



세계를 이끄는 힘
Global Leader

인버터 기술자료

| | |
|------|-----------------------|
| 문서번호 | TSD-N700-COM-001K(00) |
| 적용모델 | N700 모델 전체 |
| 개정일자 | 2009년 6월 |

N700 인버터 RS232/485 통신 인터페이스 지침서

목 차

| | |
|--|----|
| 1장. 시리얼 통신 기능..... | 1 |
| 1.1 RS232 통신 기능 | 1 |
| 1.1.1 인버터의 RS232 통신 설정..... | 1 |
| 1.1.2 PC~인버터 Cable 연결..... | 3 |
| 1.2 RS485 통신 기능 | 4 |
| 1.2.1 인버터의 RS-485 통신 설정..... | 4 |
| 1.2.2 PC~RS232/485converter~인버터 Cable 연결..... | 5 |
| 2장. Modbus 통신 프로토콜..... | 6 |
| 2.1 인버터 기능코드 | 6 |
| 2.2 통신 순서..... | 7 |
| 2.3 파라미터 설정값 읽어오기 | 8 |
| 2.3.1 송신프레임 (Frame 1)..... | 8 |
| 2.3.2 응답프레임 (Frame 2)..... | 9 |
| 2.3.3 전송 프레임 예제 | 10 |
| 2.4 파라미터 설정값 변경하기 | 12 |
| 2.4.1 송신프레임 (Frame 1)..... | 12 |
| 2.4.2 응답프레임 (Frame 2)..... | 13 |
| 2.4.3 전송 프레임 예제 | 14 |
| 2.5 주파수 지령 | 15 |
| 2.5.1 주파수 지령 방법 설정 | 15 |
| 2.5.2 전송 프레임 예제 | 16 |
| 2.6 운전 지령..... | 17 |
| 2.6.1 운전 지령 방법 설정 | 17 |
| 2.6.2 전송 프레임 예제 | 18 |
| 2.7 CRC-16 생성 절차..... | 20 |
| 2.7.1 CRC-16을 생성 절차..... | 20 |
| 2.7.2 CRC-16 생성의 예 : D001의 출력주파수를 읽고자 할 경우 | 21 |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

| | |
|--|----|
| 3장. 통신 관련 Trouble Shooting..... | 28 |
| 3.1 PC와 인버터간 통신이 되지 않습니다. (Case 1)..... | 28 |
| 3.1.1 Step 1 : PC ~ 인버터간 Cable 연결을 점검합니다..... | 28 |
| 3.1.2 Step 2 : 인버터 제어단자대의 통신 선택 스위치 상태를 점검합니다. | 29 |
| 3.1.3 Step 3 : RS485 통신인 경우, N번째 인버터의 종단저항 설정을 점검합니다. | 30 |
| 3.1.4 Step 4 : 인버터 오퍼레이터에서 통신 관련 설정을 점검합니다..... | 30 |
| 3.1.5 Step 5 : PC의 COM Port가 정상 동작하는지 점검합니다..... | 31 |
| 3.1.6 Step 6 : 인버터 통신 Port 상태를 점검합니다..... | 34 |
| 3.1.7 Step 7 : PC ~ 인버터간 통신 상태를 점검합니다..... | 34 |
| 3.2 인버터 운전중예, 통신으로 운전방향 전환이 안됩니다. (Case 2)..... | 35 |
| 3.2.1 Step 1 : 인버터 운전방향제한(B001) 코드값을 점검합니다..... | 35 |
| 3.2.2 Step 2 : 통신으로 인버터 운전(정/역방향)을 실시합니다..... | 35 |
| 3.2.3 Step 3. 인버터에서 오퍼레이터로 정/역방향 운전이 되는지 점검합니다. | 35 |
| 3.2.4 Step 4. 문제점이 해결되지 않으면, 당사 고객센터센터로 기술지원을 요청하시기 바랍니다..... | 35 |

1장. 시리얼 통신 기능

1.1 RS232 통신 기능

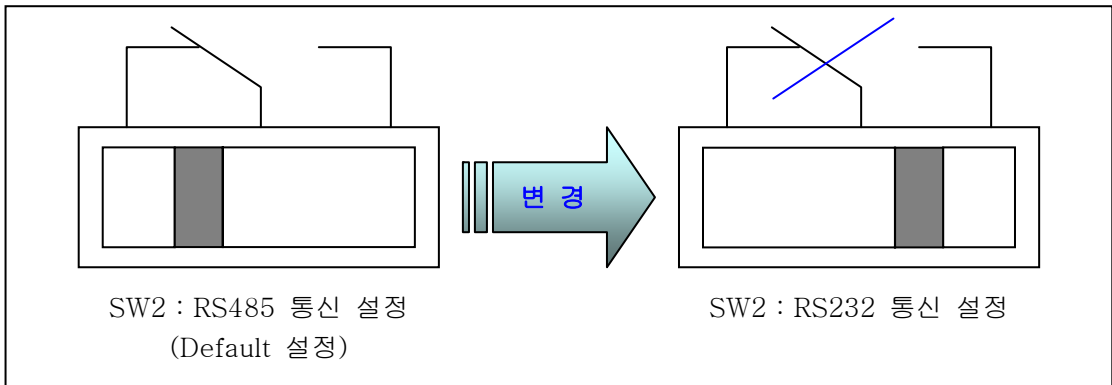
인버터의 제어기에 부착된 콘넥터를 이용하여 RS-232로 인버터와 외부 제어기기와의 통신을 할 수 있습니다.

외부 제어기기는 단지 1대의 인버터와 RS-232 통신을 할 수 있습니다.

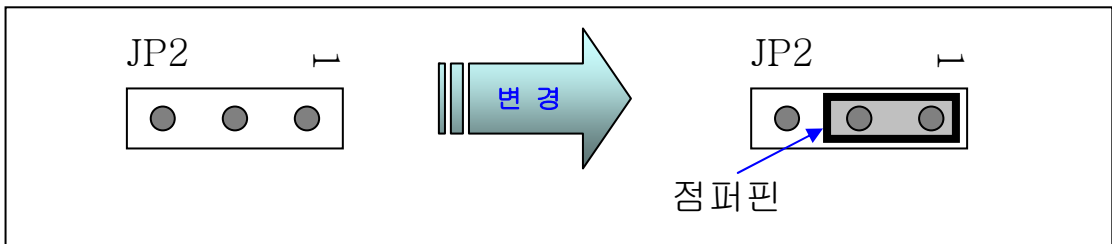


1.1.1 인버터의 RS232 통신 설정

(1) 인버터 제어단자대 PCB의 SW2 스위치를 RS232 통신으로 변경한다.



(2) 인버터 제어단자대 PCB의 J2를 점퍼핀을 사용 단락시켜 종단저항을 유효로 설정한다.
(RS232 통신설정시에는, 점퍼핀 설정을 하지 않아도 됨.)

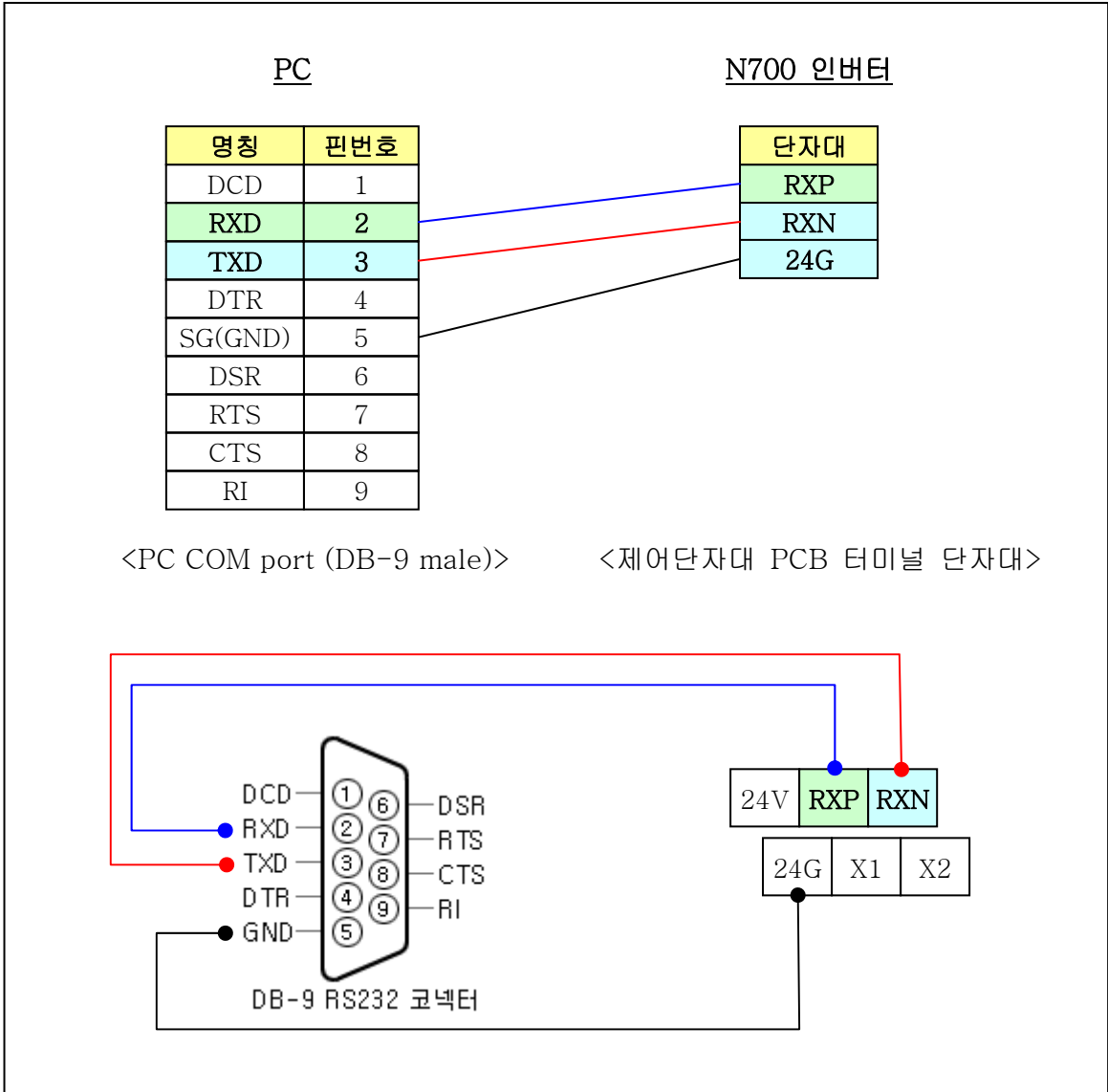


(3) 인버터 기능 코드 설정

- ① **b037** 데이터지령 선택
초기값 0(operator) → 4(RS232)으로 변경후 저장
- ② **b038** 통신 속도
초기값 2(9600bps) 확인
- ③ **b039** 통신 국번
초기값 1(국번1) 확인
- ④ **b040** 통신 비트 길이
초기값 8(8bit) 확인
- ⑤ **b041** 통신 패리티
초기값 0(No parity) 확인
- ⑥ **b042** 통신 정지 비트
초기값 1(1bit) 확인

