



USER GUIDE FOR IEEE488.2 COMPLIANT UNITS

State-of-the-Art Switching Solutions

CONTENTS

1. INTRODUCTION.....	9
2. FRONT PANEL.....	11
2.1. Front Panel LEDs	11
2.2. Keypad Control.....	12
2.2.1. LCL (local) Key.....	12
2.2.2. Connect Function.....	13
2.2.3. Disconnect Function.....	14
2.2.4. Verify Function.....	14
2.2.5. Store and Recall Switching Configurations.....	15
2.2.6. Clear Function.....	16
2.2.7. Cancel Key	16
2.3. Display Features.....	16
2.3.1. Power-up Display Screens.....	16
2.3.1.1. Keypad and Display Test Screens.....	17
2.3.1.2. Unit ID and Configuration Screen.....	18
2.3.2. Normal Screen after Power-Up	19
2.4. Display Menus.....	19
2.4.1. Normal Screen.....	20
2.4.2. Local Status Screen.....	21
2.4.3. Remote Interface Screen(s)	22
2.4.4. GPIB Interface Screen	23
2.4.5. Serial Port Screen	23
2.4.6. Operations Menu 1	24
2.4.6.1. Auto Interlock.....	24
2.4.6.2. Power On AutoRestore.....	25
2.4.6.3. Auto-Route Mode or Ganged Mode.....	25
2.4.6.4. Network ID	25
2.4.7. Operations Menu 2.....	26
2.4.7.1. Display Saver	26
2.4.7.2. Beep On Key-Press.....	27
2.4.7.3. Beep On Error	27
2.4.7.4. Enable DHCP	28
2.4.8. IP Address after Next Reset.....	28
2.4.9. Subnet Mask after Next Reset	29
2.4.10. Default Gateway after Next Reset.....	29
3. STO COMMANDS	31
4. COMMUNICATING WITH THE SYSTEM.....	33
4.1. Remote Resources.....	34
4.1.1. Example Resource Descriptors.....	34

4.1.2.	GUI and Commands Line Level of Control.....	35
5.	REMOTE CONTROL COMMANDS.....	37
5.1.	Changing Between Control Protocols.....	37
5.2.	Syntax Conventions.....	38
5.2.1.	CONnect command.....	39
5.2.2.	DISconnect command.....	39
5.2.3.	MAKe? command.....	39
5.2.4.	BREak? command.....	40
5.2.5.	QUERy? command.....	41
5.2.6.	GET? and SET commands.....	42
5.2.7.	LOCK and UNLock commands.....	43
5.2.8.	ETHernet? command.....	44
5.2.9.	*SAV and *RCL commands.....	44
5.2.10.	FORceclose command:.....	45
5.2.11.	RESEt command.....	45
5.2.12.	*IDN? command.....	46
5.2.13.	*TST? command.....	46
5.2.14.	*RST command.....	46
5.2.15.	*CLS command.....	47
5.2.16.	*STB? command.....	47
5.2.17.	*ESR? command.....	47
5.2.18.	*ESE command.....	47
5.2.19.	*ESE? command.....	48
5.2.20.	*SRE command.....	48
5.2.21.	*SRE? command.....	48
5.2.22.	*PSC command.....	48
5.2.23.	*PSC? command.....	48
5.2.24.	*OPC command.....	49
5.2.25.	*OPC? command.....	49
5.2.26.	*WAI command.....	49
6.	GET? AND SET VALUES.....	51
6.1.	GET? and SET Value Table.....	51
7.	REMOTE CONTROL INFORMATION.....	55
7.1.	Status Bite Register.....	55
7.1.1.	Status Bite Register Layout and Description.....	55
7.1.2.	The Status Byte Register (SBR).....	56
7.1.2.1.	Reading the Status Byte Register (SBR).....	57
7.1.3.	The Service Request Enable Register (SRER).....	57
7.1.4.	The Event Status Enabled Register (ESER).....	58
7.1.4.1.	The ESER under GPIB Control.....	58
7.1.5.	The Event Status Register (ESR).....	59
7.1.6.	Procedure for Recovering Errors.....	61
7.1.6.1.	Example Register Interaction.....	61
7.1.6.2.	The Fault Queue.....	62
7.1.7.	Register Function Summary.....	62
7.1.8.	Commands to Evaluate and Control Status (488.2).....	64

7.1.8.1.	Commands to Evaluate and Control Status (non 488.2).....	65
7.1.9.	Overlapped Processing.....	66
7.2.	System Switching Mode	67
7.2.1.	Auto-Route Mode	67
7.2.1.1.	Output End to End System.....	67
7.2.2.	Parallel Operation.....	68
7.2.3.	Ganged Mode.....	68
7.2.4.	Poles (Virtual Modules).....	69
8.	ERROR CODE LIST	71
9.	REMOTE CONTROL ASSEMBLY	73
9.1.	CPU Configurations.....	74
9.2.	Multiple CPU Installations	75
9.3.	XC710: Ethernet and Serial Interfaces (Remote).....	76
9.4.	XC710: Ethernet Interface	77
9.5.	C710-E10: Ethernet Interface	78
9.6.	C710-488: Ethernet and GPIB Interfaces.....	79
9.7.	C710-S3: Ethernet and Serial Interfaces.....	80
9.8.	C720-S3: Ethernet and Serial Interfaces.....	81
9.9.	Remote Control Interfaces.....	82
9.9.1.	The Ethernet (10BaseT & 10/100BaseT) Port Pin Assignment.....	82
9.9.2.	Default IP Address.....	82
9.9.2.1.	Changing the Ethernet Settings.....	82
9.9.2.2.	Changing the Ethernet Settings from the Front Panel	82
9.9.3.	The US-Link Interface and Service Port (C710, C720, XC710)	83
9.9.3.1.	The US-Link/Service Port Pin Assignment	83
9.9.4.	US-Link Rear Panel G2S400 Mainframe Interface Port (XC710).....	84
9.9.4.1.	The US-Link Port Pin Assignment.....	84
9.9.5.	GPIB Port Pin Assignment (C710-488, XC710)	85
9.9.6.	Setting the GPIB Address (C710-488, XC710)	86
9.9.6.1.	Changing the GPIB and SRQ Settings from the Front Panel.....	86
9.9.7.	Serial Port Pin Assignment: J6A (C710-S3)	86
9.9.8.	RS-485 Serial Port Pin Assignment: J7A (C710-S3).....	87
9.9.9.	Setting the RS-485 Serial Port: J7A (C710-S3)	87
9.9.10.	Serial Port Pin Assignment (C720-S3)	88
9.9.11.	Serial Port Pin Assignment: DE-9S (XC710)	89
9.9.12.	Changing the Serial Settings from the Front Panel.....	89
9.9.13.	Changing the Serial Mode.....	90
9.9.13.1.	Setting the Jumpers for C710-S3	90
9.9.13.2.	Setting the Jumpers for C720-S3	90
9.9.13.3.	Setting the Jumpers for XC710.....	90
9.9.14.	Setting the Serial Mode SW2 DIP Switch (C710, C720)	91

9.9.15.	Setting the Serial Control Mode SW1 DIP Switch (XC710)	91
9.9.16.	Setting the RS-485 Address	91
9.10.	Upgrading the CPU's Firmware Driver.....	93
9.10.1.	The Firmware Upgrade Package	93
9.10.1.1.	Upgrading Procedure:	94
10.	RECORD OF CHANGES	95

TECHNICAL SUPPORT

Phone +1 818-381-5111

Fax +1 818-252-4868

Email support@uswi.com

7671 North San Fernando Road
Burbank, CA 91505-1073 USA

1. Introduction

The User Guide for 488.2 Compliant Units documents the commands and protocols used in operating Universal Switching Corporation programmable switching systems, and the Remote Control Unit Assemblies (C710 and C720).

This document is maintained to reflect current products and does not describe older control protocols or Remote Control Unit Assembly C700.

For support of older US2, US3 and US4 control protocols, or Remote Control Unit Assembly C700, contact Customer Support.

2. Front Panel

This section describes the features of the front panel that include the LEDs, control keypad and the vacuum-fluorescent display. The explanation of each feature also provides a basis for their application and usage.

2.1. Front Panel LEDs

The mainframe front panel features status LED indicators located to the left of the control keypad. The LEDs are operational whenever the system is powered ON.

The STAT1 and STAT2 LEDs provide visual confirmation of command activity. The STAT1 LED flashes GREEN whenever command activity occurs.

The STAT2 LED is reserved for use on a project-to-project basis. It is not implemented on this project.

The ERR1 and ERR2 LEDs provide visual confirmation of errors/faults. The ERR1 LED displays RED whenever an error/fault occurs. The ERR1 LED remains RED until a non-error/fault command is issued.

The ERR2 LED is reserved for use on a project-by-project basis. It is not implemented on this project.

LED Indicator	Color	Status
STAT1	GREEN	Command activity
STAT2	GREEN	Not implemented
ERR1	RED	Error/Fault
ERR2	RED	Not implemented

2.2. Keypad Control

The control keypad has been designed for ease-of-use and functionality. All control keys are backlit with a green LED for easy reading in either direct sunlight or a completely dark room. To make operation of the unit even easier, only the valid keys are lit during any operation. Keys that are not illuminated are not valid.

The specific functions of the 24-position control pad may be divided into the following categories:

- OPERATORS: Connect, Disconnect, Verify, or Clear All
- CONTROL: Store, Recall, Cancel or Local Unlock
- INPUT: Numerical Keys, 0 through 9
- MISC.: Enter, Menu, and Arrow Keys

1	2	3	CLR	∧	LCL
4	5	6	<	MNU	>
7	8	9	CON	∨	STO
ENT	0	CNL	DIS	VFY	REC

When a button is pressed, it provides both audible (if enabled) and tactile feedback to the operator. You may control the unit locally from the manual control pad, or by means of any of the installed remote interfaces.

2.2.1. LCL (local) Key

The **LCL** key returns control to the front panel manual control pad if the controls have been locked out locally or remotely (by a security code). When pressing the "LCL" key, the user is prompted to enter the security code if the unit is locked. After entering the security code, the CPU unlocks the controls. The local keypad can also be locked to prevent unintended switch manipulation by pressing the "LCL" key and then entering a 4-digit security code.



NOTE: If the security code is unknown, press LCL, enter 7145, and press ENT on the keypad to unlock it.

2.2.2. Connect Function

The connect (**CON**) key is used to connect an input to an output. It is used with the numerical keys to enter an input number, output number combination. The example below connects port input 1 to output 2 (on module 1).

- Press connect: **CON**
- Enter, if required, the module slot number (two digits): **0 1**
- Enter the input port number (three digits): **0 0 1**
- Enter the output number (three digits): **0 0 2**
- Entries may be edited by using the arrow keys (up, down, left, or right) to select to the entry needing change. Once to the proper location, key in the desired number.
- To execute the command press the **ENT** key, or you may press the cancel **CNL** key to abort the command at any time.

After pressing the enter key, the unit verifies that the port numbers represents a valid entry. If not a valid entry, the System displays an error code and sounds a beeper. Other possible errors include attempting to connect an input to an already connected output or a switch malfunction.

After a valid entry, the unit makes the specified connection and confirms the closure by verifying with each of the embedded CPU's on the selected module that the command was received and confirmed. An OK is displayed if no errors were detected. If an error occurs while making a connection, the beeper sounds and an error message is displayed in place of the "OK".

2.2.3. Disconnect Function

The disconnect (**DIS**) key is used to disconnect (open) an input to output connection. To invoke, follow this sequence:

- Press disconnect (**DIS**)
- Enter the input number followed by the output number.
- Entries may be edited by using the arrow keys (up, down, left, or right) to select the entry to be edited. To execute the command press the **ENT** key, or you may press the cancel (**CNL**) key to abort the command at any time.

After pressing the enter key, the unit verifies that the input/output combination is a valid entry.

The unit then disconnects the specified combination and confirms it by verifying with the embedded CPUs on the modules that the command was received and confirmed. An OK is displayed.



NOTE: It is not considered an error to reconnect an already connected input/output combination or to disconnect an already disconnected output.


2.2.4. Verify Function

The verify (**VFY**) key may be used to verify the connection of a path through the unit without changing its status. The unit automatically performs this function each time a connect or disconnect is performed, but this command may be used to re-verify status.

To use the verify function, press the VFY key, enter the input and output combination and press the enter key. If the combination is connected, the unit displays VERIFY-CON as well as OK in the status field. If the combination is not connected, the unit displays "VERIFY-DIS OK" indicating that the pair is not connected. If the port is connected to another item, an error code is returned.


2.2.5. Store and Recall Switching Configurations

The System can store and recall up to 50 different configurations from the CPU's battery backed RAM. When a configuration is stored, the entire switching array configuration is saved into the storage memory location. This is useful when changing between configurations.

