

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*



the smart approach to instrumentation ™

IOtech, Inc.
25971 Cannon Road Cleveland, OH 44146
Phone: (440) 439-4091 Fax: (440) 439-4093
E-mail (sales): sales@iotech.com Internet: www.iotech.com

Serial488/4 User's Manual

p/n Serial488/4-901, Rev.1.2

1996-1997 by IOtech, Inc. Part No. Serial488-901, Rev, 1.2
Printed in the United States of America

Warranty Information -Your IOtech warranty is as stated on the product warranty card. You may contact IOtech by phone,fax machine, or e-mail in regard to warranty-related issues**.**
Phone: (440) 439-4091, fax: (440) 439-4093, e-mail: sales@iotech.com

Limitation of Liability - IOtech, Inc. cannot be held liable for any damages resulting from the use or misuse of this product.

Copyright, Trademark, and Licensing Notice - All IOtech documentation, software, and hardware are copyright with all rights reserved. No part of this product may be copied, reproduced or transmitted by any mechanical, photographic, electronic, or other method without IOtech's prior written consent. IOtech product names are trademarked; other product names, as applicable, are trademarks of their respective holders. All supplied IOtech software (including miscellaneous support files, drivers, and sample programs) may only be used on one installation. You may make archival backup copies.

FCC Statement - IOtech devices emit radio frequency energy in levels compliant with Federal Communications Commission rules (Part 15) for Class A devices. If necessary, refer to the FCC booklet How To Identify and Resolve Radio-TV Interference Problems (stock # 004-000-00345-4) which is available from the U.S. Government Printing Office, Washington, D.C. 20402.

CE Notice - Many IOtech products carry the CE marker indicating they comply with the safety and emissions standards of the European Community. As applicable, we ship these products with a Declaration of Conformity stating which specifications and operating conditions apply.



Warnings, Cautions, Notes, and Tips - Refer all service to qualified personnel. This caution symbol warns of possible personal injury or equipment damage under noted conditions. Follow all safety standards of professional practice and the recommendations in this manual. Using this equipment in ways other than described in this manual can present serious safety hazards or cause equipment damage.



This warning symbol is used in this manual or on the equipment to warn of possible injury or death from electrical shock under noted conditions.



This ESD caution symbol urges proper handling of equipment or components sensitive to damage from electrostatic discharge. Proper handling guidelines include the use of grounded anti-static mats and wrist straps, ESD-protective bags and cartons, and related procedures.

Specifications and Calibration - Specifications are subject to change without notice. Significant changes will be addressed in an addendum or revision to the manual. As applicable, IOtech calibrates its hardware to published specifications. Periodic hardware calibration is not covered under the warranty and must be performed by qualified personnel as specified in this manual. Improper calibration procedures may void the warranty**.**



Quality Notice - IOtech has maintained ISO 9001 certification since 1996. Prior to shipment, we thoroughly test our products and review our documentation to assure the highest quality in all aspects. In a spirit of continuous improvement, IOtech welcomes your suggestions.

Table of Contents

Section1	**Introduction**	**Page**
1.1	General Description	1.1
1.2	Available Accessories	1.2
1.3	Specifications	1.3
1.4	Abbreviations	1.4
Section 2	**Getting Started**	**Page**
2.1	Inspection	2.1
2.2	Front and Rear Panels	2.1
2.3	IEEE 488 Bus Configuration	2.3
2.3.1	Serial 488/4 Addressing Modes	2.4
2.3.1.1	Secondary Addressing Mode	2.5
2.3.1.2	Dual Primary Addressing Mode	2.6
2.3.2	IEEE Address Selection	2.7
2.3.3	IEEE 488 Bus Terminator Selection	2.9
2.4	Memory Usage	2.9
2.5	Serial Port Configuration	2.10
2.5.1	RS-232/422 Configuration Jumpers	2.11
2.5.2	Asynchronous Communications Setup	2.13
2.5.3	Handshake Configuration	2.15
2.5.4	Serial Signal Descriptions	2.16
2.6	IEEE 488 Bus Implementation	2.20
2.6.1	My Talk Address (MTA), (MSA)	2.20
2.6.2	My Listen Address (MLA), (MSA)	2.21
2.6.3	Device Clear (DCL and SDC)	2.21
2.6.4	Interface Clear (IFC)	2.21
2.6.5	Serial Poll Enable (SPE)	2.21
2.6.6	Serial Poll Disable (SPD)	2.22
2.6.7	Serial Poll Response	2.22
2.6.8	Unlisten (UNL)	2.22
2.6.9	Untalk (UNT)	2.22
2.7	Hardware Installation	2.22
2.8	Powerup	2.24
2.9	Programming Examples	2.24
Section 3	**Command Descriptions**	**Page**
	Introduction	3.1
	Stop Bits (An)	3.2

Table of Contents

Section 3	Command Descriptions (cont'd)	Page
	Baud Rate (Bn)	3.3
	Parity (Cn)	3.4
	Data Bits (Dn)	3.5
	Error Query (E?)	3.6
	Flush (Fn)	3.7
	Handshake Type (Gn)	3.8
	Input Query (I?)	3.9
	Status EOI (Kn)	3.10
	Data EOI (Ln)	3.11
	Service Request (Mn)	3.12
	Handshake Control (Nn)	3.14
	Output Query (O?)	3.16
	Port Select (Pn)	3.17
	Break (Qn)	3.18
	Save Configuration (Sn)	3.19
	Serial Terminator (Tn)	3.21
	Status (Un)	3.22
	Version Query (V?)	3.27
	Execute (X)	3.28
	Bus Terminator (Yn)	3.29
	Memory Query (Z?)	3.30
	Serial Poll Status	3.31

Section 4	IEEE 488 Primer	Page
4.1	History	4.1
4.2	General Structure	4.1
4.3	Send It To My Address	4.4
4.4	Bus Management Lines	4.4
4.4.1	Attention (ATN)	4.4
4.4.2	Interface Clear (IFC)	4.5
4.4.3	Remote Enable (REN)	4.5
4.4.4	End Or Identify (EOI)	4.5
4.4.5	Service Request (SRQ)	4.5
4.5	Handshake Lines	4.6
4.5.1	Data Valid (DAV)	4.6
4.5.2	Not Ready For Data (NRFD)	4.6
4.5.3	Not Data Accepted (NDAC)	4.6

Table of Contents

Section 4	IEEE 488 Primer (cont'd)	Page
4.6	Data Lines	4.7
4.7	Multiline Commands	4.7
4.7.1	Go To Local (GTL)	4.7
4.7.2	Listen Address Group (LAG)	4.8
4.7.3	Unlisten (UNL)	4.8
4.7.4	Talk Address Group (TAG)	4.8
4.7.5	Untalk (UNT)	4.8
4.7.6	Local Lockout (LLO)	4.8
4.7.7	Device Clear (DCL)	4.8
4.7.8	Selected Device Clear (SDC)	4.9
4.7.9	Serial Poll Disable (SPD)	4.9
4.7.10	Serial Poll Enable (SPE)	4.9
4.7.11	Group Execute Trigger (GET)	4.9
4.7.12	Take Control (TCT)	4.9
4.7.13	Secondary Command Group (SCG)	4.9
4.7.14	Parallel Poll Configure (PPC)	4.10
4.7.15	Parallel Poll Unconfigure (PPU)	4.10
4.8	More On Service Requests	4.10
4.8.1	Serial Poll	4.11
4.8.2	Parallel Poll	4.11

Section 5	Service Information	Page
5.1	Factory Service	5.1
5.2	Theory of Operation	5.1
5.3	Serial488/4 Main Board Component Layout	5.2
5.4	Serial488/4 Parts List	5.3

Appendix A **Serial488/4 Command Summary**

Appendix B **IEEE Command and Address Messages**

Introduction

1.1 General Description

The Serial488/4 is a four port serial RS-232 or RS-422 interface to the IEEE 488 bus. Each port can be independently programmed for baud rate, stop bits, data bits, parity and handshaking. The Serial488/4 can inform the bus controller that it has received data through the use of the SRQ facility or by direct inquiry of the number of serial input bytes available.

The Serial488/4 uses a command address in which the serial port parameters may be specified and serial port or command status is returned. The serial data is then sent and received on a separate address or addresses, providing data transparency between the serial device and host controller.

The Serial488/4 serial port configuration may be saved in non-volatile RAM to be used as the default settings when next powered on, allowing a once only configuration of the device for many applications.

1.2 Available Accessories

Additional accessories that can be ordered for the Serial488/4 include:

CA-7-1	1.5 foot IEEE 488 Cable
CA-7-2	6 foot IEEE 488 Cable
CA-7-3	6 foot shielded IEEE 488 Cable
CA-7-4	6 foot reverse entry IEEE 488 Cable
CA-23	12 foot 9 pin to 25 pin RS-232 Cable
CN-20	Right angle IEEE 488 adapter, male and female
CN-22	IEEE 488 Multi-tap bus strip, four female connectors in parallel
CN-23	IEEE 488 panel mount feed-through connector, male and female
Serial488/4 -901	Additional Instruction Manual

CAUTION



Please read this manual carefully! If equipment is used in any manner not specified in this manual, the protection provided by the equipment may be impaired.

1.3 Specifications

Serial Interface

CAUTION



Use Ports 1 through 4 only for connection devices having signals at serial communication levels

Terminal Installation Category:
Standard: Not Applicable.

- CE: Category 1.

EIA RS-232C:

- AB, BA, BB, CA, CB

EIA RS-422:

- Balanced Voltage

Character Set:

- Asynchronous bit serial

Duplex:

- Full

Word Length:

- 7 or 8 data bits

Stop Bits:

- 1 or 2

Parity:

- None, odd, or even

Baud Rates:

- 110, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200, or 16x external clock. The sum of all four channel baud rates not to exceed 38,400 bits/second.

Terminator:
- any 8 bit value

Handshaking:
- Clear to Send (CTS) / Request to Send (RTS), XON/XOFF, or none

I/O buffers:
- 56,000 characters - dynamically allocated

Output Voltage:
- ± 5 volts minimum (RS-232C); 5 volts typical (RS-422A)

Input Voltage:
- ± 3.0 volts minimum: ± 15 volts maximum

Connector:
- 9-pin sub-D male.

IEEE 488

CAUTION



The IEEE 488 terminal must only be used to control a non-isolated IEEE system. The common mode voltage (cable shell to earth) must be zero.

Terminal Installation Category:
- Standard: Not Applicable.
- CE: Category 1.

Implementation:
- SH1, AH1, T6, TE8, L4, LE4, SR1, RL0, PP0, DC1, DT0, C0, E1

Terminators:
- Selectable CR, LF, LF-CR and CR-LF with EOI

Programmable:
- IEEE Terminators, Serial Terminator, EOI, SRQ Mask, Data Port, Baud Rate, Word Length, Parity, Stop Bits, Handshake Type, Handshake Control

Connector:
- Standard IEEE 488 connector with metric studs

General

WARNING



Service: This product contains no operator serviceable parts, or operator replaceable parts within the case. Fuse replacement and changing of selected line voltage must be performed by qualified service personnel. Never open the Serial488/4 case while the unit is connected to the AC line.

Terminal Installation Category:

- Standard: Not Applicable.
- CE: Category 2 for Line Voltage Input terminal. All other terminals are Category 1.

Dimensions:

- 425mm wide x 45mm high x 203mm deep (16.75" x 1.75" x 8")

Weight:

- 2 kg. (4.5 lbs)

Environment:

- Standard: Indoor use, 0° to 50°C; 0 to 70% R.H. to 35°C. Linearly derate 3% R.H./°C from 35° to 50°C.
- CE: Indoor use at altitudes below 2000 m, 0 to 40°C; 80% maximum RH up to 31°C decreasing linearly 4% RH/°C to 40°C

Controls:

- Power Switch, external dip switch for IEEE addressing mode and IEEE address, and serial port configuration jumpers (internal).

Indicators:

- LEDs for Talk, Listen, SRQ, Error, Send, and Receive for each Port, and Power

Power:

- Internally selectable 105-125V or 210-250V; 50-60 Hz, 20 VA Max.

CAUTION



Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock.

Fuse:

- 1/4A Slo Blo (for 105-125V power line) or 1/8A Slo Blo (for 210-250V power line)

CAUTION



Fuse Failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, the fuse designated for use with that line voltage must be used.

1.4 Abbreviations

The following IEEE 488 abbreviations are used throughout this manual.

addr n	IEEE bus address "n"	PPC	Parallel Poll Configure
ATN	Attention line	PPU	Parallel Poll Unconfigure
CA	Controller Active	SC	System Controller
CR	Carriage Return	SDC	Selected Device Clear
data	Data String	SPD	Serial Poll Disable
DCL	Device Clear	SPE	Serial Poll Enable
GET	Group Execute Trigger	SRQ	Service Request
GTL	Go To Local	TA	Talker Active
LA	Listener Active	TAD	Talker Address
LAG	Listen Address Group	TCT	Take Control
LF	Line Feed	term	Terminator
LLO	Local Lock Out	UNL	Unlisten
MLA	My Listen Address	UNT	Untalk
MTA	My Talk Address	*	Unasserted

Getting Started

2.1 Inspection

The Serial488/4 was carefully inspected, both mechanically and electrically, prior to shipment. When you receive the interface, carefully unpack all items from the shipping carton and check for any obvious signs of physical damage which may have occurred during shipment. Report any such damage to the shipping agent immediately. Retain all shipping materials in the event that shipment back to the factory becomes necessary.

Every Serial488/4 is shipped with the following....

- Serial488/4 Four Channel IEEE 488/Serial Converter
- Serial488/4-901 Instruction Manual
- Serial488/4-002 Accessory Kit which includes:
    - CA-1 Power Cable
    - FU-1-.25 Spare 1/4 amp Fuse
    - FE-1 Rubber Feet (4)
    - HA-31 Rack Ears (2)
    - HA-41-6 Rack Ear Screws (4)
    - CN-19-9 9 Pin Female Connector (4)
    - CN-17-9 Hood for CN-19-9 (4)

2.2 Front and Rear Panels

Pictured in Figure 2.1 are the front and rear panels of the Serial488/4. The front panel houses thirteen LED indicators which display the status of the Serial488/4. The function of each indicator is described below.

SEND (1-4) ON when the designated port is transmitting data to the serial interface, OFF when the designated port transmitter is idle.

RECEIVE (1-4) ON when the designated port is receiving data from the serial interface, OFF when the port 1 receiver is idle.

