

# **Rhio10**

## **Remote I/O Manager**

### **User Manual**

**Version 1.4.4**

**2011-12-26**

## **Guide for the Rhio10**

Version 1.4.4

Firmware version 1.4.x

Printed in Korea

## **Copyright**

Copyright 2005~2012, Sena Technologies, Inc. All rights reserved.

Sena Technologies reserves the right to make changes and improvements to its product without providing notice.

## **Trademark**

Rhio™ is a trademark of Sena Technologies, Inc.

Windows® is a registered trademark of Microsoft Corporation.

Ethernet® is a registered trademark of XEROX Corporation.

## **Notice to Users**

When a system failure may cause serious consequences, protecting life and property against such consequences with a backup system or safety device is essential. The user agrees that protection against consequences resulting from system failure is the user's responsibility.

This device is not approved for life-support or medical systems.

Changes or modifications to this device not explicitly approved by Sena Technologies will void the user's authority to operate this device.

## **Technical Support**

Sena Technologies, Inc.

210 Yangjae-dong, Seocho-gu

Seoul 137-130, Korea

Tel: (+82-2) 576-7362

Fax: (+82-2) 573-7710

E-Mail: [support@sena.com](mailto:support@sena.com)

Website: <http://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	"Table 2-2. Terminal Block Assignment of the Rhio10 " is added.
V1.0.2	2005-11-02	D.H. Shin	Typo errors corrected.
V1.0.3	2005-11-08	JS. Kim	Operating and storage temperature is updated.
V1.4.1	2006-11-15	JS. Kim	Added IP filtering section
V1.4.2	2007-03-22	JS. Kim	Fixed the contents and added the calculation example of LRC field
V1.4.3	2007-05-07	H.R. Zo	Copyright changed.
V1.4.4	2011-12-26	H.R. Zo	Package does not contain CD-ROM

# Table of Contents

<b>1. Introduction .....</b>	<b>7</b>
1.1 Overview .....	7
1.2 Package Check List .....	7
1.3 Product Specification .....	8
<b>2. Getting Started .....</b>	<b>10</b>
2.1 Panel Layout .....	10
2.2 Connecting the Hardware .....	12
2.2.1 Setting up DIN Rail mount kit .....	12
2.2.2 Connecting the Power .....	13
2.2.3 Connecting to the Network .....	14
2.3 Rhio Manager Installation .....	14
2.3.1 Rhio Manager Installation .....	14
2.3.2 Basic configuration using Rhio Manager .....	15
<b>3. System Configuration and Administration .....</b>	<b>17</b>
3.1 Network Configuration .....	17
3.1.1 Static IP .....	19
3.1.2 DHCP .....	20
3.1.3 PPPoE .....	21
3.1.4 IP Filtering .....	21
3.2 Host Mode Configuration .....	22
3.2.1 TCP Server Mode Operations .....	25
3.2.2 TCP Client Mode Operation .....	27
3.2.3 TCP Server/Client Mode Operation .....	29
3.3 System Administration .....	30
3.4 Restoring Factory Default .....	31
<b>4. I/O Setting and Application .....</b>	<b>32</b>
4.1 I/O Monitoring and Control .....	32
4.1.1 LED .....	32
4.1.2 Specifying monitoring interval .....	33
4.1.3 Monitoring I/O port .....	33
4.1.4 Controlling digital output port .....	33
4.2 Digital Input Setting .....	34
4.2.1 Setting Enable/Disable .....	34
4.3 Digital Output Port Setting .....	35

4.3.1 Setting Enable/Disable .....	35
4.3.2 Setting run condition .....	36
4.3.3 Delay & Pulse Operation .....	37
4.3.4 Setting Power-out Post Recovery .....	37
4.4 ADC Input Port Setting .....	37
4.4.1 Setting Reference .....	38
4.4.2 Setting ADC Input Port .....	38
4.5 I/O Port Connection .....	39
4.5.1 Digital Output Port .....	39
4.5.2 Digital Input Port .....	39
4.5.3 ADC Input Port .....	40
<b>5. Software Development &amp; Application .....</b>	<b>42</b>
5.1 The Rhio Library .....	42
5.1.1 Overview of the Rhio Library .....	42
5.1.2 Reference .....	42
5.2 Creating and demonstrating a sample program with Rhio library .....	45
5.2.1 Program UI Configuration and their related classes .....	45
5.2.2 Processing .....	46
5.3 Rhio Communication Protocol .....	49
5.3.1 Overview .....	49
5.3.2 ON/OFF Control .....	51
5.3.3 Input/Output State .....	54
5.3.4 Set/Run .....	54
5.3.5 Output Port Setting .....	55
5.3.6 ADC Input Port Setting .....	58
5.3.7 Port Enable Setting .....	59
5.3.8 Port Power-out Post Recovery Setting .....	62
<b>Appendix A. Connection .....</b>	<b>64</b>
A.1 Ethernet Pin Outs .....	64
A.2 Console Port Pin Outs .....	64
A.3 Ethernet Wiring Diagram .....	65
A.4 Serial Console Wiring Diagram .....	66
<b>Appendix B. Accessing Console Port and Command Usage .....</b>	<b>67</b>
B.1 Accessing Console Port .....	67
B.1.1 Using Serial Console .....	67
B.1.2 Using Remote Console .....	69

B.2 Command Usage.....	70
B.2.1 'set' Command.....	71
B.2.2 'get' Command.....	72
B.2.3 'help' Command.....	74
B.2.4 'save' Command.....	75
B.2.5 'exit' Command.....	75
B.2.6 'reboot' Command.....	75
B.3 System Configuration using Console command.....	75
B.3.1 Network Configuration.....	75
B.3.2 Host Mode Configuration.....	76
B.3.3 System Administration.....	78
<b>Appendix C. Troubleshooting.....</b>	<b>79</b>
C.1 Power/LED Status Troubleshooting.....	79
C.2 Serial Console Troubleshooting.....	79
C.3 Remote Console Troubleshooting.....	80
C.4 IP Address Troubleshooting.....	80
C.5 DHCP Troubleshooting.....	80
<b>Appendix D. Rhio Linrary.....</b>	<b>81</b>
D.1 Enumeration Type.....	81
D.2 Structure.....	83
D.3 Function.....	84
<b>Appendix E. Warranty.....</b>	<b>97</b>
E.1 GENERAL WARRANTY POLICY.....	97
E.2 LIMITATION OF LIABILITY.....	97
E.3 HARDWARE PRODUCT WARRANTY DETAILS.....	98
E.4 SOFTWARE PRODUCT WARRANTY DETAILS.....	99
E.5 THIRD-PARTY SOFTWARE PRODUCT WARRANTY DETAILS.....	99

# 1. Introduction

## 1.1 Overview

Rhio10 is a remote I/O manager which enables users to control/monitor various I/O devices such as sensors and actuators through 10Base-T Ethernet.

Rhio10 supports 10 Digital Relay Output ports and has basic logic function capability such as AND, OR, NOT and Delay/Pulse along with the status of the input ports. Rhio10 supports 12 optically isolated digital inputs for monitoring of the digital sensors. Rhio10's Analog ports support both level mode for data acquisition and switch mode for threshold detection. The data communication between host computers is done by event-driven method which is triggered when the status is changed.

Users may probe, configure, test the Rhio10 system using Windows application, Rhio Manager and create their own Windows based application program by using MFC DLL library.

Rhio10 is designed to meet the requirements for the various applications such as remote data acquisition system, distributed I/O system, industrial automation control/monitoring/metering.

***Please note that this manual assumes user knowledge of Internetworking protocols and serial communications.***

## 1.2 Package Check List

- Rhio10 external box
- CAT5 cable
- RJ45 to DB9 Female cable connector
- DIN rail mount kit

### 1.3 Product Specification

<b>Network Interface</b>	10 Base-T Ethernet with RJ45 Ethernet connector
	Supports static and dynamic IP address
<b>Digital Input</b>	<ul style="list-style-type: none"> <li>-Number of channels: 12</li> <li>-Input type: Voltage</li> <li>-Input circuitry: Optically isolated photo-coupler</li> <li>-Input range: 0V ~ ±24V OFF 0V ~ ±1.2V, ON ±3.3V ~ ±24V</li> <li>-Sampling rate: 20ms</li> <li>-Isolation voltage: 5KV</li> </ul>
<b>Digital Output</b>	<ul style="list-style-type: none"> <li>- Number of channels: 10</li> <li>- Output type: Relay</li> <li>- Rated load: 3A/240VAC</li> <li>- Insulation resistance: 1000MΩ Min (DC500V)</li> <li>- Isolation voltage(coil and contact): 4KV</li> <li>- Reaction within 10ms</li> </ul>
<b>Analog Input</b>	<ul style="list-style-type: none"> <li>- Number of channels: 4</li> <li>- Effective resolution: 10-bit</li> <li>- Input type: Voltage, Direct Coupling</li> <li>- Input range: 0V ~ Aref (Analog reference voltage, 2~5V)</li> <li>- Sampling rate: 1000 samples/sec</li> </ul>
<b>Protocol</b>	ARP, IP/ICMP, TCP telnet, DHCP client, PPPoE
<b>Management</b>	- Rhio Manager Windows Utility, Serial Console or Telnet
<b>Software Support</b>	<ul style="list-style-type: none"> <li>- Windows MFC DLL library</li> <li>- I/O configuration, I/O status monitoring/control</li> </ul>
<b>Diagnostic LED</b>	<ul style="list-style-type: none"> <li>- Power, Link, Act</li> <li>- Digital Output, 1~10</li> <li>- Digital Input, 1~12</li> <li>- Analog Input, 1~4</li> </ul>
<b>Power</b>	9V~48VDC, Max. 5W
<b>Environmental</b>	<ul style="list-style-type: none"> <li>- Operating temperature: 0℃ to 50℃</li> <li>- Storage temperature: -20℃ to 66℃</li> <li>- 90% Non-condensing</li> </ul>



<b>Physical properties</b>	137 x 111 x 58 (mm), 5.4 x 4.4 x 2.3 (in.) Weight: 730g
<b>Certification</b>	FCC (A), CE, MIC
<b>Warranty</b>	5-year limited warranty

## 2. Getting Started

This chapter describes how to set up and configure the Rhio10 in the first place.

- 2.1 *Panel Layout* explains the panel layout and LED indicators.
- 2.2 *Connecting the Hardware* describes how to set up DIN rail mount kit and how to connect the power and the network to the Rhio10.

Following items are pre-required to get started.

- DIN rail mount kit (included in the package).
- CAT5 cable for configuration or network (included in the package).
- RJ45 to DB9 Female connector for configuration (included in the package).
- PC with Network Interface Card (hereafter, NIC) and/or one RS232 serial port.

### 2.1 Panel Layout

The Rhio10 has LED indicator lamps for status display. The lamps in the left hand side indicate the system power-on status, link and act for 10Base-T Ethernet status. There are 10 lamps for displaying digital output status, 12 lamps for digital input status, and lamps for 4 analog port status. There is a DIP switch on the side of the Rhio10 for data/console switching and hole for factory reset operation. Table 2-1 shows the description of the indicator lamps of the Rhio10.

Table 2-1. LED indicator lamps

Lamps		Function
10Base-T	Link	Turned on to Green if connected to 10 Base-T Ethernet network.
	Act	Blink whenever there is any activities such as incoming or outgoing packets through the Rhio10 Ethernet port
Staus	Power	Turned on to RED if power is supplied
Digital Input	DI 1 ~ DI 12	Turned on to GREEN if input status
Digital Output	DO 1 ~ DO 10	Turned on to GREEN if output status
Analog Input	AI 1 ~ AI 4	In Level Input mode, it is turned on to GREEN if the value is larger than 512. In Switch Input mode, it is turned on to GREEN if it is larger than threshold value.

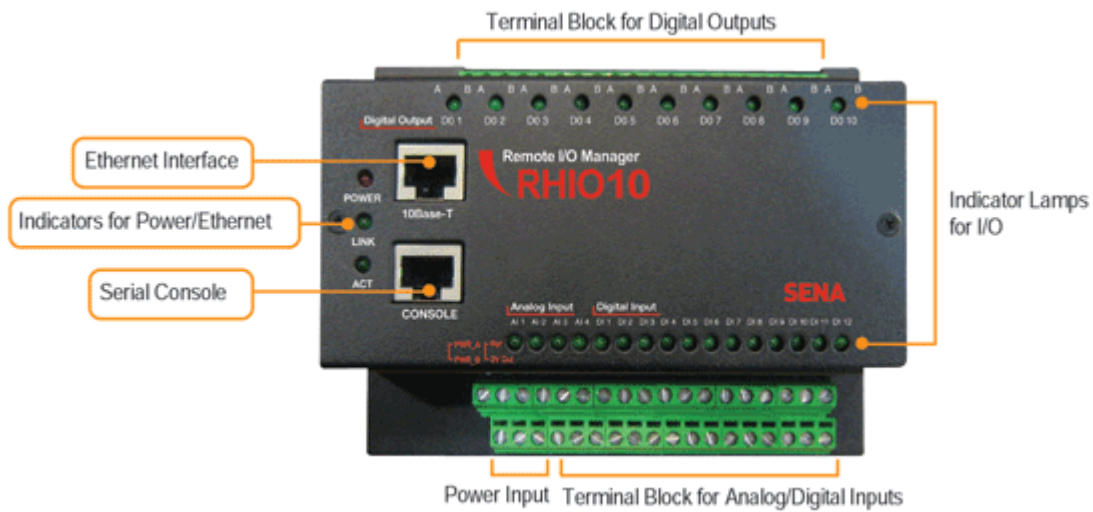


Figure 2-1. The panel layout of the Rhio10

Table 2-2. Terminal Block Assignment of the Rhio10

	Digital Output												Analog Input / Digital Input																								
	DO1		DO2		DO3		DO4		DO5		DO6		DO7		DO8		DO9		DO10		AI1	AI2	AI3	AI4	DI1	DI2	DI3	DI4	DI5	DI6	DI7	DI8	DI9	DI10	DI11	DI12	
	A	B	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	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36		
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36			
Input Blocks assignment	<ul style="list-style-type: none"> <li>• Power Input ( PWR_A , PWR_B ) : (1,2)</li> <li>• Analog reference voltage ( Ref ) : (3) <ul style="list-style-type: none"> <li>-It is analog reference voltage input block. Rhio represents voltage from 0 V to a reference voltage in 1,024 steps. A reference voltage may not exceed 5 V. For details, refer to section 4.4 ADC Input Port Setting.</li> </ul> </li> <li>• 5V Out : (4) <ul style="list-style-type: none"> <li>- It is AVCC output block. User can input the power of devices. Ground is no.5,7,9 and 11 blocks.</li> </ul> </li> <li>• Analog Input ( AI1 ~ AI4 ) : (5,6),(7,8),(9,10),(11,12)</li> <li>• Digital Input ( 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><b>Note : 1. Each Input is composed of one pair of upper and lower blocks.</b></p> <p><b>2. Except for the no.3 and 4 input blocks, users can connect the wire to the blocks without considering the polarity.</b></p>																																				
	Output Blocks assignment	<p><b>Each Output is composed of one pair of A and B blocks as follows.</b></p> <ul style="list-style-type: none"> <li>• Digital Output ( 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 Connecting the Hardware

### 2.2.1 Setting up DIN Rail mount kit

Users may use DIN rail mounting kit included in the package to install the Rhio10 on to the DIN rail.

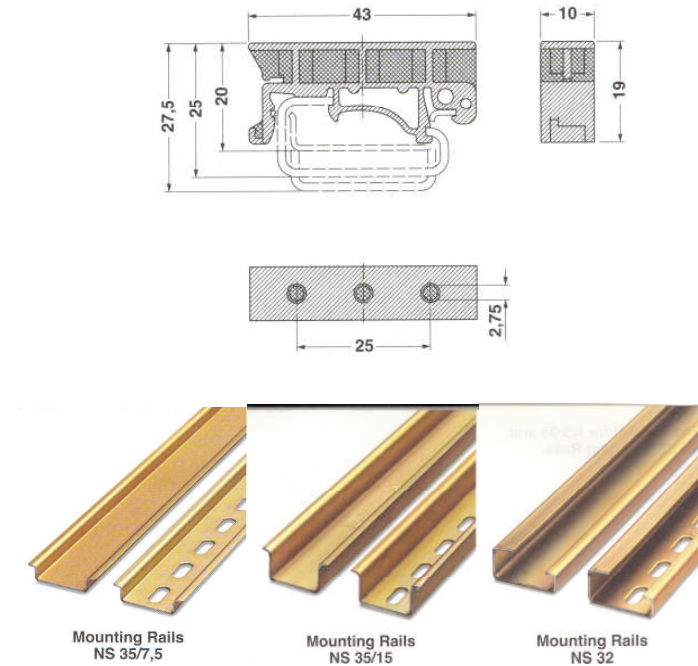


Figure 2-2. Dimension of DIN Rail mount kit and applicable DIN Rails

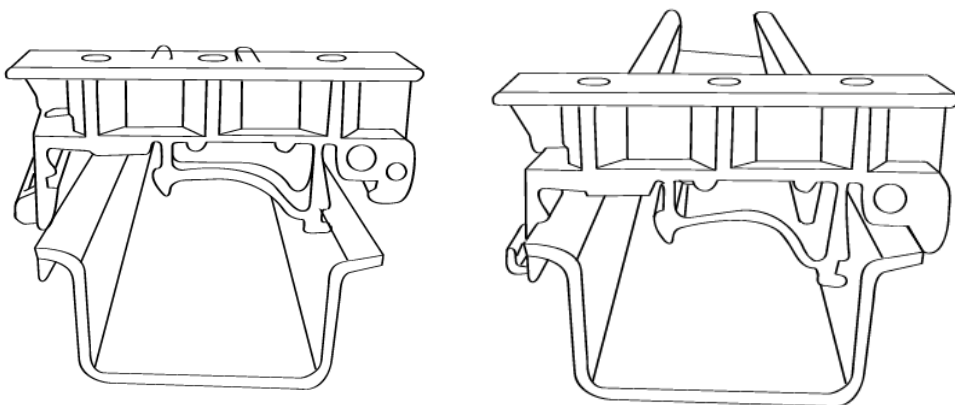


Figure 2-3. Installing DIN Rail mount kit into DIN Rail

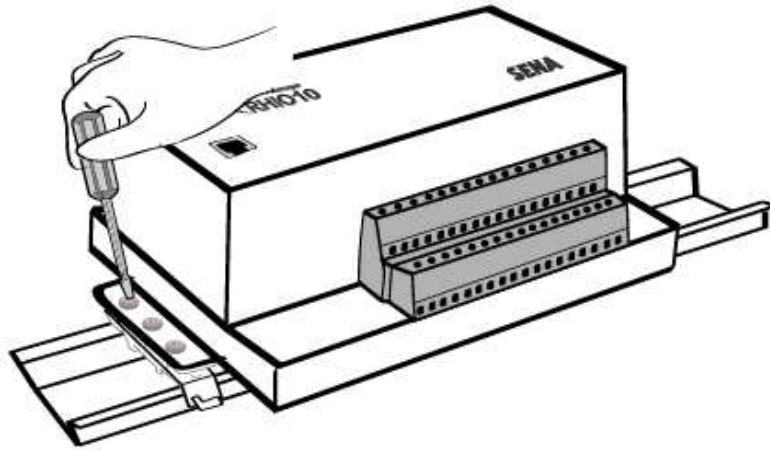


Figure 2-4. Setting up the Rhio10 to DIN Rail

### 2.2.2 Connecting the Power

Supply the proper power according to the power specification of the Rhio10, i.e. 9V~48VDC, MAX. 5W. If the power is properly applied, **[Power]** indicator will maintain RED. Be sure not to use the cable longer than 3m for normal operation.