 **NOTE: Stored configurations may be lost if the main CPU's lithium battery jumper is removed, or if the battery life has exceeded (10 years).**


Storing a switching configuration is accomplished by the following key stroke sequence:

- Press the store (**STO**) key
- Enter the storage memory location (001 to 050) using the numerical keys. The arrow keys may be used to edit the number prior to the next step.
- Press the enter (**ENT**) key to store the configuration.

 **NOTE: When a configuration is stored in a location, it automatically overwrites any previously stored configuration in that memory location.**

The operation is confirmed with an OK in the status field of the display. The storage location 000 is reserved by the System to store the current configuration should power be lost to the unit. Upon powering the unit, the System re-configures to the last matrix setting if the Auto-restore feature is enabled in the setup menu.

To recall a configuration, press the recall (**REC**) key and enter the memory location (001 to 050) to be recalled. Press the **ENT** key to execute.

 **NOTE: When a configuration is recalled, the unit clears the entire matrix array, and then restores the saved configuration.**

2.2.6. Clear Function

Use the clear (**CLR**) key to clear the entire switching configuration. The clear function makes all signal paths go to the default positions (normal mode), disconnecting all previously connected backup connections. To clear the configuration, press the **CLR** key once, then press the ENTER key to confirm.

If the clear key was pressed by accident, you may press the cancel key (**CNL**) to abort the function. **Once the matrix is cleared, the previous configuration is lost unless previously stored in a memory location.**

2.2.7. Cancel Key

The cancel (**CNL**) key may be used at any time to abort a menu selection or command entered from the manual control pad.

2.3. Display Features

The front panel display is a high contrast vacuum fluorescent display constructed for rugged environments and long life. During normal operation (after self-test is complete), the display is divided into functional areas.



NOTE: The factory default for the display is to have it dim after about 10 minutes of no keypad entries. This is to assist in prolonging the life of the display.

2.3.1. Power-up Display Screens

This unit displays different screens during the power-up sequence of the system. Each of these screens is displayed for only 2-3 seconds, at which point the next screen is displayed.

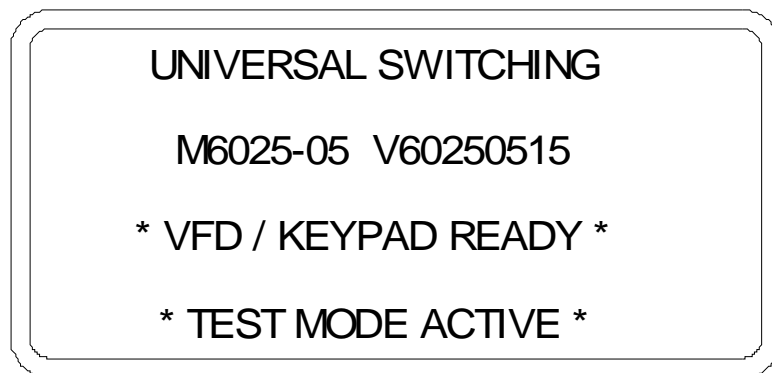
The screens are discussed in detail in the following paragraphs.

2.3.1.1. Keypad and Display Test Screens

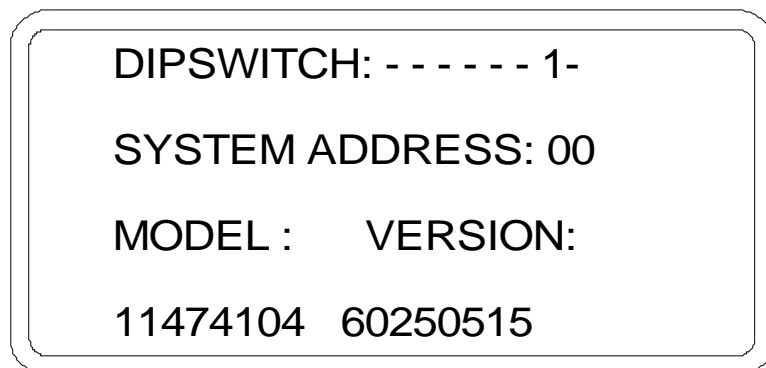
The first two (2) screens displayed are the CPU display and keypad screen test, and version display screen



NOTE: The following diagrams are examples. The actual numbers shown on display screens may differ.



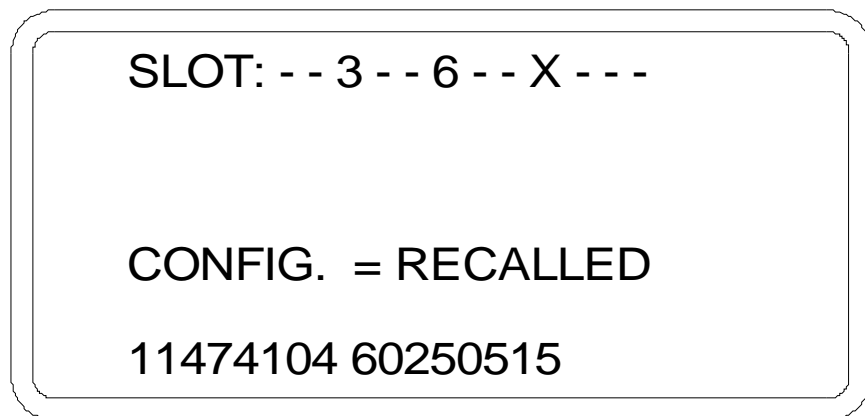
G01006a



G01007

2.3.1.2. Unit ID and Configuration Screen

This screen display comes up after the keypad CPU test only if the main CPU and interface module is installed and functioning. Proper module-to-module communication must be operational for this screen to appear.



G01006

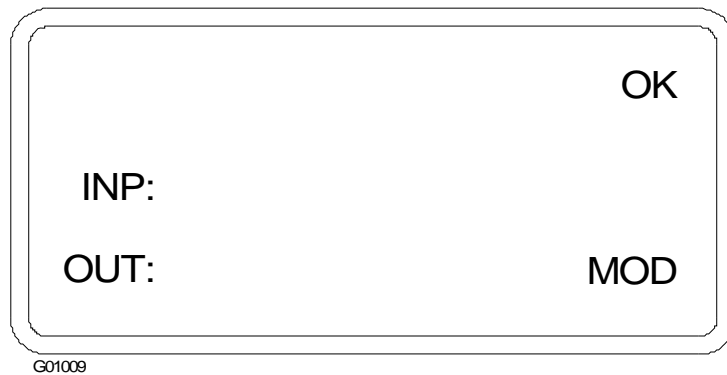
As shown above, the first line displays any modules installed within the mainframe, starting with Slot 01 (represented by the first "-"). Empty, non-operational, or non-recognized modules are shown with a dash (-). Modules that the CPU thinks should be there but are not there are designated with an "X". Installed modules are shown in order from top to bottom with their accompanying slot number. In this example, modules are installed in slots 3 and 6.

The next line of this screen indicates whether the unit is recalling "RECALL" the last configuration in use prior to being powered OFF, or if the configuration is "CLEARED" of all cross-point configuration closures. This option can be accessed via a menu selection at the front panel under the AUTO-RESTORE option (yes or no). The factory default is to have the unit auto-restore "RECALL" the last cross-point configuration.

The third line identifies the unit Series and Model numbers of the system, along with the installed Firmware version information. This line is a carry-over from the previous menu screen.

2.3.2. Normal Screen after Power-Up

Below is the normal screen displayed before the unit has processed any commands. The top line of the display is different if a GPIB type CPU is installed in place of a serial type. The top line for a GPIB type CPU displays "GPIB - OK".



2.4. Display Menus

The unit can be configured to perform various tasks by setting the unit up via menu driven selections. By pressing the MNU key on the front panel control keypad, the user can cycle through seven (7) different menu screens.

The "normal display" (shown in 5.4.1 below) is the default screen shown after the unit has finished its power ON tests and is awaiting commands.

2.4.1. Normal Screen

Below are examples of the normal screen during operation. The first line displays the last command attempted and the result. The second line displays the input port "INP". The third line displays the output port designation "OUT", and either a module number (MOD or if a tri-stage unit MID), a GANGED designation, or nothing, depending upon the specific system type.

```
CONNECT                OK
INP:  2
OUT:  8                MID: 3
```

```
DISCONNECT            OK
INP:  7
OUT:  2                MOD: 4
```

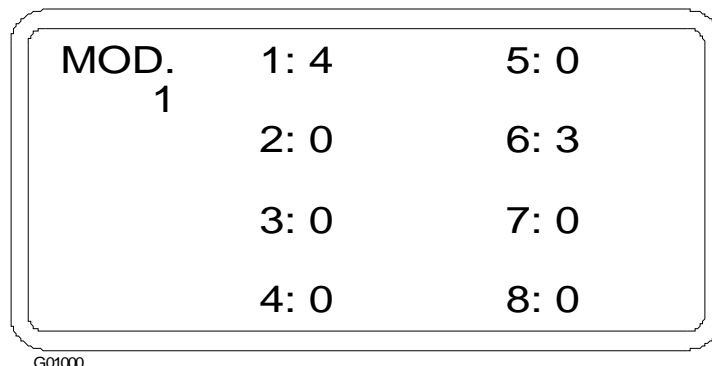
G00986

2.4.2. Local Status Screen

The user can view the cross-point connection status of the unit by pressing the MENU key once. This display lists each output in order and displays the connected input port (if any).

An example of the Local Status Screen is shown below. Outputs are displayed first, then a colon, then the inputs. Outputs are not connected if they do not have any input numbers associated with them. The screen example displays Module 1 with Output 1 connected to Input 4; Outputs 2 through 5 are not connected; Output 6 is connected to Input 3; and Outputs 7 and 8 are not connected.

Note: The display order is Output : Input



MOD.	1: 4	5: 0
1	2: 0	6: 3
	3: 0	7: 0
	4: 0	8: 0

G01000

When this screen is displayed, keypad control illuminates the MNU key and the arrows keys. To view more output connections, press the arrow keys. Continue pressing the arrow keys to view all of the output connections.

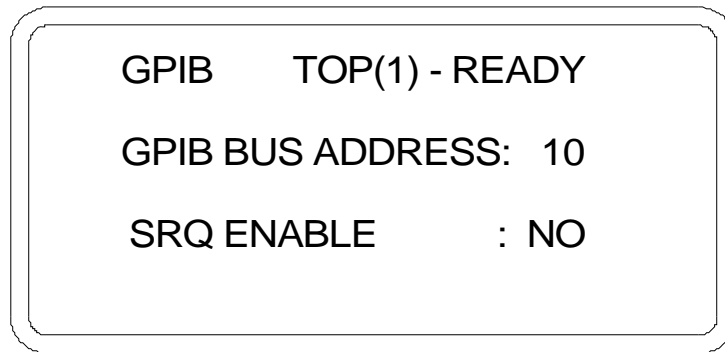


NOTE: This screen displays for about 10 seconds, and then returns to the “normal” screen, unless another key is pressed.

2.4.3. Remote Interface Screen(s)

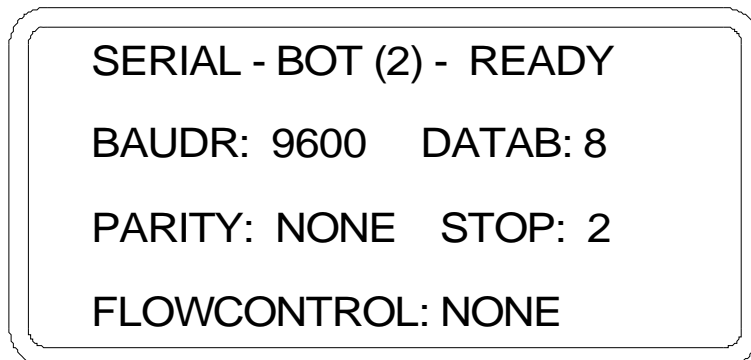
By pressing the MENU key twice (from the normal screen), the interface configuration menu appears on the display. This screen displays the parameters for any installed remote interface cards (IFC). Units have either one GPIB (IEEE-488) CPU, one serial (RS-232C / RS-422A / RS-485) CPU, one Ethernet CPU, one of each type, or two (2) of the same type depending upon system needs. On units with two (2) installed CPUs, pressing the menu key once more brings up the additional CPU screen. Additional interfaces installed in the field are displayed via this menu.

The display below shows a GPIB type CPU installed in the TOP (upper) position.



