

# **Rhio10**

## **Remote I/O Manager**

사용자 설명서

**Version 1.4.3**

**2007-05-07**

## **Rhio10 사용 설명서**

버전 1.4.3

펌웨어 버전 1.4.x

Printed in Korea

### **저작권**

Copyright 2002~2007, 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®는 Microsoft Corporation의 등록 상표입니다.

Ethernet®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

전화: (02) 573-7772

팩스: (02) 573-7710

E-Mail: support.kr@sena.com

Website: www.sena.co.kr

## 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	JS.Kim	동작온도, 저장온도 및 습도 범위 값 변경
V1.4.1	2006-11-15	JS.Kim	IP Filtering 추가
V1.4.2	2007-03-22	JS.Kim	LRC 설명 수정 및 계산에 대한 예제 추가
V1.4.3	2007-05-07	H.R. Zo	저작권, 기술 지원 수정

# 목 차

<b>1. 서론.....</b>	<b>7</b>
1.1 개요.....	7
1.2 패키지 체크 리스트.....	7
1.3 제품 사양.....	8
<b>2. 시작하기.....</b>	<b>10</b>
2.1 패널 배치.....	10
2.2 하드웨어 연결.....	12
2.2.1 설치 및 부착.....	12
2.2.2 전원 연결.....	13
2.2.3 네트워크 연결.....	14
2.3 Rhio Manager 설치.....	14
2.3.1 Rhio Manager 설치.....	14
2.3.2 Rhio Manager를 이용한 기본 설정.....	15
<b>3. 시스템 설정 및 관리.....</b>	<b>17</b>
3.1 네트워크 설정.....	17
3.1.1 Static IP.....	19
3.1.2 DHCP.....	20
3.1.3 PPPoE.....	21
3.1.4 IP 필터링.....	21
3.2 호스트 모드 설정.....	22
3.2.1 TCP Server 모드 동작.....	25
3.2.2 TCP Client 모드 동작.....	27
3.2.3 TCP Server/Client 모드 동작.....	29
3.3 시스템 관리.....	30
3.4 공장 출하 시 기본값 복구.....	31
<b>4. I/O 설정 및 응용.....</b>	<b>32</b>
4.1 I/O 감시 및 제어.....	32
4.1.1 LED.....	32
4.1.2 감시 주기 설정.....	33
4.1.3 I/O Port 감시 동작.....	33
4.1.4 Digital Output Port 제어.....	33
4.2 Digital Input 설정.....	34
4.2.1 Enable/Disable 설정.....	34

4.3 Digital Output Port 설정 .....	35
4.3.1 Enable/Disable 설정 .....	35
4.3.2 동작 조건 설정 .....	36
4.3.3 Delay 와 Pulse 동작 .....	37
4.3.4 정전 후 복구 설정 .....	37
4.4 ADC Input Port 설정 .....	37
4.4.1 Reference 설정 .....	38
4.4.2 ADC Input Port 동작 설정 .....	38
4.5 I/O Port 연결 방법 .....	39
4.5.1 Digital Output Port.....	39
4.5.2 Digital Input Port.....	39
4.5.3 ADC Input Port.....	40
<b>5. 소프트웨어 개발 및 응용 .....</b>	<b>42</b>
5.1 Rhio 라이브러리 .....	42
5.1.1 Rhio 라이브러리 개요 .....	42
5.1.2 Reference .....	43
5.2 Rhio 라이브러리를 이용한 샘플 프로그램 제작 및 데모 .....	45
5.2.1 프로그램 UI 구성 및 관련 클래스 .....	45
5.2.2 처리 절차.....	46
5.3 Rhio 통신 프로토콜 .....	49
5.3.1 개요 .....	49
5.3.2 ON/OFF 제어 .....	52
5.3.3 Input/Output 상태 .....	54
5.3.4 설정/동작 .....	54
5.3.5 Output Port 설정 .....	55
5.3.6 ADC Input Port 설정 .....	58
5.3.7 Port Enable 설정 .....	59
5.3.8 Port 정전 후 복구 설정.....	61
<b>부록 A. 연결.....</b>	<b>64</b>
A.1 Ethernet Pin Outs .....	64
A.2 콘솔 Port Pin Outs .....	64
A.3 Ethernet 배선도 .....	65
A.4 시리얼 콘솔 배선도 .....	66
<b>부록 B. Rhio 콘솔 사용법 및 콘솔 명령어 일람.....</b>	<b>67</b>
B.1 콘솔 Port 접근.....	67
B.1.1 시리얼 콘솔 사용 .....	67

B.1.2 원격 콘솔 사용.....	69
B.2 명령어 사용법.....	70
B.2.1 'set' 명령어 .....	71
B.2.2 'get' 명령어 .....	72
B.2.3 'help' 명령어.....	74
B.2.4 'save' 명령어.....	75
B.2.5 'exit' 명령어 .....	75
B.2.6 'reboot' 명령어 .....	75
B.3 콘솔 명령어들을 이용한 시스템 설정 .....	75
B.3.1 네트워크 설정 .....	75
B.3.2 호스트 모드 설정 .....	76
B.3.3 시스템 관리.....	78
<b>부록 C. 문제 해결 .....</b>	<b>79</b>
C.1 전원/LED 상태.....	79
C.2 시리얼 콘솔 .....	79
C.3 원격 콘솔 .....	80
C.4 IP 주소 .....	80
C.5 DHCP.....	81
<b>부록 D. Rhio 라이브러리.....</b>	<b>82</b>
D.1 Enumeration Type .....	82
D.2 Structure .....	84
D.3 Function .....	85
<b>부록 E. 품질 보증 정책.....</b>	<b>98</b>
E.1 제품 품질 보증 정책.....	98
E.2 책임의 한계 .....	98
E.3 하드웨어 제품 보증 상세 내용 .....	98
E.4 소프트웨어 제품 보증의 상세 .....	99
E.5 제3자 소프트웨어 제품 보증의 상세.....	99

# 1. 서론

## 1.1 개요

Rhio10은 10Base-T Ethernet 네트워크를 통해 원격에서 각종 Digital 센서 및 액츄에이터와 통신할 수 있게 하는 원격 I/O 관리 장치입니다.

Rhio10은 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, Rhio10은 최대 12개의 절연 Digital Input 접점들을 감시할 수 있는 기능을 제공합니다. Rhio10의 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.

Rhio10은 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

- Rhio10 외장 박스
- CAT5 케이블
- RJ45 to DB9 Female 케이블 커넥터
- DIN 레일 거치형 키트
- Quick Start Guide
- Rhio Manager, Rhio10 DLL 및 사용자 설명서가 포함된 CD-ROM

### 1.3 제품 사양

<b>Network Interface</b>	10Base-T Ethernet 네트워크 지원, RJ45 커넥터
	고정 및 유동 IP주소 지원
<b>Digital Input</b>	<ul style="list-style-type: none"> <li>- Input 개수: 12</li> <li>- Input 방식: 전압</li> <li>- Input 회로: 절연 photo-coupler</li> <li>- Input 전압: 0V ~ ±24V OFF 0V ~ ±1.2V, ON ±3.3V ~ ±24V</li> <li>- Input 검지시간: 20ms</li> <li>- Input 절연전압: 5KV</li> </ul>
<b>Digital Output</b>	<ul style="list-style-type: none"> <li>- Output 개수: 10</li> <li>- Output 방식: Relay 접점Output</li> <li>- Output 정격부하: 3A/240VAC</li> <li>- 접점절연저항: 1000MΩ Min (DC500V)</li> <li>- Output 절연전압(coil and contact): 4KV</li> <li>- Output 지연시간 : 10msec이하</li> </ul>
<b>Analog Input</b>	<ul style="list-style-type: none"> <li>- Input 개수: 4</li> <li>- 분해능: 10-bit</li> <li>- Input 방식: 전압, Direct Coupling 방식</li> <li>- Input 범위: 0V ~ Aref (레퍼런스 전압, 2~5V)</li> <li>- Input 검지시간: 1000 samples/sec</li> </ul>
<b>프로토콜</b>	<p>ARP, IP/ICMP, TCP telnet, DHCP client, PPPoE</p>
<b>관리</b>	- Rhio Manager Windows Utility, 시리얼 콘솔 및 Telnet
<b>소프트웨어 지원</b>	<ul style="list-style-type: none"> <li>- Windows MFC DLL library</li> <li>- I/O 설정, 상태 감시 및 제어</li> </ul>
<b>I/O 표시부</b>	<ul style="list-style-type: none"> <li>- 전원, Link, Act (Power, Link, Act)</li> <li>- Digital Output(Digital Output), 1~10</li> <li>- Digital Input(Digital Input), 1~12</li> <li>- Analog Input(Analog Input), 1~4</li> </ul>
<b>전원</b>	- 공급전압 9V~48VDC, 최대 5W
<b>환경</b>	<ul style="list-style-type: none"> <li>- 작동 온도: 0℃ to 50℃</li> <li>- 보관 온도: -20℃ to 66℃</li> <li>- 습도: 90%이하(단, 결로현상이 없을 것)</li> </ul>



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 Rhio10을 설정 및 구성하는 방법을 설명합니다.

- 2.1 패널 배치에서는 패널의 배치와 LED 표시등에 대해 설명합니다.
- 2.2 하드웨어 연결에서는 Rhio10를 DIN 레일에 부착 및 설치 하는 방법과 Rhio10에 전원 및 네트워크를 연결하는 방법을 설명합니다.

시작하려면 아래의 장치들이 필요합니다.

- DIN 레일 거치형 키트(패키지에 포함)
- 설정 및 네트워크용 CAT5 케이블(패키지에 포함)
- 설정용 RJ45 to DB9 Female 커넥터(패키지에 포함)
- 네트워크 카드(이하 NIC)와 RS232 시리얼 포트가 있는 PC

### 2.1 패널 배치

Rhio10은 시스템 상태 표시를 위한 LED 표시등이 있습니다. 좌측의 표시등은 POWER, LINK 및 ACT 인데, 이 중 POWER는 시스템 전원 인가 상태를 나타냅니다. 다음 두개의 LINK, ACT 표시등은 10Base-T Ethernet 연결 및 활동 상태를 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.

표시등		기능
10Base-T	Link	10 Base-T Ethernet 네트워크에 연결되어 있는 경우 녹색으로 점등 된다.
	Act	Ethernet Port를 통해 패킷이 들어오고 나갈 때 녹색으로 깜박거린다.
상태	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.

표 2-1 LED 표시등

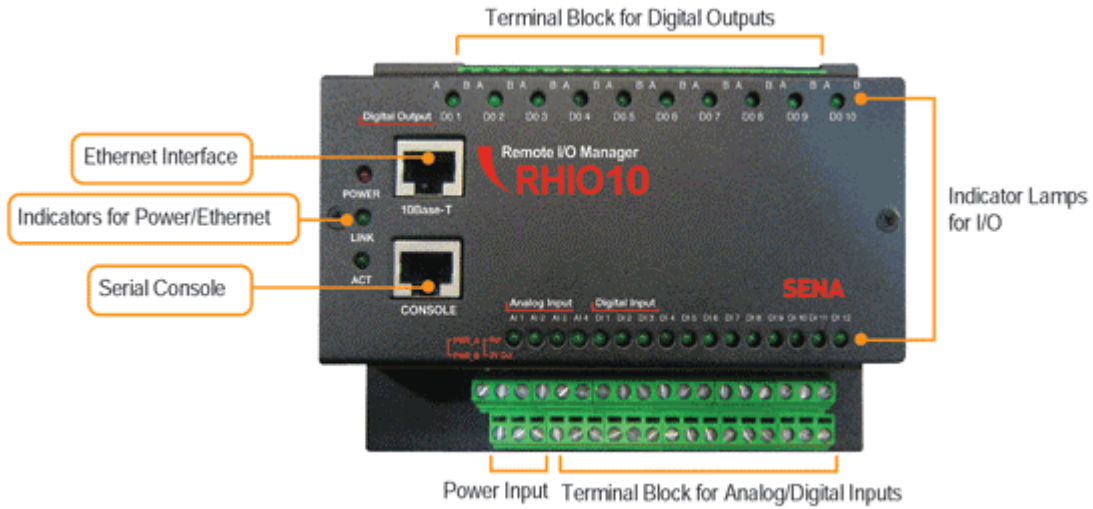


그림 2-1 Rhio10의 패널 배치

표 2-2 Rhio10의 터미널 블록 배치

		AI1	AI2	AI3	AI4	DI1	DI2	DI3	DI4	DI5	DI6	DI7	DI8	DI9	DI10	DI11	DI12		
입력부 블록 배치	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"> <li>• 전원 입력 ( PWR_A , PWR_B ) : (1,2)</li> <li>• 아날로그 입력 참조 전압 ( Ref ) : (3)               <ul style="list-style-type: none"> <li>- 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.</li> </ul> </li> <li>• 5V Out : (4)               <ul style="list-style-type: none"> <li>- AVCC 출력 블록으로, 사용자의 장비에 전원을 제공할 수 있습니다. 그라운드는 5,7,9,11번 블록입니다.</li> </ul> </li> <li>• 아날로그 입력 ( AI1 ~ AI4 ) : (5,6),(7,8),(9,10),(11,12)</li> <li>• 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)</li> </ul> <p>참고 : 1. 각각의 입력은 위/아래 각각 한 쌍의 블록으로 구성됩니다. 2. 3,4번 블록을 제외한 나머지 입력은 극성에 상관 없이 연결이 가능합니다.</p>																		
출력부 블록 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	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<p>다음과 같이 연달아 배치된 A, B 한 쌍의 블록이 각각의 출력을 구성합니다.</p> <ul style="list-style-type: none"> <li>• 디지털 출력 ( DO1 ~ DO10 ) : (1,2), (3,4), (5,6), (7,8), (9,10), (11,12), (13,14), (15,16), (17,18), (19,20)</li> </ul>																			

## 2.2 하드웨어 연결

### 2.2.1 설치 및 부착

패키지에 포함되어 있는 DIN 레일 거취형 키트를 Rhio10에 부착하여 사용자는 원하는 위치에 Rhio10을 용이하게 설치할 수 있습니다.

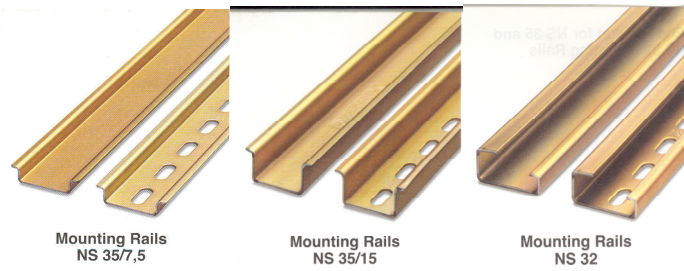
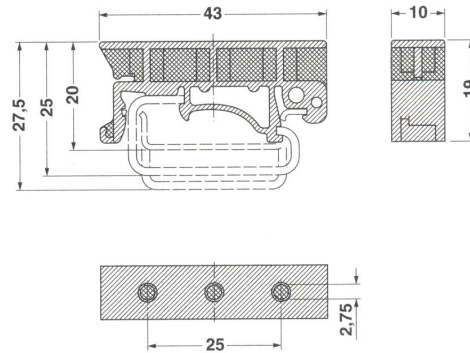


그림 2-2 DIN 레일 거취형 키트 치수 및 적용되는 DIN 레일

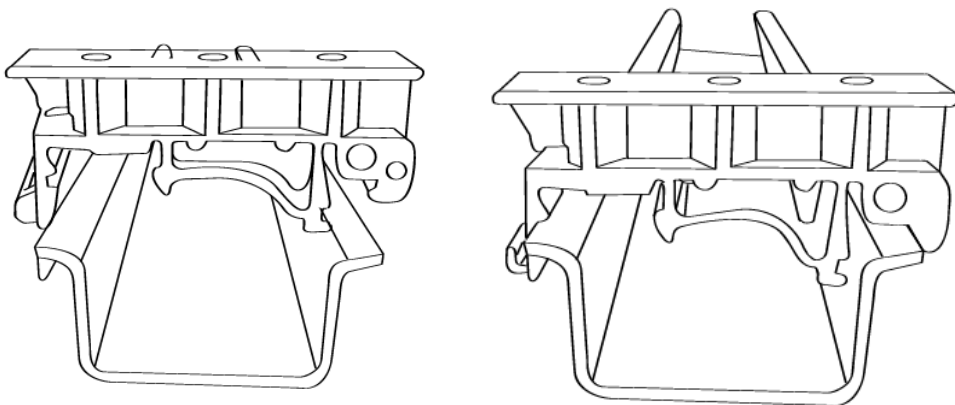


그림 2-3 DIN 레일에 DIN 레일 거취형 키트를 부착하는 방법

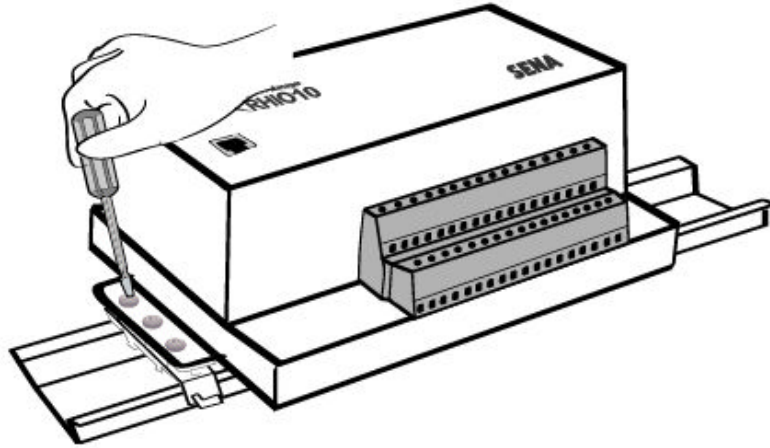


그림 2-4 DIN 레일에 Rhio10 설치 및 부착

## 2.2.2 전원 연결

사용자가 9V~48VDC, MAX. 5W의 제품 전원 사양에 맞추어 적합한 전원을 직접 Rhio10 전원 단자에 연결합니다. 전원이 제대로 공급된 경우, Rhio10의 **[Power]** 표시등이 적색 점등 상태를 유지합니다. 전원을 공급하는 선은 3M(미터)를 넘지 않도록 합니다.

