

Rhio232

Serial I/O Manager

사용자 설명서

Version 1.0.3

2005-11-08

Rhio232 사용 설명서

버전 1.0.3

펌웨어 버전 1.3.2

Printed in Korea

저작권

Copyright 2002~2005, 세나테크놀로지. All rights reserved.

세나테크놀로지는 자사 제품을 예고 없이 변경 및 개선할 수 있는 권리를 가지고 있습니다.

등록 상표

Rhio™은 (주)세나테크놀로지의 등록 상표입니다.

Windows(r)는 Microsoft Corporation의 등록 상표입니다.

Ethernet(r)은 XEROX Corporation의 등록 상표입니다.

사용자 고지

시스템 고장이 심각한 결과를 유발하는 응용 분야인 경우, 백업 시스템이나 안전 장치를 사용하여 심각한 결과로부터 인명 및 재산을 보호하는 것이 필요합니다. 시스템 고장으로 인한 결과에 대한 보호는 사용자 책임입니다.

본 기기는 생명 유지 또는 의료 시스템 승인을 받지 않은 제품입니다.

본 기기에 대하여 세나테크놀로지의 서면 허가 없이 이루어진 변경 또는 개조에 따른 어떠한 결과에 대해서도 세나테크놀로지는 책임지지 않습니다.

기술 지원

(주)세나테크놀로지

서울시 서초구 양재동 210번지

137-130, 대한민국

Tel: (02) 573-5422

Fax: (02) 573-7710

E-Mail: support@sena.com

Website: www.sena.com

Revision History

| Revision | Date | Name | Description |
|----------|------------|-----------|---------------------------------|
| V1.0.0 | 2005-05-21 | D.H. Shin | Initial Release |
| V1.0.1 | 2005-08-08 | D.H. Shin | " 표 2-2. Rhio10의 터미널 블록 배치 " 추가 |
| V1.0.2 | 2005-11-02 | D.H. Shin | 오타 수정. |
| V1.0.3 | 2005-11-08 | J.S. Kim | 동작온도, 보관온도의 범위값 변경 |

목 차

| | |
|-------------------------------------|----|
| 1. 서론 | 6 |
| 1.1 개요 | 6 |
| 1.2 패키지 체크 리스트 | 7 |
| 1.3 제품 사양 | 8 |
| 2. 시작하기 | 10 |
| 2.1 패널 배치 | 10 |
| 2.2 하드웨어 연결 | 11 |
| 2.2.1 설치 및 부착 | 12 |
| 2.2.2 전원 연결 | 13 |
| 2.2.3 시리얼 장비에 연결 | 14 |
| 2.3 Rhio Manager 설치 | 15 |
| 2.3.1 Rhio Manager 설치 | 15 |
| 2.3.2 Rhio Manager를 이용한 기본 설정 | 16 |
| 2.4 공장 출하 시 기본값 복구 | 17 |
| 3. I/O 설정 및 응용 | 18 |
| 3.1 I/O 감시 및 제어 | 18 |
| 3.1.1 LED | 18 |
| 3.1.2 감시 주기 설정 | 19 |
| 3.1.3 I/O Port 감시 동작 | 20 |
| 3.1.4 Digital Output Port 제어 | 20 |
| 3.2 Digital Input 설정 | 20 |
| 3.2.1 Enable/Disable 설정 | 21 |
| 3.3 Digital Output Port 설정 | 22 |
| 3.3.1 Enable/Disable 설정 | 22 |
| 3.3.2 동작 조건 설정 | 23 |
| 3.3.3 Delay 와 Pulse 동작 | 24 |
| 3.3.4 정전 후 복구 설정 | 24 |
| 3.4 ADC Input Port 설정 | 24 |
| 3.4.1 Reference 설정 | 25 |
| 3.4.2 ADC Input Port 동작 설정 | 25 |
| 3.5 I/O Port 연결 방법 | 26 |
| 3.5.1 Digital Output Port | 26 |
| 3.5.2 Digital Input Port | 26 |
| 3.5.3 ADC Input Port | 27 |

| | |
|---|----|
| 4. 소프트웨어 개발 및 응용 | 29 |
| 4.1 Rhio 라이브러리..... | 29 |
| 4.1.1 Rhio 라이브러리 개요..... | 29 |
| 4.1.2 Reference..... | 29 |
| 4.2 Rhio 라이브러리를 이용한 샘플 프로그램 제작 및 데모 | 32 |
| 4.2.1 프로그램 UI 구성 및 관련 클래스..... | 32 |
| 4.2.2 처리 절차..... | 33 |
| 4.3 Rhio 통신 프로토콜..... | 37 |
| 4.3.1 개요..... | 37 |
| 4.3.2 ON/OFF 제어..... | 40 |
| 4.3.3 Input/Output 상태..... | 42 |
| 4.3.4 설정/동작 | 42 |
| 4.3.5 Output Port 설정 | 43 |
| 4.3.6 ADC Input Port 설정 | 46 |
| 4.3.7 Port Enable 설정..... | 47 |
| 4.3.8 Port 정전 후 복구 설정..... | 50 |
| 5. 세나 디바이스 서버와 Rhio232 연결 | 52 |
| 5.1 연결 | 52 |
| 5.2 세나 디바이스 서버와 Rhio232의 응용..... | 54 |
| 부록 A. 연결 | 59 |
| A.1 시리얼 포트 Pin-Out | 59 |
| A.2 시리얼 연결 배선도..... | 60 |
| 부록 B. 문제 해결 | 61 |
| B.1 전원/LED 상태..... | 61 |
| B.2 시리얼 통신..... | 61 |
| 부록 C. Rhio 라이브러리 | 62 |
| C.1 Enumeration Type..... | 62 |
| C.2 Structure | 64 |
| C.3 Function..... | 66 |

1. 서론

1.1 개요

Rhio232는 디바이스 서버가 입/출력 장치의 제어 및 감시를 할 수 있도록 하는 시리얼 입/출력 관리자입니다. Rhio232는 디바이스 서버와 RS232 통신을 할 수 있습니다.

Rhio232는 릴레이에 의해 최대 10개의 Digital Output 접점들을 220V 레벨까지 제어할 수 있음, Input 접점 상태와 연동하여 AND, OR, NOT, DELAY 등의 간단한 논리 연산을 통해 Output 상태를 제어할 수 있습니다. 또한, Rhio232는 최대 12개의 절연 Digital Input 접점들을 감시할 수 있는 기능을 제공합니다. Rhio232의 Analog Port들은 threshold 설정을 통해 일반적인 Digital Input과 동일한 방법으로 응용하는 모드와 수치 자체로서 데이터를 획득하는 모드를 제공합니다. Digital/Analog Input 시스템은 상태가 변화할 때 데이터를 보내도록 하는 event-driven 방식으로 동작합니다.

사용자들은 Windows 응용 프로그램인 Rhio Manager를 이용하여 제품을 설정하고, 시스템 I/O 시험 및 감시를 수행할 수 있으며, 제공되는 Windows MFC DLL 라이브러리를 사용하여 목적에 맞는 응용 프로그램을 개발할 수도 있습니다.

Rhio232는 단독으로 또는 디바이스 서버와 연동하여 원격 데이터 취합 시스템, 원격 분산 I/O 시스템, 자동화 기기 모니터링, 이산 제어 및 분산 데이터 취합 시스템 분야 등에서의 일반적인 요구사항을 수용할 수 있도록 설계되었습니다.

본 매뉴얼을 이해하려면 인터넷 프로토콜의 개념을 어느 정도 알고 있어야 합니다. 이런 개념을 잘 모르는 경우에는 각 주제에 대한 명세 또는 문서를 참조하십시오.

1.2 패키지 체크 리스트

- Rhio232 외장 박스
- CAT5 케이블
- RJ45 to DB9 Female 케이블 어댑터
- DIN 레일 거치형 키트
- Quick Start Guide
- Rhio Manager, Rhio DLL 및 사용자 설명서가 포함된 CD-ROM

1.3 제품 사양

| | |
|-------------------------|---|
| Serial Interface | RS232 시리얼 통신 지원, RJ45 커넥터 Baud rate: 9,600/Flow control: None/Data: 8 bit/Stop: 1 bit |
| Digital Input | -Input 개수: 12 -Input 방식: 전압 -Input 회로: 절연 photo-coupler -Input 전압: 0V ~ ±24V OFF 0V ~ ±1.2V, ON ±3.3V ~ ±24V -Input 검지시간: 20ms -Input 절연전압: 5KV |
| Digital Output | - Output 개수: 10 - Output 방식: Relay 접점Output - Output 정격부하: 3A/240VAC - 점접절연저항: 1000MΩ Min (DC500V) - Output 절연전압(coil and contact): 4KV - Output 지연시간 : 10msec이하 |
| Analog Input | - Input 개수: 4 - 분해능: 10-bit - Input 방식: 전압, Direct Coupling 방식 - Input 범위: 0V ~ Aref (레퍼런스 전압, 2~5V) - Input 검지시간: 1000 samples/sec |
| 관리 | - Rhio Manager Windows Utility |
| 소프트웨어 지원 | - Windows MFC DLL library - I/O 설정, 상태 감시 및 제어 |
| I/O 표시부 | - Power, Serial Rx, Serial Tx - Digital Output(Digital Output), 1~10 - Digital Input(Digital Input), 1~12 - Analog Input(Analog Input), 1~4 |
| 전원 | - 공급전압 9V~48VDC, 최대 5W |
| 환경 | - 작동 온도: 0℃ to 50℃ - 보관 온도: -20℃ to 66℃ - 습도: 90%이하(단, 결로현상이 없을 것) |
| 물리적 특성 | 137 x 111 x 58 (mm), 5.4 x 4.4 x 2.3 (in.) 무게 730g |

| | |
|---------|------------------|
| 인증 | FCC (A), CE, MIC |
| 품질 보증기간 | 5년 |

2. 시작하기

이 장은 먼저 Rhio232를 설정 및 구성하는 방법을 설명합니다.

- 2.1 패널 배치에서는 패널의 배치와 LED 표시등에 대해 설명합니다.
- 2.2 하드웨어 연결에서는 Rhio232를 DIN 레일에 부착 및 설치 하는 방법과 Rhio232에 전원 및 시리얼 장비를 연결하는 방법을 설명합니다.

시작하려면 아래의 장치들이 필요합니다.

- DIN 레일 거치형 키트(패키지에 포함)
- 설정 및 디바이스 서버 연결용 CAT5 케이블(패키지에 포함)
- 설정 및 디바이스 서버 연결용 RJ45 to DB9 Female 커넥터(패키지에 포함)

2.1 패널 배치

Rhio232는 시스템 상태 표시를 위한 LED 표시등이 있습니다. 좌측의 표시등은 POWER, Serial Rx 및 Serial Tx 인데, 이 중 POWER는 시스템 전원 인가 상태를 나타냅니다. 다음 두개의 Serial RX/TX 표시등은 Rhio232의 시리얼 통신상태를 나타냅니다. 표 2-1은 각 LED 표시등의 기능을 설명합니다. 우측의 I/O 커넥터 주변에는 I/O 상태를 나타내기 위한 표시등들이 있는데, 상부의 표시등들은 Digital Output 점들의 상태를 나타내며, 하부의 표시등들은 Analog/Digital Input 점들의 상태를 나타냅니다.

표 2-1. LED 표시등

| 표시등 | | 기능 |
|----------------|--------------|--|
| Serial | Rx | 시리얼포트를 통해 명령을 받을 때 녹색으로 깜박거린다. |
| | Tx | 시리얼포트를 통해 응답을 보낼 때 녹색으로 깜박거린다. |
| Status | Power | 전원이 공급된 경우 적색으로 점등 된다. |
| Digital Input | DI 1 ~ DI 12 | Input 접점이 ON되면 녹색으로 점등 된다. |
| Digital Output | DO 1 ~ DO 10 | Output 접점이 ON되면 녹색으로 점등 된다. |
| Analog Input | AI 1 ~ AI 4 | Level Input모드일 경우에는 Level값 512를 기준으로 512보다 크면 녹색으로 점등 된다. Switch Input모드일 경우에는 사용자가 지정한 Threshold 값보다 크면 녹색으로 점등 된다. |

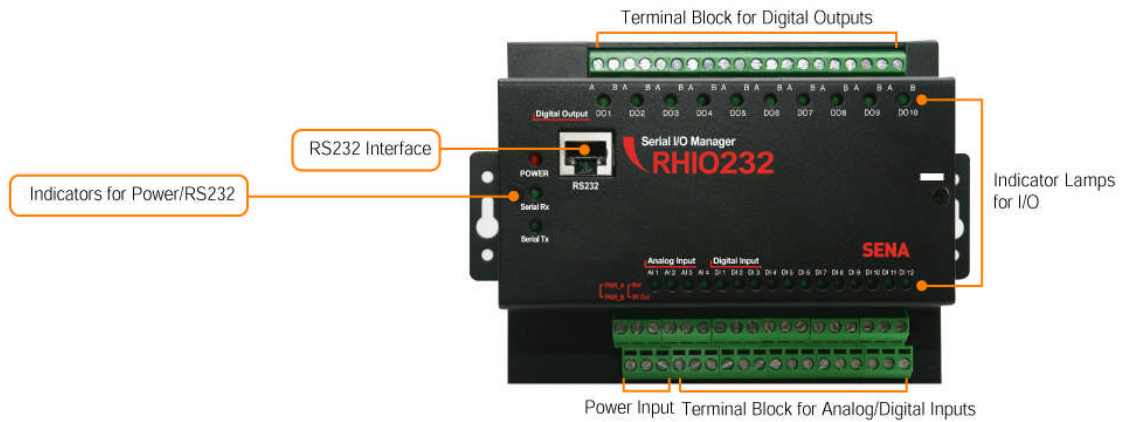


그림 2-1. Rhio232의 패널 배치

표 2-2. Rhio232의 터미널 블록 단자 배치

| 입력 블록 단자 배치 | Terminal Block Pinout | | | | | | | | | | | | | | | | | | | |
|----------------|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|----|
| | | | AI1 | AI2 | AI3 | AI4 | DI1 | DI2 | DI3 | DI4 | DI5 | DI6 | DI7 | DI8 | DI9 | DI10 | DI11 | DI12 | | |
| | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 31 | 33 | 35 | | |
| | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 | 26 | 28 | 30 | 32 | 34 | 36 | | |
| | <ul style="list-style-type: none"> • 전원 입력 (PWR_A , PWR_B): (1,2) - 극성 상관 없음. • 아날로그 입력 참조 전압 (Ref) : (3) <ul style="list-style-type: none"> - Analog Input 데이터의 Reference 전압을 지정합니다. Rhio는 0V부터 Reference 전압 까지를 1024 단계로 표시합니다. Reference 전압은 절대 5V 이상 지정해서는 안 됩니다. 자세한 내용은 4.4 ADC Input Port 설정을 참조하세요. • 5V Out : (4) <ul style="list-style-type: none"> - AVCC 출력 단자로, 사용자의 장비에 전원을 제공할 수 있습니다. 전원 그라운드는 5,7,9,11번 단자 중 하나를 사용하면 됩니다. • 아날로그 입력 (AI1 ~ AI4) : (5,6),(7,8),(9,10),(11,12) • 디지털 입력 (DI1 ~ DI12) : (13,14),(15,16),(17,18),(19,20),(21,22),(23,24), (25,26),(27,28),(29,30),(31,32),(33,34),(35,36) <p>참고 : 1. 각각의 입력은 위/아래 각각 한 쌍의 단자들로 구성됩니다. 2. 3,4번 단자를 제외한 나머지 입력은 극성에 상관 없이 연결이 가능합니다.</p> | | | | | | | | | | | | | | | | | | | |
| 출력 블록 단자 배치 | DO1 | | DO2 | | DO3 | | DO4 | | DO5 | | DO6 | | DO7 | | DO8 | | DO9 | | DO10 | |
| | A | B | A | B | A | B | A | B | A | B | A | B | A | B | A | B | A | B | A | B |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

다음과 같이 연달아 배치된 한 쌍의 단자들이 각각의 출력을 구성합니다.

- 디지털 출력 (DO1 ~ DO10) : (1,2), (3,4), (5,6), (7,8), (9,10),
(11,12), (13,14), (15,16), (17,18), (19,20)

2.2 하드웨어 연결

2.2.1 설치 및 부착

패키지에 포함되어 있는 DIN 레일 거취형 키트를 Rhio232에 부착하여 사용자는 원하는 위치에 Rhio232를 용이하게 설치할 수 있습니다.