G00998

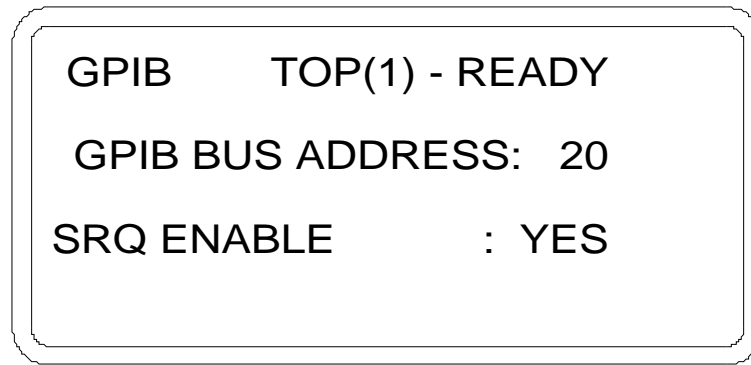
The example below shows a serial type CPU installed in the BOT (bottom) CPU position.



G00991

2.4.4. GPIB Interface Screen

With the interface configuration menu screen, the factory default GPIB port settings are displayed. The factory default settings are GPIB "ready", GPIB address 20, and SRQ enabled.

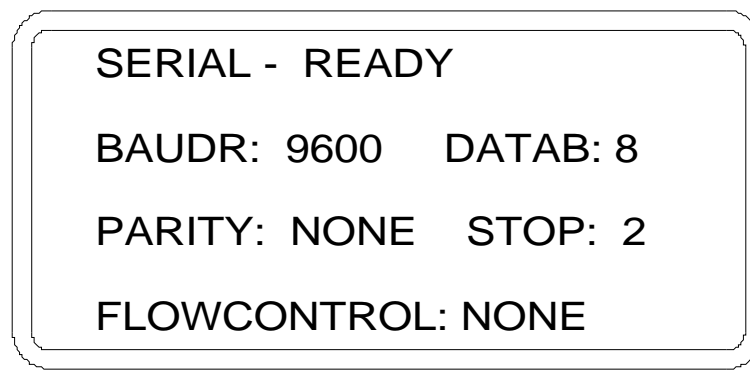


G00998a

To edit these parameters, move the blinking cursor with the arrow keys to the specific parameter you wish to change and press the ENT key to alter its value. When you have achieved the required setting, press the CNL key, or select different parameters to alter using the up/down arrow keys, and changing their value using the enter key. Once you leave this screen, the selected settings remain in non-volatile RAM until changed again.

2.4.5. Serial Port Screen

With the interface configuration menu displayed, the factory default serial port settings can also be displayed. The factory default serial port settings are 9600 baud, 8 data bits/char, no parity, 2 stop bits, and no handshaking.

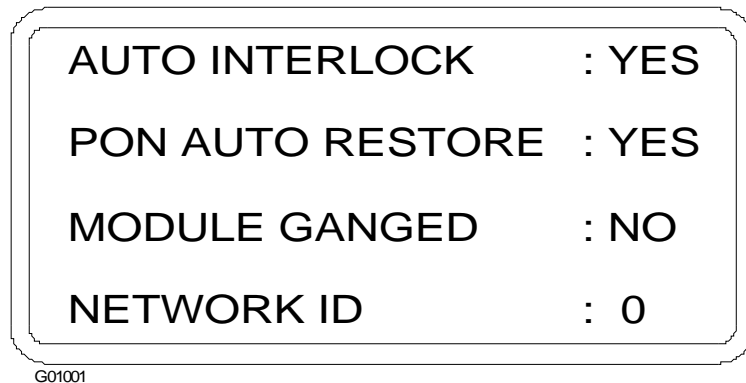


G00998b

To edit these parameters, move the blinking cursor with the arrow keys to the specific parameter you wish to change, and press the ENT key to alter its value. When you have changed the selected parameter, press the CNL key, or select different parameters to alter using the up/down arrow keys, changing their value using the enter key. Once you leave this screen, the selected settings remain in non-volatile RAM until changed again.

2.4.6. Operations Menu 1

By pressing the MNU key three times (from the normal screen), the Operation Menu 1 appears on the display. From this menu the items described in the following paragraphs may be changed.



2.4.6.1. Auto Interlock

Two (2) modes of "AUTO INTERLOCK" control are available. With the Auto-Interlocking feature enabled (YES), the unit automatically disconnects any input connected to the specified output port (or relay port) before making the new input connection.

With the input Auto-Interlocking feature disabled (NO), the user must send a disconnect command for the existing cross-point connection (or relay port connection) prior to connecting any new input port.



NOTE: The factory default is Auto-Interlock mode enabled (YES).

To configure the unit in the AUTO INTERLOCK mode, move the cursor to the line and press ENT to change the option. Once you have changed the setting, press the CNL key when done, or select the next parameter you want to change with the up/down arrow keys. Once you leave this screen, the selected settings remain in non-volatile RAM until changed again.

2.4.6.2. Power On AutoRestore

The Power On AutoRestore parameter may be enabled (YES) or disabled (NO). When enabled, the unit automatically restores the last configuration from before the unit was powered OFF (or power was lost to the unit). When disabled, all cross-points are automatically cleared on power ON.



NOTE: The factory default is Power On AutoRestore enabled (YES), so that the unit is returned to the last known cross-point state when powered ON.

To edit this parameter, move the blinking cursor on the display with the arrow keys to the specific interface needing changes, and then press the ENT key to view the options. Select the settings and press the CNL key when you are done, or continue selecting parameters with the up/down arrow keys. Once you leave this screen, the selected settings remain in non-volatile RAM until changed again.

2.4.6.3. Auto-Route Mode or Ganged Mode

Depending upon the system requirements, the system may be enabled in either "AUTO-ROUTE" mode or "GANGED" mode.

The AUTO-ROUTE mode routes the connection command without the need to address a specific module number. Tri-Stage™ or Output End-to-End systems are configured in AUTO-ROUTE mode.

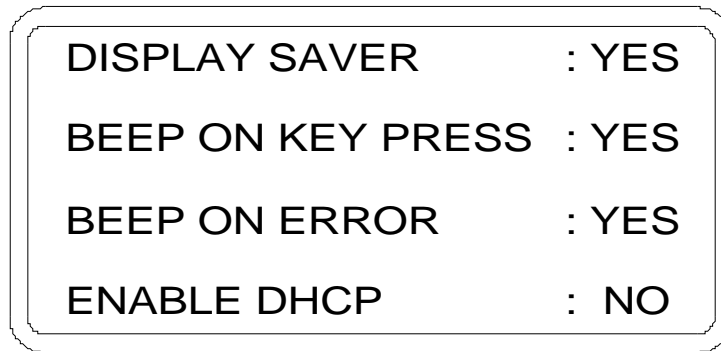
The GANGED mode may or may not require a specific module number depending on the selection of "virtual poles" (See 7.2.4.). Parallel systems may be configured in GANGED mode.

2.4.6.4. Network ID

The Network ID default number is "0". A number between 0 and 255 may be assigned. If there are two (2) systems with the same product code, the keypad can be used to assign different Network IDs. GET?/SET78 may be used to remotely set the Network ID.

2.4.7. Operations Menu 2

By pressing the MNU key four (4) times (from the normal screen) the Operation Menu 2 appears on the display. From this menu the items described in the following sections may be changed.



G01002

2.4.7.1. Display Saver

The unit incorporates a high contrast vacuum fluorescent display (VFD). To prolong the life and contrast of the display over many years, the display can be set to dim automatically when no entries have been made at the front panel controls for five (5) minutes or more. The display instantly goes to full brightness if any key on the front panel control keypad is pressed.



NOTE: The factory default is display saver enabled (YES).

To edit this parameter, move the blinking cursor on the display with the arrow keys to the specific interface needing change, then press the ENT key to view the options. Select the setting and press the CNL key when done, or continue selecting parameters with the up/down arrow keys. Once you leave this screen, the selected settings remain in non-volatile RAM until once again changed.

2.4.7.2. Beep On Key-Press

When the unit is equipped with a front panel control keypad and display option, each time a front panel key is pressed, a short beep sounds. This may not be desirable for all applications and may be disabled.



NOTE: The factory default is Beep On Key Press enabled (YES).

To edit this parameter, move the blinking cursor on the display with the arrow keys to the specific interface needing changes, then press the ENT key to view the options. Select the setting and press the CNL key when you are done, or continue selecting parameters with the up/down arrow keys. Once you leave this screen, the selected settings remain in non-volatile RAM until changed again.

2.4.7.3. Beep On Error

The unit has an internal beeper to indicate an error. This assists the user during firmware development and alerts the user of equipment failure. The unit also has a RED front panel LED labeled ERR1 that illuminates during an error condition.



NOTE: The factory default is Beep On Error enabled (YES).

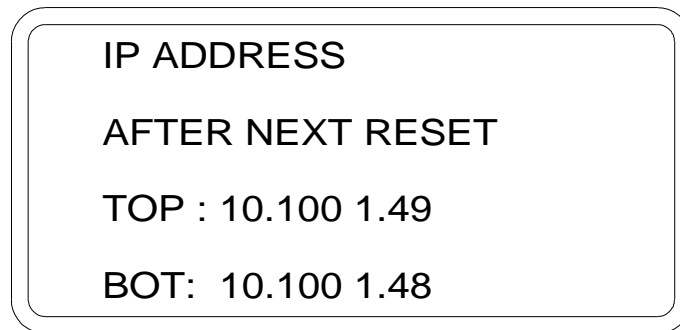
To edit this parameter, move the blinking cursor on the display with the arrow keys to the specific interface needing changes, then press the ENT key to view the options. Select the setting and press the CNL key when you are done, or continue selecting parameters with the up/down arrow keys. Once you leave this screen, the selected settings remain in non-volatile RAM until changed again.

2.4.7.4. Enable DHCP

The Dynamic Host Configuration Protocol (DHCP) is an Internet protocol for automating the configuration of CPUs that use TCP/IP stack. If enabled, after the next reset of the system resets the TCP/IP stack, the CPU attempts to obtain its IP address, subnet mask and default gateway from a DHCP server instead of using the manually set values. Application with two (2) CPU bays display both the top and bottom addresses.

2.4.8. IP Address after Next Reset

By pressing the MNU key five times (from the normal screen) the IP Address after Next Reset appears on the display. This screen sets the internet protocol parameters after next reset. Applications with two (2) CPU bays display top CPU and bottom CPU addresses.



G01003a

The RESET command performs a system reset including a reset of the TCP/IP stack. Three (3) ways of resetting the system are available to user:

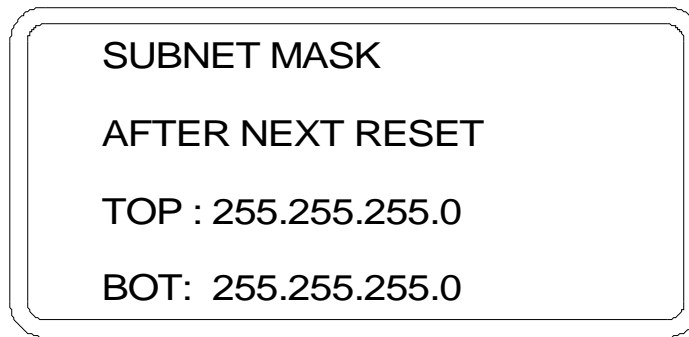
- by power-on the system or by the reset button on the main CPU or by a RESet command.
- a reset of the TCP stack triggered by a FORceclose command.
- from the front panel keypad using STOrE 224 command or from the Telnet port using an exclamation point " ! ".



NOTE: For additional information and samples about the RESet and FORceclose commands, refer to the REMOTE CONTROL COMMANDS section.

2.4.9. Subnet Mask after Next Reset

By pressing the MNU key six (6) times (from the normal screen) the Subnet Mask after Next Reset appears on the display. This screen sets the indicated parameters after next reset. Applications with two (2) CPU bays display both top and bottom addresses.

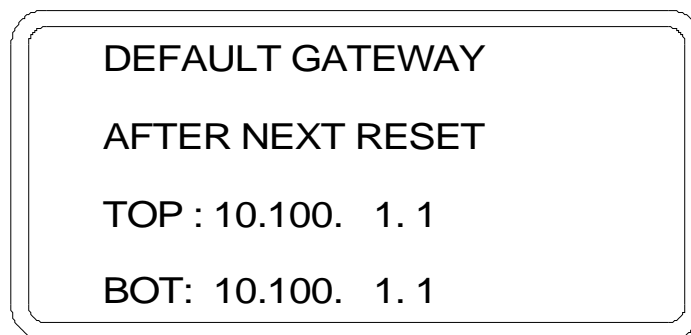


G01004

See the IP Address after Reset section for additional instructions on how to reset the system.

2.4.10. Default Gateway after Next Reset

By pressing the MNU key seven (7) times (from the normal screen) the Default Gateway after Next Reset appears on the display. This screen sets the indicated parameters after next reset. Applications with two (2) CPU bays display both top and bottom addresses.



G01005

See the IP Address after Reset section for additional instructions on how to reset the system.