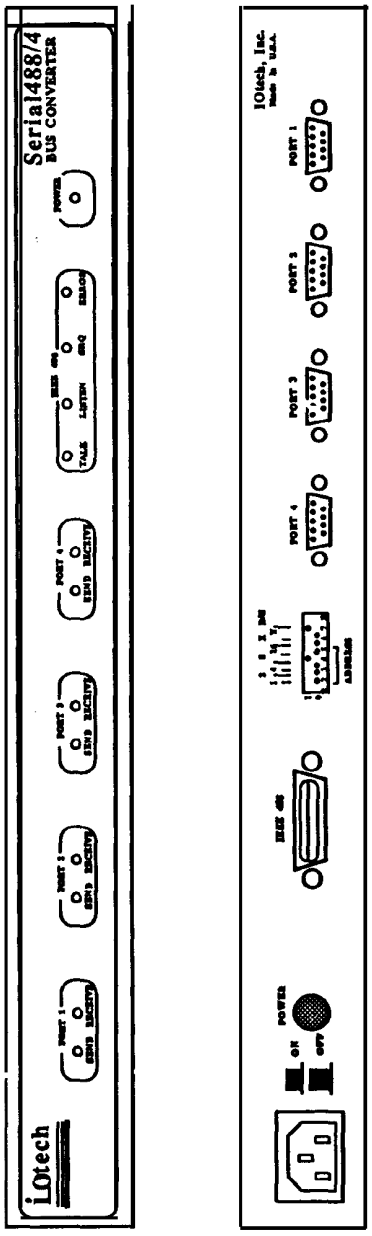


Figure 2.1 Serial488/4 Front and Rear Panels

TALK	ON when the Serial488/4 is in the Talker active state, OFF when the Serial488/4 is in the Idle or Listener active state
LISTEN	ON when the Serial488/4 is in the Listener active state, OFF when the Serial488/4 is in the Idle or Talker active state
SRQ	ON when the Serial488/4 has generated a service request, OFF when no SRQ is pending. (See Section 3 for more information.)
ERROR	ON when an error has occurred, OFF when no error condition exists. (See Section 3 for more information.)
POWER	ON when power is applied to the Serial488/4 and the power switch on the back panel is in the ON position (depressed). OFF when power is not present or the internal fuse or power supply is faulty.

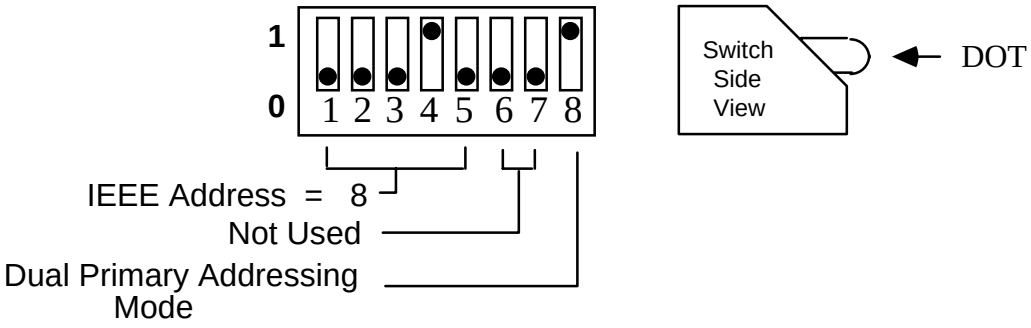
Mounted on the rear panel of the Serial488/4 are the power cable connector, the power on/off switch, the IEEE 488 bus connector, the addressing mode/address switch and four 9 pin Sub-D male connectors, one for each serial port.

2.3 IEEE 488 Bus Configuration

The Serial488/4 must be user-configured for proper operation over the IEEE 488 bus. Bus configuration includes selection of addressing mode, bus address, and bus terminator. Each of these configuration selections are described below.

The Serial488/4 bus addressing mode and bus address are determined by the eight position switch located on the rear panel (see Figure 2-1). Switch positions SW1-1 through SW1-5 determine the IEEE address (0 - 30); SW1-6 and SW1-7 are not used; SW1-8 determines the addressing mode (dual primary or secondary). SW1 is set at the factory for IEEE address 8 and dual primary addressing mode as shown in Figure 2-2. SW1 switch settings are read only when the Serial488/4 is powered on and should only be set before power is applied. If it is necessary to change the switch setting, first be sure the power cord is disconnected from AC line. Then, with a small flat blade screwdriver, slide each switch into the position desired. Refer to Sections 2.3.1 and 2.3.2 for a discussion of the various mode and address selections.

Figure 2-2. SW1 Factory Default Setting



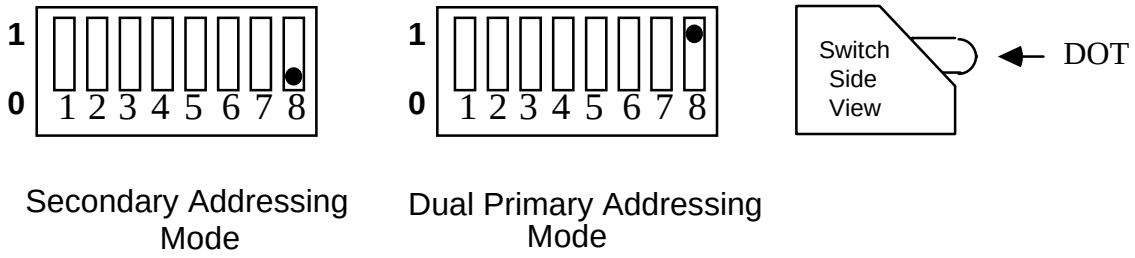
The bus terminator is determined by the last Bus Terminator (`Yn`) command sent to the bus as described in Section 3. The bus terminator can be set to any of four available options: CR only (`Y0`), LF only (`Y1`), CR LF (`Y2`), and LF CR (`Y3`). On initial power-up, the default value is (`Y2`) CR LF. To change the power-up default terminator, first set the terminator using the `Yn` command. Then save the selection using the Save Configuration (`S1`) command as described in Section 3. Refer to section 2.3.3 for a discussion of the bus terminator.

2.3.1 Serial488/4 Addressing Modes

To provide transparent serial data communications, the Serial488/4 occupies either 1 or 2 primary addresses on the IEEE 488 bus, dependent upon the addressing mode used. The two available addressing modes are secondary and dual primary. One address is used for sending commands to and receiving status from the Serial488/4. The other addresses are used exclusively for serial data transfer. The addressing mode is determined by position 8 of SW1 (located on the rear panel). Switch position SW1-8 for each mode is shown in Figure 2-4.

Examples are given at the end of this section to illustrate the differences between the two modes and how each is used to communicate with the Serial488/4.

Figure 2-3. SW1 View for Addressing Mode Selection

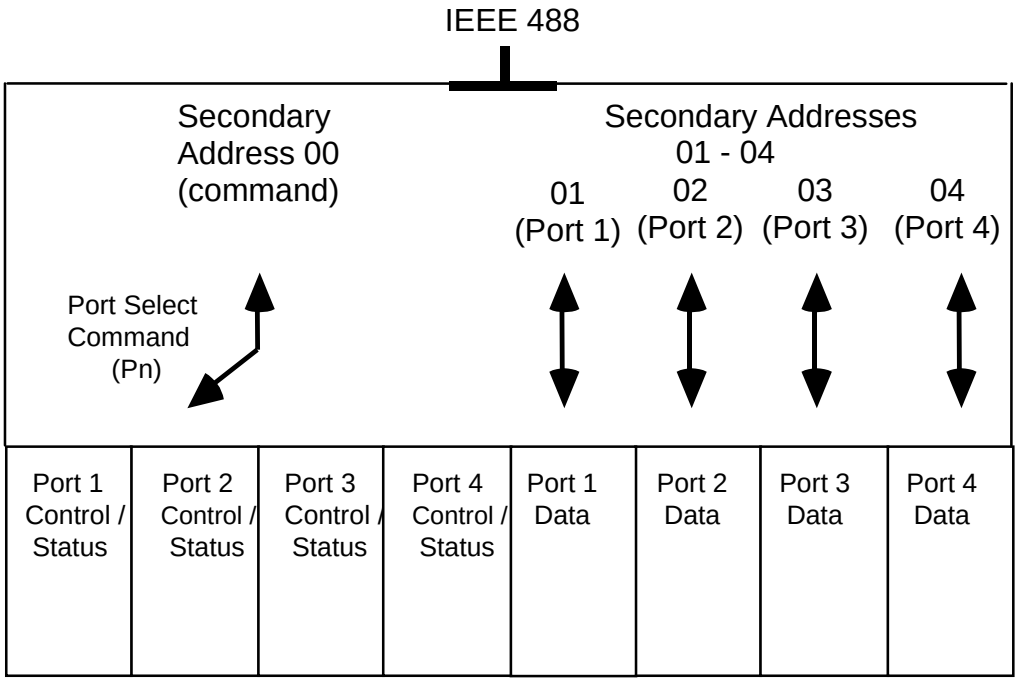


2.3.1.1 Secondary Addressing Mode

When Secondary addressing mode is selected, communications with the Serial488/4 is accomplished by using the primary bus address selected by SW1-1 through SW1-5 and an additional secondary bus address. Secondary address 00 is used in transferring command and status information while secondary addresses 01 through 04 are used for serial data transfers with serial ports 1 through 4 respectively. In this mode, control of the serial ports are multiplexed through secondary address 00, the command/status address, with the Port (`Pn`) command selecting the desired port. Data transfers, however, are not multiplexed as each serial data port has its own unique secondary address.

Figure 2-4 illustrates the relationship between the command/status and serial data addresses when using the secondary addressing mode.

Figure 2-4. Pictorial View of the Secondary Addressing Mode



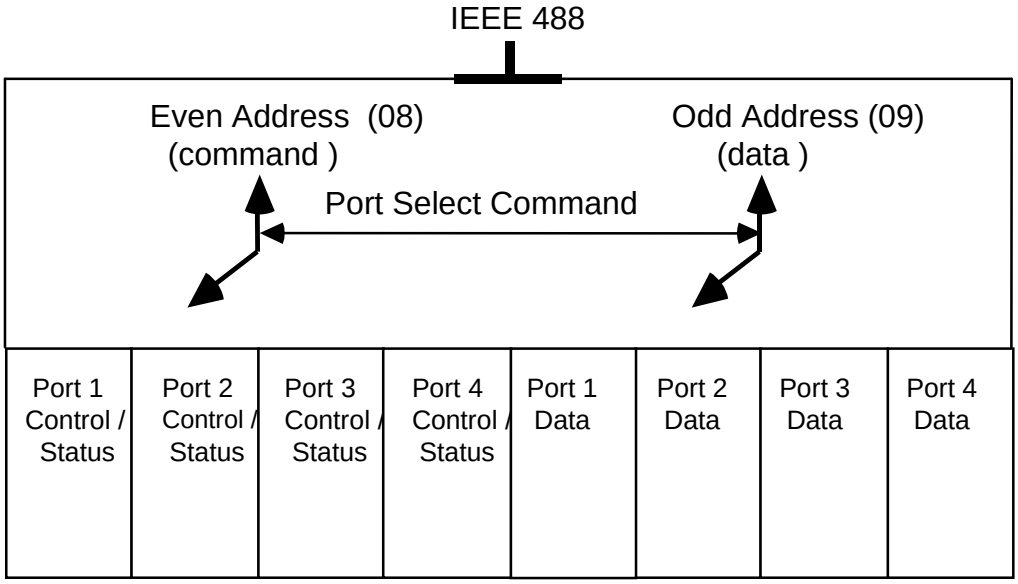
2.3.1.2 Dual Primary Addressing Mode

When Dual Primary addressing mode is selected, communications with the Serial488/4 is accomplished by using two consecutive primary IEEE bus addresses. These addresses are selected by SW-2 through SW1-5, SW1-1 (the LSB) of the address is ignored. Of the two addresses specified, the even address is used in transferring command and status information and the odd address is used for serial data transfers. In this mode, control of the serial ports is multiplexed through the even or command/status address, with the Port (`Pn`) command selecting the desired port. Serial data transfers are multiplexed through the odd or data address, again with the Port (`Pn`) command selecting the desired port.

Figure 2-5 illustrates the relationship between the

command/status and serial data addresses when using the dual primary addressing mode.

Figure 2-5 Pictorial View of the Dual Primary Addressing Mode



2.3.2 IEEE 488 Address Selection

The IEEE 488 bus address is set by SW1-1 through SW1-5 (located on rear panel). The address is read only at power on and can be set from 0 through 30 when using the secondary addressing mode, or 0 to 28 using the dual primary addressing mode. The address is selected by simple binary weighting with SW1-1 being the least significant bit and SW1-5 the most significant bit (see Figure 2-6). SW1 switch positions for IEEE addresses 0 through 30 are listed in Table 2-1. The factory default is address 8.

NOTE: If address 30 / 31 is selected, when using dual primary addressing mode, it defaults to address 28 / 29 because the IEEE 488 standard has reserved address 31. Similarly if address 31 is selected, when using the secondary addressing mode, it defaults to address 30.

Figure 2-6. SW1 View for IEEE Bus Address Selection

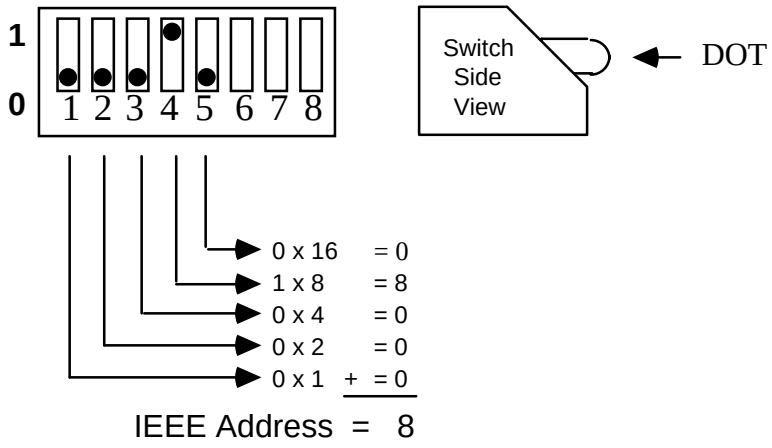


Table 2-1. SW1 Switch Positions (IEEE Address Selection)

IEEE Bus Address	SW1 setting 1 2 3 4 5	IEEE Bus Address	SW1 setting 1 2 3 4 5
0	0 0 0 0 0	16	0 0 0 0 1
1	1 0 0 0 0	17	1 0 0 0 1
2	0 1 0 0 0	18	0 1 0 0 1
3	1 1 0 0 0	19	1 1 0 0 1
4	0 0 1 0 0	20	0 0 1 0 1
5	1 0 1 0 0	21	1 0 1 0 1
6	0 1 1 0 0	22	0 1 1 0 1
7	1 1 1 0 0	23	1 1 1 0 1
8	0 0 0 1 0	24	0 0 0 1 1
9	1 0 0 1 0	25	1 0 0 1 1
10	0 1 0 1 0	26	0 1 0 1 1
11	1 1 0 1 0	27	1 1 0 1 1
12	0 0 1 1 0	28	0 0 1 1 1
13	1 0 1 1 0	29	1 0 1 1 1
14	0 1 1 1 0	30	0 1 1 1 1
15	1 1 1 1 0	31	1 1 1 1 1

2.3.3 IEEE 488 Bus Terminator Selection

The terminating characters sent when status is output by the Serial488/4 are set by using the Terminator (Yn) command. As shipped from the factory, the unit's power on IEEE 488 bus terminators are CR LF. The power on default terminators can be changed to any of the four available options discussed in detail in the Command Description section. The four available terminator command options are : CR only (Y0), LF only (Y1), CR LF (Y2), and LF CR (Y3). If it is desired to have particular power on default terminators, the terminators are set using the Terminator Command and then saved using the Save Configuration command (S1).

The Serial488/4 ignores all terminators received from the host controller on the command address. Only the Execute command (X) is used to signal the Serial488/4 that a command string has been completed.