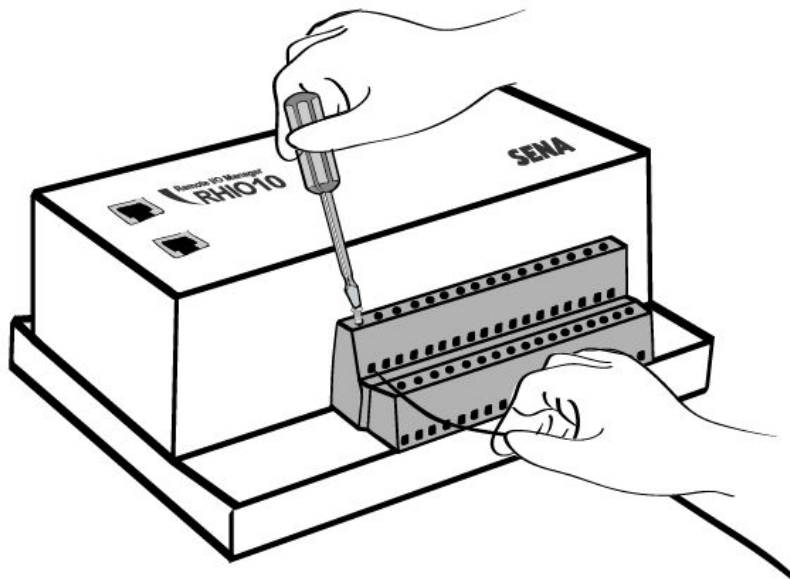


그림 2-5 Rhio10에 전원 연결

### 2.2.3 네트워크 연결

Ethernet 케이블 한쪽 끝을 Rhio10의 10Base-T Port에 연결하고 반대쪽 끝을 Ethernet 네트워크에 연결합니다. 케이블이 올바르게 연결된 경우, Rhio10과 Ethernet 네트워크간의 연결표시는 다음과 같이 나타납니다.

- Rhio10의 [Link] 표시등은 녹색 점등 상태를 유지합니다.
- [Act] 표시등은 계속해서 깜박거리면서 Ethernet 패킷의 송수신 여부를 나타냅니다.

위와 같이 되지 않는다면 Rhio10의 Ethernet 네트워크 연결이 올바르지 않은 것입니다.

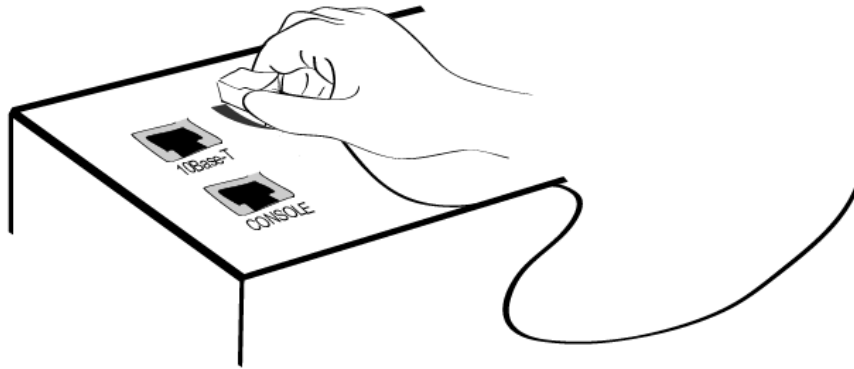


그림 2-6 Rhio10에 네트워크 케이블 연결

## 2.3 Rhio Manager 설치

Rhio Manager 는 Windows O/S 기반에서 동작하는 사용자 유틸리티 프로그램으로서 사용자는 이 프로그램을 사용하여 Rhio10 의 네트워크, 호스트 모드, 시스템 설정 등의 동작과 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 ㈜세나테크놀로지 홈페이지 다운로드 센터 ([www.sena.com/support/downloads](http://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.

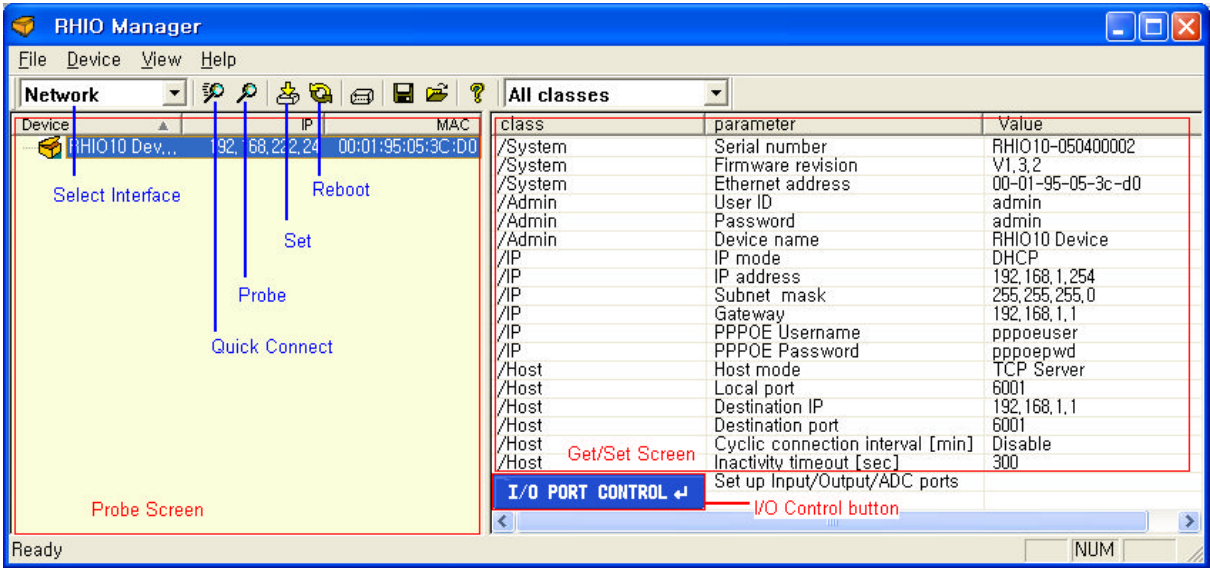


그림 2-7 Rhio Manager 실행 화면

### 2.3.2 Rhio Manager를 이용한 기본 설정

#### Select Interface

Rhio10 과의 Rhio Manager와의 통신 방법을 지정합니다. Rhio10을 사용할 때는, “Network” 모드로 설정하여야 합니다.

#### Quick Connect

원격지에 연결된 Rhio10의 설정 화면에 Rhio Manager를 통해 접속합니다. 이때, 원격지의 Rhio10의 IP 주소를 입력하여야 합니다. [Quick Connect] 명령을 수행하면 접속된 Rhio10이 왼쪽 Probe screen에 트리 구조로서 나타나며, 제품을 선택하면 ID와 Password를 묻는 창이 나타납니다. 사용자가 정확한 ID/Password를 입력하여야 Rhio10의 기본 환경 정보가 Get/Set screen 에 표시됩니다 (공장 초기 ID:PASSWORD 설정치는 admin : admin입니다.)

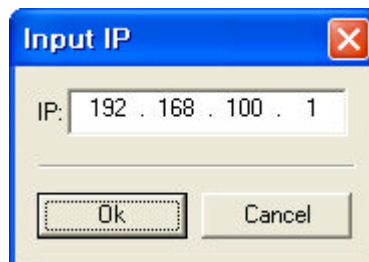


그림 2-8 Quick connect 실행 화면

### **Probe**

네트워크 상에 연결된 Rhio10들을 검색하여 표시합니다. [Probe] 명령을 수행하면 검색된 Rhio10이 Probe screen에 나타나며, 설정 또는 접속을 원하는 제품을 선택하면 ID와 Password를 묻는 창이 나타납니다. 사용자는 설정된 ID 와 Password를 입력하여 Rhio10의 기본 환경 등을 불러와 Get/Set screen에 표시합니다.

### **Set**

사용자는 Get/Set screen에 나타난 Rhio10의 네트워크 설정, 호스트 모드 설정 및 시스템 설정 등을 확인하고 원하는 값으로 클릭한 후 변경합니다. 설정 값 변경 후 Set을 실행하면 환경을 저장하고 적용하도록 합니다. Set을 실행하면 자동으로 Reboot 되어 설정값이 저장되고 적용됩니다.

### **Reboot**

Rhio10을 Reboot 시킵니다. 이 때 Rhio10과의 TCP 연결은 종료되며 I/O 부분의 동작은 계속 유지됩니다.

### **I/O PORT CONTROL**

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



### 3. 시스템 설정 및 관리

#### 3.1 네트워크 설정

Rhio10이 사용자의 네트워크 환경에서 올바르게 작동하려면 먼저 유효한 IP 주소를 할당해야 합니다. 네트워크를 위한 이 IP 주소 설정 정보는 네트워크 시스템 관리자가 제공합니다. IP 주소는 네트워크에서 유일해야 합니다. 그렇지 않으면 Rhio10이 네트워크에 올바르게 연결될 수 없습니다.

사용자는 세 개의 IP 작동 모드, 즉 Static IP, DHCP 및 PPPoE 중 원하는 IP 모드를 선택할 수 있습니다. 출하 시 기본 IP 모드는 DHCP 모드입니다. 표 3-1은 IP 설정 메뉴의 파라미터 항목을 보여줍니다.

표 3-1 IP 설정 파라미터

Static IP	IP address
	Subnet mask
	Default gateway
DHCP	No parameters required
PPPoE	PPPoE username
	PPPoE password

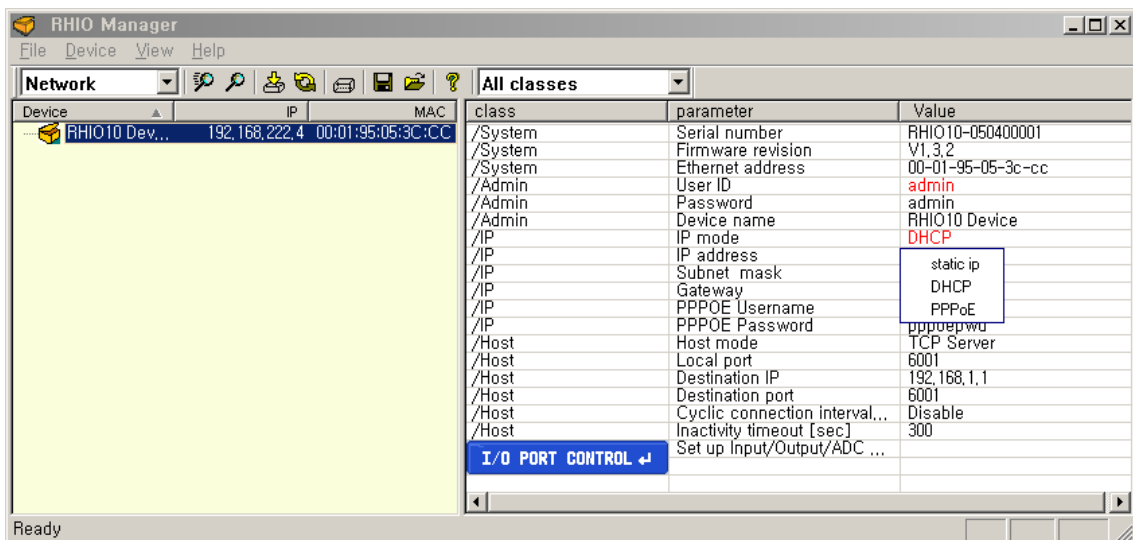


그림 3-1 Rhio10 네트워크 설정 화면

Rhio10의 공장 초기화 IP 모드 설정치는 DHCP 입니다. 최초로 Rhio10에 접속할 때는 먼저 Rhio Manager 의 [Probe] 기능을 이용하여 Rhio10 을 검색합니다. 사용자 네트워크 상에 DHCP 서버가 있는 경우, DHCP 서버로부터 부여 받은 IP 주소가 창에 표시됩니다. 만약 DHCP 서버가 없으면 0.0.0.0으로 IP 주소가 표시되며, 이 경우에는, Rhio10 의 IP 모드를 먼저 Static IP 모드로 변경한 후 사용자의 네트워크 환경을 감안한 IP 주소를 입력해야 합니다. Rhio10이 검색되고 난 후에는 접속하기 원하는 Rhio10을 클릭합니다. 이때, Rhio10 접속을 위해서는 사용자 ID 및 Password를 입력하여야 합니다.

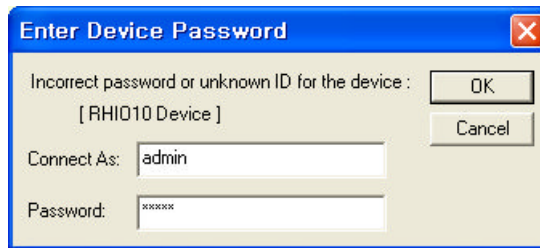


그림 3-2 패스워드 입력화면

ID와 Password가 정확하면 Rhio Manager의 Get/Set Screen에 Rhio10의 시스템 설정 값과 I/O Port 감시, 제어 및 설정을 위한 [I/O Control] 버튼이 나타납니다. 사용자는 현재 설정된 값을 확인할 수 있으며 원하는 설정 값으로 파라미터들을 변경할 수 있습니다. 변경된 값은 저장하기 전까지는 붉은 색으로 나타납니다.

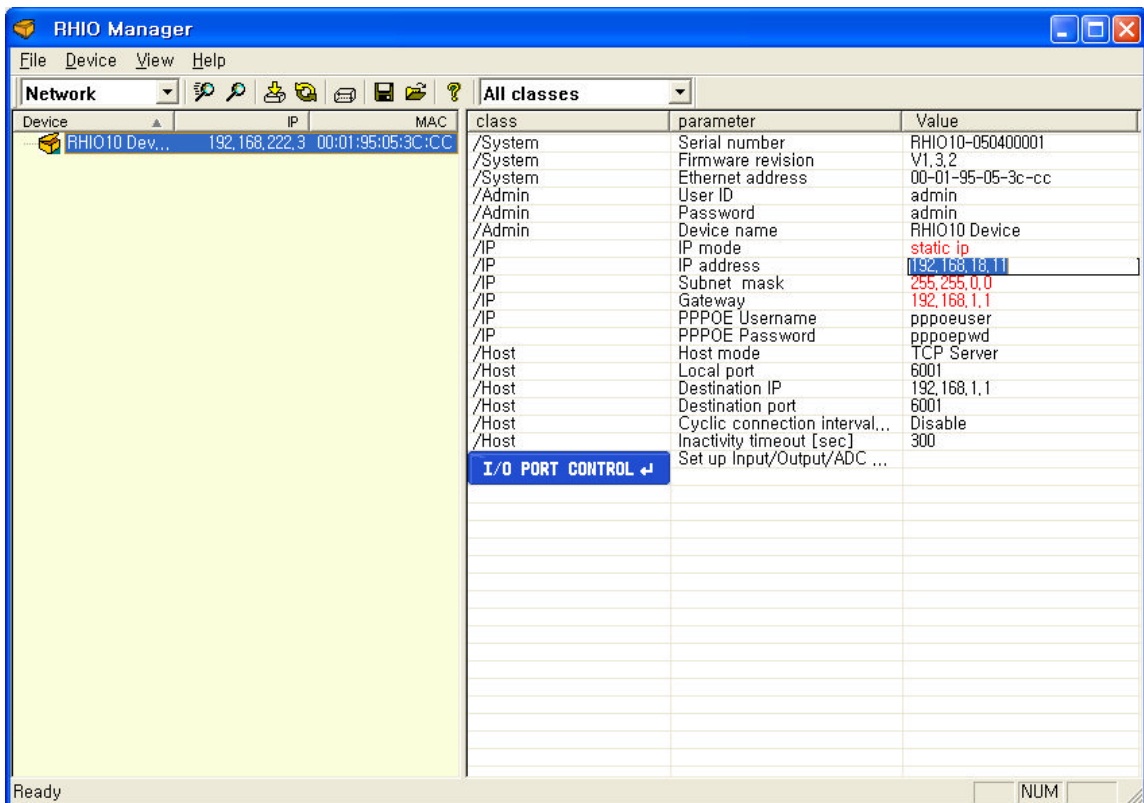


그림 3-3 네트워크 설정 예

Rhio10의 설정을 변경한 후, 설정치를 저장하고 적용하기 위해서는 반드시 [Set] 버튼을 클릭하여야 한다. [Set] 동작이 실행되면, Rhio10은 자동적으로 파라미터 저장/적용 후, Reboot 동작을 수행합니다. Reboot 완료 후, 다시 [Probe] 하여 새로운 설정으로 적용된 Rhio10의 파라미터를 확인합니다. 아래 그림은 설정이 적용된 후 다시 설정을 읽어온 화면입니다.