3. STO Commands

The following STO commands executed from the front panel are currently applicable:

- **STO 213**

Press the STO button on the front panel keypad and 213 to perform a self-test or verify module(s) installation.

- **STO222**

Press the STO button on the front panel keypad and 222 to set the switching system to 488.2 protocol.

- **STO 223**

Press the STO button on the front panel keypad and 223 to set the system to backwards compatibility.

- **STO 224**

Press the STO button on the front panel keypad and 224 to perform a Forceclose.

- **STO 225**

Press the STO button on the front panel keypad and 225 to display system CPU state (dual or single).

- **STO 226**

Press the STO button on the front panel keypad and 226 to POWER OFF the second power supply.

4. Communicating with the System

This system contains the C710 or C720 type of CPU offered by Universal Switching Corporation. This type of CPU offers the flexibility of remotely controlling the system via the following interfaces:

- **IEEE-488.2.**

Also known as the General – Purpose Interface Bus (GPIB). This type of parallel control is very popular in automated test equipment (ATE) applications. It is high speed and provides data handshaking.

- **RS-232C/422A/485.**

Also known as serial interface were originally developed for modems and data terminals. These types of interfaces are now commonly used for controlling a wide range of hardware configurations.

- **TCP/IP (Ethernet).**

The C710 CPU supports 10Base-T Ethernet networks. The C720 CPU supports 10Base-T and 10/100Base-T Ethernet networks.

The GPIB interface complies with IEEE 488.2 protocols or one of its derivatives in the case of the serial and TCP/IP interface. These interfaces allow computers to access the system's operating features providing remote access to configure, control and receive operating status of the switching system.

The exact type of C710 or C720 controller CPU/Interface used with this system is dependent on the user's system requirements.



NOTE: Please refer to REMOTE INTERFACE PORTS section for additional information on these remote control interfaces.

4.1. Remote Resources

The most versatile way to communicate is through Virtual Instrument System Architecture (VISA) library. Typically, this API contains a series of functions designed for interface-independent, device-independent and platform-independent access to instruments connected to a "host". To use, the calling program calls the function `viOpen()` with a "resource name" as one of the arguments. If successful, the function returns a handle to a VISA "session" which can be used to call the other VISA functions.

A download of VISA usually contains an application for accessing the API from either a command line or a GUI. This can also be used to access the instrument. Don't forget to append the "linefeed" character (`'\n'` or `0x0a`) as the last character.

The VISA API requires the resource descriptor to figure out how to communicate with the instrument. Some tasks of the VISA API are explained below:

- It figures out which calls to make to which interface (socket, GPIB, serial or future interfaces)
- It knows the capabilities of each interface. For example, the `viReadSTB` function "serial polls" a GPIB instrument, but send `"*STB?\n"` and read the response on serial or TCP/IP. It knows that serial has a baud rate and that GPIB has, among other lines, attention and trigger.
- Buffers characters, manages time-out.
- Multiple sessions can be opened to each instrument. It automatically handles contention issues (including locks).

4.1.1. Example Resource Descriptors

Examples of resource descriptors are:

- **GPIB0::20::INSTR**
The instrument at address 20 on GPIB interface card 0.
- **ASRL1::INSTR**
The instrument connected to serial port 1.
- **TCPIP::10.100.1.49::7145::SOCKET**
The instrument at the indicated IP address listening in on port 7145.

4.1.2. GUI and Commands Line Level of Control

At the GUI and command line levels, there are several ways to control the instrument. Probably the most familiar is HyperTerminal from Hilgraeve. (the one that is shipped with Windows). It works with serial as long as the user appends the linefeeds.

Most operating systems include some form of Telnet. It also works as long as the user changes the port from Telnet's default 23 to one of the C710 or C720's ports (start with 7145).

For GPIB, National Instruments includes a command line utility (IBIC) and a GUI utility (Communicator). Do not forget the terminating linefeed and to change the read-termination method to linefeed.

5. Remote Control Commands

The system can be controlled through any of the remote interfaces installed on the Series C710 or C720 plug-in controller. The control commands are the same whether GPIB, Ethernet or Serial.

Universal Switching products utilize a 488.2 compliant control protocol. For support of older US2, US3 and US4 control protocols, contact Customer Support. This manual only explains the 488.2 control protocol.

5.1. Changing Between Control Protocols

The 488.2 protocols are the factory default and are enabled when a system is delivered from the factory. **The user may change the protocol to the older USx type, but it is recommended that any software developed for new applications utilize 488.2 compliant protocols.** Only the 488.2 protocols as used on Universal Switching Corporation Products are detailed in this manual. For older protocols, contact the factory.

Use the following command to change the switching system to the 488.2 compatibility (factory default):

- **ST 222**

or by pressing the STO button on the front panel keypad and STOrE location 222 to set the switching system to 488.2 compatible protocol.

Use the following command to change the switching system to backwards compatibility (US2, US3 or US4):

- **SET 29, 0**

or by pressing the STO button on the front panel keypad and STOrE location 223 to set the switching system to backwards compatible protocol.

5.2. Syntax Conventions

The following outlines the proper syntax of the commands for the 488.2 protocol:

- Words in the format XXXxxxx represent keywords. The upper case part is required; the lower case part is optional. Any combination of upper and lower case characters is acceptable (example: CONnect - the CON is required and minimally acceptable).
- Italicized words represent values (example: *output*)
- () represents optional words or combinations, | means that a choice can be made among the possibilities.
- Numbers must be whole numbers in base 10. Leading zeros are ignored. As a departure from the 488.2 protocol, decimal numbers, scientific notation and arithmetic expressions are not accepted.
- Where spaces are shown, at least one space must appear unless there's a comma. An arbitrary number of spaces can be used in addition to the required one.
- A "compound" command may be formed by placing a ; between commands.
- A command must be terminated with the ASCII linefeed character (0x0a, 10 or Ctrl-J).
- Responses are sent only in reply to queries (commands which end with "?").

The switching mode for the system may be enabled in AUTO-ROUTE or GANGED mode. AUTO-ROUTE mode does not need the "MODule" parameter in the command syntax for the following commands:

- **CONnect**
- **DISconnect**
- **QUERy**



NOTE: For additional information about the functionality of MODES, please refer to the REMOTE CONTROL INFORMATION section.

5.2.1. CONnect command

This command is used to make a connection within the system.
CONnect (FRom) (OUtput) *output*, (TO) (INput) *input* [, (ON)
(MOdule) *module* | Any)

examples:

- CONnect from output 1, to input 1
- CON 1,1 (identical to the above)
- CON 1,1,any

5.2.2. DISconnect command

This command is used to perform disconnections for crosspoints (switchpoints) which are currently connected.
DISconnect ((FRom) (OUtput) *output* [, (TO) (INput) *input*] [, (ON)
(MOdule) *module* | Any)

examples:

- DISconnect from output 1, to input 1
- DISconnect all (disconnects all outputs to inputs)
- DISconnect output 1
- DISconnect 1,1

5.2.3. MAKE? command

(Rev. Code 0507140000 and up)

The MAKE? query is identical to the connect command except that it returns an integer response:

- zero: connect completed successfully
- non-zero: connect failed due to an execution error or a fault.

This is a departure from other queries which do not return a response if there was an execution error or a fault.



NOTE: Please refer to the BREak? command for an example and additional information on the responses to this command.

5.2.4. BREak? command

(Rev. Code 0507140000 and up)

The BREak? query is identical to the disconnect command except that it returns an integer response:

- zero: disconnect completed successfully
- non-zero: disconnect failed due to an execution error or a fault.

This is a departure from other queries which do not return a response if there was an execution error or a fault.

If the MAKE? or BREak? queries are used in a compound command and there's an execution error or a fault, execution continues and a response is returned. This is a departure from the other commands which cause the rest of a compound command to be discarded if there's an execution error or a fault.

Examples:

Connect output 1 input 1; disconnect output 1 input 2; disconnect output 1 input 1

Connects output 1 to input 1 but when it attempts to disconnect from output 1 input 2, the ensuing execution error stops the rest of the command from executing and output 1 remains connected to input 1.

Make? output 1 input 1; break? output 1 input 2; break? output 1 input 1

Return:

0, 4, 0

The MAKE? and BREak? queries alleviate a lot of the awkwardness that results from attempting to implement IEEE Standard 488.2 on serial and TCP/IP. Much of the standard is conceived around GPIB's ability to request attention from its controller. Lacking that ability, users of TCP/IP and serial are left with unsatisfactory alternatives. For control over GPIB, these queries probably do not offer any advantage over the CONnect and DISconnect commands except for their suitability to interface independent applications.

A non-zero result indicates the need to check systematically the various status registers to determine the exact problem. It does not indicate the nature of the problem. The actual value returned is at best meaningless and at worst, misleading. A non-zero result is of no value in determining what the problem is only in determining that there was a problem.

Reception of the response can be interpreted by the controller as an indication that the switch has completed the previous operation and is ready to accept the next command.

5.2.5. QUERy? command

This command is used to query the switch module whether it is a single relay, or all of the relays.

QUERy? (FRom) (OUput) *output* [, (TO) (INput) *input*] [, (ON) (MOdule) *module* | Any]

The response is the input to which the output is connected or 0 if it's disconnected. If an input is specified and the output is not connected or connected to a different input, it is flagged as an execution error. Queries regarding one output are checked against the module's response to a Verify command. Queries with the ALL parameter are "verified" against the system's current configuration memory.

Command examples:

- QUE? 1 (the response would be the input number that is connected to the specified output port)
- QUE? 1,1
- QUE? all (the response would be the number of outputs being reported on, followed by the inputs that are connected starting with output 1 and ending with the last output of the unit)
- QUE? all , ,1 (returns connected outputs on module 1)
- QUE? all , ,1; QUE? all , ,2; QUE? all , ,3 (returns connected outputs on three installed modules)

The following is an example response to the "que? all" command, which displays a single string from the output to the input connection.

que? all response is 1,2

Interpreted as:

1 output,
output 1 is connected to input 2

5.2.6. GET? and SET commands

These commands allow the user to get and/or set various properties and values of the system.

GET? *property*

SET *property, value*

Various "properties" can be "set" with a value; Some "properties" "value" can be read. A separate table is updated as properties are added (and the list grows!). Note that each one behaves differently: consult the GET? and SET? table (Section 6.1).

examples:

- set 21, 1 (sets auto interlock on)
- get? 21 (gets current auto interlock status)
- get? 1 (gets number of outputs)
- set 1,1 (causes an execution error 12)
- set 29,0 (changes system to "backwards compatibility mode")

5.2.7. LOCK and UNLOCK commands

These commands allow the user to lock and/or unlock the front panel keypad. A four (4) digit code is entered for security.

LOCK *nnnn* any four (4) digit code

UNLOCK

Use the lock and unlock commands as necessary. The code is not required to unlock the keypad remotely. Sending UNLOCK returns the front panel status to unlocked.

examples:

- LOCK 2121 locks front panel keypad with code 2121
- UNLOCK unlocks front panel keypad

5.2.8. EThernet? command

This command is a query that returns the MAC address of the processor board. It is a unique alphanumeric string of hexadecimal numerals arranged as six (6) octets separated by colons.

example:

ETH?

Might return a string like this (without spaces):

1 2 : 3 4 : 5 6 : 7 8 : 9 a : b c



NOTE: Each controller module has a unique MAC address. If the controller module is changed or replaced, the MAC address changes.

5.2.9. *SAV and *RCL commands

These commands allow the user to save and recall complete switchpoint configurations in non-volatile memory locations in the system.

**SAV memory*

**RCL memory*

The user can save and recall complete switchpoint configurations. As a departure from 488.2, however, 0 is not acceptable. The highest number that is acceptable can be found by the GET? 28 query.

examples:

- *SAV 99 (saves current switchpoint configuration to memory location 99)
- *RCL 12 (recalls switchpoint configuration saved in memory location 12)

5.2.10. FORceclose command:

Force close command closes all TCP/IP sessions. New sessions can be established immediately.

example:

- FORceclose (closes all open TCP/IP ports)

From the front panel keypad, this command can be executed using STOrE 224. Also, an exclamation point (!) followed by a linefeed (Ctrl-J if your Telnet program is not set up to automatically send it) on the Telnet port (23) can be used to force close the ports.

Background: *Unlike web sites that must allow thousands (or sometimes millions) of simultaneous connections to a "port", the C710 or C720 allows only one connection to each port. This is the virtual equivalent of the physical cable that connects to the serial or GPIB ports. The use of inactivity time-outs is not practical for this application.*

When the host is done with its session, it tells the other device so that it can close its port and make it available to another session. In certain conditions, it's possible for the host to close its session without the device finding out. (Examples: an Ethernet hub or cable is disconnected in mid-session and the host terminates its session.) In these conditions, the FORceclose command can be used to make the port available again.

5.2.11. RESet command

This command performs a system reset.

