13	14	15
↓	STX	상태	현재 위치값	R	모터 상태	ETX	LRC
PC	0x02	Flag		0x52		0x03	

PC	0
↓	ACK
제어기	0x06

- 현재 위치값  
변경 값의 뒤에서 3자리는 소수점으로 설정 됩니다.  
사용하지 않은 자리는 0x20으로 설정됩니다.

현재 위치 값(10bytes)										
예 1) 12345.678	2	3	4	5	6	7	8	9	10	11
위치 값	0x20	0x20	1	2	3	4	5	6	7	8

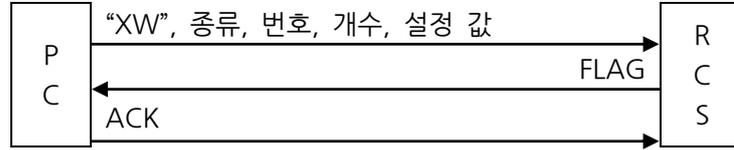
- 모터 상태 : 모터의 INPOS 상태 확인..

모터 상태	
값	설명
0	INPOS 도달
1	INPOS 미 도달

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

4.4.60 지정한 번호부터 정수형 및 위치형 변수의 개수를 선택하여 값 변경(XW)

원하는 정수형, 위치형 변수의 개수를 선택하여 값을 변경하는 명령어입니다.



- Protocol DATA

변수 번호가 4자리인 경우

PC	0	1	2	3	4	5	6	7	8~17	18~27	27~n*	n+1	N+2
↓	STX	X	W	변수	확인 할 변수의			변경 할	변경 할	변경 할	변경 할	ETX	LRC
제어기	0x02	0x58	0x57	종류	번호 설정			변수의 개수	값	값	값	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

변수 번호가 3자리인 경우

PC	0	1	2	3	4	5	6	7	8	9~18	19~28	28~n*	n+1	N+2
↓	STX	X	W	변수	확인 할 변수의			변경 할	변경 할	변경 할	변경 할	변경 할	ETX	LRC
제어기	0x02	0x58	0x57	종류	번호 설정			변수의 개수	값	값	값	값	0x03	

제어기	0	1	2	3
↓	STX	상태	ETX	LRC
PC	0x02	Flag	0x03	

PC	0
↓	ACK
제어기	0x06

- 변수 종류 : 확인 할 변수 종류를 설정합니다.

변수 종류(1Byte)	
값	설정
0	정수형 변수
1	위치형 변수

- 확인 할 변수의 번호 설정: 확인 할 변수 번호를 설정합니다.  
3자리, 4자리수 자리수 설정 없이 번호를 설정 할 수 있습니다.

확인 할 변수의 번호 설정(3Bytes)			
순서	4	5	6
내용	위치, 정수형 번호 설정		
	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 '(0x20) 또는 '(0x30)으로 작성합니다.

확인 할 변수의 번호 설정(4Bytes)				
순서	4	5	6	7
내용	위치, 정수형 번호 설정			
	천의 자리	백의 자리	십의 자리	일의 자리

※ 비어있는 패킷은 '(0x20) 또는 '(0x30)으로 작성합니다.

- 확인 할 변수 개수 : 확인 할 변수의 개수를 설정합니다.

확인 할 변수 개수(1Byte)	
순서	7 또는 8
내용	일의 자리

- 변경 값 : 변경 할 변수의 값입니다.  
변경 값의 뒤에서 3자리는 소수점으로 설정 됩니다.  
사용하지 않은 자리는 0x00또는 0x20으로 설정합니다.

위치 값(10bytes)										
예 1) 12345.678	5	6	7	8	9	10	11	12	13	14
위치 값	0x20	0x20	1	2	3	4	5	6	7	8

예 2) 456	5	6	7	8	9	10	11	12	13	14
정수 값	0x20	0x20	0x20	0x20	4	5	6	0x20	0x20	0x20

- STX, ETX, LRC, FLAG: “4.1통신 규칙”을 참조 하십시오.

Rev.	수정일자	내용	수정자	S/W Version
V.1		초판 인쇄		

RCS ROBOT CONTROLLER

---

# CONTROLLER MANUAL

FIRST EDITION JANUARY 2016

ROBOSTAR CO, LTD

ROBOT R&D CENTER

---