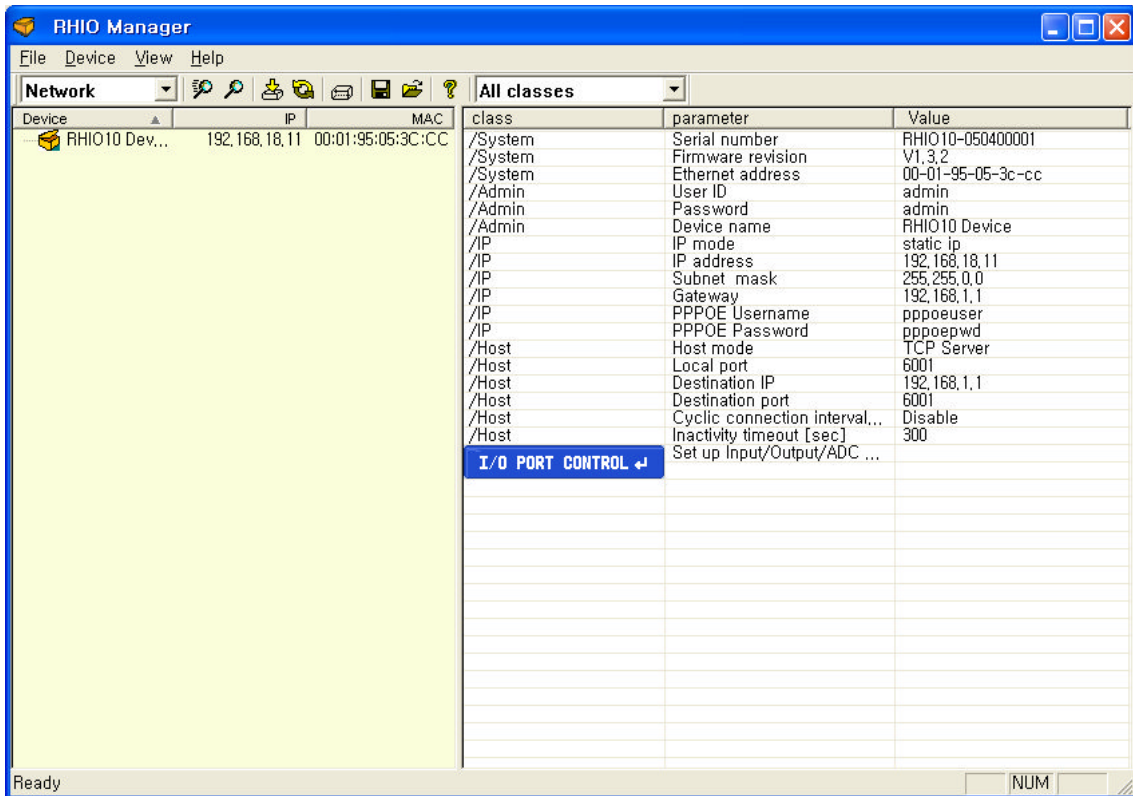


그림 3-4 네트워크 설정 완료 후 화면

### 3.1.1 Static IP

Static IP 모드에서 사용자는 Rhio10의 IP 주소, 유효한 서브넷 마스크 및 기본 게이트웨이 IP 주소와 같은 모든 파라미터를 수동으로 설정합니다.

IP 주소는 네트워크상의 영구적인 주소로서 컴퓨터에 할당된 식별 번호입니다. 컴퓨터들은 IP 주소를 이용하여 네트워크상에서 서로 구별하고 대화합니다. IP 주소는 네트워크 환경에서 고유하며 유효한 것으로 선택합니다.

서브넷은 같은 지리적 위치, 한 건물 또는 동일한 LAN상에 있는 모든 네트워크 호스트를 뜻합니다. 네트워크를 통해 나가는 패킷이 있는 경우 Rhio10은 패킷에 지정된 TCP/IP 호스

트가 로컬 네트워크 세그먼트에 있는지 서브넷 마스크를 통해 확인합니다. 주소가 Rho10과 동일한 네트워크 세그먼트에 있음이 판명되면 Rho10에서 직접 연결됩니다 그렇지 않으면 주어진 기본 게이트웨이를 통해 연결됩니다.

게이트웨이는 다른 네트워크로 들어가는 입구 역할을 하는 네트워크 접점입니다. 일반적으로 네트워크 내에서 또는 지역 ISP에서 트래픽을 제어하는 컴퓨터는 게이트웨이 노드입니다. Rho10이 로컬 네트워크 환경 밖의 호스트와 통신하기 위해서는 기본 게이트웨이 컴퓨터의 IP 주소를 알아야 합니다. 게이트웨이 IP 주소에 대한 정확한 정보는 네트워크 관리자에게 문의하십시오.

### 3.1.2 DHCP

동적 호스트 설정 통신 규약(DHCP)은 네트워크 관리자가 IP 주소의 할당을 조직의 네트워크에서 중앙 관리하고 자동화할 수 있게 하는 통신 프로토콜입니다. DHCP는 네트워크 관리자가 IP 주소를 중심점에서 관리하고 배포하게 하며, 컴퓨터가 네트워크의 다른 곳에 연결되면 새로운 IP 주소를 자동으로 보내게 합니다.

IP 주소는 Static IP 모드에서는 각 컴퓨터에 수동으로 입력해야 하고, 네트워크의 다른 부분의 다른 위치로 컴퓨터를 이동할 경우 새로운 IP 주소를 입력해야 합니다. 한편, IP 주소가 DHCP 모드에서 할당되면 IP 주소, 서브넷 마스크, 게이트웨이, DNS 서버를 포함하는 모든 파라미터가 자동으로 설정됩니다. DHCP는 임의의 IP 주소가 하나의 컴퓨터에 대하여 유효한 시간, 즉 "대여(lease)"의 개념을 사용합니다. IP 주소를 할당해야 하는 모든 파라미터는 DHCP 서버 쪽에서 설정하고, 각 DHCP 클라이언트 컴퓨터는 부팅시 IP 주소가 제공될 때 이 정보를 받습니다.

Rho10은 부팅할 때마다 IP 주소를 얻기 위해 해당 DHCP 요청을 네트워크에 브로드캐스트로 보냅니다. DHCP 서버가 생성한 응답에는 IP 주소를 비롯하여 서브넷 마스크, 게이트웨이 주소, DNS 서버 및 대여 시간이 포함되어 있습니다. Rho10은 이 정보를 비휘발성 메모리에 즉시 저장합니다. 대여 시간이 다 된 경우, Rho10은 DHCP 서버에 대여 시간 연장을 요청합니다. DHCP 서버가 대여 연장을 승인할 경우 Rho10은 계속해서 현재 IP 주소로 작동할 수 있습니다. 그렇지 않으면 Rho10은 DHCP 서버에 새로운 IP 주소를 요청하는 절차를 시작합니다.

DHCP 서버는 네트워크 관리자가 관리하고 있는 IP 주소 풀에서 IP 주소를 유동적으로 할당합니다. 따라서 DHCP 클라이언트, 즉 Rho10은 부팅할 때마다 다른 IP 주소를 받게 됩니다.. 이러한 상황에서 사용자가 Rho10의 IP 주소를 알려면 해당 IP 주소를 DHCP 서버 쪽에서 예약해야 합니다. DHCP 네트워크에서 IP 주소를 예약하려면 관리자는 Rho10의 바닥 라벨 스티커에 있는 MAC 주소가 필요합니다.

예) MAC=00:01:95:04:0c:a1

### 3.1.3 PPPoE

PPPoE는 Ethernet LAN상의 여러 컴퓨터 사용자를 보통의 가입자 구내 장치(Customer Premises Equipment)를 통해 원격 사이트에 연결하기 위한 규격입니다. (가입자 구내 장치(Customer Premises Equipment)는 모뎀 및 유사 장치를 일컫는 전화 회사 용어입니다.) PPPoE를 사용하면 사무실 또는 많은 사용자가 있는 건물에서 인터넷에 ADSL, 케이블 모뎀 또는 무선 연결을 할 수 있습니다. PPPoE는 보통 ADSL과 같은 광대역 인터넷 접속에 사용됩니다.

Rhio10이 PPPoE 모드에서 작동하려면 PPPoE 계정 및 ADSL 모뎀과 같은 PPPoE 접속용 장치가 있어야 합니다. Rhio10은 PPPoE 프로토콜을 제공하므로 ADSL 연결을 통해 인터넷상의 원격 호스트에 접근할 수 있습니다. PPPoE 계정의 사용자 이름과 비밀번호를 설정해야 합니다. IP 모드가 PPPoE로 설정되어 있는 경우 Rhio10은 매번 부팅 시 PPPoE 서버와 PPPoE 연결을 교섭합니다. 교섭하는 동안 IP 주소, 게이트웨이, 서브넷 마스크 및 DNS 서버와 같은 인터넷 연결에 필요한 정보가 수신됩니다. 연결이 되면 Rhio10은 연결을 가능한 한 오래 유지하려 합니다. 연결 해제가 감지되면 Rhio10은 새로운 연결을 요청하여 새로운 PPPoE 연결을 시도합니다.

### 3.1.4 IP 필터링

Rhio10은 IP 주소 기반 필터링 규칙을 이용하여 승인 권한이 없는 호스트가 Rhio10에 접근하는 것을 막을 수 있습니다. 사용자는 파라미터의 설정을 수정함으로써 다음과 같은 규칙을 만들 수 있습니다.

- 특정 IP 주소를 가진 호스트만이 Rhio10에 접근
- 특정 서브넷에 있는 호스트만이 Rhio10에 접근
- 모든 호스트가 Rhio10에 접근

사용자는 Telnet 콘솔, TCP/IP 접속 등의 접근을 제한하기 위해서 IP 필터링 규칙을 지정할 수 있습니다. Rhio10에 접근할 수 있는 호스트나 호스트의 그룹을 만들기 위해서는 반드시 접근이 허용된 IP 주소와 서브넷을 입력해야만 합니다. 원격 호스트상의 모든 사용자는 Rhio10에 접근하기 위해서 반드시 입력된 서브넷의 범위 안에 있어야만 합니다.

특정 호스트만이 Rhio10에 접근하도록 규칙을 변경하려면, 특정 호스트의 IP 주소와 255.255.255.255로 서브넷을 입력해야만 합니다.

모든 호스트가 Rhio10에 접근하도록 규칙을 변경하려면, IP 주소와 서브넷을 모두 0.0.0.0으로 설정합니다. 자세한 사항은 표 3-2를 참조하세요.

표 3-2 Input examples of allowed remote hosts

접근 가능 호스트	Input format	
	Base Host IP address	Subnet mask
Any host	0.0.0.0	0.0.0.0
192.168.1.120	192.168.1.120	255.255.255.255
192.168.1.1 ~ 192.168.1.254	192.168.1.0	255.255.255.0
192.168.0.1 ~ 192.168.255.254	192.168.0.0	255.255.0.0
192.168.1.1 ~ 192.168.1.126	192.168.1.0	255.255.255.128
192.168.1.129 ~ 192.168.1.254	192.168.1.128	255.255.255.128

### 3.2 호스트 모드 설정

호스트 모드는 Rhio10의 작동 세션 모드입니다. I/O 장치와 원격 호스트 사이의 데이터 통신을 위한 몇 개의 호스트 모드가 있습니다. TCP는 연결 지향 프로토콜이므로 서버, 클라이언트, 서버/클라이언트 모드 등이 제공됩니다. Rhio Manager를 이용하여 I/O를 감시, 제어할 경우에는 Host Mode 가 TCP Server 모드로 설정되어 있어야 합니다. 표 3-3는 호스트 모드에 대한 간략한 설명입니다. 출하 시 기본 호스트 모드는 TCP Server 입니다.

표 3-3 Rhio10 TCP/IP 세션 모드

모드	설명
TCP Server	Rhio10을 TCP 서버로 운용할 때 선택합니다. Rhio10은 TCP 연결 요청이 올 때까지 대기합니다. 이미 TCP 연결이 성립되어 있지 않은 경우 Rhio10이 요청을 수락하여 세션이 성립됩니다. 연결이 성립된 상태에서는 원격 호스트로부터 I/O 제어 명령 등을 수신하여 I/O 제어 동작을 할 수 있습니다. Rhio10은 하나의 TCP 세션을 지원하므로 이미 연결이 성립된 경우, 추가 TCP 연결 요청은 거부됩니다. 사용자가 원할 때마다 I/O 제어를 할 경우에 유용한 모드입니다.
TCP Client	Rhio10을 TCP 클라이언트로 운용할 때 선택합니다. Rhio10은 I/O 상태 등이 변경되거나 사전 지정된 타이머가 만료되면 Rhio10은 TCP Port를 통하여 원격 서버로의 TCP 연결을 시도합니다. 이들 간에 TCP 세션이 성립되면 Rhio10이 서버에 I/O 상태 데이터를 송신합니다. 세션이 연결된 동안에는 원격서버로부터 I/O 제어 명령 등을 수신할 수 있습니다. 그러나 Rhio10이 원격 서버로의 연결에 실패할 경우 I/O 상태 데이터는 폐기됩니다. I/O 상태가 바뀌는 상태를 모니터링하는 경우 유용한 모드입니다.

TCP Server /Client	대부분의 응용 프로그램에 사용되는 것이므로 어느 모드를 선택해야 할지 모르는 경우 선택합니다. 이 모드에서 Rhio10은 TCP 서버이자 동시에 클라이언트로 작동합니다. 연결이 성립되지 않은 경우 들어오는 연결이 모두 수락되어 I/O 제어가 가능하며 연결이 성립되지 않은 경우 I/O 상태가 바뀌면 원격 호스트에 연결되어 I/O 데이터를 주고 받습니다.
--------------------	---

각 호스트 모드 설정에 필요한 파라미터는 표 3-4에 요약되어 있습니다.

표 3-4 호스트 모드 설정 파라미터

TCP Server	Listening TCP port
	Inactivity timeout (sec)
TCP Client	Destination IP
	Destination TCP Port
	Cyclic connection Interval
	Inactivity timeout (sec)
TCP Server/Client	Listening TCP port
	Destination IP
	Destination TCP Port
	Cyclic connection Interval
	Inactivity timeout (sec)

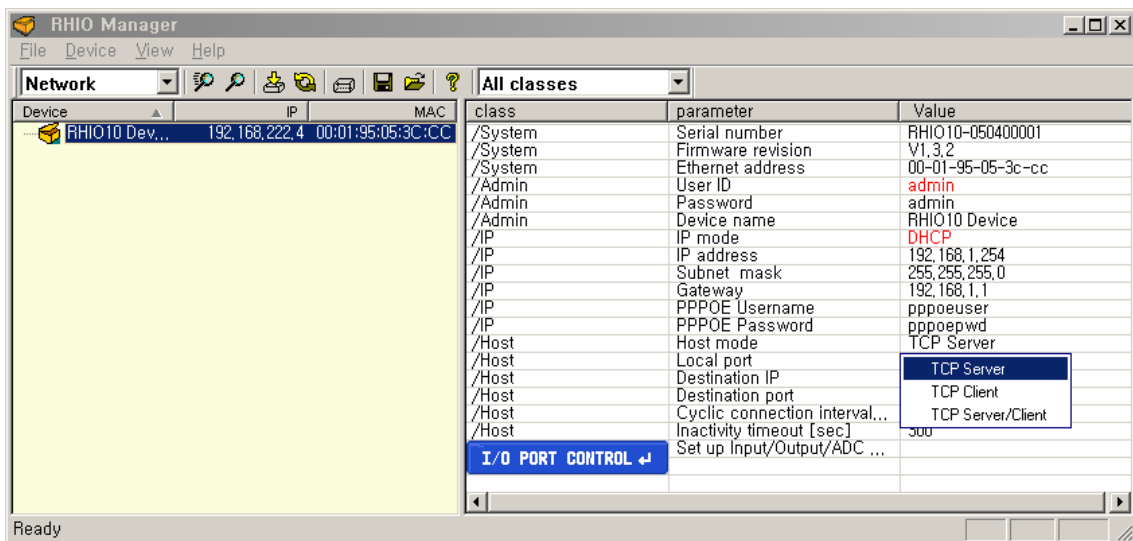


그림 3-5 Rhio10 호스트 모드 설정 화면

TCP 모드를 더 쉽게 설명하기 위해 흔히 간이 상태 전이도(State Transition Diagram)를 사용합니다. 또한 사용자가 전이도를 쉽게 이해할 수 있도록 Rhio10의 TCP 상태를 다음과 같이 간략하게 설명할 수 있습니다.

### **Listen**

"원격 호스트로부터의 연결 요청을 기다리는" 상태입니다. TCP 서버 모드로 설정되어 있는 경우 기본 시동 모드입니다. 이 상태는 TCP Server 모드 동작에서만 유효합니다.

### **Closed**

"연결되지 않은" 상태입니다. 데이터 전송이 완료된 경우 호스트 중에 하나가 연결 해제 요청을 보내면 이 상태로 변경됩니다. TCP Server 모드에 있는 경우 상태는 곧바로 [Listen] 모드로 자동 변경됩니다. TCP 클라이언트 모드나 TCP Server/Client 모드로 설정되어 있는 경우 기본 시동 모드입니다.

### **Sync-Received**

TCP server 모드에서는 외부 원격 호스트에서 연결 요청을 접수하면 상태는 [Listen]에서 [Sync-Received]로 변경됩니다. Rhio10이 요청을 수락하는 경우 상태는 [Established]로 변경됩니다. 이 상태는 TCP Client 모드에서는 유효하지 않습니다.

### **Sync-Sent**

Rhio10이 원격 호스트에 연결 요청을 보내면 [Closed]에서 [Sync-Sent] 상태로 변경됩니다. 이 상태는 원격 호스트가 연결 요청을 수락할 때까지 유지됩니다. 이 상태는 TCP Client 모드에서만 유효합니다.