N700 인버터 RS232/RS485 통신 인터페이스 지침서

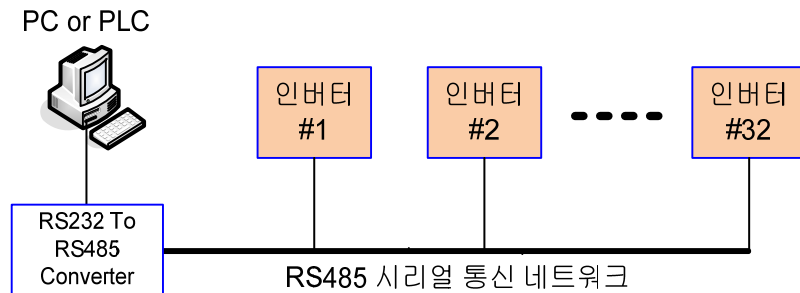
1.1.2 PC~인버터 Cable 연결



1.2 RS485 통신 기능

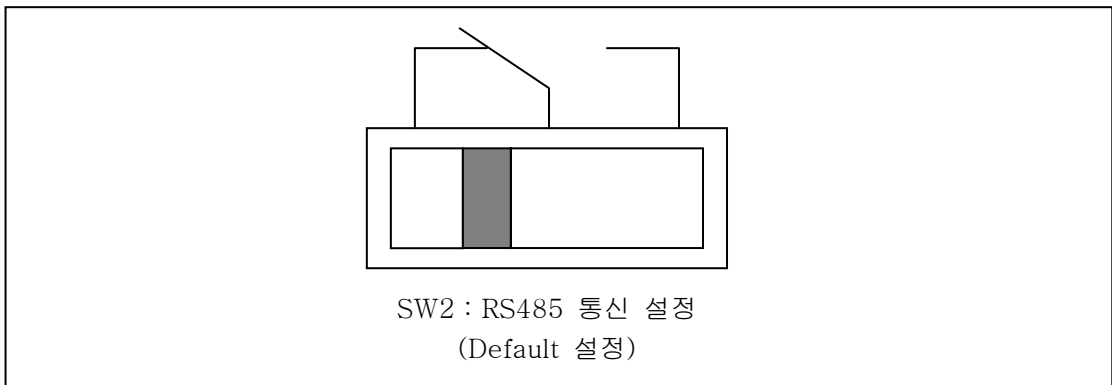
인버터의 제어기에 부착된 콘넥터를 이용하여 RS-485로 인버터와 외부 제어기기와의 통신을 할 수 있습니다.

외부 제어기기는 최대 32대의 인버터와 RS-485 시리얼 통신을 할 수 있습니다.

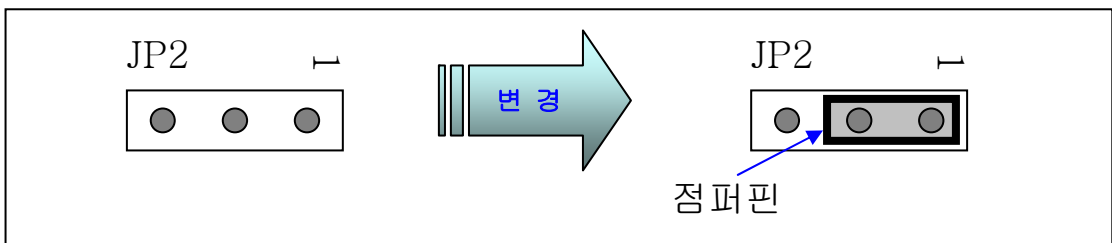


1.2.1 인버터의 RS-485 통신 설정

(1) 인버터 제어단자대 PCB의 SW2 스위치가 RS-485 통신으로 되어있는지 확인한다.



(2) 인버터 제어단자대 PCB의 J2를 점퍼핀으로 단락시켜 종단저항을 유효로 설정한다.
(1:N 방식의 Multi-drop 통신인 경우는 종단 인버터에만 설정함.)

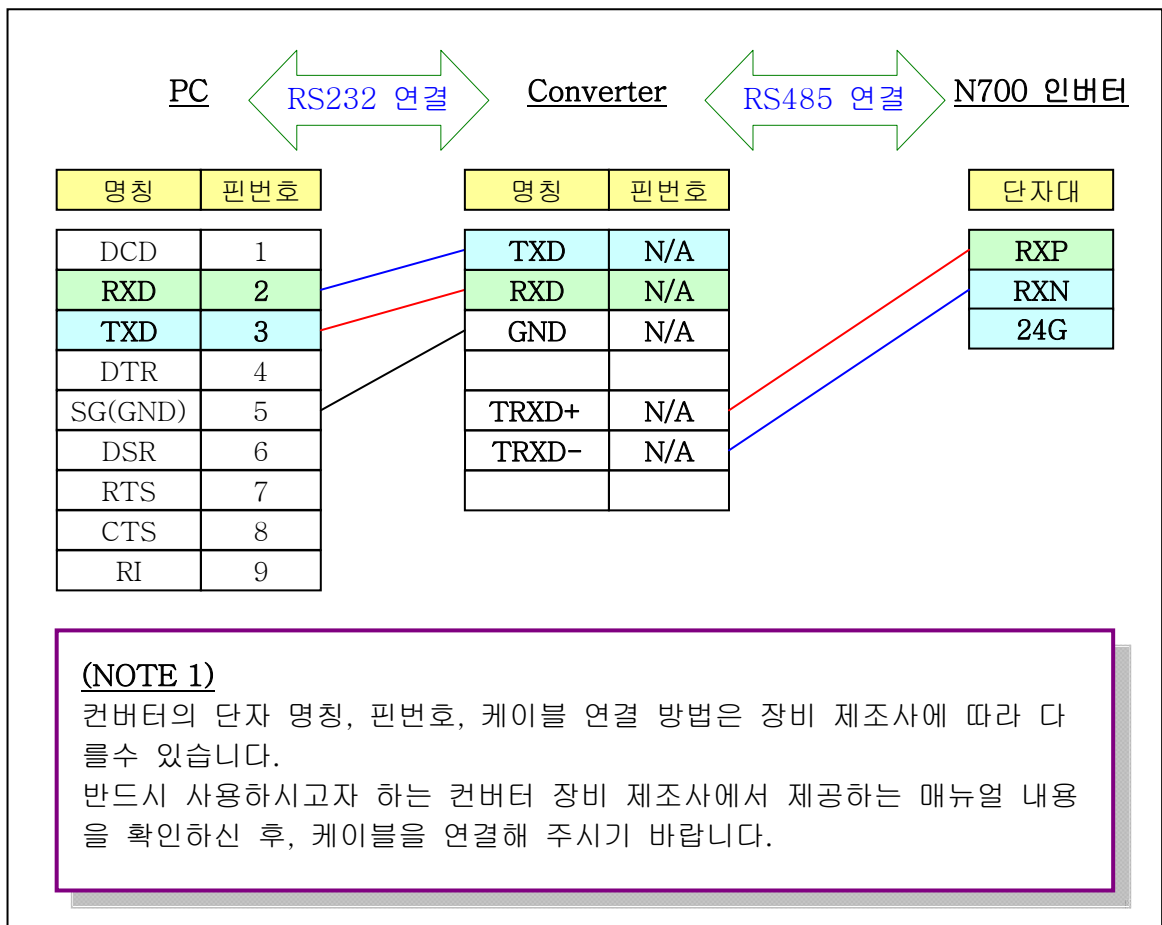


N700 인버터 RS232/RS485 통신 인터페이스 지침서

(3) 인버터 기능 코드 설정

- ① b037 데이터지령 선택
초기값 0(operator) → 1(RS-485)로 변경후 저장
- ② b038 통신 속도
초기값 2(9600bps) 확인
- ③ b039 통신 국번
초기값 1(국번1) 확인
- ④ b040 통신 비트 길이
초기값 8(8bit) 확인
- ⑤ b041 통신 패리티
초기값 0(No parity) 확인
- ⑥ b042 통신 정지 비트
초기값 1(1bit) 확인

1.2.2 PC~RS232/485converter~인버터 Cable 연결



2장. Modbus 통신 프로토콜

2.1 인버터 기능코드

N700 인버터는 Modbus RTU 프로토콜을 지원합니다.

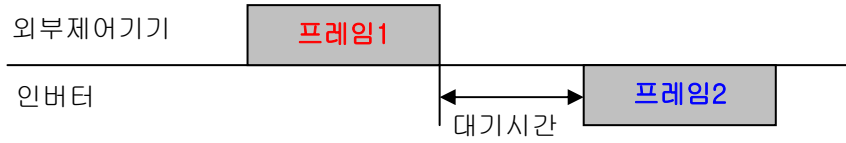
통신을 하기 위해 인버터 기능 코드를 아래와 같이 설정해 주십시오.

| 기능코드 | 기능명칭 | 초기값 | 데이터 | 내용 |
|------|-------------|-----|----------------|---|
| b037 | 데이터지령 선택 | 0 | 0(OPERATOR) | OPERATOR |
| | | | 1(RS485) | RS485 |
| | | | 2(OPT1) | 옵션1 (Profibus) |
| | | | 3(OPT2) | 옵션2 (DeviceNet) |
| | | | 4(RS232) | RS232 |
| b038 | 통신속도 | 2 | 0(2400BPS) | 2,400 BPS |
| | | | 1(4800BPS) | 4,800 BPS |
| | | | 2(9600BPS) | 9,600 BPS |
| | | | 3(19200BPS) | 19,200 BPS |
| | | | 4(38400BPS) | 38,400 BPS |
| b039 | 통신국번 | 1 | 1~32 | 인버터 국번을 설정합니다. 동시에 복수대를 제어할 시 설정합니다. |
| b040 | 통신비트 길이 | 8 | 7(7BIT) | 7비트 |
| | | | 8(8BIT) | 8비트 |
| b041 | 통신 패리티 | 0 | 0(NO PARITY) | 패리티 없음 |
| | | | 1(EVEN PARITY) | 우수 패리티 |
| | | | 2(ODD PARITY) | 기수 패리티 |
| b042 | 통신정지비트 | 1 | 1(1BIT) | 1비트 |
| | | | 2(2BIT) | 2비트 |

※ 상기 코드중 b037 설정값을 통신으로 변경한 후에는 오퍼레이터에서 다른 코드값 변경이 불가하오니, 반드시 b037 코드값은 마지막에 변경해 주시기 바랍니다.

2.2 통신 순서

외부제어기와 인버터간의 통신은 아래와 같은 순서로 행해집니다.



프레임 개시 : 신호선에 Data가 전송되면 프레임이 시작된 것으로 인식

프레임 종료 : 4.5 character에 해당하는 시간 이상 data가 없으면 프레임 종료로 인식

프레임1 : 외부제어기기에서 인버터로 송신하는 프레임

프레임2 : 인버터에서 외부제어기기로 반송하는 프레임을 표시

인버터에서의 프레임2는 인버터가 프레임1을 받고 나서 적절한 프레임을 인식하고 응답하는 신호로써 출력하는 것이며, 능동적인 출력은 하지 않습니다.