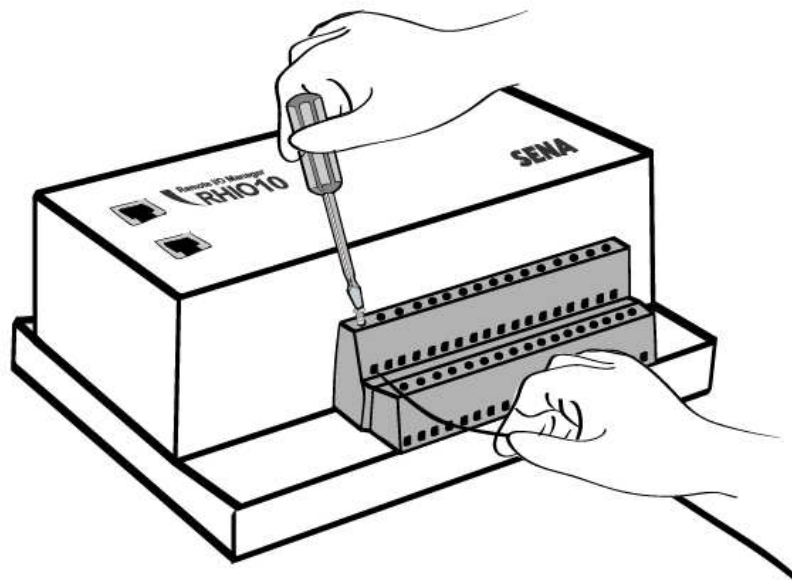


Figure 2-5. Connecting the power to the Rhio10

### 2.2.3 Connecting to the Network

Connect the one end of the Ethernet cable to the Rhio10 10Base-T port and the other to the Ethernet network. If the cable is properly hooked up, the Rhio10 will have a valid connection to the Ethernet network by indicating:

- [Link] lamp of the Rhio10 maintains solid green
- [Act] lamp continuously blinks to indicate the incoming/outgoing Ethernet packets

If any of the above does not happen, the Rhio10 is not properly connected to the Ethernet network.

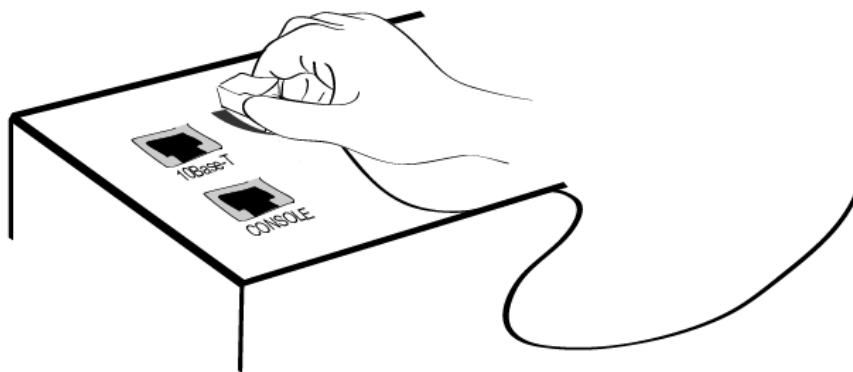


Figure 2-6. Connecting a network cable to the Rhio10

## 2.3 Rhio Manager Installation

Rhio Manager is a Windows Utility program for system configuration and I/O test of the Rhio10.

### 2.3.1 Rhio Manager Installation

Users may install the Rhio Manager software by downloading at Sena web site <http://www.sena.com/support/downloads>. If it is installed normally, then it will be placed on to the menu of [Start]->[Program]->[SENA]->[RHIO Manager]. The screen layout is shown in Figure 2-7.

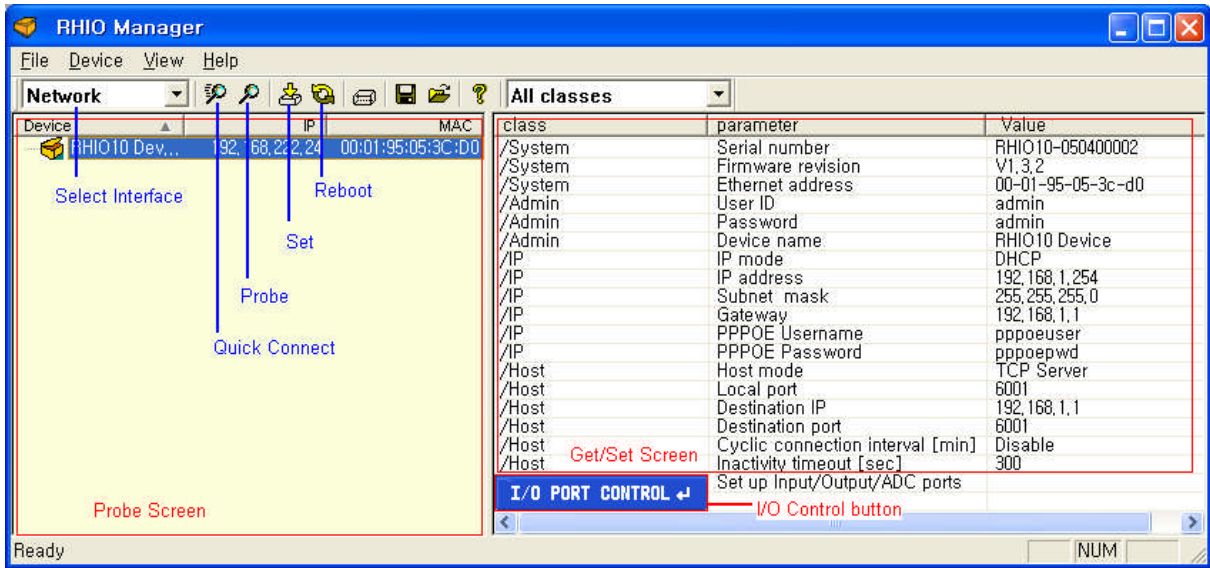


Figure 2.7 Rhio Manager screen layout

## 2.3.2 Basic configuration using Rhio Manager

### Select Interface

It specifies the communication method between Rhio10 and Rhio Manager. Be sure to set it up as "Network" mode.

### Quick Connect

Users may connect to the Rhio10 from using Rhio Manager by specifying the destination IP address. Once users click [Quick Connect] button, then Rhio10 detected is shown on the Probe screen as a tree structure form. If users click the one, then they will be asked to enter the ID and Password. The Rhio10 parameters are displayed on the Get/Set screen with correct ID and password (Factory default ID:PASSWORD is **admin** : **admin**.)

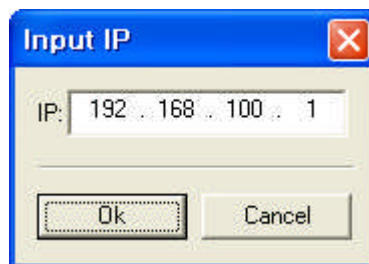


Figure 2.8 Quick connect dialog box

**Probe**

Users may probe the Rhio10 on the network. If users perform [Probe] command, then the Rhio10 detected will be shown on the Probe screen. If users select the one in the Probe screen, then they are asked to enter ID and Password. Users are able to configure the Rhio10's parameters within the Get/Set screen.

**Set**

After users complete the configuration, they have to click [Set] button to reflect the changes. Once it is done, the changes will be stored on to the flash memory and the system will be rebooted.

**Reboot**

Users may reboot the Rhio10 by using the botton. In this case, the TCP connection is disconnected, whereas the I/O management status is still maintained.

**I/O PORT CONTROL**

Users may monitor, control and configure the Rhio10's I/O ports by clicking [I/O PORT CONTROL] button.



### 3. System Configuration and Administration

#### 3.1 Network Configuration

A valid IP address of the Rhio10 needs to be assigned before it starts to work in the user's network environment. A network system administrator may provide the user with this IP address setting information for the network. The IP address must be unique within the network. Otherwise, the Rhio10 will not have a valid connection to the network.

Users can choose the desired IP mode out of the three IP operating modes, i.e., **Static IP**, **DHCP** and **PPPoE**. The factory default IP mode is **DHCP** mode. Table 3-1 shows the parameter items for IP configuration.

Table 3-1 IP Configuration Parameter

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

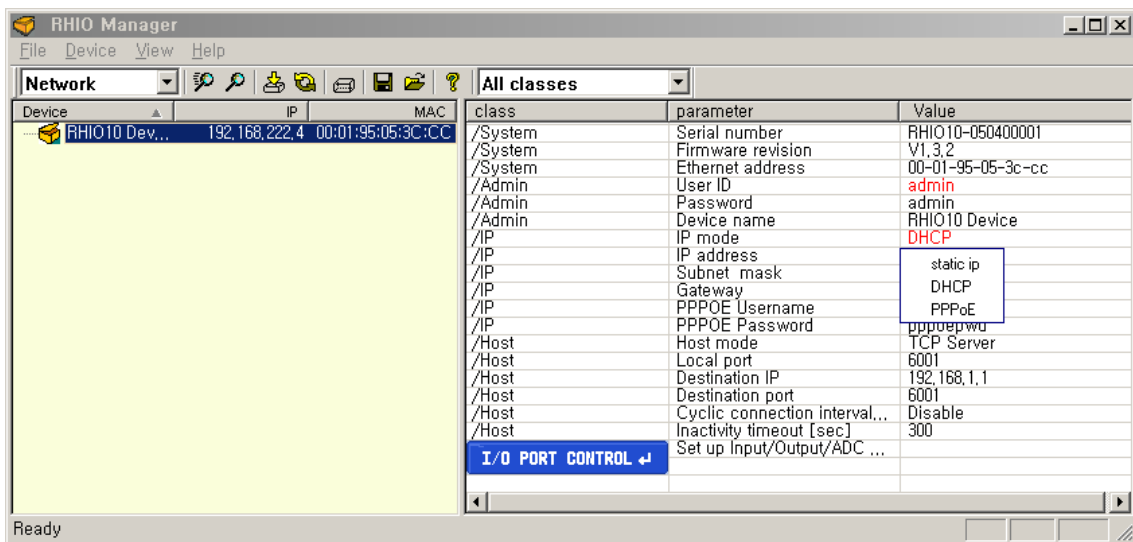


Figure 3-1. Rhio10 Network Configuration screen

The factory default configuration of IP mode is **DHCP** mode. In order to detect the Rhio10 for the first time, users have to perform [Probe] function by using Rhio Manager. If there is a DHCP server on the users' environment, then the IP address assigned by the DHCP server is displayed. Otherwise, it will display the IP address as 0.0.0.0. In this case, users have to change the IP mode as **Static IP** mode, and then enter the proper IP address for their environment. After the Rhio10 is displayed on the [Probe screen], click the unit that users want to connect.



Figure 3-2 Password Enter Dialog box

During the configuration, users may change the parameters as they want. The parameter value will be maintained as RED color until the changes are reflected by using [Set] button.

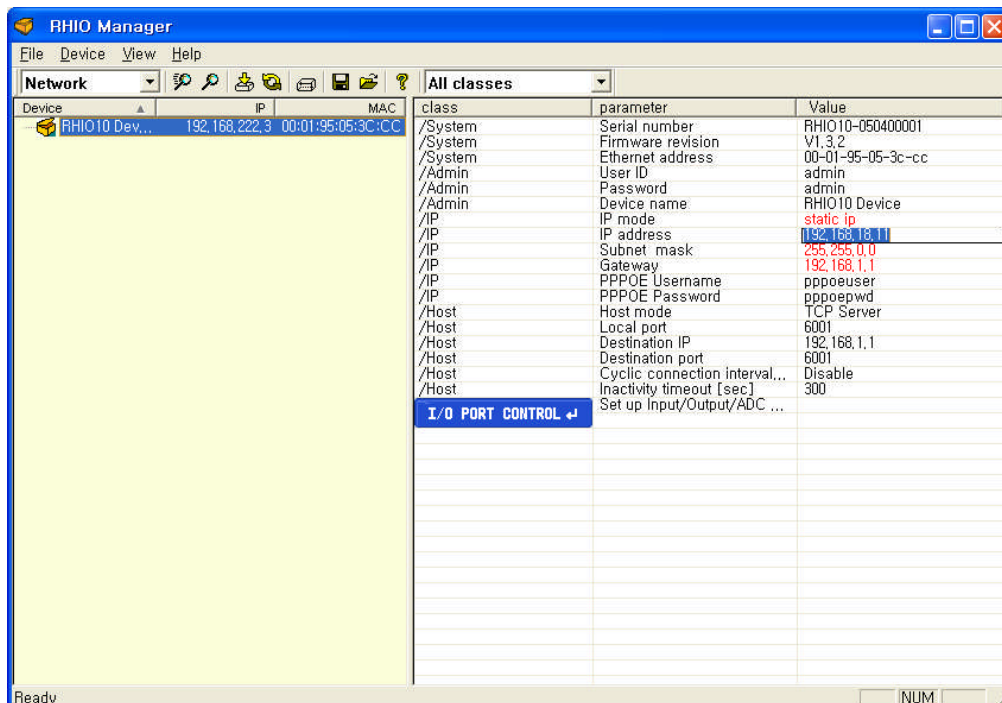


Figure 3-3 Network Configuration Screen example

Users will have to click [Set] button in order to reflect the parameter changes. Once [Set] operation is done, Rhio10 automatically store the changes and then reboot the system. Be sure to perform [Probe] function to check whether the changes are made correctly after [Set] operation.

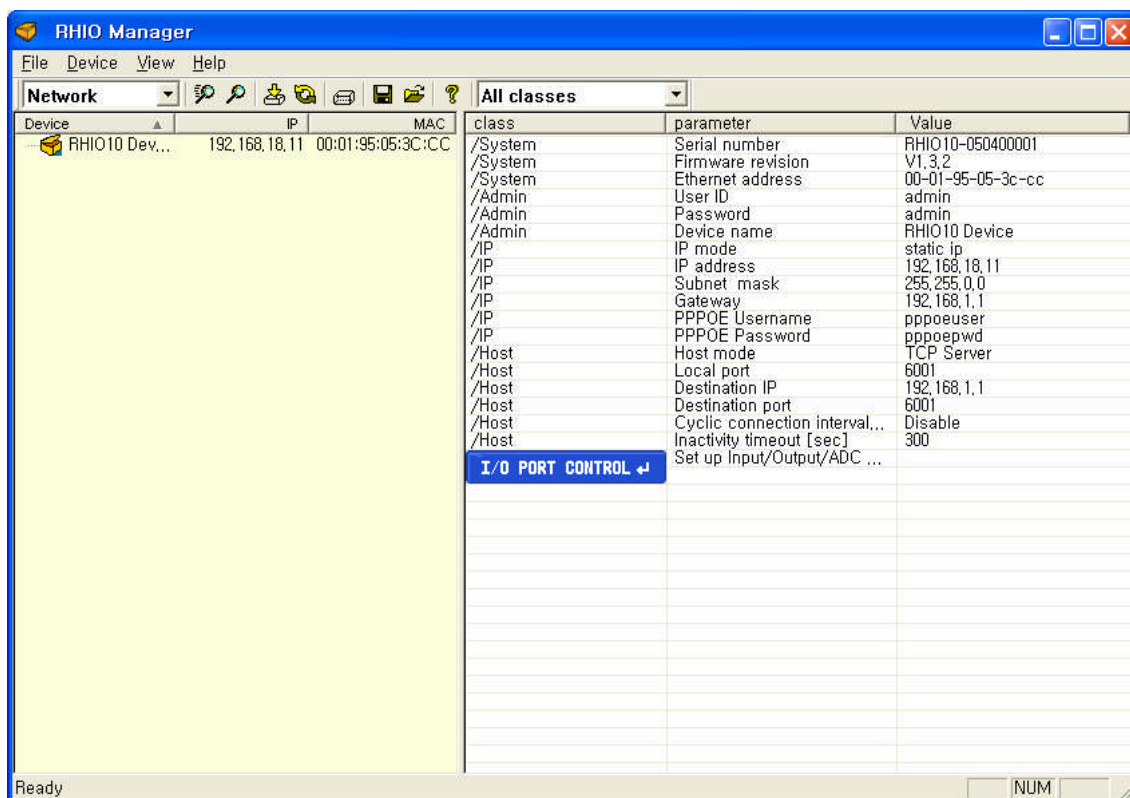


Figure 3-4 Network Configuration screen

### 3.1.1 Static IP

In the **Static IP** mode, users have to manually specify all the parameters such as **IP addresses** of the Rhio10, valid **subnet mask** and the **default gateway** IP address.

**IP address** is an identification number assigned to a computer as a permanent address on the network. Computers use IP addresses to identify and talk to each other on the network. Choose the proper IP address which is unique and valid on the network environment.

A subnet represents all the network hosts at one geographic location, in one building, or on the same local area network. When there is any outgoing packet over the network, the Rhio10 will check whether the desired TCP/IP host specified in the packet is on the local network segment

with the help of the **subnet mask**. If the address is proven to be on the same network segment as the Rhio10, the connection is established directly from the Rhio10. Otherwise, the connection is established through the given default gateway.

A **gateway** is a network point that acts as an entrance to another network. Usually, the computers that control traffic within the network or at the local Internet service provider are gateway nodes. The Rhio10 needs to know the IP address of the default gateway computer in order to communicate with the hosts outside the local network environment. For correct information on the gateway IP address, please refer to the network administrator.

### 3.1.2 DHCP

Dynamic Host Configuration Protocol (DHCP) is a communications protocol that lets network administrators manage and automate the assignment of IP addresses centrally in an organization's network. DHCP lets a network administrator supervise and distribute IP addresses from a central point and automatically send a new IP address when a computer is plugged into a different place in the network.

As described in the section 3.1, the IP address must be entered manually at each computer in Static IP mode and, if computers move to another location in another part of the network, a new IP address must be entered. Meanwhile, all the parameters including the IP address, subnet mask, gateway, DNS servers will be automatically configured when the IP address is assigned in DHCP mode. DHCP uses the concept of a "lease" or amount of time for which a given IP address will be valid for a computer. All the parameters required to assign an IP address are configured on DHCP server side, and each DHCP client computer receives this information when the IP address is provided at its boot-up.

To obtain an IP address, the Rhio10 sends a corresponding DHCP request as a broadcast over the network after each reset. The reply generated by the DHCP server contains the IP address as well as the subnet mask, gateway address, DNS servers and the lease time. The Rhio10 immediately places this information in its non-volatile memory. If the operating time reaches the lease time, the Rhio10 will request the DHCP server for renewal of its lease time. If the DHCP server approves extending the lease, the Rhio10 can continue to work with the current IP address. Otherwise, the Rhio10 will start the procedure to request a new IP address to the DHCP server.

A DHCP sever assigns IP addresses dynamically from an IP address pool, which is managed by the network administrator. This means DHCP client, i.e. the Rhio10, receives a different IP address each time it boots up. To prevent the case that users do not know the IP address of the

Rhio10 in such environments, its IP address should be reserved on the DHCP server side. In order to reserve the IP address in the DHCP network, the administrator needs the MAC address of the Rhio10 found on the label sticker at the bottom of the Rhio10:

MAC=00:01:95:04:0c:a1

### **3.1.3 PPPoE**

PPPoE (PPP over Ethernet) is a specification for connecting multiple computer users on an Ethernet local area network to a remote site through common customer premises equipment, which is the telephone company's term for a modem and similar devices. PPPoE can be used to have an office or building-full of users share ADSL, cable modem, or wireless connection to the Internet. Usually, it is used in broadband Internet access such as ADSL.

To make the Rhio10 work in PPPoE mode, users should have a PPPoE account and the equipments for PPPoE access such as an ADSL modem. Since the Rhio10 provides the PPPoE protocol, it can access the remote host on the Internet over ADSL connection. It is required to set up the user name and password of the PPPoE account.