### **Established**

"연결이 개설된" 상태입니다. 호스트 중 하나가 다른 호스트의 연결 요청을 수락한 경우 연결이 개설되며 상태는 [Established]로 변경됩니다.

### **Data**

[Established] 상태에 있으면 데이터가 호스트에서 다른 곳으로 옮겨집니다. TCP 세션 동작에 한 이해를 더 쉽게 하기 위해서 실제 데이터 전송이 수행되었을 때의 상태를 [Data] 상태라고 부릅니다. 사실상 [Data] 모드는 RFC 793 [Transmission Control Protocol]에 설명되어 있듯이 [Established] 상태의 일부입니다. 이는 연결의 데이터 전송 단계에 있어서 정상 상태입니다.



### 3.2.1 TCP Server 모드 동작

이 모드에서는 Rhio10이 TCP 서버로 작동하고 기본 TCP 상태는 [Listen]입니다. Rhio10은 TCP 소켓 연결 하나만을 지원합니다. 현재 연결이 성립되어 있는 경우 추가 연결 요청은 거부됩니다. 원격 호스트는 TCP 클라이언트 역할을 하는 소켓 프로그램이어야 합니다.

#### 1) 일반적인 상태 전이

[Listen] --> [Sync-Received] --> [Established] --> [Data] --> [Closed] --> [Listen]

시동 시 초기 TCP 상태는 [Listen]입니다. TCP 연결 요청이 들어오는 경우 상태는 [Sync-Received]로 변경된 후 [Established]로 변경되는데, 이는 세션이 오픈 되었다는 것을 의미합니다. 다음에는 연결된 호스트들 사이에서 데이터가 전송됩니다. 이 상태가 [Data] 상태입니다. 둘 중 하나의 요청으로 인해 세션이 끊겨지는데 이 상태가 [Closed] 상태입니다. 그 다음 상태는 본래의 상태, 즉 [Listen]으로 자동 변경됩니다.

#### 2) 동작

##### I/O 상태 데이터 전송

세션이 성립되면 Rhio10은 사용자의 명령에 따라 응답 데이터를 원격 호스트의 IP 주소로 전송합니다. Rhio10에 연결된 원격 호스트가 없는 경우, 변경된 I/O 상태의 데이터는 무시됩니다.

##### 세션 연결 해제

원격 호스트가 연결 해제 요청을 보내거나 일정 시간동안 I/O 데이터 전송 활동이 없어서 "Inactivity timeout"된 경우 세션의 연결이 해제됩니다.

그림 3-6은 TCP 서버 모드에서 세션 동작의 상태 전이도를 나타냅니다.

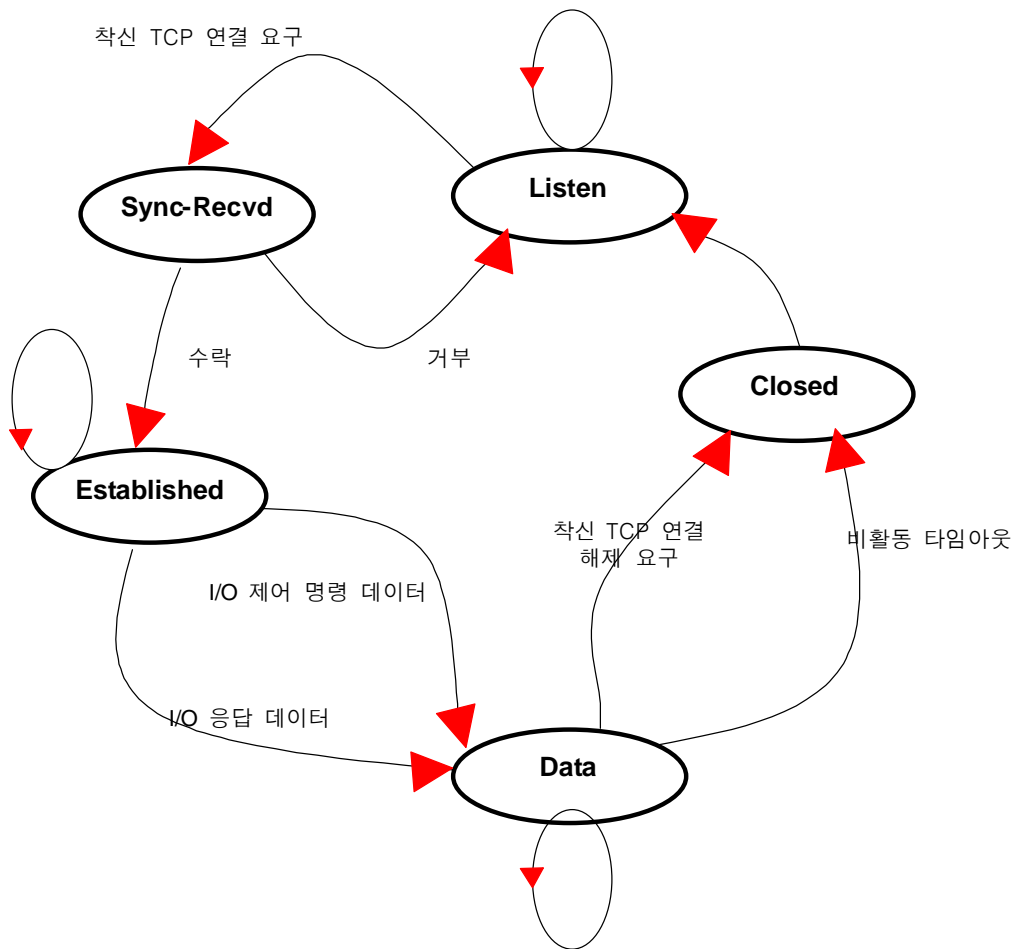


그림 3-6 TCP Server 모드 상태 전이도

**Listening TCP port**는 원격 호스트가 TCP 세션을 연결하고 데이터를 송수신할 수 있는 TCP Port 번호입니다. Listening TCP Port 이외의 Port로의 착신 연결 요청은 거부됩니다. Rhio10은 이 번호를 특정 범위로 제한하지는 않습니다. 그러나 일부 응용 프로그램용으로 잘 알려진 Port는 사용하지 않는 것이 좋습니다. (부록 D. 잘 알려진 Port 번호를 참조하십시오.) Port 번호를 변경하려면 TCP 서버 모드 설정 화면에서 2번 메뉴를 선택하십시오.

**Inactivity timeout**은 TCP host 모드에서 I/O 데이터 전송이 없는 경우 TCP 연결 상태를 Closed 또는 Listen으로 유지하기 위해 설정합니다. 지정된 Inactivity timeout 간격 동안 I/O 제어 명령이 들어오지 않거나 I/O상태 데이터가 발생하지 않는 경우 기존 TCP 연결은 자동으로 종료됩니다. Inactivity timeout 값을 0으로 설정할 경우, 연결 종료 요청이 없는 한 현재 TCP 연결이 유지됩니다. Inactivity timeout이 비활성화되어도 Rhio10은 "keep alive" 패킷을 주기적으로 보냄으로써 Rhio10과 원격 호스트 간의 연결 상태를 확인합니다. 원격 호스트가 패킷에 응답하지 않는 경우 실수로 연결이 끊어진 것으로 간주됩니다. 그러면 Rhio10은 강제로 기존 TCP 연결을 종료합니다.

### 3.2.2 TCP Client 모드 동작

이 모드에서는 Rhio10이 TCP 클라이언트로 작동하고, 기본 TCP 상태는 [Closed]입니다. 원격 호스트는 TCP 서버 역할을 하는 소켓 프로그램입니다.

#### 1) 일반적인 상태 전이

[Closed] --> [Sync-Sent] --> [Established] --> [Data] --> [Closed]

시동시 초기 TCP 상태는 [Closed]입니다. I/O 상태가 변경된 경우 Rhio10은 사용자 정의된 원격 호스트에 연결을 시도합니다. 이때 상태는 [Sync-Sent]로 변경되며, 이것은 연결 요청이 보내지고 있다는 것을 뜻합니다. 원격 호스트가 요청을 수락할 경우 상태는 [Established]로 변경되며, 이는 세션이 오픈 되었다는 것을 의미합니다. 다음에는 연결된 호스트들 사이에서 I/O 상태 데이터가 전송됩니다. 이는 [Data] 상태입니다. 둘 중 하나의 요청으로 인해 세션이 해제되면 이는 본래 상태인 [Closed]입니다.

#### 2) 동작

##### I/O 상태 데이터 전송

I/O 상태가 변경되면 Rhio10은 TCP 세션이 아직 성립되지 않았을 때 원격 호스트의 사용자 정의 IP 주소에 연결을 시도합니다. Rhio10이 원격 호스트에 성공적으로 연결된 경우 I/O 상태 데이터는 호스트로 이동됩니다. 그렇지 않으면 I/O 상태 데이터가 삭제됩니다.

##### 세션 연결 해제

원격 호스트가 연결 해제 요청을 보내거나 일정 시간 동안 호스트와 Rhio10간의 데이터 전송 활동이 없어서 "Inactivity timeout"된 경우 세션의 연결이 해제됩니다.

##### 원격 호스트의 연결 요청

TCP 클라이언트 모드에서는 들어오는 모든 TCP 연결 요청이 거부됩니다.

##### 순환 연결

Cyclic Connection 기능이 활성화된 경우 Rhio10은 I/O 상태가 변경되지 않아도 사용자 정의된 원격 호스트에 일정한 시간 간격으로 연결을 시도합니다. 원격 호스트가 일정한 데이터를 준비하고 이를 연결될 때마다 전송하는 경우, I/O 제어 명령이 전송됩니다. 결국 사용자는 Rhio10이 원격 호스트에 연결할 때마다 원격 호스트가 Rhio10으로 I/O 제어 명령을 보내게 하여 Rhio10의 I/O 상태를 주기적으로 감시하게 됩니다. 이 옵션은 I/O 상태가 바뀌지 않아도 장치 정보를 주기적으로 수집 할 필요가 있을 때 유용합니다. 그림 3-7은 TCP Client 모드에서 세션 동작의 상태 전이도를 나타냅니다.

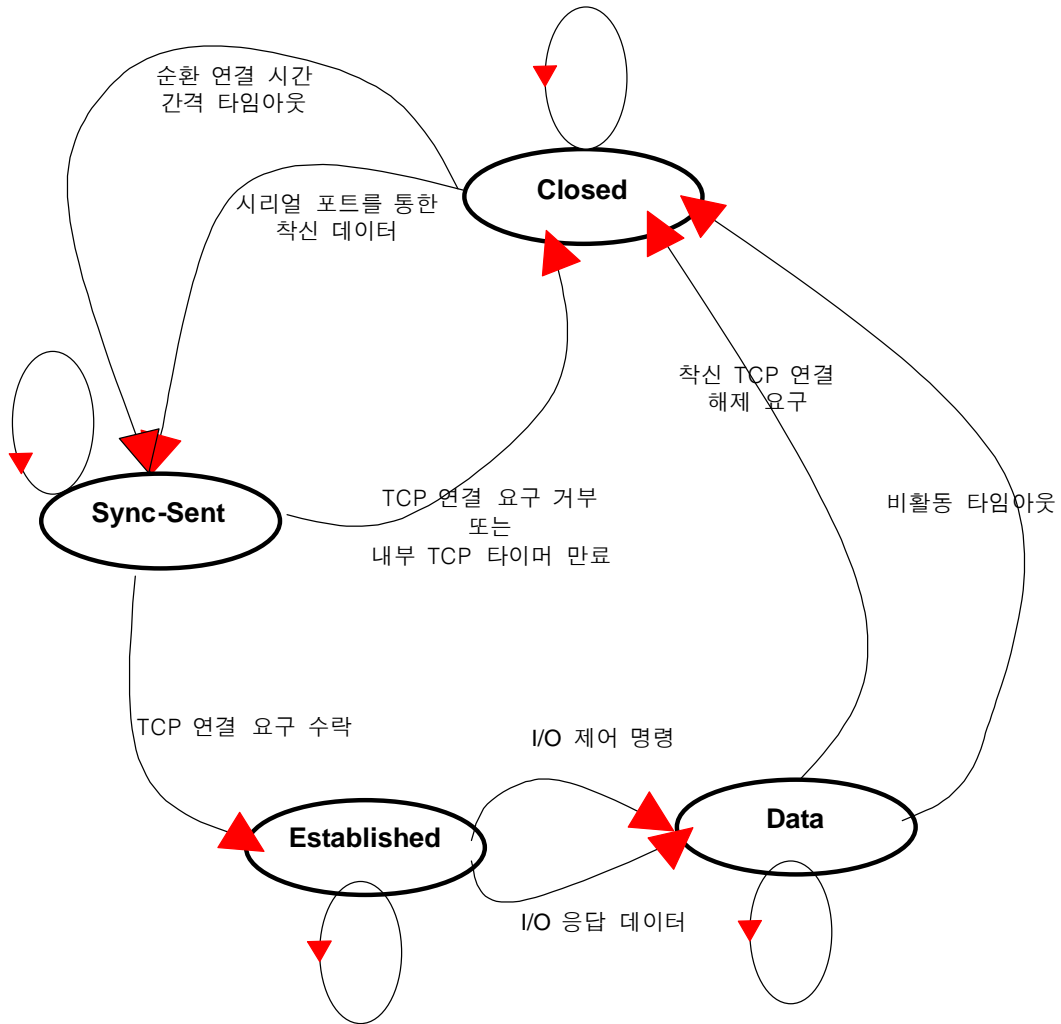


그림 3-7 TCP 클라이언트 모드의 상태 전이도

목적지 IP 주소와 목적지 TCP Port는 Rhio10이 TCP client 모드에서 연결을 시도할 원격 호스트의 정보입니다. IP 주소(또는 도메인 네임)는 TCP Port 번호와 함께 기입해야 합니다. 순환 연결 시간 간격은 Rhio10이 I/O 상태 변화 유무와 상관없이 원격 호스트에 연결을 시도하는 시간 간격입니다. 0을 제외한 유효값으로 시간 간격을 지정하면 이 기능이 활성화됩니다. 시간 간격은 분 단위로 지정됩니다. 시간 간격을 0으로 입력할 경우, 순환 연결 기능이 비활성화 됩니다

Inactivity timeout은 TCP 서버 모드 설정 절에서 설명한 것과 동일합니다

### 3.2.3 TCP Server/Client 모드 동작

Rhio10은 상황에 따라 TCP 서버 또는 클라이언트로 작동합니다. 대부분의 응용 프로그램의 경우 이것이 일반적인 모드입니다. 기본 TCP 상태는 TCP Server 모드와 동일한 [Listen]입니다.

#### 1) 일반적인 상태 전이

[Listen] --> [Sync-Received] --> [Established] --> [Data] --> [Closed] --> [Listen]

Or

[Listen] --> [Sync-Sent] --> [Established] --> [Data] --> [Closed] --> [Listen]

초기 상태는 [Listen]입니다. I/O 상태 변화가 있으면 원격 호스트에 TCP 클라이언트로서 연결합니다. 원격 호스트로부터 착신 연결 요청이 있으면 TCP 서버로서 연결을 수락한 후 데이터를 전송합니다. 따라서 사용자는 Rhio10이 항상 지정된 원격 호스트에 연결되었다고 생각할 수 있습니다.

#### 2) 동작

TCP Server 모드와의 유일한 차이점은 TCP 세션이 성립되지 않더라도 Rhio10이 원격 호스트에 연결을 시도하여 I/O 상태변화 데이터를 보낸다는 점입니다. TCP Client 모드와의 차이점은 세션이 성립되지 않더라도 원격 호스트의 착신 연결 요청을 수락하는 것입니다. 세부적인 작동 원리는 TCP Server 및 TCP Client 모드와 동일합니다.