RESet

Performs a system reset. If SET 30, 0 was performed prior, the system resets with factory defaults, otherwise, it performs a "normal" reset. If Autorestore is TRUE (Set/Get? 22), the connections are restored.

5.2.12. ***IDN? command**

This command returns an identification string.

*IDN?

Returns an identification string of the format:

“Universal Switching,pppppppp,s,rrrrrrrr\n”

where pppppppp is the model code, s is 0 (in lieu of a serial number) and rrrrrrrr is the revision code. The model code and the revision code are extremely important for resolving support issues. Note that the revision code is subject to change with any firmware updates/revisions.

5.2.13. ***TST? command**

This command performs a system self-test.

*TST?

The response is 0 if it completed successfully. A non-zero response indicates the number of modules that were not in their proper slots. Allow several seconds for the test to complete. As the result of the self-test:

Routing to modules that were removed or have stopped responding since the last reset or self-test generates an execution error 47 instead of generating a fault.

Modules that were inserted since the last restart or self-test now are used.

Interface setting made via the keypad or through SETs take effect.

5.2.14. ***RST command**

This command is the switchpoint reset command.

*RST

This command clears all the switchpoint routings in all the modules. No other status registers are affected.

5.2.15. *CLS command

This command clears registers as described below.

*CLS

This clears the interface's Last Error Registers, Event Status Register and the Status Byte Register (except for MAV, FLT and PSFLT) but not its queued responses. It has no affect on the other interfaces. As a departure from the standard, the Fault Queue is not affected. As a result, after the command, MAV, FLT and PSFLT reflects the status of their underlying conditions.

5.2.16. *STB? command

This command returns the contents of the Status Byte Register.

*STB?

Returns the contents of the Status Byte Register. Note that bit 6 is the MSS bit and not RQS and MSS is cleared by the operation. (This is in contrast to the way GPIB serial polls clear RQS).

5.2.17. *ESR? command

This command returns the contents of the Event Status Register.

*ESR?

Returns the contents of the Event Status Register (ESR). The register is cleared after the read. See below for details of the relationship between the ESR and the Last Error Registers.

5.2.18. *ESE command

This command sets the value of the Events Status Register.

*ESE *n*

This sets the value *n* (*n* can be from 0 to 255) in the Event Status Enable Register (ESER). See below.

5.2.19. *ESE? command

This command reads the Event Status Enable Register (ESER).

*ESE?

Returns the current value of the ESER.

5.2.20. *SRE command

This command sets the value of the Service Request Enable Register.

*SRE *n*

Sets the value *n* (*n* can be from 0 to 255) in the Service Request Enable Register (SRER).

5.2.21. *SRE? command

This command reads the value of the Service Request Enable Register.

*SRE?

Returns the current value of the SRER.

5.2.22. *PSC command

This command sets the value of the PSC Register to *n*.

*PSC *n*

Set the value *n* (*n* can be 0 or non-zero) in the PSC register.

5.2.23. *PSC? command

Sending this command reads the value of the PSC register.

*PSC?

Returns the current value of the PSC register as value of zero or 1.

5.2.24. *OPC command

This command sets the OPC bit in the ESR.

*OPC

Sets the OPC bit in the ESR. This causes the SBR to be updated. The bit can only be cleared by *CLS or ESR?.

5.2.25. *OPC? command

*OPC?

Immediately returns 0.

5.2.26. *WAI command

This command is only included for compatibility to the 488.2 standard.

*WAI

Since the device does not support “over-lapped” commands, this command does nothing. It is included for compatibility with the standard.

6. GET? and SET Values

6.1. GET? and SET Value Table

Below is a table of the available properties that can be read and changed. Note that each one behaves differently so consult the table.

Code	Title	Access	Function
1	Outputs	RO	Largest output
2	Inputs	RO	Largest input
3	Modules	RO	number of modules for which the system is configured
4	Last query error	RO	last query error or zero (see note 1)
5	Slot that the module is in	RW	set: (see Note 2) Get: returns the slot where the module should be
6	Inputs on module	RW	set: (see Note 2) Get: returns the number of inputs on the module
7	Outputs on module	RW	set: (see Note 2) Get: returns the number of outputs on the module
8	Slots	RO	number of slots in the system
9	Module in slot	RW	set: (see Note 2) Get: returns module number in slot or 0
10	Module ID in slot	RW	set: (see Note 2) Get: returns ID of module in slot or 0
11	(Project)		
12	(Project)		
13	(Project)		
14	Dual State	RO	1 is alone, 2 is primary, 3 is secondary
15	Fault FIFO	RO	Returns next fault or zero if empty, see note 3
16	Last Execution Error	RO	last error or zero (see note 1)
17	(Project)		
18	(Project)		
19	(Project)		
20	Ganged Modules	RW	0 is un-ganged, 1 is ganged
21	Auto Interlock	RW	auto interlock on (1) or off (0)
22	Auto Restore	RW	auto restore on(1) off(0)
23	Debug mode	RW	1: SPI traffic appears on Telnet port, 0: no traffic on port
24	Beep on Error	RW	beep on error (1) or silent (0) default is 1
25	Inactivity Timeout	RW	TCP/IP port inactivity timeout in seconds. Values: 0 - disabled (i.e. port remains open until closed by host or the FORceclose command) 1 - 28800 (8 hours) - number of seconds of inactivity until the port is closed. Values outside the range are coerced to 28800. The default value is 0 (disabled).
26	Power Supply 1	RO	Last status returned from PS1
27	Power Supply 2	RO	Last status returned from PS2
28	Memories	RO	number of "memories" available to *SAV or *RCL
29	Use 488.2	RW	use backwards-compatibility mode (0)
30	System Integrity	RW	if anything other than 21930, system restores factory defaults on next reset
31	System cleared	RO	1 if system restored defaults on last restart
32	Last Command Error		last error or zero (see note 1)
33	"This" CPU's Current IP address (MS)	RO	nnn of nnn.XXX.XXX.XXX for current IP address

Code	Title	Access	Function
34	"This" CPU's Current IP address	RO	nnn of XXX.nnn.XXX.XXX for current IP address
35	"This" CPU's Current IP address	RO	nnn of XXX.XXX.nnn.XXX for current IP address
36	"This" CPU's Current IP address (LS)	RO	nnn of XXX.XXX.XXX.nnn for current IP address
37	"This" CPU's Current Subnet Mask (MS)	RO	nnn of nnn.XXX.XXX.XXX for current subnet mask
38	"This" CPU's current Subnet Mask	RO	nnn of XXX.nnn.XXX.XXX for current subnet mask
39	"This" CPU's current Subnet Mask	RO	nnn of XXX.XXX.nnn.XXX for current subnet mask
40	"This" CPU's current Subnet Mask (LS)	RO	nnn of XXX.XXX.XXX.nnn for current subnet mask
41	CPU-1 IP Address after reset (MS)	RW	nnn of nnn.XXX.XXX.XXX for IP address after next reset
42	CPU-1 IP Address after reset	RW	nnn of XXX.nnn.XXX.XXX for IP address after next reset
43	CPU-1 IP Address after reset	RW	nnn of XXX.XXX.nnn.XXX for IP address after next reset
44	CPU-1 IP Address after reset (LS)	RW	nnn of XXX.XXX.XXX.nnn for IP address after next reset
45	CPU-1 Subnet Mask after reset (MS)	RW	nnn of nnn.XXX.XXX.XXX for subnet mask after next reset
46	CPU-1 Subnet Mask after reset	RW	nnn of XXX.nnn.XXX.XXX for subnet mask after next reset
47	CPU-1 Subnet Mask after reset	RW	nnn of XXX.XXX.nnn.XXX for subnet mask after next reset
48	CPU-1 Subnet Mask after reset (LS)	RW	nnn of XXX.XXX.XXX.nnn for subnet mask after next reset
49	"This" CPU's Current gateway (MS)	RO	nnn of nnn.XXX.XXX.XXX for current gateway address
50	"This" CPU's Current gateway address	RO	nnn of XXX.nnn.XXX.XXX for current gateway address
51	"This" CPU's Current gateway address	RO	nnn of XXX.XXX.nnn.XXX for current gateway address
52	"This" CPU's Current gateway (LS)	RO	nnn of XXX.XXX.XXX.nnn for current gateway address
53	CPU-1 Gateway after reset (MS)	RW	nnn of nnn.XXX.XXX.XXX for gateway mask after next reset
54	CPU-1 Gateway after reset	RW	nnn of XXX.nnn.XXX.XXX for gateway mask after next reset
55	CPU-1 Gateway after reset	RW	nnn of XXX.XXX.nnn.XXX for gateway mask after next reset
56	CPU-1 Gateway after reset (LS)	RW	nnn of XXX.XXX.XXX.nnn for gateway mask after next reset
57	CPU-2 IP Address after reset (MS)	RW	nnn of nnn.XXX.XXX.XXX for IP address after next reset
58	CPU-2 IP Address after reset	RW	nnn of XXX.nnn.XXX.XXX for IP address after next reset
59	CPU-2 IP Address after reset	RW	nnn of XXX.XXX.nnn.XXX for IP address after next reset
60	CPU-2 IP Address after reset (LS)	RW	nnn of XXX.XXX.XXX.nnn for IP address after next reset
61	CPU-1 serial port settings	RW	Weighted value
62	CPU-2 serial port settings	RW	Weighted value
63	CPU-2 Subnet Mask after reset (MS)	RW	nnn of nnn.XXX.XXX.XXX for subnet mask after next reset
64	CPU-2 Subnet Mask after reset	RW	nnn of XXX.nnn.XXX.XXX for subnet mask after next reset
65	CPU-2 Subnet Mask after reset	RW	nnn of XXX.XXX.nnn.XXX for subnet mask after next reset
66	CPU-2 Subnet Mask after reset (LS)	RW	nnn of XXX.XXX.XXX.nnn for subnet mask after next reset
67	CPU-1 GPIB port settings	RW	Weighted value
68	CPU-2 GPIB port settings	RW	Weighted value
69	CPU-1 has serial port installed	RO	0 is no, 1 is yes

Code	Title	Access	Function
70	CPU-2 has serial port installed	RO	0 is no, 1 is yes
71	CPU-2 Gateway after reset (MS)	RW	nnn of nnn.XXX.XXX.XXX for gateway mask after next reset
72	CPU-2 Gateway after reset	RW	nnn of XXX.nnn.XXX.XXX for gateway mask after next reset
73	CPU-2 Gateway after reset	RW	nnn of XXX.XXX.nnn.XXX for gateway mask after next reset
74	CPU-2 Gateway after reset (LS)	RW	nnn of XXX.XXX.XXX.nnn for gateway mask after next reset
75	CPU-1 has GPIB port installed	RO	0 is no, 1 is yes
76	CPU-2 has GPIB port installed	RO	0 is no, 1 is yes
77	DHCP Client (Enable/Disable)	RW	0 is Disable (default), 1 is Enable
78	Network ID Number	RW	A number between 0 and 255. Default is "0". See note 6.
79	Power Supply Monitoring Disabling	RW	0, 1, 2 or 3. Where: 0 – both power supplies are monitored. 1 – power supply 1 is not monitored. 2 – power supply 2 is not monitored. 3 – no power supply is being monitored. See note 7.
80	Dual Fault Condition Mask	RW	A number between 0 and 15. Default is "0" for backwards compatible but a different factory default can be specified for each project. See note 8.
83	The Pole to which the module belongs	RW	Set: (See Note 2) Get: for a valid Pole, returns -1. For a module, returns the pole to which it belongs. For the case where Pole 0 is invalid, returns 0.

Note 1: As long as the register's corresponding bit in SESR is set, the GET? Command returns the last error. Once the SESR has been cleared (by the *ESR? Query), it returns the last error on the next call then clear itself.

Note 2: The SET determines which module or slot the next GET? references. Example: to find out which slots module 1 & 2 are in, the following may be used:

set 5,1; get? 5; set 5,2; get? 5

Note 3: This is shared between all the interfaces. The Fault bit in the SBR clears once it's empty.

Note 4: The "Access" column designations are defined as:

- RW = Read and Write
- RO = Read Only

Note 5: Some systems have the capability to have two CPU's installed. Those with only one CPU slot (or built-in like the G2S400CE mainframe) only have "CPU-1" installed.

Note 6: When a CPU receives a UDP packet, either addressed to it or as a broadcast on port 5417 of at least one character, the CPU responds with a UDP packet with the format "PROJECT_ xxx" where PROJECT_ is the eight character product code and xxx is the Network ID, with any leading zeros (Example: 11888101007). The application for this is when DHCP is used. By sending out a broadcast packet addressed to port 5417, one can discover which IP addresses were assigned to what products by their DHCP server(s). If there are two systems with the same product code, the keypad can be used to assign different Network IDs.

Note 7: Refer to the MAINFRAME section of the system manual to see additional information on power supply monitoring.