If the IP mode is set to PPPoE, The Rhio10 negotiates the PPPoE connection with PPPoE server whenever it boots up. During the negotiation, it receives the information required for Internet connection such as IP address, gateway, subnet mask and DNS servers. If the connection is established, the Rhio10 tries to maintain the connection as long as possible. If the disconnection is detected, the Rhio10 will attempt to make a new PPPoE connection by requesting the new connection.

### **3.1.4 IP Filtering**

The Rhio10 has an IP address based filtering feature to prevent unauthorized remote hosts from accessing Rhio10. The user can allow one of the following scenarios by changing the parameter settings:

- Only one host of a specific IP address can access the Rhio10
- Hosts on the a specific subnet can access the Rhio10
- Any host can access the Rhio10

The user may allow a host or a group of hosts to access the Rhio10. Then the user must enter the IP address and subnet to be allowed for accessing.

To allow only a specific host to the Rhio10, enter the IP address of the specific host and just

give 255.255.255.255 for the subnet. To allow any hosts to the Rhio10, give 0.0.0.0 for both of the IP address and subnet. Please refer to Table 3-2 for more details.

Table 3-2 Input examples of allowed remote hosts

Allowable Hosts	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 Host Mode Configuration

Host mode represents the operating session mode of the Rhio10. Several host modes are available for the data communication between the I/O device and remote hosts. Since TCP is connection-oriented protocol, server, client, server/client modes are provided. Table 3-3 shows the brief description of the host modes. A factory default host mode is **TCP Server**.

**Note:**

**In order to use Rhio Manager, users have to set Host Mode as TCP Server mode.**

Table 3-3 The Rhio10 TCP/IP session modes

Mode	Description
TCP Server	Select this mode, when users want the Rhio10 to operate as a TCP server. The Rhio10 stands by until there is any TCP connection request. If TCP connection is not already established at that time, the Rhio10 accepts the request and the session is established. If it is established, then the system may receive the I/O control command and perform the corresponding operation. Since the Rhio10 supports only one TCP session, the additional TCP connection request will be rejected if already established. This mode is useful when users want to control I/O at any time.
TCP Client	Select this mode, when users want the Rhio10 to operate as a TCP client. Select this mode, when users want the Rhio10 to operate as a TCP client. When the I/O port status of the Rhio10 is changed or pre-defined timer is expired, the Rhio10 tries to establish a TCP connection to a remote server through its TCP port. If a

	TCP session is established between them, the Rhio10 will send data to the server. If there's any data from the server during the session, it will also send the I/O status data. However, if the Rhio10 failed to connect to the remote server, the I/O status data will be discarded. This is useful when users want to monitor the I/O status changes.
TCP Server /Client	If you are not sure which mode to choose, select this mode since it will be applied in most applications. In this mode, the Rhio10 operates as TCP server AND client. If the connection is not established, it will accept all incoming connection and connect to the remote host if there are any I/O status data. Otherwise, it will send data back and forth. In summary, the Rhio10 will work as if it is virtually connected to the remote host.

For each host mode, required parameters for configuration is summarized in Table 3-4.

Table 3-4 Host mode configuration parameters

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

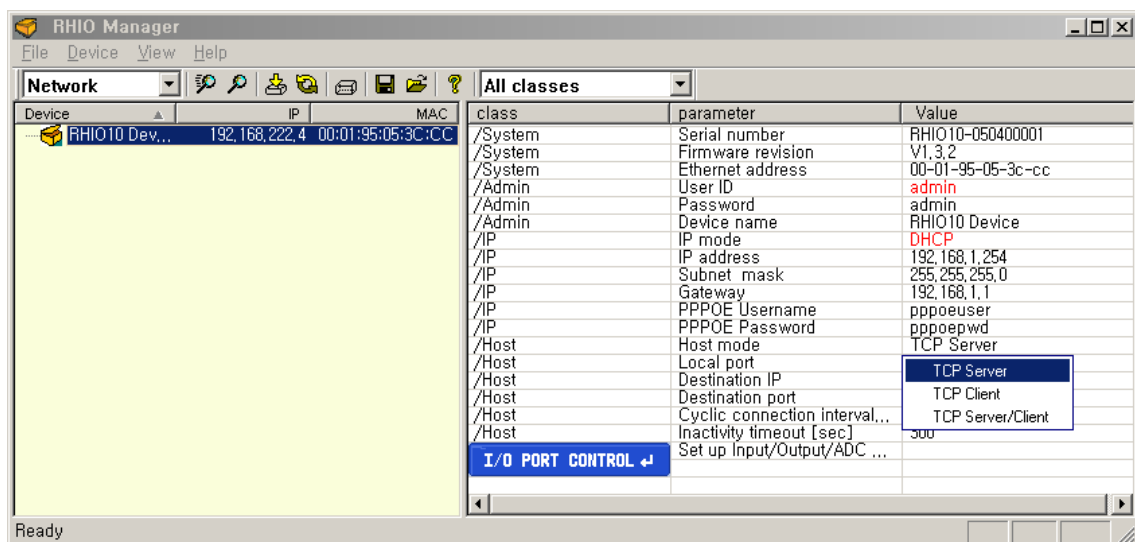


Figure 3-5. Rhio10 host mode configuration

For easier understanding of TCP modes, a simplified **State Transition Diagram** is often used. And too help users understand the diagram, the TCP state of the Rhio10 is briefly described as follows.

### **Listen**

It represents “a waiting for a connection request from any remote host”. It is a default start-up mode when it is set as *TCP server* mode. This state is valid only in *TCP server* mode operation.

### **Closed**

It means “No connection state at all”. If the data transfer is completed, the state is changed to this state if one of the host requests disconnection request. If it is in *TCP server* mode, the state is automatically changed to [Listen] mode. It is a default start-up mode when it is set as *TCP client* mode or *TCP server/client* mode.

### **Sync-Received**

In *TCP server* mode, the state will be changed from [Listen] to [Sync-Received], if any remote host sends connection request. If the Rhio10 accepts the request, the state will be changed into [Established]. This state is not valid in *TCP client* mode.

### **Sync-Sent**

If the Rhio10 sends a connection request to a remote host, the state is changed from [Closed] to [Sync-Sent]. This state is maintained until the remote host accepts the connection request. This state is valid only in *TCP client* mode.

### **Established**

It represents “an open connection”. If one of the hosts accepts a connection request from the other host, the connection is opened and state is changed into [Established].

### **Data**

When it is in [Established] state, data from a host will be transferred to the other one. For easier understanding of the TCP session operation, we called the state as [Data] state when actual data transfer is performed. Actually, the [Data] mode is a part of [Established] state as is described in the RFC 793 [Transmission Control Protocol]. This is a normal state for the data transfer phase of the connection.



### 3.2.1 TCP Server Mode Operations

The Rhio10 works as a TCP server, and the default TCP state is [Listen] in this mode. The Rhio10 supports only one TCP socket connection. If a connection is currently established, the additional connection requests will be rejected. The remote host will be a socket program acting as a TCP client running on user' PC.

#### 1) Typical State Transition

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

At start-up, an initial TCP state is [Listen]. If there is any incoming TCP connection request, the state will be changed into [Sync-Received], then [Established], which means a session is opened. For a while, data will be transferred between the hosts. This is the [Data] state. The session will be disconnected due to the request of one of them, which is [Closed] state. And then, the state is automatically changed to its original state, [Listen].

#### 2) Operations

##### I/O status data transfer

When a session has been established, the Rhio10 transfers the I/O status data to the IP address of the remote host. If there's no remote host connected to the Rhio10, all the incoming I/O status data are discarded.

##### Session disconnection

The connected session will be disconnected when the remote host sends disconnection request or when no I/O data transfer activity is found for a certain amount of time, which is "*Inactivity timeout*".

Figure 3-6 shows the State Transition Diagram of the session operations in *TCP server mode*.

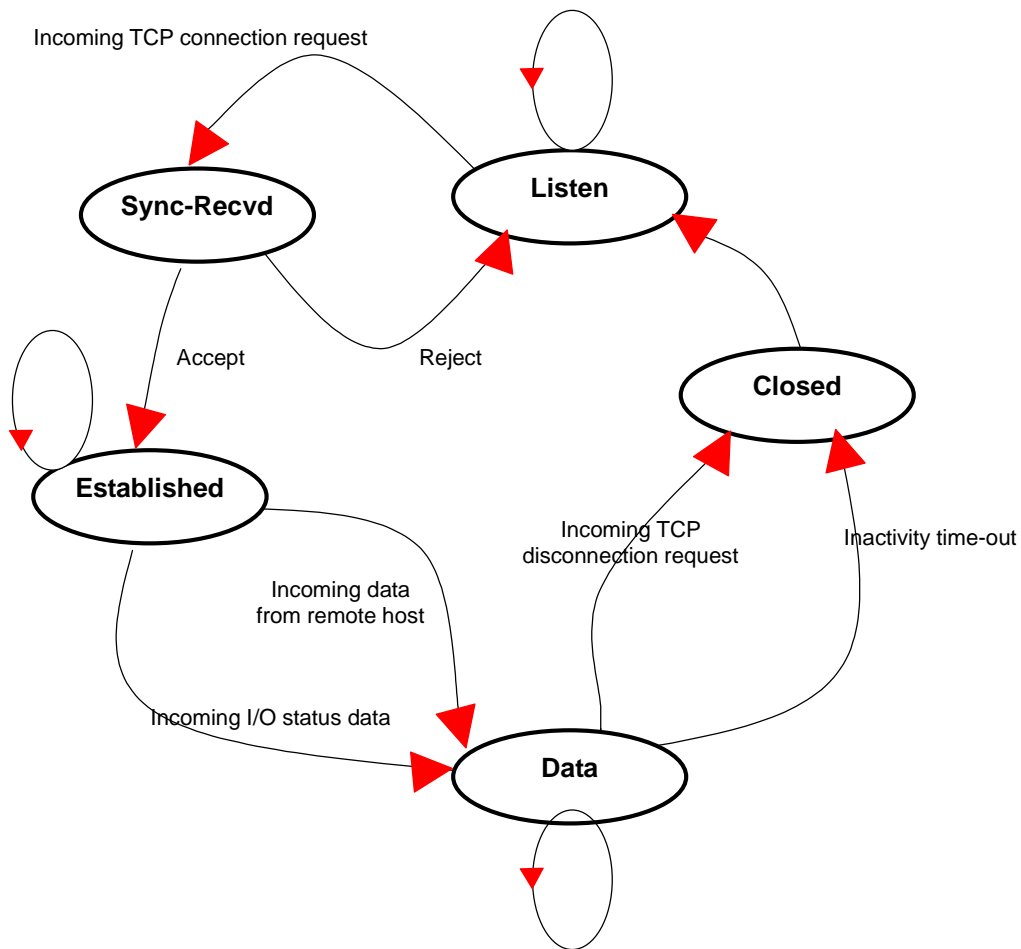


Figure 3-6 State Transition Diagram of TCP Server mode

**Listening TCP port** is the TCP port number through which remote host can connect a TCP session, and, send and receive data. Incoming connection request to the ports other than *Listening TCP Port* will be rejected. The Rhio10 does not restrict the number to a specific range, but it is strongly recommended not to use the well-known ports for certain application (See Appendix D. *Well-known Port Numbers*). To change the port number, select menu 2 on the *TCP Server mode* configuration screen.

**Inactivity timeout** is set to maintain the TCP connection state as *Closed* or *Listen* in TCP host mode unless there is any I/O data transfer between the I/O device and the Rhio10. If there is no incoming or outgoing I/O data during the specified *inactivity timeout* interval, the existing TCP connection will be closed automatically.

If the value of *inactivity timeout* is set to 0, the current TCP connection is maintained unless

there's no connection close request. Although *inactivity timeout* is disabled, the Rhio10 will check the connection status between the Rhio10 and the remote host by sending "keep alive" packets periodically. If the remote host does not answer the packets, it is regarded that the connection is down unintentionally. Then, the Rhio10 will force to close the existing TCP connection.

### 3.2.2 TCP Client Mode Operation

The RHIO10 works as a TCP client, and the default TCP state is [Closed] in this mode. The remote host will be a socket program acting as a TCP server running on users' PC.

#### 1) Typical State Transition

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

At start-up, an initial TCP state is [Closed]. If there is any changed in I/O state, the Rhio10 will try to connect to a user-defined remote host. Then, the state will be changed to [Sync-Sent], which means the connection request is being sent. If the remote host accepts the request, the state will be changed into [Established], which means a session has been opened. For a while, I/O state data will be transferred between the hosts. This is [Data] state. The session will be disconnected due to the request of one of them, which is its original state, [Closed].

#### 2) Operations

##### I/O state data transfer

Whenever there is any change in the I/O status, it tries to connect to the user-defined IP address of the remote host, if TCP session is not established yet. If the RHIO10 succeeds in connecting to the remote host, the I/O status data will be transferred to the host. Otherwise, I/O status data will be cleared.

##### Session disconnection

The connected session will be disconnected when the remote host sends disconnection request or when no I/O data transfer activity is found for certain amount of time, which is "*Inactivity timeout*".

##### Connection request from remote host

All the incoming TCP connection requests will be rejected in *TCP client* mode.

### Cyclic Connection

If *Cyclic Connection* function is enabled, the Rhio10 will make an attempt to connect to the user-defined remote host at certain interval even if there's no change in the I/O status data. If the remote host prepares I/O control/monitor command, it will be transferred to the Rhio10 after the connection is established. Eventually, users can monitor the I/O device periodically by making the remote host send the serial command to the Rhio10 whenever it is connected to the remote host. This option is useful when users need to gather the device information periodically even if the status of the I/O device is not changed. Figure 3-7 shows the State Transition Diagram of the session operations in *TCP client* mode.

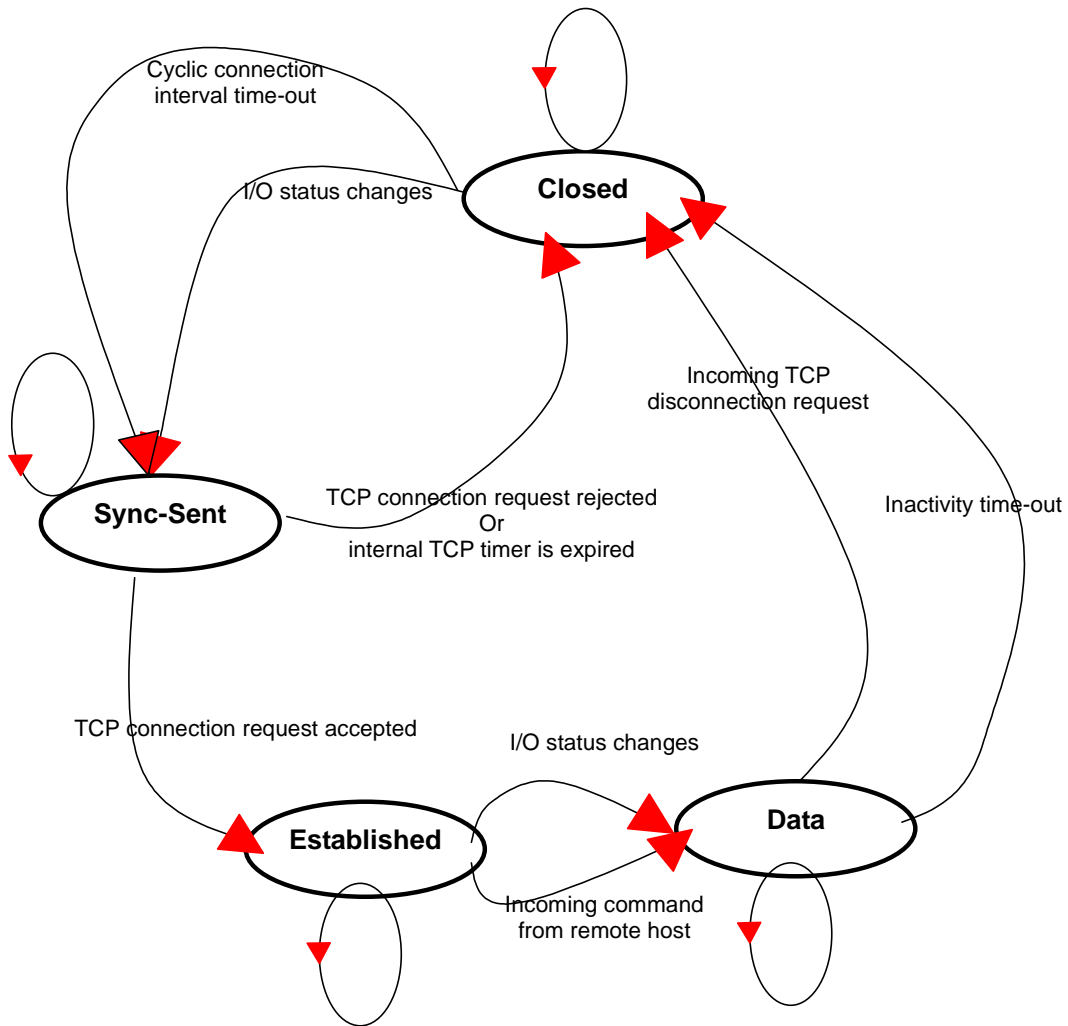


Figure 3-7 State Transition Diagram of TCP client mode

**Destination IP address** and **destination TCP Port** are the information on the remote host to which the Rhio10 will try to connect in TCP client mode. The IP address (or domain name) should be specified together with the TCP port number.

**Cyclic connection interval** is the time interval at which the Rhio10 will try to connect to the remote host regardless of the I/O status change. If the interval is specified with a valid value other than 0, the function is enabled. The time interval will be the specified value by the unit of minute. If the interval is entered as 0, cyclic connection feature will be disabled.

**Inactivity timeout** is the same as described in TCP server mode setting section.

### 3.2.3 TCP Server/Client Mode Operation

The RHIO10 works as either TCP server or client according to the situation. This will be the typical mode for most applications, since it will transfer the data either from serial port or from TCP port. The default TCP state is [Listen] which is the same as that of *TCP server* mode.

#### 1) Typical State Transition

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

Or

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

The initial state is [Listen]. If there is any change in I/O state, it will connect to the remote host as a TCP client. If there is incoming connection request from the remote host, it will accept the connection as a TCP server, and then transfer I/O data. Thus, users can assume that the RHIO10 is always connected to the specified remote host.

#### 2) Operations

The only difference from *TCP server* mode is that the RHIO10 will try to connect and send I/O state data to the remote host even if the TCP session is not established. The difference from *TCP client* mode is that it will accept incoming connection request from remote host if the session is not established. The detailed operation principles are the same as that of *TCP server* and *TCP client* mode.

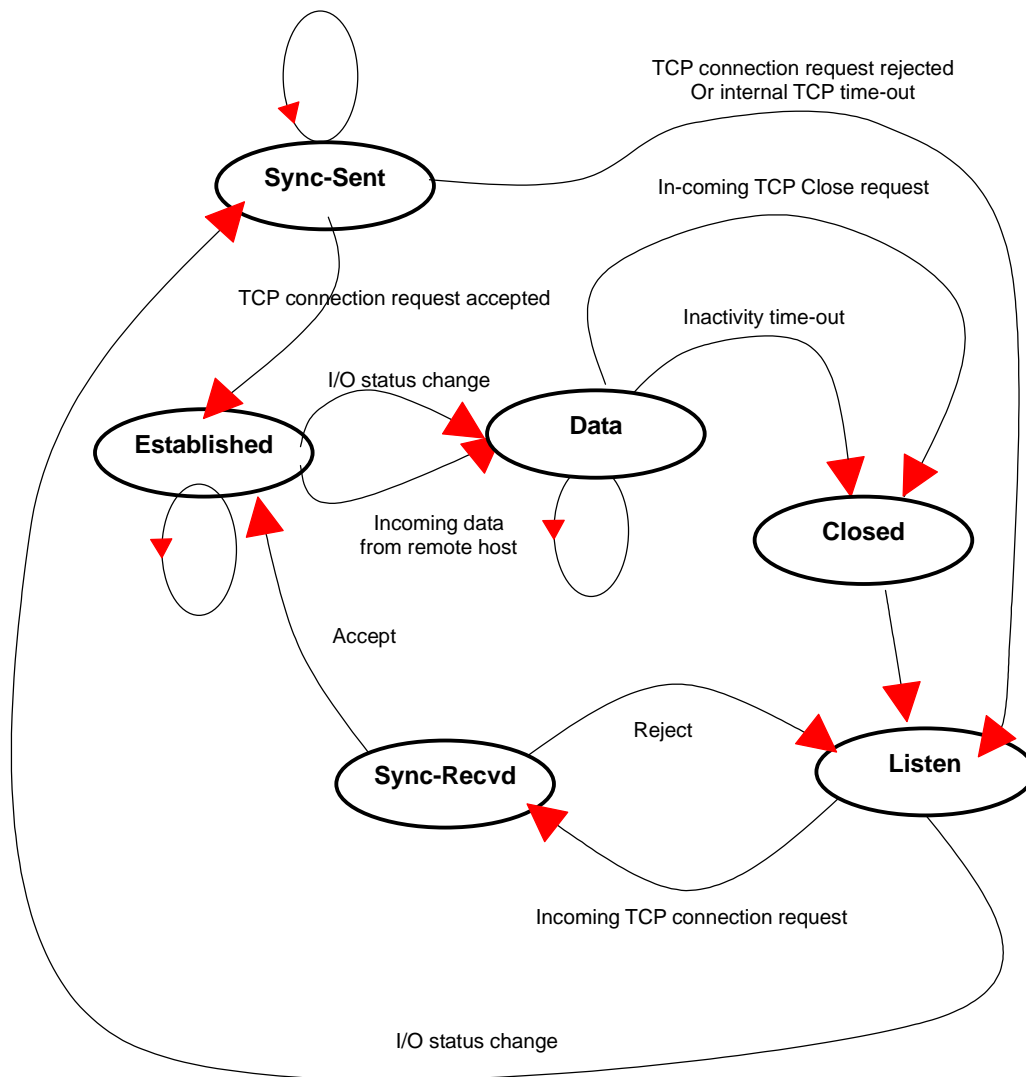


Figure 3-8 State Transition Diagram of TCP server/client mode

### 3.3 System Administration

Users may change the ID/Password and Device Name within the Rhio Manager admin section. Users have to click the [Clear] button after they change the ID/Password to clear the ID/Password configuration in the Rhio Manager as well. Device Name can be set up to discriminate the Rhio10 devices on the network.

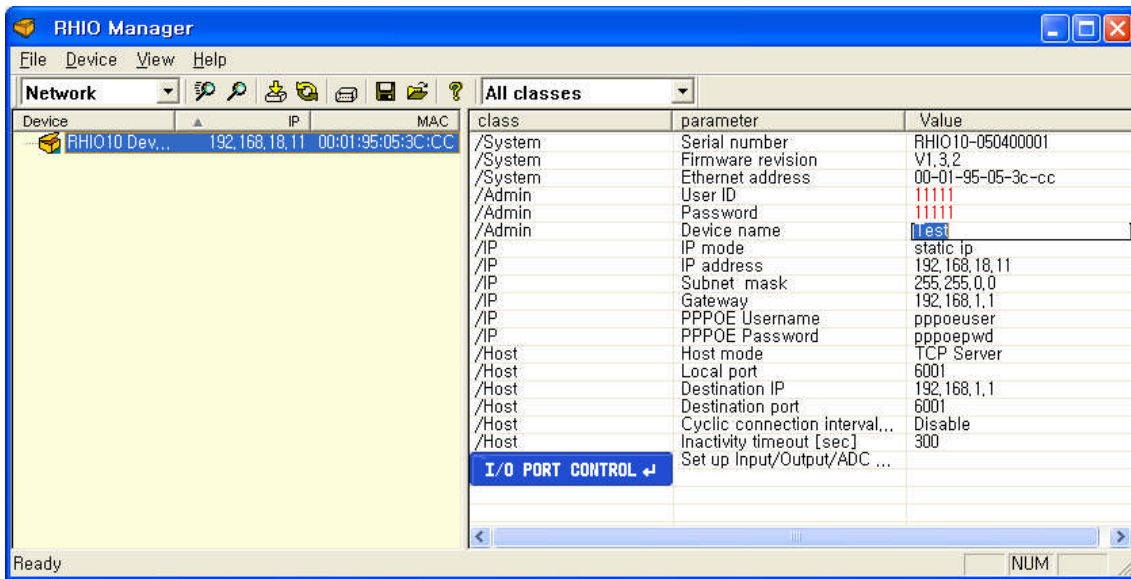


Figure 3-9 System Administration Configuration

### 3.4 Restoring Factory Default

Users may restore the Rhio10 parameters into factory default value by pressing factory reset switch on the hole of Rhio10 side panel. They will have to put the sharp pin into the hole and press it for around 1 sec to reset the Rhio10. Rhio10 will be rebooted after the operation.

The following is the factory default value of the parameters.

IP Mode:	DHCP
Host Mode:	TCP Server, TCP port 6001
ID / Password:	admin / admin
Device Name:	Rhio10 Device
I/O Port status:	Enable
Power-out Post Recovery:	Enable
ADC Operation Mode :	Level Mode
ADC Threshold value :	512
Output Port operation condition :	None
Run/Stop status :	Run

## 4. I/O Setting and Application

### 4.1 I/O Monitoring and Control

You can monitor, control and set I/O states by pressing the [I/O PORT CONTROL] button. Once [I/O PORT CONTROL] is invoked, Rhio Manager begins to monitor the I/O state by connecting to a local port of Rhio10 via a TCP socket. To allow this process, the Rhio10 should be set to TCP server mode.

#### 4.1.1 LED

- ON : Red Icon
- OFF : Blue Icon
- Disable : Grey Icon
- Condition ON/OFF : Green Icon
- Macro: M
- Delay ON: Red D
- Delay OFF: Blue D
- Pulse: P
- Level Mode ADC Port : Green

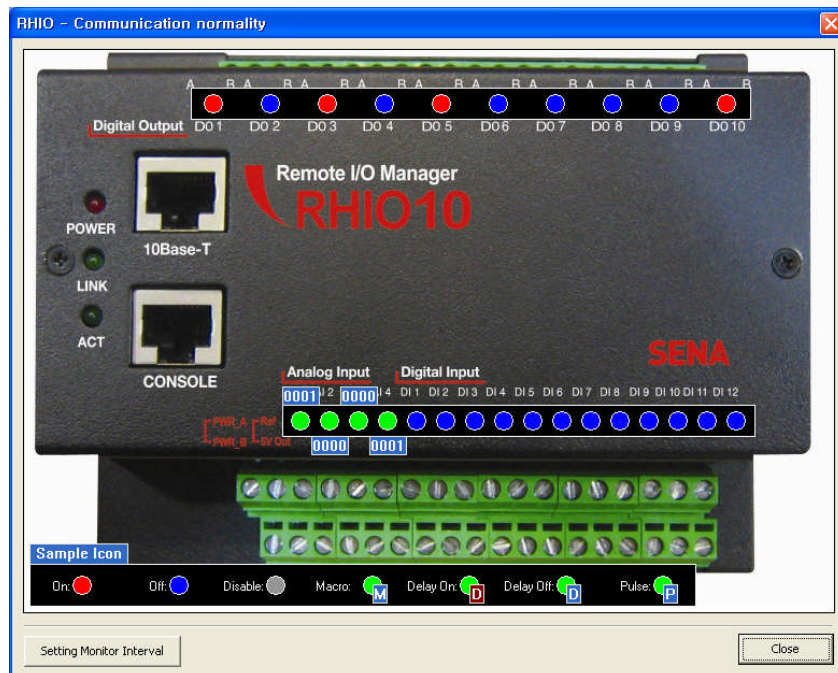


Figure 4-1 The I/O port Control screen



## 4.1.2 Specifying monitoring interval

You can continue to monitor the I/O state of Rhio10 at a specified time interval by setting [Setting Monitor Interval]. The valid value for monitoring intervals is any number between 2 and 10 seconds. In the specified time interval, Rhio Manager sends a state request command and receives a response from Rhio10 and displays it on the screen.

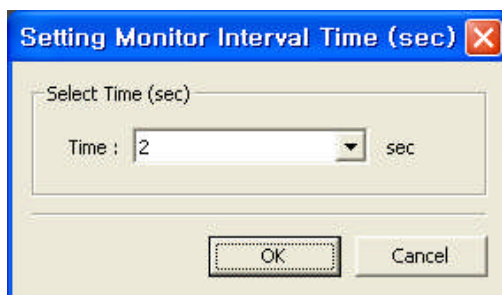


Figure 4-2 Setting Monitoring Interval

## 4.1.3 Monitoring I/O port

Upon receiving a state request command from Rhio Manager, Rhio10 returns information on the overall states of the digital input, analog input and digital output ports.

- Digital Input Port shows the ON/OFF state of input.
- In Level mode, ADC Input Port converts the analog value retrieved to a digital value in 1,024 steps and displays the converted value ("0000"- "1023").
- In Switch mode, ADC Input Port compares the input value to a specified threshold value and displays ON if it is higher and OFF if not.
- Digital Output Port displays the ON/OFF state, operation condition for an output port, and standby state.

## 4.1.4 Controlling digital output port

You can place your mouse over the Digital Output Port LED of Rhio Manager and left-click it to control ON/OFF state.

- When the operation condition for a digital output port is not specified, the ON/OFF state for the port is toggled each time you left-click your mouse.
- When the operation condition is specified, the port is set to ON if it is met, and it is set to OFF and displayed as a standby state if not.

- After output control is completed, Rhio10 returns the states of all ports to the host computer.

## 4.2 Digital Input Setting

Rhio10 has 12 digital input ports. You can enable/disable each of these digital input ports with Rhio Manager or by issuing the commands specified in "Ch. 5 I/O Port Related Protocols".

When setting the digital input with Rhio Manager, place your mouse cursor over the Digital Input LED on the I/O Port Control screen and right-click it to display the Setting window.

### 4.2.1 Setting Enable/Disable



Figure 4-3 Setting digital input

Digital input setting has two options: Enable and Disable. After selecting either of the options, press the [Setting] button to apply it in the system. A port cannot be set while it is operating in Run mode. Therefore, Rhio Manager sends a command that switches its mode to Setting mode first and then issues the set command when its operation is stopped.

When set to Enable, Rhio Manger receives the ON/OFF state from the device connected to a digital input port and then displays it. When set to Disable, it displays Disable regardless of the ON/OFF state of the device connected to the input port.

## 4.3 Digital Output Port Setting

Rhio10 has 10 digital output ports. You can set each of these digital output ports with Rhio Manager or by issuing the commands specified in "Ch. 5 I/O Port Related Protocols". Place your mouse cursor over the Digital Output Port LED on the I/O Port Control screen and right-click it to display the Setting Output Port window.

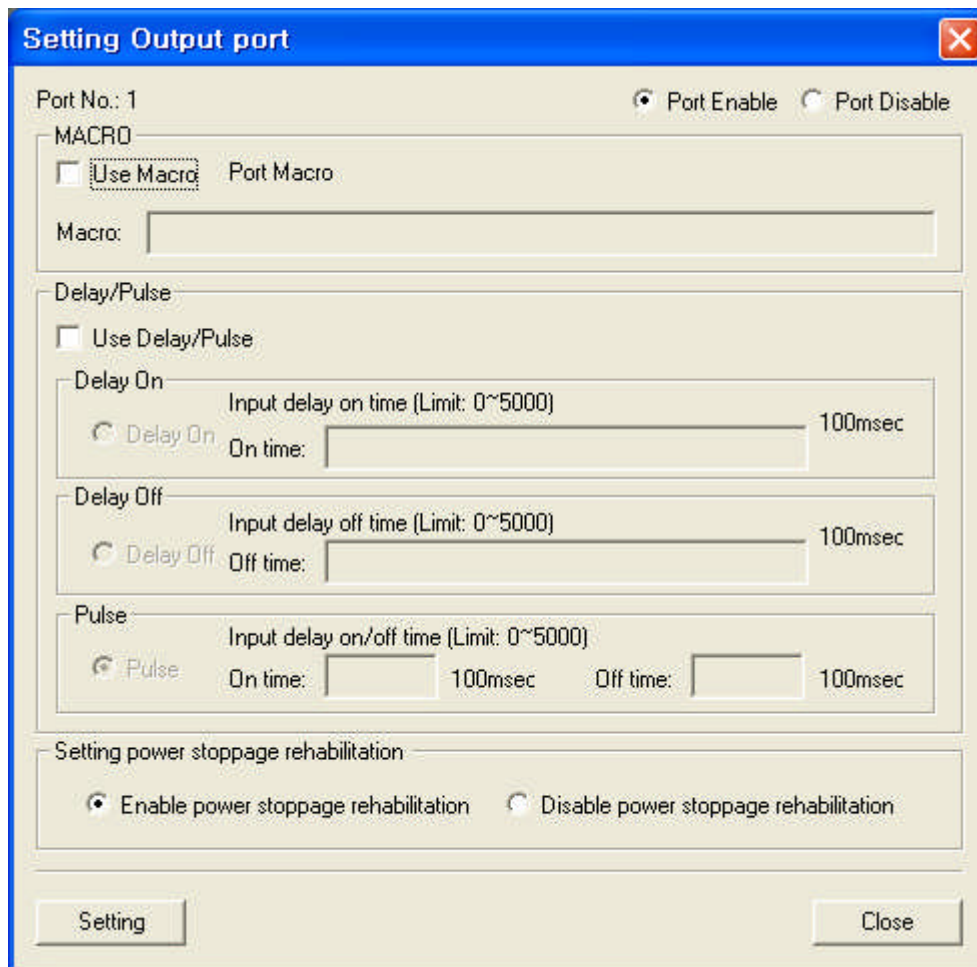


Figure 4-4 Setting digital output port window

### 4.3.1 Setting Enable/Disable

Set a specified port to Enable/Disable by selecting the [Port Enable] and [Port Disable] check boxes. If it is set to Disable, the digital output port becomes fixed to OFF.

### 4.3.2 Setting run condition

When the [Macro] check box is selected, you can enter a conditional expression and then use it to control operation of an output port. Enter the desired conditional expression in the [Macro] box.

- Specify an operation condition for each output port.
  - An output port that has not been set is regarded as a port available for direct control.
  - The final value obtained from a logical operation on the listed expression becomes the state of an output port.
  - If the operation condition expression is cleared, a port becomes available for direct access.
  - Operation condition expressions can be specified as follows:
    - 1) Port No. + Logic Expression (&,|) + Port No.
    - 2) Logic Expression (!) + Port No.
    - 3) Port No. + Logic Expression (&,|) + Logic Expression (!) + Port No.  
(Logical operator "!" can only be effective before the relevant port no.)
- ④|) -. Input #1 AND Input #2  
I1&I2
- . Input #1 OR Output #2  
I1|O2
- . The Inverse of Input #1  
!I1
- . Output #2 AND Output #3 AND the inverse of Output #4  
O2&O3&!O4
- A single logical expression can contain up to 21 ports.
  - If a port is directly set to ON when its operation condition is not met, it goes into standby state.
  - A port goes into the ON state if its operation condition is met and into OFF and standby states if not.
  - If a port is directly set to OFF, it does not operate even if the operation condition is satisfied.

### 4.3.3 Delay & Pulse Operation

If the [Use Delay/Pulse] check box is selected, the output port executes Delay and Pulse operation. The setting value for Delay and Pulse can be entered in 100 ms.

- When Delay ON is selected, you can send the *Output Port ON* command to set an output port to ON after a specified delay time has passed.
- When Delay OFF is selected, you can send the *Output Port OFF* command to set an output port to OFF after a specified delay time has passed.
- When Pulse is selected, the port continues to toggle between ON and OFF according to the specified ON/OFF time.

### 4.3.4 Setting Power-out Post Recovery

Power-out Post Recovery can be set for an output port using the [Setting power stoppage rehabilitation] pane in the Setting window.

- If it is enabled, Rhio10 retains its output port states prior to power-out when power goes out and back on
- If Power-out Post Recovery is enabled for an output port that has been set with an operation condition, the port becomes ON when power is restored if the operation condition is satisfied.
- If it is disabled, the output port state becomes OFF when power goes out and back on.

## 4.4 ADC Input Port Setting

Rhio10 has 4 ADC input ports. You can set an ADC input port with Rhio Manager or by issuing commands via an I/O port protocol (refer to "*Ch. 5 I/O Port Related Protocols*").

When using Rhio Manager, place the mouse cursor over the ADC Port LED on the [I/O Port Control] screen and right-click it to display the Setting ADC port windows as shown below:

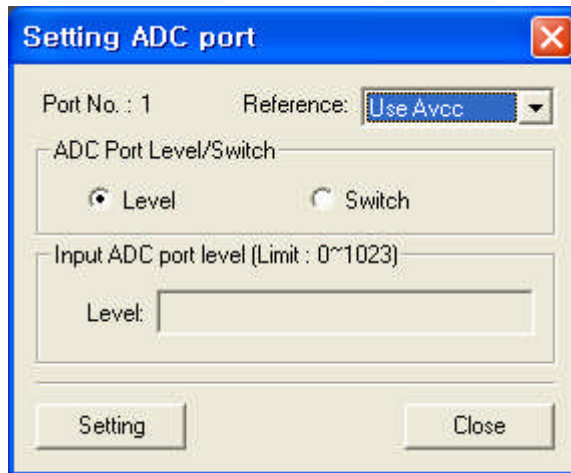


Figure 4-5 Setting ADC Input Port

#### 4.4.1 Setting Reference

Specify a reference voltage for analog input data. Rhio represents voltage from 0 V to a reference voltage in 1,024 steps. A reference voltage may not exceed 5 V. Specify a reference voltage in the Reference list box.

- **Use Avcc:** Specify Avcc (5V) as a reference voltage.
- **Use inside:** Specify the internal reference voltage (2.56V) as a reference voltage.
- **Use Outside :** Specify voltage issued to Aref as a reference voltage.

#### 4.4.2 Setting ADC Input Port

Analog Input Port has two modes: Level mode and Switch mode.

- **Level mode:** Display voltage from 0 V to a reference voltage in 1,024 steps (“0000” ~ “1023”).
- **Switch mode:** Compare input voltage level to threshold level setting and send a state change response when the input level is higher or lower than the threshold. An ADC input recognizes it is changed only when it is changed larger than 8 steps from the pre-configured threshold.

## 4.5 I/O Port Connection

This section describes how to make a required connection with the digital output, digital input and ADC input port of Rhio10 for the users' devices. The length of cables used for I/O port connection should be less than 3 meters to ensure normal operation.

### 4.5.1 Digital Output Port

All digital output ports are equipped with a status LED, which illuminates when a relay point is set to ON. An electric load can be connected as shown below by using OUTA1 and OUTB1 as driving switches.