The Serial488/4 ignores all terminators sent to the serial data address(es), but will monitor the serial data it receives for the character specified with the Serial Terminator command (Tn). If enabled by the Data EOI command (Ln), EOI will be asserted when the serial terminator character is sent to the host controller. Each serial port may use a separate serial terminator.

2.4 Memory Usage

Memory in the Serial488/4 is dynamically allocated for the serial input and output buffers. This allows for the most efficient utilization of memory for any given application. This memory is kept in the USER 'heap' (a vernacular for heap of memory) until required by the system.

At power on each serial port is allocated two 127 byte mini-buffers or queues, one for input and the other for output. When the serial port requires more buffer space, additional queues are allocated. When a queue is emptied, it is released from the serial port so that it may be reallocated when, and where required.

There are approximately 430 queues available for a total of over 54,000 bytes of

buffer (character) space. Of the available queues, 398 are issued without regard to controlling the receipt of additional serial input data (handshaking).

When one of the last 32 queues is allocated the Serial488/4 signals the serial devices that they should stop sending serial data. This is accomplished by either un-asserting RTS or sending the 'XOFF' control character depending upon how the individual serial ports are programmed. Also the Serial488/4 sets a bit in its serial poll status byte and optionally issues an SRQ to inform the host bus controller of the 'low memory' condition. When more than 32 queues become available the serial devices are signaled to allow more serial input data by asserting RTS or issuing the 'XON' control character as appropriate. The 'low memory' bit of the serial poll status byte is also cleared by more than 32 queues becoming available, however if SRQs were enabled the bit will remain set until at least one serial poll request is received after the original 'low memory' warning.

When on of the last 16 queues is allocated the Serial488/4 will stop accepting serial output data from the host bus controller. In this situation the host bus controller should read any available serial input data or wait for the existing serial output data to be sent to the serial devices before continuing.

2.5 Serial Port Configuration

The Serial488/4 has four separate serial ports, each of which can be configured independently of the others. Complete serial port configuration is accomplished by using both hardware jumpers and software commands. Hardware jumpers, located on the main board behind each serial connector, are used to to configure the port electrically as either RS-232 or RS-422. Software configuration consists of various commands for setting baud rate, stop bits, parity, and other asynchronous communications characteristics.

2.5.1 RS-232/422 Configuration Jumpers

Each serial port is capable of operating with either RS-232 or RS-422 electrical characteristics. This selection is done via a set of hardware jumpers located on the main board behind each serial connector. A twelve position jumper plug must be inserted in one of the two available positions for proper operation. These jumpers are factory set to configure all ports for RS-232 levels as illustrated in figure 2.7.

If configured for RS-422, each port's Receive Data (RxD) and Clear to Send (CTS) input lines may optionally be terminated with a 100 ohm resister. Termination resisters are selected by positioning the two flea clips located behind each serial connector towards the front panel, or upper position. These jumpers are factory set to the unterminated position, towards the rear panel. The jumpers associated with each port are: port 1 JP104 &JP105, port 2 JP109 & JP110, port 3 JP114 & JP115, and port 4 JP119 & JP120.

Figure 2.8 illustrates port 1 jumpered in RS-232, RS-422 unterminated, and RS-422 terminated with 100 ohm resister (left to right).
NOTE: When using RS-422 in a single ended configuration ports must be unterminated for proper operation.
NOTE: Termination jumpers have no effect when ports are configured for RS-232 operation.

WARNING

Never open the Serial488/4 case while it is connected to the AC line. Internal voltage potentials exist which could cause personal injury or death.

To open the Serial488/4 place it on a flat surface then remove the four screws on top of the case and remove the top cover. Located in front of the four serial port connectors, are four sets of connectors with configuration jumpers, one set per serial port.

To reconfigure a serial port for RS-422 operation, remove the twelve position configuration jumper and reinsert it so that the center socket is now connected to the left socket as viewed from the rear of the unit. If single ended operation is desired move the two flea clips to the position farthest from the serial port connectors.

Once the jumper(s) have been repositioned for your particular application, make note of the new jumper settings for later reference and carefully reassemble the unit.

Figure 2-7. View of Configuration Jumpers - Factory Setting
All Ports configured as RS-232

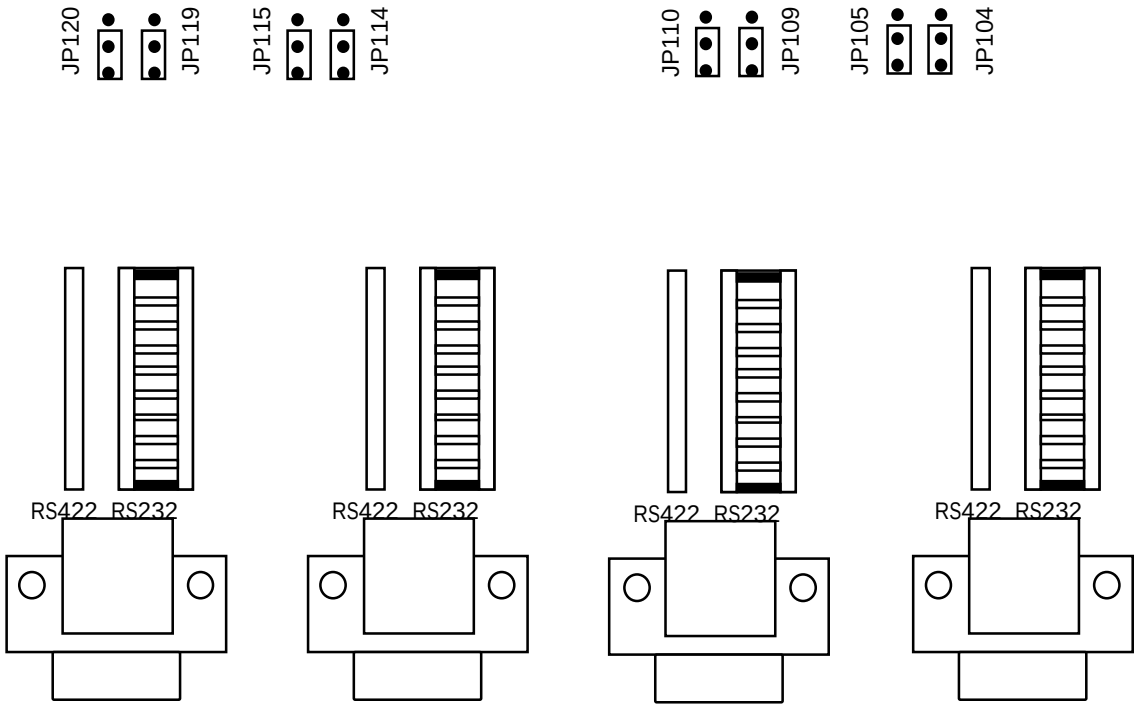
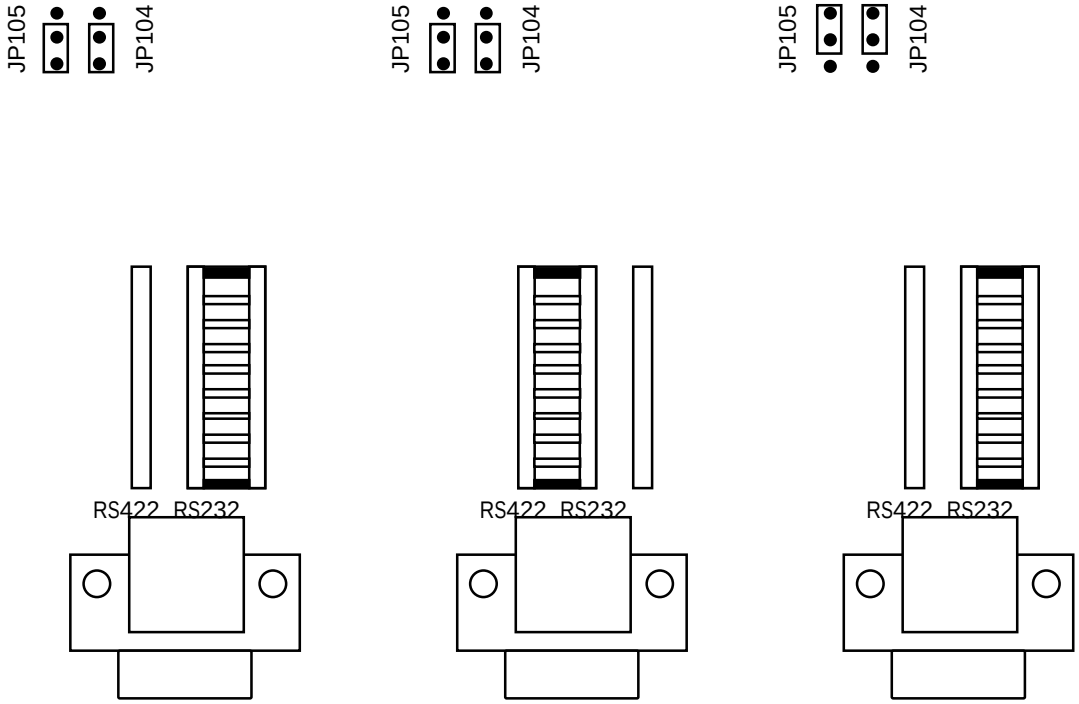


Figure 2-8. View of Alternate Port 1 Configurations
RS-232, RS-422 Unterminated, & RS-422 Terminated



2.5.2 Asynchronous Communications Setup

Associated with each serial port are a number of parameters which control the the transfer of serial data. These parameters, and the commands used to set them, are listed in Table 2-4.

Table 2-4. Serial Port Configuration Parameters

Stop Bits	`An`	Number of stop bits; 1 or 2.
Baud Rate	`Bn`	Baud rate; 110 - 19200, or external.
Parity	`Cn`	Parity; even, odd, or none.
Data Bits	`Dn`	Number of data bits; 7 or 8.
Handshake Type	`Gn`	Handshake type; RTS/CTS, XON/XOFF, or none
Handshake Control	`Nn`	Handshake control; automatic, hold-off, or accept.

Configuration of the serial ports may be accomplished at any time by sending one or more command strings to the Serial488/4 command/status address. Configuration commands may be sent individually or grouped together to fully setup a port using only one message. Furthermore all four serial ports may be configured using only one message so long as each port setup begins with a Port (`Pn`) command and ends with an Execute (`X`) command. Several examples are presented below to illustrate different configuration methods.

Example 1: Configure all ports using dual primary addressing mode.

PRINT#1,"CLEAR08"	reset the Serial488/4
`PRINT#1,"OUTPUT08;P1X"`	Select port 1
`PRINT#1,"OUTPUT08;A0B3C1D0X"`	Specify 1 stop bit, 1200 baud, odd parity, and 7 data bits
`PRINT#1,"OUTPUT08;P2X"`	Select port 2
`PRINT#1,"OUTPUT08;A1B9C0D1X"`	Specify 2 stop bits, 9600 baud, no parity, and 8 data bits
`PRINT#1,"OUTPUT08;P3X"`	Select port 3
`PRINT#1,"OUTPUT08;A0B7C2D1X"`	Specify 1 stop bit, 4800 baud, even parity, and 8 data bits
`PRINT#1,"OUTPUT08;P4X"`	Select port 4
`PRINT#1,"OUTPUT08;A0B10C1D1X"`	Specify 1 stop bit 19200 baud, no parity, and 8 data bits

Example 2: Configure all ports using secondary addressing mode.

`PRINT#1,"CLEAR0800"`	Reset the Serial488/4
`PRINT#1,"OUTPUT0800;P1A0B3C1D0X"`	Select port 1 and specify 1 stop bit, 1200 baud, odd parity, and 7 data bits
`PRINT#1,"OUTPUT0800;P2A1B9C0D1X"`	Select port 2 and specify 2 stop bits, 9600 baud, no parity, and 8 data bits
`PRINT#1,"OUTPUT0800;P3A0B7C2D1X"`	Select port 3 and specify 1 stop bit, 4800 baud, even parity, and 8 data bits

`PRINT#1,"OUTPUT0800;P4A1B10C1D1X"` Select port 4 and specify 2 stop bits 19200 baud, no parity, and 8 data bits

2.5.3 Handshake Configuration

The Serial488/4 uses handshaking to prevent loss of data while both transmitting and receiving serial data. The function of handshaking is to prevent a sender from transmitting more data than a receiver can accept. The Serial488/4 accomplishes this in one of two ways, by using hardware or software handshaking.

Hardware handshaking utilizes the Request to Send (`RTS`) line of the serial interface to inform the connected device that the Serial488/4 is able to accept data. Inversely the Clear to Send (`CTS`) line of the serial interface is monitored by the Serial488/4 to determine whether or not the connected device is able to accept data. The Serial488/4 may be configured to use the hardware (`RTS/CTS`) handshaking method with the `G0` command. In addition the automatic handshaking control option (`N0`) is normally used. This option will negate and assert the `RTS` line depending on the amount of memory available for receiving serial data. The manual handshake control options, `N1` and `N2`, may be used in special circumstances to un-assert and assert the `RTS` line under program control if desired. The clock output handshake control option (`N3`) is not allowed when hardware handshaking is being used and will generate a command conflict error if attempted.

The software handshaking method uses special control characters, transmitted along with the serial data, to prevent loss of serial data. The `XOFF` character (`&H11`) is transmitted to inform the connected device that the Serial488/4 is no longer able to accept data. The `XON` character (`&H13`) is transmitted to inform the connected device that the Serial488/4 is again able to accept data. Similarly the Serial488/4 monitors the serial data it receives for the occurrence of `XON/XOFF` control characters and stops or starts the transmission of data to the connected device accordingly. The Serial488/4 may be configured for software (`XON/XOFF`) handshaking by using the `G1` command. In addition the automatic handshaking control

option (N0) is normally used. This option will cause the Serial488/4 to send XOFF and XON control characters depending on the amount of memory available for receiving serial data. The manual handshake control options (N1 and N2) may be used in special circumstances to send XOFF and XON characters under program control if desired.

The clock output handshake control option (N3) operates the same as the automatic handshake option, but will also cause the serial baud clock x 16 signal to be output on the RTS line.

In addition to software and hardware handshaking the Serial488/4 provides a No Handshaking mode which may be used in special circumstances where handshaking is not desired. This mode is selected by using the G2 command. In the no handshake mode the RTS line may be manually un-asserted or asserted using the N1 and N0 or N2 handshake control commands, or the serial baud clock x 16 may be output on the RTS line if the N3 command is used.

2.5.4 Serial Signal Descriptions

The Serial488/4 is equipped with four DB-9S connectors on its rear panel and requires DB-9P mating connectors. The Serial488/4's connector is configured as an IBM AT when RS-232 levels are selected and Macintosh II/SE/Plus when RS-422 levels are selected. Tables 2.2 and 2.3 list the RS-232 and RS-422 pin connections. Each of these signals are described below.

RxD- Receive Data (input)
This pin accepts serial data sent by an RS-232 or RS-422 device. The serial data received is expected to match the word length, baud rate, stop bits, and parity configuration of the particular port. This signal is low true.

RxD+ Receive Data (input)
This pin accepts serial data sent by an RS-422 device. It functions identically to RxD- except that its polarity is inverted. This signal is high true.

TxD- Transmit Data (output)
This pin transmits serial data to an RS-232 or RS-422 device. The serial data received is sent with the word length, baud rate, stop bits, and parity configured for the particular port. This signal is low true.