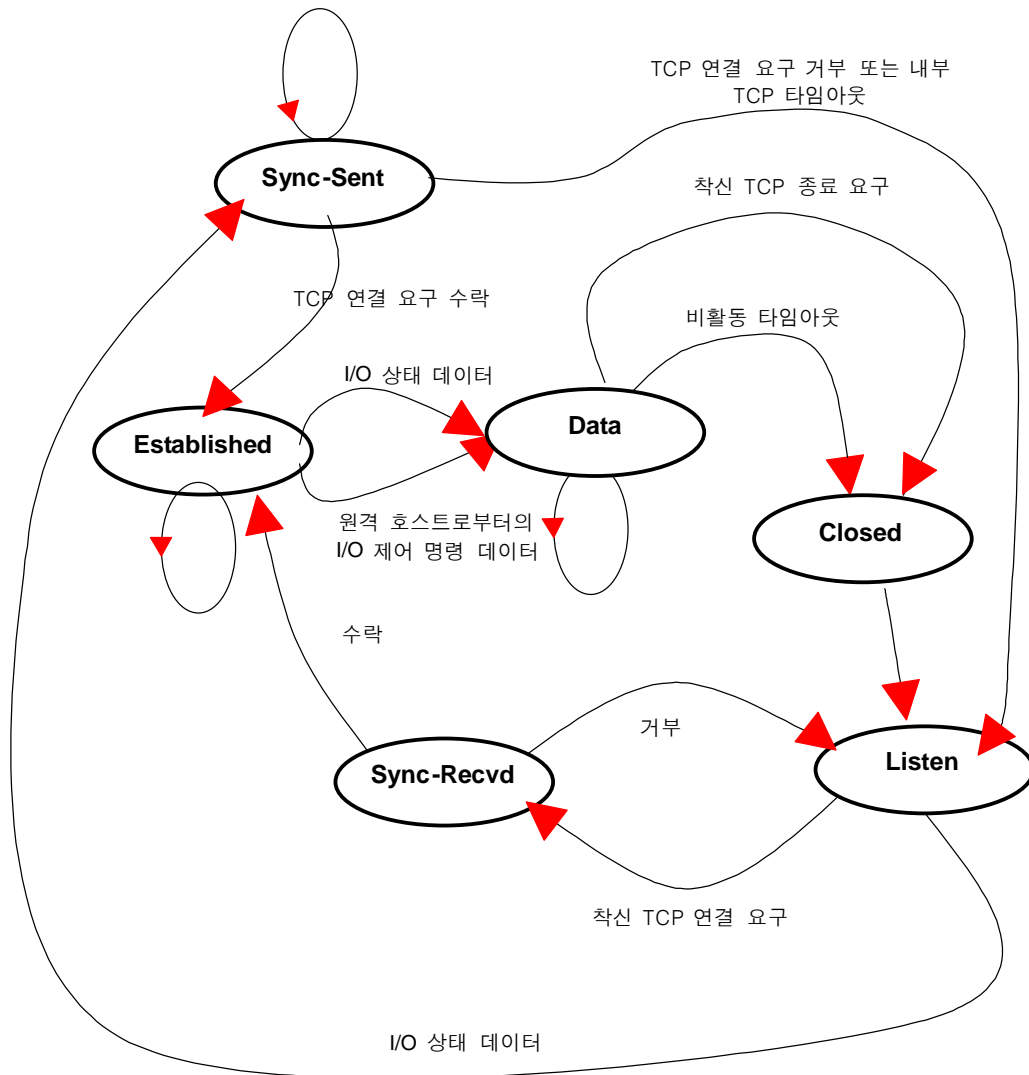


그림 3-8 TCP 서버/클라이언트 모드의 상태 전이도

### 3.3 시스템 관리

사용자는 Rhio Manager의 admin 항목에서 ID/Password 및 Device Name을 설정할 수 있습니다. ID와 Password를 수정한 후에는 [Clear] 버튼을 눌러야 Rhio Manager에 저장된 기존의 ID와 Password가 초기화됩니다. Device Name은 여러 개의 Rhio10 이 연결된 경우 각 장비를 구별하기 위한 장비 이름을 나타냅니다.

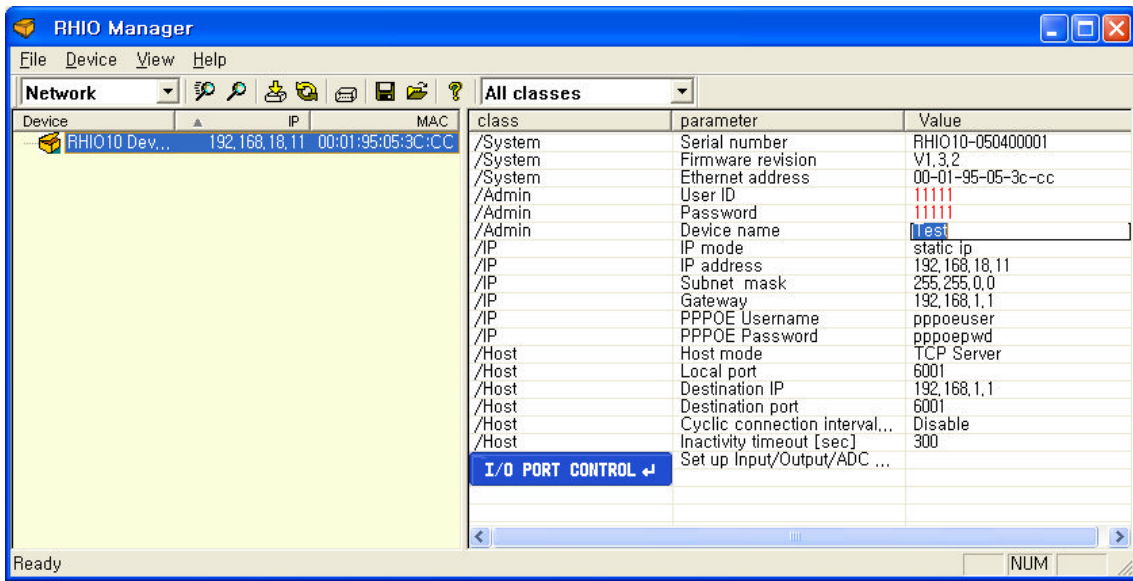


그림 3-9 시스템 관리 설정

### 3.4 공장 출하 시 기본값 복구

사용자는 Rhio10의 Factory reset 스위치를 눌러 공장 출하 시의 기본값으로 설정을 복구할 수 있습니다. Factory Reset 스위치는 Rhio10 제품 좌측면에 위치해 있으며 핀셋 등 뾰족한 기구를 이용하여 약 1초 정도 누르고 있으면 작동 후 설정치가 초기화되며 Rhio10이 Reboot 됩니다.

공장 출하 시 기본값은 아래와 같습니다.

IP Mode:	DHCP
Host Mode:	TCP Server, TCP port 6001
ID / Password:	admin / admin
Device Name:	Rhio10 Device
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

## 4. I/O 설정 및 응용

### 4.1 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 TCP 소켓 접속을 통해 Rhio10의 Local port로 접속하여 I/O 상태의 감시를 시작합니다. 이때 Rhio10의 호스트 모드는 TCP Server 모드로 동작하도록 설정하여야 합니다.

#### 4.1.1 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

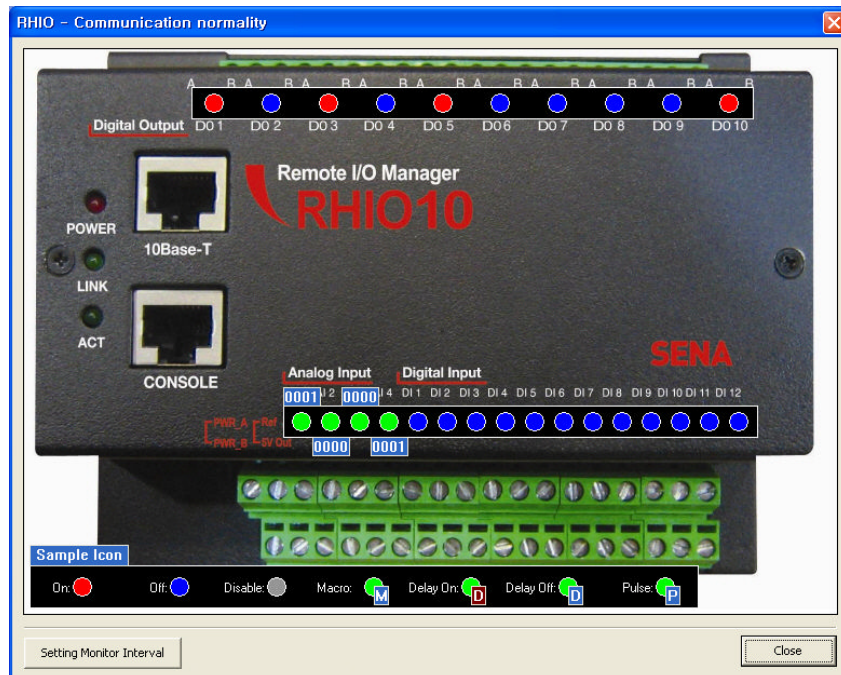


그림 4-1 I/O port Control 화면



### 4.1.2 감시 주기 설정

사용자는 [Setting Monitor Interval] 을 설정하여 Rhio10의 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 Rhio10으로부터 응답을 받아 화면에 표시합니다.

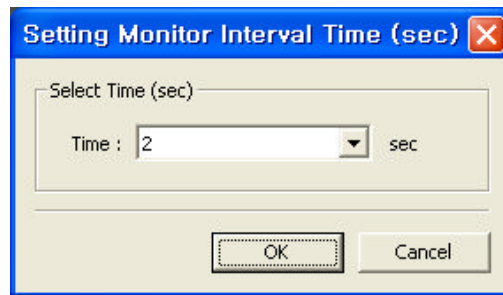


그림 4-2 Setting Monitoring Interval

### 4.1.3 I/O Port 감시 동작

Rhio10이 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.

### 4.1.4 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 일치

하지 않으면 실행 대기 상태로 표시됩니다.

Output 제어 동작이 완료되면 Rhio10은 전체 Port의 상태를 호스트 컴퓨터에 회신 합니다.

## 4.2 Digital Input 설정

Rhio10 에는 12개의 Digital Input 단자가 존재합니다. 사용자는 각각의 Digital Input Port를 Enable/Disable 할 수 있습니다. Digital Input Port에 대한 설정은 Rhio Manager를 이용하거나 5장의 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.

### 4.2.1 Enable/Disable 설정

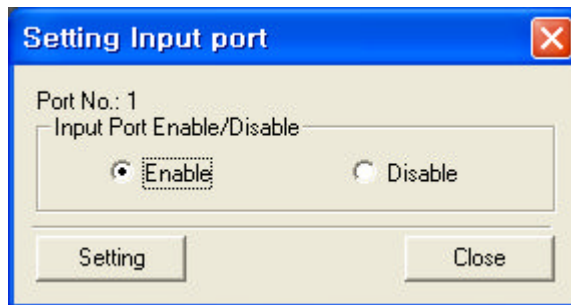


그림 4-3 Digital Input 설정

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.

### 4.3 Digital Output Port 설정

Rhio10은 10개의 Digital Output 단자가 장착되어 있습니다. Digital Output Port에 대한 설정은 Rhio Manager를 이용하거나 5장의 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.

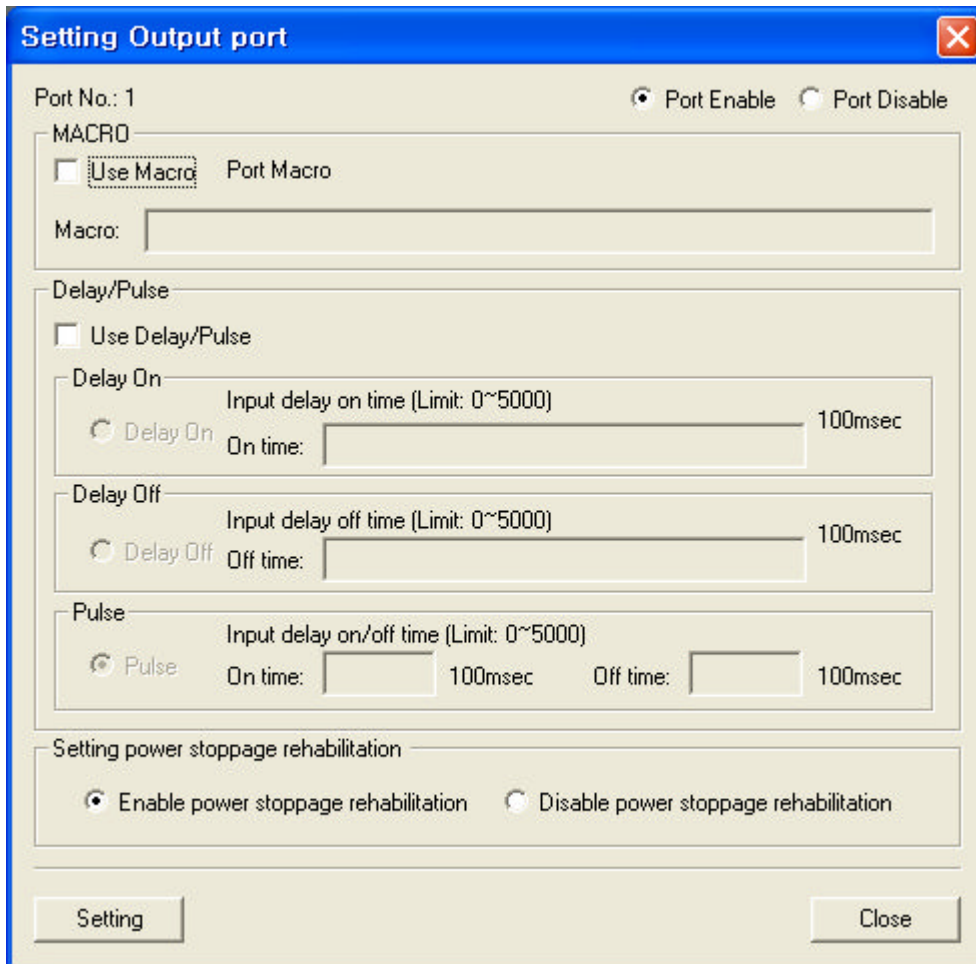


그림 4-4 Digital Output Port 설정창

#### 4.3.1 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.

### 4.3.2 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.

### 4.3.3 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.

### 4.3.4 정전 후 복구 설정

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

정전복구 Enable 시에는 Rhio10의 전원이 꺼졌다가 다시 켜진 경우 꺼지기 직전의Output Port 상태를 유지합니다.

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

정전복구 Disable 시에는 Rhio10의 전원이 꺼졌다가 다시 켜진 경우 Output Port의 상태는 OFF 됩니다.

## 4.4 ADC Input Port 설정

Rhio10에는 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 5장을 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.

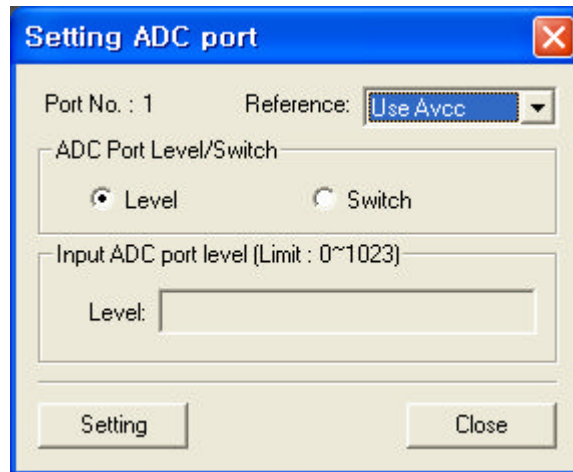


그림 4-5 ADC Input Port 설정

#### 4.4.1 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.

#### 4.4.2 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.

## 4.5 I/O Port 연결 방법

본 장에서는 Rhio10의 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.

### 4.5.1 Digital Output Port

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

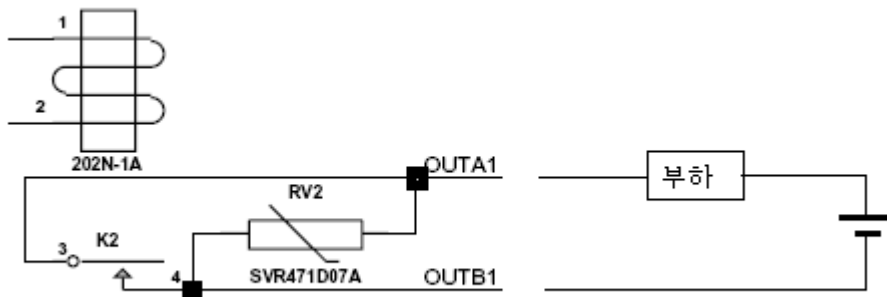


그림 4-6 Digital Output Port 연결

### 4.5.2 Digital Input Port

Digital Input 시스템은 전압의  $\pm$  극성에 상관 없이 동작하며 시스템의 내부 회로와는 절연되어 있습니다. 그림 4-7의 회로와 같이 연결되어 사용할 수 있으며 각각의 Input에 대하여 상태 확인용 LED가 있어 전압 Input 시에 점등됩니다

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.

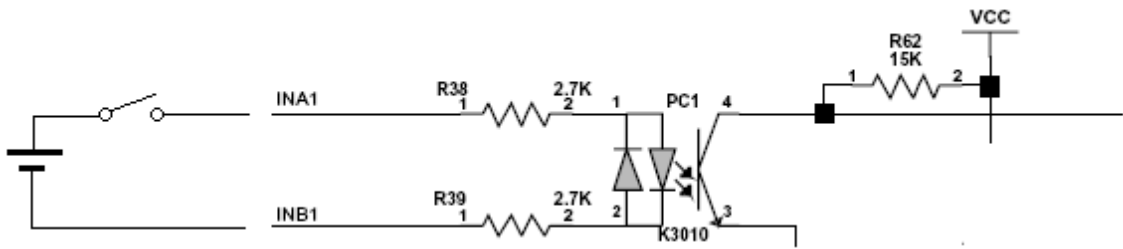


그림 4-7 Digital Input Port 연결

### 4.5.3 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

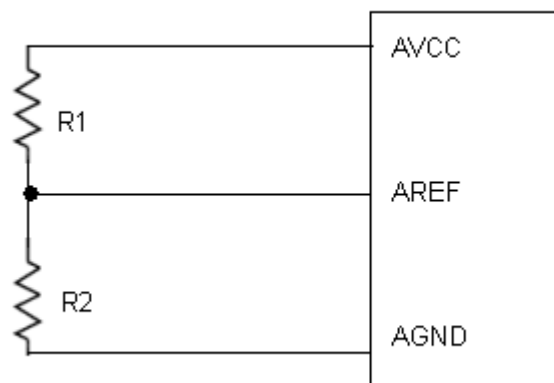


그림 4-8 Aref 의 연결방법



## 2) Analog Input Potentiometer 사용 시 방법

Analog Input은 입력되는 전압을 기준으로 동작하며 Input 임피던스는 100k $\Omega$  입니다. 그림 4-9와 같이 Potentiometer를 사용할 경우에는 1k $\Omega$  ~ 5k $\Omega$  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

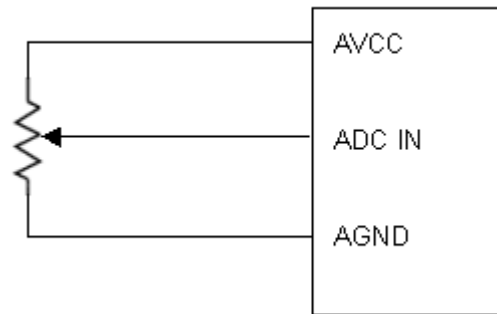


그림 4-9 Potentionmeter 사용시의 연결

## 3) AVCC 전압보다 높은 전압을 사용할 때의 연결방법

그림 4-10과 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.