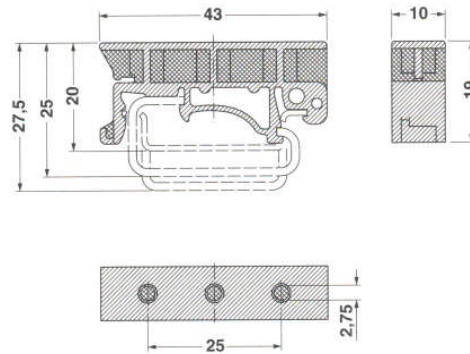


그림 2-2. DIN 레일 거취형 키트 치수 및 적용되는 DIN 레일

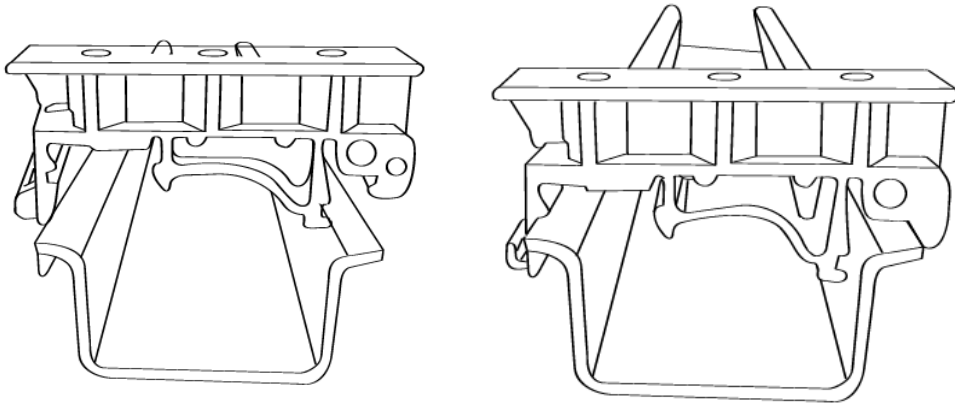


그림 2-3. DIN 레일에 DIN 레일 거취형 키트를 부착하는 방법

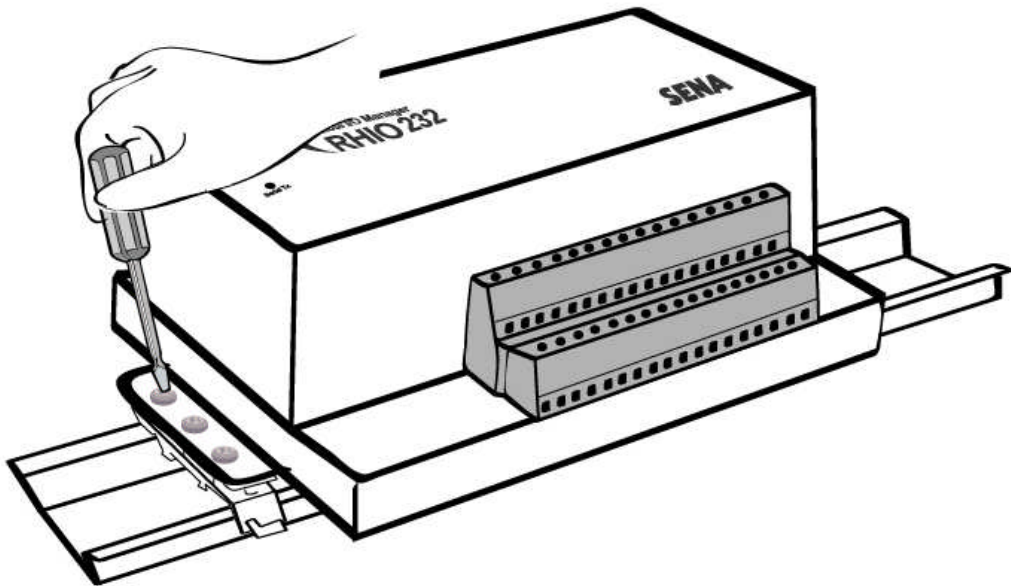


그림 2-4. DIN 레일에 Rhio232 설치 및 부착

2.2.2 전원 연결

사용자가 9V~48VDC, MAX. 5W의 제품 전원 사양에 맞추어 적합한 전원을 직접 Rhio232 전원 단자에 연결합니다. 전원이 제대로 공급된 경우, Rhio232의 **[Power]** 표시등이 적색 점등 상태를 유지합니다. 전원을 공급하는 선은 3M(미터)를 넘지 않도록 합니다.

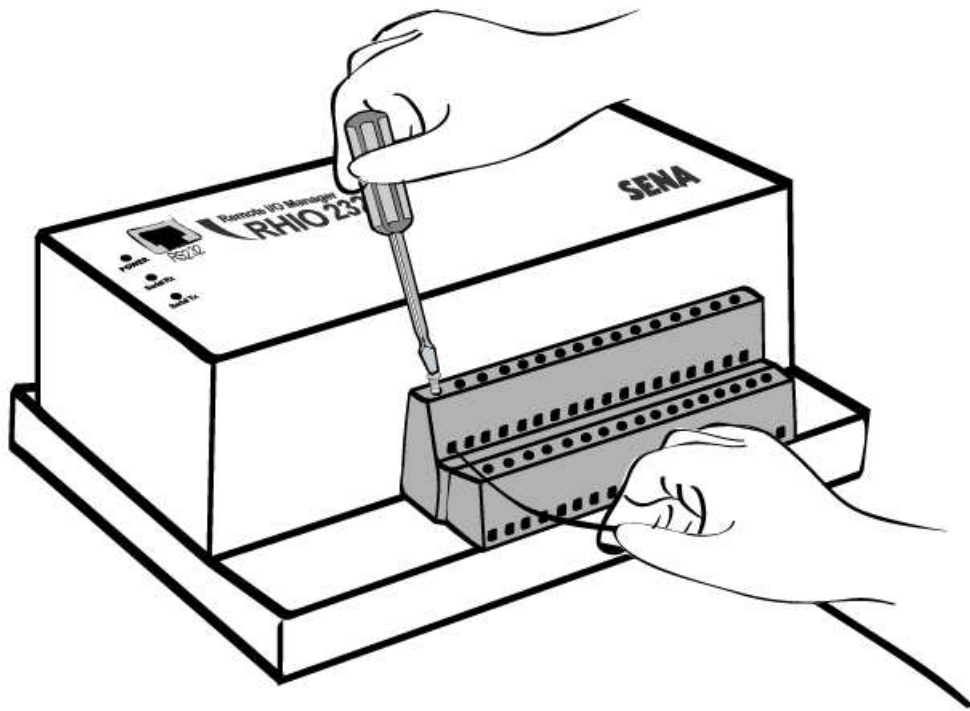


그림 2-5. Rhio232에 전원 연결

2.2.3 시리얼 장비에 연결

CAT5 케이블 한쪽 끝을 Rhio232의 RS232 Port에 연결하고 반대쪽 끝을 호스트 또는 디바이스 서버의 시리얼 포트에 연결합니다. CAT5 케이블의 커넥터가 RJ45이므로 호스트 또는 디바이스 서버의 시리얼 포트 타입이 DB9인 경우에는 패키지에 포함되어 있는 RJ45 to DB9 Female 스트레이트 케이블 어댑터를 끼워서 사용합니다.

케이블이 올바르게 연결된 경우, Rhio232와 연결된 장비간의 연결표시는 다음과 같이 나타납니다.

- Rhio232의 [Serial Rx], [Serial Tx] 표시등이 계속해서 녹색으로 점등 점멸을 반복하면 RS232 시리얼 통신을 수행하고 있음을 나타냅니다. RS232 통신을 하지 않을 경우 표시등은 점멸 상태가 됩니다.

위와 같이 나타나지 않는다면 Rhio232의 RS232 시리얼 통신 연결이 올바르지 않은 것입니다.

주의: 연결되는 호스트 또는 디바이스 서버의 시리얼 통신 파라미터를 Rhio232와 동일하게 설정해야 합니다. Rhio232의 시리얼 파라미터는 다음과 같습니다.

9600 Baud rate, Data bits 8, Parity None, Stop bits 1, No flow control

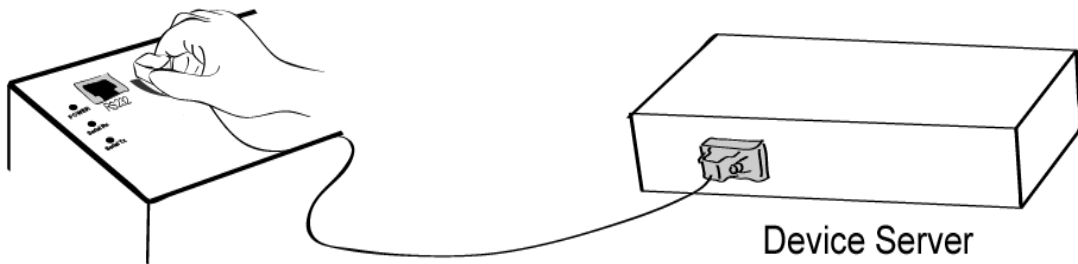


그림 2-6. Rhio232와 호스트 또는 디바이스서버와의 시리얼 케이블 연결

2.3 Rhio Manager 설치

Rhio Manager 는 Windows O/S 기반에서 동작하는 사용자 유틸리티 프로그램으로서 사용자는 이 프로그램을 사용하여 Rhio232 의 I/O Port에 대한 환경 설정, 제어 및 감시가 가능합니다.

2.3.1 Rhio Manager 설치

제품 패키지의 CD-ROM의 Rhio 시리즈 관련 폴더 또는 (주)세나테크놀로지 홈페이지 다운로드 센터 (www.sena.com/support/downloads)에서 Rhio_manager_setup.exe 파일을 다운로드 받아 설치합니다.

프로그램이 설치가 되면 [시작]->[프로그램]->[SENA]->[RHIO Manager]에서 Rhio Manager를 실행합니다. 화면 구성은 아래와 같습니다.

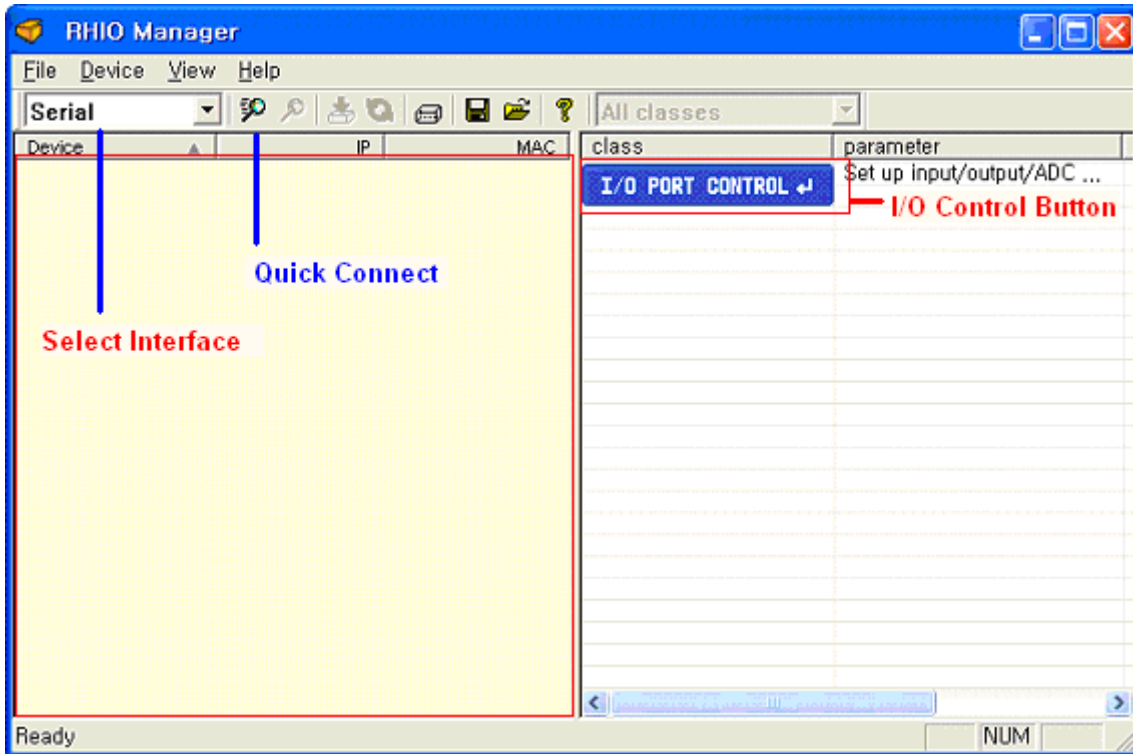


그림 2-7 Rhio Manager 실행 화면

2.3.2 Rhio Manager를 이용한 기본 설정

Select Interface

Rhio232 과의 Rhio Manager와의 통신 방법을 지정합니다. Rhio232을 사용할 때는, “Serial” 모드로 설정하여야 합니다.

Quick Connect

[Quick Connect] 명령을 수행하여 Rhio232와 연결된 RS232포트를 선택합니다.



그림 2-8 Quick connect 실행 화면

I/O PORT CONTROL

사용자는 [I/O PORT CONTROL] 버튼을 눌러 I/O 상태의 감시, 제어 및 설정이 가능합니다.

2.4 공장 출하 시 기본값 복구

사용자는 Rhio232 의 Factory reset 스위치를 눌러 공장 출하 시의 기본값으로 설정을 복구할 수 있습니다. Factory Reset 스위치는 Rhio232 제품 좌측 면에 위치해 있으며 핀셋 등 뾰족한 기구를 이용하여 약 1초 정도 누르고 있으면 작동 후 설정치가 초기화되며 Rhio232가 Reboot 됩니다.

공장 출하 시 기본값은 아래와 같습니다.

| | |
|-------------------------|------------|
| I/O Port 상태: | Enable |
| Output Port 정전 후 복구 기능: | Enable |
| ADC 동작 모드 : | Level Mode |
| ADC Threshold 값 : | 512 |
| Output Port 동작 조건 : | 없음 |
| 실행/정지 상태 : | 실행 |

3. I/O 설정 및 응용

3.1 I/O 감시 및 제어

사용자는 [I/O PORT CONTROL] 버튼을 눌러 I/O 상태 감시, 제어 및 설정이 가능합니다.

[I/O PORT CONTROL] 이 실행되면 Rhio Manager는 RS232 시리얼 통신을 통해 Rhio232의 시리얼 포트에 접속하여 I/O 상태의 감시를 시작합니다.

3.1.1 LED

ON : 적색 Icon

OFF : 청색 Icon

Disable : 회색 Icon

조건 ON/OFF : 녹색 Icon

Macro: M

Delay ON:적색 D

Delay OFF:청색 D

Pulse: P

Level Mode ADC Port : 녹색

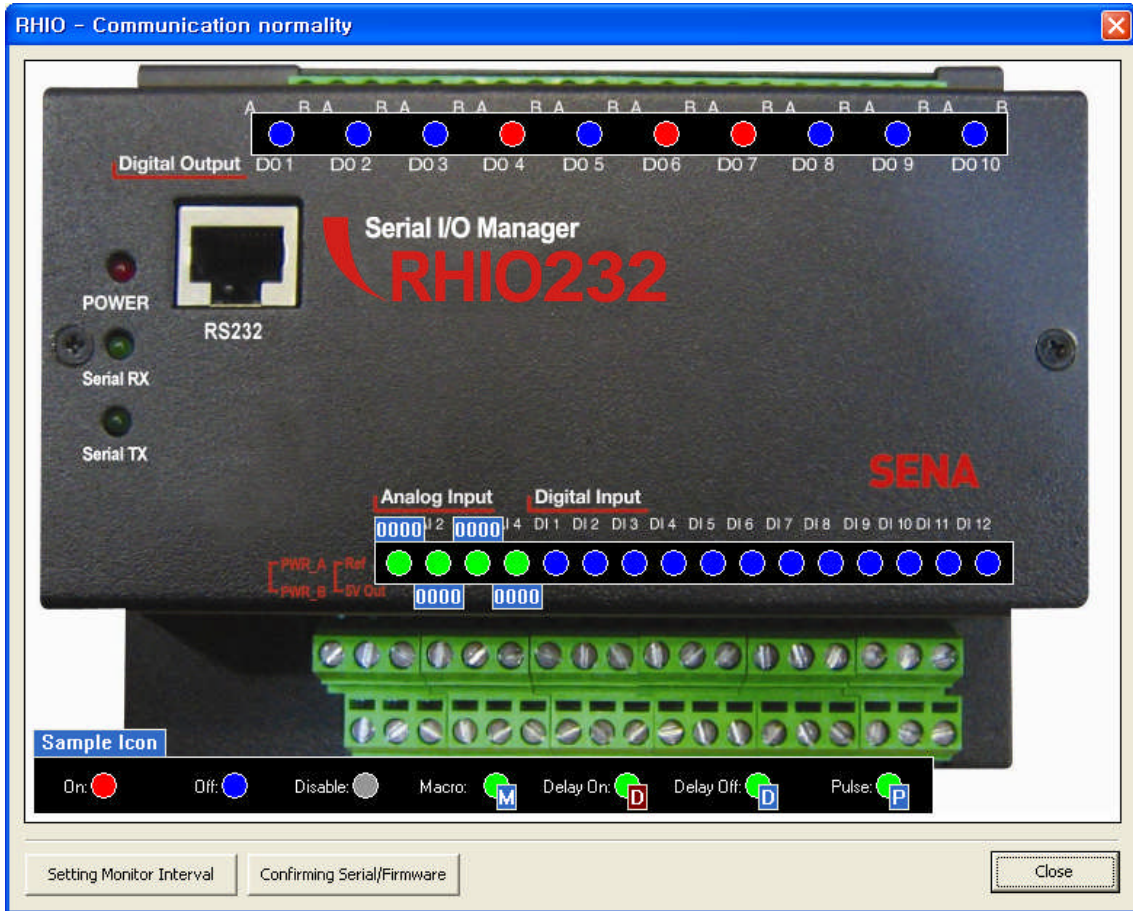


그림 3-1 I/O port Control 화면

3.1.2 감시 주기 설정

사용자는 [Setting Monitor Interval] 을 설정하여 Rhio232의 I/O 상태를 설정된 값의 시간 간격을 가지고 지속적으로 감시할 수 있습니다. Monitoring interval은 2-10초 사이의 값으로 지정합니다. Rhio Manager 는 지정된 주기에 따라 상태 요구 명령을 전송하고 Rhio232로부터 응답을 받아 화면에 표시합니다.

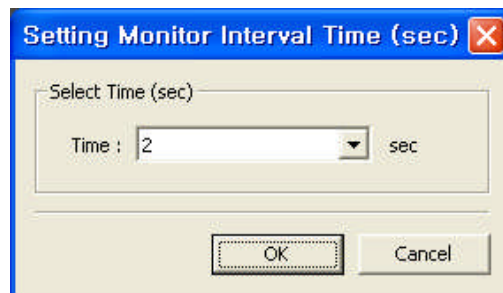


그림 3-2 Setting Monitoring Interval

3.1.3 I/O Port 감시 동작

Rhio232가 Rhio Manager로부터 상태 요구 명령을 수신하면 Digital Input, Analog Input 및 Digital Output Port의 전체 상태를 송신합니다.

Digital Input Port는 Input의 ON/OFF상태를 표시합니다

ADC Input Port는 Level 모드 설정 시에는 검출된 Analog값을 1024 단계의 Digital 값으로 표시("0000"~"1023")합니다.

ADC Input Port는 Switch 모드 시에는 입력 치와 설정된 Threshold 값을 비교하여 입력치가 설정된 값보다 높으면 ON, 설정된 값보다 낮으면 OFF로 표시합니다.