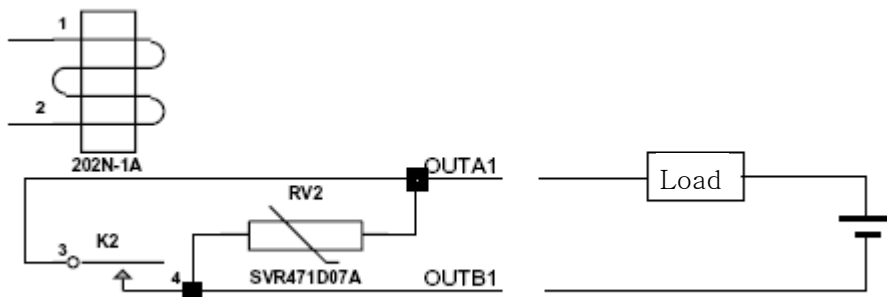


Figure 4-6 Connection of Digital Output Port

### 4.5.2 Digital Input Port

A digital input system operates regardless of the  $\pm$  polarity of the voltage and is insulated from the internal circuits in the system. It can be configured as shown in Figure 4-7 and has a status LED for each input, which illuminates when input voltage is issued.

Note)

Whilst it may operate in a voltage other than that specified (ON >  $\pm 3.2V$  / OFF <  $\pm 1.3V$ ), be sure to use the specified voltage to ensure the stable operation.

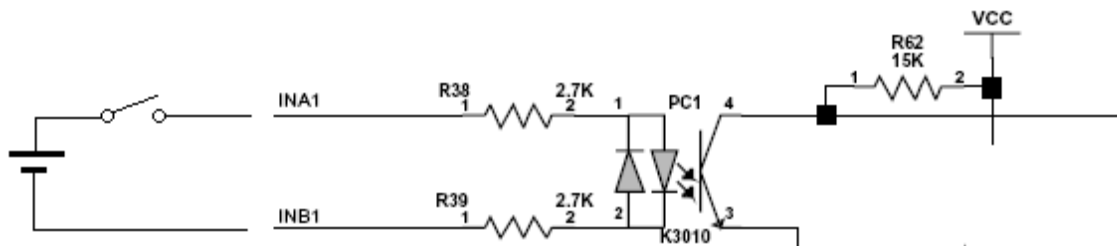


Figure 4-7 Connection of Digital Input Port

### 4.5.3 ADC Input Port

An ADC input port is a non-insulated input port. If possible, a circuit should be made using AVCC (+5V) voltage supplied from internal circuits. When operating in Switch mode, a threshold voltage should be specified. Input is set to ON and the ADC status LED is ON if the input voltage is higher than the specified threshold value. Conversely, input is set to OFF and the LED turns off when the input voltage is lower than the threshold. The ADC status LED operates only when it is set as Switch mode.

#### 1) Connecting reference voltage (AREF) in analog input

AREF can be set as either internal 2.56 V, internal AVCC or external AREF point. Internal input can be set using command. The external input can be set by splitting the AVCC voltage into R1 and R2 as shown below. The ideal resistance of split resistors R1 and R2 should be within the range of 1k $\Omega$  - 5k $\Omega$ .

Note) AREF voltage cannot be set to the value less than 2 V.

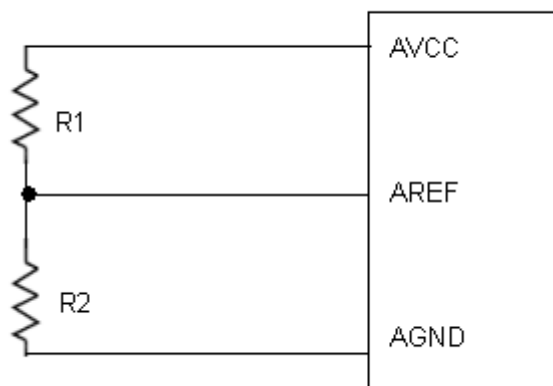


Figure 4-8 Connecting Aref



## 2) When using an Analog Input Potentiometer

An analog input operates in reference to input voltage and has an impedance of  $100\text{k}\Omega$ . When using a Potentiometer as shown in Figure 4-9, an impedance in the range of  $1\text{k}\Omega$  -  $5\text{k}\Omega$  is ideal. When using an external signal source, lower impedance ensures stable operation against various noises.

Note: Make sure that analog Input voltage does not exceed AREF voltage.

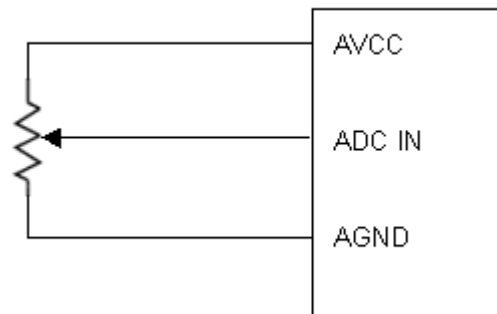


Figure 4-9 Circuit connection when a Potentiometer is used

## 3) Connecting when a voltage higher than AVCC voltage is used

For voltage input, the circuit should be split as shown in Figure 4-10. In case that the input wire is long or there is a strong noise nearby, it is recommended to have additional clamp diode in order to minimize the effect to other ADC channels although there is an internal clamp diode available.

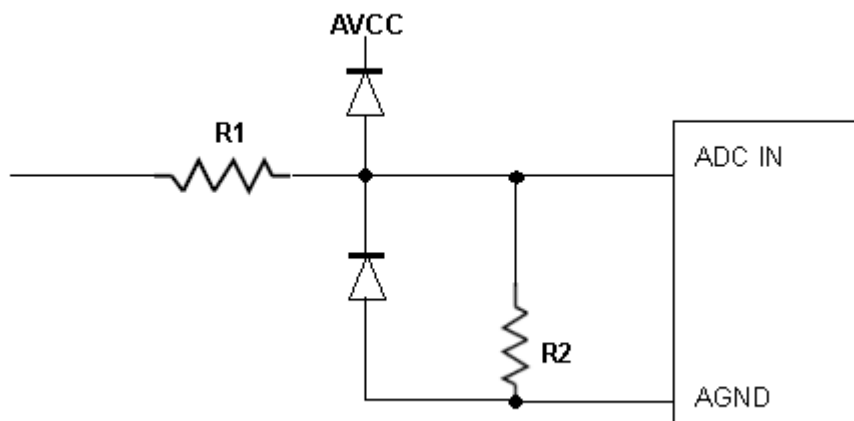


Figure 4-10 Circuit connection when a voltage higher than Avcc is used

## 5. Software Development & Application

You can use the Rhio library to develop application software that is used to communicate with Rhio10.

### The Rhio Library Files

`RHIO_Proc.dll`, `RHIO_Process.h`

You must link these two files in order to develop software using the Rhio library.

### Test Program

A sample test program that has been developed using the Rhio library is provided to you in the form of source and setup files (`RHIO_TEST_Setup.exe`). The test program shows developers how to utilize the Rhio library more easily.

## 5.1 The Rhio Library

### 5.1.1 Overview of the Rhio Library

The Rhio library is an MFC library that allows you to implement a communication protocol between Rhio and PC in a Windows environment. Since the Rhio library contains `CSocket` Class, it should be linked to a Microsoft Winsock component during program development. The `RHIO_CommProcessCreate` function must also be used to create `Process` Class for use of the library.

### 5.1.2 Reference

For definition of enumeration(s), structure(s) and function(s), refer to `RHIO_Process.h`.

#### 1) Enumeration (See Appendix D.1)

Type	Description
<code>EOnOffFlag</code>	ON/OFF - a flag indicating run state
<code>SendStateFlag</code>	A flag indicating transmission state of a command sent to Rhio
<code>ESetOutputFlag</code>	A flag related to enable/disable state when setting output port macro/delay/pulse
<code>EADCMode</code>	A flag specifying whether ADC is in Level mode or in Switch mode

2) Structure (See Appendix D.2)

Structure	Description
SADCDData	ON/OFF state of each port
SOnOffStateData	ON/OFF state of all ports
SSetOutput	Output port configuration information
SSetADC	ADC configuration information
SSetInput	Input port configuration information
SRHIOSetting	All ports configuration information

3) Function (See Appendix D.3)

Function	Description
<b>RHIO_CommProcessCreate</b>	A function that creates Process Class that must be created for use of the library.
RHIO_SockConnect	Connect to RHIO via a socket (TCP/IP).
RHIO_CommConnect	Connect to RHIO via a serial port.
RHIO_Close	Disconnect from RHIO.
RHIO_SndCmd_SetOnOff	Send a command that controls ON/OFF.
RHIO_SndCmd_GetOnOff	Send a command that checks ON/OFF setting.
RHIO_SndCmd_SetSettingMode	Send a command that sets setting mode.
RHIO_SndCmd_SetRunMode	Send a command that sets run mode.
RHIO_SndCmd_SetMACRO	Send a command that sets macro for input port.
RHIO_SndCmd_GetMACRO	Send a command that checks macro setting for input port.
RHIO_SndCmd_SetDelayPulse	Send a command that sets delay/pulse for input port.
RHIO_SndCmd_GetDelayPulse	Send a command that checks delay/pulse setting for input port.
RHIO_SndCmd_SetADC	Send a command that sets the level of all ADC ports (1-4).
RHIO_SndCmd_GetADC	Send a command that checks the level of all ADC ports (1-4).

RHIO_SndCmd_SetPortEnable	Send a command that enables/disables all ports.
RHIO_SndCmd_GetPortEnable	Send a command that checks enable/disable state of all ports.
RHIO_SndCmd_SetPwrStopEnable	Send a command that enables/disables power-out recovery for all ports.
RHIO_SndCmd_GetPwrStopEnable	Send a command that checks enable/disable state of power-out recovery for all input ports.
RHIO_SndCmd_SetFactoryReset	Send Rhio Factory Reset command.
RHIO_SndCmd_SetSerial	Send a command that sets Rhio serial number.
RHIO_SndCmd_GetSerial	Send a command that checks Rhio serial number.
RHIO_SndCmd_GetFirmware	Send a command that checks Rhio Firmware version.
RHIO_GetSettingData	Retrieve corresponding data when an event occurs such that a response on a check for setting is received from each port.
RHIO_GetOnOffData	Retrieve corresponding data when an event occurs such that responses from a change in and control of ON/OFF setting are received.

## 5.2 Creating and demonstrating a sample program with Rhio library

The sample program (RHIO\_TEST) is a dialog box based application that has been created with Microsoft Visual Studio .NET linked to **Rhio** library

(RHIO\_Proc.dll, RHIO\_Process.h).

### 5.2.1 Program UI Configuration and their related classes

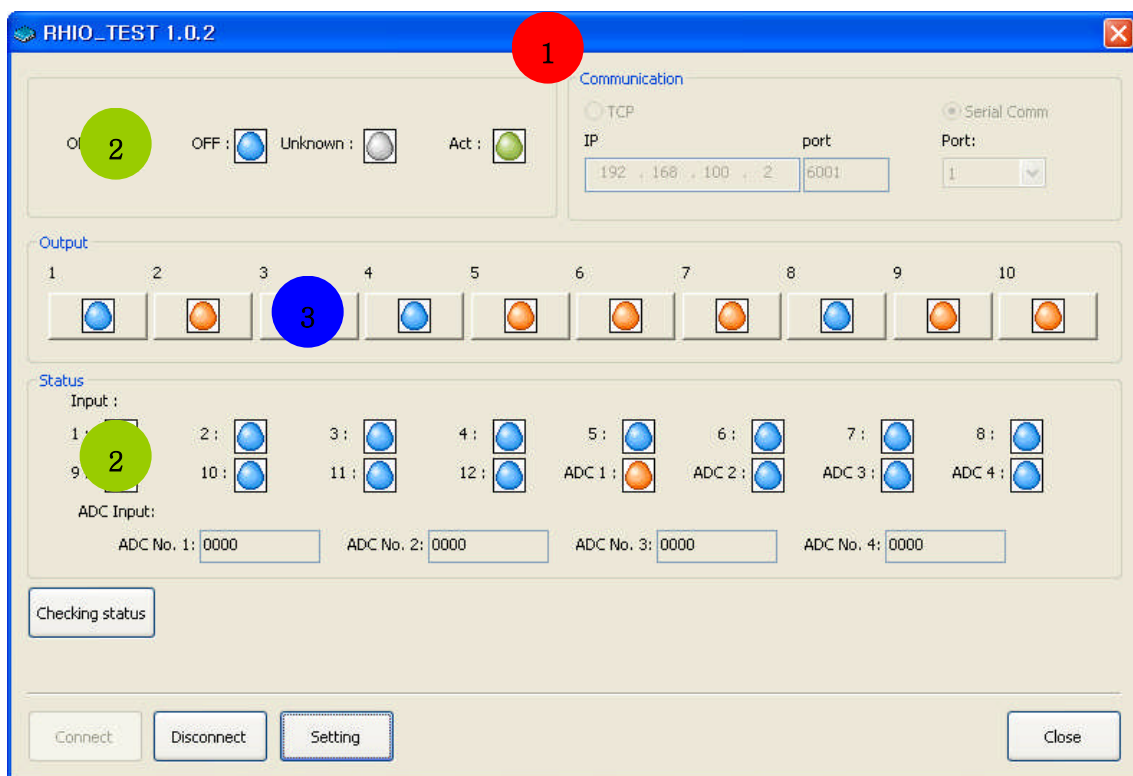


Figure 5-1 Main Window of a Sample Program

No.	Class	Related Files
1	<b>CWEB_IO_TESTDlg</b>	WEB_IO_TESTDlg.h, WEB_IO_TESTDlg.cpp
2	CStateWnd	StateWnd.h, StateWnd.cpp
3	COutputButton	OutputButton.h, OutputButton.cpp

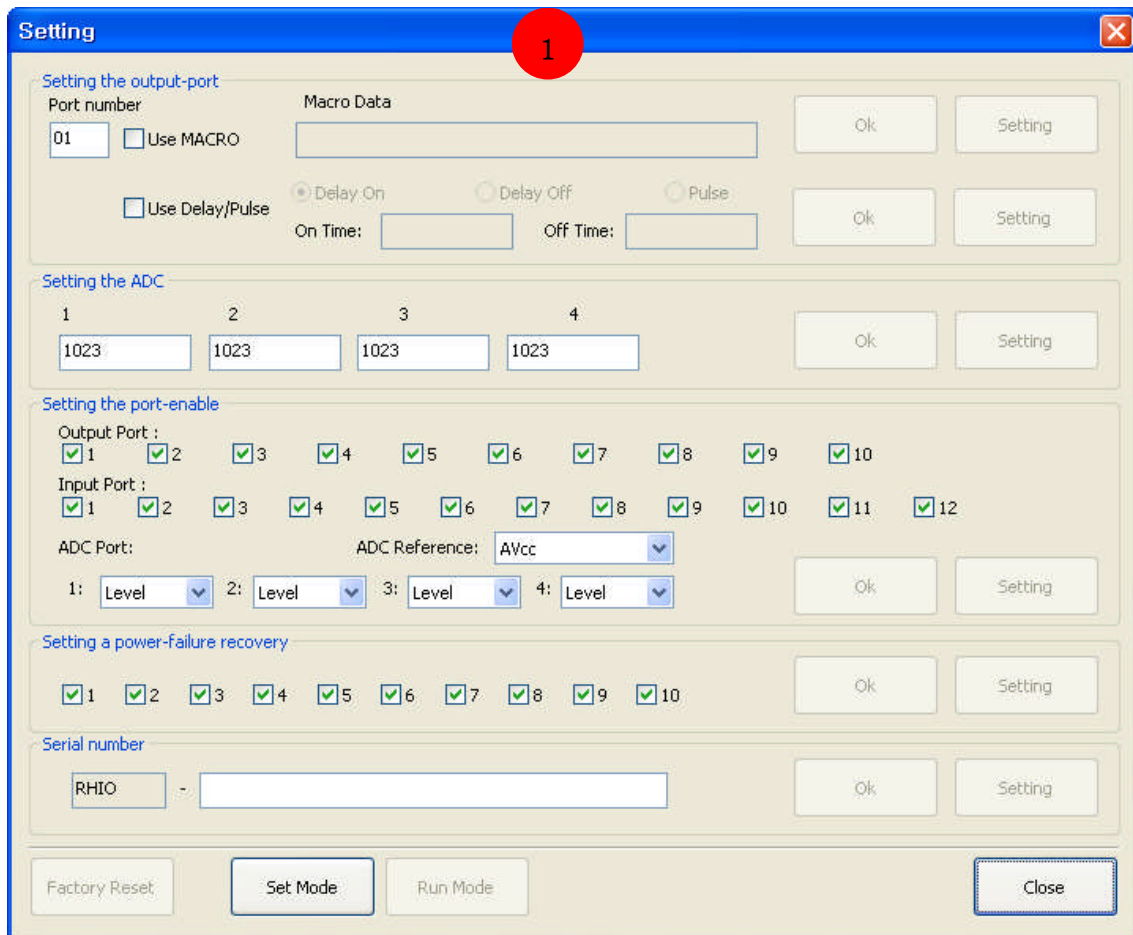


Figure 5-2 Setting Window of a Sample Program

No.	Class	Related Files
1	CSettingDlg	SettingDlg.h SettingDlg.cpp

## 5.2.2 Processing

### 5.2.2.1 Initializing Main window

- 1) Link the library and get the address of a required library function.

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

- 2) Initialize dialog items in Main window.

```
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();
```

3) Create Process Class.

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

### 5.2.2.2 Event Handling procedure

1) When an event occurs in the main window of the program:

- ① The Connect button is clicked on.

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

- ✓ TCP Connection

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

- ✓ Serial Connection

```
m_rhCommConnect (int iPort)
```

- ② The Disconnect button is clicked on.

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

- ③ The Set Button is clicked on.

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

- ④ The State View button is clicked on.

```
CWEB_IO_TESTDlg::OnBnClickedButtonStateView()
```

- ⑤ The Output Port button is clicked on.

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

2) When an event occurs in the Setting window of the program:

- ① The Factory Reset button is clicked on.

```
CSettingDlg::OnBnClickedButtonFactoryReset()
```

- ② The Set Mode button is clicked on.

```
OnBnClickedButtonSetmode()
```

- ③ The Run Mode button is clicked on.

```
CSettingDlg::OnBnClickedButtonRunMode()
```

- ④ The Monitor Serial button is clicked on.

```
OnBnClickedButtonMonitorSerial()
```

- ⑤ The Set Serial button is clicked on.

```
CSettingDlg::OnBnClickedButtonSetSerial()
```

- ⑥ The Monitor Power Stop button is clicked on.  
CSettingDlg::OnBnClickedButtonMonitorPwrStop()
- ⑦ The Set Power Stop button is clicked on.  
CSettingDlg::OnBnClickedButtonSetPwrStop()
- ⑧ The Monitor Enable State of the Port button is clicked on.  
CSettingDlg::OnBnClickedButtonMonitorEnable()
- ⑨ The Enable Port button is clicked on.  
CSettingDlg::OnBnClickedButtonSetEnable()
- ⑩ The Monitor ADC Input button is clicked on.  
CSettingDlg::OnBnClickedButtonMonitorInput()
- ⑪ The Set ADC Input button is clicked on.  
CSettingDlg::OnBnClickedButtonSetInput()
- ⑫ The Check Delay/Pulse State button is clicked on.  
CSettingDlg::OnBnClickedButtonMonitorOutput2()
- ⑬ The Set Delay/Pulse button is clicked on.  
CSettingDlg::OnBnClickedButtonSetOutput2()
- ⑭ The Check Macro Setting button is clicked on.  
CSettingDlg::OnBnClickedButtonMonitorOutput()
- ⑮ The Set Macro button is clicked on.  
CSettingDlg::OnBnClickedButtonSetOutput()

3) When an event occurs in a Rhio device:

```
CWEB_IO_TESTDlg::OnUpdateState(WPARAM wParam,
                                LPARAM lParam)
```

- ① A port ON/OFF event occurs.  
m\_rhGetOnOffData (SOnOffStateData &sOnOffData)
- ② Events other than a port ON/OFF occur.  
CSettingDlg::OnReceive(WPARAM wParam, LPARAM lParam)





- LRC (BCC)

Checks for any Errors in the command block.