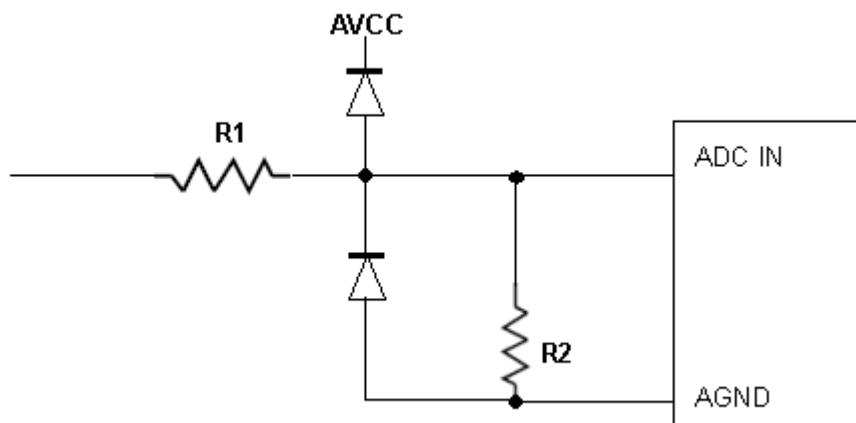


그림 4-10 Avcc 전압보다 높은 전압사용시의 연결

## 5. 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 Rhio10의 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.

## 5.1 Rhio 라이브러리

### 5.1.1 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.

## 5.1.2 Reference

Enumeration, Structure, Function의 정의는 RHIO\_Process.h를 참조 하십시오.

### Enumeration (부록 D.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 D.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 D.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.

## 5.2 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.

### 5.2.1 프로그램 UI 구성 및 관련 클래스

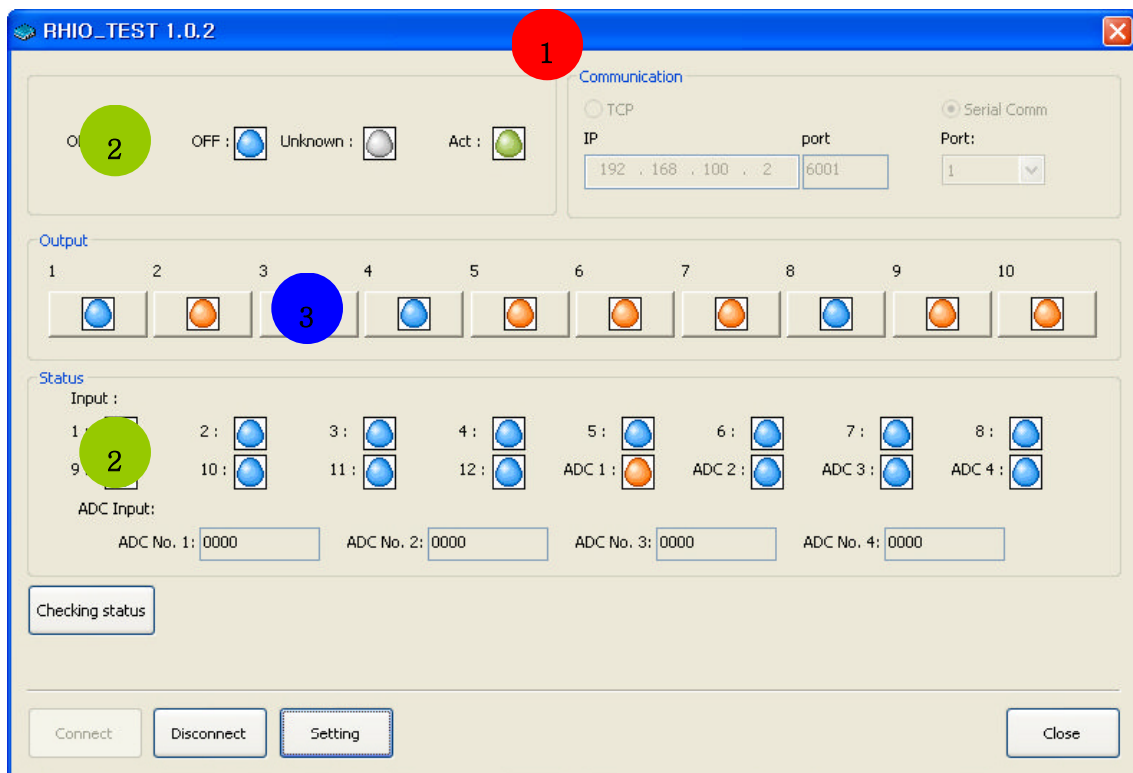


그림 5-1 샘플 프로그램 메인 윈도우

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

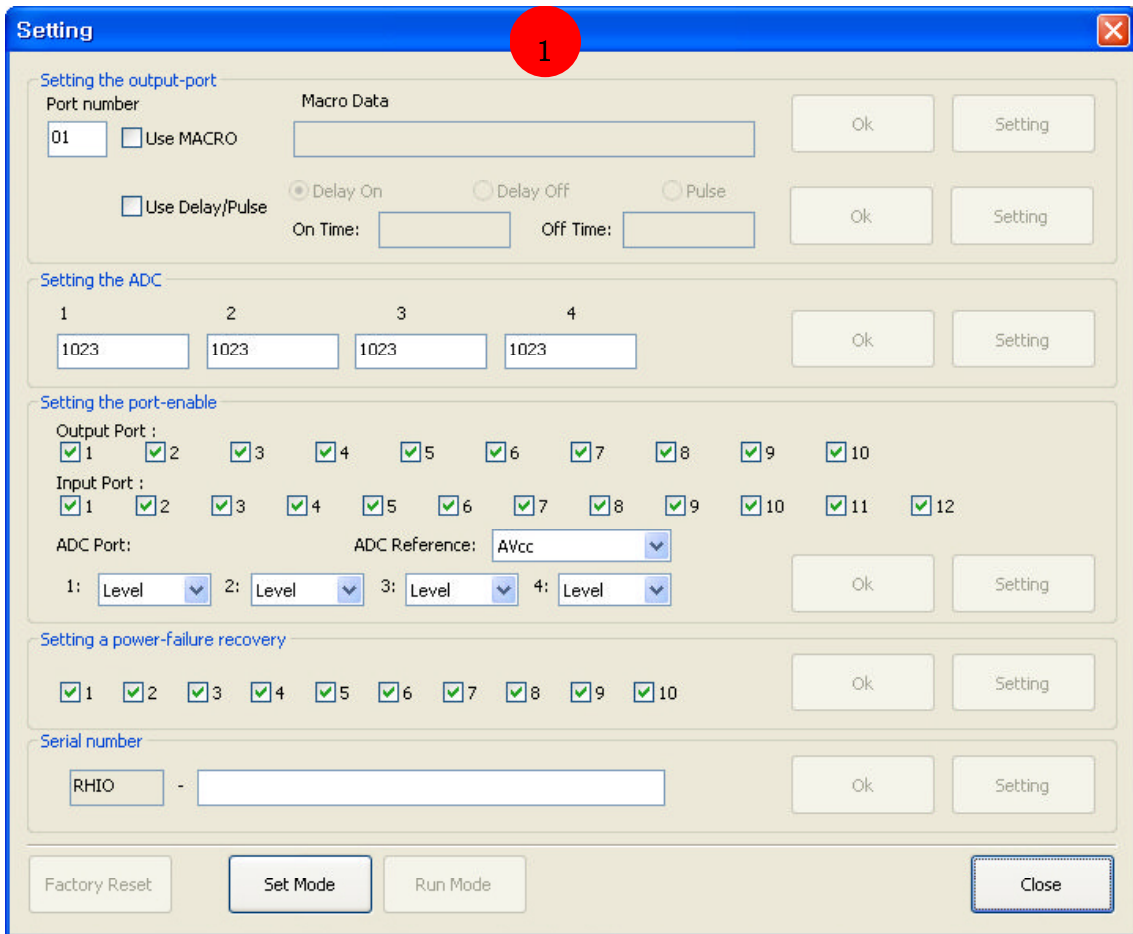


그림 5-2 샘플 프로그램 설정 윈도우

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

## 5.2.2 처리 절차

### 5.2.2.1 메인 윈도우 초기화

라이브러리를 링크하고, 라이브러리 함수의 주소를 얻어온다.

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

메인 윈도우의 다이얼로그 아이템을 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 Class**를 생성한다.

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

#### 5.2.2.2 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)`



## 5.3 Rhio 통신 프로토콜

이 장에서는 사용자가 직접 Rhio 통신 프로그램을 작성할 경우 사용자가 참고할 Rhio10의 통신 프로토콜에 대해 설명합니다.

### 5.3.1 개요

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

#### 5.3.1.1 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
	전송BLOCK의 시작을 표시한다. 0x3A (":")	FUNCTION에서 LRC 앞까지의 길이	Command Response	Data	START FLAG부터 DATA까지 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 DATA 까지의 data들을 1 Byte단위로 XOR한 값입니다.

(아래의 예제를 참조)

- 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, Rhio10에서 응답이 없으면 최소 5초 동안 Response를 대기하고, 다음 Command를 전송해야 합니다.)

\* LRC 값 계산 예제 :

LRC 값 계산(Start Flag에서 Data 까지의 XOR)의 예를 보여주기 위하여 다음과 같은 임의의 전송 블록을 사용합니다. :

(3A) (30 33) (30 33) (30) **(30 41)** (0D 0A)

LRC 계산 방식은 다음과 같습니다. :

$((((3A \text{ xor } 30) \text{ xor } 33) \text{ xor } 30) \text{ xor } 33) \text{ xor } 30 = 0A$

0A는 0과 A 라는 두 바이트의 값으로 변환 되며, 이 값에 대한 바이트 변환은 다음과 같습니다.

0x0 ~ 0x9 : 0x30 ~ 0x39 로 변환됩니다.

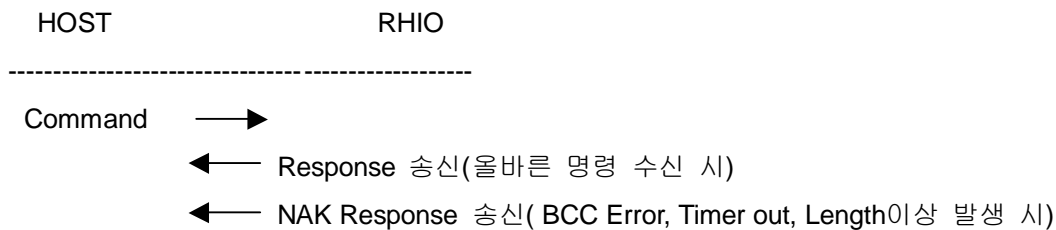
0xA ~ 0xF : 0x41 ~ 0x46 로 변환됩니다.

위의 설명에 의하여 0 =30, A=41 로 변환되며, LRC 값은 30 41 이 됩니다.

### 5.3.1.2 NAK Response

- NAK Response 조건
- 수신된 Data의 BCC Error
- HOST에서 송신한 Data가 수신이 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
	전 송BLOCK의 시작을 표시한다. 0x3A (":")	FUNCTION 에서 LRC 앞 까지의 길이	Response "00" 0x30,0x30	NAK Data "NAK" 0x4E,0x41, 0x4B	START FLAG부터 DATA까지 XOR한 값	CR (0x0D) LF (0x0A)



- 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

### 5.3.2 ON/OFF 제어

#### 5.3.2.1 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

#### 5.3.2.2 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 : 지연OFF 대기, 0x35:PULSE 동작	
	45	-	Field 구분 0x2C (“,”)	
	46~49	O5~O8	0x30 : OFF, 0x31 : ON, 0x32 : 조건실행대기, 0x33 : 지연ON대기, 0x34 : 지연OFF 대기, 0x35:PULSE 동작	
	50	-	Field 구분 0x2C (“,”)	
	51,52	O9,O10	0x30 : OFF, 0x31 : ON, 0x32 : 조건실행대기 0x33 : 지연ON대기, 0x34 : 지연OFF 대기, 0x35:PULSE 동작	

### 5.3.3 Input/Output 상태

#### 5.3.3.1 I/O 상태 요구 Command

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

#### 5.3.3.2 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.

### 5.3.4 설정/동작

#### 5.3.4.1 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.

### 5.3.4.2 설정/동작 Response

5.3.2.2 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.

### 5.3.5 Output Port 설정

#### 5.3.5.1 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
"50000"	"00000"	5000sec지연 후 ON
"00000"	"00001"	100ms지연 후 OFF
"00000"	"00020"	2sec지연 후 OFF



"0000"	"50000"	5000sec지연 후 OFF
"00001"	"00001"	100ms ON, 100ms OFF 반복
"00020"	"00020"	2sec ON, 2sec OFF 반복
"50000"	"50000"	5000sec ON, 5000sec OFF 반복
"00010"	"00030"	1sec ON, 3sec OFF 반복
"00300"	"00150"	30sec ON, 15sec OFF 반복

### 5.3.5.2 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.

### 5.3.5.3 Output Port 설정 확인 Command

	Command	Data		비 고
Byte Size	2	2	1	
	"07" ( 0x30,0x37 ) 설정확인	Port번호 "01" ~ "10"	0x31 : 동작조건설정 확인 0x33 : 지연, Pulse동작 설정 확인	개별 Output Port의 속성을 확인한다.

### 5.3.5.4 Output Port 설정 확인 Response

5.3.5.2 항목의 Output Port 설정 Response와 동일합니다.

## 5.3.6 ADC Input Port 설정

### 5.3.6.1 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"

### 5.3.6.2 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

### 5.3.6.3 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

### 5.3.6.4 ADC Input Port 설정 확인 Response

5.3.6.2 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.

## 5.3.7 Port Enable 설정

### 5.3.7.1 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.

### 5.3.7.2 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	Port구분	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

### 5.3.7.3 Port Enable 설정 확인 Command

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

### 5.3.7.4 Port Enable 설정 확인 Response

5.3.7.2 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)

## 5.3.8 Port 정전 후 복구 설정

### 5.3.8.1 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

### 5.3.8.2 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

### 5.3.8.3 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

#### **5.3.8.4 Port 정전 복구 설정 확인 Response**

5.3.8.2 정전 복구 설정 Response와 동일합니다.

회신 시 설정 상태 field는 정상(0x30)으로 회신 합니다.

공장 출하 시 초기값은 모두 Enable 상태입니다.

## 부록 A. 연결

### A.1 Ethernet Pin Outs

Rhio10은 AT&T258 규격을 따르는 차폐 커넥터인 표준 Ethernet 커넥터를 사용합니다. 표 A-1은 핀 지정 및 전선 색깔을 나타냅니다.

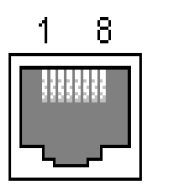


그림 A-1 RJ45 커넥터의 핀 배치

표 A-1 RJ45 커넥터의 핀 지정

핀	설명	색깔
1	Tx+	주황색과 흰색
2	Tx-	주황색
3	Rx+	녹색과 흰색
4	NC	청색
5	NC	청색과 흰색
6	Rx-	녹색
7	NC	갈색과 흰색
8	NC	갈색

### A.2 콘솔 Port Pin Outs

Rhio10 RJ45 커넥터의 핀 지정은 표 A-2에 요약되어 있습니다. 각 핀에는 시리얼 통신 유형 설정에 따른 기능이 있습니다.

주의 : 콘솔 Port와 호스트의 연결 시 CAT5 케이블(패키지에 포함)의 한쪽 끝에 RJ45 to DB9 Female 커넥터(패키지에 포함)를 끼워서 사용합니다.