Digital Output Port는 ON/OFF 및 Output Port의 동작 조건, 실행 대기 상태를 표시합니다.

3.1.4 Digital Output Port 제어

사용자는 Rhio Manager의 Digital Output Port LED에 마우스를 포커스하여 왼쪽 버튼을 클릭함으로써 ON/OFF 등의 제어를 할 수 있습니다.

Digital Output Port 동작 조건이 설정되어 있지 않은 경우에는 마우스의 왼쪽 버튼을 한번 클릭할 때마다 ON/OFF 상태가 변화합니다.

동작 조건이 설정된 Port는 조건이 일치되면 ON 동작을 하며 OFF 상태에서 조건이 일치하지 않으면 실행 대기 상태로 표시됩니다.

Output 제어 동작이 완료되면 Rhio232는 전체 Port의 상태를 호스트 컴퓨터에 회신 합니다.

3.2 Digital Input 설정

Rhio232 에는 12개의 Digital Input 단자가 존재합니다. 사용자는 각각의 Digital Input Port를 Enable/Disable 할 수 있습니다. Digital Input Port에 대한 설정은 Rhio Manager를 이용하거나 4장의 I/O Port 관련 프로토콜에 규정된 명령을 전송함으로써 설정할 수 있습니다. Rhio Manager로 Digital Input 설정을 할 경우에는 I/O Port Control 화면의 Digital input LED에서 마우스 오른쪽 버튼을 클릭하면 설정 창이 나타나게 됩니다.

3.2.1 Enable/Disable 설정

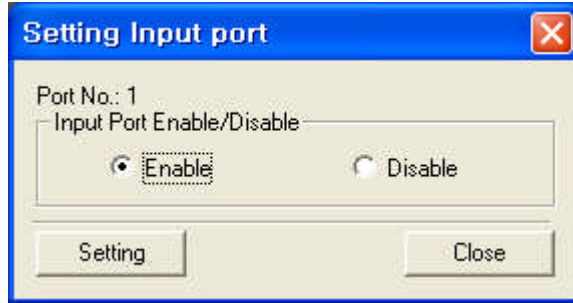


그림 3-3 Digital Input 설정

Digital Input의 설정은 Enable / Disable 설정이 있습니다. Enable / Disable 선택 후 [Setting] 버튼을 누르면 시스템에 적용됩니다. I/O Port가 동작하는 동작 모드 시에는 설정이 되지 않습니다. 따라서, Rhio Manager는 I/O Port 환경 설정 시에 먼저 설정 모드를 변경하는 명령을 전송한 후 이에 따라 동작이 정지하면 설정 명령을 전송하고 있습니다.

Enable 되어 있는 경우, Digital Input 단자에 연결된 하드웨어 장치로부터 ON/OFF 상태를 받아 표시합니다. Disable로 설정되어 있는 경우에는 Input 단자에 연결된 하드웨어 장치의 ON/OFF 상태와 상관없이 Disable로 표시됩니다.

3.3 Digital Output Port 설정

Rhio232는 10개의 Digital Output 단자가 장착되어 있습니다. Digital Output Port에 대한 설정은 Rhio Manager를 이용하거나 4장의 I/O Port 관련 프로토콜에 규정된 명령을 전송함으로써 설정할 수 있습니다. 사용자는 설정하기 원하는 Digital Output Port의 LED에 마우스를 포커스한 후 오른쪽 버튼을 클릭하면 Digital Output 설정 창이 나타납니다.

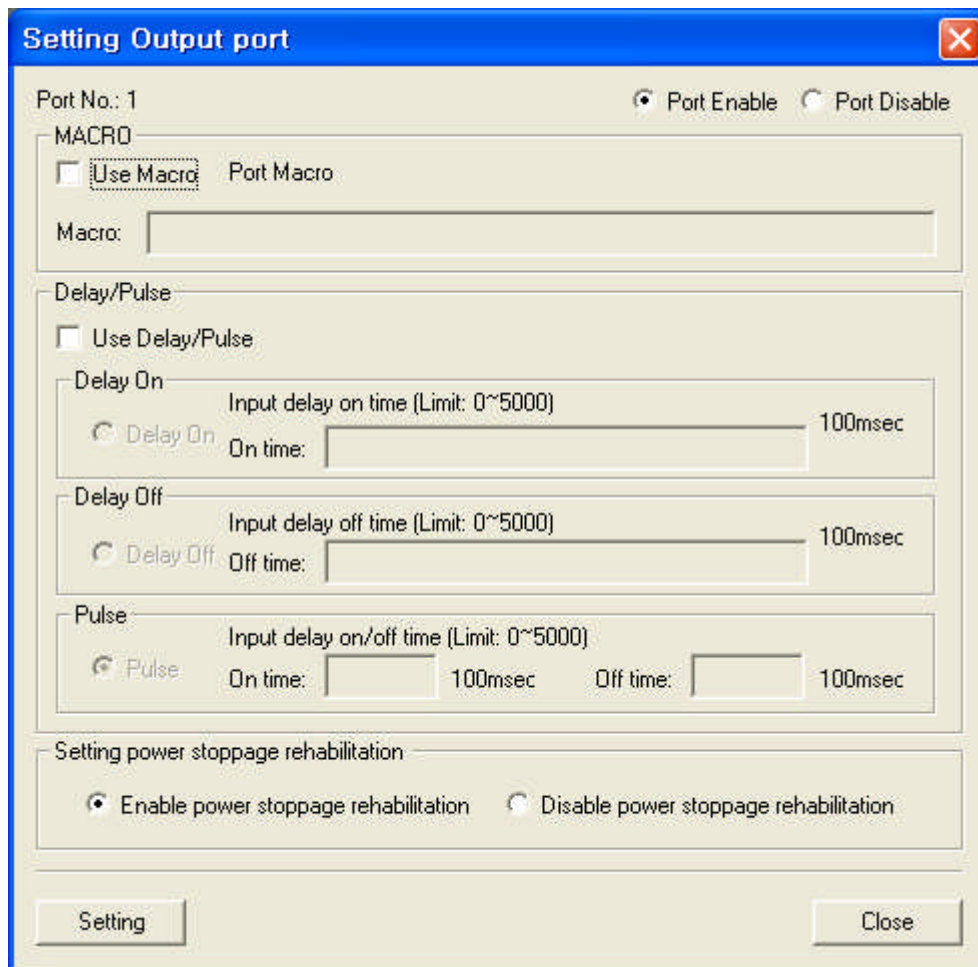


그림 3-4 Digital Output Port 설정창

3.3.1 Enable/Disable 설정

지정된 Port의 Enable/Disable 의 설정은 설정 창의 [Port Enable]과 [Port Disable] 체크박스를 선택하여 설정합니다. Disable 시에는 Digital Output Port의 상태는 OFF로 고정됩니다.

3.3.2 동작 조건 설정

MACRO 체크 박스를 선택한 경우, 사용자는 조건식을 입력하고 조건식에 따라 Output Port의 동작을 제어할 수 있습니다. Macro 박스에 원하는 동작 조건식을 입력합니다.

Output Port의 동작조건을 Port 별로 하나씩 설정합니다.

설정되지 않은 Output Port는 직접 제어하는 Port로서 설정됩니다.

나열된 식의 논리적 연산에 따른 최종값이 Output Port의 상태가 됩니다.

동작 조건식의 설정을 Clear 하면 직접 제어하는 Port가 됩니다.

동작 조건식의 설정방법

1) Port 번호 + 논리식(&,|) + Port 번호

2) 논리식(!) + Port 번호

3) Port 번호 +논리식(&,|) + 논리식(!) + Port 번호

(논리식 !은 해당되는 Port 번호의 앞에서만 유효합니다)

예) - Input 1번과 Input2번이 AND

I1&I2

- Input 1번과 Output2번이 OR

I1|O2

- Input 1번의 반전

!I1

- Output2번 AND Output 3번 AND Output 4번의 반전

O2&O3&!O4

동작 조건식의 설정 최대치는 Port 수를 기준으로 최대 21 Port까지 설정할 수 있습니다.

동작 조건식이 일치 하지 않는 상태에서 제어 명령을 통해 직접 ON 상태로 설정하면 상태는 동작 실행 대기 상태로 됩니다.

동작 조건이 일치하면 ON 상태가 되며, 일치 하지 않으면 Port는 OFF되고, 동작 실행 대기 상태가 됩니다.

ON 상태에서 명령을 통해 직접 OFF 상태로 설정하면 OFF 상태가 되며 이때에는 동작 조건이 일치하여도 동작되지 않습니다.

3.3.3 Delay 와 Pulse 동작

Use Delay/Pulse 체크박스에 체크한 경우 Output Port는 Delay 및 Pulse 동작을 합니다.
Delay 및 Pulse 동작의 설정 값은 100ms 단위로 입력해야 합니다.

Delay ON을 선택한 경우 사용자가 Output Port ON 명령을 보내면 설정된 Delay 시간 후 Output Port가 실제로 ON 상태로 됩니다.

Delay OFF를 선택한 경우 사용자가 Output Port OFF 명령을 보내면 설정된 Delay 시간 후 Output Port가 실제로 OFF 상태로 됩니다.

Pulse를 선택한 경우 사용자가 입력한 ON/OFF 시간에 따라 반복하여 ON/OFF 상태를 반복합니다.

3.3.4 정전 후 복구 설정

Output Port의 정전 후 복구 설정은 설정 창의 [Setting power stoppage rehabilitation] 프레임에서 설정합니다.

정전복구 Enable 시에는 Rhio232의 전원이 꺼졌다가 다시 켜진 경우 꺼지기 직전의 Output Port 상태를 유지합니다.

동작 조건이 설정된 Output Port가 정전복구 Enable되어 있으면 전원이 복구되었을 때 동작 조건이 일치하면 ON이 됩니다.

정전복구 Disable 시에는 Rhio232 의 전원이 꺼졌다가 다시 켜진 경우 Output Port의 상태는 OFF 됩니다.

3.4 ADC Input Port 설정

Rhio232에는 4개의 ADC Input 단자가 존재합니다. ADC Input Port을 설정하는 방법은 Rhio Manager를 이용한 방법과 I/O Port 프로토콜을 이용한 직접 설정 명령 전송 방법이 있습니다 (I/O Port 관련 프로토콜은 4장을 참조합니다).

Rhio Manager로 ADC Input Port 설정을 할 경우에는 [I/O Port Control] 화면의 ADC Port LED에 마우스를 포커스한 후, 오른쪽 마우스 클릭을 하면 ADC Input Port 설정 화면이 나타납니다.

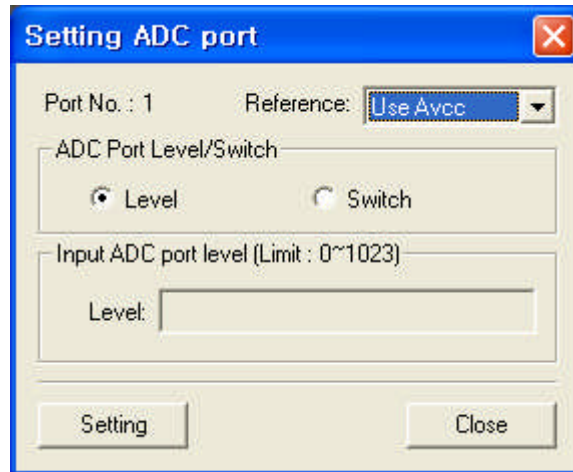


그림 3-5 ADC Input Port 설정

3.4.1 Reference 설정

Analog Input 데이터의 Reference 전압을 지정합니다. Rhio는 0V부터 Reference 전압까지를 1024 단계로 표시합니다. Reference 전압은 절대 5V 이상 지정해서는 안 됩니다. Reference 리스트 박스에서 사용할 reference 전압의 기준치를 지정합니다.

Use Avcc: Avcc(5V)를 reference 전압으로 지정합니다.

Use inside: Rhio 시스템 내부의 Reference 전압(2.56V)을 reference 전압으로 지정합니다.

Use Outside : Aref에 입력된 전압을 reference 전압으로 지정합니다.

3.4.2 ADC Input Port 동작 설정

Analog Input Port는 레벨 모드와 스위치 모드의 두 가지 모드가 있습니다.

레벨 모드 : 입력된 전압을 0V부터 Reference 전압까지를 “0000” ~ “1023”까지의 1024단계로 표시하도록 합니다.

스위치 모드 : 입력된 전압의 Level 값을 Threshold Level 설정 값을 기준으로 하여 Input Level이 Threshold보다 높거나 낮으면 상태가 변화 할 때 상태변화를 전송합니다. 스위치 모드 시 ADC Input은 Threshold값 기준으로 상위는 +8 이상, 하위는 -8 이하로 변화 하였을 때 상태의 변화로 인식합니다.

3.5 I/O Port 연결 방법

본 장에서는 Rhio232의 Digital Output, Digital Input 및 ADC Input Port에 사용자 회로를 구성할 경우의 연결방법에 대해 설명합니다. 모든 I/O Port에 연결할 케이블의 길이는 정확한 동작을 위하여 3M 이내의 길이로 연결하여 주어야 합니다.

3.5.1 Digital Output Port

모든 Digital Output Port에는 상태 확인용 LED가 부착되어 있으며 Relay 점접 ON시에 점등됩니다. 부하의 연결은 아래의 그림과 같이 OUTA1 과 OUTB1을 구동 스위치로 보고 접속하여 사용합니다

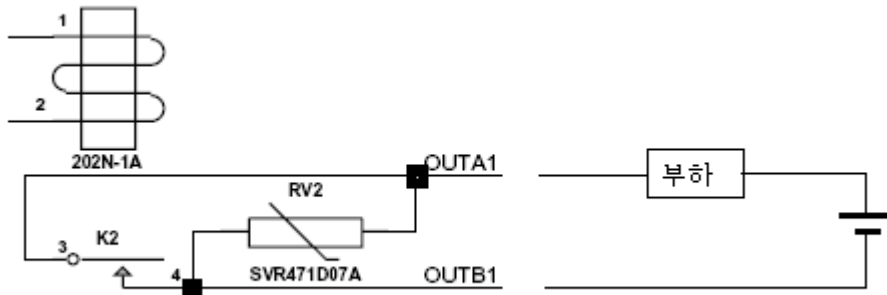


그림 3-6 Digital Output Port 연결

3.5.2 Digital Input Port

Digital Input 시스템은 전압의 \pm 극성에 상관 없이 동작하며 시스템의 내부 회로와는 절연되어 있습니다. 그림 3-7의 회로와 같이 연결되어 사용할 수 있으며 각각의 Input에 대하여 상태 확인용 LED가 있어 전압 Input 시에 점등 됩니다

주의)

규정된 전압(ON $\pm 3.2V$ 이상, OFF $\pm 1.3V$ 이하) 이외의 전압에서도 동작이 되는 경우가 있지만 안정된 동작을 하기 위하여 반드시 규정 전압을 사용하여 주십시오.

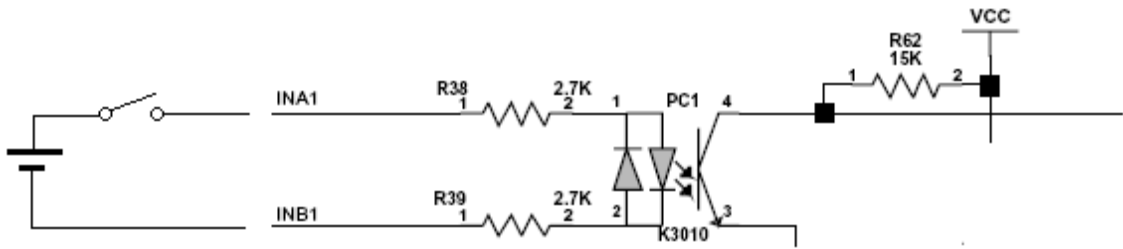


그림 3-7 Digital Input Port 연결

3.5.3 ADC Input Port

ADC Input Port는 비 절연 Input입니다. 가능하면 내부 회로에서 공급하는 AVCC(+5V) 전압을 사용하여 회로를 구성해야 합니다. 스위치 모드로 동작 시에는 Threshold 전압을 설정하고 Input 전압이 설정된 전압보다 높으면 Input ON으로 처리되며 ADC 상태표시 LED가 ON됩니다. 반대로 Input 전압이 Threshold 전압보다 낮으면 Input OFF로 처리하고 LED는 소등됩니다. AD 상태 표시 LED는 스위치 모드 시에만 동작하여 상태를 표시하게 됩니다.

1) Analog Input부 기준 전압(AREF) 연결 방법

AREF의 설정은 내부의 2.56V, 내부의 AVCC, 외부의 AREF단자의 3가지가 있습니다. 내부의 설정은 명령에 의하여 설정할 수 있으며 외부 Input의 설정은 아래와 같이 AVCC 전압을 R1, R2로 분압하여 사용할 수 있습니다. 분압용 저항 R1, R2는 1kΩ ~ 5kΩ 정도가 적당합니다.