Note*8: The following is effective on all revisions greater than 0502221100 regardless of how many processors are installed.

- GET?/SET register 80 is defined as the "Dual Fault Condition Mask." It is readable and writeable. The factory default is 0 but a different factory default can be specified for each project.
- When the register is set to 0, it is "backwards compatible" and no difference is apparent from previous versions.
- The register is a 4 bit binary mask to enable new functionality. When a bit is set, its function is enabled and when it's cleared, that function is disabled. The function of each bit is:
 - ◆ Bit 0 (weight 1): If the CPU resets and does not detect another CPU, it logs fault 7900 if it's in the top slot and fault 8000 if it's in the bottom slot.
 - ◆ Bit 1 (weight 2): Assuming that two processors had been working, if a CPU fails to get responses from the other CPU, it logs 8100 if it's in the top slot and fault 8200 if it's in the bottom slot.
 - ◆ Bit 2 (weight 4): If a CPU that has been working with a second CPU detects that the other processor has reset for any reason including a power-on reset, it logs fault 8300 if it's in the top slot and fault 8400 if it's in the bottom slot. Note that in this case, it's to be assumed that regardless of what state it was in, the processor that logged the fault is now the "primary."
 - ◆ Bit 4 (weight 8): If the top CPU resets and assumes "secondary" status, it logs fault 8500.

Example: assuming GET? 80 responds with 15 (the sum of 1 + 2 + 4 + 8) and two CPUs are installed and working (which you can verify with GET? 14 returning 2 from one processor and 3 from the other):

- remove the top processor: fault 8200 is logged.
- reinsert the top processor: faults 8400 and 8500 is logged.

These features allow the user to tailor which incidents should log faults. Examples (with the corresponding value to set in the register):

- A user with only one CPU probably does not want to be informed at every reset that the "other" CPU is missing (0).
- A user might want to know if one of the two CPUs has a "hard" failure but doesn't care if one temporarily "wigs out" (3).

A user might want to know if anything suspicious happened and may decide to discuss it with the factory (15).

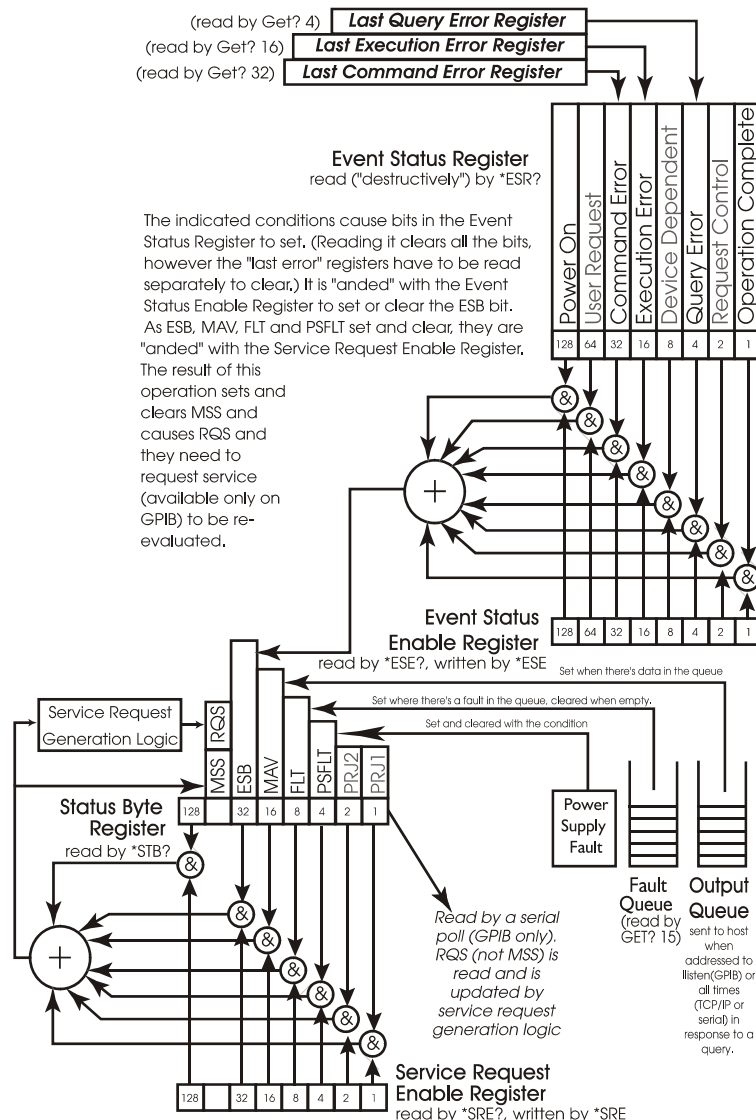
7. Remote Control Information

This section provides a detailed description of the Status Bite Register control protocol and the System Switching Modes.

7.1. Status Bite Register

The Status Bite Register tracks and reports the operating status of the system using the IEEE Standard 488.2 control protocol.

7.1.1. Status Bite Register Layout and Description



Italics indicate Universal Switching extensions to the model. dimmed items are not applicable to this product and are always 0.

7.1.2. The Status Byte Register (SBR)

The current bite status is summarized in the standard Status Byte Register (SBR). The SBR consists of the following bits with the following weights:

- **The standard Master Summary Status**

(MSS, numeric weight of 64) – this bit is dependent on the state of the other bits of the register and the current value of the Service Request Enable Register (SRER, see below for details on it and how the MSS is determined).

- **The standard Event Status Bit**

(ESB, numeric weight of 32) – this bit is dependent on the state of the Event Status Register (ESR) and the current value of Event Status Enable Register (ESER) (see below for both).

- **The standard Message Available bit**

(MAV, numeric weight of 16) – this bit is set when the Output Queue (see below) has responses for the host in it. It gets cleared when it's empty.

- **The Fault bit**

(FLT, numeric weight of 8) – this bit is set when there are faults recorded in the Fault Queue (see below). The usage of this bit is defined by Universal Switching.

- **The Power Supply Fault bit**

(PSFLT, numeric weight of 4) – this bit is set and cleared as faults are found or cleared in the power supplies.

The remaining bits (numeric weights 128, 2 and 1) are not defined by either 488.2 or Universal Switching Corp. They are currently always cleared. Universal Switching Corp reserves the right to use the undefined bits in the future or in project-specific applications. **It is suggested that your host application mask these bits out.**

7.1.2.1. Reading the Status Byte Register (SBR)

The SBR is read by issuing the *STB? common command and interpreting the response. The response is a decimal number that, in practice, is between 0 and 127. The value indicates which of the five bits are set or cleared. The SBR is read-only.

On GPIB interfaces, the SBR is returned to the host in response to a serial poll. The 488.1 Request Service bit (RQS) is returned instead of the MSS bit in this case. For details, please consult the 488.2 standard.

On a technical note, serial polls are handled directly by interface ASICs without the intervention of the processor. As service requests are acted upon by the interface ASICs, the moment-by-moment status of RQS may be changed by the ASIC. MSS, on the other hand, is changed by the processor only. (Note that the serial poll causes RQS to clear but has no direct effect on MSS.)

7.1.3. The Service Request Enable Register (SRER)

The standard Service Request Enable Register (SRER) can be read by the common *SRE? command and written by the *SRE command. With the exception of the bit with weight 64, each bit in the SRER corresponds to a bit in the SBR. In general, if at least one bit in the SRER is set and its corresponding bit in the SBR is also set, then MSS is set. Otherwise, MSS is cleared.

As an example, if the bit with weight 16 is set in the SRER, then MSS is set whenever MAV is set. Any of the following are examples of commands which set SRER that way: *SRE 16; *SRE 48; *SRE 56. The bit with weight 64 (which corresponds to the MSS itself) is ignored. In this implementation, all unused and undefined bits are always zero.

As an example, the response from *SRE 255; *SRE? is 56 unless one of the above mentioned bits has been designated to support a project specific function which shall be documented separately.

7.1.4. The Event Status Enabled Register (ESER)

In a similar situation, each bit in the standard Event Status Enabled Register (ESER) corresponds to a bit in the Event Status Register (ESR, see below). All eight bits in both registers are defined. If at least one bit in the ESER and its corresponding bit in the ESR are set, then the ESB bit of the SBR is set, otherwise the ESB is cleared.

The SRER is written to by the *SRE command and read by *SRE?

The Message Available (MAV) bit of the SBR tracks the status of the standard Output Queue. If there are responses waiting in the queue, MAV is set, if the queue is empty, MAV is cleared.

In the following example:

```
STB?;*STB?
```

The response could possibly be 0;16 or possibly 32;48 (if ESB was previously set). This is because the first command sees an empty queue while the second *STB? sees effect of the first command in the queue on MAV. Using *STB? to poll MAV is futile. (In order to read the response from *STB?, you have to address the device to talk as if MAV is set.)

Again, GPIB behaves differently from serial or TCP/IP. The latter two immediately sends their responses back to the host on their own initiative. GPIB, on the other hand, waits until it is addressed to talk. See below for details on what happens if it's not address to talk prior to the arrival of the next command.

7.1.4.1. The ESER under GPIB Control

GPIB behaves differently from serial or TCP/IP. The latter two immediately send their responses back to the host on their own initiative. GPIB, on the other hand, waits until it is addressed to talk.

If the bit with weight 16 is set in SRER, then there is a request for service. Even without that, the host can still perform a serial poll and from the presence of the 16 in the SBR, it can infer that it needs to be addressed to talk.

The FLT bit of the SBR tracks the condition of the Fault Queue. The Fault Queue (see below) is a list of “faults” detected by the system.

Faults require operator and/or manufacturer intervention.

See below. The FLT bit can be monitored by the host to detect such conditions. Additionally, its corresponding bit in SRER can be set and thus cause a service request in the event of a fault.

7.1.5. The Event Status Register (ESR)

The standard Event Status Register (ESR) tracks several events. As mentioned above, it works in conjunction with the ESER to determine the value of the SBR’s ESB bit. As set forth in 488.2, when a condition that is monitored by one of the bits is encountered, that bit is set and remains set until all the bits are cleared by:

- Reading the ESR with the *ESR? common command
- The common *CLS command
- The common *RST command
- A processor reset

The ESR consists of the following bits and their weights:

- **Power-on (PON, weight 128)**
It is set when the power is turned on.
- **User Request (URQ, weight 64)**
It is not used and is always cleared.
- **Command Error (CME, weight 32)**
It is set when a syntax error (see below).is encountered in a command
- **Execution Error (EXE, weight 16)**
It is set when an execution error (see below) is encountered in a command.
- **Device Specific Error (DDE, weight 8)**
It is not used and is always cleared.

- **Query Error (QYE, weight 4)**

It indicates that the device was addressed to talk when the output queue was empty or that a new command was received before the output queue was emptied. Both cases are possible under GPIB, but highly unlikely with other interfaces.

- **Request Control (RQC, weight 2)**

It is not used and always clear.

- **Operation Complete (OPC, weight 1)**

Execution of the *OPC common command causes this bit to be immediately set. See the description of the *OPC, *OPC? and *WAI commands for details.

A command error (CME) results from the processor encountering a syntax error in a command. If the command is a compound command, all commands up to the offending command are still scheduled to execute. The command with the error and any following commands are discarded.

As indicated previously, the CME bit of ESR is set. Additionally, a numeric code is placed in the Last Command Error register (see below) to indicate the nature of the syntax error.

An execution error (EXE) results from the processor attempting to execute an otherwise syntactically correct command that violates some condition. Some examples might be an output that's out of range or trying to connect to an output that's already in use when auto-interlock is not activated. Again, if it's a compound command, all commands up to the offending command are executed.

The command with the execution error and any commands following it are discarded. Again, the EXE bit of ESR is set and a numeric code is placed in the Last Execution Error register (see below).

A query error (QYE) sets the QYE bit of the ESR and causes a numeric code to be entered in the Last Query Error register (see below). It does not affect the parsing or execution of otherwise valid commands except to indicate that valid responses may have been lost.

The Last Command Error (LCE), Last Execution Error (LEE) and Last Query Error (LQE) registers and their behavior are specific to UNIVERSAL SWITCHING products. As indicated above, when one of the errors is encountered, a numeric error code placed in one of the registers and its flag in ESR is set. The error code can be read using the GET? Command with the code for that register.

The error code stays in the register until:

- It is replaced by a new error code
- The register is cleared with a *RST, *CLS or processor reset
- The register was read with a *GET? command when its corresponding bit in the ESR is cleared.

7.1.6. Procedure for Recovering Errors

The suggested procedure for recovering these errors is to first read ESR with an *ESR?. If its bit in ESR is set, then perform a GET? to recover the code.

7.1.6.1. Example Register Interaction

The following example should help illustrate the interaction between these registers and ESR.