TxD+ Transmit Data (output)
This pin transmits serial data to an RS-422 device. The pin functions identically to TxD- except that its polarity is inverted. This signal is high true.

CTS+ Clear To Send (input) / 16 x Baud Clock (input)
The CTS+ input is used as a hardware handshake line to prevent the Serial488/4 from transmitting serial data to an RS-232 or RS-422 device when it is not able to accept it. When RTS/CTS handshaking is selected using the (G0) command, the Serial488/4 will not transmit data out TxD+ while this signal is un-asserted (low). If XON/XOFF (G1) or no handshaking (G2) is selected, the CTS+ line is ignored when transmitting data.

If XON/XOFF (G1) or no handshaking (G2) is selected the CTS line may be used to input a clock signal which is sixteen times the desired baud rate. This mode is selected using the Baud Rate command (B11).

CTS- Clear To Send (input) / 16 x Baud Clock (input)
The CTS- input is used as a hardware handshake line with RS-422 devices. The pin functions identically to CTS+ except that its polarity is inverted. This signal is low true.

RTS+ Request To Send (output) / 16 x Baud Clock (output)
The RTS+ output is used as a hardware handshake line to prevent an RS-232 device from transmitting serial data to the Serial488/4 when it is not able to accept it. When automatic RTS/CTS handshaking is selected using the (G0) and (N0) commands, the Serial488/4 will assert (high) the RTS+ signal when greater than

4096 memory locations are available in its internal buffers. If available memory drops below 4096 bytes, the Serial488/4 un-asserts (low) the RTS+ signal. With RTS/CTS handshaking selected, the RTS+ signal may also asserted and un-asserted under program control by using the (N2) and (N1) handshake control commands respectively.

If XON/XOFF handshaking (G1) is selected, the RTS+ line may either be programmed as either permanently asserted (high) or to output sixteen times the baud rate clock. These options are selected using the handshake control command (Nn).

If No handshaking (G2) is selected, the RTS+ line may either be programmed as either permanently asserted (high), permanently un-asserted (low), or to output a signal sixteen times the baud rate clock. These options are selected using the handshake control command (Nn).

RTS Request To Send (output) / 16 x Baud Clock (output)

The RTS- output is used as a hardware handshake line with RS-422 devices The pin functions identically to RTS+ except that its polarity is inverted. This signal is low true.

GND Ground

This signal sets the ground reference point for the other RS-232 input and output signals.

Table 2-2. RS-232 Serial Port Pin Connections

Pin	Description	Input/Output
1	No connection	
2	Receive data (RxD-)	Input
3	Transmit data (TxD-)	Output
4	No connection	
5	Ground	
6	No connection	
7	Request to send (RTS+)	Output
	or 16 x baud clock	Output
8	Clear to send (CTS+)	Input
	or 16 x baud clock	Input
9	No connection	

Table 2-3. RS-422 Serial Port Pin Connections

Pin	Description	Input/Output
1	Ground	
2	Clear to send positive (CTS +)	Input
	or 16 x baud clock	Input
3	Clear to send negative (CTS -)	Input
	or 16 x baud clock	Input
4	Transmit data positive (TxD +)	Output
5	Transmit data negative (TxD -)	Output
6	Request to send positive (RTS +)	Output
	or 16 x baud clock	Output
7	Request to send negative (RTS -)	Output
	or 16 x baud clock	Output
8	Receive data positive (RxD +)	Input
9	Receive data negative (RxD -)	Input

2.6 IEEE 488 Bus Implementation

The Serial488/4 implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following sections. The bus uniline and multiline commands that the Serial488/4 does not support or respond to include:

Remote Enable (REN)
Go to Local (GTL)
Local Lockout (LLO)
Take Control (TCT)
Group Execute Trigger (GET)
Parallel Poll (PP)
Parallel Poll Configure (PPC)
Parallel Poll Unconfigure (PPU)
Parallel Poll Disable (PPD)

2.6.1 My Talk Address (MTA), My Secondary Address (MSA)

The Serial 488/4 is addressed to talk in one of two ways dependent upon the addressing mode selected. When dual primary addressing mode is selected, the Serial488/4 is addressed to talk upon receipt of MTA. If secondary addressing mode is selected the Serial488/4 is addressed to talk upon receipt of an MTA and MSA sequence.

When the Serial488/4 is addressed to talk on its command/status address, it outputs either a `U0`, `Un`, or query (`?`) response message depending on which was last requested. The status output bus terminators are then appended to the output and EOI is asserted if the status EOI option was enabled. The Serial488/4 must be readdressed to talk to perform subsequent reads.

When the Serial488/4 is addressed to talk on a serial data address, it retrieves data from the associated serial input buffer and outputs it to the IEEE bus. If the serial terminator character is encountered and the data EOI option is enabled (`L0`), EOI is asserted on that character. EOI may also be asserted on the last character in the serial data buffer if this data EOI option is enabled (`L2`).

2.6.2 My Listen Address (MLA), My Secondary Address (MSA)

The Serial 488/4 is addressed to listen in one of two ways dependent upon the addressing mode selected. When dual primary addressing mode is selected, the Serial488/4 is addressed to listen upon receipt of MLA. If secondary addressing mode is selected the Serial488/4 is addressed to listen upon receipt of an MLA and MSA sequence.

When the Serial488/4 is addressed to listen on its command/status address it accepts characters from the active talker and interprets these characters as commands and command parameters. These commands are fully explained in Section 3.

When the Serial488/4 is addressed to listen on a serial data address it accepts data from the active talker and places this data into the appropriate serial output buffer, to be transmitted verbatim.

2.6.3 Device Clear (DCL and SDC)

Device Clear resets the Serial488/4 to power on defaults and empties all the serial input and output buffers. Any pending data and Service Requests (SRQ), including the information they convey are lost.

2.6.4 Interface Clear (IFC)

IFC places the Serial488/4 in the Talker/Listener Idle State and clears any pending request from service (SRQ). The condition which caused the SRQ, however, remains unchanged.

2.6.5 Serial Poll Enable (SPE)

When Serial Poll Enabled, the Serial488/4 sets itself to respond to a serial poll with its serial poll status byte if addressed to talk. When the serial poll byte is accepted by the controller, any pending SRQs are cleared.

The Serial488/4 will continue to try to output its serial poll response until it is serial poll disabled by the controller.

2.6.6 Serial Poll Disable (SPD)

Disables the Serial488/4 from responding to serial polls by the controller.

2.6.7 Serial Poll Response

Whenever the Serial488/4 generates an SRQ, a serial poll of either the command/status address or a serial data address will return a serial poll status byte of at least 64 decimal showing that an SRQ has been generated by the Serial488/4. The status byte returned can be used to determine the cause of the Service Request. These responses are explained in detail in Section 3.

2.6.8 Unlisten (UNL)

UNL places the Serial488/4 in the Listener Idle State.

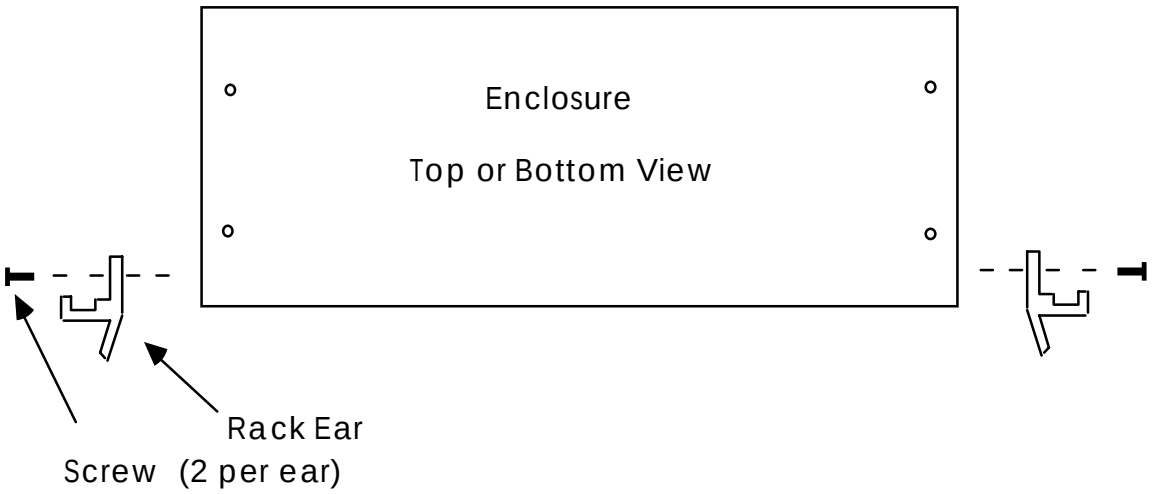
2.6.9 Untalk (UNT)

UNT places the Serial488/4 in the Talker Idle State.

2.7 Hardware Installation

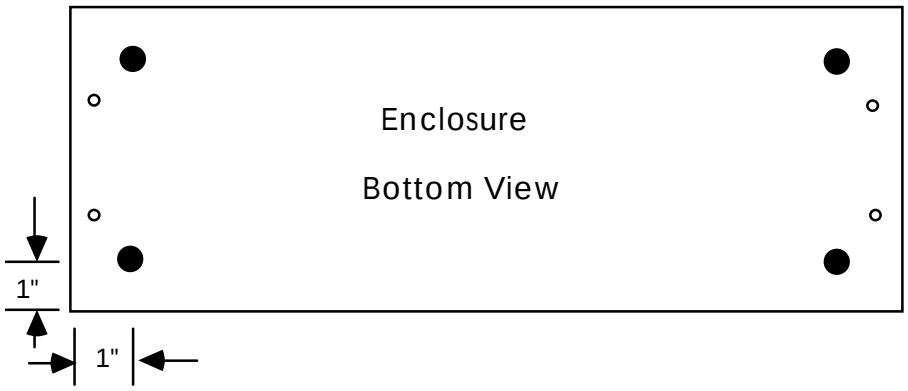
Included with the Serial488/4 are accessories for rack or bench use. If rack mount installation is required, install the two rack ears using the enclosed screws. as shown in Figure 2-6. These ears can be installed so either the front or the rear of the unit faces the front of the rack fixture.

Figure 2-6. View of Enclosure showing Rack Hardware Installation



If bench installation is required, the rubber feet can be installed on the underside of the unit approximately one inch from each corner as shown in Figure 2-7.

Figure 2-7. View of Enclosure Showing Feet Placement



2.8 Powerup

To begin operating the Serial488/4, plug the power cord into an AC outlet.

WARNING
The interface is intended for <u>INDOOR USE ONLY.</u>
Failure to observe this warning could result in equipment failure, personal injury or death.

After plugging in the power cord, turn on the Serial488/4 by depressing the rear panel power switch (Figure 2-1). All the front panel LEDs should light for approximately two seconds while the Serial488/4 performs an internal ROM and RAM self check. At the end of this self check all indicators should turn off except POWER.

If you obtain the above response then your Serial488/4 is alive and well. If all LEDs remain on, then a ROM error has occurred. If all LEDs (except the power LED) flash continuously, then a RAM error has occurred. Try cycling the power to the Serial488/4 to determine that the error is repeatable.

If the LEDs do not flash and the POWER indicator does not remain lit, there may not be any power supplied to the interface. In this event, check to make sure the AC power is supplied to the unit. If the problem persists, refer to the Service Information section of this manual.

2.9 Programming Examples

These examples use the IOtech Personal488 and the BASIC Keyboard Controller program listed in appendix D. The Keyboard Controller program allows direct communication with the Serial488/4 (or any IEEE device on the bus) and is the most convenient method of exercising the Serial488/4 and becoming familiar with the commands and their actions.

First, set SW1 for Dual Primary addressing mode and bus address 8. Connect an IEEE cable from the Serial488/4 to the IEEE connector on the Personal488. Turn on

the Serial488/4 by depressing the rear panel power switch. All of the front panel LEDs should light for approximately one second while the Serial488/4 performs an internal ROM and RAM self test. At the end of this self test all indicators should turn off except Power.

Run the Keyboard Controller program and verify that the CMD> prompt appears on the screen. The following is a listing of commands given to the Serial488/4 and the response received. Responses are separated by extra vertical space for easier reading although these spaces will not actually appear when using the program.

```
CMD>                        Command prompt from Keyboard Controller program
CMD>OUTPUT08;S0X            Restore configuration to factory default values
CMD>CLEAR                   Reset the Serial488/4
CMD>ENTER08
1.3C0E0F0G0I000K0M000P0R0Y0Z55880

                            Status message shows all default options are in use
CMD>OUTPUT08;P1U1X          select port 1 and port 1 status
CMD>ENTER08
1.3A0B009C0D1G0I00000L1N0O00000T010U1

                            Status message shows all default options are in use
CMD>OUTPUT08;a1b7d0x        Change the configuration of port 1
CMD>ENTER08
1.3A1B007C0D0G0I00000L1N0O00000T010U1

                            Status message shows new options are in use
CMD>OUTPUT08;P3U3X          select port 3 and port 3 status
CMD>ENTER08
1.3A0B009C0D1G0I00000L1N0O00000T010U3

                            Status message shows all default options are in use
CMD>OUTPUT08;C2X            Change parity option for port 3
CMD>OUTPUT08;C?X            Request current parity option
CMD>ENTER08
C2                          C2 is returned since we changed the default
CMD>OUTPUT09;hello message
                            send "hello message" out port 3
CMD>OUTPUT08;P1X            select port 1
```

```
CMD>OUTPUT09;hello message
```
send "hello message" out port 1
```
CMD>ENTER08
1.3A0B009C2D1G0I00000L1N0O00000T010U3
```
port 3 status is returned since U3 has not changed
```
CMD>ENTER09
???
```
port 3 serial data is returned if any has been received

The above examples may also be repeated using Secondary addressing. To do this, turn off the Serial488/4, set SW1 for Secondary addressing mode. The session would then appear as presented below.

```
CMD>
```
Command prompt from Keyboard Controller program
```
CMD>OUTPUT0800;S0X
```
Restore configuration to factory default values
```
CMD>CLEAR
```
Reset the Serial488/4
```
CMD>ENTER0800
1.3C0E0F0G0I000K0M000P0R0Y0Z55880
```
Status message shows all default options are in use
```
CMD>OUTPUT0800;P1U1X
```
select port 1 and port 1 status
```
CMD>ENTER0800
1.3A0B009C0D1G0I00000L1N0O00000T010U1
```
Status message shows all default options are in use
```
CMD>OUTPUT0800;a1b7d0x
```
Change the configuration of port 1

```
CMD>ENTER0800
1.3A1B007C0D0G0I00000L1N0O00000T010U1
```
Status message shows new options are in use
```
CMD>OUTPUT0800;P3U3X
```
select port 3 and port 3 status
```
CMD>ENTER0800
1.3A0B009C0D1G0I00000L1N0O00000T010U3
```
Status message shows all default options are in use
```
CMD>OUTPUT0800;C2X
```
Change parity option for port 3
```
CMD>OUTPUT0800;C?X
```
Request current parity option

```
CMD>ENTER0800
C2                      C2 is returned since we changed the default
CMD>OUTPUT0803;hello message
                        send "hello message" out port 3
CMD>OUTPUT0801;hello message
                        send "hello message" out port 1
CMD>ENTER0800
1.3A0B009C2D1G0I00000L1N0O00000T010U3
                        port 3 status is returned since U3 has not changed
CMD>ENTER0801
???
                        port 1 serial data is returned if any has been received
CMD>ENTER0803
???
                        port 3 serial data is returned if any has been received
```

Command Descriptions

Control of the Serial488/4 is implemented with 22 bus commands, described here in detail. Examples are given for the commands using a PC functioning as an IEEE 488 controller by using the IOtech GP488B board and associated driver software, Driver488. All examples are given using GWBASIC or BASICA. The Serial488/4 bus address is assumed to be set to 8 for all examples.