주의) AREF 전압은 2V 이하로는 설정할 수 없습니다

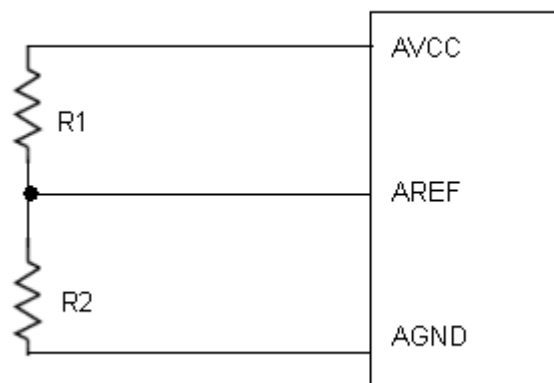


그림 3-8 Aref 의 연결방법

2) Analog Input Potentiometer 사용 시 방법

Analog Input은 입력되는 전압을 기준으로 동작하며 Input 임피던스는 100k Ω 입니다. 그림3-9과 같이 Potentiometer를 사용할 경우에는 1k Ω ~ 5k Ω 정도가 적당하며 외부 신호 소스를 사용할 때는 Input 임피던스를 낮게 사용하는 것이 각종 노이즈에 대하여 안정된 동작을 할 수 있습니다

주의) Analog Input 전압은 AREF 전압을 초과하지 않도록 하여 주십시오

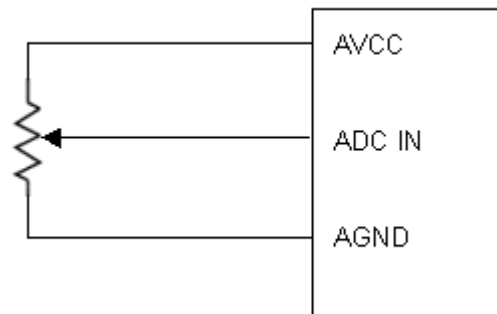


그림 3-9 Potentionmeter 사용시의 연결

3) AVCC 전압보다 높은 전압을 사용할 때의 연결방법

그림 3-10와 같이 분압하여 입력하고 Input 배선의 길이가 길거나 주변에 강한 노이즈가 있을 경우에는 내부에 Clamp Diode가 있으나 다른 ADC 채널에 영향이 없게 하기 위하여 외부에 Clamp Diode를 추가로 설치하는 것을 권장합니다.

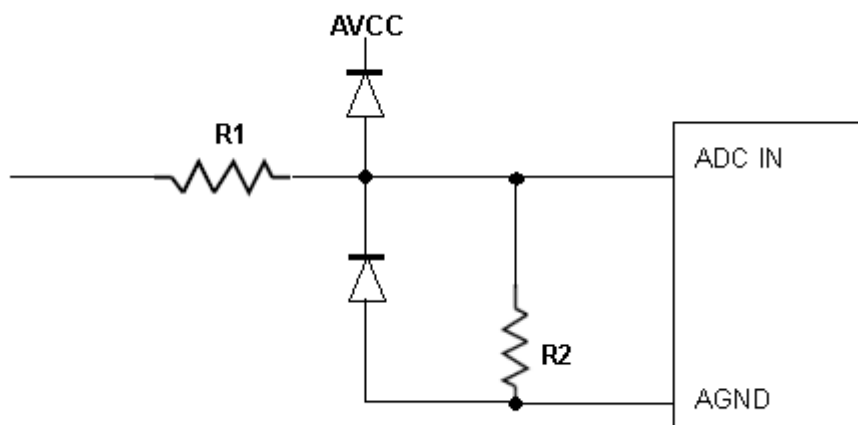


그림 3-10 Avcc 전압보다 높은 전압사용시의 연결

4. 소프트웨어 개발 및 응용

사용자는 Rhio 통신 프로토콜에 따라 사용자가 직접 응용 소프트웨어 작성할 수 있습니다. Rhio 프로토콜은 제어, 설정 명령 및 모니터링 명령에 대한 Rhio232의 응답으로 구성되어 있습니다.

사용자는 Rhio 라이브러리를 이용하여 응용 프로그램을 작성할 수 있습니다. Rhio 라이브러리는 사용자가 직접 Rhio 통신 프로토콜에 따라 작성하는 수고를 덜어주기 위해 각 명령 및 응답에 대한 처리를 함수로 만들어 쉽게 사용자 응용 프로그램을 작성할 수 있도록 도와줍니다.

- Rhio 라이브러리 파일

RHIO_Proc.dll, RHIO_Process.h

사용자가 Rhio 라이브러리를 이용하여 소프트웨어를 개발 하기 위해서는 두 파일을 반드시 링크하여야 합니다.

- 테스트 프로그램

Rhio 라이브러리를 이용해 개발한 샘플 프로그램으로서 테스트 프로그램이 제공되면, 사용자에게는 테스트 프로그램의 소스 및 설치 파일(RHIO_TEST_Setup.exe)이 제공됩니다. 테스트 프로그램은 개발자로 하여금 Rhio 라이브러리를 쉽게 적용할 수 있도록 적용 방법을 제시합니다.

4.1 Rhio 라이브러리

4.1.1 Rhio 라이브러리 개요

Rhio 라이브러리는 Rhio와 PC 간의 통신 프로토콜을 Windows 환경에서 쉽게 응용 프로그램으로 구현하기 위한 MFC 라이브러리입니다. Rhio 라이브러리는 CSocket Class를 내포하고 있으므로 사용자는 프로그램 개발시에 Microsoft Winsock component를 링크하여 사용해야 합니다. 또한 RHIO_CommProcessCreate 함수를 이용하여 Process Class를 생성해야만 라이브러리를 사용할 수 있습니다.

4.1.2 Reference

Enumeration, Structure, Function의 정의는 RHIO_Process.h를 참조 하십시오.

Enumeration (부록 C.1 참조)

| 타입명 | 설 명 |
|----------------|---|
| EOnOffFlag | ON, OFF 동작 조건 상태에 대한 flag |
| SendStatusFlag | Rhio로 보낸 명령에 대한 송신 상태 flag |
| ESetOutputFlag | Output Port Macro/Delay/Pulse 설정 시에 설정/제거에 관련된 flag |
| EADCMode | ADC가 Level 모드인지 Switch 모드인지를 설정하는 flag |

Structure (부록 C.2 참조)

| 구조체명 | 설 명 |
|------------------|------------------------|
| SADCData | 각각의 Port의 ON/OFF 상태 정보 |
| SOnOffStatusData | 전체 Port의 ON/OFF 상태 정보 |
| SSetOutput | Output Port 설정 정보 |
| SSetADC | ADC 설정 정보 |
| SSetInput | Input Port 설정 정보 |
| SRHIOSetting | 전체 Port 설정 정보 |

Function (부록 C.3 참조)

| 함 수 명 | 설 명 |
|----------------------------|---|
| RHIO_CommProcessCreate | Process Class를 생성하는 함수. 라이브러리 사용을 위해서는 반드시 생성해야 한다. |
| RHIO_SockConnect | RHIO와 소켓(TCP/IP)으로 접속한다. |
| RHIO_CommConnect | RHIO와 시리얼 Port로 접속한다. |
| RHIO_Close | RHIO와의 접속을 해제한다. |
| RHIO_SndCmd_SetOnOff | ON/OFF 제어 명령을 송신한다. |
| RHIO_SndCmd_GetOnOff | ON/OFF 확인 명령을 송신한다. |
| RHIO_SndCmd_SetSettingMode | 설정 모드 전환 명령을 송신한다. |
| RHIO_SndCmd_SetRunMode | 동작 모드 전환 명령을 송신한다. |
| RHIO_SndCmd_SetMACRO | Input Port의 매크로 설정 명령을 송신한다. |
| RHIO_SndCmd_GetMACRO | Input Port의 매크로 확인 명령을 송신한다. |
| RHIO_SndCmd_SetDelayPulse | Input Port의 Delay/Pulse 설정 명령을 송신한다. |
| RHIO_SndCmd_GetDelayPulse | Input Port의 Delay/Pulse 확인 명령을 송신한다. |

| | |
|------------------------------|---|
| RHIO_SndCmd_SetADC | 전체 ADC Port(1~4)의 Level 설정 명령을 송신한다. |
| RHIO_SndCmd_GetADC | 전체 ADC Port(1~4)의 Level 확인 명령을 송신한다. |
| RHIO_SndCmd_SetPortEnable | 전체 Port의 Enable/Disable 설정 명령을 송신한다. |
| RHIO_SndCmd_GetPortEnable | 전체 Port의 Enable/Disable 확인 명령을 송신한다. |
| RHIO_SndCmd_SetPwrStopEnable | 전체 Input Port의 정전 복구 Enable/Disable 설정 명령을 송신한다. |
| RHIO_SndCmd_GetPwrStopEnable | 전체 Input Port의 정전 복구 Enable/Disable 확인 명령을 송신한다. |
| RHIO_SndCmd_SetFactoryReset | Rhio Factory Reset 명령을 송신한다. |
| RHIO_SndCmd_SetSerial | Rhio 시리얼 번호 설정 명령을 송신한다. |
| RHIO_SndCmd_GetSerial | Rhio 시리얼 번호 확인 명령을 송신한다. |
| RHIO_SndCmd_GetFirmware | Rhio Firmware 버전 확인 명령을 송신한다. |
| RHIO_GetSettingData | 각각의 Port에 설정 확인 응답을 수신하는 이벤트가 발생하였을 경우 해당 데이터를 읽어 들인다. |
| RHIO_GetOnOffData | ON/OFF 상태 변화에 대한 응답 및 ON/OFF 제어에 대한 응답 수신 이벤트가 발생하였을 경우 해당 데이터를 읽어 들인다. |

4.2 Rhio 라이브러리를 이용한 샘플 프로그램 제작 및 데모

본 샘플 프로그램(RHIO_TEST)은 Rhio 라이브러리(RHIO_Proc.dll, RHIO_Process.h)를 링크하여 Microsoft Visual Studio .NET으로 제작되었으며, 다이얼로그 박스 기반의 어플리케이션입니다.

4.2.1 프로그램 UI 구성 및 관련 클래스

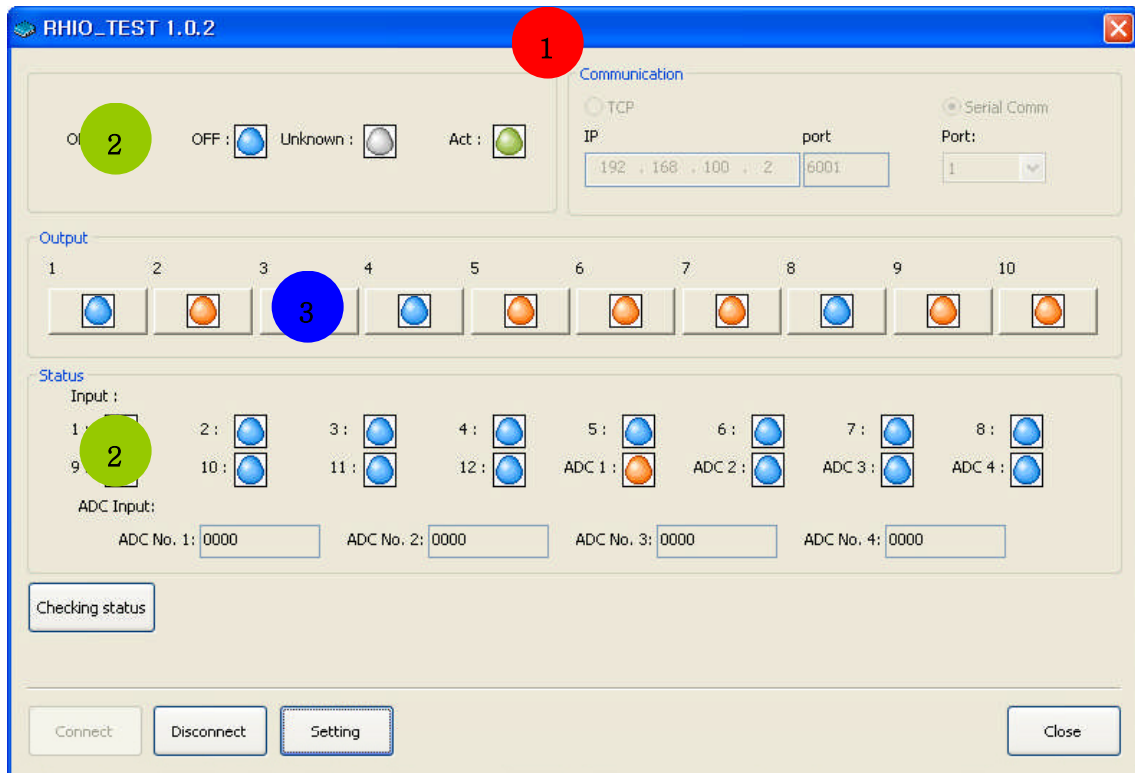


그림 4-1 샘플 프로그램 메인 윈도우

| 순 번 | 관련 클래스명 | 관련 파일명 |
|-----|-----------------|--------------------------------------|
| 1 | CWEB_IO_TESTDlg | WEB_IO_TESTDlg.h, WEB_IO_TESTDlg.cpp |
| 2 | CStatusWnd | StatusWnd.h, StatusWnd.cpp |
| 3 | COutputButton | OutputButton.h, OutputButton.cpp |

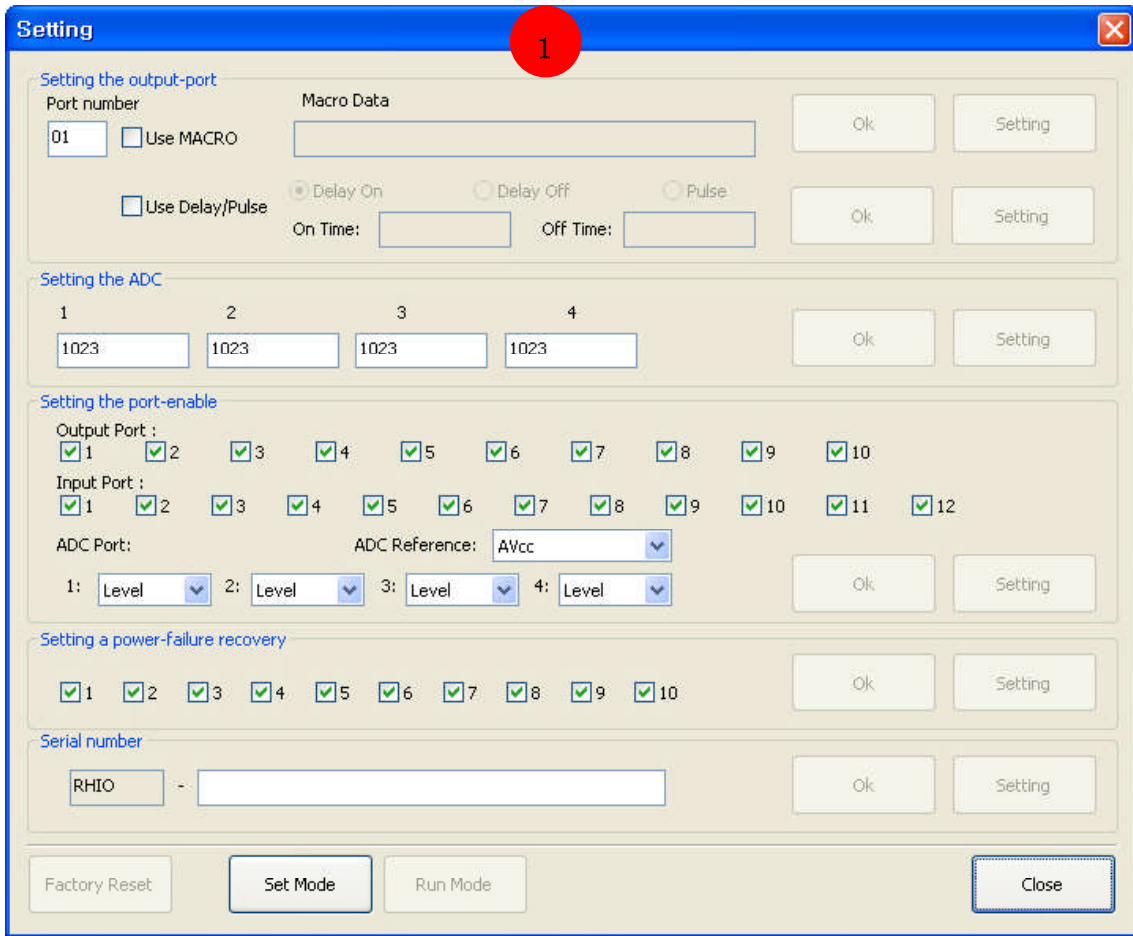


그림 4-2 샘플 프로그램 설정 윈도우

| 순번 | 관련 클래스명 | 관련 파일명 |
|----|-------------|-----------------------------|
| 1 | CSettingDlg | SettingDlg.h SettingDlg.cpp |

4.2.2 처리 절차

4.2.2.1 메인 윈도우 초기화

라이브러리를 링크하고, 라이브러리 함수의 주소를 얻어온다.

```
CWEB_IO_TESTDlg::RHIODllLoad()
```

메인 윈도우의 다이얼로그 아이টে를 초기화 한다.

```
CWEB_IO_TESTDlg::InitOutPutButton();
```

```
CWEB_IO_TESTDlg::InitInput(CPoint pntStart, int iWidth,  
int iHeight, int iTerm)
```

```
CWEB_IO_TESTDlg::InitSample(CPoint pntStart, int iWidth,  
int iHeight, int iTerm)  
CWEB_IO_TESTDlg::InitSelComm();
```

Process Class를 생성한다.

```
m_rhCreate(CWnd *pParentWnd)
```

4.2.2.2 이벤트 발생 시점별 처리 절차

프로그램 메인 윈도우에서의 이벤트 발생

접속 버튼 클릭

```
CWEB_IO_TESTDlg::OnBnClickedButtonConnect()
```

TCP 접속

```
m_rhSockConnect (BYTE bAddr1, BYTE bAddr2, BYTE bAddr3,  
BYTE bAddr4, int iPort)
```

Serial 접속

```
m_rhCommConnect (int iPort)
```

접속 끊기 버튼 클릭

```
CWEB_IO_TESTDlg::OnBnClickedButtonClose()
```

설정 버튼 클릭

```
CWEB_IO_TESTDlg::OnBnClickedButtonSetting()
```

상태확인 버튼 클릭

```
CWEB_IO_TESTDlg::OnBnClickedButtonStatusView()
```

Output Port 버튼 클릭

```
COutputButton::OnBnClicked()
```

프로그램 설정 윈도우에서의 이벤트 발생

Factory Reset 버튼 클릭

CSettingDlg::OnBnClickedButtonFactoryReset()

Set Mode 버튼 클릭

OnBnClickedButtonSetmode()

Run Mode 버튼 클릭

CSettingDlg::OnBnClickedButtonRunMode()

Serial 번호의 확인 버튼 클릭

OnBnClickedButtonMonitorSerial()

Serial 번호의 설정 버튼 클릭

CSettingDlg::OnBnClickedButtonSetSerial()

정전 복구 설정의 확인 버튼 클릭

CSettingDlg::OnBnClickedButtonMonitorPwrStop()

정전 복구 설정의 설정 버튼 클릭

CSettingDlg::OnBnClickedButtonSetPwrStop()

Port Enable 설정의 확인 버튼 클릭

CSettingDlg::OnBnClickedButtonMonitorEnable()

Port Enable 설정의 설정 버튼 클릭

CSettingDlg::OnBnClickedButtonSetEnable()

ADC Input 설정의 확인 버튼 클릭

CSettingDlg::OnBnClickedButtonMonitorInput()

ADC Input 설정의 설정 버튼 클릭

CSettingDlg::OnBnClickedButtonSetInput()

Use Delay/Pulse의 확인 버튼 클릭

CSettingDlg::OnBnClickedButtonMonitorOutput2()

Use Delay/Pulse의 설정 버튼 클릭

CSettingDlg::OnBnClickedButtonSetOutput2()

Use MACRO의 확인 버튼 클릭

CSettingDlg::OnBnClickedButtonMonitorOutput()

Use MACRO의 설정 버튼 클릭

CSettingDlg::OnBnClickedButtonSetOutput()

Rhio Device에서의 이벤트 발생

CWEB_IO_TESTDlg::OnUpdateStatus(WPARAM wParam,
LPARAM lParam)

Port ON/OFF 이벤트 발생

m_rhGetOnOffData (SOnOffStatusData &sOnOffData)

Port ON/OFF 이외의 이벤트 발생

CSettingDlg::OnReceive(WPARAM wParam, LPARAM lParam)

4.3 Rhio 통신 프로토콜

이 장에서는 사용자가 직접 Rhio 통신 프로그램을 작성할 경우 사용자가 참고할 Rhio232의 통신 프로토콜에 대해 설명합니다.