- Suppose that the prior command has caused an execution error 1 (invalid output) and there were no other problems prior to it.
- Assume that LEE is read by the GET? 16 command. In that case, the following:

```
GET? 16; GET? 16; *ESR?; GET? 16; GET? 16; *ESR?  
yield: 1;1;16;1;0;0.
```

- The first and second reads of the LEE both return 1 since nothing has happened to change it. Reading the ESR returns 16 (the set EXE bit) and clears it. The next read of LEE also returns 1, but since EXE is now cleared, LEE is also cleared. Finally, the last read of LEE and ESR show the results of the clearing.

7.1.6.2. The Fault Queue

The Fault Queue is a FIFO queue of system faults. Faults require intervention by either the operator (plug in the power supply that the janitor unplugged), or the manufacturer (repair the power supply that failed).

The queue can be read with repeated calls to the FAULT? Command. If it's empty, FAULT? returns a 0. If the queue should fill-up, it over-writes the older entries in a "circular" fashion. If there are any entries in the queue, the FLT bit in SBR is set. Upon reading the last fault (if any) in the queue, FLT is cleared. **As a departure from 488.2, *CLS and *RST does not clear the queue or the FLT bit.** Only a complete system reset (or reading them as specified above) clears them.

7.1.7. Register Function Summary

To summarize the register functions:

- The overall status can be determined by reading SBR with the *STB? command:
- A set PSFLT bit indicates a fault currently occurring in one of the power supplies. Note that is bit unique in that it sets and clears independently of host or keypad commands.
- A set FLT bit indicates the need to retrieve serious faults with the FAULT? command
- A set MAV bit indicates that there's data in the output queue
- A set ESB bit indicates that one or more of the bits in the ESR that were enabled in ESER are set.
- A set MSS bit indicates that one or more of the above three bits is set along with its corresponding bit in the SRER.
- The SRER is "anded" with the FLT, MAV and ESB bit of the SBR to generate the current MSS. It is set with *SRE and read by *SRE?

- The ESR contains several bits which in practice indicate the success or failure of a command to execute and respond. It is set by the various conditions and read by the *ESR? command. Reading it clears it.
- The ESER is “and’ed” with the ESR to generate the SBR’s ESB bit. It is written with the *ESE command and can be read with the *ESE? command.
- Set CME, EXE or QYE bits in the ESR indicate that an error code is stored in LCE, LEE or LQE. These are read by GET? xx.

The device maintains a separate set of registers for each interface (assuming there’s more than one). In practical terms, this means that each interface can have, for example, a separate SRER or ESER. The Fault Queue, on the other hand, is common to all interfaces in the device.

Several conditions ultimately determine the current value of MSS. For example, setting the OPC bit of the ESR does not necessarily set MSS. First OPC’s bit in the ESER must be set in order for ESB to be set, then ESB’s bit in the SRER must be set before the condition exists to set MSS.

As each change occurs, the value of MSS is re-evaluated and updated regardless of whether the change was an internal condition or a commanded change of one of the enable registers. On GPIB interfaces, as MSS is set or cleared, the need to request service or cancel a previous service request is also evaluated.

7.1.8. Commands to Evaluate and Control Status (488.2)

The following commands are used to evaluate and control the status of the device. Unless noted, these commands are part of the IEEE 488.2 standard command set. Recall that each interface maintains its own set of registers (but share the Fault Queue).

***STB?** – returns the value of the SBR at the start of the command. Note that the effect of the command on MAV won't be visible until after the command is executed. The most effective place for this command is as the last command of a compound command. Bits 7, 2, 1 and 0 are not used by this device at this time. Host programs should mask them out.

SRE *n – *n* is an integer value between 0 and 255. *n* is "anded" with 00111000 and set in the SRER. This corresponds to the bits that are in use in SBR. If at least one bit is set in SRER and its corresponding bit in SBR is also set then the MSS bit in SBR is set. Each time the SRER is changed (by this command), the value of MSS is re-evaluated.

***SRE?** – returns the value of the SRER as an integer between 0 and 255. Note that any unused bits are mapped out whenever a value is set in SRER. To illustrate the following command: SRE 255;SRE? returns 56.

***ESR?** – returns the value of the ESR as an integer between 0 and 255. Per 488.2, the reading of the ESR clears it.

***ESE?** – returns the current value of the ESER as an integer between 0 and 255. The value of ESER does not change.

ESE *n – sets the integer value *n* (between 0 and 255) into ESER. If at least one of the bits in ESER is set at the same time as its corresponding bit in ESR, then the ESB bit in SBR is also set. Each time ESER is changed (by this command), the value of ESB and MSS are re-evaluated.

PSC *n – sets the integer value *n* (0 or 1) in the PSC register. If the register is 1, ESER and SRER are cleared on each restart. If its value is zero, ESER and SRER are preserved. One possible use of this feature is that by setting the PON bit in ESER, setting the ESB bit in SRER and setting 0 in PSC, the device requests service from its controller on power-up. (This works only on GPIB.)

***PSC?** – returns the current value of PSC

***CLS** – This clears the interface’s Last Error Registers, Event Status Register and the Status Byte Register (except for MAV, FLT and PSFLT) but not its queued responses. It has no affect on the other interfaces. As a departure from the standard, the Fault Queue is not affected. As a result, after the command, MAV, FLT and PSFLT reflects the status of their underlying conditions.

***RST** – This command clears all the switchpoint routings in all the modules. No other status registers are affected

7.1.8.1. Commands to Evaluate and Control Status (non 488.2)

The following commands are used to evaluate the status of the device. They are in addition to IEEE 488.2 commands and their functionality is not defined by that standard.

GET? 15 – returns the oldest entry in the Fault Queue as a number between -32768 and 32767. If the queue is empty it returns 0. If there are entries in the queue, the FLT bit of SBR is set. Reading the last entry causes the FLT bit to clear and for MSS to be evaluated.

The following three read-only registers behave differently from the others properties that can be read with the GET? query commands. Upon detecting an error of one of the above types, its corresponding flag in the ESR is set and the error is recorded in the register. (Any previous, unread error is over-written.) At this point, the register can be read with the GET? query. Once the ESR is cleared or read (which clears it), the Last Error Registers clear after the next time they are read

GET? 4 - returns contents of the Last Query Error Register

GET? 16 - returns contents of the Last Execution Error Register

GET? 32 - returns contents of the Last Command Error Register

7.1.9. Overlapped Processing

The due to the nature of switching operations, all commands are processed sequentially. There is no overlapped processing. As such the following commands are included for compatibility with other 488.2 devices. They provide no added functionality beyond performing their required functions in as little time as possible.

Per the standard these operations wait while the no-operation-pending flag is false. On devices that have only sequential operations, the standard calls for this flag to always be true.

***WAI** – does not have any effect.

***OPC?** – returns “1”.

***OPC** – sets the OPC bit in the ESR. The values of ESB and MSS are re-evaluated.

7.2. System Switching Mode

The switching system can be enabled in two modes; AUTO-ROUTE mode or GANGED mode. The mode of the system affects the allowed syntax for the following operational commands:

- **CONnect** Makes a connection within the system.
- **DISconnect** Makes a disconnection within the system.
- **QUERy** Queries the complete system or a single switchpoint.

Depending upon the system mode, these commands may or may not require a module number in the command syntax entered by the user.

7.2.1. Auto-Route Mode

The AUTO-ROUTE mode automatically routes the command entry without the need to address a specific module number.

7.2.1.1. Output End to End System

The output end-to-end system assumes a single configuration that may comprise one or several modules. The total number of outputs and inputs of the system is the result of all modules combined. The system is programmed to route automatically the command entry to the correct module without the need to address a specific module number. However, for diagnostic purposes only, the user has the option to specify a module when querying the system. The keypad on this type of configuration automatically displays the following:

“AUTO-ROUTE”

Output End-to-End systems are configured in AUTO-ROUTE mode.

7.2.2. Parallel Operation

PARALLEL operation requires a module number in the command string to correctly interpret the command entry. The PARALLEL system is configured to control each module individually. When controlling each module individually, if ALL is specified in the command syntax, the command operation is performed collectively on all modules.

- When Module 1 is specified in the command syntax, the command operation is performed on Module 1.
- When Module 2 is specified in the command syntax, the command operation is performed on Module 2.
- When Module 3 is specified in the command syntax, the command operation is performed on Module 3.

7.2.3. Ganged Mode

GANGED Mode does not require a specific module number to correctly route the command entry. The command operation is performed collectively on all GANGED modules.

Note that if GANGED is enabled, the entry of a single module is not possible and ALL is assumed. The keypad on this type of configuration automatically displays the following:

“GANGED”

For systems configured in GANGED mode, POLES or “virtual modules” may be formed to assist the programming and operation of the system.

7.2.4. Poles (Virtual Modules)

The concept and functionality of POLES or “virtual” modules is summarized as follows:

- POLES are used only in PARALLEL systems.
- POLES are “virtual” modules that gang together two or more “actual” modules. The system can have as many POLES as required.
- Operation on a POLE affects all the module’s members of the POLE.
- Any module member of a POLE maintains its independent functionality.

For example, in a system with three modules, Module 2 and Module 3 may be identified as POLE 4. POLE 4 is addressed in command syntax as “Module 4”. Command operations specifying “Module 4” simultaneously perform the command operation on Module 2 and Module 3. However, any module maintains its independent functionality and may be addressed as follows:

- When Module 1 is specified in the command syntax, the command operation is performed on Module 1.
- When Module 2 is specified in the command syntax, the command operation is performed on Module 2.
- When Module 3 is specified in the command syntax, the command operation is performed on Module 3.

8. Error Code List

The following is a standard list of error codes from the Series C710 and C720 plug-in CPU assemblies. Please note that some codes may not be applicable to all types of equipment or configurations. See notes at the bottom of the table for applicability and other information.

The types of errors are divided into four classifications:

1. E = Execution
2. C = Command
3. F = Fault
4. Q = Query

All errors are 1 or 2-digits (except faults which are 3 to 4 digits). The first two digits are defined in the table below. The last two digits are application specific.

Error	Type	Description
1	E	Invalid output parameter
2	E	Invalid input parameter
3	E	Invalid command
4	E	Output already connected to different Input
5	E	Blocking input connected already
6	E	Output not connected to anything
8	E	Memory location not used yet
9	E	Bad command argument
10	E	Invalid slot parameter
11	E	Not a valid "GET" property
12	E	Not a valid "SET" property
14	E	Store (*SAV) or Recall (*RCL) out of range
15	E	Set or Get not defined for this system
16	F	Remote missing (Master only). See note 1.
17	F	No top route (Master only). See note 1.
18	F	No bottom route (Master only). See note 1.
19	F	Excessive faults in remote. See note 1.
21	E	Command length error
22	E	No mid-stage available
26	E	Non existing module
27	E	Coil error from relay(s)
28	E	Memory location for "**SAV" temporarily not available. See note 2.
31	Q	Lost command (new command sent before last response read)
32	E	Wrong mid-stage addressed - Partial ganging (parallel). See note 3.
33	E	Path is not excluded. See note 3.

Error	Type	Description
34	E	List is full, no more exclusions. See note 3.
35	E	Path is excluded. See note 3.
40	F	Power supply #1 fault
41	F	Power supply #2 fault
42	F	Power supply #1 is missing
43	F	Power supply #2 is missing
47	E	Missing module in system
50	F	SPI "time-out" error caused by module not responding
52	F	SPI error (internal)
53	F	SPI error (internal)
54	F	SPI error (internal)
55	F	SPI error (internal)
56	F	SPI error (internal)
57	F	SPI error (internal)
58	F	SPI error (internal)
59	F	SPI error (internal)
61	C	Wrong 1st argument
62	C	Wrong 2nd argument
63	C	Wrong 3rd argument
64	C	Blank program message unit (IEEE-488.2 standard term)
65	C	Firmware issue (consult factory)
66	C	Invalid command header (IEEE-488.2 defined)
67	C	Command has too many arguments
68	C	Command has too few arguments
73	Q	Addressed to talk but has no response
74	F	US-Link Timeout
75	F	US-Link address wrong
76	F	US-Link bad command
77	F	US-Link remote has a fault
78	F	US-Link remote has an ongoing fault condition
79	F	Dual transition Fault CPU1 (Top) is alone
80	F	Dual transition Fault CPU2 (Bottom) is alone
81	F	Dual transition Fault CPU2 (Bottom) stopped responding
82	F	Dual transition Fault CPU1 (Top) stopped responding
83	F	Dual transition Fault CPU2 (Bottom) reset
84	F	Dual transition Fault CPU1 (Top) reset
85	F	Dual transition Fault CPU1 (Top) came up as secondary
86	F	Invalid mid-stage parameter. See note 4.
88	F	Dual processor firmware mismatch

Note 1: Faults 16, 17, 18, and 19 apply only to Master/Remote switching system.