In order to establish communication with Driver488 from GWBASIC or BASICA, the following sequence must be used:

```
10 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
20 IOCTL#1,"BREAK"
30 PRINT#1,"RESET"
40 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
```

All of the command examples that follow assume that the driver has been properly opened and reset by the above sequence.

All commands may be entered in upper or lower case. Each command offers a query (?) option. This option can be used to determine the present configuration or mode of a command previously executed. In addition, there are four commands that are query only (i.e., they are used as status commands and have no set options). Any number of query commands may be combined into one string to allow the user to construct a specialized status command requesting the Serial488/4 to return only that information which is of interest .

Attention!

It is necessary that the **EXECUTE** command **(X)** follow all command strings sent to the Serial488/4. No commands are executed until an **X** is received by the Serial488/4**.** The only exception to this rule are the query commands. Query commands do not have to be followed by an **X**.

Stop Bits An

The Stop Bits command programs the number of stop bits contained in each character transmitted and received by the selected serial port. The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use. The factory default setting as set by the Save Configuration command (`S0`) is `A0`, 1 stop bit.

`A0`	One stop bit selected.
`A1`	Two stop bits selected.
`A?`	Return current stop bit setting.

Example 1: Set stop bits for ports 1 and 3 using dual primary addressing mode

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1X"         select port 1 as the current port
PRINT#1,"OUTPUT08;A0X"         select 1 stop bit
PRINT#1,"OUTPUT08;P3A1X"       select 2 stop bits for port 3
```

Example 2: Set stop bits for ports 1 and 3 using secondary addressing mode

```
PRINT#1,"CLEAR0800"            reset the Serial488/4
PRINT#1,"OUTPUT0800;P1X"       select port 1 as the current port
PRINT#1,"OUTPUT0800;A0X"       select 1 stop bit
PRINT#1,"OUTPUT0800;P3A1X"     select 2 stop bits for port 3
```

Baud Rate Bn

The Baud Rate command programs the speed at which serial data is transmitted and received by the selected serial port. The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use. Additionally the `CTS` line may be used to input a clock signal which is sixteen times the desired baud rate if `B11` is selected. The factory default setting as set by the Save Configuration command (`S0`) is `B9`, 9600 baud.

NOTE: The sum of the baud rates for all four ports cannot exceed 38,400 baud or data may be lost.

`B0`	110 baud selected
`B1`	300 baud selected
`B2`	600 baud selected
`B3`	1200 baud selected
`B4`	1800 baud selected
`B5`	2400 baud selected
`B6`	3600 baud selected
`B7`	4800 baud selected
`B8`	7200 baud selected
`B9`	9600 baud selected
`B10`	19200 baud selected
`B11`	16 x External clock selected
`B?`	Return current baud setting

Example1: Set baud rates for ports 1 and 2 using dual primary addressing mode

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1X"         select port 1 as the current port
PRINT#1,"OUTPUT08;B3X"         select 1200 baud
PRINT#1,"OUTPUT08;P2B10X"      select 19200 baud on  port 2
```

Example 2: Set baud rates for ports 1 and 2 using secondary addressing mode

```
PRINT#1,"CLEAR0800"            reset the Serial488/4
PRINT#1,"OUTPUT0800;P1B3X"     select 1200 baud for port 1
PRINT#1,"OUTPUT0800;P3b10X"    select 19200 baud for port 3
```

Parity Cn

The Parity command specifies the parity, if any, to be transmitted and received with each character of the selected serial port. The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use. If even or odd parity is enabled the Serial488/4 will append a bit of the correct parity to each character transmitted. However, the Serial 488/4 does not check parity on characters received. The factory default setting as set by the Save Configuration command (`S0`) is `C0`, parity disabled.

`C0`	Parity disabled.
`C1`	Odd parity enabled.
`C2`	Even parity enabled.
`C?`	Return current parity setting.

Example 1: Set parity for port 2 using dual primary addressing mode.

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P2C1X"       select odd parity for port 2
```

Example 2: Set parity for port 4 using secondary addressing mode.

```
PRINT#1,"CLEAR0800"            reset the Serial488/4
PRINT#1,"OUTPUT0800;P4C2X"     select even parity for port 4
```

Data Bits Dn

The Data Bits command programs the number of data bits contained in each character transmitted and received by the selected serial port. The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use. The factory default setting as set by the Save Configuration command (`S0`) is `D1`, eight data bits.

`D0`	Seven data bits selected.
`D1`	Eight data bits selected.
`D?`	Return current data bits setting

Example1: Set data bits for ports 1 and 3 using dual primary addressing mode.

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1X"         select port 1 as the current port
PRINT#1,"OUTPUT08;D0X"         select 7 data bits
PRINT#1,"OUTPUT08;P3D1X"       select 8 data bits for port 3
```

Example 2: Set data bits for ports 1 and 3 using secondary addressing mode.

```
PRINT#1,"CLEAR0800"            reset the Serial488/4
PRINT#1,"OUTPUT0800;P1X"       select port 1 as the current port
PRINT#1,"OUTPUT0800;d0X"       select 7 data bits
PRINT#1,"OUTPUT0800;P3d1X"     select 8 data bits for port 3
```

Error Query E?

The Error Query is used to determine the present error condition of the Serial488/4. After execution of the Error Query, the present error condition is cleared. The error condition is also returned and cleared by the status `Un` command.

`E?` Return and clear present error condition

Example1: Query the error status using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;E?"          request error status
PRINT#1,"ENTER08"
INPUT#2,A$
PRINT A$                       display shows E0, No errors
```

Example2: Query the error status using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;W5X"       send an illegal command to the Serial488/4
PRINT#1,"OUTPUT0800;E?"        request error status
PRINT#1,"ENTER0800"
INPUT#2,A$
PRINT A$                       display shows E1, unrecognized command

PRINT#1,"OUTPUT0800;E?"        request error status again
PRINT#1,"ENTER0800"
INPUT#2,A$
PRINT A$                       display shows E0, error condition has
                               been cleared after reading error status
```

Flush Fn

The Flush command is used to discard serial data for a selected port. The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use. Data waiting to be read over the IEEE bus, transmitted out the serial port, or both, may be flushed.

`F0`	Flush serial input data.
`F1`	Flush serial output data.
`F2`	Flush both serial input and output data.
`F?`	Return last F command executed.

Example 1: Flush serial input data for port 1 using dual primary addressing mode.

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1X"         select port 1 as the current port
PRINT#1,"OUTPUT08;F0X"         flush serial input data
```

Example 2: Flush serial output data for port 2 using secondary addressing mode.

```
PRINT#1,"CLEAR0800"            reset the Serial488/4
PRINT#1,"OUTPUT0800;P2X"       select port 2 as the current port
PRINT#1,"OUTPUT0800;f1X"       flush serial output data
```

Handshake Type Gn

The Handshake Type command determines the flow control method to be used in serial data transfers for the selected serial port. The Port Select command (Pn) is used to select the serial port regardless of addressing mode in use.

The hardware handshake method uses the RTS and CTS signals of the serial interface to prevent data overruns by asserting the RTS line when the Serial488/4 is able to accept data and requiring the CTS signal to be asserted before transmitting data.

The software handshake method uses two special serial characters, XON (11 hex) and XOFF (&H13) to control data flow. The XOFF character is sent to stop data transmission and the XON character is sent to resume data transmission. It should be noted that the software method is not compatible with binary data as the data itself may contain XON and XOFF characters.

If no handshaking is desired, handshaking may be disabled entirely. The factory default setting as set by the Save Configuration command (S0) is G0, hardware handshaking (RTS/CTS).

G0	Hardware handshaking (RTS/CTS) is selected.
G1	Software handshaking (XON/XOFF) is selected.
G2	No handshaking is selected.
G?	Return present handshaking method.

Example 1: Select software handshaking on port 1 using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;P1X"          select port 1 as the current port
PRINT#1,"OUTPUT08;G1X"          select software handshaking
```

Example 2: Select hardware handshaking on port 2 using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;P2G0X"      select hardware handshaking for port 2
```

Input Query I?

The Input Query command returns the number of characters available to be read on the the IEEE bus from the selected serial port (i.e. the port's serial input buffer). The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use.

`I?` Return number of characters in the input buffer.

Example 1: Find the number of input characters available on port 1 using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;P1x"          select port 1
PRINT#1,"OUTPUT08;I?"           request number of serial input characters
PRINT#1,"ENTER08"
INPUT#2,A$                      read the response
PRINT A$                        display shows I00010, 10 characters
```

Example 2: Find number of input characters available on port 1 using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;P1xI?"      select port 1 and request number of input
PRINT#1,"ENTER0800"             characters available
INPUT#2,A$                      read the response
PRINT A$                        display shows I00033, 33 characters
```

Status EOI Kn

The EOI line is one of five interface management lines of the IEEE 488 bus. It is used to indicate the end of a multiple byte message. The Status EOI command specifies how this line will be used when the Serial488/4 sends port and command status messages to the addressed listener, usually the bus controller. In the K0 mode, the EOI line is asserted by the Serial488/4 on the last byte of every status message. In the K1 mode the EOI line is not used. The factory default setting as set by the Save Configuration command (S0) is K1, EOI is disabled.

K0	Assert EOI on last byte of status message.
K1	EOI is disabled.
K?	Return last EOI command received.

Example 1: Select EOI on last byte of message using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;k0X"
```

Example 2: Disable EOI on status messages using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;K1X"
```

Data EOI Ln

The Data EOI command specifies how the IEEE 488 EOI line will be used when the Serial488/4 sends serial port data to the addressed listener, usually the bus controller, for the selected serial port. The Port Select command (Pn) is used to select the serial port regardless of addressing mode in use. In the L0 mode, the EOI line is asserted by the Serial488/4 when the data byte matches the selected serial terminator (Txxx). In the L1 mode the EOI line is not used. If the L2 mode is selected, the EOI line is asserted when the last serial input character is sent over the IEEE bus. In the L3 mode the EOI line is asserted on either a match with the serial terminator or the last serial character sent. The factory default setting as set by the Save Configuration command (S0) is L1, EOI is disabled.

L0	Assert EOI on serial terminator character.
L1	EOI is disabled.
L2	Assert EOI on last serial input character.
L3	Assert EOI on serial terminator character or last serial character.
L?	Returns last Data EOI command selected.

Example 1: Select EOI on serial terminator using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;P1X"          select port 1 as the current port
PRINT#1,"OUTPUT08;L0X"          select data EOI on serial terminator
```

Example 2: Disable EOI on selected port using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;p1L1X"      disable EOI on port 1
```

Service Request (SRQ) Mn

The Service Request (SRQ) mode is used by the Serial488/4 to alert the controller to one of several conditions described below. Multiple SRQ conditions can be enabled simultaneously by issuing them separately or by combining them in one command. If multiple SRQ commands are contained within the same command string, each SRQ command should be followed by an Execute command (X). The programmed SRQ modes will remain enabled until the M0 command is sent, or the controller sends a Device Clear (DCL) or Selected Device Clear (SDC). The factory default setting as set by the Save Configuration command (S0) is M0, SRQ is disabled.

M0	SRQ is disabled
M1	SRQ on receipt of data on serial port 1
M2	SRQ on receipt of data on serial port 2
M4	SRQ on receipt of data on serial port 3
M8	SRQ on receipt of data on serial port 4
M16	SRQ on Ready
M32	SRQ on Error condition
M128	SRQ on Memory low condition
M?	Returns current SRQ mask

MO default mode disables the SRQ function, preventing the Serial488/4 from generating a Service Request.

M1, M2, M4 and M8 will generate a Service Request when the data is received on serial ports 1, 2, 3, and 4 respectively.

M16 will generate a Service Request when the Serial488/4 has completed the execution of a set of commands from the bus controller. This is used by the controller to assure the completion of a set of commands before sending a subsequent set of commands.

M32 will generate a Service Request when an error occurs. The most common bus error is sending an invalid command to the Serial488/4. For example, attempting to select an 'B15' baud rate when no 'B15' baud rate exists will generate a Service Request when the M32 mode is selected.

`M128` will generate a Service Request when the Serial488/4 has used up most of its available memory. This provides an indication that no further data should be sent to the serial ports until data is freed. Data may be freed either by reading serial input data from the Serial488/4, waiting for serial output data to finish being transmitted, or by flushing a serial port if data is not moving as expected.

Example 1: Enable SRQ on serial data or low memory using dual primary addressing mode.

PRINT#1,"OUTPUT08;M0X"	clear SRQ mask
`PRINT#1,"OUTPUT08;M1X"`	select SRQ on serial port 1 data
`PRINT#1,"OUTPUT08;M2X"`	select SRQ on serial port 2 data
`PRINT#1,"OUTPUT08;M4X"`	select SRQ on serial port 3 data
`PRINT#1,"OUTPUT08;M8X"`	select SRQ on serial port 4 data
`PRINT#1,"OUTPUT08;M128X"`	select SRQ memory low

Example 2: Enable SRQ on serial data or low memory using dual primary addressing mode. This example has the same effect as example 1.

`PRINT#1,"OUTPUT08;M132X"`	select SRQ on serial data or memory low

Example 3: Enable SRQ on error using secondary addressing mode.

`PRINT#1,"OUTPUT0800;M0X"`	clear SRQ mask
`PRINT#1,"OUTPUT0800;M32X"`	select SRQ on command error

Handshake Control Nn

The Handshake Control, in conjunction with the Handshake Type command, determines what flow control method is to be used in serial data transfers for the selected port. The Port Select command (Pn) is used to select the serial port regardless of addressing mode in use. If the hardware handshake method (G0) is being used, the handshake control command will effect the serial port's RTS line. Handshaking control may be selected to asserted and un-asserted the RTS line automatically as memory is available, asserted the RTS line continually or un-asserted RTS line continually.

When the software handshake method (G0) is used, the handshake control command will effect the sending of the handshake characters XON and XOFF. Handshaking control may be selected to enable the automatic output of XON/XOFF serial characters as memory dictates, cause the XOFF character to be issued, or cause the XON character to be issued. Additionally this command may be used to enable the output of sixteen times the baud rate clock on the RTS line.

If the no handshake method (G2) is in use, the Handshake Control command controls the assertion, un-assertion, or output of a signal sixteen times baud clock output on the RTS line.

The factory default setting as set by the Save Configuration command (S0) is N0, automatic handshaking.

N0	Automatic handshaking, RTS/CTS or XON/XOFF.
N1	Hold off serial input, un-assert RTS or send XOFF.
N2	Release holdoff of serial input, assert RTS, or send XON.
N3	Automatic XON/XOFF handshake with 16 x baud clock output on RTS line, or no handshaking and 16 x baud clock output on RTS line.
N?	Return present handshaking control selection.

Example 1: Select automatic hardware handshaking using dual primary addressing mode.

`PRINT#1,"CLEAR08"`	reset the Serial488/4
`PRINT#1,"OUTPUT08;P1X"`	select port 1 as the current port
`PRINT#1,"OUTPUT08;G0N0X"`	select automatic hardware handshaking

Example 2: Select software handshaking with baud clock output using secondary addressing mode.

`PRINT#1,"CLEAR0800"`	reset the Serial488/4
`PRINT#1,"OUTPUT0800;P2X"`	select port 2 as the current port
`PRINT#1,"OUTPUT0800;g1n3X"`	select software handshaking with baud clock output

Output Query O?

The Output Query command returns the number of characters waiting to be transmitted out on the selected serial port (i.e. the port's serial output buffer). The Port Select command (`Pn`) is used to select the serial port regardless of addressing mode in use.

`O?` Return number of characters in the serial output buffer.

Example 1: Find number of output characters on port 1 using dual primary addressing mode.

`PRINT#1,"OUTPUT08;P1x"`	select port 1
PRINT#1,"OUTPUT08;O?"	request number of serial output characters
`PRINT#1,"ENTER08"`	
INPUT#2,A$	read the response
`PRINT A$`	display shows `O00010`, 10 characters

Example 2: Find number of output characters on port 1 using secondary addressing mode.

`PRINT#1,"OUTPUT0800;P1xO?"`	select port 1 and request number of output
PRINT#1,"ENTER0800"	characters waiting to be sent
`INPUT#2,A$`	read the response
`PRINT A$`	display shows `O00033`, 33 characters

Port Select Pn

The Port Select command determines which port is selected for execution of serial setup commands and for data input/output when using the Dual Primary addressing mode. The factory default setting as set by the Save Configuration command (`S0`) is `P1`, port 1 selected.

`P1`	Port 1 is selected
`P2`	Port 2 is selected
`P3`	Port 3 is selected
`P4`	Port 4 is selected
`P?`	Return present port selection

Example 1: Select port 1, then send a message using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;P1X"          select port 1 as the current port
PRINT#1,"OUTPUT08;a0b5c1d1X"    select 1 stop bit, 2400 baud, odd parity
                                8 data bits
PRINT#1,"OUTPUT08;g1n0t13X"     select Xon/Xoff, automatic
        handshaking, terminator CR
PRINT#1,"OUTPUT09;Startup       send a startup message out port 1
        message"
```