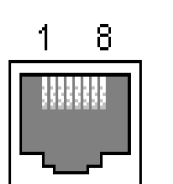


그림 A-2 RJ45 커넥터의 핀 배치



표 A-2 RJ45 커넥터의 핀 지정

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

### A.3 Ethernet 배선도

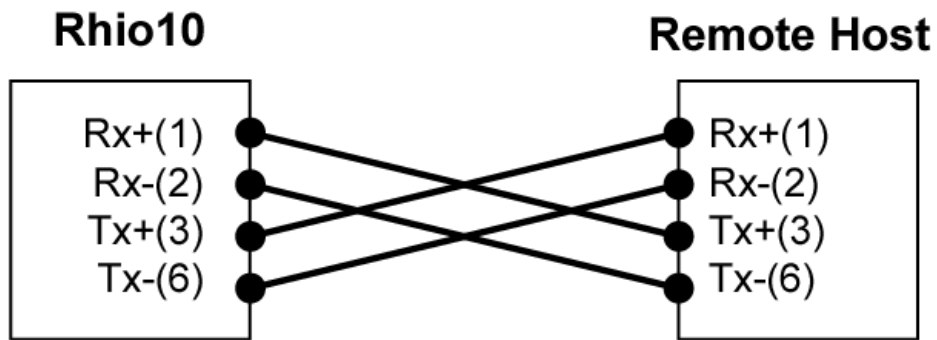


그림 A-3 크로스오버 Ethernet 케이블을 사용한 Ethernet 직접 연결

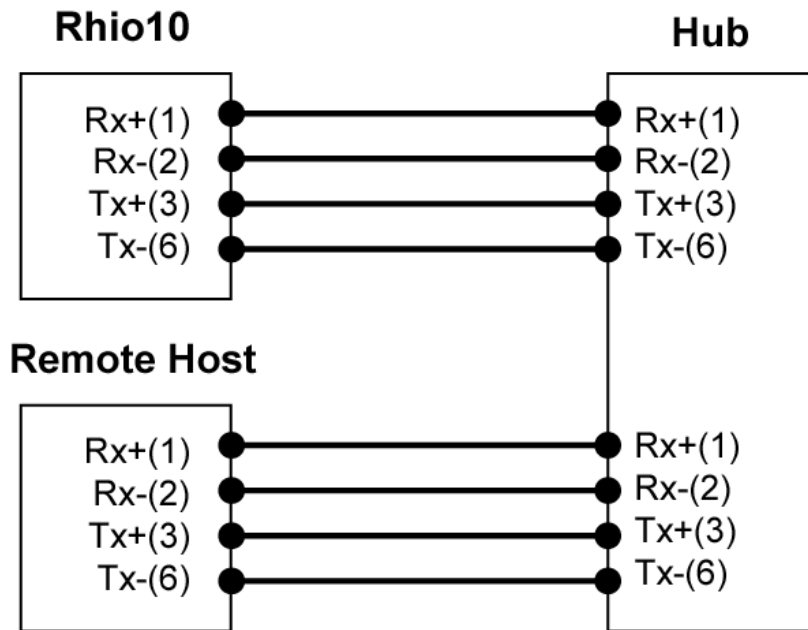


그림 A-4 스트레이트 Ethernet 케이블을 사용한 Ethernet 연결

#### A.4 시리얼 콘솔 배선도

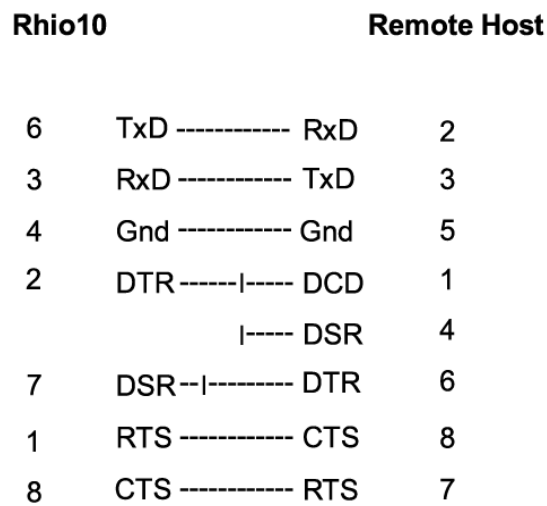


그림 A-5 RS232 배선도

## 부록 B. Rhio 콘솔 사용법 및 콘솔 명령어 일람

### B.1 콘솔 Port 접근

로컬 사이트와 원격 사이트 중 사용자가 위치하는 곳에 따라 Rhio10의 콘솔 Port에 접근하는 방법은 두 가지가 있습니다.

- 시리얼 콘솔: 로컬 사용자는 시리얼 콘솔 케이블을 사용하여 Rhio10의 시리얼 콘솔 Port에 직접 연결할 수 있습니다. Rhio10의 콘솔 Port는 기본적인 네트워크 설정등에 사용됩니다. 콘솔 Port로 사용하려면 데이터/콘솔 스위치를 콘솔쪽으로 합니다.

주의 : 콘솔 Port와 호스트의 연결시 CAT5 케이블(패키지에 포함)의 한쪽 끝에 RJ45 to DB9 Female 커넥터(패키지에 포함)를 끼워서 사용합니다.

- 원격 콘솔: 원격 사용자는 TCP/IP 네트워크를 통해 Rhio10의 원격 콘솔 Port( Port 23)에 telnet 연결을 할 수 있습니다.

어느 방법을 사용하든 계속하려면 Rhio10에 로그인해야 합니다.

#### B.1.1 시리얼 콘솔 사용

시리얼 콘솔 케이블의 한쪽 끝을 Rhio10의 콘솔 Port에 연결합니다.

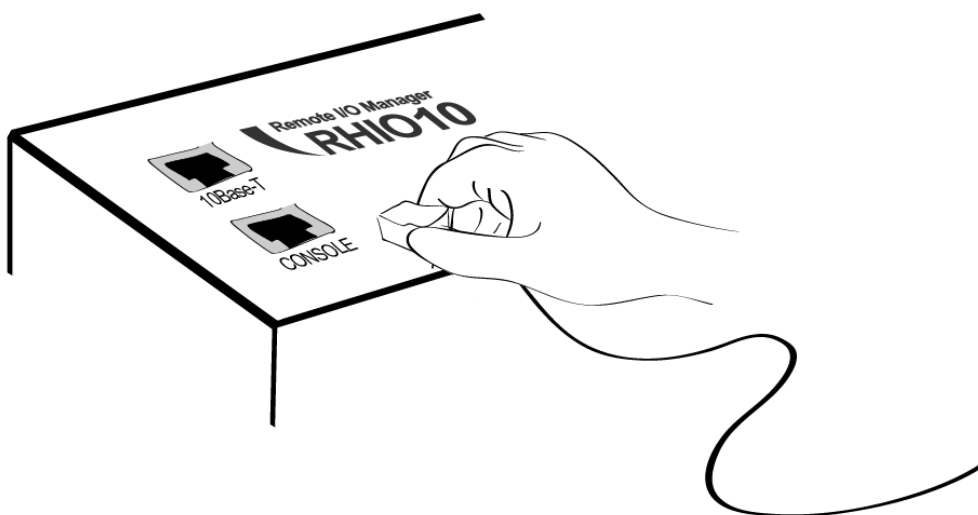


그림 B-1 Rhio10에 시리얼 콘솔 케이블 연결

케이블의 반대쪽 끝을 사용자 컴퓨터의 시리얼 Port에 연결합니다.  
 데이터/콘솔 스위치를 콘솔 쪽으로 맞춥니다.

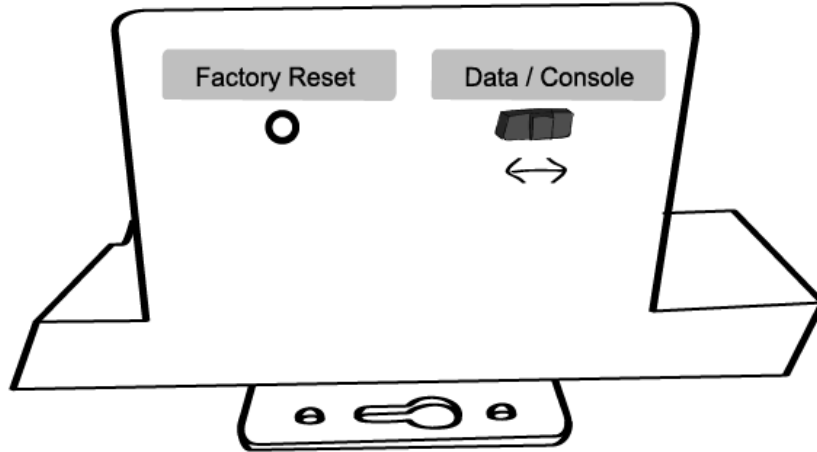


그림 B-2 Rhio10의 데이터/콘솔 스위치

하이퍼터미널 (HyperTerminal)과 같은 터미널 에뮬레이터 프로그램을 실행합니다. 다음과 같이 터미널 에뮬레이션 프로그램의 시리얼 구성 파라미터를 설정합니다.

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

- ① [ENTER]를 누릅니다.
- ② 사용자 이름 및 비밀번호를 입력하여 Rhio10에 로그인 합니다. 출고 시 사용자 이름 및 비밀번호의 기본 설정은 모두 admin입니다.
- ③ 사용자가 Rhio10에 성공적으로 로그인 하면 그림 B-3과 같이 컴퓨터 명령어 프롬프트 화면이 나타납니다.

```
login: admin
password: *****
Type 'help' to get command usages
> help
set group par1 [par2 ...] + <CR>
- group = 'ip', 'host', or 'admin'
- par1 ... = configuration parameters. Use * to keep a parameter's value
get [group] + <CR>
- group = 'ip', 'host', 'admin' or 'status'
- If group is specified, shows settings of the group.
- If group is omitted, shows settings of all groups.
factorydefault [option] + <CR>
- if option is omitted, all parameters are set with factory default values.
- if option='-ip',
  all parameters except IP settings are set with factory default values.
help [group] + <CR>
- If group is omitted, shows this screen.
- If group is specified, shows 'set' command usage of the group.
save + <CR>
```

```

- Save changes
exit + <CR>
- Exit without rebooting the device
reboot + <CR>
- Exit and reboot the device
>

```

그림 B-3 Rhio10 콘솔 화면

명령어 프롬프트 화면에서 사용자는 'set', 'get' 및 'save' 명령어를 사용하여 구성 파라미터를 설정하고, 보고, 저장할 수 있습니다. 사용자는 또한 'exit' 및 'reboot' 명령어를 사용하여 콘솔을 종료하거나 장치를 재부팅할 수 있습니다. 'help' 명령어를 사용하면 명령어 사용법을 볼 수 있습니다. 명령어 사용법에 대한 설명은 3.2 단원을 참조하십시오.

### B.1.2 원격 콘솔 사용

Rhio10은 시리얼 콘솔 뿐만 아니라 telnet을 통한 원격 콘솔 기능도 제공하므로 사용자는 원격 사이트에서 Rhio10에 접속하여 시스템 환경 설정 등을 할 수 있습니다. 사용자가 원격 콘솔 Port에 접속하려면 Rhio10의 IP 주소를 알아야 합니다. 원격 콘솔용 Port 번호는 telnet에 할당된 TCP Port 번호 23입니다.

사용자는 원격 콘솔 또는 시리얼 콘솔에 한번에 한 명씩만 로그인할 수 있습니다. 원격 콘솔 접속 중에 시리얼 콘솔이 접속된 경우, 접속 중인 원격 콘솔은 중단되고 시리얼 콘솔이 종료될 때까지 더 이상 원격 콘솔이 접속할 수 없습니다.

Rhio10의 원격 콘솔에 접속하려면 다음과 같은 단계를 수행하십시오.

① TeraTerm-Pro나 하이퍼터미널과 같은 telnet 기능을 지원하는 프로그램을 실행합니다. Target IP 주소 및 Port 번호는 Rhio10의 것이어야 합니다. 필요한 경우 Port 번호를 23으로 지정합니다.

컴퓨터의 명령줄에 다음 명령을 입력합니다.

```
telnet 192.168.1.254
```

또는 다음과 같은 파라미터로 telnet 프로그램을 실행합니다.

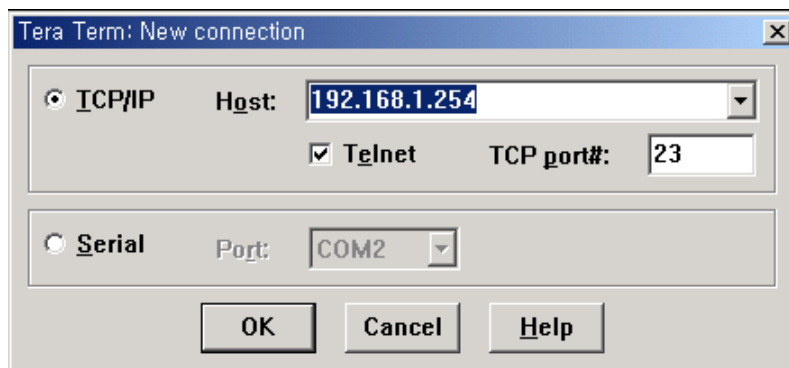


그림 B-4 Telnet 프로그램 설정 예

② Rhio10에 사용자 이름과 비밀번호를 입력하여 로그인 합니다. 출하 시 사용자 이름 및 비밀번호의 기본 설정은 모두 **admin** 입니다.

③ Rhio10에 로그인하면 시리얼 콘솔과 동일한 명령어 프롬프트 화면이 나타납니다. 시리얼 콘솔과 같이 구성 파라미터를 설정하고, 보고, 저장하며 콘솔을 종료하고 장치를 재부팅할 수 있습니다.

시리얼 콘솔 또는 기타 원격 콘솔이 이미 연결된 경우, 새로운 콘솔은 연결되지 않습니다.

## B.2 명령어 사용법

Rhio10은 구성 및 제어를 위한 간단한 몇 가지 명령어를 제공합니다. 표 B-1 은 RHIO10이 지원하는 명령어 모음이 요약되어 있습니다.

표 B-1 Rhio10 명령어 모음 개요

명령어	설명	결과
set group par1 [par2 ...] + <CR>	구성 파라미터 설정 - group = 'ip', 'host', 또는 'admin' - par1 ... = 구성 파라미터. 파라미터 값을 유지하려면 * 사용	성공하면, "OK" + <CR> + <LF> If error "ERROR" + <CR> + <LF>
get [group] + <CR>	구성 파라미터 값 보기 - group = 'ip', 'host', 'admin' 또는 'status' - 그룹이 지정된 경우, 해당 그룹의 설정을 표시한다. - 그룹을 지정하지 않은 경우, 모든 그룹의 설정을 표시한다.	파라미터 값 표시
help [group] + <CR>	명령어 사용법 화면을 표시한다. - 그룹을 지정하지 않은 경우, 도움말 화면을 표시한다.. - 그룹을 지정한 경우, 해당 그룹의 'set' 명령어 사용법을 표시한다.	도움말 메시지 표시
save + <CR>	변경 사항 저장	성공하면, "OK" + <CR> + <LF>

		오류가 발생하면 "ERROR" + <CR> + <LF>
exit + <CR>	장치를 재부팅하지 않고 종료(변경 사항 적용 안됨)	성공하면, "OK" + <CR> + <LF> 오류가 발생하면 "ERROR" + <CR> + <LF>
reboot + <CR>	장치 종료 및 재부팅	None

### B.2.1 'set' 명령어

'set' 명령어를 사용하여 각 환경에 대해 Rhio10의 파라미터 값을 설정할 수 있습니다. 기본적인 'set' 명령어 사용법은 다음과 같습니다.

**set group par1 [par2 ...] + <CR>**

**where,**

**group = 'ip', 'host', or 'admin'**

**par1 par2 ... = configuration parameters. Use \* to keep a parameter's value**

'group'은 파라미터를 입력할 범주입니다. 예를 들어, IP 구성과 관련된 파라미터를 설정하려면 그림 B-5와 같이 set 명령어를 사용하십시오.

```
> set ip static 192.168.1.100 255.255.255.0 192.168.1.1
OK
>
```

그림 B-5 IP 구성 화면 예

위의 예에서 첫번째 파라미터 'ip'는 뒤따르는 파라미터가 IP 설정 파라미터임을 뜻합니다. 두번째 파라미터 'static'은 RHIO10에서 세번째 파라미터 '192.168.1.100'을 static IP 주소로 사용할 것을 나타냅니다. 다섯번째 파라미터는 서브넷 마스크를 나타내고 그 다음 파라미터는 기본 게이트웨이 IP 주소를 나타냅니다.

그룹의 파라미터 중 하나만 변경하려면 뒤에 오는 파라미터를 누락시키거나 '\*'을 사용하여 파라미터 값을 유지하면 됩니다. 아래 화면은 IP 주소와 게이트웨이 IP 주소는 변경하지 않고 서브넷 마스크만 변경하는 방법을 보여줍니다.

```
> set ip static * 255.255.0.0
OK
>
```

그림 B-6 하나의 파라미터 값만 변경한 예

'set' 명령어의 사용법은 그룹에 따라 달라집니다. 'help group' 명령어를 사용하면 각 그룹의 'set' 명령어 사용법을 볼 수 있습니다. 예를 들어, IP 구성 시 'set' 명령어 사용법을 보려는 경우, 'help ip' + <CR>을 입력하면 그림 B-7과 같이 IP 구성을 위한 'set' 명령어 사용법이 나타납니다.

```
> help ip
set ip ipmode par1 par2 ...
- ipmode: static=Static IP / dhcp=DHCP / pppoe=PPPoE
- parameters:
if ipmode = static,
    par1 = IP address,
    par2 = subnet mask,
    par3 = gateway
if ipmode = dhcp,
    no parameters required
if ipmode = pppoe,
    par1 = PPPoE username,
    par2 = PPPoE password
>
```

그림 B-7 도움말 화면 예

#### 주의:

변경된 값은 'save' 및 'reboot' 명령어를 실행해야만 적용됩니다. 자세한 내용은 B.2.4 ~ B.2.6 단원을 참조하십시오.

### B.2.2 'get' 명령어

'get' 명령어를 사용하여 Rhio10의 현재 파라미터 값과 상태를 볼 수 있습니다. 기본적인 'get' 명령어 사용법은 다음과 같습니다.

