



ACCES I/O PRODUCTS INC
10623 Roselle Street, San Diego, CA 92121
TEL (858)550-9559 FAX (858)550-7322

MODEL RAG128

USER MANUAL

Notice

The information in this document is provided for reference only. ACCES does not assume any liability arising out of 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 ACCES, nor the rights of others.

IBM PC, PC/XT, and PC/AT are registered trademarks of the International Business Machines Corporation.

Printed in USA. Copyright 2001 by ACCES I/O Products Inc, 10623 Roselle Street, San Diego, CA 92121. All rights reserved.

WARNING!!

ALWAYS CONNECT AND DISCONNECT YOUR FIELD CABLING WITH THE COMPUTER POWER OFF. ALWAYS TURN COMPUTER POWER OFF BEFORE INSTALLING A CARD. CONNECTING AND DISCONNECTING CABLES, OR INSTALLING CARDS INTO A SYSTEM WITH THE COMPUTER OR FIELD POWER ON MAY CAUSE DAMAGE TO THE I/O CARD AND WILL VOID ALL WARRANTIES, IMPLIED OR EXPRESSED.

Warranty

Prior to shipment, ACCES equipment is thoroughly inspected and tested to applicable specifications. However, should equipment failure occur, ACCES assures its customers that prompt service and support will be available. All equipment originally manufactured by ACCES which is found to be defective will be repaired or replaced subject to the following considerations.

Terms and Conditions

If a unit is suspected of failure, contact ACCES' Customer Service department. Be prepared to give the unit model number, serial number, and a description of the failure symptom(s). We may suggest some simple tests to confirm the failure. We will assign a Return Material Authorization (RMA) number which must appear on the outer label of the return package. All units/components should be properly packed for handling and returned with freight prepaid to the ACCES designated Service Center, and will be returned to the customer's/user's site freight prepaid and invoiced.

Coverage

First Three Years: Returned unit/part will be repaired and/or replaced at ACCES option with no charge for labor or parts not excluded by warranty. Warranty commences with equipment shipment.

Following Years: Throughout your equipment's lifetime, ACCES stands ready to provide on-site or in-plant service at reasonable rates similar to those of other manufacturers in the industry.

Equipment Not Manufactured by ACCES

Equipment provided but not manufactured by ACCES is warranted and will be repaired according to the terms and conditions of the respective equipment manufacturer's warranty.

General

Under this Warranty, liability of ACCES is limited to replacing, repairing or issuing credit (at ACCES discretion) for any products which are proved to be defective during the warranty period. In no case is ACCES liable for consequential or special damage arriving from use or misuse of our product. The customer is responsible for all charges caused by modifications or additions to ACCES equipment not approved in writing by ACCES or, if in ACCES opinion the equipment has been subjected to abnormal use. "Abnormal use" for purposes of this warranty is defined as any use to which the equipment is exposed other than that use specified or intended as evidenced by purchase or sales representation. Other than the above, no other warranty, expressed or implied, shall apply to any and all such equipment furnished or sold by ACCES.

Table of Contents

Notice	iii
Warranty	iv
FOREWORD	iii
Guarantee	iv
Limited Warranty	iv
Return Procedure	v
Limitation of Liability	v
Advisories	vi
Chapter 1: Introduction	1-1
Description	1-1
Specifications	1-3
Chapter 2: Installation	2-1
CD Installation	2-1
3.5-Inch Diskette Installation	2-1
Directories Created on the Hard Disk	2-2
Installing the Pod	2-4
Input/Output Pin Connections	2-7
Chapter 3: Software	3-1
Command Structure	3-2
Command Functions	3-7
Error Codes	3-15
Appendix A: Application Considerations	A-1
Introduction	A-1
Balanced Differential Signals	A-1
RS485 Data Transmission	A-3
Cable Termination	A-4

List of Figures

Figure 1-1: RAG128 Block Diagram	1-7
Figure 2-1: Option Selection Map	2-6
Figure A-1: Typical RS485 Two-Wire Multidrop Network	A-3

List of Tables

Table 2-1: 37 Pin Connector Assignments	2-7
Table 3-1: RAG128 Command List	3-6
Table A-1: Specification Summary for the RS232c, RS422 and RS485 Communications Standards	A-2

Chapter 1: Introduction

Features

- Remote Intelligent Analog and Digital I/O Units with Opto-Isolated RS485 Serial Interface to Host Computer.
- Eight Single-ended Analog Inputs, Software Programmable Gain on a Channel-by-Channel Basis, 12-Bit Resolution A/D Converter.
- Eight Bits of TTL/CMOS-Compatible Digital I/O Configured on a Bit-by-Bit Basis as either 8 Inputs or 7 Outputs.
- Digital Output Voltages up to 50V.
- Open Collector Digital Outputs for External Loads up to 350mA.
- Protective Metal Enclosure.
- Type 8031 Microcontroller with 32K x 8 bits RAM and 8K x 8 bits EEPROM (32K x 8 Bits EEPROM Optional).
- All Programming in Software, No Switches to Set. Jumper wires to select two or four wire communications mode, and Jumpers (3) to By-Pass Opto-Isolators if desired.
- On-board Crystal Clock and Three 16-Bit Counter/Timers

Description

RAG128 is an intelligent, 8-channel, analog-to-digital converter unit that communicates with the host computer via EIA RS422, Half-Duplex, serial communications protocol. ASCII-based command/response protocol permits communication with virtually any computer system. RAG128 is one of a series of remote intelligent Pods called the "REMOTE ACCES Series". As many as 32 REMOTE ACCES Series Pods (or other RS485 devices) may be connected on a single two- or four-wire multidrop RS485 network. RS485 repeaters may be used to extend the number of Pods on a network. Each unit has a unique address. Communication uses a master/slave protocol wherein the Pod talks only if questioned by the computer.

An 8031 (with 32k x 8 bits RAM, 32K x 8 bits non-volatile EEPROM, and a watchdog timer circuit) gives RAG128 the capability and versatility expected from a modern distributed control system. RAG128 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 57.6 Kbaud and distances up to 4000 feet with low-attenuation twisted-pair cabling, such as Belden #9841 or equivalent. 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.

All programming of RAG128 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 REMOTE ACCES Series Pods with virtually any computer that has an RS485 port.

The module, or Pod address is programmable from 00 to FF hex and whatever address is assigned is stored in EEPROM and used at the next Power-ON. Similarly, the baud rate is programmable for 1200, 2400, 4800, 9600, 14400, 19200, 28800, and 57600 baud. The baud rate is stored in EEPROM and used at the next Power-ON. The Pod is shipped from the factory with preprogrammed address 00 and with the selected baud rate of 9600.

Note

Always write the Pod address and selected baud rate on the label of the steel enclosure.

Analog Inputs

The RAG128 accepts up to eight single-ended analog inputs. Input ranges of $\pm 10\text{V}$, $\pm 5\text{V}$, 0-10V, and 0-5V are available and all inputs are internally protected up to $\pm 16.5\text{V}$. RAG128 uses a 12-bit successive approximation analog-to-digital converter (A/D) capable of achieving a system throughput of 10,000 conversions per second. A/D conversions may be initiated either by software command, or by a microcontroller-provided programmable timer.

The RAG128 provides full compatibility with AIM-16 analog input expansion cards. Each AIM-16 provides capability to connect sixteen differential inputs, allowing as many as 128 differential inputs if you use a RAG128 with a full complement of AIM-16s. The $\pm 12\text{VDC}$ power required by the AIM-16 must be supplied locally because of the amount of current required. Alternatively, you can use the AIM-16P-S06, which requires only +5VDC of local power. The RAG128 is also compatible with the LVDT-8 and SSH-08 signal conditioning cards, however, the +12VDC power for these must be supplied locally.

Digital I/O (Port 0)

Eight bits of TTL/CMOS-compatible digital I/O, (8 input / 7 output) are provided. These can be software configured on a bit-by-bit basis as either inputs or outputs. When used as outputs, a maximum of seven bits are available. Each output provides compliance with user-supplied voltages up to 50 VDC and the maximum current per output bit is 350mA. When all seven-bits are used as outputs, there is a maximum total current of 650mA.

There are an additional eight bits which are used strictly for digital output. This is identified as Port 1, which handles bits 8-15. The setup software has the option of installing operating firmware supporting either the AIM-16 external analog multiplexer or more digital (output only) bits. When the RAG128 is used with an AIM-16 Multiplexer card then output bits GN0-GN2 and SEL0-SEL3 are used for gain and channel select. The final output, OP4, is available for general use. Each output has the ability to drive 10 LSTTL loads.

Counter/Timers

Three 16-bit Counter/Timers are provided on the RAG128. Counter/Timer 0 is enabled by a digital input and may be clocked either by the output of Counter 1 or by an external source up to 10Mhz. This counter is not committed on the card and its clock, gate, and output lines are available at the I/O connector.

Counter/Timers 1 and 2 are concatenated to form a 32-bit counter/timer. Counter 1 may be enabled by a digital input and is clocked by an on-board, crystal-controlled 10Mhz oscillator. The output of Counter 1 provides the clock for Counter 2 and is also available at the I/O connector. The output of Counter 2 is connected to input T0 of the microprocessor and is available for custom firmware options.

Watchdog Timer

The built-in watchdog timer resets the Pod if for some unexpected reason the microcontroller "hangs up" or the power supply voltage drops below 4.75 VDC.

Specifications

Serial Communications Interface

- Serial Port: Opto-isolated Matlabs type LTC491 Transmitter/Receiver. Compatible with RS485 specification. Up to 32 drivers and receivers allowed on the line. I/O bus addresses programmable from 00 to FF hex (0 to 255 decimal) and stored in EEPROM as default at next Power-On.
- Asynchronous Data Format: 7 data bits, even parity, one stop bit
- 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: 12KW minimum
- Transmitter Output Drive: 60 mA, 100 mA short circuit current capability.
- Serial Data Rates: Programmable for 1200, 2400, 4800, 9600, 14400, 19200, 28800, and 57600 baud. Crystal oscillator provided.

Analog Inputs

- Channels: Eight, single-ended with common ground.
- Voltage Ranges: Software selectable, $\pm 5V$, $\pm 10V$, $0-5V$, $0-10V$.
- Resolution: 12 binary bits.
- Accuracy: 0.025% of Reading ± 1 LSB (est.).
- Input Impedance: 21K Ω Unipolar, 16K Ω Bipolar, 10M Ω with AIM-16.
- Coding: True binary for unipolar inputs and offset binary for bipolar inputs.
- Throughput: 10,000 conversions per second in foreground mode. 6.67K conversions per second in background mode.
- A/D Type: Successive Approximation.
- Gain Drift: ± 5 ppm/ $^{\circ}C$.
- Trigger Source: Software command or on-board programmable timer.
- Non-Linearity and Noise: $1\frac{1}{2}$ LSB.

Digital I/O

- Eight bits configured as input or output on an individual basis except for bit D17 which is dedicated as input only.
- Digital Inputs Logic High: $+2.0V$ to $+5.0V$ at 20 A max. (5mA max at 50V in)
Logic Low: $-0.5V$ to $+0.8V$ at 0.4 mA max.
- Digital Outputs Logic-Low Output Current: 350 mA maximum. (See note below.)
Inductive kick suppression diode included in the each circuit.

Note

Maximum allowable current per output bit is 350 mA. When all eight bits are used, there is a maximum total current of 650 Ma.

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

Digital Outputs

- Eight bits, each with 10 LSTTL load drive capability.
- There are eight additional bits used strictly for digital outputs. If you are not using a multiplexer such as the AIM-16, the setup software allows you to use the GN0-GN2 and SEL0-SEL3 bits directly. OP4 is available for general use.

Programmable Timer

- Type: 82C54-2 programmable interval timer.
- Counters: Three 16-bit down counters, two permanently concatenated with 10 MHz clock as programmable timer. One counter is uncommitted.
- Output Drive: 4 mA at 0.4V (10 LSTTL loads).
- Input Gate: TTL/DTL/CMOS compatible.
- Clock Input Frequency: DC to 10 MHz.
- Active Count Edge: Negative edge.
- Minimum Clock Pulse Width: 30 nS high/50 nS low.
- Timer Range: 2.5 MHz to < 1 pulse per hour.

Environmental

- Operating Temperature Range: 0 °C. to 65 °C. (Optional -40 °C. to +80 °C.).
- Temperature De-rating: Based on the power applied, maximum operating temperature may have to be de-rated because internal power regulators dissipate some heat.

For example, when 7.5VDC is applied, the temperature rise inside the enclosure is 7.3 °C. above the ambient temperature.

Maximum operating temperature can be determined according to the following equation:

$$V_{I(T_J = 120)} < 30 - T_A/4.4$$

Where T_A is the ambient temperature in °C. and $V_{I(T_J = 120)}$ is the voltage at which the integral voltage regulator junction temperature will rise to a temperature of 120 °C.
(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 25V.
At an ambient temperature of 100°F (37.8°C), the voltage V_I can be up to 21.4V.

- Storage Temperature Range: -20 °C. to +70 °C.
- Humidity: 5% to 95% RH non-condensing.
- Size: Metal Enclosure 8.5" long by 5.25" wide by 2" high or watertight dustproof NEMA4 enclosure.

Power Required

- Opto-Isolated Section: 7.5 to 25 VDC @ 40 mA. (Note: Due to the small amount of current required, voltage drop in long cables is not significant.)
- Local Power: 7.5 to 16 VDC @ 150 mA. (See box that follows.)

Power can be applied from the computer's +12VDC power supply for the opto-isolated section in the serial communication cable and from a local power supply for the rest of the unit. If you do not wish to use power from the computer, a separate power supply may be used for the opto-isolated section.

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 $8.2 \times 0.12 = 1$ watt.

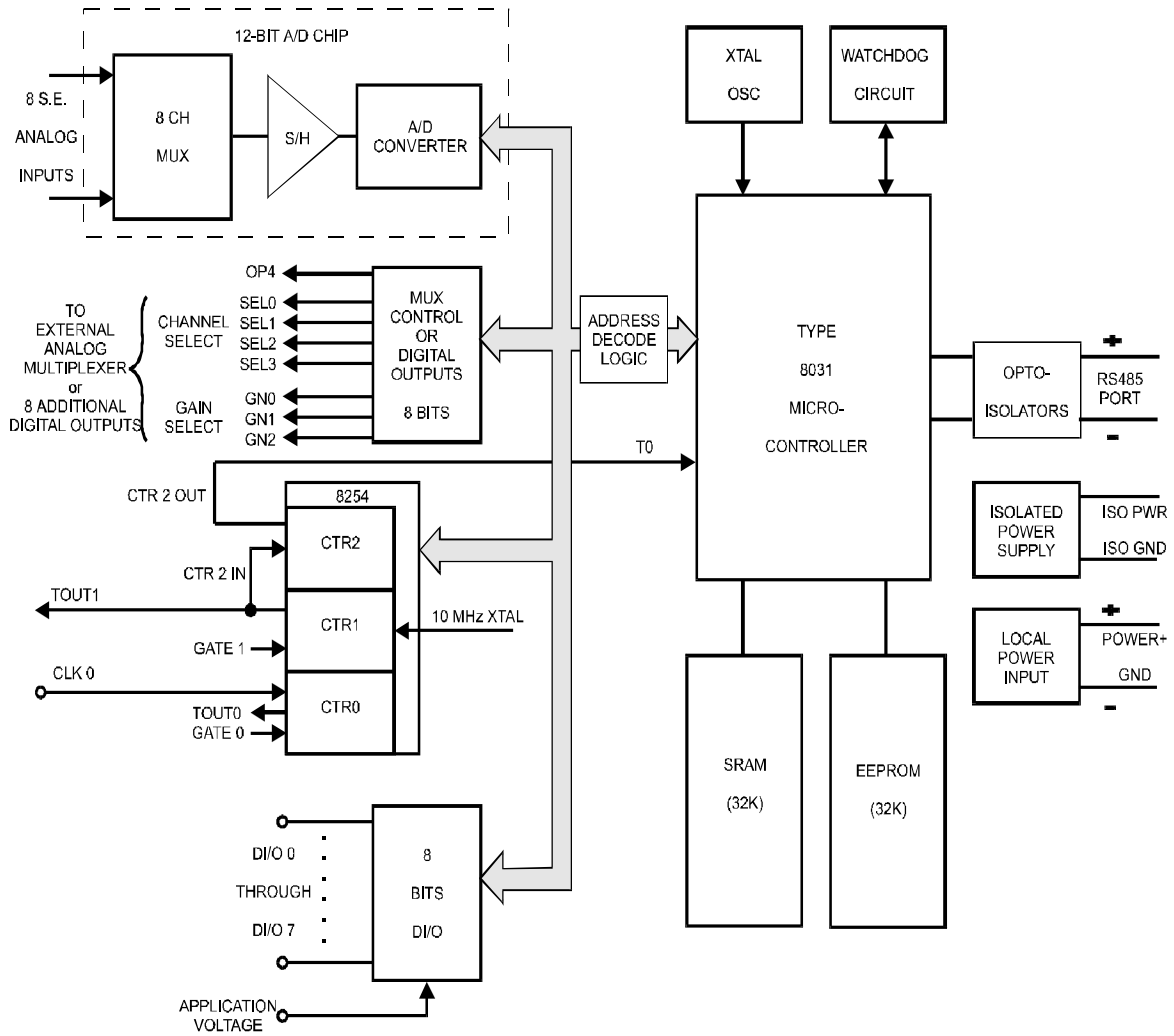


Figure 1-1: RAG128 Block Diagram

Chapter 2: Installation

The software provided with this card is contained on either one CD or multiple diskettes and must be installed onto your hard disk prior to use. To do this, perform the following steps as appropriate for your software format and operating system. Substitute the appropriate drive letter for your CD-ROM or disk drive where you see `d:` or `a:` respectively in the examples below.

CD Installation

DOS/WIN3.x

1. Place the CD into your CD-ROM drive.
2. Type `d:K` to change the active drive to the CD-ROM drive.
3. Type `installK` to run the install program.
4. Follow the on-screen prompts to install the software for this card.

WIN95/98/NT/2000

- a. Place the CD into your CD-ROM drive.
- b. The CD should automatically run the install program after 30 seconds. If the install program does not run, click `START | RUN` and type `d:install`, click `OK` or press `K`.
- c. Follow the on-screen prompts to install the software for this card.

3.5-Inch Diskette Installation

As with any software package, you should make backup copies for everyday use and store your original master diskettes 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:K
```

You will need to swap disks as requested by the system.

In a two-disk system, the command is:

```
diskcopy a: b:K
```

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

To copy the files on the master diskette to your hard disk, perform the following steps.

- a. Place the master diskette into a floppy drive.
- b. Change the active drive to the drive that has the diskette installed. For example, if the diskette is in drive A, type a:K.
- c. Type `installK` and follow the on-screen prompts.

Directories Created on the Hard Disk

The installation process will create several directories on your hard disk. If you accept the installation defaults, the following structure will exist.

[CARDNAME]

Root or base directory containing the `SETUP.EXE` setup program used to help you configure jumpers and calibrate the card.

DOS\PSAMPLES: A subdirectory of [CARDNAME] that contains Pascal samples.

DOS\CSAMPLES: A subdirectory of [CARDNAME] that contains "C" samples.

Win32\language: Subdirectories containing samples for Win95/98 and NT.

WinRISC.exe

A Windows dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem.

ACCES32

This directory contains the Windows 95/98/NT driver used to provide access to the hardware registers when writing 32-bit Windows software. Several samples are provided in a variety of languages to demonstrate how to use this driver. The DLL provides four functions (`InPortB`, `OutPortB`, `InPort`, and `OutPort`) to access the hardware.

This directory also contains the device driver for Windows NT, `ACCESNT.SYS`. This device driver provides register-level hardware access in Windows NT. Two methods of using the driver are available, through `ACCES32.DLL` (recommended) and through the `DeviceIOControl` handles provided by `ACCESNT.SYS` (slightly faster).

SAMPLES

Samples for using ACCES32.DLL are provided in this directory. Using this DLL not only makes the hardware programming easier (MUCH easier), but also one source file can be used for both Windows 95/98 and WindowsNT. One executable can run under both operating systems and still have full access to the hardware registers. The DLL is used exactly like any other DLL, so it is compatible with any language capable of using 32-bit DLLs. Consult the manuals provided with your language's compiler for information on using DLLs in your specific environment.

VBACCES

This directory contains sixteen-bit DLL drivers for use with VisualBASIC 3.0 and Windows 3.1 only. These drivers provide four functions, similar to the ACCES32.DLL. However, this DLL is only compatible with 16-bit executables. Migration from 16-bit to 32-bit is simplified because of the similarity between VBACCES and ACCES32.

PCI

This directory contains PCI-bus specific programs and information. If you are not using a PCI card, this directory will not be installed.

SOURCE

A utility program is provided with source code you can use to determine allocated resources at run-time from your own programs in DOS.

PCIFind.exe

A utility for DOS and Windows to determine what base addresses and IRQs are allocated to installed PCI cards. This program runs two versions, depending on the operating system. Windows 95/98/NT displays a GUI interface, and modifies the registry. When run from DOS or Windows3.x, a text interface is used. For information about the format of the registry key, consult the card-specific samples provided with the hardware. In Windows NT, NTioPCI.SYS runs each time the computer is booted, thereby refreshing the registry as PCI hardware is added or removed. In Windows 95/98/NT PCIFind.EXE places itself in the boot-sequence of the OS to refresh the registry on each power-up.

This program also provides some COM configuration when used with PCI COM ports. Specifically, it will configure compatible COM cards for IRQ sharing and multiple port issues.

WIN32IRQ

This directory provides a generic interface for IRQ handling in Windows 95/98/NT. Source code is provided for the driver, greatly simplifying the creation of custom drivers for specific needs. Samples are provided to demonstrate the use of the generic driver. Note that the use of IRQs in near-real-time data acquisition programs requires multi-threaded application programming techniques and must be considered an intermediate to advanced programming topic. Delphi, C++ Builder, and Visual C++ samples are provided.

Findbase.exe

DOS utility to determine an available base address for ISA bus , non-Plug-n-Play cards. Run this program once, before the hardware is installed in the computer, to determine an available address to give the card. Once the address has been determined, run the setup program provided with the hardware to see instructions on setting the address switch and various option selections.

Poly.exe

A generic utility to convert a table of data into an nth order polynomial. Useful for calculating linearization polynomial coefficients for thermocouples and other non-linear sensors.

Risc.bat

A batch file demonstrating the command line parameters of RISCTerm.exe.

RISCTerm.exe

A dumb-terminal type communication program designed for RS422/485 operation. Used primarily with Remote Data Acquisition Pods and our RS422/485 serial communication product line. Can be used to say hello to an installed modem. RISCTerm stands for Really Incredibly Simple Communications TERMinal.

Installing the Pod

The RAG128 enclosure is an easily-mounted metal 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 and serial communications wiring and a deeper notch at the other end for input and output wiring.

In applications where a watertight and/or dust-tight protection is needed the RAG128-SO5 is packaged in an aluminum NEMA 4 enclosure. When the RAG128 is included into a RIDACS configuration it is installed in a gasketed 14-gauge steel box with a hinged cover and stainless steel clamps.

Electrical power and RS485 connections to the Pod are made at Terminal Block TB1. Terminal markings and their functions are as follows:

- ISO GND and ISO PWR: This is the power connection for the isolator section which may be supplied from a separate network power supply or the computer's +12VDC supply via a pair of wires on the RS485 network. This power is independent of "local power". The voltage level can be from 7.5 VDC to 25 VDC. (An on-board voltage regulator regulates the power to +5 VDC.) RAG128 will require about 40 mA of current and, thus, the resistive voltage drops on the wires has to be taken into consideration. The resistance of 22 gauge wire is approximately 15 ohms/1000 feet. If separate power is not available, then these terminals can be jumpered to the "local power" terminals, which defeats the optical isolation.
- PWR+ and GND: These terminals are used to apply local power to the Pod from a local power supply. The voltage can be anywhere in the range from 7.5 VDC to 16 VDC. Higher voltage can be used, 24 VDC for example, if an external zener diode is used to reduce the voltage applied to the RAG128. (See the Specification section of this manual to determine the zener diode power rating required.)
- RX+ and RX-: These are differential Receive/transmit inputs for two wire RS485 communications or receive inputs for four-wire Half-Duplex RS422 communications. For RS422 four-wire communications cut jumpers X+ and X-.
- TX+ and TX-: These are the differential "Transmit" outputs for RS422, four-wire communications. For two-wire RS485 communications they are connected to Rxt/Txt and RX-/TX- through jumpers X+, X- on the printed circuit board.
- EXV: This terminal is for the "application power" or the user provided voltage level to which digital outputs are connected through the loads. Open-collector Darlington amplifiers are used at the outputs. Inductive suppression diodes are included in the output driver circuits to clamp their outputs to the EXV power. The application power level (EXV) can be as high as 50 VDC.

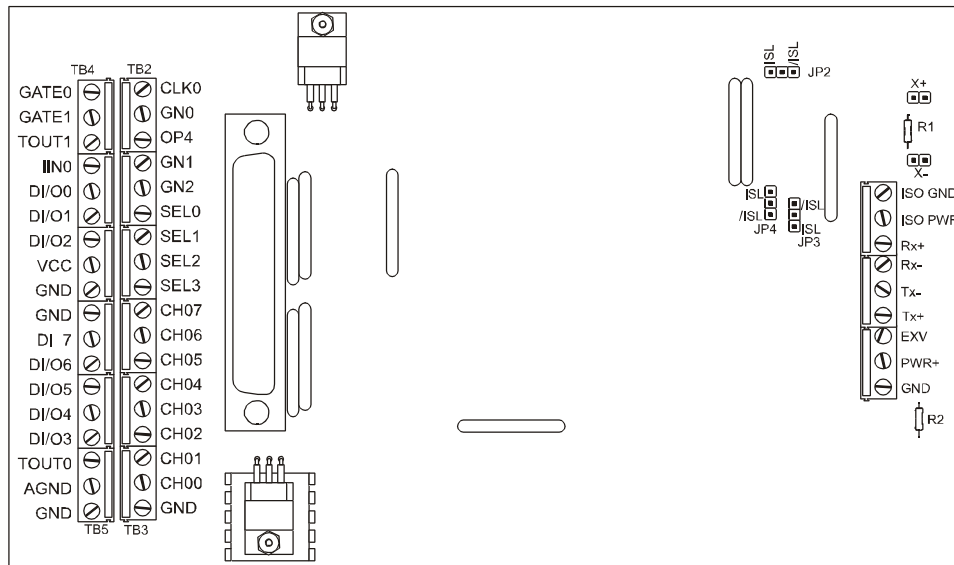


Figure 2-1: Option Selection Map

There are three jumper locations on the card and their functions are as follows:

- JP2, JP3, and JP4:** Normally these jumpers should be in the "ISL" position. If you wish to by-pass the opto-isolators, then you can move these jumpers to the "/ISL" Position.

Input/Output Pin Connections

Input and output connections can be made one of two ways: (a) at Euro style screw terminals TB2, TB3, TB4, and TB5; or (b) at 37-pin, male, sub-D connector J1. I/O connection points at the screw terminal connectors are as marked on the printed circuit board at terminals TB2 thru TB5. Connector pin assignments for the 37-pin are as follows:

Pin	Signal	Pin	Signal
1	N/C	19	N/C
2	CLK0 - Clock 0	20	N/C
3	GN0 - Digital Output Bit 8	21	GATE0 - Timer Gate 0
4	OP4 - Digital Output Bit 11	22	GATE1 - Timer Gate 1
5	GN1 - Digital Output Bit 9	23	TOUT1 - Timer Out 1
6	GN2 - Digital Output Bit 10	24	INT0 - Interrupt 0
7	SEL0 - Digital Output Bit 12	25	DI/O0 - Digital I/O Bit 0
8	SEL1 - Digital Output Bit 13	26	DI/O1 - Digital I/O Bit 1
9	SEL2 - Digital Output Bit 14	27	DI/O2 - Digital I/O Bit 2
10	SEL3 - Digital Output Bit 15	28	DIGITAL GROUND
11	DIGITAL GROUND	29	VCC - Regulated +5 DC Out
12	DI/O7 - Digital I/O Bit 7	30	CH07 - Analog Channel 7
13	DI/O6 - Digital I/O Bit 6	31	CH06 - Analog Channel 6
14	DI/O5 - Digital I/O Bit 5	32	CH05 - Analog Channel 5
15	DI/O4 - Digital I/O Bit 4	33	CH04 - Analog Channel 4
16	DI/O3 - Digital I/O Bit 3	34	CH03 - Analog Channel 3
17	TOUT0 - Timer Output 0	35	CH02 - Analog Channel 2
18	AGND - Analog Ground	36	CH01 - Analog Channel 1
		37	CH00 - Analog Channel 0

Table 2-1: 37 Pin Connector Assignments

Units that are marked for CE (European) Certification are installed in a NEMA 4 enclosure rather than the T-BOX metal enclosure referred to earlier. Further, to ensure that there is minimum susceptibility to EMI and minimum radiation, it is important that there be a positive chassis ground. Proper EMI cabling techniques (cable shields connected to chassis ground at the aperture, twisted pair wiring, and, in extreme cases, ferrite-level of EMI protection) must be used.

Chapter 3: Software

General

The RAG128 comes with software provided on CD. ASCII programming permits you to write applications in any high-level language that supports ASCII text string functions, allowing the "REMOTE ACCES" series modules to be used with virtually any computer that has an RS485 port.

The communication protocol has two forms: addressed and non-addressed. Non-addressed protocol is used when only one REMOTE ACCES Pod is to be used. Addressed protocol must be used when more than one REMOTE ACCES Pod is to be used. 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 specific Pod and the host computer. It enables communication with that specific Pod and disables all other REMOTE ACCES devices on the network.

If the non-addressed mode is used when there are multiple cards or pods in the system, all the cards or pods would attempt to respond simultaneously on the RS485 lines and this would corrupt the transmitted data. Also, of course, that would be a violation of the RS485 specification.

The sample programs are in forms suitable for use with QuickBASIC, C, and Pascal. The programs are as follows:

SETUP.EXE A menu-driven program that assists you in configuring and calibrating your RAG128. SETUP.EXE instructs you on the use of the opto-isolation jumpers and allows you to specify the Pod Address and the communication baud rate. This program can also auto-detect the address and/or baud rate of an attached card in the host computer. The setup software may also install one of two firmware sets into the RAG128. One provides support for the AIM-16 external analog multiplexer, and the other trades AIM-16 support for support of another digital I/O port. Finally, a calibration procedure tests the unit and determines the calibration correction values and stores them into internal storage. A sample program is provided to demonstrate the use of the calibration feature.

The Setup program should be run with only a single Pod attached to the RS485 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.

The setup software supports the ability to check calibration and to write correction values into EEPROM so they are available automatically on power-up. Calibration checks need only be performed periodically, not every time power is cycled. The calibration procedure requires the ability to ground an analog channel and a voltage source providing 4.5 VDC \pm 2 mV.

RISCTerm: This program provides a simple, non-automated interface to RS485 devices and can be used to configure or test RAG128 setup and operations. Help using RISCTerm is provided in-program and is accessed by pressing `ü`.

DRIVERS: `COMMDRV.PAS`, `COMMDRV.H`, and `COMMDRV.C` are communication shortcuts or drivers provided in C and Pascal source code. Three functions are provided: "initCommCard", "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.

Command Structure

All communication must be seven data bits, even parity, 1 stop bit. (If your RAG128 is a special version that works differently, there will be a Manual Addendum sheet at the front of this manual.) All numbers sent to and received from the Pod are in ASCII-coded 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 select 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 "[CR]". 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. When there's only one Pod connected as Pod 00, no address select command is needed.

You can issue commands listed in the RAG128 Command List that follows. 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. Commands are not case-sensitive, i.e., upper or lower case may be used.
- f. The symbol '*' means zero or more valid characters (total msg length < 255 decimal)

Concepts: Onboard Digital I/O

The RAG128 has 15 digital outputs, and eight digital inputs. Seven of the outputs are on the same pins at the cable terminus as the eight inputs, and are considered to comprise the DI/O port, or Port 0. The second eight digital outputs are generally dedicated to the control of sub-multiplexer products such as the LVDT-8, AIM-16, and SSH-0x. These second eight outputs are considered the MUX port, or Port 1. The use of Port 1 for multiplexer control or as general purpose outputs is determined by firmware installed from the setup program. The use of Port 1 by the user should be constrained to the non-multiplexer application. When Port 1 is used for multiplexer control, the firmware controls these outputs.

Concepts: The Point List

A Point, as used in the RAG128, is a logical channel number. The RAG128 has eight analog inputs. However, using sub-multiplexer boards, each of the inputs might be expanded to 16 separate channels, yielding up to 128 channels total. To determine the Point number for any given channel, place the A/D channel number in the upper nibble of a byte, and the MUX channel number in the lower nibble, as follows:

D7	D6	D5	D4	D3	D2	D1	D0
0	AD2	AD1	AD0	MA3	MA2	MA1	MA

This can be done algorithmically by multiplying the A/D channel by 16 and adding the mux channel.

In C: `pointnum = (adchan << 4) + muxchan;`

In Pascal: `pointnum = (adchan shl 4) or muxchan;`

Concepts: The Point List Buffer

Because the Pod has up to 128 channels of Analog data, and each channel can be acquired at a different gain, and even at a different polarity, some setup is necessary to improve usability. The setup consists of telling the Pod what input range and polarity each channel is going to be acquired at, and which channels are going to be acquired in what order, using the Point List (PL) series of commands. The Point List is a list of the points to be acquired, along with all of the configuration information necessary to select polarity, range and mux gain. All of the commands to set PL entries take a point number (or the keyphrase "ALL"), and a 2-byte (4 hex characters) value describing the configuration for that point (or the keyphrase "DEFAULT"). The format of the 2-byte entry follows:

Polarity				Gain/Range Code				Analog Channel				Mux Channel			
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	BIP/ UNI	5/10	GN2	GN1	GN0	0	AD2	AD1	AD0	MA3	MA2	MA1	MA0

BIP/UNI and 5/10: See following table:

BIP/UNI	5/10	Input Range (V)
0	0	0 to 5
0	1	0 to 10
1	0	±5
1	1	±10

GN2-GN0: The GN2-GN0 bits of Port 1 (MUX Control). Used to control gain control bits on AIM-16 and other submultiplexers. Refer to AIM-16 manual, fig. 3-1, for description of required control bits.

Note

ALL numbers passed to and from the Pod are in hexadecimal.

Command	Description	Returns
PLnn=xxxx	Set Point List Entry for point nn to xxxx. See Concepts: The Point List Buffer for more information on the point configuration setup required to use the PL commands.	[CR]
PLnn?	Reads the Point List Entry for point number nn.	xxxx[CR]
PLALL=xxxx-xxxx	Reserved for future development.	
PLALL?	Reads the Point Configuration for all points (0-127). Data is returned as XXXX XXXX XXXX . . . XXXX XXXX[CR] (A total of 128 entries).	See Description
PLnn=DEFAULT	Sets point nn to default configuration (±5V and GN2-GN0 all zero). The default point list consists of A/D channels 0-7 acquired at ±5V, and all further list entries set to A/D channel 0.	[CR]
PLALL=DEFAULT	Sets all point to default configuration (±5V and GN2-GN0 all zero, entries 0-7= A/D channel 0-7, all other = A/D channel 0).	[CR]
BACKUP=PL	Stores the point list into the onboard EEPROM for power-off safekeeping. The point list is restored automatically every time the Pod is reset, or the PLALL=BACKUP command is used.	[CR]
PLALL=BACKUP	Restores the EEPROM point list into the current point list. This command is executed automatically upon Pod reset.	[CR]
BACKUP=CAL: xxxx, xxxx	Writes scale and offset calibration constants for Y=MX+B calibration. Used by setup/calibration program	[CR]

CAL?	Returns stored calibration constants. See sample code on disk for usage.	xxxx,xxxx [CR]
PROGRAM=	Use with caution. Loads new firmware into the Pod EEPROM. Use ESC (ASCII 27) to abort.	n/a
?(vertical bar)	Same as PROGRAM=, except used only after a reset to gain access to pod.	[CR]
BAUD=nnn	Sets baud rate.	=:Baud:0x
POD=xx	Sets the Pod address to xx	=:Pod#xx
S=xxxx	Sets the sample rate of the A/D. Valid ranges are 00A2 through FFFF. The value passed is the desired divisor of the clock rate. 75H is 6666Hz, or 150? s/sample. 385H is 1000Hz, or 1 ms/sample.	[CR] 1/12 x Divisor +22 uSEC
S?	Returns the Clock Divisor from EEPROM on the Pod as above.	xxxx[CR]
!xx	Address Select command. Selects the Pod at address"xx" for communication.	[CR]
H*	"Hello" command. H followed by anything returns Pod name, revision, and copyright message.	See Descr.
N	Resend last message. Causes the Pod to re-transmit the last message. For example, should be used upon parity error.	See Descr.
V	Reads the version number form the firmware.	X.xx[CR]
Mxx	Sets bits as inputs or outputs. Each bit of xx controls a single bit of the DIO port. One's set the corresponding bit as output, zeroes set the port as input. Writing to a port marked as input will fail. Power-on default is all inputs .	[CR]
Mx+	Turns bit x to output.	[CR]
Mx-	Turns bit x to input, lower case	[CR]
O0xx	Writes xx to DIO port 0.	[CR]
O1xx	Writes to Port 1. Should be used with Nomux firmware variant.	[CR]
Ox+	Sets output bit x (0-F0) to One The DIO port bits are 0-7, MUX bits are 8-F.	[CR]
Ox-	Sets output bit x to ZERO.	[CR]
I	This command reads the eight digital inputs (from port 0).	Xx[CR]
In	Single bit read from bit n (0-7, the port 0 bits).	0[CR] or 1[CR]
CMxx[CR]	Writes byte value xx to the counter/timer control byte.	[CR]

CLn,xxxx[CR]	Loads hex value xxxx to counter number n divisor.	[CR]
CRn[CR]	Reads the value xxxx of the current count on counter 00, 01, or 02	Cnn=xxxx[CR]
ACnn ₁ -nn ₂ ,xxxx	Acquires point list entries nn ₁ through nn ₂ , with a total of xxxx conversions performed. xxxx must be equal to or less than 2710hex (10,000 decimal--all numbers must be presented to the Pod in hex). 10,000 data samples is the maximum internal storage capacity of the standard Pod. (Optional firmware can increase this capacity. Contact factory for further information.) Note that this function does not return any data. Data is stored in internal SRAM storage for later retrieval using the R command.	[CR]
R*	This command reads the last buffer of data acquired using the AC command. The data is returned in CCXXXXCCXXXX . . CCXXXXCCXXXX[CR] format, where CC is the point number that was acquired, and XXXX is the analog conversion data. The total number of return values is identical to the XXXX parameter passed to the most-recently issued AC command.	See Descr.
Axxxx	Acquires a single channel and returns the data immediately via the serial port, bypassing the point list. This command takes as its sole argument the same 2-byte parameter passed to the PL series of commands. The channel specified in the parameter is acquired using the range, polarity, and mux gain specified. Unlike the AC command, the data is returned immediately.	XXXX[CR]
Ann ₁ -nn ₂ ,xxxx	The A command works very similarly to the AC command. All parameters are the same. However, the A command runs in the foreground, occupying all of the attention of the Pod's microcontroller. This yields higher throughput but sacrifices flexibility. The Pod acquires data at its fastest rate.	Data is returned upon completion as if "R" had been issued.

Table 3-1: RAG128 Command List

Note

Pod reset occurs upon power-up, programming process, or watchdog time-out.

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.

Configure Point List

PLnn=xxxx	Sets point list entry for point number nn to xxxx.
PLnn= DEFAULT	Sets Point nn to default configuration
PLALL=DEFAULT	Sets all points to default configuration

Note: Default configuration is $\pm 5V$., GN2-GN0 all zero.

Read Point List

Plnn?	Reads the Point configuration for point number nn.
PLALL?	Reads the Point configuration for all points (0-127)

Note: Data is returned as xxxx xxxx xxxx ... xxxx xxxx[CR] with a total of 128 entries.

Write Point List to EEPROM

BACKUP=PL	Stores the point list configuration in the onboard EEPROM, for power-off safekeeping.
-----------	---

Note: The point list configuration is automatically restored every time the Pod is reset, or when PLALL=BACKUP command is issued.

Restore Point List

PLALL=BACKUP	Restores Point List configuration in EEPROM to the current point list.
--------------	--

Note: This command is executed automatically upon Pod reset.

Set Sample Rate

S=xxxx	Set Sample Rate of the A/D Converter.
--------	---------------------------------------

This function sets the sample rate of the A/D converter. Valid values range from 00A2 to FFFF. The value passed is the desired divisor of the rate clock (11.0592 MHz). The equation to use in calculating the divisor is:

$$\text{Divisor} = [(1/\text{Rate}) - 22? \text{ Sec}] * (\text{Clock}/12)$$

Example:

Program the RAG128 for 1K samples per second.

SEND:	S=0385
RECEIVE:	[CR]

Note: The sample rate configured is stored in EEPROM on the Pod, and will be used as the default (power-on) sample rate. The factory default sample rate (100Hz) can be restored by sending "S=0000" to the Pod.

Read Sample Rate

S? Returns the clock Divisor from EEPROM on the Pod.

Configure Bits as Input or Output

Mxx Configures digital bits as inputs or outputs.
Mx+ Configures digital bit 'x' as output.
Mx- Configures digital bit 'x' as input.

These commands program the digital bits, on a bit-by-bit basis, as input or output. A "zero" in any bit position of the xx control byte designates the corresponding bit to be configured 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: MAA
RECEIVE: [CR]

Program bits 0-3 as input, and bits 4-7 as output.

SEND: MF0
RECEIVE: [CR]

Read Digital Inputs

I Read 8 bits
In Read bit number n

These commands read the digital input bits from the Pod. All byte responses are sent most-significant nibble first.

Examples:

Read ALL 8 bits.

SEND: I
RECEIVE: FF[CR]

Read only bit 2.

SEND: I02
RECEIVE: 1[CR]

Write Digital Outputs

O0xx	Write to all 8 digital output bits. (Port 0)
Ox±	Set bit x hi or low
O1xx	Write to AIM-16 output port. (Port 1)

Warning

Use of the 'O1xx' command can disrupt data acquisition in process by changing MUX channel and/or gain during acquisition, if MUX firmware is installed. These commands write outputs to digital bits. Any attempt to write to a bit configured as an 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 bit asserts the pull-down for that bit. Writing a "zero"(-) de-asserts the pull-down. Therefore, if the factory default +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.

Examples:

Write a one to bit 7 (set output to zero volts, assert the pull-down).

```
SEND:      O7+
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:      O00
RECEIVE:   [CR]
```

Write zeros to every odd bit.

```
SEND:      OAA
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 RAG128 version number.

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

Note: The "H" command returns the version number along with other information. See "Hello Message" on next page.

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:   FF[CR]            ;or whatever the data was
```


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, RAG128 Rev rr Firmware Ver:x.xx ACCES W/MUX"

aa is the Pod address
rr is the hardware revision, such as "B1"
x.xx is the software revision, such as "1.00" (See the "V" command.)

Mux support as "W/MUX" or "NOMUX"

Example:

Read the greeting message.

SEND: Hello?

RECEIVE: =Pod 00, RAG128 Rev B1 Firmware Ver:1.00 ACCES NOMUX[CR]

Configure Baud Rate (When Shipped, the Baud Rate Is Set at 9600.)

BAUD=nnn Program the Pod with a new baud rate.

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

Code(n)/(nnn)	Baud Rate
0 (000)	1200
1 (111)	2400
2 (222)	4800
3 (333)	9600
4 (444)	14400
5 (555)	19200
6 (666)	28800
7 (777)	57600

Therefore, valid values for the command's "nnn" are 000, 111, 222, 333, 444, 555, 666, or 777. 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=nnn" 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]
```

Configure Pod Address

POD=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 RS485 specification allows only 32 drops on the line, so some addresses may be unused even if RS485 "Repeaters" are used.

The new Pod address is saved in EEPROM and will be used even after power-down until the next "Pod=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:      POD=01
RECEIVE:   =:Pod#01[CR]
```

Set the Pod address to F3.

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

Take the Pod out of addressed mode.

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

Address Select

!xx Selects the Pod addressed 'xx'

Note: When using more than one Pod in a system, each Pod is configured with a unique address. This command must be issued prior to any other commands to that particular Pod. This command needs to be issued only once prior to executing any other commands. Once the address select command has been issued, that Pod will respond to all other commands until a new address select command is issued.

Enter a New Program

PROGRAM= This command initiates transfer of a new program to the RAG128.

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 ACCESS to provide field-upgrades to the RAG128 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 CD, or is available separately for a small fee.

| (Vertical Bar)= Same as PROGRAM= except it is used only after a reset has occurred. Allows access to Pod to download new firmware. This command is embedded in BOOT code and will work even if the "Personality" code is corrupted.

Counter/timer Commands

Cmxx[CR]: Writes the byte value xx to the type 8254 counter/timer control byte.

Cln,xxxx[CR]: Loads the hex value xxxx to counter/timer n.

Crn[CR]: Reads the xxxx value in counter n (00, 01, or 02).

Special Acquisition Commands

ACnn1-nn2,xxxx: Acquires Point List entries nn1 through nn2, with a total of xxxx conversions performed.

xxxx must be equal to or less than 2710hex (which is 10000 decimal--all numbers must be presented to the Pod in hex). 10000 data samples is the maximum internal storage capacity of the standard Pod. (Optional firmware can increase this capacity. Contact the factory for further information.) Please note that this function does not return any data. Data is stored in the internal SRAM storage for later retrieval using the R command.

R: This command reads the last buffer of data acquired using the AC command.

The data is returned in CCXXXX CCXXXX ... CCXXXX CCXXXX[CR] format, where CC is the point number that was acquired, and XXXX is the analog conversion data. The total number of return values is identical to the XXXX parameter passed to the most recently issued AC command.

Axxxx: Acquires a single channel and returns the data immediately via the serial port, bypassing the point list.

This command takes as its sole argument the same 2-byte parameter passed to the PL series of commands. The channel specified in the parameter is acquired using the range, polarity, and mux gain specified. The data is returned immediately, unlike the AC command.

Ann-nn,xxxx The A command works very similarly to the AC command. All parameters are the same. However, the A command runs in the foreground, occupying all of the attention of the Pod's microcontroller. This yields higher throughput, but sacrifices flexibility. The Pod acquires data at its fastest possible rate.

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 07, in hex).
- 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].

Appendix A: Application Considerations

Introduction

Working with RS422 and RS485 devices is not much different from working with standard RS232 serial devices and these two standards overcome deficiencies in the RS232 standard. The RS422 and RS485 standard permits cable lengths up to 4000 feet and, because it operates in differential mode and is more immune to induced noise.

An important deficiency of the RS232 Standard is that more than two devices cannot share the same cable.

This is also true for RS422 but RS485 offers all the benefits of RS422 plus allows up to 32 devices to share the same twisted pairs. An exception to the foregoing is that multiple RS422 devices can share a single cable if only one will talk and the others will all receive.

Balanced Differential Signals

The reason that RS422 and RS485 devices can drive longer lines with more noise immunity than RS232 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 ± 2 to ± 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. RS485 drivers must have this control capability. RS422 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 ± 7 V 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.

Parameter	RS232C	RS422	RS485
Mode of Operation	Single Ended	Differential	Differential
Number of Drivers and Receivers	1 Driver 1 Receiver	1 Driver 10 Receivers	32 Drivers 32 Receivers
Maximum Cable Length (FT)	50	4000	4000
Maximum Data Rate (Kbits/sec)	20	10000	10000
Maximum Common Mode Voltage (V)	N/A	+/- 7	+12V to -7V
Driver Outputs (V)	25 V Maximum	+/- 6 V Maximum	+/- 5 V Maximum
Driver Output Open Circuit Vo Loaded	25 V Maximum	2 V or 50% of Vo	+/- 1.5 Min, +/- 5 V Max
Driver Load (Ohms)	3000-7000	100	60
Driver Output Short-Circuit Current (mA)	500 to VCC or Ground	150 to Ground	150 to Ground
Driver Output Resistance (Power On) in Tristate (Kohms)	N/A	N/A	120
Driver Output Resistance (Power Off) in Tristate (Kohms)	0.3	60	120
Receiver Input Resistance (Kohms)	3 to 7	4 Minimum	12 Minimum
Receiver Sensitivity (uV)	+/- 3000	200 Minimum	+/- 200 Minimum
Driver Output Rise Time		10% of Bit Interval Max	25 nSec

Table A-1: Specification Summary for the RS232c, RS422 and RS485 Communications Standards

To prevent signal reflections in the cable and to improve noise rejection in both the RS422 and RS485 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 RS422 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.)

All Remote ACCES pods use photo couplers to provide galvanic isolation between network devices (remote pods), and between the remote pods and the host controller. This prevents the harmful effects of possible differences in ground potentials between different parts of the network with each operating from its own local power source.

RS485 Data Transmission

The RS485 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 RS422 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.

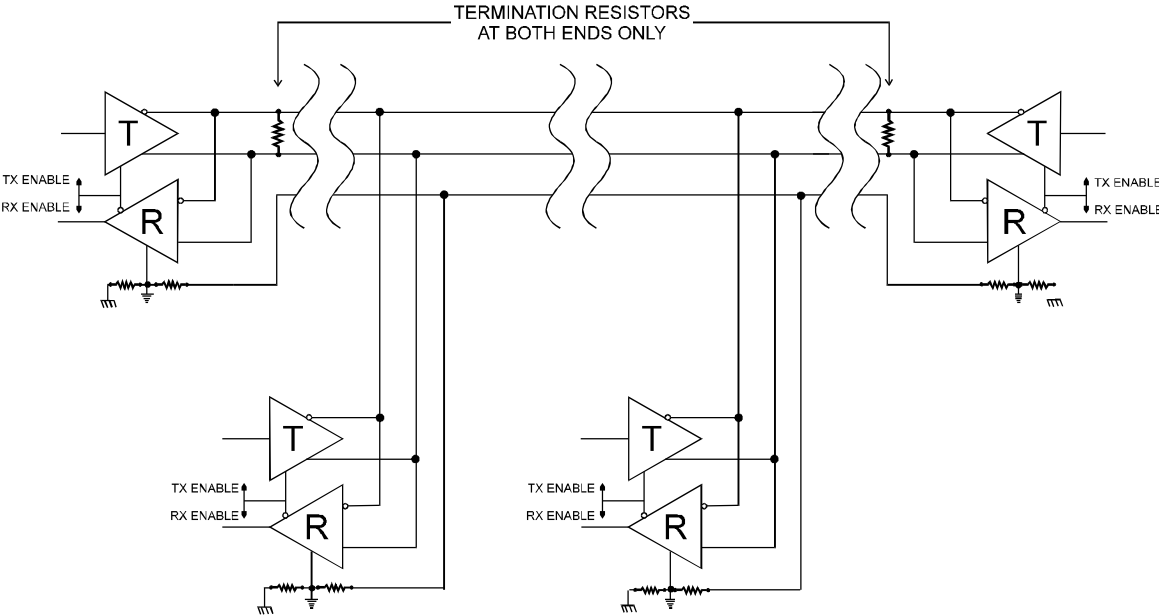
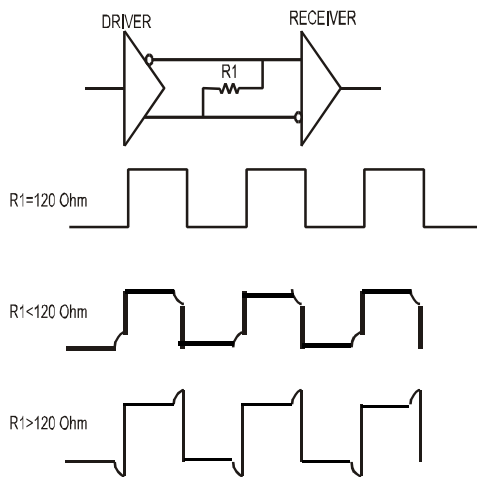


Figure A-1: Typical RS485 Two-Wire Multidrop Network

For longer cable runs, the proper termination of the cable is extremely important. If the cable is not terminated with its characteristic impedance distorted waveforms may result. A quick look at the output of the driver will tell how the cable is terminated. It is best to look at the driver output connected to the end of a cable when transmitting square wave data.



Rt: Termination Resistance
 Rch: Characteristic Impedance

This waveform represents proper termination being used.

In this waveform the signal is reflected back out-of-phase because $R_t < R_{ch}$ and subtracts from the signal

In this waveform the signal is reflected in phase because $R_t > R_{ch}$ and adds to the amplitude of the signal at the output driver.

The width of the distortion is equal to twice the electrical length of the cable (approximately 1.5 nSec/foot). The transmission line of choice for the RS485 application is a twisted pair with a characteristic impedance of 120 ohms. Many manufacturers offer a broad range of 120 ohm cables designated for RS485 applications.

Losses in a transmission line are a complex contribution of DC conductor loss, AC losses in the form of skin effect, leakage and AC losses in the dielectric. Polyethylene insulated cables (such as Belden 9481 or equivalent) have lower losses for high frequencies. For data rates < 100 Kbytes the PVC insulated cables are an economically and technically acceptable choice.

Customer Comments

If you experience any problems with this manual or just want to give us some feedback, please email us at: *manuals@accessioproducts.com*.. Please detail any errors you find and include your mailing address so that we can send you any manual updates.



10623 Roselle Street, San Diego CA 92121
Tel. (858)550-9559 FAX (858)550-7322
www.accessioproducts.com