Example 2: Select port 1, then send a message using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;P1X"          select port 1 as the current port
PRINT#1,"OUTPUT0800;a0b5c1d1X"    select 1 stop bit, 2400 baud, odd parity,
                                  8 data bits
PRINT#1,"OUTPUT0800;g1n0t13X"     select Xon/Xoff, automatic
                                  handshaking, terminator CR
PRINT#1,"OUTPUT0801;Startup       send a startup message out port 1
        message"
```

Break Qn

The Break command enables or disables the sending of break (a constant stream of '0' bits) by the selected port . While a port is sending break no characters can be transmitted. When sending break is subsequently disabled, normal character transmission will resume. The factory default setting as set by the Save Configuration command (`S0`) is `Q0`, disable sending break.

`Q0`	Disable sending break.
`Q1`	Enable sending break.
`Q?`	Return current break setting.

NOTE: The programmer must ensure the break command is enabled for more than one character time before issuing the disable sending break command to ensure that break is recognized.

NOTE: Sending a break while a port is transmitting data may cause the character currently being transmitted to become corrupted. The Output Query (O?) may be used to verify that the serial output buffer is empty. Once the serial output buffer is empty a delay of two character times is required to ensure that all serial data has been transmitted and will not be corrupted by sending break.

Example 1: Program port 1 to send a break using dual primary addressing mode

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1X"         select port 1 as the current port
PRINT#1,"OUTPUT08;Q1X"         send break
FOR LUP = 1 to 100:NEXT LUP    delay a short while
PRINT#1,"OUTPUT08;Q0X"         stop sending break
```

Example 2: Program port 3 to send a break using secondary addressing mode

```
PRINT#1,"CLEAR0800"            reset the Serial488/4
PRINT#1,"OUTPUT0800;P3X"       select port 3 as the current port
PRINT#1,"OUTPUT0800;Q1X"       send break
FOR LUP = 1 to 100:NEXT LUP    delay a short while
PRINT#1,"OUTPUT0800;Q0X"       stop sending break
```

Save Configuration Sn

The Save Configuration command is used to save a non-volatile copy of the present configuration for the Serial 488/4 to be used as the power-up configuration. Note that this command is independent of port selection; it saves all Serial 488/4 port and command settings. The `S0` command is used to restore the factory default configuration.

`S0`	Save the default configuration in non-volatile RAM.
`S1`	Save the current configuration in non-volatile RAM.
`S?`	Return the last Save configuration command.

Example 1: Configure each port; save settings using dual primary addressing mode.

`PRINT#1,"OUTPUT08;k0m16u0y2X"`	select EOI with last bus terminator, SRQ on ready, command status, terminator CR LF
`PRINT#1,"OUTPUT08;P1X"`	select port 1 as the current port
`PRINT#1,"OUTPUT08;a0b5c1d1X"`	select 1 stop bit, 2400 baud, odd parity, 8 data bits
`PRINT#1,"OUTPUT08;g1n0t13X"`	select Xon/Xoff, automatic handshaking, terminator CR
`PRINT#1,"OUTPUT08;P2X"`	select port 2 as the current port
`PRINT#1,"OUTPUT08;a1b3c0d0X"`	select 2 stop bits, 1200 baud, no parity, 7 data bits
`PRINT#1,"OUTPUT08;g0n0t10X"`	select RTS/CTS, automatic handshaking, terminator LF
`PRINT#1,"OUTPUT08;P3X"`	select port 3 as the current port
`PRINT#1,"OUTPUT08;a0b10c2d1X"`	select 1 stop bit,19200 baud, even parity, 8 data bits
`PRINT#1,"OUTPUT08;g1n0t13X"`	select Xon/Xoff, automatic handshaking, terminator CR
`PRINT#1,"OUTPUT08;P4X"`	select port 4 as the current port
`PRINT#1,"OUTPUT08;a1b1c0d0X"`	select 2 stop bit, 300 baud, no parity, 8 data bits
`PRINT#1,"OUTPUT08;g1n3t13X"`	select Xon/Xoff, automatic handshaking, output baud clock , terminator CR
`PRINT#1,"OUTPUT08;S1x"`	save the configuration

Example 2: Restore the default configuration using secondary addressing mode.

PRINT#1,"OUTPUT0800;S0x"	restore the factory default configuration
PRINT#1,"CLEAR0800"	reset the Serial488/4 to re-configure

Serial Terminator Tn

The Serial Terminator command is used in conjunction with the Data EOI (Ln) command to specify which character to assert EOI on for the selected serial port. The Port Select command (Pn) is used to select the serial port regardless of addressing mode in use. The terminator value may range from 0 to 255 allowing any 7 bit ASCII character or 8 bit data value to be used. The factory default setting as set by the Save Configuration command (S0) is T010, the line feed character.

Tn	Select serial terminator where n is a value in the range 0 thru 255
T?	Return the currently selected serial terminator.

Example 1: Set terminators for ports 1 and 3 using dual primary addressing mode.

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1X"         select port 1 as the current port
PRINT#1,"OUTPUT08;T13X"        select CR terminator
PRINT#1,"OUTPUT08;P3X"         select port 3 as the current port
PRINT#1,"OUTPUT08;T0X"         select NUL character terminator
```

Example 2: Set terminators for ports 1 and 3 using secondary addressing mode.

```
PRINT#1,"OUTPUT0800;P1X"       select port 1 as the current port
PRINT#1,"OUTPUT0800;T13X"      select CR terminator
PRINT#1,"OUTPUT0800;P3X"       select port 3 as the current port
PRINT#1,"OUTPUT0800;T0X"       select NUL terminator
```

Status Un

The Status command selects the type of status message the Serial488/4 will send when addressed to talk on the command address. The `U0` selection causes the command status to be selected. The `U1` thru `U4` selections cause a serial port status to be selected. The status of the Serial488/4 may be read at any time without interfering with normal operation. Any error conditions are cleared after the command status string is read by the controller. The factory default setting as set by the Save Configuration command (`S0`) is `U0`, command status is selected.

NOTE: The Status selected is independent of the serial port selected via the Port command (`Pn`).

`U0`	Send the Serial488/4 command status when addressed to talk
`U1`	Send the status of serial port 1 when next addressed to talk
`U2`	Send the status of serial port 2 when next addressed to talk
`U3`	Send the status of serial port 3 when next addressed to talk
`U4`	Send the status of serial port 4 when next addressed to talk
`U?`	Return the current status command selection

The format of the status byte returned by the Serial488/4 after receiving a `U0` command is as follows:

```
*.*E#K#M###P#U#Y#Z#####
```

where each # equals the number corresponding to that command. The leading information `*.*` is the revision level of the Serial488/4 firmware.

Example 1: Read Serial488/4 command status using dual primary addressing mode.

```
DIM A$(50)                      dimension A$
PRINT#1,"CLEAR08"               reset the Serial488/4
PRINT#1,"OUTPUT08;U0X"          send U0 to the Serial488/4
PRINT#1,"ENTER08"
INPUT#2,A$                      read the status byte
PRINT A$                        shows 1.0E0K1M000P1Y2
```

The format of the status byte returned by the Serial488/4 after receiving a U1 through U4 command is as follows:

```
*.*A#B#C#D###G#I#####L#N#O#####Q#T###U#
```

where each # equals the number corresponding to that command. The leading information *.* is the revision level of the Serial488/4 firmware.

Example 1: Read Serial488/4 command status using secondary addressing mode.

```
DIM A$(50)                          dimension A$
PRINT#1,"OUTPUT0800;S0X"            restore Serial488/4 defaults
PRINT#1,"CLEAR0800"                 reset the Serial488/4
PRINT#1,"OUTPUT0800;U1X"            send U1 to the Serial488/4
PRINT#1,"ENTER0800"
INPUT#2,A$                          read the port 1 status message
PRINT A$                            shows

1.0A0B009C0D1G0I00000L1N0O00000Q0T010U1
```

Below is a summary of the Command Status (U0) information.

E#	Error Message
E0	No error
E1	Unrecognized command (ex. W3)
E2	Illegal command option (ex. F8)
E3	Conflict (handshake type, handshake control, baud select)
K#	End Or Identify
K0	EOI enabled on last byte of status message
K1	EOI disabled
M###	Service Request
M0	SRQ is disabled
M1	SRQ on serial port #1 data in
M2	SRQ on serial port #2 data in
M4	SRQ on serial port #3 data in

M8	SRQ on serial port #4 data in
M16	SRQ on Ready
M32	SRQ on error condition
M128	SRQ on memory low condition

Note: the status indication reflects the sum of all received Service Request commands.

P#	Selected Port
P1	Port 1 selected
P2	Port 2 selected
P3	Port 3 selected
P4	Port 4 selected

U#	Status message
U0	Command status

Y#	Terminator
Y0	CR only
Y1	LF only
Y2	CR LF
Y3	LF CR

Z#####	Free buffer memory available for serial data

Below is a summary of the Serial Port Status (U1-4) information.

A#	Stop Bits
A0	1 stop bit
A1	2 stop bits

B###	Baud Rate
B0	110 Baud selected
B1	300 Baud selected
B2	600 Baud selected
B3	1200 Baud selected
B4	1800 Baud selected
B5	2400 Baud selected

B6	3600 Baud selected
B7	4800 Baud selected
B8	7200 Baud selected
B9	9600 Baud selected
B10	19200 Baud selected
B11	16 x External clock selected
C#	Parity
C0	Parity is disabled.
C1	Odd parity is enabled.
C2	Even parity is enabled.
D#	Data Bits
D0	Seven data bits selected.
D1	Eight data bits selected.
G#	Handshake Type
G0	Hardware handshaking (RTS/CTS) is selected.
G1	Software handshaking (XON/XOFF) is selected.
G2	No handshaking is selected.
I#####	Serial input bytes available to read
L#	Data EOI
L0	Assert EOI on serial terminator character.
L1	EOI is disabled.
L2	Assert EOI on last serial input character.
L3	Assert EOI on serial terminator character or last serial character.
N#	Handshake Control
N0	Automatic handshaking, RTS/CTS or XON/XOFF
N1	Hold off serial input, un-assert RTS or send XOFF.
N2	Release holdoff of serial input, assert RTS or Send XON.
N3	Automatic XON/XOFF handshake with Baud Clock x 16 output on RTS line, or no handshaking and Baud Clock x 16 output on RTS line.
O#####	Serial output bytes not yet transmitted out the serial port

`Q#`	Break
`Q0`	Stop sending break characters.
`Q1`	Send break characters.
`T###`	Current Serial Terminator character
`U#`	Selected Serial Port Status
`U1`	Serial port 1 status
`U2`	Serial port 2 status
`U3`	Serial port 3 status
`U4`	Serial port 4 status

Version Query V?

The Version Query command returns the Serial488/4 firmware revision number.

`V?` Return Serial488/4 firmware revision number.

Example 1: Get Serial488/4 firmware revision using dual primary addressing mode.

```
PRINT#1,"OUTPUT08;V?"          ask for version information
PRINT#1,"ENTER08"              read the response
INPUT#2,A$
PRINT A$                       display shows   1.3   (or higher)
```

Execute X

Commands sent to the Serial488/4 will result in no action until the unit is instructed to execute these commands. This is done by sending an X, usually as the last character of a command string. Commands sent without an X are stored in the internal buffer until an X is received. Any number of Execute commands may be inserted into the same command string.

X Execute previous Serial488/4 command(s).

Example 1: Execute a Serial488/4 command using dual primary addressing mode.

```
PRINT#1,"CLEAR08"              reset the Serial488/4
PRINT#1,"OUTPUT08;P1F2"        send "F2" to the Serial488/4 command
                               Flush port 1 input and output buffers
PRINT#1,"OUTPUT08;X"           instruct the Serial488/4 to execute its
                               command input buffer
```

Bus Terminator Yn

The IEEE 488 bus terminator command selects the characters sent at the end of status messages returned by the Serial488/4. The terminator may be programmed for any combination of Carriage Return (CR) and Line Feed (LF). `Y0` sends CR only and `Y1` sends LF only. The `Y2` mode, the most commonly accepted terminator, sends CR-LF. `Y3` reverses the sequence to send LF-CR. The factory default setting as set by the Save Configuration command (`S0`) is `Y2`, CR-LF.

`Y0`	CR only
`Y1`	LF only
`Y2`	CR LF
`Y3`	LF CR
`Y?`	Return current bus terminator selection.

Example 1: Set terminator to linefeed carriage return using dual primary addressing mode.