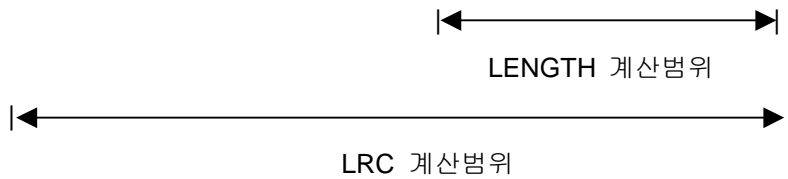
4.3.1 개요

사용자는 아래 프로토콜을 사용하여 RHIO를 설정, 제어, 모니터링할 수 있습니다.

4.3.1.1 전송 BLOCK의 구성

Rhio 통신 프로토콜의 아래와 같은 형태로 구성됩니다.

| | START FLAG | LENGTH | FUNCTION | DATA | LRC (BCC) | END FLAG |
|-----------|-----------------------------------|------------------------|---------------------|------|-------------------------------|------------------------|
| Byte Size | 1 | 2 | 2 | N | 2 | 2 |
| | 전송 BLOCK의 시작을 표시한다. 0x3A (":") | FUNCTION에서 LRC 앞까지의 길이 | Command Response | Data | START FLAG이후에서 LRC 앞까지 XOR한 값 | CR (0x0D) LF (0x0A) |



- START FLAG

전송 BLOCK의 시작을 의미합니다.

0x3A (":")

- LENGTH

FUNCTION+DATA Field의 길이를 나타냅니다.

- FUNCTION

제어, 설정, 확인, 상태 Command 와 Response Code를 표시합니다.

- DATA

제어,설정,확인,상태 Data 입니다.

- LRC (BCC)

전송 BLOCK의 Error를 검사합니다.

START FLAG에서 LRC이전까지의 data를 1 Byte단위로 XOR한 값입니다.

- END FLAG

CR+LF (0x0D+0x0A)

- LENGTH, LRC의 Data는 각 Byte를 상위 4bit nibble과 하위 4bit nibble로

나누어 각 nibble이 1 Byte로 표현됩니다.

각 Byte의 배열순서는 상위,하위 순서로 배열합니다.

4bit data의 1 Byte변환은 0x0 ~ 0x9는 0x30 ~ 0x39로 되고,

0xA ~ 0xF는 0x41 ~ 0x46으로 변환합니다.

변환된 data를 수신하면 수신측은 다시 역으로 변환합니다.

- Frame의 수신 개시 후 LF가 수신될 때까지의 Timeout은 5초입니다.

- Rhio는 수신 Command이상(BCC Error, Time Out) 발생 시 수신된 Command를 무시하며 다음 수신을 대기합니다.

(HOST에서는 송신 후, Rhio232에서 응답이 없으면 최소 5초 동안 Response를 대기하고, 다음 Command를 전송해야 합니다.)

4.3.1.2 NAK Response

- NAK Response 조건

- 수신된 Data의 BCC Error

- HOST에서 송신한 Data가 수신에 시작되고 정해진 시간(1초)이내에 Frame이 종료 되지 않을 때(수신대기 Time out)

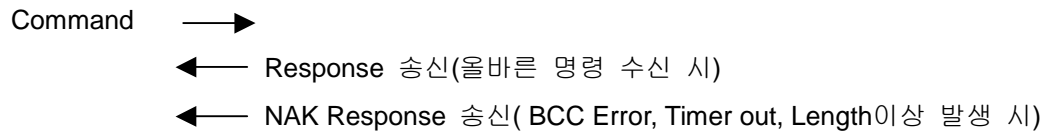
- 수신된 Data Frame의 길이가 Command의 Data 길이와 불일치 할 때
(Frame Length error)

- NAK Response의 구성

| | START FLAG | LENGTH | FUNCTION | DATA | LRC (BCC) | END FLAG |
|-----------|----------------------------------|------------------------|----------------------------|--------------------------------------|-------------------------------|------------------------|
| Byte Size | 1 | 2 | 2 | 3 | 2 | 2 |
| | 전송BLOCK의 시작을 표시한다. 0x3A (“:”) | FUNCTION에서 LRC 앞까지의 길이 | Response “00” 0x30,0x30 | NAK Data “NAK” 0x4E,0x41, 0x4B | START FLAG이후에서 LRC 앞까지 XOR한 값 | CR (0x0D) LF (0x0A) |

HOST

RHIO



- 사용자 응용프로그램에서는 NAK Response를 수신하면 Command를 재송신하거나 NAK 가 표시되도록 프로그램을 작성해야 합니다.

* 참고 : 프로토콜에 대한 아래 설명에서 각 Port의 표시

Input Port의 표시 : I1 ~ I12

ADC Input의 표시 : A1 ~ A4 (Level Input, Switch Input 모드)

Output Port의 표시 : O1 ~ O10

4.3.2 ON/OFF 제어

4.3.2.1 ON/OFF 제어 Command

| | | | |
|-----------|------------------------------------|---|-----------------------------------|
| | Command | Data | 비 고 |
| Byte Size | 2 | | |
| | “01” (0x30, 0x31) ON/OFF제어 | Output 10 Point의 Mask와 ON/OFF제어 Data | ON/OFF할 Output Point를 일괄 전송한다. |

Data

| | | | |
|---------|-------------------------|------------------------------------|-------------------------|
| | MASK Data | Data 구분 표시 | ON/OFF Data |
| Data 순서 | 1 ~ 10 | 11 | 12~21 |
| Port 번호 | 1 ~ 10 | - | 1~10 |
| Data 내용 | 제어 : 0x31 비제어 : 0x30 | 0x2C (“;”) MASK와 ON/OFF 구분 | ON : 0x31 OFF : 0x30 |

MASK 및 ON/OFF Data는 각 Port별로 순서대로, 1 Byte당 1 Port를 지정합니다.

Data 순서별 Port 위치

| | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|-----|
| Data | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 순서 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| Port | O1 | O2 | O3 | O4 | O5 | O6 | O7 | O8 | O9 | O10 |

4.3.2.2 ON/OFF 제어 Response

| | | | |
|-----------|---|--|--------------------------------------|
| | Response | Data | 비 고 |
| Byte Size | 2 | 52 | |
| | “02” (0x30, 0x32) Input, Output 전체상태 | ADC Level 4 Input 12 Point Output 10 Point 의 ON/OFF 상태 Data | Input, Output Point의 상태를 일괄 전송한다. |

Data

| Field 구분 | Data 순서 | Port 구분 | Data 내용 |
|-----------------|---------|---------|---|
| 제어상태 | 1 | - | 0x30 : 제어 정상, 0x31 : 동작 Mode 아님, 0x39 : 제어 비정상 |
| ADC Input LEVEL | 2~6 | A1 | 0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode "0000"~"1023" : Level |
| | 7 | - | Field 구분 0x2C (",") |
| | 8~12 | A2 | 0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode "0000"~"1023" : Level |
| | 13 | - | Field 구분 0x2C (",") |
| | 14~18 | A3 | 0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode "0000"~"1023" : Level |
| | 19 | - | Field 구분 0x2C (",") |
| | 20~24 | A4 | 0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode "0000"~"1023" : Level |
| | 25 | - | Field 구분 0x2C (",") |
| Input Status | 26~29 | I1~I4 | 0x30 : OFF, 0x31 : ON |
| | 30 | - | Field 구분 0x2C (",") |
| | 31~34 | I5~I8 | 0x30 : OFF, 0x31 : ON |
| | 35 | - | Field 구분 0x2C (",") |
| | 36~39 | I9~I12 | 0x30 : OFF, 0x31 : ON |
| | 40 | - | Field 구분 0x2C (",") |
| Output Status | 41~44 | O1~O4 | 0x30 : OFF, 0x31 : ON, 0x32 : 조건실행대기, 0x33 :지연ON대기, 0x34 :지연OFF대기, 0x35:PULSE동작 |
| | 45 | - | Field 구분 0x2C (",") |
| | 46~49 | O5~O8 | 0x30 : OFF, 0x31 : ON, 0x32 : 조건실행대기, 0x33 :지연ON대기, 0x34 :지연OFF대기, 0x35:PULSE동작 |
| | 50 | - | Field 구분 0x2C (",") |

| | | | |
|--|-------|--------|--|
| | 51,52 | 09,010 | 0x30 : OFF, 0x31 : ON, 0x32 : 조건실행대기 0x33 : 지연ON대기, 0x34 : 지연OFF대기, 0x35:PULSE동작 |
|--|-------|--------|--|

4.3.3 Input/Output 상태

4.3.3.1 I/O 상태 요구 Command

| | Command | Data | 비 고 |
|-----------|--------------------------------|---------------------|-----|
| Byte Size | 2 | 1 | |
| | “03” (0x30,0x33) 상태 요구 | 0x30 : Dummy값 고정 | |

4.3.3.2 I/O 상태 요구 Response

ON/OFF 제어 Response와 동일합니다. Input Port, ADC(Switch Input Mode일 때)의 상태변화 시(Threshold 전압 기준)에도 전송합니다.(Input 상태변화는 15ms 이상 변화된 값이 유지되면 상태변화 한다.).

상태 Response 일 때, 제어상태 field는 동작 모드일 때 0x30으로 회신하고, 설정 모드일 때는 0x31로 회신 합니다.

4.3.4 설정/동작

4.3.4.1 설정/동작 Command

| | Command | Data | 비 고 |
|-----------|---------------------------------|------------------------|--------------|
| Byte Size | 2 | 1 | |
| | “04” (0x30, 0x34) 설정/동작 | 0x30 : 설정 0x31 : 동작 | 설정/동작 Mode전환 |

설정 Command가 수신되면 상태 Response를 1회 회신하고, 상태가 바뀌어도 동작 Command 수신 이전까지는 상태 회신을 하지 않으며, 설정 모드로 변환되어 각종 설정 Command 외의 명령을 무시합니다.

Output Port 설정, ADC Input Port 설정, Port Enable 설정 및 Port 정전 복구 설정 시 설정 Command를 송신하여 설정 모드로 만든 후 명령을 전송해야 합니다.(각 설정 값의 확인 Command는 설정/동작 모드를 구분하지 않습니다.)

사용자가 원하는 각 설정이 종료되면 동작 Command를 송신하여 동작 상태로 만들어야 합니다. 동작 Command가 수신되면 상태 Response를 1회 회신하고, 정지되었던 동작을 계속 실행합니다.

4.3.4.2 설정/동작 Response

4.3.2.2 ON/OFF 제어 Response와 동일함.

설정 상태에서는 Response 1회 회신 후 동작 Command가 수신될 때 까지는 회신하지 않습니다. 상태 Response의 제어 상태 field는 설정 모드로 되면 0x31, 동작 모드로 되면 0x30을 회신 합니다.

4.3.5 Output Port 설정

4.3.5.1 Output Port 설정 Command

| | Command | Data | | | 비 고 |
|-----------|---|-----------------------------|---|------------------------------------|---|
| Byte Size | 2 | 2 | 1 | N | |
| | “05” (0x30, 0x35) Output Port 설정 | Port번호 “01” ~ “10” | 0x30 :동작조건 Clear 0x31 :동작조건설정 0x32 :지연, Pulse 동작 Clear 0x33 :지연, Pulse 동작 설정 0x39 :설정 ALL Clear | 설정 Data (Max 106 Byte) | 개별 Output Port의 속성을 설정한다. (Clear시 설정 Data는 0x30 1 Byte) |

설정 모드 상태에서만 설정이 가능합니다.

각각의 Output Port는 동작 조건 설정 또는 Pulse, 지연동작을 중복 또는 단독으로 설정할 수 있으며, 중복 설정 시 동작 조건 설정이 우선적으로 실행됩니다. 각 모드 별로 Clear 하거나 전체 설정을 모두 Clear할 수 있습니다. 설정 Clear 시 Data는 0x30, 1 Byte 입니다.

동작 조건 설정

지정된 개별 Output Port의 동작 조건을 설정합니다.

(Output Port가 ON일 때만 동작하고. OFF로 하면 Port를 직접 OFF 시킵니다.)

Input, Output Port의 상태 + 조건 + Input, Output Port의 상태 (AND, OR)

조건 + Input, Output Port의 상태 (NOT)

AND => &, OR => |, NOT => !

조건식은 AND, OR, NOT으로 단일식만을 사용합니다.

eg. 1번 Input과 2번 Input의 AND 조건 : I1 & I2

3번 Output과 1번 Input의 OR 조건 : O3 | I1

10번 Input의 반전과 10번 Output의 AND 조건 : !I10 & O10

1번 Input의 반전 조건 : !I1

설정 Data의 1 Output Point 당 동작 조건 지정의 Point 개수는?

21 Point 이내로 합니다.

eg. I1&I2&I3|I4!|I5|I6&I7&I8&I9&I10!|I11|I12|O2|O3|O4|O5|O6|O7|O8|O9|O10

설정할 Output Port는 동작 조건에 포함하지 않아야 합니다.

eg. O1 설정 시 동작 조건에 O1을 포함하지 않아야 합니다.

지연, Pulse 동작 설정

Output을 정해진 시간에 ON/OFF 또는 ON/OFF를 반복 하도록 지정합니다.

(지연 ON, Pulse는 ON제어 Command 일 때부터 동작이 시작되면, OFF 제어를 하면 동작을 종료합니다. 지연 OFF는 OFF 제어 시 동작을 시작합니다.)

설정 Data

| | 설정 Data (Time) | |
|-----------|------------------|-----------------|
| | ON Time | OFF Time |
| Byte Size | 5 | 5 |
| 동작 설정 | "00000"~"50000" | "00000"~"50000" |

100ms 단위로 설정하며 최대 500sec 까지 가능합니다.

eg.

| ON Time | OFF Time | 비 고 |
|---------|----------|------------------|
| "00000" | "00000" | 지연,Pulse동작 하지 않음 |
| "00001" | "00000" | 100ms지연 후 ON |
| "00020" | "00000" | 2sec지연 후 ON |

| | | |
|--------|--------|----------------------------|
| "5000" | "0000" | 5000sec지연 후 ON |
| "0000" | "0001" | 100ms지연 후 OFF |
| "0000" | "0020" | 2sec지연 후 OFF |
| "0000" | "5000" | 5000sec지연 후 OFF |
| "0001" | "0001" | 100ms ON, 100ms OFF 반복 |
| "0020" | "0020" | 2sec ON, 2sec OFF 반복 |
| "5000" | "5000" | 5000sec ON, 5000sec OFF 반복 |
| "0010" | "0030" | 1sec ON, 3sec OFF 반복 |
| "0030" | "0150" | 30sec ON, 15sec OFF 반복 |

4.3.5.2 Output Port 설정 Response

| | Response | Data | | | | 비 고 |
|-----------|-------------------------------|------------|-----------------------------|--|------------|-------------------------------|
| Byte Size | 2 | 1 | 2 | 1 | N | |
| | "06" (0x30,0x36) 설정회신 | 설정 FLAG | Port번호 "01" ~ "10" | 0x30 : 동작조건 Clear 0x31 : 동작조건설정 0x32 : 지연, Pulse동작 Clear 0x33 : 지연, Pulse동작 설정 0x39 : 설정 ALL Clear | 설정 Data | Output Port의 설정을 회신 한다. |

설정 Flag

0x30 : 설정 OK

0x39 : 설정 NG

0x31 : 설정 모드 아님(동작 모드일 때 송신)

0x32 : 미설정 (미설정 상태에서 Clear시 0x32로 회신)

(미설정 상태일 때 설정 Data는 0x30 1Byte)

설정 Clear 시 설정 Data는 0x30, 1Byte 입니다.

4.3.5.3 Output Port 설정 확인 Command

| | Command | Data | | 비 고 |
|-----------|--------------------------------|------------------------|---|------------------------------|
| Byte Size | 2 | 2 | 1 | |
| | “07” (0x30,0x37) 설정 확인 | Port 번호 “01” ~ “10” | 0x31 : 동작조건 설정 확인 0x33 : 지연, Pulse 동작 설정 확인 | 개별 Output Port의 속성을 확인한다. |

4.3.5.4 Output Port 설정 확인 Response

4.3.5.2 항목의 Output Port 설정 Response와 동일합니다.