Bitwise XORed value from **START FLAG** field to **DATA** field

(refer to example below)

- END FLAG

CR+LF ( 0x0D+0x0A )

- The data of LENGTH and LRC is expressed as follows.

Each 4-bit nibble (upper 4-bit nibble and lower 4-bit nibble) is expressed as 1 Byte data. The 1-byte conversion of the 4-bit data is as follows.

**0x0 ~ 0x9 → 0x30 ~ 0x39, 0xA ~ 0xF → 0x41 ~ 0x46**

If the Rhio receives the data converted, then it converts it to original by inverse.

- The timeout from the start of the frame to the time when it receives **LF** is 5 sec.

- Rhio will discard the command if there is any error in the command received (BCC Error or Time Out) and will wait for the next command (Users have to write a code so that it will have to wait at least 5 sec if there is no response from Rhio after sending the command.)

\* An Example of the LRC calculation is as follows :

XOR from Start Flag field to the Data Field use the following as an example :

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

Do the following :

(((3A xor 30) xor 33) xor 30) xor 33) xor 30 = 0A

0A gets translated as 0 and A, the byte conversion, which is as follows:

0x0 ~ 0x9 gets converted as 0 and A, the byte conversion, which is as follows :

0x0 ~ 0x9 gets converted to 0x30 ~ 0x39

0xA ~ 0xF gets converted to 0x41 ~ 0x46

In the case of above 0 =30 and A = 41, thus the LRC is 30 41, as shown above.

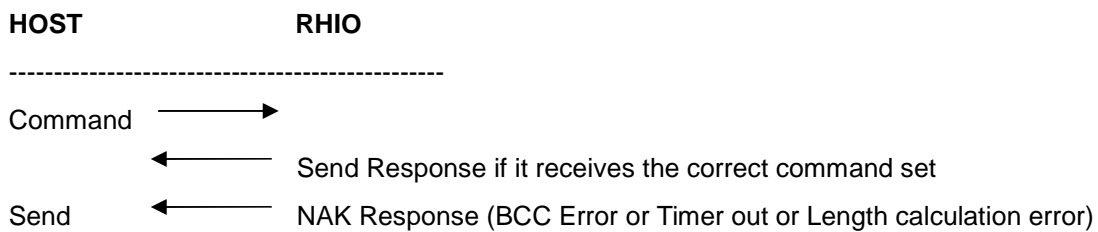
### 5.3.1.2 NAK Response

- NAK Response condition

- When there is Data BCC Error when receiving the command
- When the command data is not completed within 1 sec after it is started (Time out)
- When the length of the Data frame is not same as the one in the command  
(Frame Length error)

- NAK Response

	<b>START FLAG</b>	<b>LENGTH</b>	<b>FUNCTION</b>	<b>DATA</b>	<b>LRC (BCC)</b>	<b>END FLAG</b>
<b>Byte Size</b>	1	2	2	3	2	2
	Start of the Command Block, <b>0x3A</b> ( “:” )	The length from <b>FUNCTION</b> field to <b>DATA</b> field	Response “00” 0x30,0x30	NAK Data “NAK” 0x4E, 0x41, 0x4B	XOR from <b>START FLAG</b> field to <b>DATA</b> field	CR (0x0D) LF (0x0A)



- Users have to write code to send the command again or display NAK status if the program receives NAK Response.

\* **NOTE:** In the following descriptions, each port is indicated as below:

- Input Port: I1 - I12
- ADC Input: A1 - A4 (Level Input and Switch Input modes)
- Output Port: O1 - O10

### 5.3.2 ON/OFF Control

#### 5.3.2.1 ON/OFF Control Command

	Command	Data	Remark
Byte Size	2		
	“01” ( 0x30, 0x31 ) ON/OFF control	10-point output masking and ON/OFF control data	Send output points to set ON/OFF in a batch.

- Data

	MASK Data	Data Separator	ON/OFF Data
Data Order	1 - 10	11	12-21
Port No.	1 - 10	-	1-10
Data Content	Control: 0x31 Non-control: 0x30	0x2C ( “,” ) separates MASK from ON/OFF.	ON: 0x31 OFF: 0x30

MASK and ON/OFF Data correspond to one port per byte for each port in sequence.

- Port location by Data Order

Data	1	2	3	4	5	6	7	8	9	10
Order	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 Response for ON/OFF Control

	Response	Data	Remark
Byte Size	2	52	
	“02” ( 0x30, 0x32 ) <b>input/output</b> overall state	ON/OFF state data for ADC Level 4, Input 12 Point and Output 10 Point.	Send state of input/output points in a batch.

- Data

Field	Data Order	Port	Data Content
Control State	1	-	0x30: Normal Control, 0x31: No Run Mode, 0x39: Abnormal Control
ADC Input LEVEL	2-6	A1	0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode
	7	-	Field Separator 0x2C ( “,” )

	8-12	A2	0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode	"0000"-1023": Level
	13	-	Field Separator 0x2C ( " , " )	
	14-18	A3	0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode	"0000"-1023": Level
	19	-	Field Separator 0x2C ( " , " )	
	20-24	A4	0x30: OFF, 0x31: ON (Switch Input), 0x39: Level Mode	"0000"-1023": Level
	25	-	Field Separator 0x2C ( " , " )	
Input State	26-29	I1-I4	0x30: OFF, 0x31: ON	
	30	-	Field Separator 0x2C ( " , " )	
	31-34	I5-I8	0x30: OFF, 0x31: ON	
	35	-	Field Separator 0x2C ( " , " )	
	36-39	I9-I12	0x30: OFF, 0x31: ON	
	40	-	Field Separator 0x2C ( " , " )	
Output State	41-44	O1-O4	0x30: OFF, 0x31: ON, 0x32: Wait for a conditional execution, 0x33: Wait for Delay ON, 0x34: Wait for Delay OFF, 0x35: Run PULSE	
	45	-	Field Separator 0x2C ( " , " )	
	46-49	O5-O8	0x30: OFF, 0x31: ON, 0x32: Wait for a conditional execution, 0x33: Wait for Delay ON, 0x34: Wait for Delay OFF, 0x35: Run PULSE	
	50	-	Field Separator 0x2C ( " , " )	
	51,52	O9, O10	0x30: OFF, 0x31: ON, 0x32: Wait for a conditional execution 0x33: Wait for Delay ON, 0x34: Wait for Delay OFF, 0x35: Run PULSE	

- Indication that the system is standby when conditional execution for output port, pulse/delay operations are active (with ON command).

### 5.3.3 Input/Output State

#### 5.3.3.1 I/O State Request Command

	Command	Data	Remark
Byte Size	2	1	
	“03” ( 0x30, 0x33 ) State Request	0x30: Fixed to a dummy value	

#### 5.3.3.2 Response for I/O State Request Command

- Identical with ON/OFF control response.
- Sent even if the state of input port or ADC (in Switch Input mode) has been changed (based on a threshold voltage). A change of input state is recognized when the input value changed is maintained for 15 ms.
- In case of state response, the control state field returns 0x30 in Run mode and 0x31 in Setting mode.

### 5.3.4 Set/Run

#### 5.3.4.1 Set/Run Command

	Command	Data	Remark
Byte Size	2	1	
	“04” ( 0x30, 0x34 ) Set/Run	0x30: Set 0x31: Run	Switch between Set and Run Modes.

- Rhio10 returns a state response once after set command is received. It does not return any state response until it receives run command even if it is switched to Setting mode.
- When setting output port, ADC input port, Port Enable, Port Power-out Recovery or serial number, send set command first to switch it to Setting mode. (Check command is working in both Setting and Run mode.)
- Once each setting is completed, send run command to switch it to Run mode.
- When a run command is received, it returns a state response once and continues with the operation paused.

### 5.3.4.2 Response for Set/Run command

- Same as "5.3.2.2 Response for ON/OFF control".
- In Setting mode, Rhio10 returns a response once and it does not respond until run command is received.
- The control state field of the state response, Rhio10 returns 0x31 in Setting mode and 0x30 in Run mode.

### 5.3.5 Output Port Setting

#### 5.3.5.1 Set Output Port Command

Byte Size	Command	Data			Remark
	2	2	1	N	
	"05" ( 0x30, 0x35 ) Set output port.	Port No. "01" ~ "10"	0x30: Clear run condition 0x31: Set run condition 0x32: Clear Delay or Pulse 0x33: Set Delay/Pulse 0x39: Clear all settings	Setting Data (Max. 106 bytes)	Set the attribute of each output port. (When cleared, the setting data is 0x30 in 1 byte.)

- A setting can be done only in Setting mode.  
Users can set repeated run condition, pulse or delay. When a port is set for repeated actions, run condition set up has a priority, which means it is executed first. A setting can be cleared by mode, or the entire setting can be cleared as well. When cleared, data setting becomes 0x30 in 1 byte.

#### 1) Set run condition.

It sets the run condition for the output port specified.

(It is run only when an output port is ON and will set the port as OFF when it is configured as such.)

- Input/Output port state + a conditional operator + Input/Output port state (AND/OR)
- A conditional operator + Input/Output port state (NOT)
- AND => &, OR => |, NOT => !

- A conditional expression only takes the form of a single expression with AND, OR, NOT.

Eg. AND operation of Input #1 and Input #2 : I1 & I2

OR operation of Output #3 and Input #1 : O3 | I1

AND operation of the inversed Input #10 and Output #10 : !I10 & O10

Inverse operation of Input #1 : !I1

- The number of points that can be specified for run condition per 1 output point in setting data must be less than 21 points.

Eg. I1&I2&I3|I4|!I5|I6&I7&I8&I9&I10|!I11|I12|

O2|O3|O4|O5|O6|O7|O8|O9|O10

- An output port to be set should not be included in run condition.

Eg. When setting O1, it should not be included in its run condition expression.

## 2) Set Delay/Pulse.

- It specifies the output to toggle between ON/OFF at a given time or repeatedly. (Delay ON and Pulse are enabled with ON control command and disabled with OFF control. Delay OFF is enabled with OFF control.)

- Setting Data

	Setting Data (time)	
	ON Time	OFF Time
Byte Size	5	5
Run Setting	"00000"- "50000"	"00000"- "50000"

A setting can be specified in 100 ms and allowed up to 500 sec.

Eg.

ON Time	OFF Time	Remark
"00000"	"00000"	No Delay/Pulse
"00001"	"00000"	ON after 100 ms delay
"00020"	"00000"	ON after 2 sec delay
"50000"	"00000"	ON after 5,000 sec delay
"00000"	"00001"	OFF after 100 ms delay
"00000"	"00020"	OFF after 2 sec delay



"00000"	"50000"	OFF after 5,000 sec delay
"00001"	"00001"	Repeat 100 ms ON and 100 ms OFF.
"00020"	"00020"	Repeat 2 sec ON and 2 sec OFF.
"50000"	"50000"	Repeat 5,000 sec ON and 5,000 sec OFF.
"00010"	"00030"	Repeat 1 sec ON and 3 sec OFF.
"00300"	"00150"	Repeat 30 sec ON and 15 sec OFF.

### 5.3.5.2 Response for Set Output Port command

Byte Size	Response	Data				Remark
	2	1	2	1	N	
	"06" ( 0x30, 0x36 ) Return setting.	Setting Flag	Port No. "01" - "10"	0x30: Clear run condition 0x31: Set run condition 0x32: Clear Delay/Pulse 0x33: Set Delay/Pulse 0x39: Clear all settings	Setting Data	Return the setting of an output port.

- Setting Flag  
0x30: Set OK  
0x39: Set NG  
0x31: Not in Setting mode (when in Run mode)  
0x32: Unspecified (If the user attempts to clear in unspecified state, it will return 0x32.)  
(When it is in unspecified state, the setting data is 0x30 in 1 byte.)
- When setting is cleared, the setting data is 0x30 in 1 byte.

### 5.3.5.3 Check Output Port Setting Command

Byte Size	Command	Data		Remark
	2	2	1	
	"07" ( 0x30, 0x37 ) Check setting.	Port No. "01" - "10"	0x31: Check the run condition setting. 0x33: Check the Delay/Pulse setting	Check the attribute of each output port.

### 5.3.5.4 Response for Check Output Port Setting command

- Same as "5.3.5.2 Response for Set Output Port command".

## 5.3.6 ADC Input Port Setting

### 5.3.6.1 Set ADC Input Port Command

	Command	Data	Remark
Byte Size	2	16	
	"08" ( 0x30, 0x38 ) Set ADC.	Set the threshold level value for an input ADC port. ( "0000"- "1023" ) X 4	Set it to a 10-bit ADC level value.

- A threshold value is set for all ADC inputs in Switch Input mode.
- When an input change is measured, only changes beyond the range of +8 - -8 are recognized as an input change (based on the threshold level). It is determined as ON when an input voltage is more than the threshold level by 8 or more and OFF when an input voltage is less than the threshold level by 8 or more.

(When the threshold level is above 1015, a change between 1015-1023 is determined as ON. When it is set to less than 8, a change between 8-0 is determined as OFF.)

ADC Input No.	1	2	3	4
Settings	"0000"- "1023"	"0000"- "1023"	"0000"- "1023"	"0000"- "1023"

### 5.3.6.2 Response for Set ADC Input Port command

	Response	Data		Remark
Byte Size	2	17		
	"09" ( 0x30, 0x39 ) Return the ADC setting.	Setting FLAG	( "0000"- "1023" ) X 4	The threshold level for input ADC port

- Setting Flag
- 0x30: Set OK
- 0x31: Not in Setting mode (in Run mode)
- 0x39: Set NG

### 5.3.6.3 Check ADC Input Port Setting Command

	Command	Data	Remark
Byte Size	2	1	
	"10" ( 0x31, 0x30 ) Check the ADC setting	0x30: Fixed to a dummy value	Check the threshold level value of an input ADC port.

### 5.3.6.4 Response for Check ADC Input Port Setting command

- Same as "5.3.6.2 Response for Set Port command".
- The setting field is always OK (0x30).
- The factory default value in Level Input mode is set to "0000".

## 5.3.7 Port Enable Setting

### 5.3.7.1 Set Port Enable command

	Command	Data	Remark
Byte Size	2	34	
	"11" ( 0x31, 0x31 ) Set to Enable.	Port Enable/Disable setting data	Set all ports to Enable/Disable.

- Data

Field	Data Order	Port	Data Content
ADC Port	1	A1	0x31: Level Input mode 0x32: Switch Input mode
	2	A2	
	3	A3	
	4	A4	

	5	-	Field Separator 0x2C ( “,” )
ADC Reference Setting	6	-	0x30: AVcc ( Vcc 5V ) -> default 0x31: Internal ( 2.56V ) 0x32: External ( 2-4.5V )
	7	-	Field Separator 0x2C ( “,” )
Input Port	8-11	I1-I4	0x31: Enable, 0x32: Disable
	12	-	Field Separator 0x2C ( “,” )
	13-16	I5-I8	0x31: Enable, 0x32: Disable
	17	-	Field Separator 0x2C ( “,” )
	18-21	I9-I12	0x31: Enable, 0x32: Disable
	22	-	Field Separator 0x2C ( “,” )
Output State	23-26	O1-O4	0x31: Enable, 0x32: Disable
	27	-	Field Separator 0x2C ( “,” )
	28-31	O5-O8	0x31: Enable, 0x32: Disable
	32	-	Field Separator 0x2C ( “,” )
	33,34	O9, O10	0x31: Enable, 0x32: Disable

All I/O ports are initially set to Enable (including Factory Reset).

All ADC ports are initially set to Level Input mode (including Factory Reset).

### 5.3.7.2 Response for Set Port Enable command

	Response	Data	Remark
Byte Size	2	35	
	“12” ( 0x31, 0x32 ) Return the Enable setting.	Port Enable/Disable setting data	Return all port Enable/Disable settings.

- Data

Field	Data Order	Port	Data Content
Setting	1	-	0x30: Normal Enable, 0x31: Not in Setting mode 0x39: Abnormal Enable

ADC Port	2	A1	0x30: Avcc ( Vcc 5V ) -> default 0x31: Internal ( 2.56V ) 0x32: External ( 2V-4.5V )
	3	A2	
	4	A3	
	5	A4	
	6	-	Field Separator 0x2C ( “,” )
ADC Reference Setting	7	-	0x30: Avcc ( Vcc 5V ) -> default 0x31: Internal ( 2.56V ) 0x32: External ( 2V-4.5V )
Input Port	9-12	I1-I4	0x31: Enable, 0x32: Disable
	13	-	Field Separator 0x2C ( “,” )
	14-17	I5-I8	0x31: Enable, 0x32: Disable
	18	-	Field Separator 0x2C ( “,” )
	19-22	I9-I12	0x31: Enable, 0x32: Disable
	23	-	Field Separator 0x2C ( “,” )
Output State	24-27	O1-O4	0x31: Enable, 0x32: Disable
	28	-	Field Separator 0x2C ( “,” )
	29-32	O5-O8	0x31: Enable, 0x32: Disable
	33	-	Field Separator 0x2C ( “,” )
	34,35	O9, O10	0x31: Enable, 0x32: Disable

### 5.3.7.3 Check Port Enable Setting command

	Command	Data	Remark
Byte Size	2	1	
	“13” ( 0x31, 0x33 ) Check the Enable setting.	0x30: Fixed to a dummy value	Check all port Enable/Disable setting.

### 5.3.7.4 Response for Check Port Enable Setting command

- Same as "5.3.7.2 Response for Set Port Enable command".
- The setting field returns Normal (0x30) when it returns.
- The initial value is set to Enable for all ports at the time of shipment.  
(ADC is set to Level Input mode.)

### 5.3.8 Port Power-out Post Recovery Setting

#### 5.3.8.1 Set Port Power-out Post Recovery command

	Command	Data	Remark
Byte Size	2	12	
	"14" ( 0x31, 0x34 ) Set Power-out Post Recovery to Enable/Disable.	The Port Power-out Post Recovery setting data	Set Power-out Post Recovery to Enable/Disable for an output port

- Data

Field	Data Order	Port	Data Content
Output State	1-4	O1-O4	0x31: Enable, 0x32: Disable
	5	-	Field Separator 0x2C ( "," )
	6-9	O5-O8	0x31: Enable, 0x32: Disable
	10	-	Field Separator 0x2C ( "," )
	11, 12	O9, O10	0x31: Enable, 0x32: Disable

#### 5.3.8.2 Response for Set Port Power-out Post Recovery command

	Response	Data	Remark
Byte Size	2	13	
	"15" ( 0x31, 0x35 ) Return the Power-out Post Recovery setting.	The Port Power-out Post Recovery setting data	Return the Power-out Post Recovery setting for an output port.

- Data

Field	Data Order	Port	Data Content
Setting	1	-	0x30: Set OK, 0x31: Not in Setting mode, 0x39: Set NG

Output State	2-5	O1-O4	0x31: Enable, 0x32: Disable
	6	-	Field Separator 0x2C ( “,” )
	7-10	O5-O8	0x31: Enable, 0x32: Disable
	11	-	Field Separator 0x2C ( “,” )
	12, 13	O9, O10	0x31: Enable, 0x32: Disable

### 5.3.8.3 Check Port Power-out Post Recovery Setting command

	Command	Data	Remark
Byte Size	2	1	
	“16” ( 0x31, 0x36 ) Check the Power-out Recovery setting.	0x30: Fixed to a dummy value	Check the Power-out Recovery setting for all ports.

### 5.3.8.4 Response for Check Port Power-out Recovery Setting command

- Same as "5.3.8.2 Response for Set Port Power-out Recovery command".
- The setting field returns Normal (0x30) when it returns.