```
PRINT#1,"CLEAR08"
PRINT#1,"OUTPUT08;Y3X"          select line feed  carriage return terminator
```

Memory Query Z?

The Memory Query command returns the number of characters which may be written to the serial ports without causing a 'low memory' condition. This command may be used to for determining how many more characters can be safely written to the Serial488/4 before handshaking will attempt to stop serial input data because of a 'low memory' condition.

`Z?` Return the number of characters which may be written to the serial ports

Example 1: Find the amount of free memory remaining in the Serial488/4 using dual primary addressing mode.

PRINT#1,"OUTPUT08;Z?"	request amount of free memory
`PRINT#1,"ENTER08"`	
INPUT#2,A$	read the response
`PRINT A$`	display shows `Z23456`

Example 2: Find the amount of free memory remaining in the Serial488/4 using secondary addressing mode.

`PRINT#1,"OUTPUT0800;Z?"`	request amount of free memory
`PRINT#1,"ENTER0800"`	
INPUT#2,A$	read the response
`PRINT A$`	display shows `Z23456`

Serial Poll Status Byte

The Serial Poll Status byte is sent upon receiving the serial poll command from the controller. Refer to the SRQ description for details on how the Serial Poll Status byte is affected. Below is a description of the significance of each bit in the Serial Poll Status byte.

Bit Location	Value	(SRQ Bit Value if set to logic 1)
DIO1(LSB)	1	serial data available on port 1
DIO2	2	serial data available on port 2
DIO3	4	serial data available on port 3
DIO4	8	serial data available on port 4
DIO5	16	Ready for more commands
DIO6	32	Command Error
DIO7	64	Service Request bit
DIO8 (MSB)	128	Memory low warning

Serial Poll Bit Description

DIO1 The DIO1 bit is set by the receipt of serial data on port 1. When enabled by the `M1` command this condition will also generate a Service Request. DIO1 is cleared when no more data is available to be read on port 1.

DIO2 The DIO2 bit is set by the receipt of serial data on port 2. When enabled by the `M2` command this condition will also generate a Service Request. DIO2 is cleared when no more data is available to be read on port 2.

DIO3 The DIO3 bit is set by the receipt of serial data on port 3 When enabled by the `M4` command this condition will also generate a Service Request. DIO3 is cleared when no more data is available to be read on port 3.

DIO4 The DIO4 bit is set by the receipt of serial data on port 4. When enabled by the `M8` command this condition will also generate a Service Request. DIO4 is cleared when no more data is available to be read on port 4.

DIO5 The DIO5 bit is set after an entire command string has been received and processed by the Serial488/4. The bit is cleared while the Serial488/4 is processing commands which have been received from the controller. When used with the `M16` command, a Service Request will also be generated when

the DIO5 bit is set. An Execute command (X) must be received before the DIO5 bit can be cleared.

DIO6 DIO6 is set when an invalid command is sent to the Serial488/4. The `M32` command will enable a Service Request to occur when an invalid command is received. This bit is cleared after the controller sends a Status (`U0`), or Error Query (`E?`) command and reads the status string from the Serial488/4.

DIO7 When the Serial488/4 generates a Service Request, the DIO7 bit will be set to a logic one. This is used by the controller to determine that the Service Request was generated by the Serial488/4.

DIO8 DIO8 is when the Serial488/4 memory is low. The `M128` command enables a Service Request to occur when the memory low condition is encountered. This bit is cleared when a memory low condition no longer exists.

Example:

`PRINT#1,"CLEAR08"`	reset the Serial488/4
`PRINT#1,"OUTPUT08;M32X"`	select SRQ on Bus error
`PRINT#1,"OUTPUT08;F7X"`	send an invalid bus command. ERROR and SRQ LEDs should illuminate
`PRINT#1,"SPOLL08"`	display should be `112` (64+16+32)

64 denotes the Serial488/4 was the source of the SRQ. 16 denotes the Serial488/4 is READY for more commands. 32 denotes a Command error. When serial polled, the SRQ LED will turn off.

IEEE 488 Primer

4.1 HISTORY

The **IEEE 488** bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978. The **Digital488** conforms to this most recent revision designated **IEEE 488-1978**.

Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the **IEEE 488** interface (sometimes called the **G**eneral **P**urpose **I**nterface **B**us or **GPIB**) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the **IEEE 488** standard has moved the responsibility of the user from design of the interface to design of the high level software that is specific to the measurement application.

4.2 GENERAL STRUCTURE

The main purpose of the **GPIB** is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will talk and who will listen usually falls on the **System Controller** which is, at power on, the **Active Controller**.

The **System Controller** is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the **Active Talker**. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called **Addressing to Talk**. If the committee chairman can not attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the **GPIB**, this device becomes the **Active Controller**.

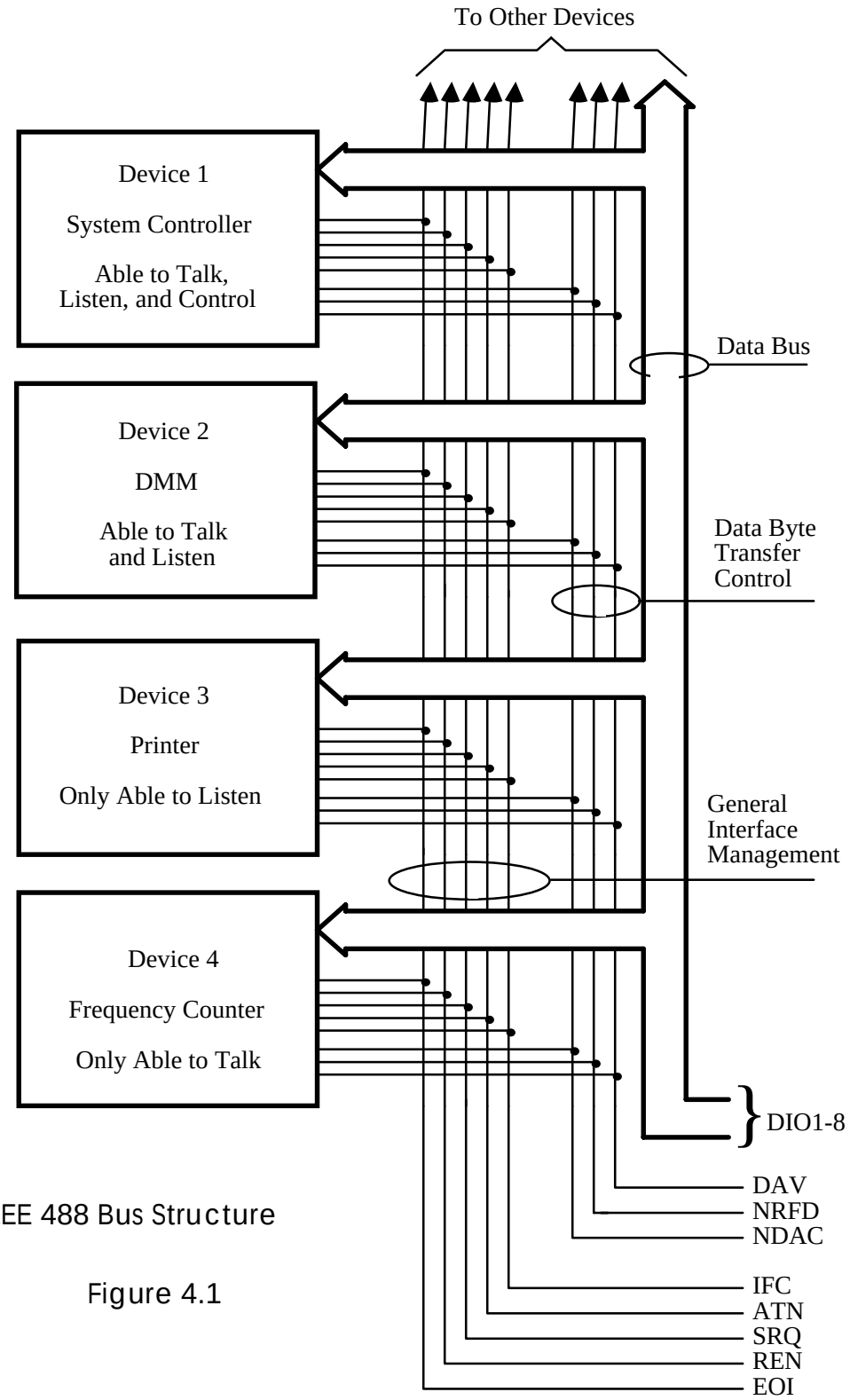
At a committee meeting, everyone present usually listens. This is not the case with the **GPIB**. The **Active Controller** selects which devices will listen and commands all other devices to ignore what is being transmitted. A device is instructed to listen by being **Addressed to Listen**. This device is then referred to as an **Active Listener**. Devices which are to ignore the data message are instructed to **Unlisten**.

The reason some devices are instructed to **Unlisten** is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (**Unlisten**) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The **GPIB** transfers information in a similar way. This method of data transfer is called **handshaking**. More on this later.

For data transfer on the **IEEE 488**, the **Active Controller** must…

a) **Unlisten** all devices to protect against eavesdroppers.
b) Designate who will **talk** by **addressing** a device to **talk**.
c) Designate all the devices who are to **listen** by **addressing** those devices to **listen**.
d) Indicate to all devices that the data transfer can take place.



IEEE 488 Bus Structure

Figure 4.1

4.3 SEND IT TO MY ADDRESS

In the previous discussion, the terms **Addressed to Talk** and **Addressed to Listen** were used. These terms require some clarification.

The **IEEE 488** standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Addresses are sent with **universal** (**multiline**) commands from the **Active Controller**. These commands include **My Listen Address** (MLA), **My Talk Address** (MTA), **Talk Address Group** (TAG), and **Listen Address Group** (LAG).

4.4 BUS MANAGEMENT LINES

Five hardware lines on the **GPIB** are used for bus management. Signals on these lines are often referred to as **uniline** (single line) commands. The signals are active low, i.e. a low voltage represents a logic "1" (asserted), and a high voltage represents a logic "0" (unasserted).

4.4.1 Attention (ATN)

ATN is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the **Active Listener**s. The **Active Controller** is the only bus device that has control of this line.

4.4.2 Interface Clear (IFC)

The **IFC** line is used only by the **System Controller**. It is used to place all bus devices in a known state. Although device configurations vary, the **IFC** command usually places the devices in the Talk and Listen Idle states (neither **Active Talker** nor **Active Listener**).

4.4.3 Remote Enable (REN)

When the **System Controller** sends the **REN** command, bus devices will respond to remote operation. Generally, the **REN** command should be issued before any bus programming is attempted. Only the **System Controller** has control of the **Remote Enable** line.

4.4.4 End or Identify (EOI)

The **EOI** line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts **EOI** during the transfer of the last data byte. The **EOI** signal is not always necessary as the end of the data may be indicated by some special character such as carriage return.

The **Active Controller** also uses **EOI** to perform a **Parallel Poll** by simultaneously asserting **EOI** and **ATN**.

4.4.5 Service Request (SRQ)

When a device desires the immediate attention of the **Active Controller** it asserts **SRQ**. It is then the Controller's responsibility to determine which device requested service. This is accomplished with a **Serial Poll** or a **Parallel Poll**.

4.5 HANDSHAKE LINES

The **GPIB** uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other **IEEE 488** lines, are active low.

4.5.1 Data Valid (DAV)

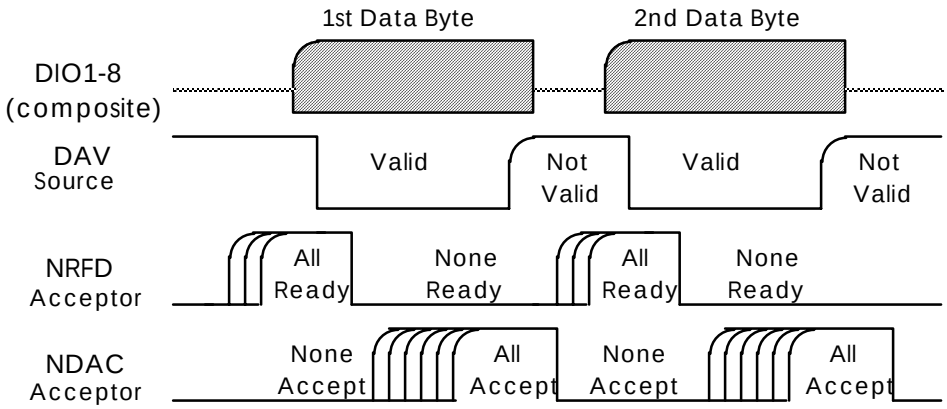
The **DAV** line is controlled by the **Talker**. The **Talker** verifies that **NDAC** is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The **Talker** then outputs data on the bus and waits until **NRFD** is unasserted (high) which indicates that all Addressed Listeners are ready to accept the information. When **NRFD** and **NDAC** are in the proper state, the **Talker** asserts **DAV** (active low) to indicate that the data on the bus is valid.

4.5.2 Not Ready for Data (NRFD)

This line is used by the **Listeners** to inform the **Talker** when they are ready to accept new data. The **Talker** must wait for each **Listener** to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.

4.5.3 Not Data Accepted (NDAC)

The **NDAC** line is also controlled by the **Listeners**. This line indicates to the **Talker** that each device addressed to listen has accepted the information. Each device releases **NDAC** (high) at its own rate, but the **NDAC** will not go high until the slowest Listener has accepted the data byte.



IEEE Bus Handshaking

4.6 DATA LINES

The **GPIB** provides eight data lines for a bit parallel/byte serial data transfer. These eight data lines use the convention of **DIO1** through **DIO8** instead of the binary designation of **D0** to **D7**. The data lines are bidirectional and are active low.

4.7 MULTILINE COMMANDS

Multiline (bus) commands are sent by the **Active Controller** over the data bus with **ATN** asserted. These commands include addressing commands for talk, listen, Untalk and Unlisten.

4.7.1 Go To Local (GTL)

This command allows the selected devices to be manually controlled. ($01)

4.7.2 Listen Address Group (LAG)

There are 31 (0 to 30) listen addresses associated with this group. The 3 most significant bits of the data bus are set to 001 while the 5 least significant bits are the address of the device being told to listen.

4.7.3 Unlisten (UNL)

This command tells all bus devices to Unlisten. The same as Unaddressed to Listen. ($3F)

4.7.4 Talk Address Group (TAG)

There are 31 (0 to 30) talk addresses associated with this group. The 3 most significant bits of the data bus are set to 010 while the 5 least significant bits are the address of the device being told to talk.

4.7.5 Untalk (UNT)

This command tells bus devices to Untalk. The same as Unaddressed to Talk. ($5F)

4.7.6 Local Lockout (LLO)

Issuing the **LLO** command prevents manual control of the instrument's functions. ($11)

4.7.7 Device Clear (DCL)