4.3.6 ADC Input Port 설정

4.3.6.1 ADC Input Port 설정 Command

| | Command | Data | 비 고 |
|-----------|---------------------------------|---|---------------------------------|
| Byte Size | 2 | 16 | |
| | “08” (0x30,0x38) ADC 설정 | Input ADC Port의 Threshold Level 값을 설정한다. (“0000”~”1023”) X 4 | 10bit ADC Level값으 로 설정한다. |

ADC Input 전체의 Threshold Level 값을 설정합니다. (Switch Input 모드로 동작합니다.)

Input의 변화량 측정 시, Threshold Level을 기준으로 하여 +8 ~ -8 이상의 변화일 때만 상태가 변한 것으로 하여, Input 전압이 Threshold Level 보다 8 이상이면 ON, 8 이하이면 OFF로 판단 합니다.

(Threshold Level 이 1015 이상 일 때는 1015~1023까지의 변화를 ON으로 하며, 8 이하로 설정 되었으면 8~0까지의 변화를 OFF로 판단합니다.)

| ADC Input번호 | 1 | 2 | 3 | 4 |
|-------------|---------------|---------------|---------------|---------------|
| 설정 값 | “0000”~”1023” | “0000”~”1023” | “0000”~”1023” | “0000”~”1023” |

4.3.6.2 ADC Input Port 설정 Response

| | Response | Data | | 비 고 |
|-----------|----------------------------------|------------|-----------------------|------------------------------------|
| Byte Size | 2 | 17 | | |
| | “09” (0x30,0x39) ADC설정회신 | 설정 FLAG | (“0000”~”1023”) X 4 | Input ADC Port의 Threshold Level |

설정 Flag

0x30 : 설정 OK

0x31 : 설정 모드 아님(동작 모드 일 때)

0x39 : 설정 NG

4.3.6.3 ADC Input Port 설정 확인 Command

| | Command | Data | 비 고 |
|-----------|----------------------------------|------------------|---|
| Byte Size | 2 | 1 | |
| | “10” (0x31,0x30) ADC설정확인 | 0x30 : Dummy값 고정 | Input ADC Port의 Threshold Level 값을 확인 |

4.3.6.4 ADC Input Port 설정 확인 Response

4.3.6.2 ADC Input Port 설정 Response와 동일.

설정 Flag는 항상 설정 OK(0x30)입니다.

공장 출하 시 Level Input 모드의 초기값은 모두 “0000” 으로 되어 있습니다.

4.3.7 Port Enable 설정

4.3.7.1 Port Enable 설정 Command

| | Command | Data | 비 고 |
|-----------|-----------------------------------|------------------------------|---------------------------------------|
| Byte Size | 2 | 34 | |
| | “11” (0x31,0x31) Enable설정 | Port의 Enable/Disable 설정 Data | 전체 Port의 Enable/Disable 설정을 한다. |

Data

| Field 구분 | Data순서 | Port구분 | Data내용 |
|---------------------|--------|--------|---|
| ADC Port | 1 | A1 | 0x31 : LevelInput Mode 0x32 : SwitchInput Mode |
| | 2 | A2 | |
| | 3 | A3 | |
| | 4 | A4 | |
| | 5 | - | Field구분 0x2C (“,”) |
| ADC Reference 설정 | 6 | - | 0x30 : AVcc (Vcc 5V) -> default 0x31 : 내부 (2.56V) 0x32 : 외부 (2V~4.5V) |
| | 7 | - | Field구분 0x2C (“,”) |
| Input Port | 8~11 | I1~I4 | 0x31 : Enable, 0x32 : Disable |
| | 12 | - | Field구분 0x2C (“,”) |
| | 13~16 | I5~I8 | 0x31 : Enable, 0x32 : Disable |
| | 17 | - | Field구분 0x2C (“,”) |
| | 18~21 | I9~I12 | 0x31 : Enable, 0x32 : Disable |
| | 22 | - | Field구분 0x2C (“,”) |
| Output Status | 23~26 | O1~O4 | 0x31 : Enable, 0x32 : Disable |
| | 27 | - | Field구분 0x2C (“,”) |
| | 28~31 | O5~O8 | 0x31 : Enable, 0x32 : Disable |
| | 32 | - | Field구분 0x2C (“,”) |
| | 33,34 | O9,O10 | 0x31 : Enable, 0x32 : Disable |

I/O Port는 최초상태에서 모두 Enable 상태로 됩니다.

ADC Port는 최초상태에서 모두 Level Input 모드입니다.

4.3.7.2 Port Enable 설정 Response

| | Response | Data | 비 고 |
|-----------|-------------------------------------|------------------------------------|--------------------------------------|
| Byte Size | 2 | 35 | |
| | “12” (0x31,0x32) Enable설정회신 | Port의 Enable/Disable 설정 Data | 전체 Port의 Enable/Disable 설정을 회신 |

Data

| Field 구분 | Data순서 | Port 구분 | Data내용 |
|------------------|--------|---------|---|
| 설정상태 | 1 | - | 0x30 : 설정 정상, 0x31 : 설정 Mode 아님 0x39 : 설정 비정상 |
| ADC Port | 2 | A1 | 0x30 : Avcc사용 (Vcc 5V) -> default 0x31 : 내부사용 (2.56V) 0x32 : 외부사용 (2V~4.5V) |
| | 3 | A2 | |
| | 4 | A3 | |
| | 5 | A4 | |
| | 6 | - | Field구분 0x2C (“,”) |
| ADC Reference 설정 | 7 | - | 0x30 : Avcc사용 (Vcc 5V) -> default 0x31 : 내부사용 (2.56V) 0x32 : 외부사용 (2V~4.5V) |
| Input Port | 9~12 | I1~I4 | 0x31 : Enable, 0x32 : Disable |
| | 13 | - | Field구분 0x2C (“,”) |
| | 14~17 | I5~I8 | 0x31 : Enable, 0x32 : Disable |
| | 18 | - | Field구분 0x2C (“,”) |
| | 19~22 | I9~I12 | 0x31 : Enable, 0x32 : Disable |
| | 23 | - | Field구분 0x2C (“,”) |
| Output Status | 24~27 | O1~O4 | 0x31 : Enable, 0x32 : Disable |
| | 28 | - | Field구분 0x2C (“,”) |
| | 29~32 | O5~O8 | 0x31 : Enable, 0x32 : Disable |
| | 33 | - | Field구분 0x2C (“,”) |
| | 34,35 | O9,O10 | 0x31 : Enable, 0x32 : Disable |

4.3.7.3 Port Enable 설정 확인 Command

| | Command | Data | 비 고 |
|-----------|--------------------------------------|------------------|--------------------------------------|
| Byte Size | 2 | 1 | |
| | “13” (0x31,0x33) Enable설정한확인 | 0x30 : Dummy값 고정 | 전체 Port의 Enable/Disable 설정을 확인 |

4.3.7.4 Port Enable 설정 확인 Response

4.3.7.2 Port Enable 설정 Response와 동일합니다.

회신 시 설정 상태 field는 정상(0x30)으로 회신 합니다.

공장 출하 시 초기값은 모두 Enable 상태 입니다. (ADC는 Level Input 모드)

4.3.8 Port 정전 후 복구 설정

4.3.8.1 Port 정전 후 복구 설정 Command

| | Command | Data | 비 고 |
|-----------|------------------------------------|--------------------------|------------------------------------|
| Byte Size | 2 | 12 | |
| | “14” (0x31,0x34) 정전 후 복구설정 | Port의 정전 후 복구 설정 Data | Output Port의 정전 후 복구 설정을 한다. |

Data

| Field 구분 | Data순서 | Port구분 | Data내용 |
|---------------|--------|--------|-------------------------------|
| Output Status | 1~4 | O1~O4 | 0x31 : Enable, 0x32 : Disable |
| | 5 | - | Field구분 0x2C (“;”) |
| | 6~9 | O5~O8 | 0x31 : Enable, 0x32 : Disable |
| | 10 | - | Field구분 0x2C (“;”) |
| | 11,12 | O9,O10 | 0x31 : Enable, 0x32 : Disable |

4.3.8.2 Port 정전 후 복구 설정 Response

| | Response | Data | 비 고 |
|-----------|--|--------------------------|-----------------------------------|
| Byte Size | 2 | 13 | |
| | “15” (0x31,0x35) 정전 후 복구설정 회신 | Port의 정전 후 복구 설정 Data | Output Port의 정전 후 복구 설정을 회신 |

Data

| Field 구분 | Data순서 | Port구분 | Data내용 |
|---------------|--------|--------|--|
| 설정상태 | 1 | - | 0x30 : 설정 정상, 0x31 : 설정 Mode 아님, 0x39 : 설정 비정상 |
| Output Status | 2~5 | O1~O4 | 0x31 : Enable, 0x32 : Disable |
| | 6 | - | Field구분 0x2C (“,”) |
| | 7~10 | O5~O8 | 0x31 : Enable, 0x32 : Disable |
| | 11 | - | Field구분 0x2C (“,”) |
| | 12,13 | O9,O10 | 0x31 : Enable, 0x32 : Disable |

4.3.8.3 Port 정전 후 복구 설정 확인 Command

| | Command | Data | 비 고 |
|-----------|-----------------------------------|------------------|----------------------------|
| Byte Size | 2 | 1 | |
| | “16” (0x31,0x36) 정전복구설정확인 | 0x30 : Dummy값 고정 | 전체 Port의 정전복구 설정을 확인 |

4.3.8.4 Port 정전 복구 설정 확인 Response

4.3.8.2 정전 복구 설정 Response와 동일합니다.
회신 시 설정 상태 field는 정상(0x30)으로 회신 합니다.
공장 출하 시 초기값은 모두 Enable 상태입니다.

5. 세나 디바이스 서버와 Rhio232 연결

Rhio232는 시리얼 통신을 통해서 각종 I/O 장비를 제어 및 감시할 수 있게 하는 제품입니다. Rhio232를 세나의 디바이스 서버 제품에 연결하여 함께 사용하면 각종 I/O 장치 및 시리얼 통신 장치들을 Ethernet 네트워크 상에서 관리할 수 있습니다.

세나 디바이스 서버 제품과 Rhio232를 함께 사용할 수 있는 전형적인 방법은 아래와 같습니다.

- Rhio Library를 이용한 **TCP 소켓 프로그램**을 통한 연결 : 세나 디바이스서버 제품 + Rhio232
- Rhio Library를 이용한 **시리얼 통신 프로그램 (COM port)**을 통한 연결 : Serial/IP COM Port Redirector + 세나 디바이스서버 제품 + Rhio232

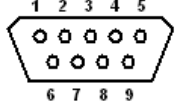

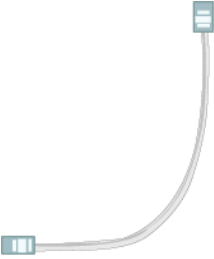
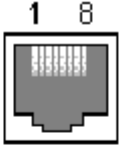
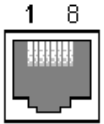
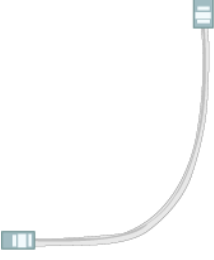
Serial/IP 소프트웨어에 대한 자세한 설명은 *tips_serial_ip_with_sena_ds-v1.0.0.pdf* 또는 Serial/IP User Manual을 참고하시기 바랍니다. Serial/IP에 관련된 문서들은 제품 패키지 CD 또는 <http://www.sena.com/korean/support/downloads/>에서 다운로드 받으실 수 있습니다.

본장에서는 사용자의 이해를 돕기 위해서 세나 디바이스 서버 제품과 Rhio232를 연결하는 방법 및 실제 응용 예를 설명합니다.

5.1 연결

Rhio232는 RS232 시리얼 통신 포트를 통해 세나의 디바이스 서버와 연결되며, 커넥터는 RJ45 타입입니다. 세나 디바이스 서버 제품 중에서 시리얼 통신 포트 형태가 RJ45 커넥터가 아닌 제품들(예:LS100, PS100 등)은 Rhio232 제품 패키지에 포함되어 있는 RJ45-DB9 Female 스트레이트 어댑터를 이용하여 연결할 수 있습니다. Super 시리즈 또는 STS 시리즈 제품들은 RJ45 형태의 시리얼 포트를 지원하므로, 직접 CAT5 스트레이트 케이블 연결을 통해 통신할 수 있습니다. 표 5-1 은 세나 디바이스 서버 제품들과 Rhio232의 유형을 요약하여 보여주고 있습니다.

표 5-1. 세나 디바이스 서버 커넥터 유형

| 세나 디바이스 서버 | 연결 | Rhio232 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------------|-------------|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|---|---|---|-----|-------------|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|
| <p>DB9:</p>  <table border="1" data-bbox="338 528 564 831"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>1</td><td>-</td></tr> <tr><td>2</td><td>Rx</td></tr> <tr><td>3</td><td>Tx</td></tr> <tr><td>4</td><td>DTR</td></tr> <tr><td>5</td><td>GND</td></tr> <tr><td>6</td><td>DSR</td></tr> <tr><td>7</td><td>RTS</td></tr> <tr><td>8</td><td>CTS</td></tr> <tr><td>9</td><td>-</td></tr> </tbody> </table> <p>예: LS100, LS100W, PS100/200/400/110/410/810, SS100</p> | Pin | Description | 1 | - | 2 | Rx | 3 | Tx | 4 | DTR | 5 | GND | 6 | DSR | 7 | RTS | 8 | CTS | 9 | - | <p>RJ45-DB9 Female 스트레이트 어댑터</p>  <p>+ CAT5 스트레이트 케이블</p>  | <p>RJ45:</p>  <table border="1" data-bbox="1002 633 1305 958"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>1</td><td>CTS</td></tr> <tr><td>2</td><td>DSR</td></tr> <tr><td>3</td><td>RxD</td></tr> <tr><td>4</td><td>GND</td></tr> <tr><td>5</td><td>DCD</td></tr> <tr><td>6</td><td>TxD</td></tr> <tr><td>7</td><td>DTR</td></tr> <tr><td>8</td><td>RTS</td></tr> </tbody> </table> | Pin | Description | 1 | CTS | 2 | DSR | 3 | RxD | 4 | GND | 5 | DCD | 6 | TxD | 7 | DTR | 8 | RTS |
| Pin | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Rx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Tx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | DTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | GND | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | DSR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | RTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | CTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pin | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | CTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | DSR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | RxD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | GND | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | DCD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | TxD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | DTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | RTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>RJ45:</p>  <table border="1" data-bbox="323 1270 576 1543"> <thead> <tr> <th>Pin</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>1</td><td>CTS</td></tr> <tr><td>2</td><td>DSR</td></tr> <tr><td>3</td><td>RxD</td></tr> <tr><td>4</td><td>GND</td></tr> <tr><td>5</td><td>DCD</td></tr> <tr><td>6</td><td>TxD</td></tr> <tr><td>7</td><td>DTR</td></tr> <tr><td>8</td><td>RTS</td></tr> </tbody> </table> <p>예: SS110/400/800 STS800/1600</p> | Pin | Description | 1 | CTS | 2 | DSR | 3 | RxD | 4 | GND | 5 | DCD | 6 | TxD | 7 | DTR | 8 | RTS | <p>CAT5 스트레이트 케이블로 연결</p>  | | | | | | | | | | | | | | | | | | | | | |
| Pin | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | CTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | DSR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | RxD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | GND | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | DCD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | TxD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | DTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | RTS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

5.2 세나 디바이스 서버와 Rhio232의 응용

본 절에서는 Serial/IP 소프트웨어를 이용한 가상 COM 포트 통신 및 8-포트 시리얼 디바이스 서버 SS800 제품을 통한 응용 예를 설명합니다.

작업 전에 구비하여야 하는 제품 및 소프트웨어 목록과 SS800 시리얼 포트 설정 및 Serial/IP COM Port Redirctor의 COM Port 설정을 아래와 같이 요약하였습니다.

Rhio232 및 SS800 제품

Serial/IP COM Port Redirector 및 Rhio Manager 소프트웨어

두개의 CAT5 스트레이트 케이블 (SS800 네트워크 연결용 및 Rhio232 시리얼 포트 연결용)

- SS800 설정

IP 주소(DHCP): 192.168.222.21

시리얼 포트 #1 설정:

Host mode: TCP

TCP Local port: 7001

Serial: RS232 type / 9600 Baud rate / 8 Data bit / None Parity / 1 Stop bit

- Serial/IP COM Port Redirector 소프트웨어 설정

COM Port #10 사용

단, 사용자의 PC에 Serial/IP COM Port Redirector 및 Rhio Manager 소프트웨어가 이미 설치되었고, Rhio232는 전원과 적절한 I/O 장치에 연결되어 있다고 가정합니다. 그림 5-1은 SS800과 Rhio232연결하여 I/O 장비를 원격으로 관리하는 모습을 보여주고 있습니다.