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.3 파라미터 설정값 읽어오기

2.3.1 송신프레임 (Frame 1)

| | | | | | |
|-------|-------|-------|-------|--------|--------|
| 1Byte | 1Byte | 2Byte | 2Byte | 1Byte | 1Byte |
| 국번 | 지령 | 파라미터 | 파라미터수 | CRC Hi | CRC Lo |

| 구분 | 설명 | 데이터크기 | 내용 |
|--------|-----------|--------|--|
| 국번 | 대상인버터 국번 | 1 byte | 1~32 |
| 지령 | 프레임 종류 | 1 byte | 0x03 → Read 옵션 |
| 파라미터 | 파라미터 | 2 byte | 1 st byte : Group D(0x01) F(0x02) U(0x03) A(0x04) B(0x05) I(0x06) O(0x07) C(0x08) H(0x09) P(0x0a) 2 nd byte : index (파라미터 번호) |
| 파라미터수 | 요구 파라미터 수 | 2 byte | 1 st byte : 0x00 2 nd byte : N(0x01~0x08) 위에서 정의된 파라미터로부터 N개의 파라미터를 요구한다. |
| CRC Hi | - | 1 byte | 16bit CRC의 상위 8bit |
| CRC Lo | - | 1 byte | 16bit CRC의 하위 8bit |

* (예) 010301010001D436

- ① 01 : 통신 국번이 1인 인버터를 호출함.
- ② 03 : Read 옵션 지정
- ③ 0101 : D 그룹의 D001(출력주파수) 지정
- ④ 0001 : 1개의 파라미터를 지정
- ⑤ D436 : D4 → CRC 상위 8 Bit 36 → CRC 하위 8 Bit

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.3.2 응답프레임 (Frame 2)

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|--------|--------|
| 1Byte | 1Byte | 1Byte | 2Byte | | 2Byte | 1Byte | 1Byte |
| 국번 | 지령 | Byte수 | 데이터1 | | 데이터 N | CRC Hi | CRC Lo |

| 구분 | 설명 | 데이터크기 | 내용 |
|--------|-----------|--------|------------------------|
| 국번 | 대상인버터 국번 | 1 byte | 1~32 |
| 지령 | 프레임 종류 | 1 byte | 0x03 → Read 옵션 |
| Byte수 | 데이터 byte수 | 1 byte | 요구 파라미터 수 x 2 |
| 데이터 1 | 파라미터 1 | 2 byte | 1 st 파라미터 값 |
| 데이터 N | 파라미터 N | 2 byte | N번째 파라미터 값 |
| CRC Hi | - | 1 byte | 16bit CRC의 상위 8bit |
| CRC Lo | - | 1 byte | 16bit CRC의 하위 8bit |

* Frame Size = 5Byte + (2 * 요구파라미터 갯수)Byte

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.3.3 전송 프레임 예제

(1) 통신 Data 분석 기준

- ① 국번 : 01 (Address 1번 인버터 지정)
- ② 지령 : 03 (Read Request)
- ③ 파라미터 수 : 1개 기준
- ④ 모니터/트립&경고 파라미터 Read Request (파라미터 1개 기준)

(2) 출력주파수값 읽어오기 (파라미터 : D001)

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 파라미터 수 | | CRC 16 | |
|------------------|----|----|------|----|--------|----|--------|----|
| | | | | | | | Hi | Lo |
| 010301010001D436 | 01 | 03 | 01 | 01 | 00 | 01 | D4 | 36 |

② 응답프레임 (In case of D001 = 0.00Hz)

| 응답 (7Byte) | 국번 | 지령 | 바이트수 | 데이터 | | CRC 16 | |
|----------------|---|----|------|-----|----|--------|----|
| | | | | | | Hi | Lo |
| 0103020000B844 | 01 | 03 | 02 | 00 | 00 | B8 | 44 |
| 데이터 설명 | $0 \times 16^3 + 0 \times 16^2 + 0 \times 16^1 + 0 \times 16^0 = 0$ (Communication Data) → divide by 100 → 0.00Hz (Parameter value) | | | | | | |

③ 응답프레임 (In case of D001 = 60.00Hz)

| 응답 (7Byte) | 국번 | 지령 | 바이트수 | 데이터 | | CRC 16 | |
|----------------|---|----|------|-----|----|--------|----|
| | | | | | | Hi | Lo |
| 0103021770B650 | 01 | 03 | 02 | 17 | 70 | B6 | 50 |
| 데이터 설명 | $1 \times 16^3 + 7 \times 16^2 + 7 \times 16^1 + 0 \times 16^0 = 6000$ (Communication Data) → divide by 100 → 60.00Hz (Parameter value) | | | | | | |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

(3) Output frequency (Function code : F001)

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 파라미터 수 | | CRC 16 | |
|------------------|----|----|------|----|--------|----|--------|----|
| | | | | | | | Hi | Lo |
| 010302010001D472 | 01 | 03 | 02 | 01 | 00 | 01 | D4 | 72 |

② 응답프레임 (In case of F001 = 60.0Hz)

| 응답 (7Byte) | 국번 | 지령 | 바이트수 | 데이터 | | CRC 16 | |
|----------------|---|----|------|-----|----|--------|----|
| | | | | | | Hi | Lo |
| 0103021770B650 | 01 | 03 | 02 | 17 | 70 | B6 | 50 |
| 데이터 설명 | $1 \times 16^3 + 7 \times 16^2 + 7 \times 16^1 + 0 \times 16^0 = 6000$ (Communication Data) → divide by 100 → 60.00Hz (Parameter value) | | | | | | |

③ 응답프레임 (In case of F001 = 50.00Hz)

| 응답 (7Byte) | 국번 | 지령 | 바이트수 | 데이터 | | CRC 16 | |
|----------------|---|----|------|-----|----|--------|----|
| | | | | | | Hi | Lo |
| 0103021388B512 | 01 | 03 | 02 | 13 | 88 | B5 | 12 |
| 데이터 설명 | $1 \times 16^3 + 3 \times 16^2 + 8 \times 16^1 + 8 \times 16^0 = 5000$ (Communication Data) → divide by 100 → 50.00Hz (Parameter value) | | | | | | |

(4) Acceleration time (Function code : F007)

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 파라미터 수 | | CRC 16 | |
|------------------|----|----|------|----|--------|----|--------|----|
| | | | | | | | Hi | Lo |
| 010302010001D472 | 01 | 03 | 02 | 0C | 00 | 01 | 45 | B1 |

② 응답프레임 (In case of F007 = 10.0sec)

| 응답 (7Byte) | 국번 | 지령 | 바이트수 | 데이터 | | CRC 16 | |
|----------------|---|----|------|-----|----|--------|----|
| | | | | | | Hi | Lo |
| 0103020064B9AF | 01 | 03 | 02 | 00 | 64 | B9 | AF |
| 데이터 설명 | $0 \times 16^3 + 0 \times 16^2 + 6 \times 16^1 + 4 \times 16^0 = 100$ (Communication Data) → divide by 10 → 10.0sec (Parameter value) | | | | | | |

2.4 파라미터 설정값 변경하기

통신을 이용하여 인버터 기능코드의 설정값을 변경할 수 있습니다.

단, 출력주파수 (F001 & F002) 및 운전 지령을 하기 위해서는 별도의 파라미터를 사용하며, 자세한 내용은 [“2.5 주파수지령”](#) 및 [“2.6 운전지령”](#)을 참조하시기 바랍니다.

2.4.1 송신프레임 (Frame 1)

| | | | | | |
|-------|-------|-------|-------|--------|--------|
| 1Byte | 1Byte | 2Byte | 2Byte | 1Byte | 1Byte |
| 국번 | 지령 | 파라미터 | 데이터 | CRC Hi | CRC Lo |

| 구분 | 설명 | 데이터크기 | 내용 |
|--------|----------|--------|--|
| 국번 | 대상인버터 국번 | 1 byte | 1~32, FF(전국통신, Broadcasting) |
| 지령 | 프레임 종류 | 1 byte | 0x06 → Write 옵션 |
| 파라미터 | 파라미터 | 2 byte | 1 st byte : Group D(0x01) F(0x02) U(0x03) A(0x04) B(0x05) I(0x06) O(0x07) C(0x08) H(0x09) P(0x0a) 2 nd byte : index (파라미터 번호) |
| 데이터 | 데이터 | 2 byte | 설정하고자 하는 값 |
| CRC Hi | - | 1 byte | 16bit CRC의 상위 8bit |
| CRC Lo | - | 1 byte | 16bit CRC의 하위 8bit |

※ 전국통신은(국번:FF) Write 지령에서만 사용되어질 수 있으며, 인버터의 응답은 없습니다.

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.4.2 응답프레임 (Frame 2)

| | | | | | |
|-------|-------|-------|-------|--------|--------|
| 1Byte | 1Byte | 2Byte | 2Byte | 1Byte | 1Byte |
| 국번 | 지령 | 파라미터 | 데이터 | CRC Hi | CRC Lo |

| 구분 | 설명 | 데이터크기 | 내용 |
|--------|----------|--------|--|
| 국번 | 대상인버터 국번 | 1 byte | 1~32 |
| 지령 | 프레임 종류 | 1 byte | 0x06 |
| 파라미터 | 파라미터 | 2 byte | 1 st byte : Group D(0x01) F(0x02) U(0x03) A(0x04) B(0x05) I(0x06) O(0x07) C(0x08) H(0x09) P(0x0a) 2 nd byte : index (파라미터 번호 16진수 값) |
| 데이터 | 데이터 | 2 byte | 설정하고자 하는 값이 응답됨 |
| CRC Hi | - | 1 byte | 16bit CRC의 상위 8bit |
| CRC Lo | - | 1 byte | 16bit CRC의 하위 8bit |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.4.3 전송 프레임 예제

(1) 통신 Data 분석 기준

- ① 국번 : 01 (Address 1번 인버터 지정)
- ② 지령 : 06 (설정 요구)

(2) 출력주파수 설정 setting (Function code : F001, F201)

[“2.5 주파수지령”](#) 내용을 참조 하시기 바랍니다.

(3) Acceleration time setting (F007)

① 송신프레임 (In case of 10sec)

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 0106020200642859 | 01 | 06 | 02 | 0C | 00 | 64 | 49 | 9A |
| 데이터 설명 | $0x16^3 + 0x16^2 + 6x16^1 + 4x16^0 = 100$ (Communication Data) → divide by 10 → 10.0sec (Parameter value) | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 0106020200642859 | 01 | 06 | 02 | 0C | 00 | 64 | 49 | 9A |

(4) Deceleration time setting (F008)

① 송신프레임 (In case of 30sec)

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|--|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 01060203012C783F | 01 | 06 | 02 | 0E | 01 | 2C | E9 | FC |
| 데이터 설명 | $0x16^3 + 1x16^2 + 2x16^1 + 12x16^0 = 300$ (Communication Data) → divide by 10 → 30.0sec (Parameter value) | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 01060203012C783F | 01 | 06 | 02 | 0E | 01 | 2C | E9 | FC |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.5 주파수 지령

2.5.1 주파수 지령 방법 설정

통신으로 주파수 지령을 하기 위해서는 인버터 기능 코드를 아래와 같이 설정해 주십시오.

(1) 기능코드

| 기능코드 | 기능명칭 | 초기값 | 데이터 | 내용 |
|------|-----------|-----|-----|-----------------------------|
| F010 | 주파수 지령 방법 | 0 | 0 | OPE VOL (통신 Data : 0000) |
| | | | 1 | Terminal (통신 Data : 0001) |
| | | | 2 | OPE Keypad (통신 Data : 0002) |
| | | | 3 | COM (통신 Data : 0003) |
| | | | 4 | OPT1 (통신 Data : 0004) |
| | | | 5 | OPT2 (통신 Data : 0005) |

(2) 통신으로 상기 기능코드 변경을 하는 경우에 Data Frame은 아래와 같습니다.