The initial value is set to Enable for all ports at the time of shipment.

## Appendix A. Connection

### A.1 Ethernet Pin Outs

The RHIO10 uses standard Ethernet connector, which is a shielded connector compliant with AT&T258 specifications. Table A-1 shows the pin assignment and the wire color.

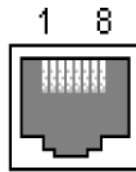


Figure A-1. Pin layout of the RJ45 connector

Table A-1. Pin assignment of the RJ45 connector

Pin	Description	Color
1	Tx+	White with orange
2	Tx-	Orange
3	Rx+	White with green
4	NC	Blue
5	NC	White with blue
6	Rx-	Green
7	NC	White with brown
8	NC	Brown

### A.2 Console Port Pin Outs

The pin assignment of the RHIO10 RJ45 connector is summarized in Table A-2. Each pin has a function according to the serial communication type configuration.

Note : When connecting console port to host, user should use the other end of CAT5 cable using RJ45 to DB9 Female connector(included package).

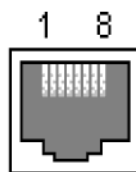


Figure A-2. Pin layout of the RJ45 connector



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

Table A-2. Pin assignment of the RJ45 connector

### A.3 Ethernet Wiring Diagram

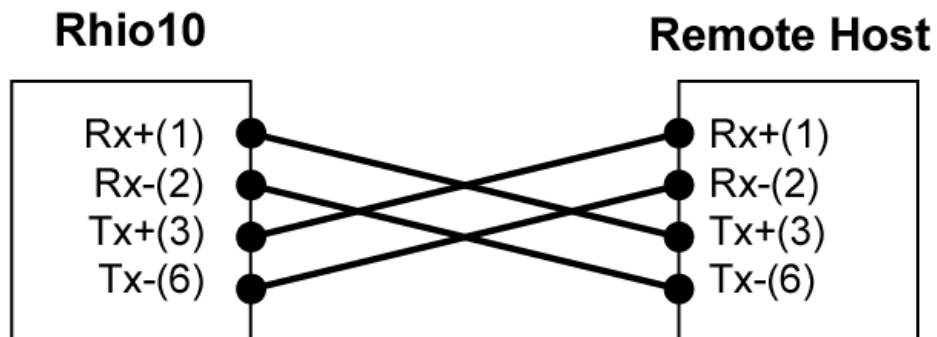


Figure A-3. Ethernet direct connection using crossover ethernet cable

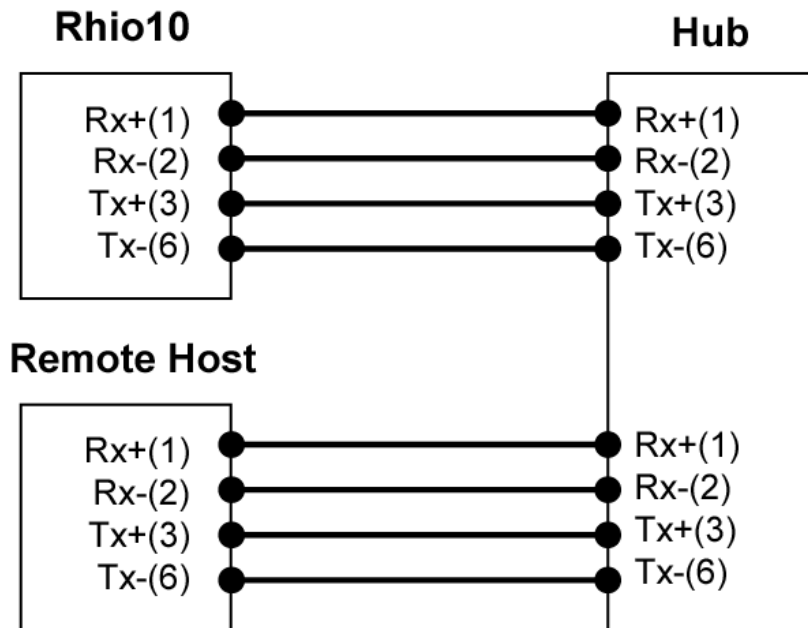


Figure A-4. Ethernet connection using straight through ethernet cable

### A.4 Serial Console Wiring Diagram

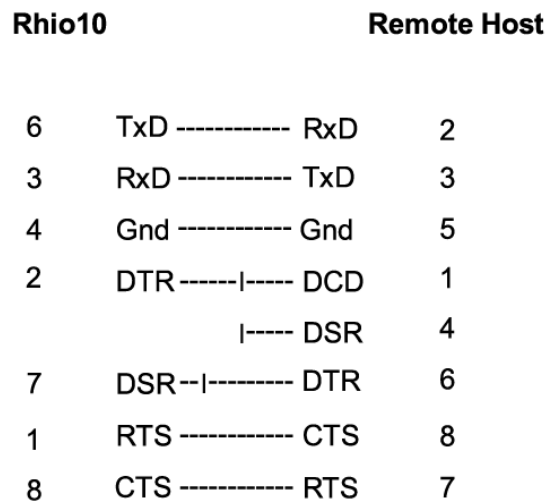


Figure A-5. RS232 wiring diagram

## Appendix B. Accessing Console Port and Command Usage

### B.1 Accessing Console Port

There are two ways to access console port of the RHIO10 depending on whether the user is located at a local site or a remote site.

- Serial console: Local users can connect directly to the serial console port of the RHIO10 using serial console/data cable (null-modem cable). The serial port of the RHIO10 is used as the console port as well as the data port. To use the serial port as the console port, slide Data/Console switch to the Console side.

Note : When connecting console port to host, user should use the other end of CAT5 cable using RJ45 to DB9 Female connector(included package).

- Remote console: Remote users can make a telnet connection to the remote console port (port 23) of the RHIO10 via TCP/IP network.

Both methods require the user to log into the RHIO10 in order to continue.

#### B.1.1 Using Serial Console

Connect the one end of the serial console cable to the console port of the RHIO10.

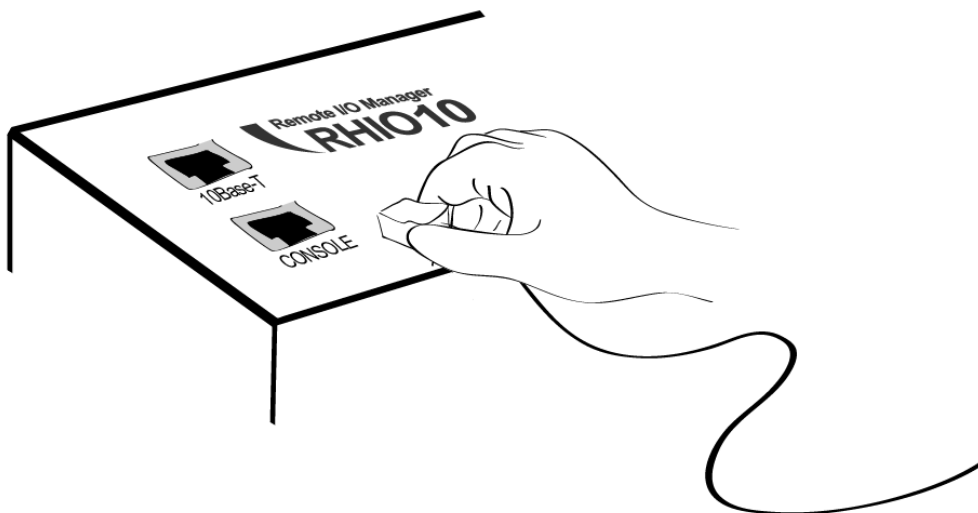


Figure B-1. Connecting a serial console cable to the RHIO10

Connect the other end of the cable to the serial port of user's computer.  
Slide Data/Console switch to Console side.

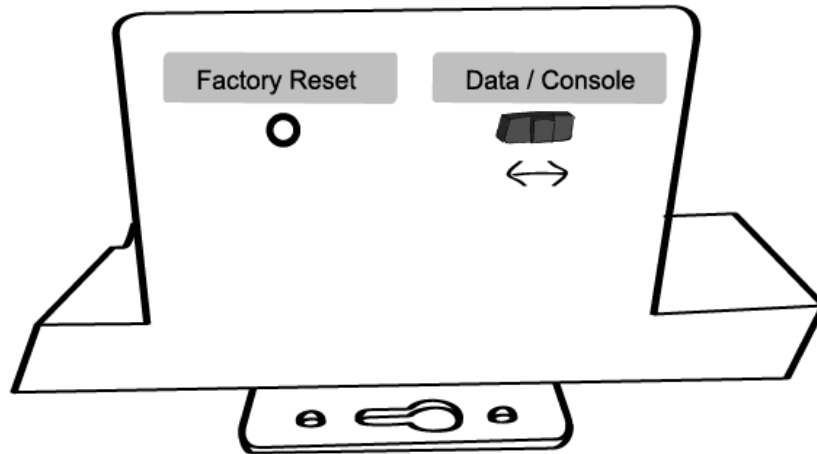


Figure B-2. Data/Console switch of the RHIO10

Run a terminal emulator program such as HyperTerminal. Set up the serial configuration parameters of the terminal emulation program as follows:

**9600 Baud rate, Data bits 8, Parity None, Stop bits 1, Hardware flow control**

- 1) Press [ENTER] key.
- 2) Type the user name and password to log into the RHIO10. A factory default setting of the user name and password are both **admin**.
- 3) If the user logged into the RHIO10 successfully, command prompt screen will appear on the computer as shown in Figure B-3.

```
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
>

```

Figure B-3. The RHIO10 console screen

From the command prompt screen, users can set, get and save configuration parameter values using 'set', 'get' and 'save' command. Users also can exit the console or reboot the device using 'exit' and 'reboot' command. The usage of the commands can be found using 'help' command. For command usages description, please refer to section *B.2 Command usage*.

### B.1.2 Using Remote Console

The RHIO10 provides remote console feature via telnet as well as serial console so that users can access the RHIO10 at remote site for configuration and monitoring purpose. The IP address of the RHIO10 must be known before users can access the remote console port. The port number for the remote console is 23, which is a TCP port number assigned for Telnet.

Only one user can log into the remote console or serial console at a time. If the serial console is established while a remote console is established, current remote console will be halted and no more remote console will be established until serial console is finished.

To access remote console of the RHIO10, please use the step as follows:

- 1) Run a telnet program or a program that supports telnet functions such as TeraTerm-Pro or HyperTerminal. The target IP address and the port number should be those of the RHIO10. If required, specify the port number as 23.

Type the following command in the command line interface of your computer.

```
telnet 192.168.1.254
```

Or run a telnet program with parameters as follows.

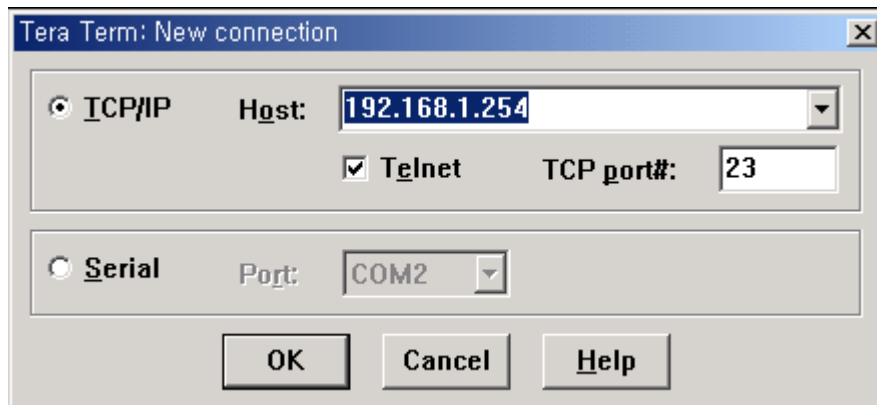


Figure B-4 Telnet program set up example

- 2) The user has to log into the RHIO10. Type the user name and password. A factory default setting of the user name and password are both **admin**.
- 3) If the user logged into the RHIO10 successfully, the same command prompt screen as the one of serial console will be displayed. The user can set, get, save configuration parameters and exit console, reboot the device as like the serial console.
- 4) If serial console or the other remote consoles are connected already, new console will not be established at all.

## B.2 Command Usage

The RHIO10 provides several simple commands for configuration and control of the RHIO10. Table B-1 summarizes command set which RHIO10 supports.

*Table B-1 RHIO10 command set summary*

Command	Description	Result
set group par1 [par2 ...] + <CR>	Set configuration parameters - group = 'ip', 'host', 'serial' or 'admin' - par1 ... = configuration parameters. Use * to keep a parameter's value	If success, "OK" + <CR> + <LF> If error "ERROR" + <CR> + <LF>
get [group] + <CR>	Get configuration parameter values - group = 'ip', 'host', 'serial', 'admin' or 'status' - If group is specified, shows settings of the group. - If group is omitted, shows settings of all groups.	Parameter value display
help [group] + <CR>	Shows command usage screen. - If group is omitted, shows help screen. - If group is specified, shows 'set' command usage of the group.	Help message display
factorydefault [option] + <CR>	Restore factory default values - if option is omitted, all parameters are set with factory default values.	If success, "OK" + <CR> + <LF> If error

	- if option='-ip', all parameters except IP settings are set with factory default values.	"ERROR" + <CR> + <LF>
save + <CR>	Save changes	If success, "OK" + <CR> + <LF> If error "ERROR" + <CR> + <LF>
exit + <CR>	Exit without rebooting the device (changes are not applied)	If success, "OK" + <CR> + <LF> If error "ERROR" + <CR> + <LF>
reboot + <CR>	Exit and reboot the device	None

### B.2.1 'set' Command

With 'set' command, users can configure parameter values of the RHIO10 for each environment. Basic 'set' command usage is as follows:

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

**where,**

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

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

The 'group' is the category where the parameters should be entered. For example, if users want to set parameters related to the IP configuration, use set command as shown in the Figure B-5.

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

*Figure B-5 IP configuration example screen*

In the above example, the first parameter 'ip' indicates that the following parameters are IP configuration parameters. The second parameter 'static' indicates that the RHIO10 will use static IP address of the third parameter '192.168.1.100'. The fifth parameter indicates the subnet mask and the next indicates the default gateway IP address.

If users want to change only one of the parameters of the group, users can omit trailing parameters and/or can use '\*' to keep a parameter value. The screen below will show how to change subnet mask only without changing IP address and gateway IP address.

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

*Figure B-6 Changing only one parameter value example*

Command usage of 'set' will differ depending on the groups. Each 'set' command usage of the group can be found using 'help group' command. For example, if users want to know how to use 'set' command to configure IP configuration, typing 'help ip' + <CR> will show 'set' command usage for the IP configuration as shown in Figure B-7.

```
> 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
>
```

*Figure B-7 Help screen example*

**Note:**

The changed values will not take effect until 'save' and 'reboot' commands are invoked. For more details, please refer to section B.2.4. ~ B.2.6.

## **B.2.2 'get' Command**

With 'get' command, users can view the current parameter values and status of the RHIO10. Basic 'get' command usage is as follows:

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

***where,***

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

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

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



The group means the category where parameters belong to as like 'set' command. For example, if users want to view parameter values related to IP configuration, get command as Figure B-8 can be used.

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

*Figure B-8 Getting ip configuration screen*

'status' group is a special group where 'set' command does not apply. 'get status' will show current system status screen.

```
> get status
Serial_no.: Rhi010-0207_test
MAC_address: 00-01-95-77-88-99
F/W_REV.: V1.2.0
Current_IP: 192.168.0.125
>
```

*Figure B-9 Getting status screen*

If group is omitted, get command will show all of the parameter values as shown in Figure B-10.

```
> get
--- Status ---
Serial_no.: Rhi010-0207_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
>
```

*Figure B-10 Getting all parameters screen*

## B.2.3 'help' Command

With 'help' command, users can find command usage help in the console screen. Basic command usage is as follows:

**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.**

Figure B-11 shows help screen when no group is specified while Figure 2-16 shows help screen with 'ip' group specified.

```
> 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
```

*Figure B-11 Help screen*

```
> 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
```

*Figure B-12 Help screen with 'ip' group specified*

## B.2.4 'save' Command

With 'save' command, current parameter changes are saved to non-volatile memory. Command usage of 'save' command is as follows:

***save + <CR>***

Saved changes will be applied if the RHIO10 is rebooted by 'reboot' command or manual rebooting.

## B.2.5 'exit' Command

With 'exit' command, current serial or remote console session will be closed. However, changed parameters are not applied until the RHIO10 is manually rebooted. Command usage of 'exit' command is as follows:

***exit + <CR>***

## B.2.6 'reboot' Command

With 'reboot' command, the RHIO10 will be rebooted immediately. Changed parameter values will be applied when the RHIO10 is up again. Command usage of 'reboot' is as follows:

***reboot + <CR>***

## B.3 System Configuration using Console command

### B.3.1 Network Configuration

Basic 'set' command usage for IP configuration is as follows:

***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**

To configure IP configuration parameters of the RHIO10, use set command as follows:

**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**

To apply changes, users have to invoke 'save' and 'reboot' command after changing configuration.

```
> 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 Host Mode Configuration**

Basic 'set' command usage for host mode configuration is as follows:

**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*

To configure the RHIO10 to work as a TCP server, use 'set' command as follows:

**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
>
```

To configure the RHIO10 to work as a TCP client, use set command as follows:

**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
>
```

To configure the RHIO10 to work as a TCP server/client mode, use 'set' command as follows:

***set host tcpsc 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.***

The parameter of the TCP Server/Client mode is same as the ones of the TCP server mode and client mode

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

### **B.3.3 System Administration**

Users can configure administrator username, password and device name using set command as follows:

***set admin username password devicename***

***username: login username***

***password: login password***

***devicename: device name***

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

## Appendix C. Troubleshooting

### C.1 Power/LED Status Troubleshooting

Problem	Cause	Action
Power LED does not light up	Power cable is not connected	Check power connection. Rhio10's power specification is DC 9V~48V.
Link LED does not light up	Ethernet cable is not connected	Check Ethernet cable connection
	Invalid Ethernet cable is used	There are two types of Ethernet cables: Straight-through cable and crossover cable. If you are using an Ethernet hub, use straight-through cable. If direct connection between the RHIO10 and remote host is used, use crossover cable instead.
ACT LED does not blink	Invalid IP configuration	Check IP configuration parameters

### C.2 Serial Console Troubleshooting

Problem	Cause	Action
Serial console is not connected	Invalid serial cable	Be sure to use a serial console cable (null-modem cable) for serial console
	Invalid serial port configuration of terminal emulation program	Check serial port configuration of terminal emulation program: 9600 bps, 8 Data bits, No parity, 1 stop bit, Hardware flow control
	Invalid Console/Data switch position	Be sure that Console/Data switch position is Console side.
Cannot login to console	Invalid username and/or password	Use valid username and password. If username and/or password are lost, perform factory default reset using factory reset switch. Factory default value of username and password are both <i>admin</i>

### C.3 Remote Console Troubleshooting

Problem	Cause	Action
Cannot connect to the RHIO10 using telnet	The RHIO10 is not assigned valid IP address	Use serial console to assign valid IP address to the RHIO10
	Someone is using serial console	Exit serial console and retry telnet connection
Cannot login to console	Invalid username and/or password	Use valid username and password. If username and/or password are lost, perform factory default reset using factory reset switch. Factory default value of username and password are both <i>admin</i>

### C.4 IP Address Troubleshooting

Problem	Cause	Action
Cannot find IP address of the RHIO10		Use serial console to find IP address
		Use HelloDeviceManager program to probe the RHIO10 on the network

### C.5 DHCP Troubleshooting

Problem	Cause	Action
Cannot lease IP address	DHCP server is not working	Check if DHCP server is working correctly
IP address of the RHIO10 is changed	DHCP server does not extend lease time	Check if DHCP server is working correctly



## Appendix D. Rhio Linrary

### D.1 Enumeration Type

#### **EOnOffFlag**

Flag for ON, OFF, Operation condition

```
enum EOnOffFlag
{
    EOF_ON,           //ON status
    EOF_ON_ADC_LEVEL, //ADC Level ON status
    EOF_OFF,          //OFF status
    EOF_NOT,          // Status uncertain status
    EOF_ON_DELAY,     //Delay ON waiting status
    EOF_OFF_DELAY,    //Delay OFF waiting status
    EOF_ONOFF_MACRO,  //Macro(condition) waiting status
    EOF_ONOFF_PULSE   //Pulse status
};
```

#### **SendStatusFlag**

Transfer status flag for indicating which command is sent to Rhio10

```
enum SendStatusFlag
{
    ESF_ONOFF,        //ON, OFF control command transfer status
    ESF_SET,           //
    ESF_SET_MODE,     // Set mode switch command transfer status
    ESF_RUN_MODE,     //operation mode change command transfer status
    ESF_SET_MACRO,    //Macro Set Command transfer status
    ESF_SET_DELAY_PULSE, //Delay/Pulse Set Command transfer status
    ESF_SET_ADC,      //ADC Set Command transfer status
    ESF_SET_ENABLE,   // Port Enable/Disable Set command transfer
                    // status
    ESF_SET_PWR_STOP, // Power-out Post Recovery Enable/Disable Set
                    // Command transfer status
};
```

```

    ESF_SET_FACTROT_RESET, //Factory Reset Command transfer status
    ESF_SET_SERIAL,        //Serial number Set Command transfer status
    ESF_MON_MACRO,         //Macro check Command transfer status
    ESF_MON_DELAY_PULSE,  //Delay/Pulse check Command transfer status
    ESF_MON_ADC,           //ADC check Command transfer status
    ESF_MON_ENABLE,        //Enable/Disable check Command transfer status
    ESF_MON_PWR_STOP,      //Power-out Post Recovery check Command transfer
                          //status
    ESF_MON_SERIAL,        //Serial number check Command transfer status
    ESF_MON_FIRMWARE,      //Firmware Version check transfer status
    ESF_NONE
};

```

### **EsetOutputFlag**

Output Port Maco/Delay/Pulse Set/Clear Flag

```

enum ESetOutputFlag
{
    ESOF_CLEAR,           // Port Set Clear
    ESOF_SETTING          // Port Set
};

```

### **EADCMode**

Flag specifies whether the current ADC mode is Level or Switch mode

```

enum EADCMode
{
    EAM_LEVEL,
    EAM_SWITCH
};

```

## D.2 Structure

### ON/OFF status Data

Structure which contains Port ON/OFF status

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

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

### Set status Data

Structure for the Set status

```
typedef struct _SetOutput        //Output Port Set 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 Set Data
{
    EADCMode eModeADC[4];      //ADC port Mode
    BYTE bReference;           //ADC Reference
}
```

```

        SADCData sADCData[4];    //ADC Level Data
}SSetADC;

typedef struct _SetInput        //Input Port Set Data
{
    bool isEnabled[12];        //Input port Enable
}SSetInput;

typedef struct _RHIOSetting    //All Port Set Data
{
    SSetOutput sOutput;        //Setting Output port Data;
    SSetADC sADC;              //Setting ADC port Data;
    SSetInput sInput;          //Setting Input port Data;
    char cGetSerial[18];        // Model name and Serial number
                                // received
    char cGetFirmVer[9];        // Firmware Version received
}SRHIOSetting;

```

## D.3 Function

### **RHIO\_CommProcessCreate**

**Description :** It creates Process Class of RHIO\_Proc Dll. In order to use Rhio Dll, be sure to create the process using RHIO\_CommProcessCreate.

**Function Prototype :**

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

**Parameter :**

CWnd \*pParentWnd : Window Pointer to Parent window of Process Class

**Return :** None

**eg. :** RHIO\_CommProcessCreate(this)

## **RHIO\_SockConnect**

**Description :** Used when users want to connect to RHIO using TCP Socket connection.

### **Function Prototype :**

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

### **Parameter :**

BYTE bAddr1 : 1st byte of the IP Address  
BYTE bAddr2 : 2nd byte of the IP Address  
BYTE bAddr3 : 3rd byte of the IP Address  
BYTE bAddr4 : 4th byte of the IP Address  
int iPort : Port number.

**Return :** TRUE if successful, False if failed.

### **eg. :**

```
if(RHIO_SockConnect(192, 168, 100, 2, 6001))  
    AfxMessageBox("Connection Successful");  
else  
    AfxMessageBox("Connection Failure");
```

## **RHIO\_CommConnect**

**Description :** Used when users want to connect to RHIO using RS232 serial connection.

(Parameter fixed as Baudrate:9600 bps, Parity: none, Data bit: 8, Stop bit: 1)

### **Function Prototype :**

```
extern "C" __declspec(dllexport) bool RHIO_CommConnect (int iPort);
```

**Parameter :** None

**Return :** TRUE if successful, False if failed.

### **eg. :**

```
if(RHIO_CommConnect(1))  
    AfxMessageBox("Connection Successful.");  
Else  
    AfxMessageBox("Connection Failure.");
```

## **RHIO\_Close**

**Description :** Used when users want to disconnect the current connection with the Rhio in both serial and TCP connection. In order to reconnect to the Rhio, be sure to use Rhio-Close function before connection attempt.

### **Function Prototype :**

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

**Parameter :** None

**Return:** TRUE if successful, False if failed.

**eg. :** RHIO\_Close();

## **RHIO\_SndCmd\_SetOnOff**

**Description :** Used to send the command to turn ON/OFF an output port.

### **Function Prototype :**

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

### **Parameter :**

int iOutputNum : Output port number (1~10)

EOnOffFlag eOnOff : Control flag (EOF\_ON, EOF\_OFF)

**Return :** TRUE if successful, False if failed.

### **Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_ONOFF)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30: Normal operation, 0x31 : Not the Run Mode, 0x32: Abnormal operation)

**eg. :** RHIO\_SndCmd\_SetOnOff(1, EOF\_ON);

#### **RHIO\_SndCmd\_GetOnOff**

**Description :** Used to send the command to get ON/OFF status of all the ports

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**eg. :** RHIO\_SndCmd\_GetOnOff();

#### **RHIO\_SndCmd\_SetSettingMod**

**Description :** Used to send the command to switch to the Set Mode.

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status Flag (ESF\_SET\_MODE)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30: Normal operation, 0x31 : Not the Run Mode, 0x32:Abnormal operation)

**eg :** RHIO\_SndCmd\_SetSettingMode();

#### **RHIO\_SndCmd\_SetRunMode**

**Description :** Used to send the command to switch to the Run Mode

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status Flag (ESF\_RUN\_MODE)

LPARAM : Control status Flag of the command (False if Time Out Error)

(0x30: Normal operation, 0x31 : Not the Run Mode, 0x32:Abnormal operation)

**eg. :** RHIO\_SndCmd\_SetRunMode();

## **RHIO\_SndCmd\_SetMACRO**

**Description :** Used to send the command to set/clear the MACRO of an input port

### **Function Prototype :**

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

### **Parameter :**

```
SRHIOSetting sSetData : Data to set
SRHIOSetting.sInput.eMacro : Flag to determine to set or clear
                             (ESOF_CLEAR, ESOF_SETTING)
SRHIOSetting.sInput.cMacro : MACRO string
int iPortNum : Input port number to set
```

**Return :** TRUE if successful, False if failed.

### **Parameter when reaction event is received :**

```
WPARAM : Command transfer status Flag (ESF_SET_MACRO)
LPARAM : Control status flag of the command (False if Time Out Error)
(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode, 0x32:Not set)
```

**eg. :**

#### **In case of Set command :**

```
SRHIOSetting sSetData;
sSetData.sInput.eMacro = ESOF_SETTING;
memset(sSetData.sInput.cMacro, '\\0', sizeof(sSetData.sInput.cMacro));
memcpy(sSetData.sInput.cMacro, "O2&O4|I1", 8);
RHIO_SndCmd_SetMACRO(sSetData, 1);
```

#### **In case of Set Clear command :**

```
SRHIOSetting sSetData;
sSetData.sInput.eMacro = ESOF_CLEAR;
RHIO_SndCmd_SetMACRO(sSetData, 1);
```



## RHIO\_SndCmd\_GetMACRO

**Description :** Used to send the command to check whether it is possible to get the MACRO of an input port

### Function Prototype :

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

### Parameter :

int iPortNum : Input port number to check

**Return :** TRUE if successful, False if failed.

### Parameter when reaction event is received :

WPARAM : Command transfer status Flag (ESF\_MON\_MACRO)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode, 0x32:Not set)

※ When reaction event is received, get data using RHIO\_GetSettingData function and check the MACRO on setting.

**eg. :** RHIO\_SndCmd\_GetMACRO(1);

## RHIO\_SndCmd\_SetDelayPulse

**Description :** Used to send the command to set the Delay/Pulse of an input port

### Function Prototype :

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

### Parameter :

SRHIOSetting sSetData : Data to set

SRHIOSetting.sInput.eDelayPulse : Flag to determine to set or clear  
(ESOF\_CLEAR, ESOF\_SETTING)

SRHIOSetting.sInput.cDelayPulse : Time string in 10 bytes

Byte Size	5(ON Time)	5(OFF Time)
Operation Set	"00000" ~ "50000"	"00000" ~ "50000"

int iPortNum : Input port number to set

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_SET\_DELAY\_PULSE)  
LPARAM : Control status flag of the command (False if Time Out Error)  
(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode, 0x32:Not set)

**eg. :**

**In case of Set command :**

```
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);
```

**In case of Set Clear command :**

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

**RHIO\_SndCmd\_GetDelayPulse**

**Description :** Used to send the command to check whether it is possible to get the Delay/Pulse of an input port

**Function Prototype :**

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

**Parameter :**

int iPortNum : Input port number to check

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_MON\_DELAY\_PULSE)  
LPARAM : Control status flag of the command (False if Time Out Error)  
(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode, 0x32:Not set)

※ When reaction event is received, get data using RHIO\_GetSettingData function and check the Delay/Pulse on setting.

**eg. :** RHIO\_SndCmd\_GetDelayPulse(1);

## RHIO\_SndCmd\_SetADC

**Description :** Used to send the command to set the Level of all the ADC ports from 1 to 4.

**Function Prototype :**

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

**Parameter :**

SRHIOSetting sSetData : Data to set

SRHIOSetting.sADC.sADCData[index].cADC : ADC Level string in 4 bytes  
between "0000" and "1023"

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_SET\_ADC)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

**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

**Description :** Used to send the command to check whether it is possible to get the Level of all the ADC ports.

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_MON\_ADC)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

※ When reaction event is received, get data using RHIO\_GetSettingData function and check the Level on setting.

**eg.** : RHIO\_SndCmd\_GetADC();

#### **RHIO\_SndCmd\_SetPortEnable**

**Description** : Used to send the command to enable/disable all the ports.

Function Prototype :

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

**Parameter** :

SRHIOSetting sSetData : Data to set  
SRHIOSetting.sADC.eModeADC[4] -> Value of the Level/Switch Mode  
 (EAM\_LEVEL, EAM\_SWITCH)  
SRHIOSetting.sADC.bReference -> Value of the ADC Reference  
 (0x30:Avcc, 0x31:Internal, 0x32:External)  
SRHIOSetting.sInput.isEnabled[12] -> Input Enable/Disable  
 (Enable:true, Disable:false)  
SRHIOSetting.sOutput.isEnabled[10] -> Output Enable/Disable  
 (Enable:true, Disable:false)

**Return** : TRUE if successful, False if failed.

**Parameter when reaction event is received** :

WPARAM : Command transfer status flag (ESF\_SET\_ENABLE)  
LPARAM : Control status flag of the command (False if Time Out Error)  
(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

**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**

**Description :** Used to send the command to check whether it is possible to get the Enable/Disable Port data of all the ports

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_MON\_ENABLE)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

※ When reaction event is received, get data using RHIO\_GetSettingData function and check the Enable/Disable Port on setting.

**eg. :** RHIO\_SndCmd\_GetPortEnable();

#### **RHIO\_SndCmd\_SetPwrStopEnable**

**Description :** Used to send the command to enable/disable the Power Failure Recovery of all the input ports

**Function Prototype :**

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

**Parameter :**

SRHIOSetting sSetData : Data to set

SRHIOSetting.sOutput.sEnablePowerStop[10] : Enable/Disable Power Failure Recovery of the input port (Enable:true, Disable:false)

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_SET\_PWR\_STOP)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

**eg. :**

```
SRHIOSetting sSetData;  
for(int index = 0; index < 10; index++)  
    sSetData.sOutput.sEnablePowerStop[index] = true;  
RHIO_SndCmd_SetPortEnable(sSetData);
```

#### **RHIO\_SndCmd\_GetPwrStopEnable**

**Description :** Used to send the command to check whether it is possible to get the Enable/Disable Power Failure Recovery of all the input ports

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_MON\_PWR\_STOP)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

※ When reaction event is received, get data using RHIO\_GetSettingData function and check the Enable/Disable Power Failure Recovery on setting.

**eg. :** RHIO\_SndCmd\_GetPortEnable();

#### **RHIO\_SndCmd\_SetFactoryReset**

**Description :** Used to send the command to request RHIO FactoryReset

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_SET\_FACTROT\_RESET)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation 0x39:Abnormal operation)

**eg. :** RHIO\_SndCmd\_SetFactoryReset();

#### **RHIO\_SndCmd\_SetSerial**

**Description :** Used to send command to set the RHIO serial number

**Function Prototype :**

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

**Parameter :**

SRHIOSetting sSetData : Data to set

CString strSerial : Serial number string limited to 12 bytes

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_SET\_SERIAL)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

**eg. :** RHIO\_SndCmd\_SetSerial("0000003");

#### **RHIO\_SndCmd\_GetSerial**

**Description :** Used to send the command to check whether it is possible to get the RHIO Serial number

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_MON\_SERIAL)

LPARAM : Control status flag of the command (False if Time Out Error)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

※ When reaction event is received, get data using RHIO\_GetSettingData function.

**eg. :** RHIO\_SndCmd\_GetSerial();

#### **RHIO\_SndCmd\_GetFirmware**

**Description :** Used to send the command to check whether it is possible to get the RHIO Firmware Version

**Function Prototype :**

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

**Parameter :** None

**Return :** TRUE if successful, False if failed.

**Parameter when reaction event is received :**

WPARAM : Command transfer status flag (ESF\_MON\_FIRMWARE)

LPARAM : Control status flag (False if Time Out)

(0x30:Normal operation, 0x39:Abnormal operation, 0x31:Not the Set Mode)

※ When reaction event is received, get data using RHIO\_GetSettingData function.

**eg. :** RHIO\_SndCmd\_GetFirmware();

#### **RHIO\_GetSettingData**

**Description :** Used to send the command to get the setting data corresponding to the command transfer status flag

**Function Prototype :**

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

**Parameter :**

SRHIOSetting &sSetData : Data to get

**Return :** TRUE if successful, False if failed.

**eg. :**

```
SRHIOSetting sSetData;
RHIO_GetSettingData(sSetData); // Setting data stored to sSetData
```

#### **RHIO\_GetOnOffData**

**Description :** Used to send the command to get the ON/OFF status of all the ports

**Function Prototype :**

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

**Parameter :**

SOnOffStatusData &sOnOffData : Data to get

**Return :** TRUE if successful, False if failed.

**eg. :**

```
SOnOffStatusData sOnOffData;
RHIO_GetOnOffData(sOnOffData);
//ON/OFF status stored to sOnOffData
```



## **Appendix E. Warranty**

### **E.1 GENERAL WARRANTY POLICY**

Sena Technologies, Inc. (hereinafter referred to as SENA) warrants that the Product shall conform to and perform in accordance with published technical specifications and the accompanying written materials, and shall be free of defects in materials and workmanship, for the period of time herein indicated, such warranty period commencing upon receipt of the Product.

This warranty is limited to the repair and/or replacement, at SENA's discretion, of defective or non-conforming Product, and SENA shall not be responsible for the failure of the Product to perform specified functions, or any other non-conformance caused by or attributable to: (a) any misapplication or misuse of the Product; (b) failure of Customer to adhere to any of SENA's specifications or instructions; (c) neglect of, abuse of, or accident to, the Product; or (d) any associated or complementary equipment or software not furnished by SENA.

Limited warranty service may be obtained by delivering the Product to SENA or to the international distributor it was purchased through and providing proof of purchase or receipt date. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to SENA, and to use the original shipping container or equivalent.

### **E.2 LIMITATION OF LIABILITY**

EXCEPT AS EXPRESSLY PROVIDED HEREIN, SENA MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, WITH RESPECT TO ANY EQUIPMENT, PARTS OR SERVICES PROVIDED PURSUANT TO THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER SENA NOR ITS DEALER SHALL BE LIABLE FOR ANY OTHER DAMAGES, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER IN AN ACTION IN CONTRACT OR TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY), SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS RESULTING FROM, OR ARISING OUT OF, OR IN CONNECTION WITH THE USE OF FURNISHING OF EQUIPMENT, PARTS OR SERVICES HEREUNDER OR THE PERFORMANCE, USE OR INABILITY TO USE THE SAME, EVEN IF SENA OR ITS DEALER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL SENA OR ITS DEALERS TOTAL LIABILITY EXCEED

THE PRICE PAID FOR THE PRODUCT.

### **E.3 HARDWARE PRODUCT WARRANTY DETAILS**

**WARRANTY PERIOD:** SENA warranties embedded hardware Product for a period of one (1) year, and external hardware Product for a period of three (3) or five (5) years according to the Product type.

**WARRANTY PROCEDURE:** Upon return of the hardware Product SENA will, at its option, repair or replace Product at no additional charge, freight prepaid, except as set forth below. Repair parts and replacement Product will be furnished on an exchange basis and will be either reconditioned or new. All replaced Product and parts become the property of SENA. If SENA determines that the Product is not under warranty, it will, at the Customers option, repair the Product using current SENA standard rates for parts and labor, and return the Product at no charge in or out of warranty.

**WARRANTY EXCLUSIONS:** Damages caused by

- Accidents, falls, objects striking the SENA product,
- Operating the Product in environments that exceed SENA's temperature and humidity specifications,
- Power fluctuations, high voltage discharges,
- Improper grounding, incorrect cabling,
- Misuse, negligence by the customer or any other third party,
- Failure to install or operate the product (s) in accordance to their SENA User Manual,
- Failure caused by improper or inadequate maintenance by the customer or any other third party,
- Floods, lightning, earthquakes,
- Water spills,
- Replacement of parts due to normal wear and tear,
- Hardware has been altered in any way,
- Product that has been exposed to repair attempts by a third party without SENA's written consent,
- Hardware hosting modified SENA Software, or non-SENA Software, unless modifications have been approved by SENA.
- Battery component capacity degradation due to usage, aging, and with some chemistry, lack of maintenance.

## **E.4 SOFTWARE PRODUCT WARRANTY DETAILS**

WARRANTY PERIOD: SENA warrants software Product for a period of one (1) year.

WARRANTY COVERAGE: SENA warranty will be limited to providing a software bug fix or a software patch, at a reasonable time after the user notifies SENA of software non-conformance.

## **E.5 THIRD-PARTY SOFTWARE PRODUCT WARRANTY DETAILS**

The warranty policy of the third-party software is conformed to the policy of the corresponding vendor.