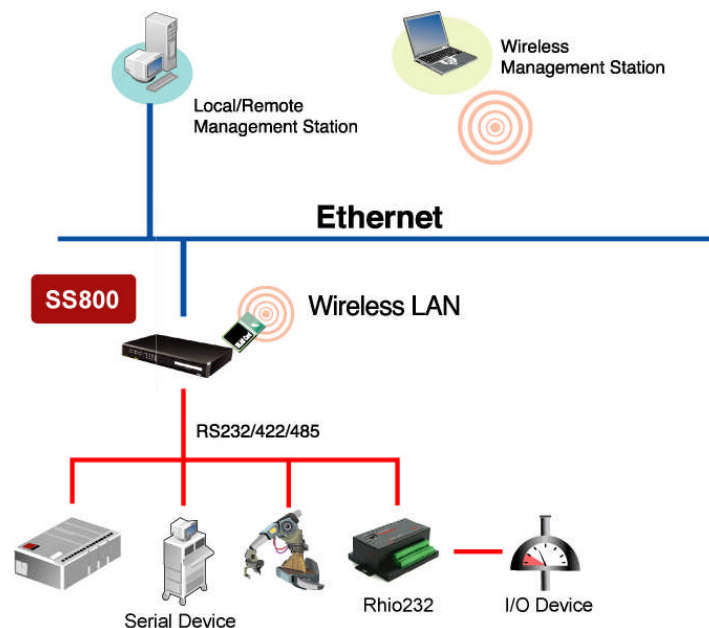
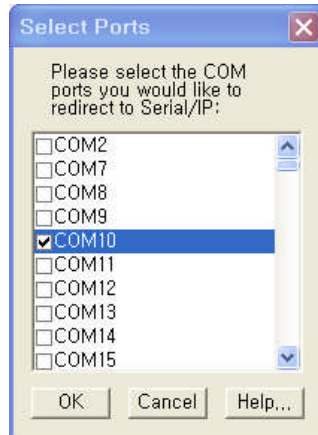


그림 5-1 Rhio232와 SS800의 연결도

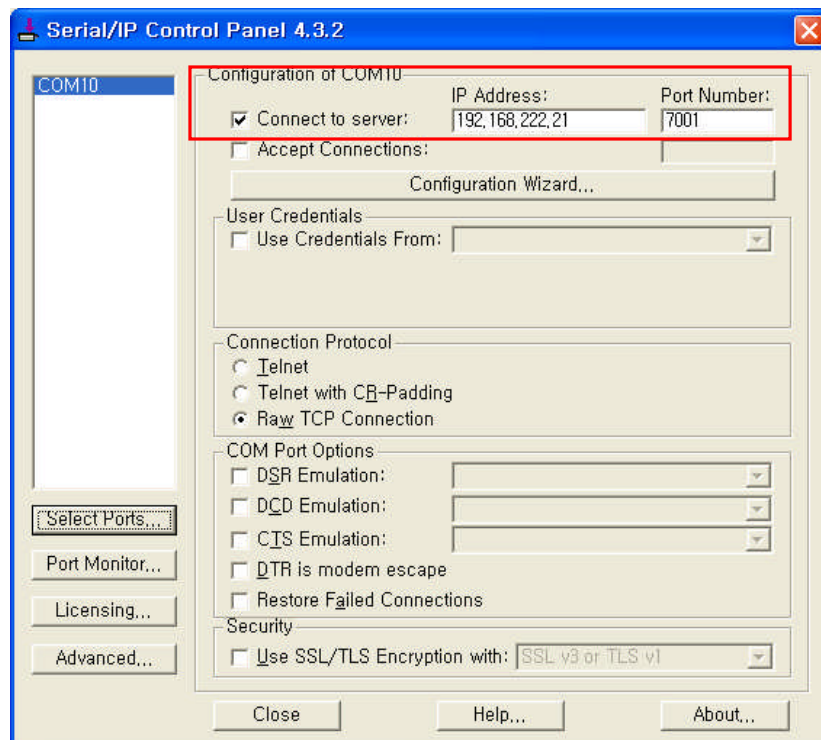
1 단계. SS800의 1번 시리얼 포트에 Rhio232를 CAT5 케이블로 연결합니다.

2 단계. Serial/IP 설정

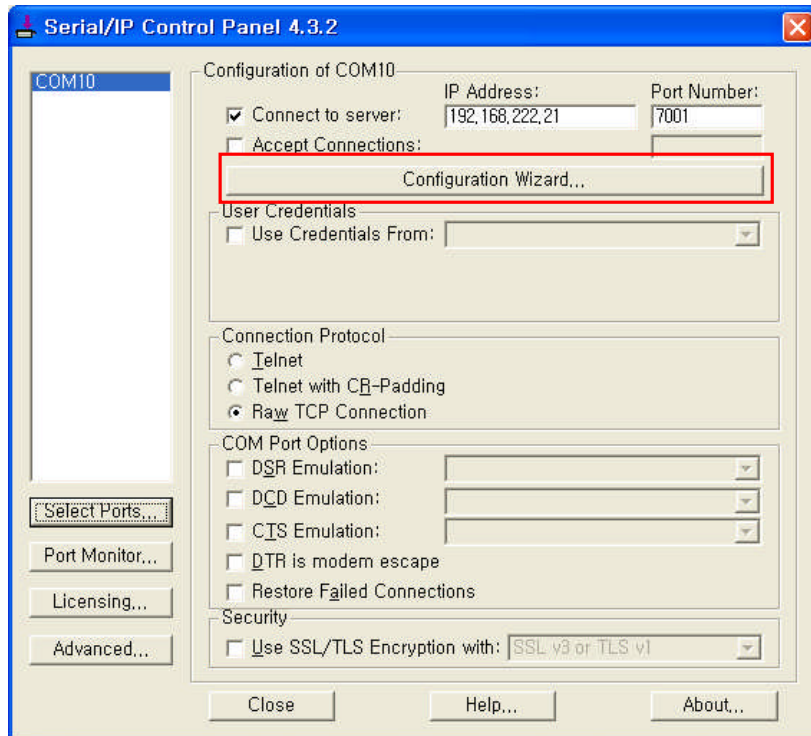
① 사용자의 Serial/IP Tray application에서 [Select Ports]를 클릭한 후, COM10을 선택합니다.



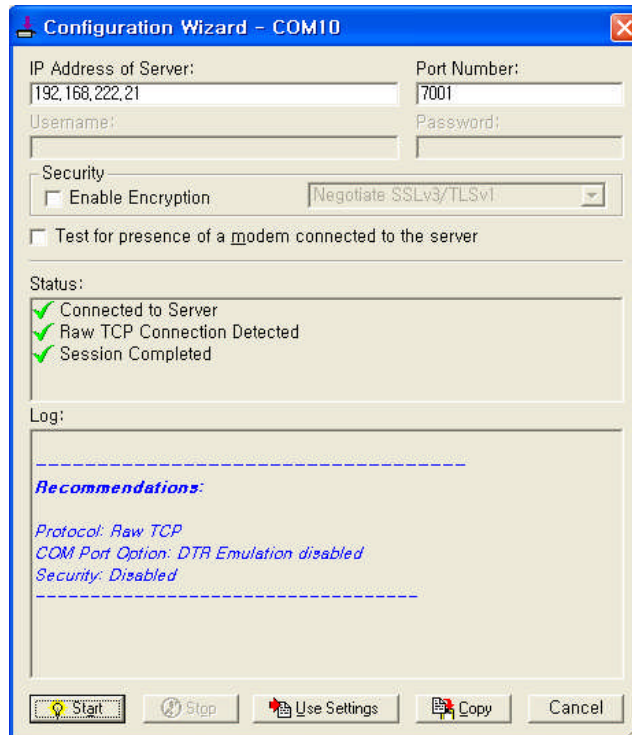
② Serial/IP Control Panel에서 해당되는 SS800의 IP 주소인 192.168.222.1과 1번 시리얼 포트 설정 값인 7001을 입력합니다.



㉓ Serial/IP Control Panel에서 [Configuration Wizard] 버튼을 클릭합니다.



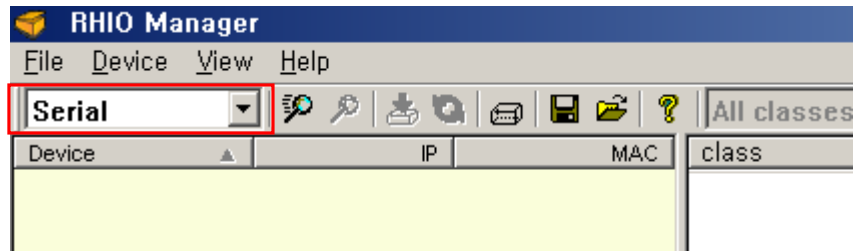
㉔ Configuration Wizard의 [Start] 버튼을 클릭하여 COM10으로 설정 완료합니다. 시험 세션이 완료되면 [Use Setting] 버튼을 클릭하여 창을 닫습니다.



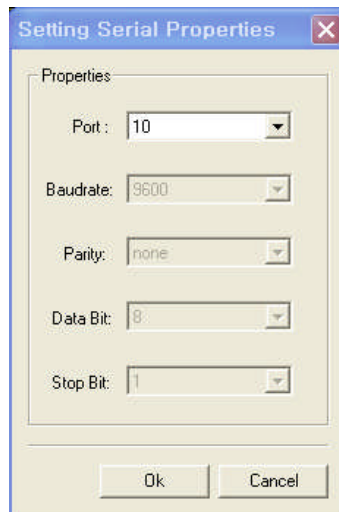
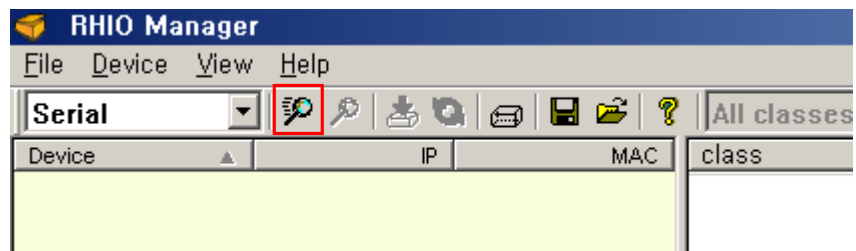
이제 사용자의 PC의 응용 프로그램에서 COM10으로 전송되는 모든 데이터는 SS800의 IP 주소, 192.168.222.21, 7001번 TCP 포트로 전송되도록 설정되었습니다.

3 단계. Rhio Manager를 이용한 I/O 장비 감시 및 제어

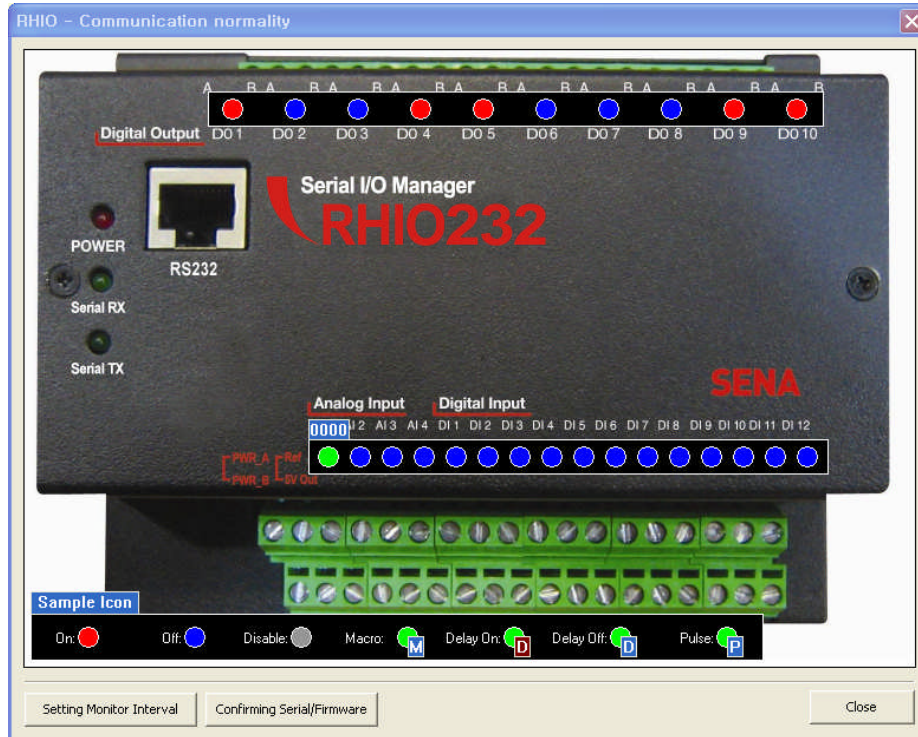
① Rhio Manager를 실행한 후, 통신 타입을 [Serial]로 선택합니다.



② [Probe]를 실행한 후, 해당되는 COM 포트 설정을 수행합니다. 이제 사용자는 COM10 포트를 이용하여 SS800에 연결된 Rhio232에 접속할 수 있게 됩니다.



㉔ [I/O CONTROL] 버튼을 선택하면 Rhio Manager I/O 관리 화면이 나타납니다. 사용자는 가상 시리얼 포트인 COM10을 통해서 원격에서 Rhio232를 설정하고 I/O 장비를 제어 및 감시할 수 있습니다. 아래 이미지는 Rhio Manager를 이용하여 Rhio232의 I/O 포트를 제어하는 모습입니다.



사용자는 위와 같은 절차를 거쳐서 Serial/IP, SS800 및 Rhio232를 이용하여 센서, 액츄에이터 등의 I/O 장치들을 원격에서 관리할 수 있습니다. 실제 응용의 경우에, 사용자는 Rhio Library를 이용하여 Rhio232와 시리얼 통신하는 응용 프로그램을 제작하여 사용자의 기존 프로그램과 통합함으로써, 원격 I/O 관리 시스템을 개발할 수 있습니다.

부록 A. 연결

A.1 시리얼 포트 Pin-Out

Rhio232 RJ45 커넥터의 핀 지정은 표 A-1에 요약되어 있습니다.

주의 : 시리얼 포트 와 호스트 PC의 연결 시 CAT5 케이블(패키지에 포함)의 한쪽 끝에 RJ45 to DB9 Female 커넥터(패키지에 포함)를 끼워서 사용합니다.

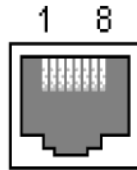


그림 A-1. RJ45 커넥터의 핀 배치

| Pin | Description |
|-----|-------------|
| 1 | RTS |
| 2 | DTR |
| 3 | TxD |
| 4 | GND |
| 5 | - |
| 6 | RxD |
| 7 | DSR |
| 8 | CTS |

표 A-1. 시리얼 포트 커넥터의 핀 지정

A.2 시리얼 연결 배선도

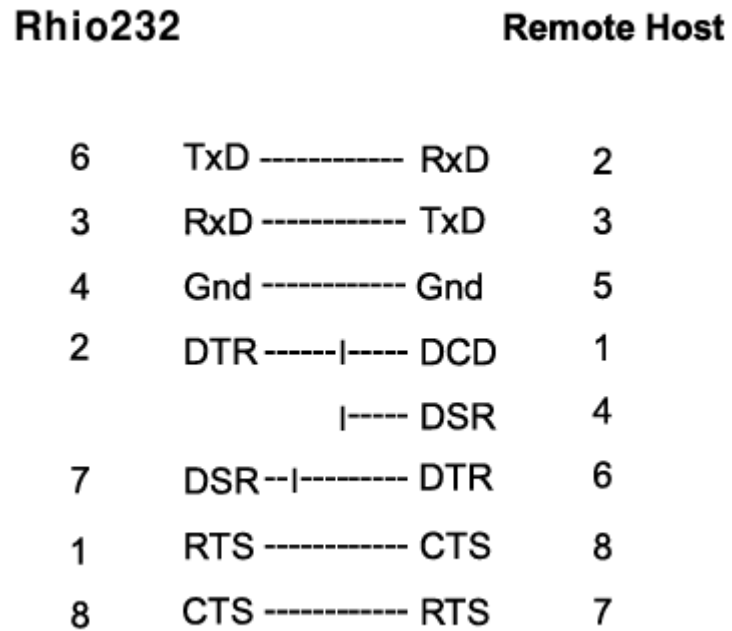


그림 A-2. RS232 시리얼 연결 배선도

부록 B. 문제 해결

B.1 전원/LED 상태

| 문제 | 원인 | 조치 |
|---------------------------|------------------------|---|
| Power LED가 점등 되지 않는다. | 전원 케이블이 연결되지 않았습니다. | 전원 연결을 확인하십시오. Rhio232 은 DC 9V 부터 48 V까지를 전원으로 입력을 받을 수 있습니다. |
| Serial Rx LED가 점멸 되지 않는다. | 시리얼케이블이 연결되지 않았습니다. | 시리얼 케이블 연결을 확인하십시오. |
| | 부적합한 시리얼 케이블이 사용되었습니다. | 시리얼 포트에는 반드시 Pin-out 정의에 맞게 제작된 시리얼 케이블(또는, CAT5 Straight Cable + DB9F Straight 어댑터)을 사용하십시오. |
| Serial Tx LED가 점멸 되지 않는다. | 시리얼케이블이 연결되지 않았습니다. | 시리얼 케이블 연결을 확인하십시오 |
| | 부적합한 시리얼 케이블이 사용되었습니다. | 시리얼 포트에는 반드시 Pin-out 정의에 맞게 제작된 시리얼 케이블(또는, CAT5 Straight Cable + DB9F Straight 어댑터)을 사용하십시오. |

B.2 시리얼 통신

| 문제 | 원인 | 조치 |
|-----------------------|------------------------|---|
| Rhio232 와 통신이 되지 않는다. | 시리얼 케이블이 연결되지 않았습니다. | 시리얼 케이블 연결을 확인하십시오. |
| | 부적합한 시리얼 케이블이 사용되었습니다. | 시리얼 포트에는 반드시 Pin-out 정의에 맞게 제작된 시리얼 케이블(또는, CAT5 Straight Cable + DB9F Straight 어댑터)을 사용하십시오. |
| | 시리얼 Baud rate 가 다릅니다. | 시리얼 Baud rate 를 9600-N-8-1 흐름제어 None 으로 설정해야 통신이 가능합니다. |

부록 C. Rhio 라이브러리

C.1 Enumeration Type

EonOffFlag

ON, OFF, 동작 조건 상태에 대한 Flag

enum EOnOffFlag

```
{
    EOF_ON,                //ON상태
    EOF_ON_ADC_LEVEL,     //ADC Level ON 상태
    EOF_OFF,              //OFF상태
    EOF_NOT,              //상태 모름 상태
    EOF_ON_DELAY,         //Delay ON 대기상태
    EOF_OFF_DELAY,        //Delay OFF 대기상태
    EOF_ONOFF_MACRO,     //Macro(조건) 대기상태
    EOF_ONOFF_PULSE      //Pulse 상태
};
```

SendStatusFlag

RHIO에 어떠한 Command를 송신하였는지에 대한 송신 상태 Flag

enum SendStatusFlag

```
{
    ESF_ONOFF,            //ON, OFF 제어 Command 송신 상태
    ESF_SET,              //
    ESF_SET_MODE,        //설정 모드로 변경 Command 송신 상태
    ESF_RUN_MODE,        //동작 모드로 변경 Command 송신 상태
    ESF_SET_MACRO,       //Macro 설정 Command 송신 상태
    ESF_SET_DELAY_PULSE, //Delay/Pulse 설정 Command 송신 상태
    ESF_SET_ADC,         //ADC 값 설정 Command 송신 상태
    ESF_SET_ENABLE,      //각각의 Port의 Enable/Disable 설정
//Command송신 상태
    ESF_SET_PWR_STOP,    //정전 복구 Enable/Disable 설정 Command
//송신상태
    ESF_SET_FACTROT_RESET, //Factory Reset Command 송신 상태
    ESF_SET_SERIAL,      //Serial 번호 설정 Command 송신 상태.
};
```

```

ESF_MON_MACRO,           //Macro 확인 Command 송신 상태.
ESF_MON_DELAY_PULSE,    //Delay/Pulse 확인 Command 송신 상태
ESF_MON_ADC,            //ADC 값 확인 Command 송신 상태
ESF_MON_ENABLE,        //Enable/Disable 확인 Command 송신 상태.
ESF_MON_PWR_STOP,      //정전 복구 확인 Command 송신 상태.
ESF_MON_SERIAL,        //Serial 번호 확인 Command 송신 상태.
ESF_MON_FIRMWARE,      //Firmware Version 확인 송신 상태.
ESF_NONE
};

```

EsetOutputFlag

Output Port Maco/Delay/Pulse 설정시 설정/제거에 대한 Flag

```
enum ESetOutputFlag
```

```

{
    ESOF_CLEAR,           //해당 Port의 설정을 Clear
    ESOF_SETTING         //해당 Port의 설정
};

```

EADCMode

ADC가 Level인지 Switch Mode인지에 대한 Flag

```
enum EADCMode
```

```

{
    EAM_LEVEL,
    EAM_SWITCH
};

```

C.2 Structure

ON/OFF 상태 Data

각각의 Port의 ON/OFF상태에 대한 Data를 저장하여 반환되는 구조체

```
typedef struct _ADCData //ADC Data
{
    EOnOffFlag eADC; //ADC On/OFF Flag
    char cADC[5]; //ADC Level 값 저장
}SADCData;

typedef struct _OnOffStatusData //전체 Port의 ON/OFF Data
{
    EOnOffFlag eOutput[10]; //Output Port 1~10까지
    EOnOffFlag eInput[12]; //Input Port 1~12까지
    SADCData sADC[4]; //ADC Port 1~4까지
}SOnOffStatusData;
```

설정 상태 Data

각각의 설정상태를 저장하여 Parameter로 넘기거나 반환되는 구조체

```
typedef struct _SetOutput //Output Port 설정 Data
{
    ESetOutputFlag eMacro; //MACRO Flag : ESOF_CLEAR,
    ESOF_SETTING
    char cMacro[106]; //MACRO Data
    ESetOutputFlag eDelayPulse; //DelayPulse Flag :
    // ESOF_CLEAR, ESOF_SETTING
    char cDelayPulse[11]; //DelayPulse Data
    bool isEnabled[10]; //Output port Enable
    bool isEnabledPowerStop[10]; //Output port power stoppage
}SSetOutput;
```

```
typedef struct _SetADC //ADC 설정 Data
{
    EADCMode eModeADC[4]; //ADC port Mode
    BYTE bReference; //ADC Reference
    SADCData sADCData[4]; //ADC Level Data
```



```
}SSetADC;
```

```
typedef struct _SetInput //Input Port 설정 Data
```

```
{
```

```
    bool isEnabled[12]; //Input port Enable
```

```
}SSetInput;
```

```
typedef struct _RHIOSetting //전체 Port 설정 Data
```

```
{
```

```
    SSetOutput sOutput; //Setting Output port Data;
```

```
    SSetADC sADC; //Setting ADC port Data;
```

```
    SSetInput sInput; //Setting Input port Data;
```

```
    char cGetSerial[18]; //수신된 모델명과 Serial번호
```

```
    char cGetFirmVer[9]; //수신된 Firmware Version
```

```
}SRHIOSetting;
```

C.3 Function

RHIO_CommProcessCreate

설명 : RHIO_Proc Dll의 Process Class를 Create하는 함수. 본 Dll을 사용하기 위해서는 반듯이 RHIO_CommProcessCreate 함수를 사용하여 Create를 하고 사용하여야 합니다.

함수 원형 :

```
extern "C" __declspec(dllexport) void RHIO_CommProcessCreate(CWnd *pParentWnd);
```

Parameter:

CWnd *pParentWnd : Process Class의 부모 Window가 되는 Wnd Class의 Pointer

Return : 없음

eg. RHIO_CommProcessCreate(this)

RHIO_SockConnect

설명 : RHIO와 Socket(TCP/IP)으로 접속하고자 할 때 사용하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SockConnect  
(BYTE bAddr1, BYTE bAddr2, BYTE bAddr3, BYTE bAddr4, int iPort);
```

Parameter :

BYTE bAddr1 : IP Address 1번.

BYTE bAddr2 : IP Address 2번.

BYTE bAddr3 : IP Address 3번.

BYTE bAddr4 : IP Address 4번.

int iPort : Port 번호.

Return : 정상적으로 Connect되면 true를 그렇지 않으면 false를 반환한다.

eg :

```
if(RHIO_SockConnect(192, 168, 100, 2, 6001))  
    AfxMessageBox("정상적으로 접속하였습니다.");  
else  
    AfxMessageBox("접속하지 못 하였습니다.");
```

RHIO_CommConnect

설명 : RHIO와 Serial Port로 접속하고자 할 때 사용하는 함수. (Baudrate:9600, Parity:none, Databit:8, Stopbit:1 고정)

함수 원형 : extern "C" __declspec(dllexport) bool RHIO_CommConnect(int iPort);

Parameter :

int iPort : Port 번호.

Return : 정상적으로 Connect되면 true를 그렇지 않으면 false를 반환한다.

eg.

```
if(RHIO_CommConnect(1))
    AfxMessageBox("정상적으로 접속하였습니다.");
else
    AfxMessageBox("접속하지 못 하였습니다.");
```

RHIO_Close

설명 : Connect후 해당 RHIO와 접속을 끊기 위한 함수. (Socket, Comm 전부 적용됨)

함수 원형 : extern "C" __declspec(dllexport) bool RHIO_Close();

Parameter : 없음

Return : 정상적으로 종료되면 true 그렇지 않으면 false를 반환한다.

eg : RHIO_Close();

※ Connect후 재접속을 하기 위해선 반드시 Close 함수를 사용해야 한다.

RHIO_SndCmd_SetOnOff

설명 : ON/OFF 제어 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetOnOff
(int iOutputNum, EOnOffFlag eOnOff);
```

Parameter :

int iOutputNum : 제어할 Output Port의 번호 (1~10)

EOnOffFlag eOnOff : 제어 Flag (EOF_ON, EOF_OFF)

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_ONOFF)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:제어 정상, 0x31 : 동작 Mode 아님, 0x32:제어 비정상)

eg : RHIO_SndCmd_SetOnOff(1, EOF_ON);

RHIO_SndCmd_GetOnOff

설명 : ON/OFF 확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetOnOff()
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

eg : RHIO_SndCmd_GetOnOff();

RHIO_SndCmd_SetSettingMod

설명 : 설정 모드 전환 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetSettingMode()
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter

WPARAM : Command 송신 상태 Flag (ESF_SET_MODE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)
(0x30:설정 Mode, 0x31:동작 Mode)

eg : RHIO_SndCmd_SetSettingMode();

RHIO_SndCmd_SetRunMode

설명 : 동작 모드 전환 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetRunMode()
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter

WPARAM : Command 송신 상태 Flag (ESF_RUN_MODE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)
(0x30:설정 Mode, 0x31:동작 Mode)

eg : RHIO_SndCmd_SetRunMode();

RHIO_SndCmd_SetMACRO

설명 : Input Port의 Macro 설정 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetMACRO  
(SRHIOSetting sSetData, int iPortNum);
```

Parameter :

SRHIOSetting sSetData : 설정할 값을 저장하는 Data.

SRHIOSetting.sInput.eMacro -> 설정할 것인지 Clear할 것인지 저장.

(ESOF_CLEAR, ESOF_SETTING)

SRHIOSetting.sInput.cMacro -> Macro 문자열 저장.

int iPortNum : 설정할 Port 번호.

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter

WPARAM : Command 송신 상태 Flag (ESF_SET_MACRO)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님, 0x32:미설정)

eg :

설정 시 :

```
SRHIOSetting sSetData;
```

```
sSetData.sInput.eMacro = ESOF_SETTING;
```

```
memset(sSetData.sInput.cMacro, '\0', sizeof(sSetData.sInput.cMacro));
```

```
memcpy(sSetData.sInput.cMacro, "O2&O4|1", 8);
```

```
RHIO_SndCmd_SetMACRO(sSetData, 1);
```

설정 Clear 시 :

```
SRHIOSetting sSetData;
```

```
sSetData.sInput.eMacro = ESOF_CLEAR;
```

```
RHIO_SndCmd_SetMACRO(sSetData, 1);
```

RHIO_SndCmd_GetMACRO

설명 : Input Port의 Macro 확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetMACRO(int iPortNum);
```

Parameter : int iPortNum : 확인할 Input Port 번호.

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter

WPARAM : Command 송신 상태 Flag (ESF_MON_MACRO)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님, 0x32:미설정)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함수에서 사용하였던 Parameter 값을 확인한다.

eg : RHIO_SndCmd_GetMACRO (1);

RHIO_SndCmd_SetDelayPulse

설명 : Input Port의 Delay/Pulse 설정 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetDelayPulse
```

```
(SRHIOSetting sSetData, int iPortNum);
```

Parameter :

SRHIOSetting sSetData : 설정할 값을 저장하는 Data.

SRHIOSetting.sInput.eDelayPulse -> 설정할 것인지 Clear할 것인지 저장.

(ESOF_CLEAR, ESOF_SETTING)

SRHIOSetting.sInput.cDelayPulse -> Time 문자열 저장 10Byte

| Byte Size | 5(ON Time) | 5(OFF Time) |
|-----------|-----------------|-----------------|
| 동작 설정 | "00000"~"50000" | "00000"~"50000" |

int iPortNum : 설정할 Port 번호.

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_SET_DELAY_PULSE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님, 0x32:미설정)

eg :

설정 시 :

```
SRHIOSetting sSetData;
```

```
sSetData.sInput.eDelayPulse = ESOF_SETTING;
```

```
memset(sSetData.sInput.cDelayPulse, '\0',
```

```
sizeof(sSetData.sInput.cDelayPulse));
```

```
memcpy(sSetData.sInput.cDelayPulse, "0050000000", 10);
```

```
RHIO_SndCmd_SetDelayPulse(sSetData, 1);
```

설정 Clear 시 :

```
SRHIOSetting sSetData;
```

```
sSetData.sInput.eDelayPulse = ESOF_CLEAR;
RHIO_SndCmd_SetDelayPulse (sSetData, 1);
```

RHIO_SndCmd_GetDelayPulse

설명 : Input Port의 Delay/Pulse 확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetDelayPulse(int iPortNum);
```

Parameter :

int iPortNum : 확인할 Input Port 번호.

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_MON_DELAY_PULSE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님, 0x32:미설정)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함수에서 사용하였던 Parameter 값을 확인한다.

eg : RHIO_SndCmd_GetDelayPulse(1);

RHIO_SndCmd_SetADC

설명 : 전체 ADC Port(1~4)의 Level 설정 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetADC(SRHIOSetting sSetData);
```

Parameter :

SRHIOSetting sSetData : 설정할 값을 저장하는 Data.

SRHIOSetting.sADC.sADCData[index].cADC -> ADC Level 문자열로 저장.

("0000"~"1023", 4 Byte 저장)

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_SET_ADC)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

eg :

SRHIOSetting sSetData;

```

for(int index = 0; index < 4; index++)
{
    memset(sSetData.sADC.sADCData[index].cADC, '\0',
sizeof(sSetData.sADC.sADCData[index].cADC));
    memcpy(sSetData.sADC.sADCData[index].cADC, "0512", 4);
}
RHIO_SndCmd_SetADC(sSetData);

```

RHIO_SndCmd_GetADC

설명 : 전체 ADC Port의 확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetADC();
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_MON_ADC)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함수에서 사용하였던 Parameter 값을 확인한다.

eg : RHIO_SndCmd_GetADC();

RHIO_SndCmd_SetPortEnable

설명 : 전체 Port의 Enable/Disable 설정 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetPortEnable(SRHIOSetting sSetData);
```

Parameter :

SRHIOSetting sSetData : 설정할 값을 저장하는 Data.

SRHIOSetting.sADC.eModeADC[4] -> Level/Switch Mode 설정.

(EAM_LEVEL, EAM_SWITCH)

SRHIOSetting.sADC.bReference -> ADC Reference 값 설정.

(0x30:Avcc, 0x31:내부, 0x32:외부)

SRHIOSetting.sInput.isEnable[12] -> Input Enable/Disable

(Enable:true, Disable:false)

SRHIOSetting.sOutput.isEnabled[10] -> Input Enable/Disable

(Enable:true, Disable:false)

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_SET_ENABLE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

eg :

```
SRHIOSetting sSetData;
```

```
for(int index = 0; index < 4; index++)
```

```
    SetData.sADC.eModeADC[index] = EAM_SWITCH;
```

```
sSetData.sADC.bReference = 0x30;
```

```
for(int index = 0; index < 12; index++)
```

```
    SetData.sInput.isEnabled[index] = true;
```

```
for(int index = 0; index < 10; index++)
```

```
    sSetData.sOutput.isEnabled[index] = true;
```

```
RHIO_SndCmd_SetPortEnable(sSetData);
```

RHIO_SndCmd_GetPortEnable

설명 : 전체 Port의 Enable/Disable 확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetPortEnable();
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_MON_ENABLE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함수에서 사용하였던 Parameter 값을 확인한다.

eg : RHIO_SndCmd_GetPortEnable();

RHIO_SndCmd_SetPwrStopEnable

설명 : 전체 Input Port의 정전 복구 Enable/Disable 설정 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetPwrStopEnable  
(SRHIOSetting sSetData);
```

Parameter :

SRHIOSetting sSetData : 설정할 값을 저장하는 Data.

SRHIOSetting.sOutput.sEnablePowerStop[10] -> Input Enable/Disable
(Enable:true, Disable:false)

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_SET_PWR_STOP)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)
(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

eg :

```
SRHIOSetting sSetData;
```

```
for(int index = 0; index < 10; index++)
```

```
    sSetData.sOutput.sEnablePowerStop[index] = true;
```

```
RHIO_SndCmd_SetPortEnable(sSetData);
```

RHIO_SndCmd_GetPwrStopEnable

설명 : 전체 Input Port의 정전 복구 Enable/Disable확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetPwrStopEnable();
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_MON_PWR_STOP)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)
(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함수에서 사용하였던 Parameter 값을 확인한다.

eg : RHIO_SndCmd_GetPwrStopEnable();

RHIO_SndCmd_SetFactoryReset

설명 : RHIO Factory Reset Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetFactoryReset();
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_SET_FACTORY_RESET)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG)

eg : RHIO_SndCmd_SetFactoryReset;

RHIO_SndCmd_SetSerial

설명 : RHIO Serial 번호 설정 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_SetSerial(CString strSerial);
```

Parameter :

CString strSerial -> Serial 번호 문자열 (Limit Length : 12)

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_SET_SERIAL)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

eg : RHIO_SndCmd_SetSerial("0000003");

RHIO_SndCmd_GetSerial

설명 : RHIO Serial 번호 확인 Command를 송신하는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetSerial();
```

Parameter : 없음

Return : 정상적으로 송신하면 true, 그렇지 않으면 false를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_MON_SERIAL)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함

수에서 사용하였던 **Parameter** 값을 확인한다.

eg : RHIO_SndCmd_GetSerial();

RHIO_SndCmd_GetFirmware

설명 : RHIO Firmware Version 확인 Command를 송신하는 함수

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_SndCmd_GetFirmware();
```

Parameter : 없음

Return : 정상적으로 송신하면 **true**, 그렇지 않으면 **false**를 반환한다.

수신 Event 발생시 Parameter :

WPARAM : Command 송신 상태 Flag (ESF_MON_FIRMWARE)

LPARAM : Response의 제어 상태 Flag(Time Out Error시 : false)

(0x30:설정 OK, 0x39:동작 NG, 0x31:설정 Mode 아님)

※ Response 수신 Event 발생시 RHIO_GetSettingData를 이용하여 Data를 읽은 후, Set 함수에서 사용하였던 **Parameter** 값을 확인한다.

eg : RHIO_SndCmd_GetFirmware();

RHIO_GetSettingData

설명 : 각각의 Port에 설정 확인 Response 수신 Event가 발생하였을 경우 해당 Data들을 읽어 들이는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_GetSettingData(SRHIOSetting &sSetData);
```

Parameter :

SRHIOSetting sSetData : 설정할 값을 저장하는 Data.

Return : 정상적으로 읽으면 **true**, 그렇지 않으면 **false**를 반환한다.

eg :

```
SRHIOSetting sSetData;
```

```
RHIO_GetSettingData(sSetData);
```

//sSetData에 해당 Port의 상태 값들이 저장된다.

RHIO_GetOnOffData

설명 : ON/OFF 상태변화 Response 및 ON/OFF 제어 Response 수신 Event가 발생하였을 경우 해당 Data들을 읽어 들이는 함수.

함수 원형 :

```
extern "C" __declspec(dllexport) bool RHIO_GetOnOffData(SOnOffStatusData &sOnOffData);
```

Parameter :

SOnOffStatusData sOnOffData : On/Off 상태 값을 저장하는 Data.

Return : 정상적으로 읽으면 true, 그렇지 않으면 false를 반환한다.

eg :

```
SOnOffStatusData sOnOffData;
```

```
RHIO_GetOnOffData(sOnOffData);
```

```
//sSetData에 ON/OFF 상태 값들이 저장된다.
```