**get [group] + <CR>**

**where,**

**group = 'ip', 'host', 'admin' or 'status'**

**- If group is specified, shows settings of the group.**

**- If group is omitted, shows settings of all groups.**

그룹은 'set' 명령어와 같이 파라미터가 속하는 범주입니다. 예를 들어, IP 구성과 관련된 파라미터 값을 보려면 그림 B-8과 같은 get 명령어를 사용할 수 있습니다.



```

> get ip
IP_mode: static
IP_address: 192.168.1.100
Subnet_mask: 255.255.255.0
Gateway: 192.168.1.1
>

```

그림 B-8 IP 구성 보기 화면

'status' 그룹은 'set' 명령어가 적용되지 않는 특수 그룹입니다. 'get status'는 현재 시스템 상태 화면을 표시합니다.

```

> get status
Serial_no.: Rhio10-0207_test
MAC_address: 00-01-95-77-88-99
F/W_REV.: V1.2.0
Current_IP: 192.168.0.125
>

```

그림 B-9 상태 보기 화면

그룹을 지정하지 않을 경우, get 명령어는 그림 B-10와 같이 모든 파라미터 값을 표시합니다.

```

> get
--- Status ---
Serial_no.: Rhio10-0207_test
MAC_address: 00-01-95-77-88-99
F/W_REV.: V1.2.0
Current_IP: 192.168.0.125
--- Admin ---
Username: admin
Password: admin
Devicename: RHIO10 Device
--- IP ---
IP_mode: dhcp
--- Host ---
Host_mode: tcps
Local_port: 6001
Inactivity_timeout(sec): 300
--- Serial ---
Baudrate: 9600
Data_bits: 8_bits
Parity: None
Stop_bits: 1_bit
Flow_control: None
DTR_option: Always_high
DSR_option: None
Interchar_timeout(ms): 50
>

```

그림 B-10 모든 파라미터 보기 화면

## B.2.3 'help' 명령어

'help' 명령어를 사용하면 콘솔 화면에서 명령어 사용법 도움말을 볼 수 있습니다. 기본적인 명령어 사용법은 다음과 같습니다.

**help [group] + <CR>**

*where,*

***if group is omitted, overall help screen will be displayed***

***if group is specified, 'set' command usage of specified group will be displayed.***

그림 B-11은 그룹을 지정하지 않은 경우의 도움말 화면이고 그림 B-12은 'ip' 그룹을 지정한 경우의 도움말 화면입니다.

```
> help
set group par1 [par2 ...] + <CR>
- group = 'ip','host' or 'admin'
- par1 ... = configuration parameters. Use * to keep a parameter's value
get [group] + <CR>
- group = 'ip','host','admin' or 'status'
- If group is specified, shows settings of the group.
- If group is omitted, shows settings of all groups.
help [group] + <CR>
- If group is omitted, shows this screen.
- If group is specified, shows 'set' command usage of the group.
factorydefault [option] + <CR>
- if option is omitted, all parameters are set with factory default values.
- if option='-ip',
    all parameters except IP settings are set with factory default
values.
save + <CR>
- Save changes
exit + <CR>
- Exit without rebooting the device
reboot + <CR>
- Exit and reboot the device
```

그림 B-11 도움말 화면

```
> help ip
- set ip ipmode par1 par2 ...
- ipmode: static=Static IP / dhcp=DHCP / pppoe=PPPoE
- parameters:
if ipmode = static,
    par1 = IP address,
    par2 = subnet mask,
    par3 = gateway
if ipmode = dhcp,
    no parameters required
if ipmode = pppoe,
    par1 = PPPoE username,
    par2 = PPPoE password
```

그림 B-12 'ip' 그룹을 지정한 경우의 도움말 화면

## B.2.4 'save' 명령어

'save' 명령어를 사용하면 현재 파라미터 변경 사항이 비휘발성 메모리에 저장됩니다. 'save' 명령어 사용법은 다음과 같습니다.

**save + <CR>**

'reboot' 명령어를 사용하거나 수동으로 Rhio10을 재부팅하면 저장된 변경 사항이 적용됩니다.

## B.2.5 'exit' 명령어

'exit' 명령어를 사용하면 현재의 시리얼 또는 원격 콘솔 세션이 닫힙니다. 그러나 변경된 파라미터는 Rhio10을 수동으로 재부팅 해야만 적용됩니다. 'exit' 명령어 사용법은 다음과 같습니다.

**exit + <CR>**

## B.2.6 'reboot' 명령어

'reboot' 명령어를 사용하면 Rhio10이 즉시 재부팅 됩니다. Rhio10이 다시 가동되면 변경된 파라미터 값이 적용됩니다. 'reboot' 명령어 사용법은 다음과 같습니다.

**reboot + <CR>**

## B.3 콘솔 명령어들을 이용한 시스템 설정

### B.3.1 네트워크 설정

IP 설정을 위한 기본적인 'set' 명령어 사용법은 다음과 같습니다.

**set ip ipmode par1 par2 ...**

**where,**

**ipmode: 'static' for Static IP / 'dhcp' for DHCP / 'pppoe' for PPPoE**

**parameters:**

**if ipmode = static,**  
     **par1 = IP address, par2 = subnet mask, par3 = gateway**  
**if ipmode = dhcp,**  
     **no parameters required**  
**if ipmode = pppoe,**  
     **par1 = PPPoE username, par2 = PPPoE password**

Rhio10의 IP 구성 파라미터를 설정하려면 set 명령을 다음과 같이 사용하십시오.

**set ip static ip\_address subnet\_mask default\_gateway + <CR>**

**where,**

**ip\_address = IP address of the RHIO10**  
**subnet\_mask = Subnet mask**  
**default\_gateway = Default gateway IP address**

변경 사항을 적용하려면 설정 후에 'save' 및 'reboot' 명령어를 실행해야 합니다.

```
> set ip static 192.168.1.10 255.255.255.0 192.168.1.1
OK
```

```
> set ip dhcp
OK
>
```

```
> set ip pppoe pppoeuser pppoepassword
OK
>
```

### **B.3.2 호스트 모드 설정**

호스트 모드 구성을 위한 기본적인 'set' 명령어 사용법은 다음과 같습니다.

**set host hostmode par1 par2 ...**

**where,**

**hostmode: tcps=TCP server / tcpc=TCP client / tcpsc=TCP server & client**

**parameters:**

**if hostmode = TCP server (tcps),**

**par1 = listening TCP port,**

**par2 = inactivity timeout (sec)**

*if hostmode = TCP client (tcpc),*  
*par1 = destination IP address,*  
*par2 = destination TCP port,*  
*par3 = cyclic connection interval (min),*  
*par4 = inactivity timeout (sec)*

*if hostmode = TCP server & client (tcpsc),*  
*par1 = listening TCP\_port,*  
*par2 = destination IP address,*  
*par3 = destination TCP port,*  
*par4 = cyclic connection interval (min),*  
*par5 = inactivity timeout (sec)*

*\* set cyclic connection interval to 0 not to use cyclic connection*

*\* set inactivity timeout to 0 for unlimited timeout*

Rhio10이 TCP 서버로 작동하도록 설정하려면 'set' 명령어를 다음과 같이 사용하십시오.

**set host tcps listening\_TCP\_port inactivity\_timeout + <CR>**

*where,*

*listening\_TCP\_port: Listening TCP port*

*Inactivity\_timeout: Inactivity timeout in seconds.*

```
> set host tcps 6001 300
OK
>
```

Rhio10이 TCP 클라이언트로 작동하도록 설정하려면 set 명령어를 다음과 같이 사용하십시오.

**set host tcpc dest\_ip dest\_port cyclic\_connection\_interval inactivity\_timeout + <CR>**

*where,*

*dest\_ip = destination IP address*

*dest\_port = destination TCP port*

*cyclic\_connection\_interval = cyclic connection interval in minutes*

*inactivity\_timeout = inactivity timeout in seconds.*

```
> set host tcpc 192.168.1.1 6001 10 300
OK
>
```

Rhio10이 TCP 서버/클라이언트 모드로 작동하도록 설정하려면 'set' 명령어를 다음과 같이 사용하십시오.

***set host tcpssc listening\_port dest\_ip dest\_port cyclic\_connection\_interval inactivity\_timeout***

***where,***

***listening\_port = listening TCP port***

***dest\_ip = destination IP address***

***dest\_port = destination TCP port***

***cyclic\_connection\_interval = cyclic connection interval in minutes***

***inactivity\_timeout = inactivity timeout in seconds.***

TCP 서버/클라이언트 모드 설정시 파라미터 정의는 TCP 서버 모드 및 TCP 클라이언트 모드 파라미터와 동일합니다.

```
> set host tcpssc 6001 192.168.1.100 7001 10 300
OK
>
```

### **B.3.3 시스템 관리**

사용자는 set 명령어를 사용하여 다음과 같이 관리자 사용자 이름, 비밀번호 및 장치 이름을 설정할 수 있습니다.

***set admin username password devicename***

***username: login username***

***password: login password***

***devicename: device name***

```
> set admin adminuser adminpassword Rhio10_test1
OK
>
```

## 부록 C. 문제 해결

### C.1 전원/LED 상태

문제	원인	조치
Power LED가 점등 되지 않는다.	전원 케이블이 연결되지 않았습니다.	전원 연결을 확인하십시오. Rhio10 은 DC 9V 부터 48 V까지를 전원으로 입력을 받을 수 있습니다.
Link LED가 점등 되지 않는다.	Ethernet 케이블이 연결되지 않았습니다.	Ethernet 케이블 연결을 확인하십시오.
	부적합한 Ethernet 케이블이 사용되었습니다.	Ethernet 케이블에는 스트레이트 케이블과 크로스오버 케이블의 두 가지 유형이 있습니다. Ethernet 허브를 사용하는 경우 스트레이트 케이블을 사용하십시오 RHIO10과 원격 호스트 사이를 직접 연결하는 경우에는, 크로스오버 케이블을 사용하십시오.
ACT LED가 깜박이지 않는다.	IP 설정이 올바르지 않습니다.	IP 설정 파라미터를 확인하십시오.

### C.2 시리얼 콘솔

문제	원인	조치
시리얼 콘솔이 연결되지 않는다.	부적합한 시리얼 케이블이 사용되었습니다.	시리얼 콘솔에는 반드시 시리얼 콘솔 케이블(CAT5 Straight Cable + DB9F Straight 아답터)을 사용하십시오.
	터미널 에뮬레이션 프로그램의 시리얼 Port 설정이 올바르지 않습니다.	터미널 에뮬레이션 프로그램의 시리얼 Port 설정이 올바르지 않습니다. 9600 bps, 8 Data bits, No parity, 1 stop bit, flow control None
	콘솔/Ethernet 스위치의 위치가 올바르지 않습니다.	콘솔/Ethernet 스위치가 콘솔 쪽에 있는지 확인하십시오.

콘솔에 로그인할 수 없다.	사용자 이름이나 비밀번호가 올바르지 않습니다.	유효한 사용자 이름과 비밀번호를 사용하십시오. 사용자 이름이나 비밀번호를 잊은 경우 기본값 복구 스위치를 사용하여 출하시 기본 설정을 복구하십시오. 사용자 이름 및 비밀번호의 출하시 기본값은 모두 admin입니다.
----------------	---------------------------	---

### C.3 원격 콘솔

문제	원인	조치
Telnet을 사용하여 RHIO10에 연결할 수 없다.	RHIO10에 유효한 IP 주소가 할당되지 않았습니다.	시리얼 콘솔 또는 Rhio Manager를 사용하여 RHIO10에 유효한 IP 주소를 할당하십시오.
	다른 사람이 시리얼 콘솔을 사용 중이다.	시리얼 콘솔을 종료한 다음 Data/Console 스위치를 Data로 옮긴 후 telnet 연결을 다시 시도하십시오.
콘솔에 로그인할 수 없다.	사용자 이름이나 비밀번호가 올바르지 않습니다.	유효한 사용자 이름과 비밀번호를 사용하십시오. 사용자 이름이나 비밀번호를 잊은 경우 기본값 복구 스위치를 사용하여 출하시 기본 설정을 복구하십시오. 사용자 이름 및 비밀번호의 출하시 기본값은 모두 admin입니다.

### C.4 IP 주소

문제	원인	조치
RHIO10의 IP 주소를 모릅니다.		시리얼 콘솔을 사용하여 IP 주소를 찾으십시오.
		Rhio10 Manager 프로그램을 사용하여 네트워크에서 RHIO10을 검색하십시오.
Rhio Manager로 RHIO10을 검색할 수 없다.	네트워크에 UDP Traffic 이 많습니다.	간혹 네트워크에 UDP 트래픽이 많아 반응이 느려 표시되지 않을 수 있습니다. 이 경우 이더넷 크로스 케이블을 연결해서 사용하면 Rhio10을 검색할 수 있습니다.



## C.5 DHCP

문제	원인	조치
IP 주소를 할당 받을 수 없다.	DHCP 서버가 작동하지 않습니다.	DHCP 서버가 올바르게 작동하는지 확인하십시오.
RHIO10의 IP 주소가 변경되었다.	DHCP 서버가 대여 시간을 연장하지 않습니다.	DHCP 서버가 올바르게 작동하는지 확인하십시오.

## 부록 D. Rhio 라이브러리

### D.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
};

```

## D.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 prot 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;

```

### D.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, "02&04|I1", 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.isEnabled[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 함수에서 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 false를 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 false를 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.
```

## 부록 E. 품질 보증 정책

### E.1 제품 품질 보증 정책

주식회사 세나테크놀로지 (이하 “SENA”) 는 제품이 기술명세 및 부속 자료에 명시된 사양에 부합하고 그에 따라 작동하며, 보증 기간 동안 재료 및 공법상 하자가 없음을 보증한다. 보증기간은 제품을 수령하는 시점부터 시작된다.

SENA의 보증 범위는, SENAs의 자체적 판단에 따라, 하자 또는 부적합 제품의 수리 또는 교체로 국한되며, (a) 제품을 잘못 적용 또는 사용하는 경우 (b) 사용자가 SENAs의 사용 지침을 준수하지 않은 경우; (c) 제품의 관리 소홀, 남용 및 우발적인 사고의 경우; 또는 (d) SENAs가 제공하지 않은 장비나 소프트웨어와 관련된 경우에 생기는 기능상 문제에 대해서는 책임지지 않는다.

사용자는 구매 또는 수령일자를 증빙하는 자료와 함께 제품을 SENAs 또는 제품을 구매한 해외 딜러에게 보냄으로써 제한적 보증 서비스를 받을 수 있다. 이 때, 사용자는 운송 중 생길 수 있는 제품 분실 또는 파손의 가능성을 인지하고, 운송비를 선지급하며, 원래의 운송 포장 등을 사용하기로 합의한다.

### E.2 책임의 한계

SENA는, 본 문서에 명시된 경우를 제외하고는, 본 계약에 따라 제공되는 장비, 부품 또는 서비스에 대해 어느 특정 용도에 대한 상업성이나 적합성 여부를 포함한 어떠한 보증도 명시적이든 묵시적이든 하지 않는다. SENAs 또는 그 딜러는, 손해 가능성에 대한 사전 인지 여부와 관계없이 본 계약에 따라 제공되는 장비, 부품 또는 서비스가 기대한대로 동작하지 않는 경우 발생할 수 있는 직접, 간접, 부수, 특별 또는 결과적 손해나 기대 이익의 손실 등 어떠한 다른 손해에 대하여 책임을 지지 않는다.

어떠한 경우에도 SENAs 또는 그 딜러의 책임 한도는 제품의 지불된 판매 가격을 초과하지 않는다.

### E.3 하드웨어 제품 보증 상세 내용

SENA는 내장 하드웨어 제품을 일(1)년간 보증하고, 외장 하드웨어 제품을 제품에 따라서 삼(3)년간 또는 오(5)년간 보증한다.

보증절차: 하드웨어 제품이 반환된 경우, **SENA**는 자체 판단에 따라 추가 비용 없이 제품을 수리 또는 교체한다. 단, 아래에 해당되는 경우는 제외한다. 수리 부품과 교체 제품은 일대일 교환 형태로 제공되며, 재생 또는 신제품으로 할 수 있다. 교체된 제품 및 부품은 **SENA**로 귀속된다. 제품에 대해 보증이 적용되지 않는 것으로 **SENA**가 판단한 경우, **SENA**는 고객의 선택에 따라 부품 및 노무에 관한 **SENA**의 표준 요율에 따라 제품을 수리하거나 또는 제품을 그냥 반환할 수 있다.

보증 제외 경우:

- 사고, 떨어뜨린 경우, **SENA** 제품에 충격을 가한 경우,
- **SENA**의 온도 및 습도 명세를 초과한 환경에서 제품을 작동한 경우,
- 전원 불안정, 고압 방전으로 인한 경우,
- 부적절한 접지 및 부정확한 배선으로 인한 경우,
- 고객 등의 오용, 부주의로 인한 경우,
- **SENA** 사용자 매뉴얼에 따라 제품을 설치 또는 작동하지 않은 경우,
- 고객 또는 제3자의 부적절한 유지보수로 인한 경우,
- 홍수, 번개, 지진으로 인한 경우,
- 물을 쏟은 경우,
- 통상의 마모로 인한 부품 교체,
- 하드웨어가 변경된 경우,
- **SENA**의 서면 합의 없이 제3자가 수리를 시도한 제품,
- 하드웨어에 **SENA** 소프트웨어의 변형, 또는 **SENA** 소프트웨어 이외의 소프트웨어를 사용한 경우, **SENA**가 변형을 승인한 경우 제외.
- 소모품인 충전용 배터리의 사용 시간이 제품의 사용 방법과 기간에 따라서 최초 구입시보다 현저히 줄어든 경우.

## E.4 소프트웨어 제품 보증의 상세

보증기간: 소프트웨어 제품의 보증기간은 일(1)년으로 한다.

보증범위: **SENA**의 보증은 사용자가 **SENA**에게 소프트웨어 부적합을 통보한 때로부터 합리적 시간 내에 소프트웨어 버그 픽스 또는 패치를 제공하는 것으로 제한된다.

## E.5 제3자 소프트웨어 제품 보증의 상세

제3자 소프트웨어의 보증정책은 해당 벤더의 품질 보증 정책을 따른다.