① 송신프레임 (인버터 국번이 1인 경우)

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---------------------------------|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 0106021100039876 | 01 | 06 | 02 | 11 | 00 | 03 | 98 | 76 |
| 데이터 설명 | 0003 → 주파수 지령방법을 COM(통신)으로 설정함. | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 0106021100039876 | 01 | 06 | 02 | 11 | 00 | 03 | 98 | 76 |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.5.2 전송 프레임 예제

(1) 통신 Data 분석 기준

- ① 국번 : 01 (Address 1번 인버터 지정)
- ② 지령 : 06 (Setting request)
- ③ Parameter : 0004 (주파수 지령)

(2) 주파수지령 (In case of 60Hz)

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600041770C61F | 01 | 06 | 00 | 04 | 17 | 70 | C6 | 1F |
| 데이터 설명 | $1 \times 16^3 + 7 \times 16^2 + 7 \times 16^1 + 0 \times 16^0 = 6000$ (Communication Data) → divide by 100 → 60.00Hz (Parameter value) | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600041770C61F | 01 | 06 | 00 | 04 | 17 | 70 | C6 | 1F |

(3) 주파수지령 (In case of 50Hz)

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600041388C55D | 01 | 06 | 00 | 04 | 13 | 88 | C5 | 5D |
| 데이터 설명 | $1 \times 16^3 + 3 \times 16^2 + 8 \times 16^1 + 8 \times 16^0 = 5000$ (Communication Data) → divide by 100 → 50.00Hz (Parameter value) | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600041388C55D | 01 | 06 | 00 | 04 | 13 | 88 | C5 | 5D |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.6 운전 지령

2.6.1 운전 지령 방법 설정

통신으로 운전 지령을 하기 위해서는 인버터 기능 코드를 아래와 같이 설정해 주십시오.

(1) 기능코드

| 기능코드 | 기능명칭 | 초기값 | 데이터 | 내용 |
|------|----------|-----|-----|---------------------------|
| F011 | 운전 지령 방법 | 0 | 1 | Terminal (통신 Data : 0001) |
| | | | 2 | OPE (통신 Data : 0002) |
| | | | 3 | COM (통신 Data : 0003) |
| | | | 4 | OPT1 (통신 Data : 0004) |
| | | | 5 | OPT2 (통신 Data : 0005) |

(2) 통신으로 상기 기능코드 변경을 하는 경우에 Data Frame은 아래와 같습니다.

① 송신프레임 (인버터 국번이 1인 경우)

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|---------------------------------|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 0106021100039876 | 01 | 06 | 02 | 12 | 00 | 03 | 68 | 76 |
| 데이터 설명 | | 0002 → 운전 지령 방법을 COM(통신)으로 설정함. | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 0106021100039876 | 01 | 06 | 02 | 12 | 00 | 03 | 68 | 76 |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

2.6.2 전송 프레임 예제

(1) 통신 Data 분석 기준

- ① 국번 : 01 (Address 1번 인버터 지정)
- ② 지령 : 06 (Setting request)
- ③ Parameter : 0002 (운전 지령)

(2) STOP 지령

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020000280A | 01 | 06 | 00 | 02 | 00 | 00 | 28 | 0A |
| 데이터 설명 | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020000280A | 01 | 06 | 00 | 02 | 00 | 00 | 28 | 0A |

(3) FWD(Forward) RUN 지령

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020001E9CA | 01 | 06 | 00 | 02 | 00 | 01 | E9 | CA |
| 데이터 설명 | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020001E9CA | 01 | 06 | 00 | 02 | 00 | 01 | E9 | CA |

N700 인버터 RS232/RS485 통신 인터페이스 지침서

(4) REV(Reverse) RUN 지령

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020002A9CB | 01 | 06 | 00 | 02 | 00 | 02 | A9 | CB |
| 데이터 설명 | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020002A9CB | 01 | 06 | 00 | 02 | 00 | 02 | A9 | CB |

(5) 출력주파수 차단 지령

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020003680B | 01 | 06 | 00 | 02 | 00 | 03 | 68 | 0B |
| 데이터 설명 | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 010600020003680B | 01 | 06 | 00 | 02 | 00 | 03 | 68 | 0B |

(6) RST(Reset) 지령

① 송신프레임

| 송신 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|---|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 01060002000429C9 | 01 | 06 | 00 | 02 | 00 | 04 | 29 | C9 |
| 데이터 설명 | 0000 : STOP 0001 : FWD, 0002 : REV, ,0003 : Cutoff 0004 : RST | | | | | | | |

② 응답프레임

| 응답 (8Byte) | 국번 | 지령 | 파라미터 | | 데이터 | | CRC 16 | |
|------------------|----|----|------|----|-----|----|--------|----|
| | | | | | | | Hi | Lo |
| 01060002000429C9 | 01 | 06 | 00 | 02 | 00 | 04 | 29 | C9 |

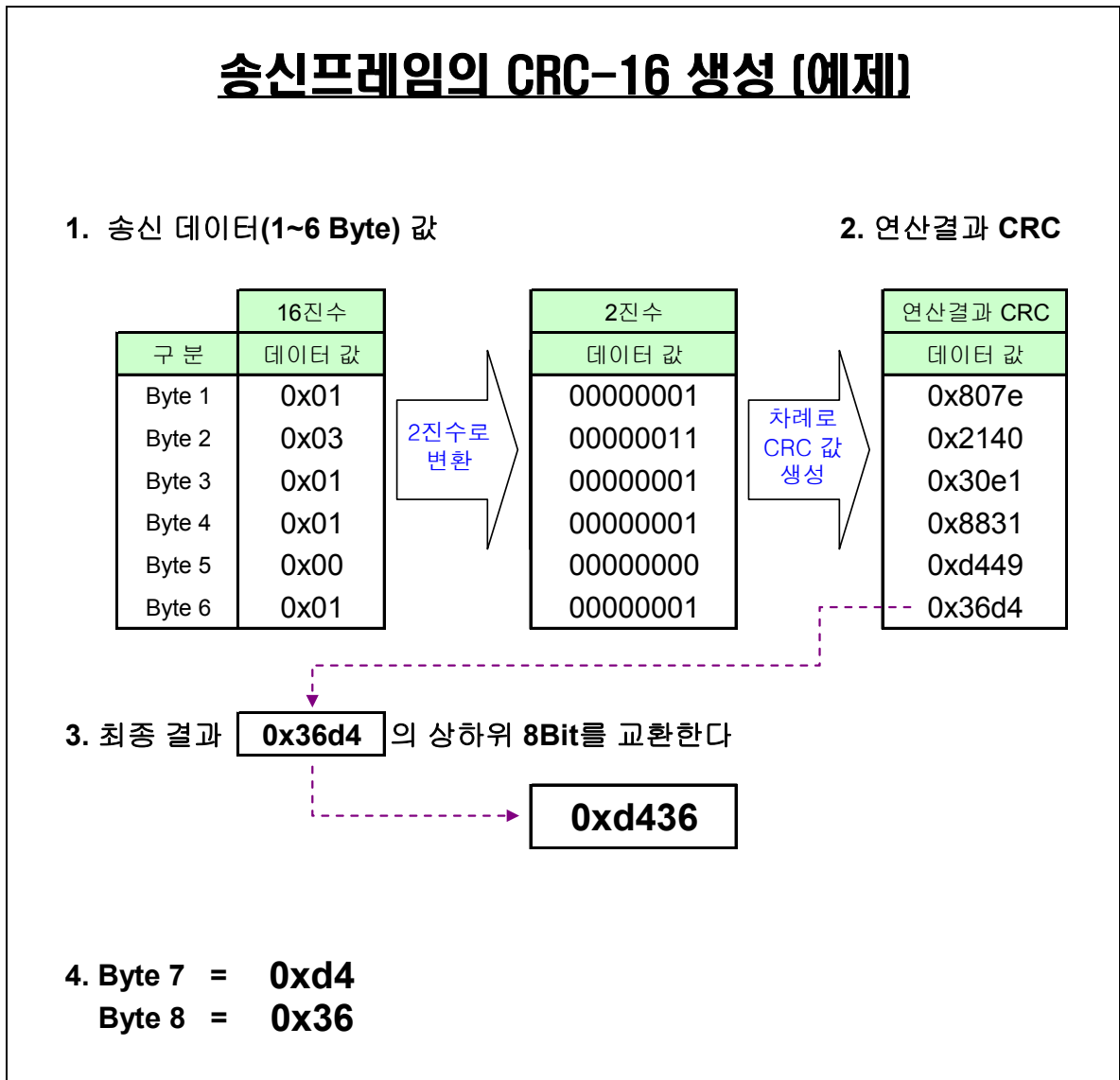
2.7 CRC-16 생성 절차

2.7.1 CRC-16을 생성 절차

- (1) 16Bit 레지스터를 모두 1로 한다. 0xFFFF
- (2) 16Bit 레지스터와 8Bit 데이터를 배타적 논리합(Exclusive OR)을 하여 다시 16Bit 한다.
- (3) 레지스터에 입력한다.
- (4) 16Bit 레지스터를 오른쪽으로 1Bit shift 한다.
- (5) Shift되어 나온 Bit가 1 이면 16Bit 레지스터와 0xA001을 Exclusive OR하여 16Bit 레지스터에 입력한다.
- (6) 3에서 4의 항목을 8번 실행한다.
- (7) 보내고자 하는 데이터가 완료될 때까지 Exclusive OR와 8번 shift에 해당하는 2~6항목을 송신하고자 하는 데이터의 Byte 수만큼 반복한다. (Request는 6 Byte 임)
- (8) 위의 결과로 나오는 16Bit 레지스터의 상위와 하위 8Bit를 교환한다.

2.7.2 CRC-16 생성의 예 : D001의 출력주파수를 읽고자 할 경우

| Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 |
|-------|-------|-------|-------|-------|-------|
| 국번 | 지령 | 파라미터 | | 파라미터수 | |
| 0x01 | 0x03 | 0x01 | 0x01 | 0x00 | 0x01 |



N700 인버터 CRC-16 생성 과정 (1/6 step)

16Bit 레지스터 : 0xffff

1. 8bit 데이터 ==> 0x01

2. 8Bit 데이터와 16Bit 레지스터를 Exclusive OR 한다.

3. 오른쪽으로 1Bit shift 한다 (1st)

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

4. 오른쪽으로 1Bit shift 한다 (2nd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

5. 오른쪽으로 1Bit shift 한다 (3rd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

6. 오른쪽으로 1Bit shift 한다 (4th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

7. 오른쪽으로 1Bit shift 한다 (5th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

8. 오른쪽으로 1Bit shift 한다 (6th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

9. 오른쪽으로 1Bit shift 한다 (7th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

10. 오른쪽으로 1Bit shift 한다 (8th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.



상기 CRC 값은 다음 Byte의 16Bit 레지스터 값으로 사용됨.

N700 인버터 CRC-16 생성 과정 (2/6 step)

16Bit 레지스터 : Byte 1의 CRC 연산값

1. 8Bit 데이터 ==> 0x03

2. 8Bit 데이터와 16Bit 레지스터를 Exclusive OR 한다.