This command causes all bus devices to be initialized to a pre-defined or power up state. ($14)

4.7.8 Selected Device Clear (SDC)

This causes a single device to be initialized to a pre-defined or power up state. ($04)

4.7.9 Serial Poll Disable (SPD)

The **SPD** command disables all devices from sending their Serial Poll status byte. ($19)

4.7.10 Serial Poll Enable (SPE)

A device which is Addressed to Talk will output its Serial Poll status byte after **SPE** is sent and **ATN** is unasserted. ($18)

4.7.11 Group Execute Trigger (GET)

This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously. ($08)

4.7.12 Take Control (TCT)

This command passes bus control responsibilities from the current **Controller** to another device which has the ability to control. ($09)

4.7.13 Secondary Command Group (SCG)

These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a talk or listen address. ($60 to $7F)

4.7.14 Parallel Poll Configure (PPC)

This configures devices capable of performing a **Parallel Poll** as to which data bit they are to assert in response to a **Parallel Poll**. ($05)

4.7.15 Parallel Poll Unconfigure (PPU)

This disables all devices from responding to a **Parallel Poll**. ($15)

4.8 MORE ON SERVICE REQUESTS

Most of the commands covered, both uniline and multiline, are the responsibility of the **Active Controller** to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, **SRQ** is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the **Controller** that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert **SRQ**. A DMM might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might **SRQ** if its output has current limited. This is a powerful bus feature that removes the burden from the **System Controller** to periodically inquire, "Are you done yet?". Instead, the **Controller** says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since **SRQ** is a single line command, there is no way for the **Controller** to determine which device requested the service without additional information. This information is provided by the multiline commands for **Serial Poll** and **Parallel Poll**.

4.8.1 Serial Poll

Suppose the **Controller** receives a service request. For this example, let's assume there are several devices which could assert **SRQ**. The **Controller** issues an **SPE** (Serial Poll enable) command to each device sequentially. If any device responds with DIO7 asserted it indicates to the **Controller** that it was the device that asserted **SRQ**. Often times the other bits will indicate why the device wanted service. This **Serial Polling** sequence, and any resulting action, is under control of the software designer.

4.8.2 Parallel Poll

The **Parallel Poll** is another way the **Controller** can determine which device requested service. It provides the who but not necessarily the why. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical OR/AND conditions on one data line by more than one device. When **SRQ** is asserted, the **Controller** (under user's software) conducts a **Parallel Poll**. The **Controller** must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a **Serial Poll** might be used to determine the why.

Of the two polling types, the **Serial Poll** is the most popular due to its ability to determine the who and why. In addition, most devices support **Serial Poll** only.

Service Information

5.1 Factory Service

IOtech maintains a factory service center in Cleveland, Ohio. If problems are encountered in using the **Serial488/4** you should first telephone the factory. Many problems can be resolved by discussing the problems with our applications department. If the problem cannot be solved by this method, you will be instructed as to the proper return procedure.

5.2 Theory of Operation

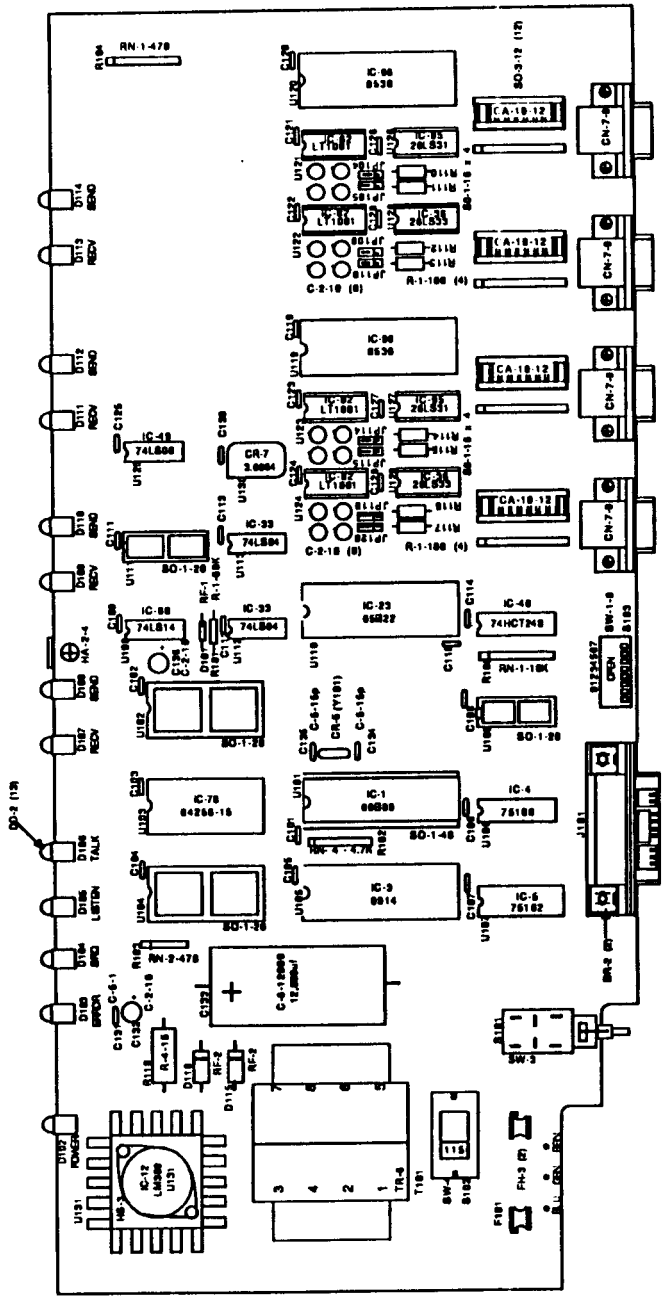
The Heart of the **Serial488/4** is a 2 Mhz 6809 microprocessor [U101] supported by 8K bytes of firmware EPROM [U102 (27128)] and 64k bytes of static RAM [U103 & U104 (58256)]. A Versatile Interface Adapter [U104 (65B22)] is used to generate real-time interrupts for the firmware operating system and drive the front panel LED annunciators through inverters [U112 & U113 (74LS04)].

The IEEE 488 bus interface is accomplished by a TMS9914A [U105] controller with drivers [U106 (75160) and U107(75162)]. The four serial interfaces use two Z8530 [U119 & U120] Serial Communications Controllers with four RS232 drivers [U121, U122, U123, &U124 (LT1081)] , two RS422 drivers [U126 & U127 (26LS31)], and two RS422 receivers [U128 & U129 (26LS33)].

Decoding of the microprocessor address space is accomplished with two Programmable Logic Array [U108 & U111 (16L8)]. The Memory space allocation is…

Address	Device	Part Number	Function
$0000-$7FFF	U103	58256	Static RAM
$8000-$E6FF	U104	58256	Static RAM
$E700-$E70F	U106	TMS9914A	IEEE Controller
$E710-$E71F	U114	74HCT240	SW1
$E760-$E76F	U120	Z8530	Serial Controller
$E770-$E77F	U119	Z8530	Serial Controller
$E780-$E7FF	U110	R65C22	VIA
$E800-$FFFF	U102	27128	Programmed EPROM

5.3 Serial488/4 Main Board Component Layout



5.4 Serial488/4 Replaceable Parts List

Schematic	Part Number	Description
C101-C131	C-5-.1	Ceramic, 25v
C132	C-2-10	Electrolytic, 25v
C133	C-1-10000	Electrolytic, 25v
C134,C135	C-5-15p	Ceramic, 25v
C136,C152	C-2-10	Electrolytic, 25v
	CN-35	Power Connector
D101	RF-1	Small Signal Diode
D102-D114	DD-2	Red PC Mount
D115,D116	RF-2	3A Diode
F101	FH-3	Fuse Clip
	FH-4	Fuse Holder
J101	CN-2	IEEE 488 Connector
J102-J105	CN-7-9	DB-9 PC Mount Male Connector
R101	R-1-68K	68K•, 1/4w carbon
R102	RN-4-4.7K	4.7K• x 7 SIP
R103,R104	RN-1-470	470• x 9 SIP
R105-R109	RN-1-10K	10K• x 9 SIP
R110-R117	R-1-100	68K•, 1/4w carbon
R118	R-4-15	15 •, 2W
S101	SW-3	DPDT Push-Push
S102	SW-4	DPDT Slide 110/220
S103	SW-1-8	8 Pole Piano Switch
T101	TR-6	Transformer 10v @1.2A
U101	IC-1	MC68B09P Microprocessor
U102	Serial488/4-600	Programmed EPROM
U103,U104	IC-46	58256 Static RAM
U105	IC-3	TMS9914ANL IEEE Controller
U106	IC-4	SN75160BN IEEE Driver
U107	IC-5	SN75162BN IEEE Driver
U108	IC-64	Programmed 16L8PAL
U109	IC-66	74LS14 Schmitt-trigger
U110	IC-23	65B22 Versatile Interface Adapter
U110	IC-64	Programmed 16L8PAL
U112,U113	IC-33	74LS04 Hex Inverter
U114,U118	IC-48	74LS240 Octal Buffer
U119,U120	IC-96	Z8530 Serial Communications Controller
U121-U124	IC-82	LT1081 RS-232 Driver Receiver
U125	IC-49	74LS08 Dual Quad Nand Gates
U126,U127	IC-95	26LS31 RS-422 Driver
U128,U129	IC-36	26LS33 RS-422/423 Receiver
U130	CR-7	3.6964 Mhz Oscillator
U131	IC-12	LM309 Regulator, + 5v
U132	IC-94	Smart Socket Battery Backup
Y101	CR-5	8.0000 MHz Crystal

Command	Code	Description
Stop Bit(s)	**A0**	Selects 1 stop bit for selected port
	A1	Selects 2 stop bits for selected port
	A?	Returns current stop bits setting
Baud Rate	**B0**	Baud rate = 110
	B1	Baud rate = 300
	B2	Baud rate = 600
	B3	Baud rate = 1200
	B4	Baud rate = 1800
	B5	Baud rate = 2400
	B6	Baud rate = 3600
	B7	Baud rate = 4800
	B8	Baud rate = 7200
	B9	Baud rate = 9600
	B10	Baud rate = 19200
	B11	Baud rate = 16 x external
	B?	Returns current baud rate setting
Parity	**C0**	No parity enabled
	C1	Odd parity
	C2	Even parity
	C?	Returns current parity setting
Data Bits	**D0**	Sets number of data bits to 7
	D1	Sets number of data bits to 8
	D?	Returns current data bit setting
Error Query	**E?**	Returns error status
Flush	**F0**	Flushes serial input buffer
	F1	Flushes serial output buffer
	F2	Flushes both input and output buffers
	F?	Returns last flush command executed

Command	Code	Description
HandshakeType	**G0**	Sets handskake to Rts/Cts (hardware)
	G1	Sets handskake to Xon/Xoff (software)
	G2	Sets handskake to none
	G?	Returns current handshake type selected
Input Query	**I?**	Returns number of characters in serial input queue
Status EOI	**K0**	Assert EOI with last terminator
	K1	EOI disabled
	K?	Returns current status EOI selected
Data EOI	**L0**	Assert EOI on serial terminator
	L1	EOI disabled
	L2	Assert EOI on last character in buffer
	L3	Assert EOI on terminator or last character
	L?	Returns current data EOI selected
Service Request Mask	**M0**	Disable bus SRQ
	M1	SRQ on Serial port 1 input
	M2	SRQ on Serial port 2 input
	M4	SRQ on Serial port 3 input
	M8	SRQ on Serial port 4 input
	M16	SRQ on Ready for next command
	M32	SRQ on Command error
	M128	SRQ on Memory low
	M?	Returns current SRQ mask
Handshake Control	**N0**	Automatic
	N1	Holdoff data
	N2	Accept data
	N3	Output 16 x baud clock on RTS line
	N?	Returns current handshake control setting
Output Query	**O?**	Returns number of characters in serial output queue
Port Select	**P1**	Select serial port 1
	P2	Select serial port 2

Command	Code	Description
	P3	Select serial port 3
	P4	Select serial port 4
	P?	Returns currently selected serial port
Break	**Q0**	Disable sending break
	Q1	Enable sending break
	Q?	Returns current break setting
Save	**S0**	Save default setup in non-volatile RAM
Configuration	**S1**	Save current setup in non-volatile RAM
	S?	Returns last save configuration command executed
Serial	**Tn**	Select n (0 - 255) as serial terminator
Terminator	**T?**	Returns currently selected serial terminator
Status	**U0**	Send command status on talk
	U1	Send serial port 1 status on talk
	U2	Send serial port 2 status on talk
	U3	Send serial port 3 status on talk
	U4	Send serial port 4 status on talk
	U?	Returns current status select setting
Version Query	**V?**	Returns the Serial488/4 firmware revision
Execute	**X**	Execute Command String
Bus Terminator	**Y0**	CR
	Y1	LF
	Y2	CR LF
	Y3	LF CR
	Y?	Returns current bus terminator setting
Memory Query	**Z?**	Returns the amount of serial data memory available

$00 0 NUL	$10 16 DLE	$20 32 SP 00	$30 48 0 16	$40 64 @ 00	$50 80 P 16	$60 96 ` SCG	$70 112 p SCG
$01 1 SOH GTL	$11 17 DC1 LLO	$21 33 ! 01	$31 49 1 17	$41 65 A 01	$51 81 Q 17	$61 97 a SCG	$71 113 q SCG
$02 2 STX	$12 18 DC2	$22 34 " 02	$32 50 2 18	$42 66 B 02	$52 82 R 18	$62 98 b SCG	$72 114 r SCG
$03 3 ETX	$13 19 DC3	$23 35 # 03	$33 51 3 19	$43 67 C 03	$53 83 S 19	$63 99 c SCG	$73 115 s SCG
$04 4 EOT SDC	$14 20 DC4 DCL	$24 36 $ 04	$34 52 4 20	$44 68 D 04	$54 84 T 20	$64 100 d SCG	$74 116 t SCG
$05 5 ENQ PPC	$15 21 NAK PPU	$25 37 % 05	$35 53 5 21	$45 69 E 05	$55 85 U 21	$65 101 e SCG	$75 117 u SCG
$06 6 ACK	$16 22 SYN	$26 38 & 06	$36 54 6 22	$46 70 F 06	$56 86 V 22	$66 102 f SCG	$76 118 v SCG
$07 7 BEL	$17 23 ETB	$27 39 ' 07	$37 55 7 23	$47 71 G 07	$57 87 W 23	$67 103 g SCG	$77 119 w SCG
$08 8 BS GET	$18 24 CAN SPE	$28 40 (08	$38 56 8 24	$48 72 H 08	$58 88 X 24	$68 104 h SCG	$78 120 x SCG
$09 9 HT TCT	$19 25 EM SPD	$29 41) 09	$39 57 9 25	$49 73 I 09	$59 89 Y 25	$69 105 i SCG	$79 121 y SCG
$0A 10 LF	$1A 26 SUB	$2A 42 * 10	$3A 58 : 26	$4A 74 J 10	$5A 90 Z 26	$6A 106 j SCG	$7A 122 z SCG
$0B 11 VT	$1B 27 ESC	$2B 43 + 11	$3B 59 ; 27	$4B 75 K 11	$5B 91 [27	$6B 107 k SCG	$7B 123 { SCG
$0C 12 FF	$1C 28 FS	$2C 44 , 12	$3C 60 < 28	$4C 76 L 12	$5C 92 \ 28	$6C 108 l SCG	$7C 124 | SCG
$0D 13 CR	$1D 29 GS	$2D 45 - 13	$3D 61 = 29	$4D 77 M 13	$5D 93] 29	$6D 109 m SCG	$7D 125 } SCG
$0E 14 SO	$1E 30 RS	$2E 46 . 14	$3E 62 > 30	$4E 78 N 14	$5E 94 ^ 30	$6E 110 n SCG	$7E 126 ~ SCG
$0F 15 SI	$1F 31 US	$2F 47 / 15	$3F 63 ? UNL	$4F 79 O 15	$5F 95 _ UNT	$6F 111 o SCG	$7F 127 DEL SCG
ACG	UCG	LAG		TAG		SCG	

ACG = Addressed Command Group
UCG = Universal Command Group
LAG = Listen Address Group
TAG = Talk Address Group
SCG = Secondary Command Group

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP