



***INDUSTRIAL COMPUTER SOURCE***<sup>®</sup>

# **Model RDIO24 Product Manual**

**MANUAL NUMBER : 00650-124-2B**



***INDUSTRIAL COMPUTER SOURCE***<sup>®</sup>



<http://www.indcompsrc.com>

6260 SEQUENCE DRIVE, SAN DIEGO, CA 92121-4371 (619) 677-0877 (FAX) 619-677-0895

INDUSTRIAL COMPUTER SOURCE EUROPE TEL 01.69.18.74.40 FAX 01.64.46.40.42 • INDUSTRIAL COMPUTER SOURCE (UK) LTD TEL 01243-523500 FAX 01243-532949



## FOREWARD

This product manual provides information to install, operate and or program the referenced product(s) manufactured or distributed by Industrial Computer Source. The following pages contain information regarding the warranty and repair policies.

Technical assistance is available at: **1-800-480-0044**.

**Manual Errors, Omissions and Bugs:** A "Bug Sheet" is included as the last page of this manual. Please use the "Bug Sheet" if you experience any problems with the manual that requires correction.

The information in this document is provided for *reference* only. Industrial Computer Source does not assume any liability arising from the application or use of the information or products described herein. This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Industrial Computer Source, nor the rights of others.

Copyright © 1995 by Industrial Computer Source, a California Corporation, 6260 Sequence Drive, San Diego, CA 92121-4371. Industrial Computer Source is a Registered Trademark of Industrial Computer Source. All trademarks and registered trademarks are the property of their respective owners. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording , or otherwise, without the prior written permission of the publisher.

## Guarantee

A thirty day money-back guarantee is provided on all **standard** products sold. **Special order products** are covered by our Limited Warranty, *however they may not be returned for refund or credit. EPROMs, RAM, Flash EPROMs or other forms of solid electronic media are not returnable for credit - but for replacement only. Extended Warranty available. Consult factory.*

## Refunds

In order to receive refund on a product purchase price, the product must not have been damaged by the customer or by the common carrier chosen by the customer to return the goods, and the product must be returned complete (meaning all manuals, software, cables, etc.) within 30 days of receipt and in as-new and resalable condition. The **Return Procedure** must be followed to assure prompt refund.

## Restocking Charges

Product returned *after 30 days, and before 90 days*, of the purchase will be subject to a **minimum 20%** restocking charge and any charges for damaged or missing parts.

Products not returned within 90 days of purchase, or products which are not in as-new and resalable condition, are not eligible for credit return and will be returned to the customer.

## Limited Warranty

One-year limited warranty on all products sold with the exception of the "Performance Series" I/O products, which are warranted to the original purchaser for as long as they own the product, subject to all other conditions below, including those regarding neglect, misuse and acts of God. Within one year of purchase, Industrial Computer Source will repair or replace, at our option, any defective product. At any time after one year, we will repair or replace, at our option, any defective "Performance Series" I/O product sold. This does not include products damaged in shipment, or damaged through customer neglect or misuse. Industrial Computer Source will service the warranty for all standard catalog products for the first year from the date of shipment. After the first year, for products not manufactured by Industrial Computer Source, the remainder of the manufacturer's warranty, if any, will be serviced by the manufacturer directly.

The **Return Procedure** must be followed to assure repair or replacement. Industrial Computer Source will normally return your replacement or repaired item via UPS Blue. *Overnight delivery or delivery via other carriers is available at additional charge.*

The limited warranty is void if the product has been subjected to alteration, neglect, misuse, or abuse; if any repairs have been attempted by anyone other than Industrial Computer Source or its authorized agent; or if the failure is caused by accident, acts of God, or other causes beyond the control of Industrial Computer Source or the manufacturer. Neglect, misuse, and abuse shall include any installation, operation, or maintenance of the product other than in accordance with the owners' manual.

No agent, dealer, distributor, service company, or other party is authorized to change, modify, or extend the terms of this Limited Warranty in any manner whatsoever. Industrial Computer Source reserves the right to make changes or improvements in any product without incurring any obligation to similarly alter products previously purchased.



**Shipments not in compliance with this Guarantee and Limited Warranty Return Policy will not be accepted by Industrial Computer Source.**

## Return Procedure

For any Limited Warranty or Guarantee return, please contact Industrial Computer Source's Customer Service at **1-800-480-0044** and obtain a Return Material Authorization (RMA) Number. All product(s) returned to Industrial Computer Source for service or credit **must** be accompanied by a Return Material Authorization (RMA) Number. Freight on all returned items **must** be prepaid by the customer who is responsible for any loss or damage caused by common carrier in transit. Returns for Warranty **must** include a Failure Report for each unit, by serial number(s), as well as a copy of the original invoice showing date of purchase.

To reduce risk of damage, returns of product must be in an Industrial Computer Source shipping container. If the original container has been lost or damaged, new shipping containers may be obtained from Industrial Computer Source Customer Service at a nominal cost.

## Limitation of Liability

In no event shall Industrial Computer Source be liable for any defect in hardware or software or loss or inadequacy of data of any kind, or for any direct, indirect, incidental, or consequential damages in connection with or arising out of the performance or use of any product furnished hereunder. Industrial Computer Source liability shall in no event exceed the purchase price of the product purchased hereunder. The foregoing limitation of liability shall be equally applicable to any service provided by Industrial Computer Source or its authorized agent.

Some *Sales Items* and *Customized Systems* are **not** subject to the guarantee and limited warranty. However, in these instances any deviations will be disclosed prior to sales and noted in the original invoice. ***Industrial Computer Source reserves the right to refuse returns or credits on software or special order items.***

## Advisories

Three types of advisories are used throughout the manual to stress important points or warn of potential hazards to the user or the system. They are the Note, the Caution, and the Warning. Following is an example of each type of advisory:

**Note:** The note is used to present special instruction, or to provide extra information which may help to simplify the use of the product.



### CAUTION!



A Caution is used to alert you to a situation which if ignored may cause injury or damage equipment.

---



### WARNING!



A Warning is used to alert you of a situation which if ignored will cause serious injury.

---

Cautions and Warnings are accented with triangular symbols. The exclamation symbol is used in all cautions and warnings to help alert you to the important instructions. The lightning flash symbol is used on the left hand side of a caution or a warning if the advisory relates to the presence of voltage which may be of sufficient magnitude to cause electrical shock.

Use caution when servicing any electrical component. We have tried to identify the areas which may pose a Caution or Warning condition in this manual; however, Industrial Computer Source does not claim to have covered all situations which might require the use of a Caution or Warning.

You must refer to the documentation for any component you install into a computer system to insure proper precautions and procedures are followed.

# Table of Contents

<b>FOREWARD</b> .....	<b>iii</b>
<b>Guarantee</b> .....	<b>iv</b>
<b>Limited Warranty</b> .....	<b>iv</b>
<b>Return Procedure</b> .....	<b>v</b>
<b>Limitation of Liability</b> .....	<b>v</b>
<b>Advisories</b> .....	<b>vi</b>
<b>Chapter 1: Installation</b> .....	<b>1</b>
Backing up the Disk .....	1
Hard Disk Installation .....	1
Installing the Pod .....	2
Digital Input/Output Pin Connections .....	4
Screw Terminal Connections .....	4
50-Pin Connector .....	5
37 Pin Connector J1 (Optional) .....	6
<b>Chapter 2: Functional Description</b> .....	<b>7</b>
Features .....	7
Description .....	7
How to remain CE Compliant .....	9
<b>Chapter 3: Software</b> .....	<b>11</b>
General .....	11
Command Structure .....	11
Addressed Mode .....	11
Non-Addressed Mode .....	11
Command List .....	12
Command Functions .....	13
Set Time Base .....	13
Configure Bits as Input or Output .....	14
Read Digital Inputs .....	14
Write Digital Outputs .....	15
Generate a Square Wave Output .....	17
Read Change-of-State .....	17
Enable Change-of-State Detection .....	17
Select Which Edge Will Increment Counters .....	18
Read Digital Input Counter & Read Time Left on Pulse or Free Run Output .....	19
Reset Counter & Turn Off Pulse or Free-Run Output .....	20
Read Firmware Revision Number .....	20
Resend Last Response .....	20
Hello Message .....	21
Set a New Baud Rate (When shipped the baud rate is set at 9600.) .....	21
Program the Pod Address .....	22

Read and Store Digital Input Data .....	23
Re-send Data .....	23
Enter a New Program .....	24
Error Codes: .....	24
<b>Chapter 4: Specifications .....</b>	<b>25</b>
Serial Communications Interface .....	25
Digital Inputs .....	25
Digital Outputs .....	25
Environmental .....	26
Power Required .....	26
<b>Appendix A: Application Considerations .....</b>	<b>27</b>
Introduction .....	27
Balanced Differential Signals .....	27
RS-485 Data Transmission .....	28

## **CE Declaration of Conformity**

## **List of Figures**

Figure 1-1: Option Selection Map .....	3
Figure 2-1: Block Diagram .....	8
Figure A-1: Typical RS-485 Two-Wire Multidrop Network .....	29

**Current Revision 2B**

**September 1997**



# Chapter 1: Installation

## Backing up the Disk

---

The software provided with the RDIO24 is in DOS on a 3½" diskette. As with any software package, you should make backup copies for everyday use and place your original master diskette in a safe location.

The easiest way to make a backup copy is to use the DOS DISKCOPY utility.

In a single-drive system the command is:

```
DISKCOPY A: A:
```

In a two-disk system the command is:

```
DISKCOPY A: B:
```

This will copy the contents of the master disk in drive A to the backup disk in drive B.

## Hard Disk Installation

---

The files contained on the master diskette may also be copied onto your hard disk. To do this perform the following:

1. Place the master diskette into the floppy drive.
2. Change the active drive to the drive that has the master diskette installed.  
For example, if the diskette is in the A drive, type A:.
3. Type `INSTALL` and follow the screen prompts.

Files contained on the disk are as follows:

**SETUP PROGRAM:** A menu-driven program designed to assist you to configure your RDIO24. The program instructs you on use of the serial opto-isolation jumpers and allows you to specify the Pod Address and the serial baud rate. This program can also auto-detect the address and/or baud rate of an attached card in the host computer. The Setup program should be run with only a single pod attached to the RS-485 port in your computer. You will need to know the setup information (base address, IRQ) of the port so that the Setup program can access it. The Setup program will create a batch file in its directory, called `RISC.BAT`, that can be used to run `RISCTerm`.

**RISCTerm:** This program provides a simple, non-automated interface to RS-485 devices and can be used to configure or test RDIO24 setup and operations. Help on using `RISCTerm` is provided in-program and is accessed by pressing "F1".

- DRIVERS:** COMMDRV.PAS, COMMDRV.H, and COMMDRV.C are communication short cuts or drivers provided in C and Pascal source code. Three functions are provided: “initComCard”, “writePod”, and “readPod”. The functions are polling based, no IRQ necessary, and deal with strings rather than character-by-character. Consult the source code files for more information.
- SAMPLE:** A sample program is provided that demonstrates reading digital input bits, writing digital output bits, configuring bits as input or output, and polling the pod for general information. The sample also demonstrates use of the driver .
- PROGRAM.EXE:** Depending on the exact configuration options requested when the unit was purchased, you may or may not have this file. This program is used to upgrade the pod. More information is provided with the program.

## Installing the Pod

---

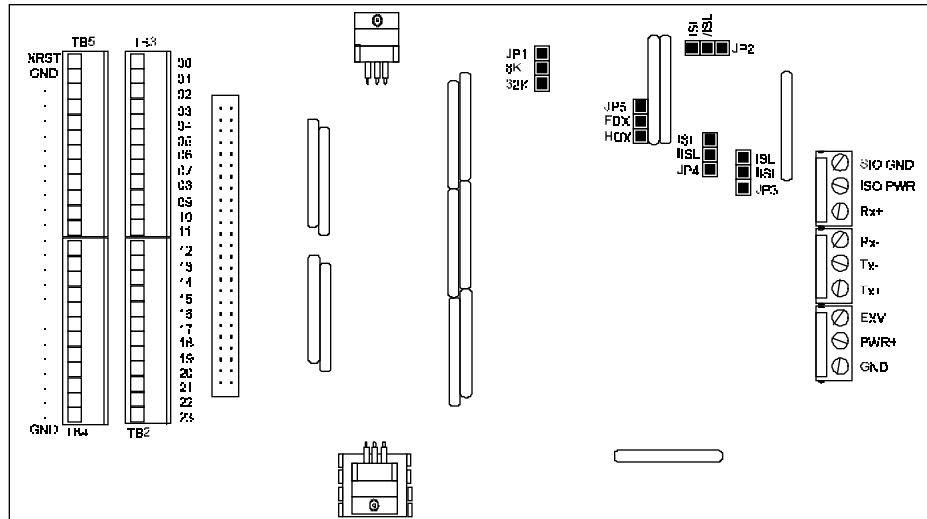
The RDIO24 enclosure is an easily-mounted steel enclosure. Outside dimensions of the enclosure are: 8.5" long by 5.25" wide by 2" high. The cover is secured to the body by four Phillips-head screws. Four threaded bosses inside the enclosure provide for mounting the printed circuit card assemblies. The enclosure has a notch at one end for external power wiring and a deeper notch at the other end for input and output wiring.

Electrical power RS-485 connections to the pod are made at Terminal Board TB1. Terminal markings and their functions are as follows:

- ISO GND and ISO.PWR:** This is the power connection from the computer’s +12VDC supply via telephone cable on the RS-485 network or from a central power supply. This powers the isolator section and is independent of “local power”. The voltage level can be from 7.5 VDC to 35 VDC. (An on-board voltage regulator regulates the power to +5 VDC.) RDIO24 will require only about 7 mA of current and, thus, any voltage drop of computer power will be inconsequential. If separate power is not available, then, with some loss of isolation, these terminals can be jumpered to the “local power” terminals.
- RX<sup>+</sup> and RX<sup>-</sup>:** These are the differential “receive” inputs for four-wire RS-485 communications. If two-wire connection is to be used, install jumpers from RX<sup>+</sup> to TX<sup>+</sup> and from RX<sup>-</sup> to TX<sup>-</sup>.
- TX<sup>+</sup> and TX<sup>-</sup>:** These are the differential “transmit” outputs for four-wire RS-485 communications. If two-wire connection is to be used, install jumpers as described above.
- EXV:** This terminal is for the “application power” or the voltage level to which digital inputs and outputs are connected through the loads or external signal sources. Open-collector Darlington amplifiers are used at the outputs. Inductive suppression diodes are included and the application power level can be as high as 50 VDC.

**PWR + and GND:**

These terminals can be used to apply power to the pod from a local power supply. If a local power supply is used, the voltage can be anywhere in the range from 7.5 VDC to 16 VDC. Higher local power can be used, 24 VDC for example, if an external zener diode is used to reduce the voltage applied to the RDIO24. ( See the Specification section of this manual to determine the zener diode power rating required.)



**Figure 1-1: Option Selection Map**

There are five jumper locations on the card. Locations are as shown on the **Option Selection Map** and their functions are as follows:

- JP1:** Normally this jumper will be in the “8K” position and need not concern you. If the RAM and EEPROM memory in your pod has been expanded, then this jumper should be in the “32K” position.
- JP2, JP3, and JP4:** Normally these jumpers should be in the “ISL” position. If you wish to bypass the opto-isolators, then you can move these jumpers to the “/ISL” position.
- JP5:** Place this jumper according to the mode in which you will operate the pod. Place this jumper in the “FDX” position if you will use four-wire communication and in the “HDX” position if you are going to use two-wire communication.

## Digital Input/Output Pin Connections

---

Digital input and output connections can be made in any of three ways: (a) at Euro style screw terminals TB2, TB3, TB4, and TB5; or (b) at 50-pin, keyed, male, insulation-displacement type, ribbon cable connector J2; or (c) at 37-pin, male, sub-D connector J1.

### Screw Terminal Connections

Digital I/O connection points at the screw terminal connectors are as marked on the printed circuit board at terminals TB2 and TB3. Bit locations are numbered 00 through 23 (decimal). Note also that a terminal labeled XRST is provided at TB5 for Reset signals to the microprocessor. The other terminals on TB5 and TB4 provide ground returns for the I/O signals.

## 50-Pin Connector

Connector pin assignments for the 50-pin connector are listed below. Note that alternate wires in the ribbon cable will be connected to ground. This provides means to isolate the digital signals from each other in the cabling.

Pin	Signal	Pin	Signal
1	I/O23	26	Gnd
2	Gnd	27	I/O10
3	I/O22	28	Gnd
4	Gnd	29	I/O09
5	I/O21	30	Gnd
6	Gnd	31	I/O08
7	I/O20	32	Gnd
8	Gnd	33	I/O07
9	I/O19	34	Gnd
10	Gnd	35	I/O06
11	I/O18	36	Gnd
12	Gnd	37	I/O05
13	I/O17	38	Gnd
14	Gnd	39	I/O04
15	I/O16	40	Gnd
16	Gnd	41	I/O03
17	I/O15	42	Gnd
18	Gnd	43	I/O02
19	I/O14	44	Gnd
20	Gnd	45	I/O01
21	I/O13	46	Gnd
22	Gnd	47	I/O00
23	I/O12	48	Gnd
24	Gnd	49	Ext Reset
25	I/O11	50	Gnd

**37 Pin Connector J1 (Optional)**

Connector pin assignments for the 37-pin connector are listed below. Note that, since this connector would normally not mate with a ribbon cable, alternate-ground pins are not provided (there would not be enough pins anyway). Also, note that there is provision for external Reset, for a Timer signal, and for Interrupts 0 and 1.

Pin	Signal	Pin	Signal
1	I/O 00	19	I/O 18
2	I/O 01	20	I/O 19
3	I/O 02	21	I/O 20
4	I/O 03	22	I/O 21
5	I/O 04	23	I/O 22
6	I/O 05	24	I/O 23
7	I/O 06	25	Gnd
8	I/O 07	26	Resv'd (/Int 1)
9	I/O 08	27	Resv'd (TMR 0)
10	I/O 09	28	Resv'd (Int 0)
11	I/O 10	29	Local Pwr Gnd
12	I/O 11	30	Local Pwr
13	I/O 12	31	R(+)
14	I/O 13	32	R(-)
15	I/O 14	33	Isolator Pwr
16	I/O 15	34	Isolator Pwr Gnd
17	I/O 16	35	Apl'n Pwr
18	I/O 17	36	Apl'n Pwr Gnd
		37	Reset

# Chapter 2: Functional Description

## Features

- Opto-Isolated RS-485 Serial Interface to Host Computer
- 24-Bit Digital I/O Programmable Bit-by-Bit, in 8-Bit Bytes, or in 24-Bit Words
- Digital Input and Output Voltages up to 50VDC
- Open Collector Digital Outputs for Loads up to 350mA
- Protective Steel Enclosure
- Type 8031 Microcontroller with 8K x 8 RAM and 8K x 8 EEPROM (32K x 8 optional)
- All Programming in Software, No Switches to Set. Jumper to Set 2-Wire or 4-Wire modes, and Jumpers (3) to By-Pass Opto-Isolators if Desired.
- 16-Bit Digital-Input Software Counters
- Change of State Flag Readable via the Serial Port
- Digital Outputs May Be Either Level or Pulse.

## Description

---

RDIO24 is an intelligent, 24-bit parallel, digital-I/O-to-computer interface unit. Communication with the host computer is via EIA RS-485, full- or half-duplex, serial communications protocol. ASCII-based command/response protocol permits communication with virtually any computer system. As many as 32 pods (or other RS-485 devices) may be connected on a single two- or four-wire multidrop RS-485 network. Each unit has a unique address. Communication uses a master/slave protocol wherein the pod talks only if questioned by the computer.

A type 8031 microcontroller (with 8x8K RAM, 8x8K non-volatile EEPROM, and a watchdog timer circuit) gives RDIO24 the capability and versatility expected from a modern distributed control system. To accommodate special programs, the RAM and EEPROM can each be expanded to 8x32K. The unit contains CMOS low-power circuitry, an optically-isolated receiver/transmitter, and power conditioners for local and external isolated power. It can operate at baud rates up to 28.8 Kbaud and distances up to 4000 feet with low-attenuation twisted-pair cabling.

All programming of RDIO24 is in ASCII-based software. ASCII-based programming permits you to write applications in any high-level language that supports ASCII string functions and you can use ACCES "R" series pods with virtually any computer.

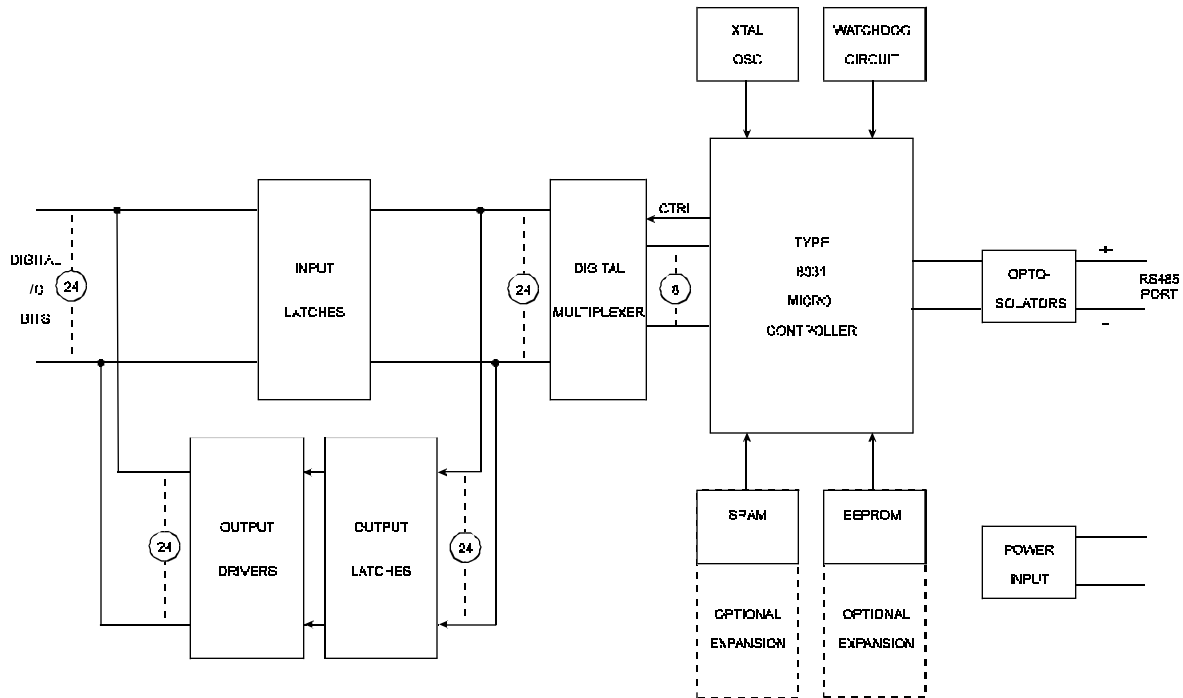
The module, or pod, address is programmable from 00 to FF hex and whatever address is assigned is stored in EEPROM and used as the default address at the next Power-ON. Similarly, the baud rate is programmable for 1200, 2400, 4800, 9600, 14400, 19200, and 28800. The address is stored in EEPROM and used as default at the next Power-ON.

The time base, used in all time-relevant operations is also software selectable between:

- a. Digital input sample rate from 14 Hz to 1 KHz
- b. Digital output square wave pulse from 7 Hz to 500 Hz

Digital inputs of up to 50V may be read individually, or in 8-bit bytes, or in 24-bit word groups. There are also digital software counters on each input. Selectable edges can be counted for up to 65,535 transitions. These counters support Read and Reset commands. Moreover, change-of-state flags can be set on any enabled input bits and can be read via the serial port. This is particularly useful in applications where it's necessary to detect contact closures or openings. This change-of-state detection capability is enabled on a bit-by-bit basis for all bits programmed for input.

Digital outputs may be programmed individually, or in 8-bit bytes, or in 24-bit words. These outputs may be latched, pulsed, or set to free-run for a prescribed period of time. The digital output drivers are open collector circuits that have 350 mA drive capability at a logic "low". The unit can comply with up to 50 VDC (voltage supplied by you) or, if no external voltage is supplied, outputs will be pulled up by a 10 K $\Omega$  resistor to 5 VDC.



**Figure 2-1: Block Diagram**

The built-in watchdog timer resets the pod if, for some unexpected reason, the microcontroller "hangs up". Data collected by the pod can be stored in local RAM and accessed later through the computer's serial port. This facilitates a stand-alone pod mode of operation.



## How to remain CE Compliant

---

This device complies with CE Directives 72/23/EEC and EMC 89/336/EEC. CE compliance is based on the interaction of all the components of a system. Any modifications made to the equipment may affect the CE compliance and must be approved in writing by Industrial Computer Source. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to claim CE compliance.

The Model RDIO24 is designed to be CE Compliant when used in an CE compliant chassis. Maintaining CE Compliance also requires proper cabling and termination techniques. The user is advised to follow proper cabling techniques from sensor to interface to ensure a complete CE Compliant system. Industrial Computer Source does not offer engineering services for designing cabling or termination systems. Although Industrial Computer Source offers accessory cables and termination panels, it is the user's responsibility to ensure they are installed with proper shielding to maintain CE Compliance.

This page intentionally left blank

# Chapter 3: Software

## General

---

You received ASCII-based software on 3½” diskette for use with RDIO24. ASCII programming permits you to write applications in any high level language that supports ASCII string functions.

The communication protocol has two forms: addressed and non-addressed. Non-addressed protocol can be used when only one RDIO24 is in use. If more than one RDIO24 is in use, you must use addressed protocol. The difference is that an address command is sent to enable the specific pod. The address command is only sent once during communication between the pod and the host computer. It enables communication with that specific pod and disables all other devices on the network.

### Command Structure

All communication must be 7 data bits, even parity, 1 stop bit. All numbers sent to or received from the pod are in hexadecimal form. The factory default baud rate is 9600 Baud. The pod is considered to be in addressed mode any time its pod address is not 00. The factory default pod address is 00 (non-addressed mode).

### Addressed Mode

The address select command must be issued before any other command to the addressed pod. The address command is as follows:

“!xx[CR]” where xx is the pod address from 01 to FF hex, and [CR] is Carriage Return, ASCII character 13.

The pod responds with either “xxN[CR]” or “xxY[CR]” if an input change of state has occurred on enabled bits since the last “Y” or address command. Once the address select command has been issued, all further commands (other than a new address select) will be executed by the selected pod. *The addressed mode is required when using more than one pod.*

### Non-Addressed Mode

When there’s only one pod connected, no address select command is needed. You can merely issue commands listed in the following table. Terminology used is as follows:

- a. The single lower case letter ‘x’ designates any valid hex digit (0-F).
- b. The single lower case letter ‘b’ designates either a ‘1’ or ‘0’.
- c. The symbol ‘±’ designates either a ‘+’ or a ‘-’.
- d. All commands are terminated with CR, the ASCII character #13.
- e. Wherever xx is used to designate a bit number, only 00-17 hex are valid.
- f. Commands are not case-sensitive; i.e., upper- or lower case letters may be used..
- g. The symbol ‘\*’ means zero or more valid characters (total msg length<255 decimal)

## Command List

Sxxxx	Set a new timebase. 039A<xxxx<=FFFF	CR
SCxxxx	Set a new timebase, and reset all free-run and pulse DOs	CR
MLxx	Configure bits 00-07 as input/output. 0=in, 1=out	CR
MMxx	Configure bits 08-0F as input/output. 0=in, 1=out	CR
MHxx	Configure bits 10-17 as input/output. 0=in, 1=out	CR
I	Read all 24 digital bits.	xxxxxCR
Ixx	Read a single digital bit. (00<=xx<=17)	bCR
IL	Read digital bits 00-07	xxCR
IM	Read digital bits 08-0F	xxCR
IH	Read digital bits 10-17	xxCR
Oxxxxxx	Output all 24 digital bits.	CR
Ox±	Output either high or low on bit x	CR
Oxx±	Output either high or low on bit xx	CR
Ox±xx	Output either high or low on bit x for time xx	CR
Oxx±xx	Output either high or low on bit xx for time xx	CR
OLxx	Output xx on bits 00-07	CR
OMxx	Output xx on bits 08-0F	CR
OHxx	Output xx on bits 10-17	CR
Oxx±xx	Output either high or low on bit xx for time xx	CR
Fxx,xx	Set Digital Output xx to free run with period xx	CR
Y	Read digital input Change Of State bit and clear bit	Y or N
TLxx	Set bit 00-07 Mask for COST bit flag, 1=change will set COST	CR
TMxx	Set bit 08-0F Mask for COST bit flag	CR
THxx	Set bit 10-17 Mask for COST bit flag	CR
Dx±	Set digital input active state high or low on bit x	CR
Dxx±	Set digital input active state high or low on bit xx	CR
Cxx	Read digital input xx counter (counts each active pulse) or Read pulse/free-run output xx counter and reload value	xxxxCR xxxxCR
Rxx	Reset digital input counter xx to 0000	CR
Rall	Reset all digital input counters to 0000	CR
V	Read the Firmware version number	x.xxCR
N	Resend last response	varies
H*	Greeting message: copyright, firmware version number	varies
BAUD=xxx	Set new baud rate. Each x is code number for new baud	
POD=xx	Set pod address to xx	=:Baud:0x varies
PROGRAM=	Begin process of uploading custom program to pod	special
D	Download historical storage of digital input data again	varies
FASTDATAL	Acquire bits 0-7 as fast as possible, then display	varies
FASTDATAM	Acquire bits 8-F as fast as possible, then display	varies
FASTDATAH	Acquire bits 10-17 as fast as possible, then display	varies
!xx	Address Select Command, Selects the POD at address 'xx' for communication.	CR

## Command Functions

---

The following paragraphs give details of the command functions, describe what the commands cause, and give examples. Please note that all commands have an acknowledgement response. You must wait for a response from a command before sending another command.

### Set Time Base

Sxxxx Set Time Base  
SCxxxx Set Time Base with Output Synchronization

This function sets the pod-global timebase which is used in all time-sensitive operations. Valid values range from 039A to FFFF. Any invalid value will result in the pod-default timebase of 2400 (10ms/100Hz).

039A corresponds to 1KHz, 2400 is 100Hz, and the longest timebase of FFFF corresponds to 14Hz. (11,059,200 Hz / 12 / timebase = Hz rate of time base)

The SCxxxx variant sets the timebase as normal, then causes all free-run and pulse outputs to change on the next timebase tick.

Examples:

Program the RDIO24 to a 1msec timebase.

SEND: S039A  
RECEIVE: [CR]

Program the RDIO24 to a 50msec timebase, and synchronize outputs.

SEND: SC4800  
RECEIVE: [CR]

**Note:** The timebase configured is stored in EEPROM on the pod, and will be used as the default (power-on) timebase. The factory default timebase (100Hz) can be restored by sending "S0000" to the pod.

**Application Note:** If it is necessary to synchronize free-running outputs, the SCxxxx command will assist in this. First, configure all desired free-run bits as outputs. Then initialize each bit as free-run using the fxx,xx command. Now, by issuing an SCxxxx command, synchronize all the free-run outputs to toggle on the next time-base tick. Now, all free-run outputs with the same time-duration will toggle on the same tick. (20usec skew max. per byte).

## Configure Bits as Input or Output

MLxx	Configure bits 0-7 as input/output
MMxx	Configure bits 8-F as input/output
MHxx	Configure bits 10-17 as input/output

These functions program, on a bit-by-bit basis, the digital bits as input or output, in groups of eight. The MLxx command control input/output on bits 0-7, MMxx control bits 8-F, and MHxx controls bits 10-17. A “zero” in any bit position of the xx control byte designates the corresponding bit as an input. Conversely, a “one” designates a bit to be configured as an output. (Note: Any bit configured as an output can still be read as an input if the current value output is a “one”.

Examples:

Program even bits as outputs, and odd bits as inputs.

```

SEND:      MLAA
RECEIVE:   [CR]
SEND:      MMAA
RECEIVE:   [CR]
SEND:      MHAA
RECEIVE:   [CR]

```

Program bits 0-7 as input, and bits 8-F as output.

```

SEND:      ML00
RECEIVE:   [CR]
SEND:      MMFF
RECEIVE:   [CR]

```

## Read Digital Inputs

I	Read 24 bits
Ixx	Read bit number xx
IL	Read bits 00-07
IM	Read bits 08-0F
IH	Read bits 10-17

These commands read the digital input bits from the pod. All byte or word wide responses are sent most-significant nibble first.

Examples:

Read ALL 24 bits.

```
SEND:      I
RECEIVE:   FFFFFFF[CR]
```

Read only bit 17 (23 decimal, the highest bit on the card).

```
SEND:      I17
RECEIVE:   1[CR]
```

Read only bit 2.

```
SEND:      I02
RECEIVE:   1[CR]
```

Read bits 8-F.

```
SEND:      IM
RECEIVE:   FF[CR]
```

## Write Digital Outputs

Oxxxxxx	Write to all 24 digital output bits
Ox±	Set bit x hi or low
Oxx±	Set bit xx hi or low
Ox±xx	Pulse bit x hi or low for time xx
Oxx±xx	Pulse bit xx hi or low for time xx
OLxx	Write to bits 0-7
OMxx	Write to bits 8-F
OHxx	Write to bits 10-17
bx±xx	Identical to Ox±xx
bxx±xx	Identical to Oxx±xx

These commands write outputs to digital bits. Any attempt to write to a bit configured as input will fail. Writing to a byte or word wherein some bits are input and some are output will cause the output latches to change to the new value, but the bits which are inputs will not output the value until/unless they are placed in output mode.

Single bit commands will return an error (4) if an attempt is made to write to a bit configured as an input.

Writing a “one” to a port asserts the pull-down. Writing a zero de-asserts the pull-down. Therefore, if the factory installed +5V pull-up is installed, writing a one will cause zero volts to be at the connector, and writing a zero will cause +5 volts to be asserted. If the factory installed pull-up has been removed, the user supplied pull-up will be asserted.

Pulsing a bit high or low uses the currently configured timebase. The bit will be set to 5V (-) or 0V (+) for a period equal to xx time-ticks, where one time-tick occurs every timebase. So, if the default timebase of 2400 (3916 dec, or 100Hz) is currently selected, and 32 (hex) was specified for a pulse duration, the bit will pulse high or low for 50 counts, or one half second. (50x10ms=500ms=half-a-second)

Examples:

Write a one to bit 13 (set output to zero volts, assert the pull-down).

```
SEND:      O13+
RECEIVE:   [CR]
```

Write a zero to bit 2 (set output to +5V or user pull-up).

```
SEND:      O2-
           or
SEND:      O02-
RECEIVE:   [CR]
```

Write zeros to bits 0-7.

```
SEND:      OL00
RECEIVE:   [CR]
```

Write zeros to every odd bit.

```
SEND:      OAAAAAA
RECEIVE:   [CR]
```

Cause bit 7 to pulse to 0 Volts for 20ms (assuming S039A was issued).

```
SEND:      O7+14
RECEIVE:   [CR]
```

Write zeros to bits 0-9, 13-17, and ones on all other bits.

```
SEND:      O07FC00
RECEIVE:   [CR]
```



## Generate a Square Wave Output

fxx,xx      Produce a free-running square wave on bit xx with period xx.

This function will cause bit xx to change state every xx timebase units, effectively generating a square wave with period xx, or frequency  $1 \text{ div } (xx*2)$ .

Examples:

Start a 1Hz square wave on bit 2. Bit two must be configured as output using ML, such as ML02. Also, the timebase is assumed to have been configured to the default (2400, or 10ms), using S2400, or S0000.

```
SEND:      f02,32
RECEIVE:   [CR]
```

**Note:** It is possible to synchronize the starting edges of any free-running outputs by following the procedure outlined in the application note of the Set Timebase command. (SCxxxx)

## Read Change-of-State

Y            Read COST bit.

The pod can set a change-of-state flag for any input that has been configured to do so. This command will read and then reset that bit. Therefore, this command will always return “N[CR]” unless the T command has first been used to enable change-of-state detect for any given bit.

If a change-of-state has been detected since the last “Y” command (see note), the pod will return “Y[CR]” otherwise “N[CR]” will be returned.

Example:

Read COST bit.

```
SEND:      Y
RECEIVE:   N[CR]
```

**Note:** The address command for any given pod will also return “Y” or “N” and clear the Change-of-state flag in the pod.

## Enable Change-of-State Detection

```
TLxx      Set COST mask for bits 0-7
TMxx      Set COST mask for bits 8-F
THxx      Set COST mask for bits 10-17
```

These commands configure the bit-by-bit mask to enable a change-of-state to set the COST flag at the pod for readback by the “Y” or address commands. If a “one” is set for a particular bit, that bit will set the COST flag if/when the bit changes state. A “zero” will disable change-of-state detection.

Examples:

Allow bit 13 and only bit 13 to set the COST flag.

```
SEND:      TL00
RECEIVE:   [CR]
SEND:      TM00
RECEIVE:   [CR]
SEND:      TH08
RECEIVE:   [CR]
```

Allow a change of state on ANY bit to set the COST flag.

```
SEND:      TLFF
RECEIVE:   [CR]
SEND:      TMFF
RECEIVE:   [CR]
SEND:      THFF
RECEIVE:   [CR]
```

**Note:** The COST Flag is read via either the “Y” command or a valid address command. The COST Flag is reset to FALSE by either command.

## Select Which Edge Will Increment Counters

```
dx±       Set Digital input active state on bit x
dxx±     Set Digital input active state on bit xx
```

These commands allow you to set whether a rising or falling edge will increment the digital input counter; i.e., if all bits are set to rising edge, the digital input counter for any given bit will increment each time a rising edge is detected. “+” is rising edge, “-” is falling edge.

Examples:

Set bit 1 to rising edge active.

```
SEND:      D1+
           or
SEND:      D01+
RECEIVE:   [CR]
```

Set bit 17 to falling edge active.

```
SEND:      D17-  
RECEIVE:   [CR]
```

**Note:** The digital input counters are read with the “cxx” command, and reset with the “rxx” command.

## Read Digital Input Counter & Read Time Left on Pulse or Free Run Output

```
cxx      Read digital input counter xx  
cxx      Read pulse or free-run status and reload counters
```

This command performs two duties, depending on whether the bit is configured as input or output. If the bit is configured as an input, this command will read how many times bit xx has changed to its active state (as configured with dx± or dxx±) since the last reset command (rxx).

If this bit is configured as an output, this command will indicate how much time (in time-base units) remains in a pulse-output or free-run-output before the pulse terminates. If the output is configured as free-run, it also returns what period is programmed into the output counter.

Input counters are configured as 16-bit counters. Counter content is provided most significant bit first.

Output return values are divided into two eight-bit counters. The first byte of the output counters is the time-remaining before the output pulse expires, the second byte is the originally-programmed period of free-run outputs. The second byte is zero for pulse outputs.

Examples:

Read digital input counter for bit #1.

```
SEND:      C01  
RECEIVE:   0213[CR]    ;assuming 213hex edges since last reset of counter
```

Read pulse output counter for bit #F.

```
SEND:      C0F  
RECEIVE    1F00        ;1F is number of timebase units remaining before pulse  
                                expires, 00 indicates pulse, not free-run.
```

Read free-run output counter for bit 17.

```
SEND      c17  
RECEIVE:  045F        ;04 is number of timebase units remaining before pulse  
                                expires, 5F indicates that the duration of each period is 5F  
                                units.
```

**Note:** It is possible to cause pulse and free-run outputs to prematurely expire or change state by using the SCxxxx variant of the Set TimeBase command. Also, it is possible to terminate pulse and free-run outputs without toggling the outputs by issuing a counter reset command (rxx) for each output to be terminated.

## Reset Counter & Turn Off Pulse or Free-Run Output

rxx            Reset digital input counter xx  
 rxx            Turn-off digital output pulse or free-run xx

This command is normally used to reset a digital input counter to zero. It can also be used to stop digital output pulses (Oxx±xx) or free-run outputs (fxx,xx)

Examples:

Reset digital input counter for digital input number 3.

```
SEND:      r03
RECEIVE:   [CR]
```

Stop free-run digital output on output number 14.

```
SEND:      r14
RECEIVE:   [CR]
```

## Read Firmware Revision Number

V            Read the firmware revision number

This command is used to read the version of firmware installed in the pod. It returns “X.XX[CR]”.

Example:

Read the RDIO24 version number.

```
SEND:      V
RECEIVE:   1.00[CR]
```

**Note:** The “H” command returns the version number along with other information.

## Resend Last Response

n            Resend last response

This command will cause the pod to return the same thing it just sent. This command works for all responses less than 255 characters in length. Normally this command is used if the host detected a parity or other line fault while receiving data, and needs the data to be sent a second time.

The “n” command may be repeated.

Example:

Assuming the last command was “I”, ask pod to resend last response.

SEND:           n  
RECEIVE:       FFFFFF[CR] ;or whatever the data was

**Note:** This command may not be used for the FASTDATA-L, -M, or -H commands, as they exceed the 255 character limit. Use the “D” command to perform the same task for these three commands.

## Hello Message

H\*           Hello message

Any string of characters starting with “H” will be interpreted as this command. (“H[CR]” alone is also acceptable.) The return from this command takes the form (without the quotes):

“=Pod aa, RDIO24 Rev rr Firmware Ver:x.xx

aa            is the pod address  
rr            is the hardware revision, such as “B1”  
x.xx          is the software revision, such as “1.00”

Example:

Read the greeting message.

SEND:           Hello?  
RECEIVE:       =Pod 00, RDIO24 Rev B1 Firmware Ver:1.00

## Set a New Baud Rate (When shipped the baud rate is set at 9600.)

BAUD=xxx       Program the pod with a new baud rate

This command sets the pod to communicate at a new baud rate. The parameter passed, xxx, is slightly unusual. Each x is the same digit from the following table:

CODE	BAUD RATE
0	1200
1	2400
2	4800
3	9600
4	14400
5	19200
6	28800

Therefore, valid values for the command's "xxx" are 000, 111, 222, 333, 444, 555, or 666.

The pod returns a message indicating it will comply. The message is sent in the old baud rate, not the new one. Once the message is transmitted, the pod changes to the new baud rate. The new baud rate is stored in EEPROM and will be used even after power-reset, until the next "BAUD=xxx" command is issued.

Example:

Set the pod to 19200 baud.

```
SEND:          BAUD=555
RECEIVE:       =:Baud:05[CR]
```

Set the pod to 9600 baud.

```
SEND:          BAUD=333
RECEIVE:       =:Baud:03[CR]
```

**Note:** There is an undocumented Code 7 which selects 57600 baud. This code can be used for testing but only with short cables. Use of this code is not recommended due to possible difficulty with character drop-out and other line noise problems.

## Program the Pod Address

A=xx      Program the currently selected pod to respond at **address xx**

This command changes the pod's address to xx. If the new address is 00, the pod will be placed into non-addressed mode. If the new address is not 00, the pod will not respond to further communications until a valid address command is issued. Hex numbers 00-FF are considered valid addresses. The R specification allows only 32 drops on the line, so some addresses may be unused.

The new pod address is saved in EEPROM and will be used even after power-down until the next "A=xx" command is issued. Note that, if the new address is not 00 (i.e., the pod is configured to be in addressed mode), it is necessary to issue an address command to the pod at the new address before it will respond.

The pod returns a message containing the pod number as confirmation.

Example:

Set the pod address to 01.

```
SEND:          A=01
RECEIVE:       =:Pod#01[CR]
```

Set the pod address to F3.

```
SEND:      A=F3
RECEIVE:   =:Pod#F3[CR]
```

Take the pod out of addressed mode.

```
SEND:      A=00
RECEIVE:   =:Pod#00[CR]
```

## Read and Store Digital Input Data

```
FASTDATAL   Read digital bits 0-7 as fast as possible
FASTDATAM   Read digital bits 8-F as fast as possible
FASTDATAH   Read digital bits 10-17 as fast as possible
```

These commands read the respective byte of digital input data and store it in RAM at the fastest possible rate: 21 microseconds between samples. The commands will store as much data as the pod can hold: RAM size-1KByte. Typically, this is 7Kbytes of data, however, a 32k RAM version is optionally available, which would provide 31Kbytes of data storage.

Once the data has been stored, it is dumped to the serial port. The data is formatted into 3-byte chunks, followed by a space: xxxxxx xxx...etc. There are no carriage returns until the last byte has been sent.

All normal pod activities (parsing commands, receiving commands, pulse output countdowns, free-run generation, COST detect, etc.) STOP until the serial data is done transmitting. NOTHING else works until the data has been dumped.

## Re-send Data

D This will dump the last stored historical data to the serial port.

Data can be resent by issuing a “D” command. This will dump the last stored historical data to the serial port and can be used, for example, if line noise or similar problems are suspected.

This command should only be used after FASTDATAL, FASTDATAM, or FASTDATAH have been issued, because random data fills the buffer until one of these commands acquire data.

The format of the data is identical to the FASTDATAx commands. See the previous description of the FASTDATAx command for more information about the format and length of returned data.

Example:

Resend the data buffer.

```
SEND:      D
RECEIVE:   xxxxxx xxxxxx xxxxxx ... xxxxxx for size of buffer.
```

## Enter a New Program

PROGRAM=        This command initiates transfer of a new program to the RDIO24.

This command should be used carefully. If you accidentally issue a “PROGRAM=” command, ESC (ASCII 27) will restart the pod as if power had been reset. This feature is designed to allow field-upgrades to the RDIO24 firmware, and, for advanced users, the opportunity to customize the firmware in the pod. Documentation relating to use of this command is provided with the upgrade diskette, or is available separately for a small fee.

## Error Codes:

---

The following error codes can be returned from the pod:

- 1: Invalid channel number (too large, or not a number. All channel numbers must be between 00 and 17, in hex. (0-24 decimal)).
- 3: Improper Syntax. (Not enough parameters is the usual culprit).
- 4: Channel number is invalid for this task (For example if you try to output to a bit that is set as an input bit, that will cause this error).
- 9: Parity error. (This occurs when some part of the received data contains a parity or framing error).

Additionally, several full-text error codes are returned. All begin with “Error, ” and are useful when using a terminal to program the pod.

Error, Unrecognized Command: {command received}[CR]

This occurs if the command is not recognized.

Error, Command not fully recognized: {Command received}[CR]

This occurs if the first letter of the command is valid, but the remaining letters are not.

Error, Address command must be CR terminated[CR]

This occurs if the address command (!xx[CR]) has extra characters between the pod number and the [CR].



# Chapter 4: Specifications

## Serial Communications Interface

Serial Port:	Opto-isolated Matlabs type LTC491 Transmitter/ Receiver. Compatible with RS-485 specification. Up to 32 drivers and receivers allowed on line. I/O bus programmable from 00 to FF hex (0 to 255 decimal). Whatever address is assigned is stored in EEPROM and used as default at next Power-On.
Input Common Mode Voltage:	300V minimum (opto-isolated). If opto-isolators are by-passed: -7V to +12V.
Receiver Input Sensitivity:	±200 mV, differential input.
Receiver Input Impedance:	12KΩ minimum
Transmitter Output Drive Capability:	60 mA, 100 mA short circuit current capability.
Serial Data Rates:	Programmable for 1200, 2400, 4800, 9600, 14400, 19200, and 28800 baud. Crystal oscillator provided.

## Digital Inputs

Number:	Up to 24. Can be programmed, on a bit-by-bit basis, on an 8-bit byte basis, or on a 24-bit word basis. In this latter case, there would be no capability for digital outputs.
Sample Rate:	Programmable from 14 Hz to 1 KHz.
Software Counters:	There are 16-bit software counters on all bits programmed to be inputs. These can be programmed to increment on either rising or falling edges.
Change of State Detection:	Change-of-state flags can be set on any enabled input bits and can be read via the serial port.
Logic Input Low:	-0.5V to +0.8V.
Logic Input High:	+2.0V to +50.0V
Low-level Input Current:	-450 mA maximum.

## Digital Outputs

Number:	Up to 24. Can be programmed, on a bit-by-bit basis, on an 8-bit byte basis, or on a 24-bit word basis. In this latter case, there would be no capability for digital inputs. (The inputs would read back the output value only.).
Type:	Outputs can be latched, pulsed, or set to free-run for a prescribed period of time. Pulsed outputs are square wave and programmable from 7 Hz to 500 Hz.

Logic-Low Output Current: 350 mA maximum. (See note below.) Inductive suppression diode included in the each circuit.

**Note:** Maximum allowable current per output bit is 350 mA but, for the seven-bit group, there is a maximum cumulative total of 650 mA. Output groups are bits 0-5, 6-11, 12-17, and 18-23 (decimal).

High-Level Output Voltage: Open Collector, compliance with up to 50VDC. If no user-supplied voltage, outputs pulled up to +5VDC via 10 K $\Omega$  resistors.

## Environmental

Operating Temperature Range: 0° to 65°C (Optional -40° to +80°C).  
 Temperature De-rating: Ambient temperature de-rating must be considered based on the power level applied. When 7.5VDC is applied, the temperature rise inside the enclosure is 7.3°C above the ambient temperature.

The equation to determine maximum temperature de-rating is:

$$V_{I(TJ=120)} < 22 - T_A / 7$$

Where  $T_A$  is the ambient temperature in °C and  $V_{I(TJ=120)}$  is the voltage at which the integral voltage regulator junction temperature will rise to a temperature of 120°. (Note: The junction temperature is rated to 150°C maximum.)

For example, at an ambient temperature of 25°C, the voltage  $V_I$  can be up to 18.4V. At an ambient temperature of 100°F (37.8°C), the voltage  $V_I$  can be up to 16.6V.

Storage Temperature Range: -50° to +120°C.  
 Humidity: 5% to 95% RH non-condensing.  
 Size: Enclosure 8.5" long by 5.25" wide by 2" high.

## Power Required

Power can be applied from the computer's +12VDC power supply for the opto-isolated section via the serial communication cable and from a local power supply for the rest of the unit.

Opto-Isolated Section: 7.5 to 35 VDC @ 7 mA. (Note: Due to the small amount of current required, voltage drop in the communication cable is not significant.)

Local Power: 7.5 to 16 VDC @ 150mA. (See box below.)

If the local power supply has an output voltage greater than 16VDC, you can install a zener diode in series with the supply voltage. The voltage rating of the zener diode ( $V_Z$ ) should be equal to  $V_I - 16$  where  $V_I$  is the power supply voltage. The power rating of the zener diode should be  $V_Z \times 0.12$  watts. Thus, for example, a 24VDC power supply would require using an 8.2V zener diode with a power rating of  $8.2 \times 0.12 \gg 1$  watt.

# Appendix A: Application Considerations

## Introduction

---

Working with RS-422 and RS-485 devices is not much different from working with standard RS-232 serial devices and these two standards overcome deficiencies in the RS-232 standard. First, the cable length between two RS-232 devices must be short; less than 50 feet at 9600 baud. Second, many RS-232 errors are the result of noise induced on the cables. The RS-422 standard permits cable lengths up to 5000 feet and, because it operates in the differential mode, it is more immune to induced noise.

Connections between two RS-422 devices (with CTS ignored) should be as follows:

Device #1		Device #2	
Signal	Pin No.	Signal	Pin No.
Gnd	7	Gnd	7
TX+	24	RX+	12
TX-	25	RX-	13
RX+	12	TX+	24
RX-	13	TX-	25

A third deficiency of RS-232 is that more than two devices cannot share the same cable. This is also true for RS-422 *but RS-485 offers all the benefits of RS-422 plus allows up to 32 devices to share the same twisted pairs.* An exception to the foregoing is that multiple RS-422 devices can share a single cable if only one will talk and the others will all receive.

## Balanced Differential Signals

The reason that RS-422 and RS-485 devices can drive longer lines with more noise immunity than RS-232 devices is that a balanced differential drive method is used. In a balanced differential system, the voltage produced by the driver appears across a pair of wires. A balanced line driver will produce a differential voltage from  $\pm 2$  to  $\pm 6$  volts across its output terminals. A balanced line driver can also have an input “enable” signal that connects the driver to its output terminals. If the “enable signal is OFF, the driver is disconnected from the transmission line. This disconnected or disabled condition is usually referred to as the “tristate” condition and represents a high impedance. RS-485 drivers must have this control capability. RS-422 drivers may have this control but it is not always required.

A balanced differential line receiver senses the voltage state of the transmission line across the two signal input lines. If the differential input voltage is greater than +200 mV, the receiver will provide a specific logic state on its output. If the differential voltage input is less than -200 mV, the receiver will provide the opposite logic state on its output. A maximum operating voltage range is from +6V to -6V allows for voltage attenuation that can occur on long transmission cables.

A maximum common mode voltage rating of  $\pm 7V$  provides good noise immunity from voltages induced on the twisted pair lines. The signal ground line connection is necessary in order to keep the common mode voltage within that range. The circuit may operate without the ground connection but may not be reliable.

### RS-422 Specification Summary

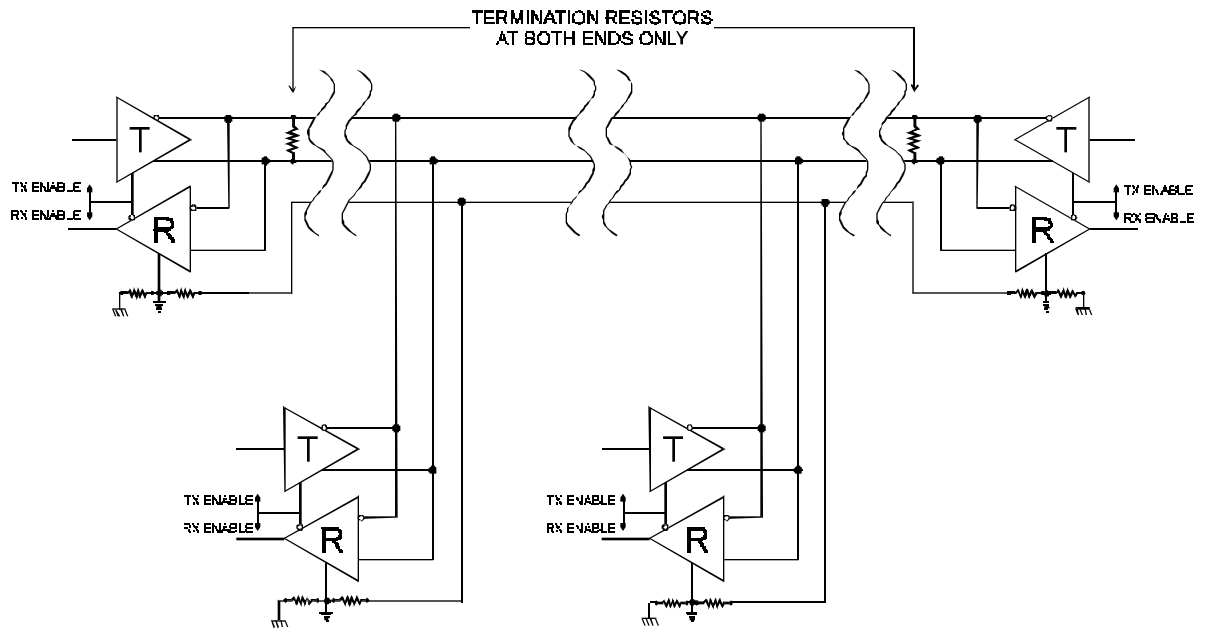
Parameter	Conditions	Min.	Max.
Driver Output Voltage (unloaded)		4V	6V
		-4V	-6V
Driver Output Voltage (loaded)	LD and LDGND jumpers in	2V	
		-2V	
Driver Output Resistance			50W
Driver Output Short-Circuit Current			$\pm 150$ mA
Driver Output Rise Time			10% unit interval
Receiver Sensitivity			$\pm 200$ mV
Receiver Common Mode Voltage Range			$\pm 7V$
Receiver Input Resistance			4KW

To prevent signal reflections in the cable and to improve noise rejection in both the RS-422 and RS-485 mode, the receiver end of the cable should be terminated with a resistance equal to the characteristic impedance of the cable. (An exception to this is the case where the line is driven by an RS-422 driver that is never “tri-stated” or disconnected from the line. In this case, the driver provides a low internal impedance that terminates the line at that end.)

### RS-485 Data Transmission

The RS-485 Standard allows a balanced transmission line to be shared in a party-line mode. As many as 32 driver/receiver pairs can share a two-wire party line network. Many characteristics of the drivers and receivers are the same as in the RS-422 Standard. One difference is that the common mode voltage limit is extended and is  $+12V$  to  $-7V$ . Since any driver can be disconnected (or tri-stated) from the line, it must withstand this common mode voltage range while in the tristate condition.

The following illustration shows a typical multidrop or party line network. Note that the transmission line is terminated on both ends of the line but not at drop points in the middle of the line.



**Figure A-1: Typical RS-485 Two-Wire Multidrop Network**

This page intentionally left blank

# *Declaration of Conformity*



6260 Sequence Drive  
San Diego, CA 92121-4371  
(800) 523-2320

Industrial Computer Source declares under its own and full responsibility that the following products are compliant with the protection requirements of the 89/336/EEC and 73/23/EEC directives.

**Only specific models listed on this declaration and labeled with the CE logo are CE compliant.**

## **RDIO24**

Conformity is accomplished by meeting the requirements of the following European harmonized standards:

**EN 50081-1:1992** Emissions, Generic Requirements.

-EN 55022 Measurement of radio interference characteristics of information technology equipment.

**EN 50082-1:1992** Immunity, Generic Requirements.

-IEC 801-2:1984 Immunity for AC lines, transients, common, and differential mode.

-IEC 801-3:1984 Immunity for radiated electromagnetic fields.

-IEC 801-4:1988 Immunity for AC and I/O lines, fast transient common mode.

**EN 60950:1992** Safety of Information Technology Equipment.

Information supporting this declaration is contained in the applicable Technical Construction file available from:



Z.A. de Courtaboeuf  
16, Avenue du Québec  
B.P. 712  
91961 LES ULIS Cedex

Mr. Steven R. Peltier  
President & Chief Executive Officer

September 10, 1997  
San Diego, CA





## BUG REPORT

While we have tried to assure this manual is error free, it is a fact of life that works of man have errors. We request you to detail any errors you find on this BUG REPORT and return it to us. We will correct the errors/problems and send you a new manual as soon as available. Please return to:



**INDUSTRIAL COMPUTER SOURCE®**

Attn: Documentation Department

P. O. Box 910557

San Diego, CA 92121-0557

Your Name: \_\_\_\_\_

Company Name: \_\_\_\_\_

Address 1: \_\_\_\_\_

Address 2: \_\_\_\_\_

Mail Stop: \_\_\_\_\_

City: \_\_\_\_\_ State: \_\_\_\_\_ Zip: \_\_\_\_\_

Phone: (\_\_\_\_) \_\_\_\_\_

Product: **RDIO24**

Manual Revision: **00650-124-2B**

Please list the page numbers and errors found. Thank you!