3. 오른쪽으로 1Bit shift 한다 (1st)

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

4. 오른쪽으로 1Bit shift 한다 (2nd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

5. 오른쪽으로 1Bit shift 한다 (3rd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

6. 오른쪽으로 1Bit shift 한다 (4th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

7. 오른쪽으로 1Bit shift 한다 (5th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

8. 오른쪽으로 1Bit shift 한다 (6th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

9. 오른쪽으로 1Bit shift 한다 (7th) ==>

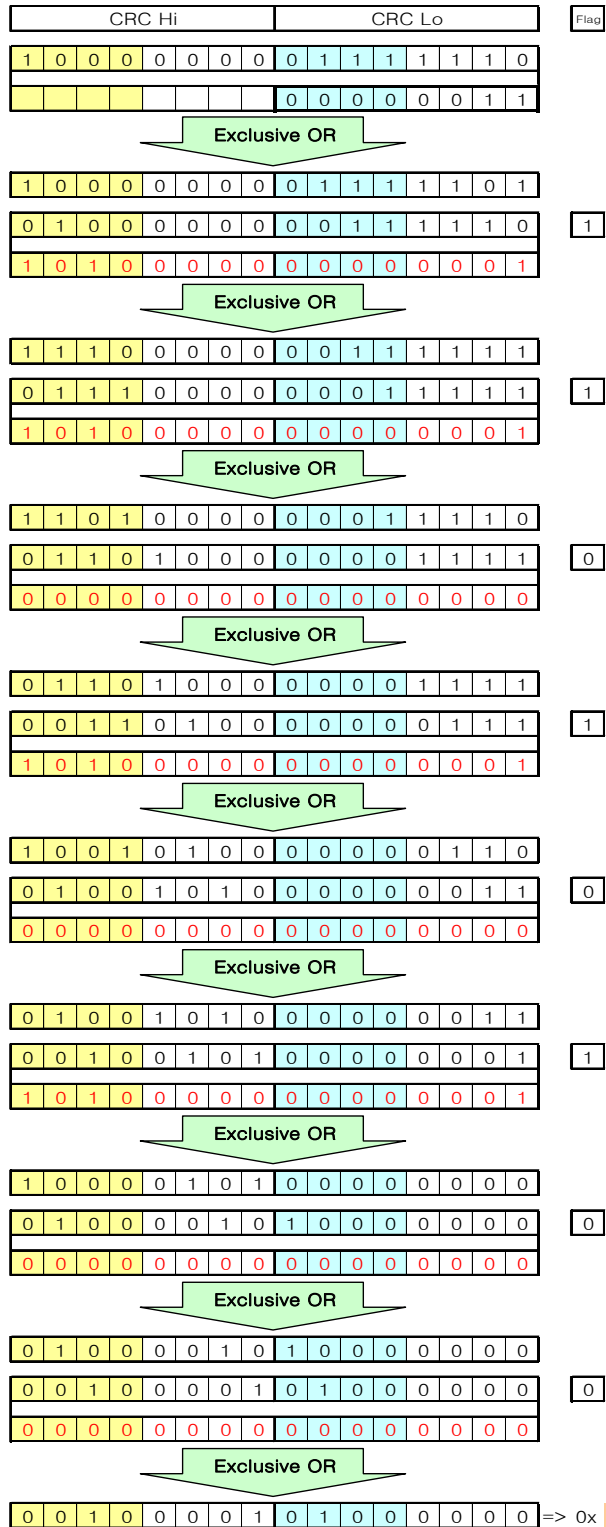
==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

10. 오른쪽으로 1Bit shift 한다 (8th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.



상기 CRC 값은 다음 Byte의 16Bit 레지스터 값으로 사용됨.

N700 인버터 CRC-16 생성 과정 (3/6 step)

16Bit 레지스터 : Byte 2의 CRC 연산값

1. 8bit 데이터 ==> 0x01

2. 8Bit 데이터와 16Bit 레지스터를 Exclusive OR 한다.

3. 오른쪽으로 1Bit shift 한다 (1st)

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

4. 오른쪽으로 1Bit shift 한다 (2nd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

5. 오른쪽으로 1Bit shift 한다 (3rd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

6. 오른쪽으로 1Bit shift 한다 (4th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

7. 오른쪽으로 1Bit shift 한다 (5th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

8. 오른쪽으로 1Bit shift 한다 (6th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

9. 오른쪽으로 1Bit shift 한다 (7th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

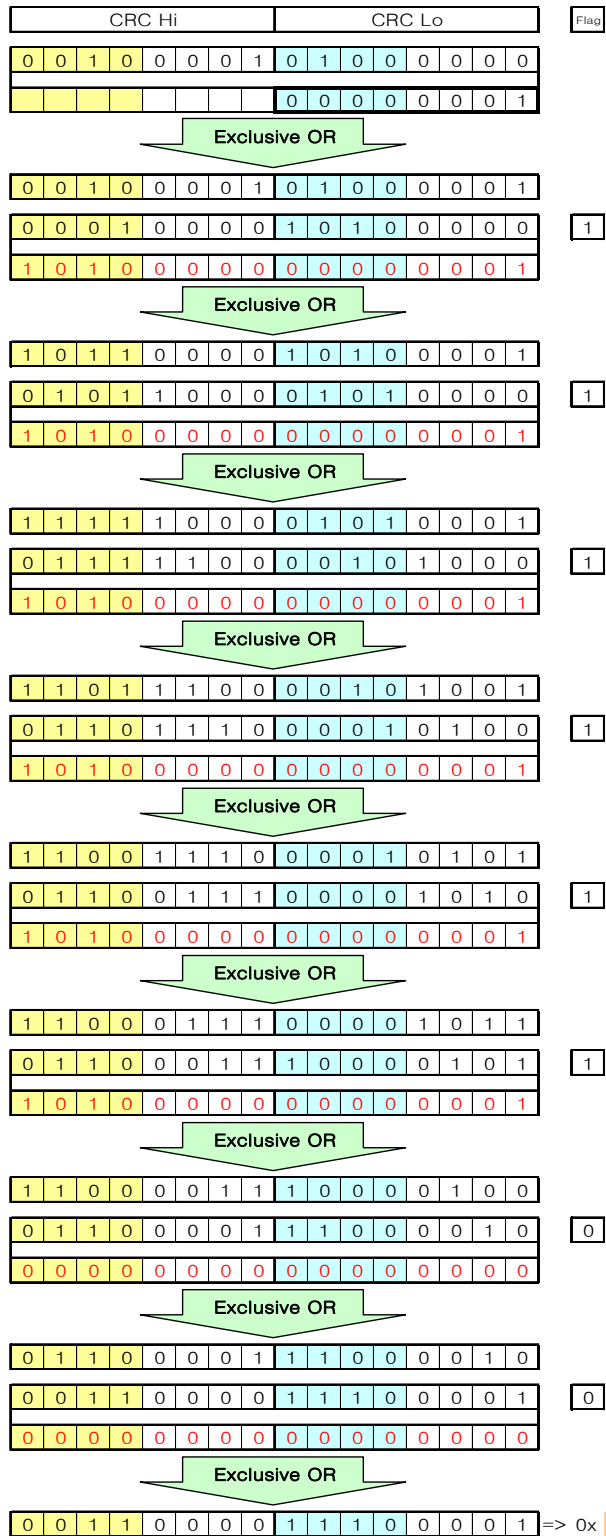
오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

10. 오른쪽으로 1Bit shift 한다 (8th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.



상기 CRC 값은 다음 Byte의 16Bit 레지스터 값으로 사용됨.

N700 인버터 CRC-16 생성 과정 (4/6 step)

16Bit 레지스터 : Byte 3의 CRC 연산값

1. 8bit 데이터 ==> 0x01

2. 8Bit 데이터와 16Bit 레지스터를 Exclusive OR 한다.

3. 오른쪽으로 1Bit shift 한다 (1st)

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

4. 오른쪽으로 1Bit shift 한다 (2nd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

5. 오른쪽으로 1Bit shift 한다 (3rd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

6. 오른쪽으로 1Bit shift 한다 (4th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

7. 오른쪽으로 1Bit shift 한다 (5th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

8. 오른쪽으로 1Bit shift 한다 (6th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

9. 오른쪽으로 1Bit shift 한다 (7th) ==>

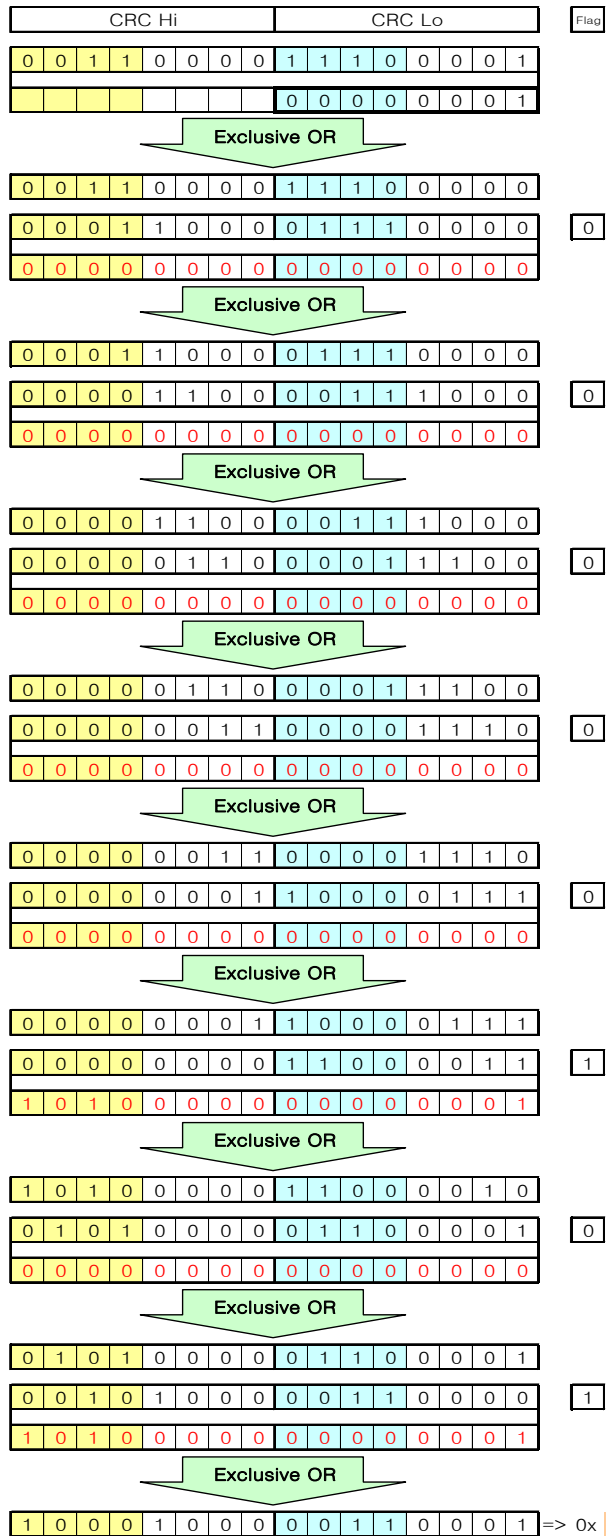
==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

10. 오른쪽으로 1Bit shift 한다 (8th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.



상기 CRC 값은 다음 Byte의 16Bit 레지스터 값으로 사용됨.

N700 인버터 CRC-16 생성 과정 (5/6 step)

16Bit 레지스터 : Byte 4의 CRC 연산값

1. 8Bit 데이터 ==> 0x00

2. 8Bit 데이터와 16Bit 레지스터를 Exclusive OR 한다.

3. 오른쪽으로 1Bit shift 한다 (1st)

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

4. 오른쪽으로 1Bit shift 한다 (2nd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

5. 오른쪽으로 1Bit shift 한다 (3rd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

6. 오른쪽으로 1Bit shift 한다 (4th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

7. 오른쪽으로 1Bit shift 한다 (5th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

8. 오른쪽으로 1Bit shift 한다 (6th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

9. 오른쪽으로 1Bit shift 한다 (7th) ==>

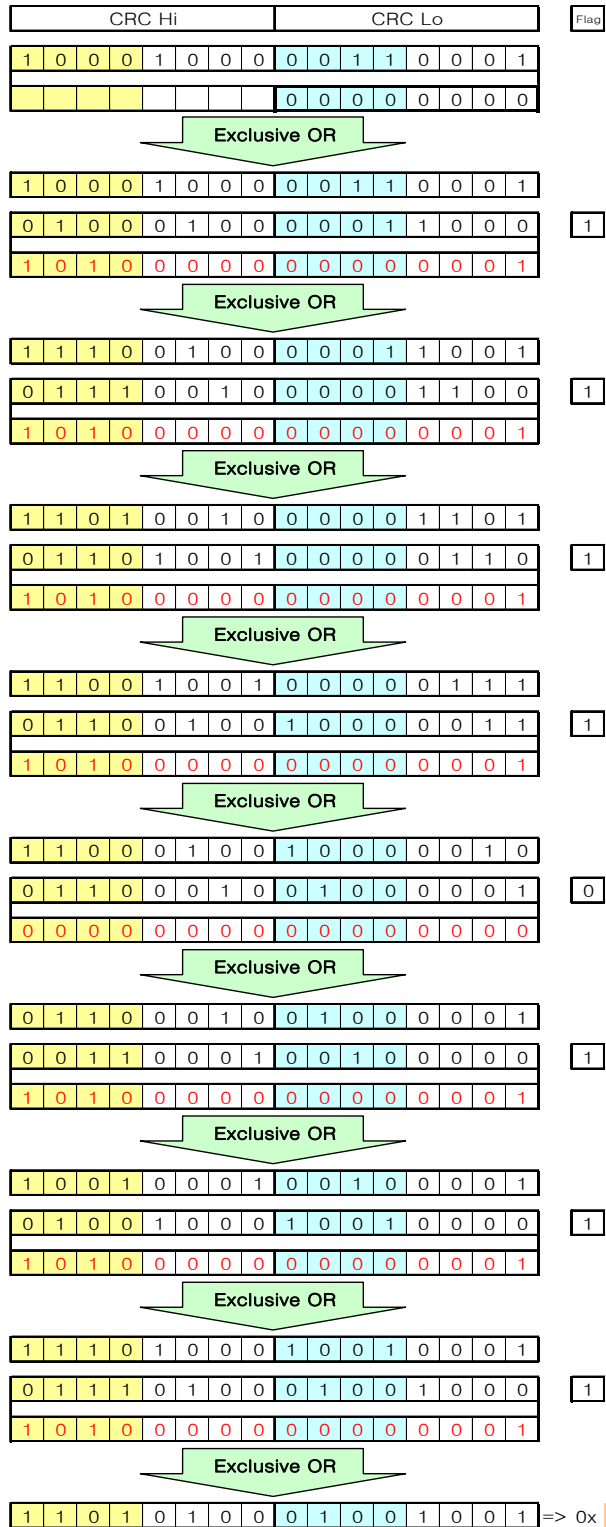
==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

10. 오른쪽으로 1Bit shift 한다 (8th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다. 오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.



상기 CRC 값은 다음 Byte의 16Bit 레지스터 값으로 사용됨.

N700 인버터 CRC-16 생성 과정 (6/6 step)

16Bit 레지스터 : Byte 5의 CRC 연산값

1. 8bit 데이터 ==> 0x01

2. 8Bit 데이터와 16Bit 레지스터를 Exclusive OR 한다.

3. 오른쪽으로 1Bit shift 한다 (1st)

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

4. 오른쪽으로 1Bit shift 한다 (2nd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

5. 오른쪽으로 1Bit shift 한다 (3rd) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

6. 오른쪽으로 1Bit shift 한다 (4th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

7. 오른쪽으로 1Bit shift 한다 (5th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

8. 오른쪽으로 1Bit shift 한다 (6th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

9. 오른쪽으로 1Bit shift 한다 (7th) ==>

==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

10. 오른쪽으로 1Bit shift 한다 (8th) ==>

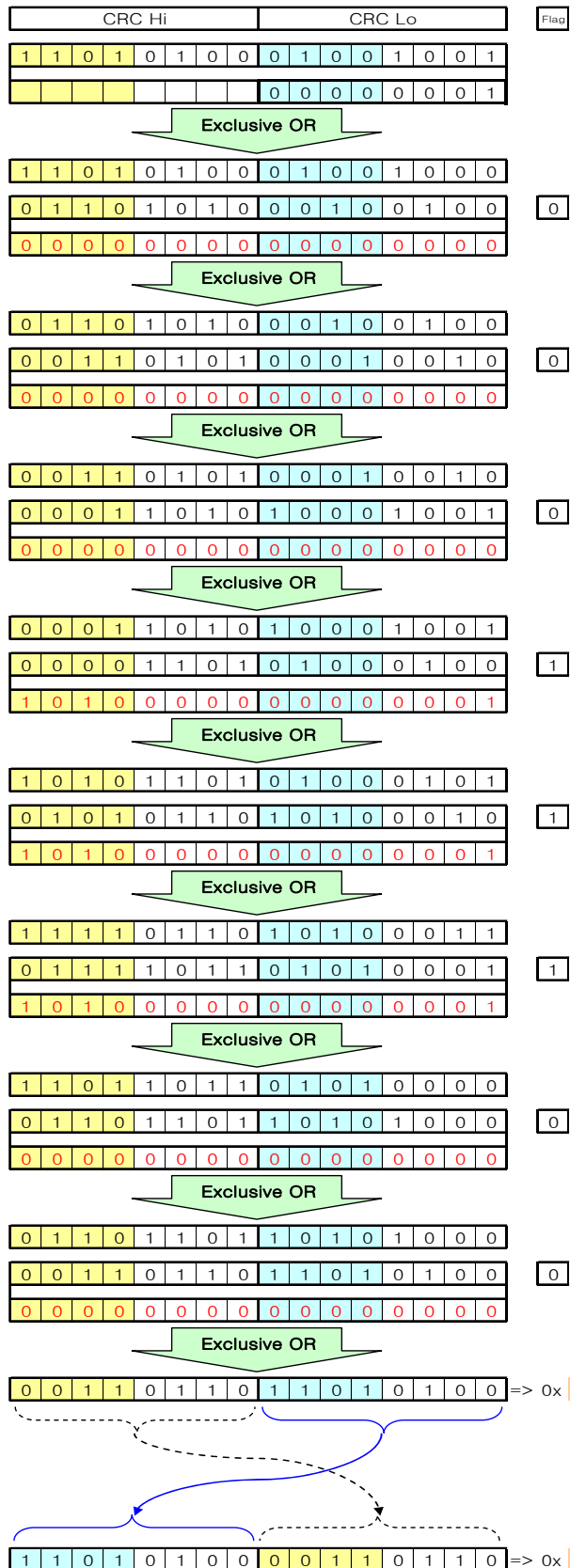
==> Shift 한 값

오른쪽으로 Shift한 Flag값이 1이면, 16Bit 레지스터와 0xa001을 Exclusive OR 한다.

오른쪽으로 Shift한 Flag값이 0이면, 16Bit 레지스터와 0x0000을 Exclusive OR 한다.

11. 위의 결과로 나온 16Bit 레지스터의 값중, 상하위 8Bit를 교환한다.

==> 상하위 8Bit 값을 교환한 결과

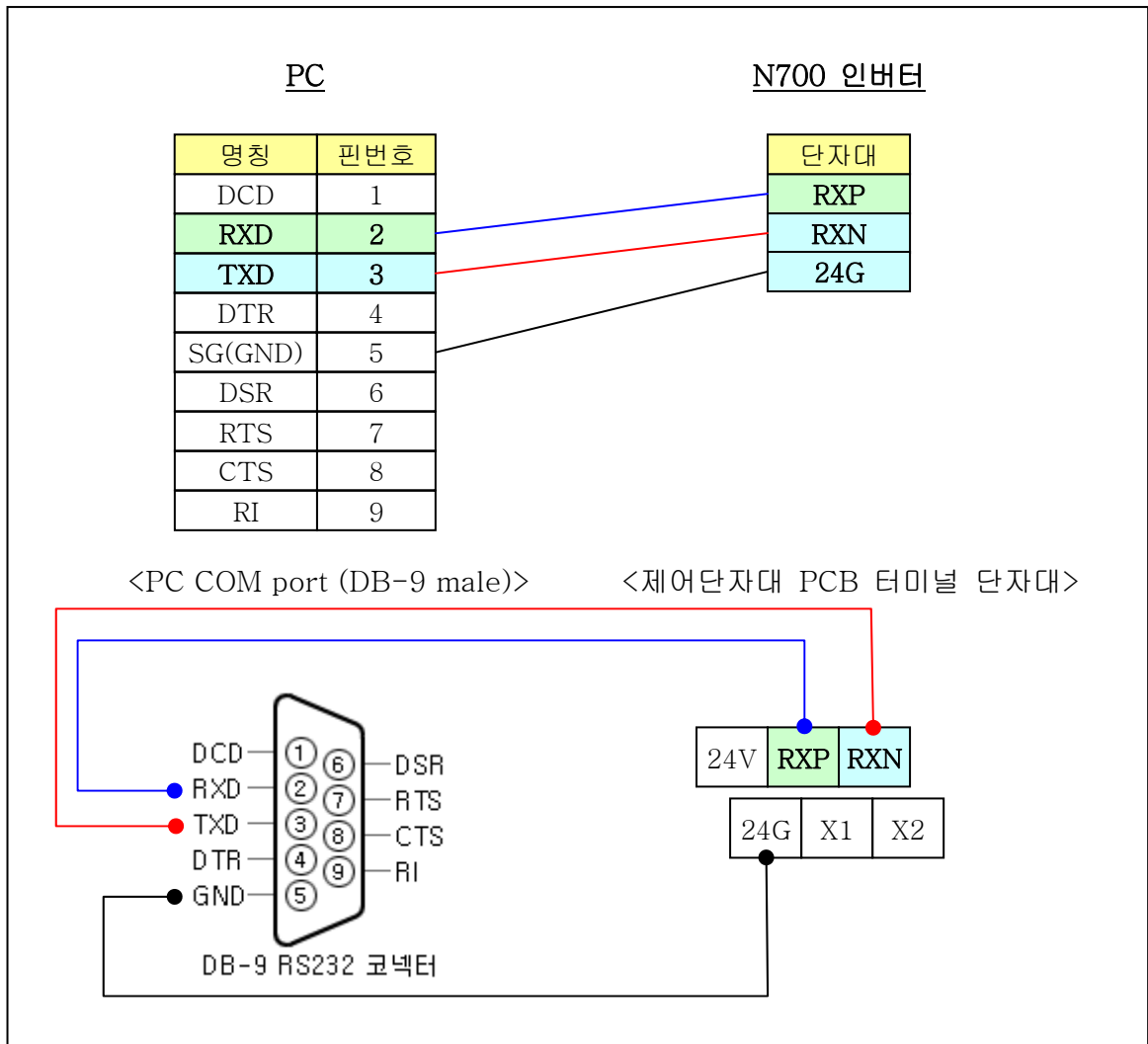


3장. 통신 관련 Trouble Shooting

3.1 PC와 인버터간 통신이 되지 않습니다. (Case 1)

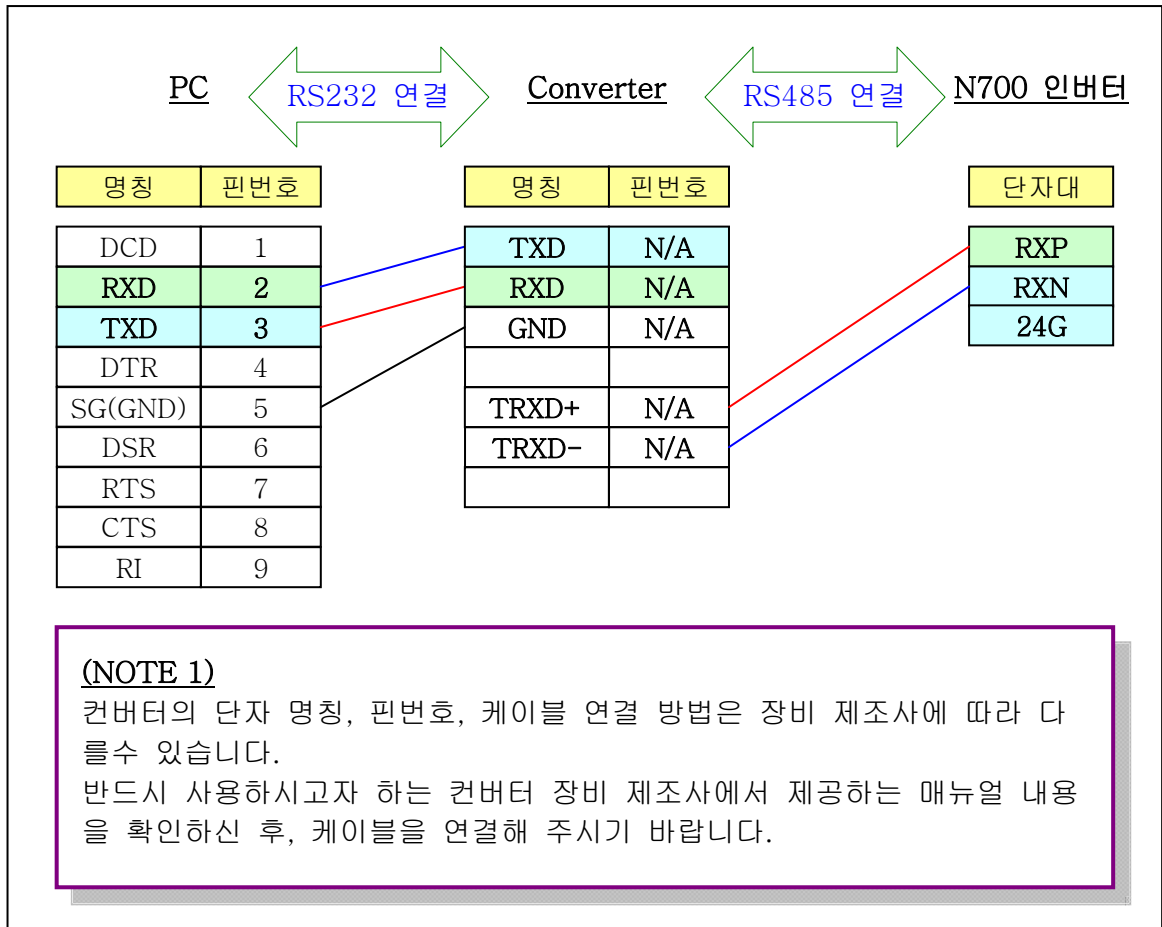
3.1.1 Step 1 : PC ~ 인버터간 Cable 연결을 점검합니다.

(1) RS232 통신을 사용하는 경우



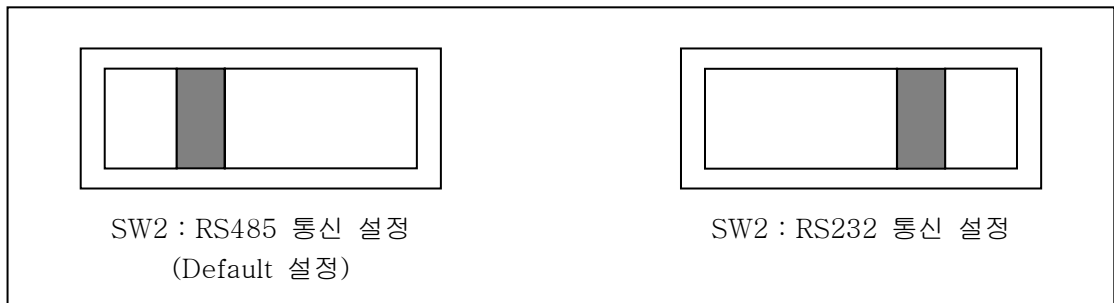
N700 인버터 RS232/RS485 통신 인터페이스 지침서

(2) RS485 통신을 사용하는 경우



3.1.2 Step 2 : 인버터 제어단자대의 통신 선택 스위치 상태를 점검합니다.

(1) RS232 & RS485 통신방법 선택 스위치



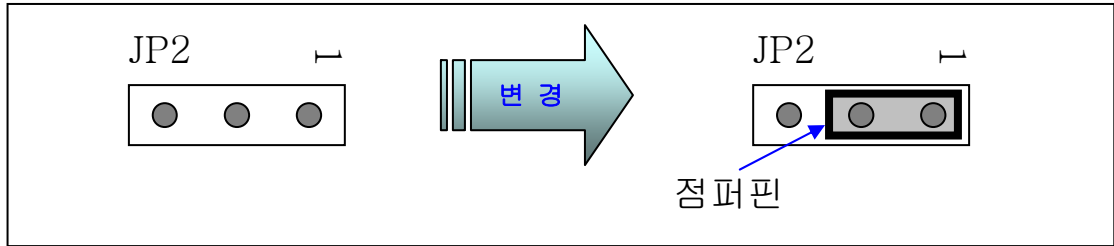
(2) 사용하고계신 인버터의 통신 선택 스위치 상태가 올바른지 확인하시고, 잘못된 경우는 위 그림과 같이 SW2 스위치를 조정해 주십시오.

N700 인버터 RS232/RS485 통신 인터페이스 지침서

3.1.3 Step 3 : RS485 통신인 경우, N번째 인버터의 종단저항 설정을 점검합니다.

(1) 인버터 제어단자대 PCB의 J2 점퍼핀을 단락시켜 종단저항을 유효로 설정한다.
(Multi-drop 방식의 RS485 통신인 경우는 종단 인버터에만 설정함.)

(2) 인버터 제어단자대 PCB의 J2 점퍼핀



3.1.4 Step 4 : 인버터 오퍼레이터에서 통신 관련 설정을 점검합니다.

| Function Code | 초기설정값 | RS485 | RS232 |
|-----------------|---------------|---------------|---------------|
| b037 (데이터지령 선택) | 0 (Operator) | 1 (RS485) | 4 (RS232) |
| b038 (통신 속도) | 2 (9600BPS) | 2 (9600BPS) | 2 (9600BPS) |
| b039 (통신 국번) | 1 (Address 1) | 1 (Address 1) | 1 (Address 1) |
| b040 (통신 비트 길이) | 8 (8 Bit) | 8 (8 Bit) | 8 (8 Bit) |
| b041 (통신 패리티) | 0 (패리티 없음) | 0 (패리티 없음) | 0 (패리티 없음) |
| b042 (통신 정지 비트) | 1 (1 Bit) | 1 (1 Bit) | 1 (1 Bit) |

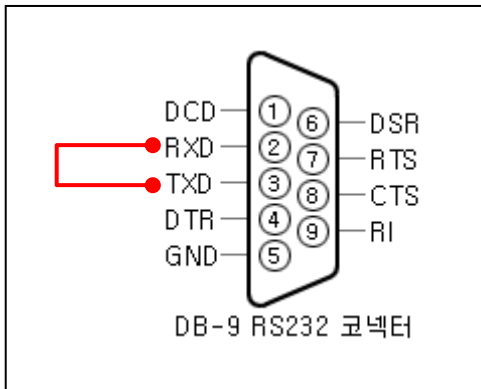
* Note : RS485 통신으로 여러대(1~N, 최대 32대)의 인버터와 통신을 하는 경우에는 인버터의 b039 Function Code값을 1~N까지 순서대로 설정하시면 됩니다.

3.1.5 Step 5 : PC의 COM Port가 정상 동작하는지 점검합니다.

Windows OS에서 기본으로 제공하는 하이퍼 터미널을 사용하여 COM Port 정상동작 상태 여부를 확인하는 것이 가능합니다.

(1) PC의 COM port 조작

아래와 같이 PC측 COM port의 2번핀과 3번핀을 SHORT 시켜 주십시오.

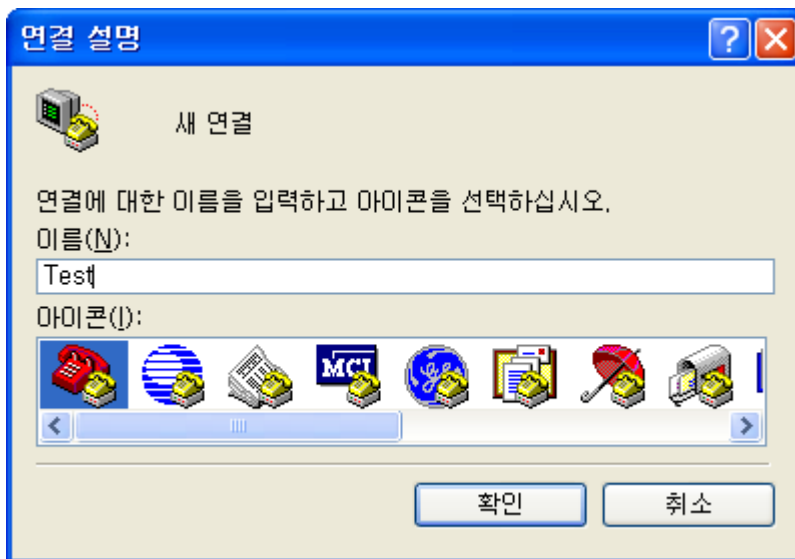


(2) 하이퍼터미널 통신 설정 (Windows XP 기준)

Windows 시작 메뉴에서 [모든프로그램] – [보조프로그램] – [통신]으로 이동한 후 “하이퍼터미널”을 마우스 왼쪽 버튼을 클릭하여 실행합니다.

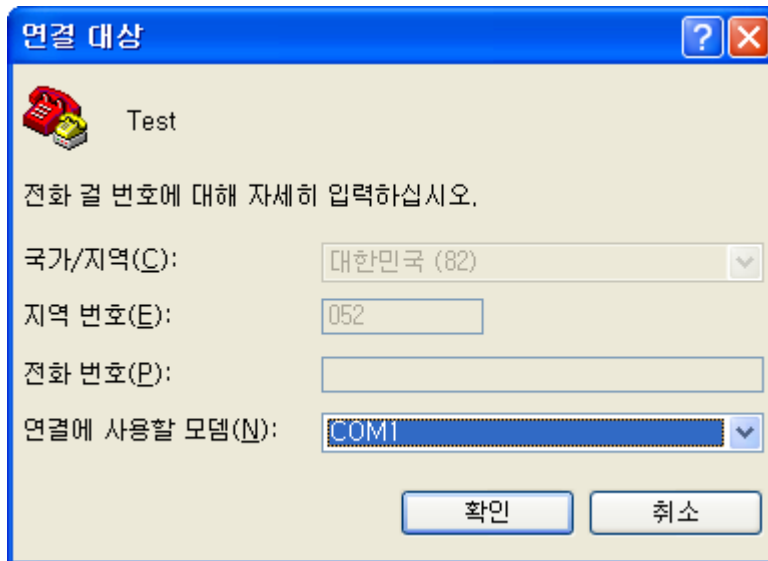
Windows 버전에 따라 팝업창 모양이 다를수 있으나, 시험 방법은 동일합니다.

- ① 이름을 임의로 입력한 후 “확인” 버튼을 클릭합니다.
→ “Test”라고 입력

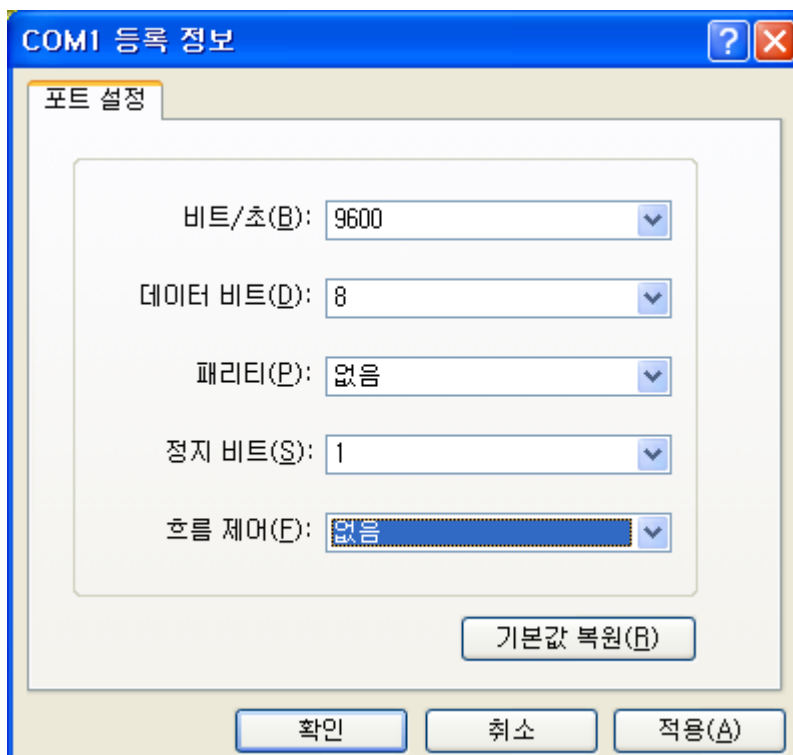


N700 인버터 RS232/RS485 통신 인터페이스 지침서

- ② 사용하시는 COM Port 번호를 선택한 후 “확인” 버튼을 클릭합니다.
→ “COM1” 선택

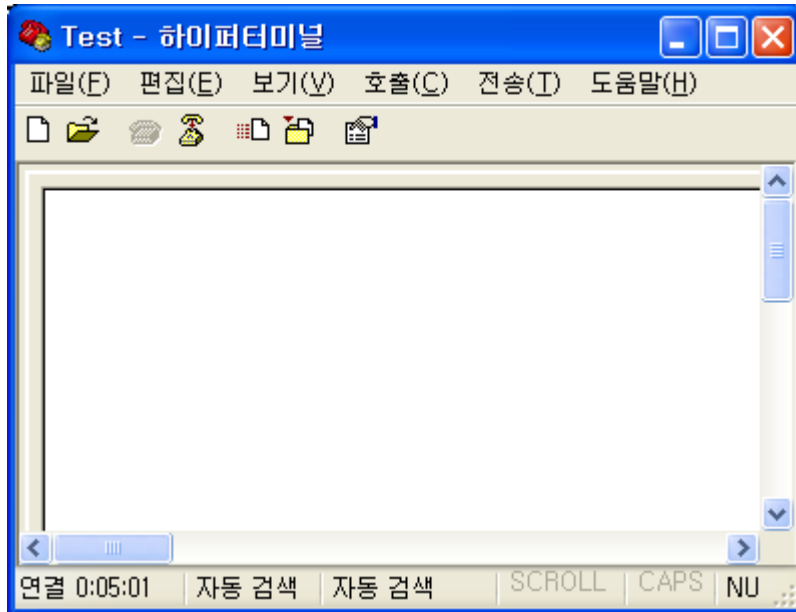


- ③ 아래 그림과 같이 COM Port의 통신 설정을 하고 “적용” 버튼과 “확인” 버튼을 차례로 클릭합니다.



N700 인버터 RS232/RS485 통신 인터페이스 지침서

- ④ COM Port에 대한 통신 준비가 완료되었으며, 아래와 같은 팝업 통신창이 뜹니다.

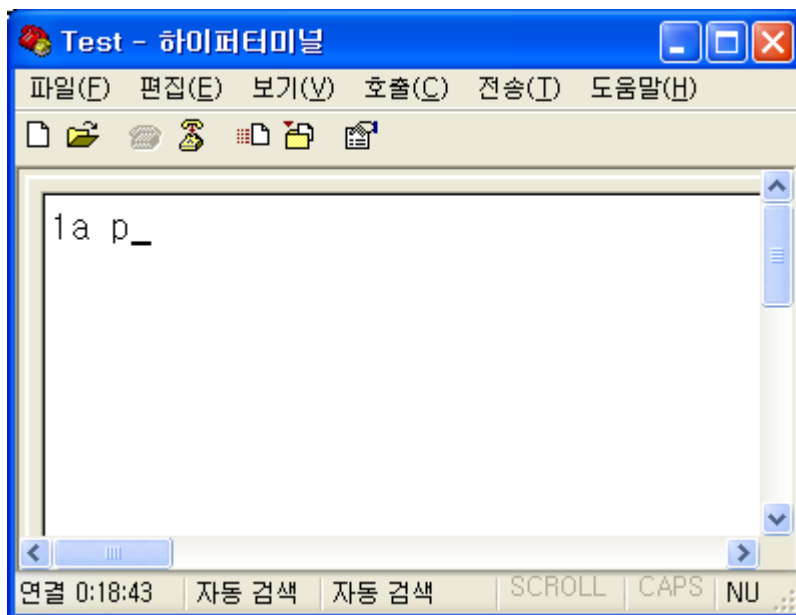


- ⑤ 이제, COM Port에 대한 Test 준비가 완료되었습니다.

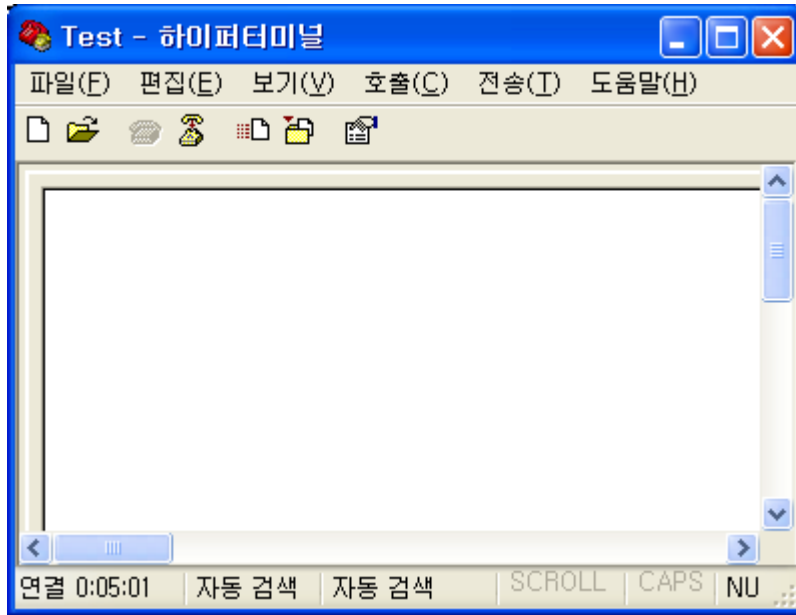
- ⑥ Keyboard에서 임의의 버튼을 쳐보시기 바랍니다.

숫자 “1”, 문자 “a”, 스페이스바 “ ”, 문자 “p”를 차례대로 쳐서 “하이퍼터미널 통신창”에 아래와 같이 문자가 순서대로 수신되어 나타나면 PC측 COM Port는 정상 동작하는 것으로 판단할 수 있습니다.

(하이퍼터미널 창에서는, 송신데이터는 표시되지 않으며 수신 데이터만 표시됩니다.)



- ⑦ 그러나, 상기 ⑥번 항목의 시험을 진행한 후에도 아래와 같이 “하이퍼터미널 통신창”에 아무런 변화가 없으면 PC측 Com Port는 고장난 것으로 판단할 수 있습니다.



3.1.6 Step 6 : 인버터 통신 Port 상태를 점검합니다.

인버터 자체 오류나 통신 Port 점검을 위해서는 별도의 시험장비와 회로 Check가 필요하오니, 사용자께서는 Step 7에서 PC ~ 인버터간 통신시험을 진행하십시오.

3.1.7 Step 7 : PC ~ 인버터간 통신 상태를 점검합니다.

Step 1에서 Step 5까지 각 항목에 대한 설정을 점검/수정한 후 PC ~ 인버터간 통신을 시도해 보시기 바랍니다.

만일, 통신이 전혀 되지 않는다면, 인버터 자체의 통신회로 고장 가능성을 의심할 수 있으니, 고객분들께서는 당사 고객센터로 연락하여 기술지원을 요청하시기 바랍니다.

3.2 인버터 운전중에, 통신으로 운전방향 전환이 안됩니다. (Case 2)

3.2.1 Step 1 : 인버터 운전방향제한(B001) 코드값을 점검합니다.

인버터 운전방향제한(B001) 코드값이 올바르게 설정되어 있는지 점검한 후, 아래 표를 참조하여 B001 코드값을 적당한 값으로 변경해 주시기 바랍니다.

| 기능코드 | 기능명칭 | 초기값 | 데이터 | 내용 |
|------|---------|-----|-----|---------------------------|
| B001 | 운전방향 제한 | 0 | 0 | 정,역운전 유효 (통신 Data : 0000) |
| | | | 1 | 정운전만 유효 (통신 Data : 0001) |
| | | | 2 | 역운전만 유효 (통신 Data : 0002) |

3.2.2 Step 2 : 통신으로 인버터 운전(정/역방향)을 실시합니다.

- (1) 운전 (정운전/역운전/정지) 실시
- (2) 인버터가 여전히 한쪽 방향으로만 운전이 되거나, 정상동작 하지 않으면 Step 3을 점검해 보십시오.

3.2.3 Step 3. 인버터에서 오퍼레이터로 정/역방향 운전이 되는지 점검합니다.

- (1) 오퍼레이터로 운전하기 위해서는 아래 기능 코드값을 변경해 주시기 바랍니다.

| Function Code | 코드 값 | 설명 |
|-----------------|------|----------------|
| b037 (데이터지령 선택) | 0 | 오퍼레이터 데이터지령 모드 |
| F011 (운전지령 방법) | 2 | 오퍼레이터 운전지령 모드 |

- (2) 인버터를 오퍼레이터로 운전하여 정운전/역운전이 되는지 확인합니다.
- (3) 인버터가 정상으로 동작한다면 통신 관련 기능코드 설정값을 재 확인한 후, 통신으로 운전을 해 보십시오.

3.2.4 Step 4. 문제점이 해결되지 않으면, 당사 고객센터로 기술지원을 요청하시기 바랍니다.