Note 2: If error 28 does occur, try splitting the compound command with all those QUERY? ALL commands and the *SAV into two or more commands.

Note 3: Errors 32, 33, 34 and 35 apply only to Tri-stage switching system.

Note 4: Fault 86 applies only to Tri-stage switching system.

9. Remote Control Assembly

The Series C710, C720-S3 and XC710 controller CPU/interface provides the brain and the remote control capacity of the switching system. The CPU offers a host of features and is compatible with all current Universal Switching Corp. switching systems.

The CPU/interface features a **US-Link**/Serial Service port for firmware upgrades and is currently available in the following configurations:

- **XC710-400 & XC710-SS244:** Ethernet (10BaseT), GPIB (IEEE-488), Serial (RS-232C/422A/485)
- **C710-E10:** Ethernet control only (10BaseT)
- **C710-488:** Ethernet and GPIB (IEEE-488)
- **C710-S3:** Ethernet and Serial (RS-232C/422A/485)
- **C720-S3:** Ethernet (10-base-T and 10/100-base-T) and Serial (RS-232C/422A/485)

The unit model number determines the type of CPU included with the unit when is delivered from the factory. Some unit configurations can have two (2) of these assemblies installed for additional flexibility or for a redundant means of control.

These CPU assemblies are installed into the rear of most systems and are secured by two Phillips-head screws at the top, and two at the bottom.

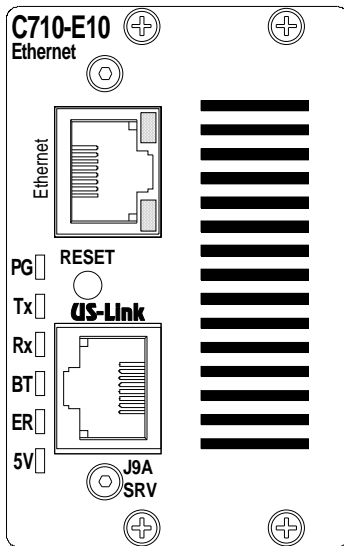


CAUTION: *The CPU Assembly contains a battery. Contact Technical Support before replacing the battery.*

9.1. CPU Configurations

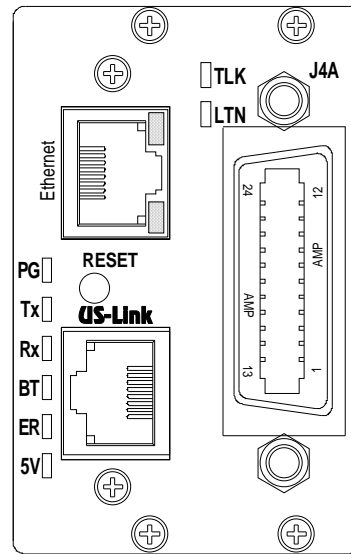
C710-E10

ETHERNET ONLY



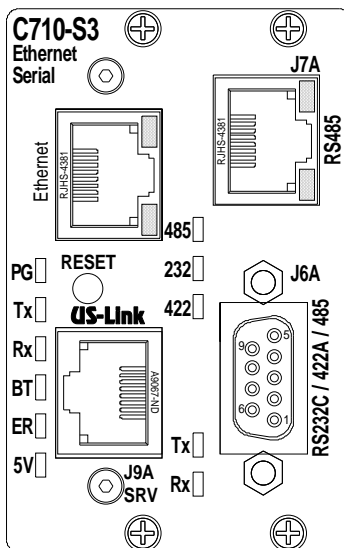
C710-488

ETHERNET AND GPIB



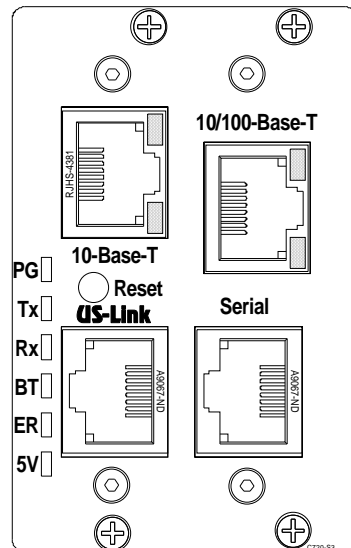
C710-S3

ETHERNET AND SERIAL



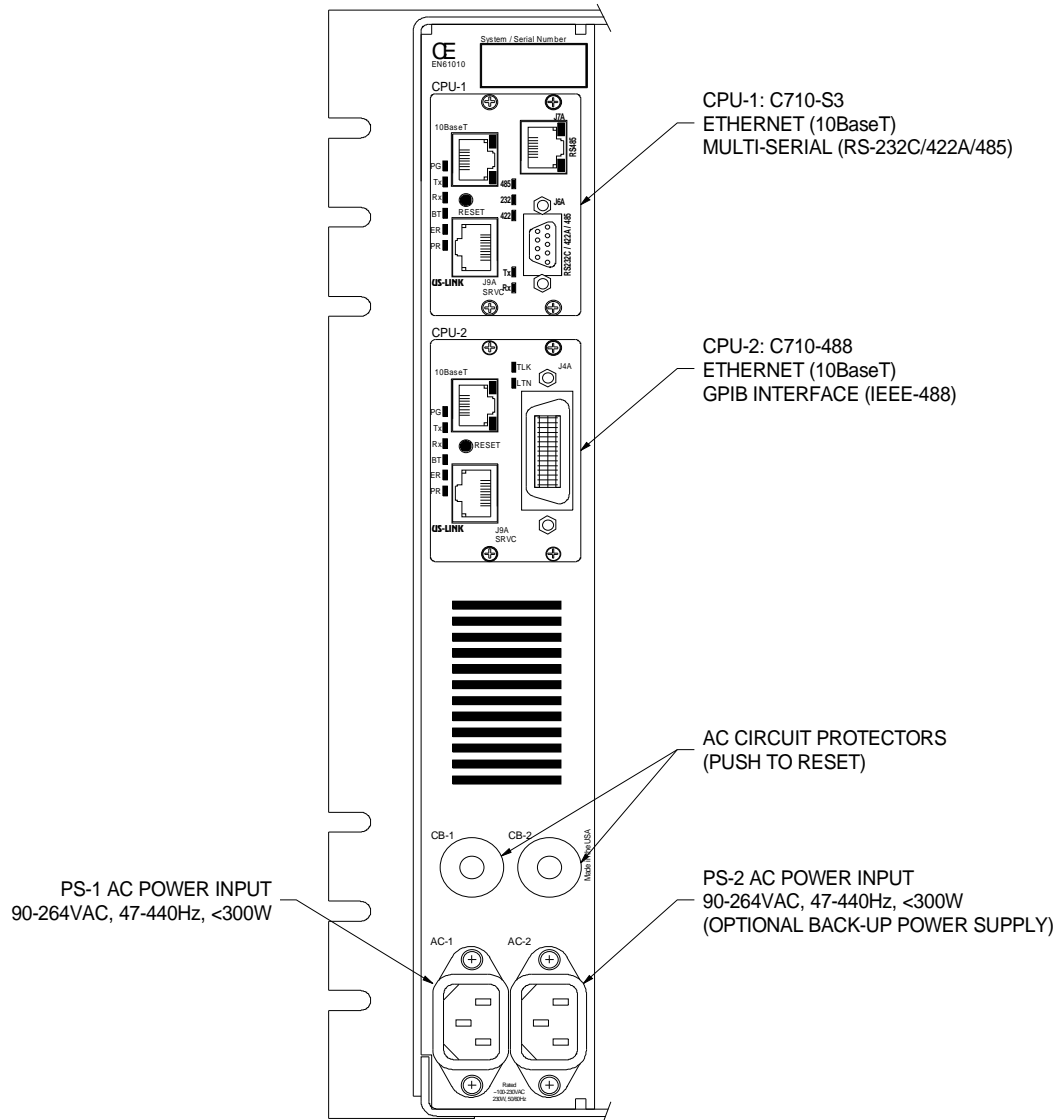
C720-S3

ETHERNET (10-Base-T, 10/100-Base-T)
AND SERIAL



9.2. Multiple CPU Installations

Some unit configurations can have two (2) of these assemblies installed for additional flexibility, or for a redundant means of control. The CPU/interface assemblies allow for flexible control scenarios to be constructed. Any two (2) different types or two (2) identical types may be installed in the unit at the same time to support redundant control schemes. Below is a rear view of a larger unit (8RU) that contains two (2) CPU slots. The upper slot is assigned as CPU-1 and the lower slot is assigned as CPU-2.



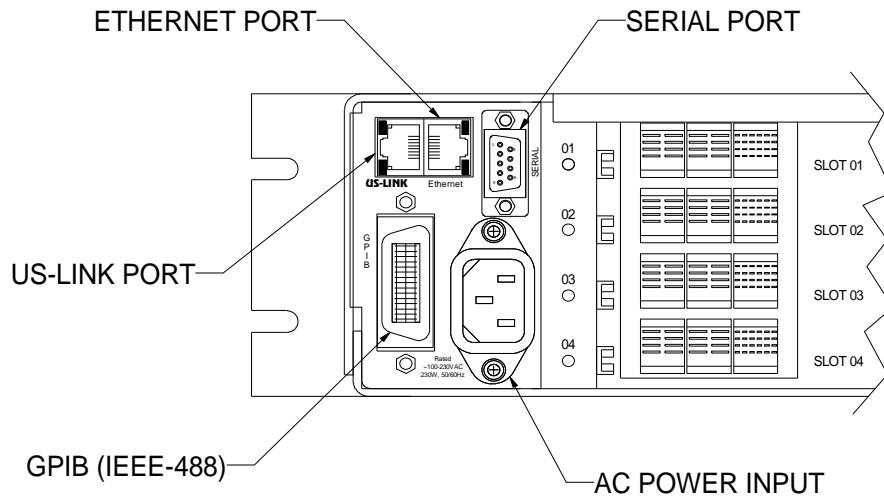
9.3. XC710: Ethernet and Serial Interfaces (Remote)



NOTE: The XC710-400 & XC710-SS244 are referred to as the XC710.

The XC710 installed in a G2S400 mainframe is equipped with rear mounted remote control interfaces. The XC710 can accept control from Ethernet (10-base-T), RS-232C/422A/485 serial, or GPIB via the rear mounted control interfaces. Choosing between these interfaces is automatic as they are serviced on a "first-come, first-served" basis.

Remote interface ports are located on the rear of the G2S400 mainframe as indicated in the illustration below.

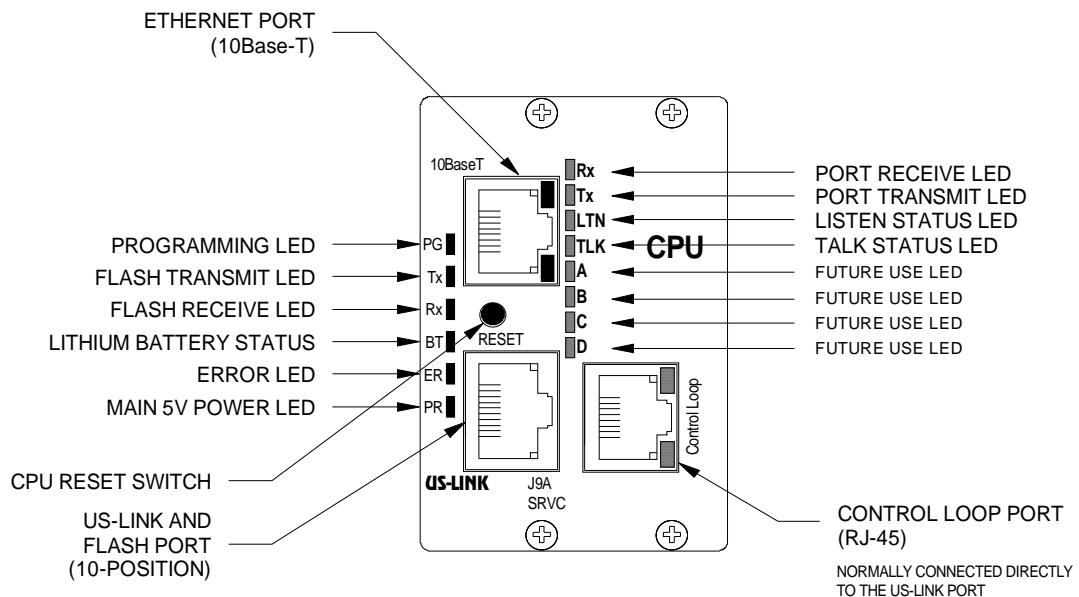


9.4. XC710: Ethernet Interface

The G2S400 mainframe is equipped with a front mounted XC710 CPU. The CPU unit can accept control from Ethernet (10-base-T) via the front mounted interface, or or receive firmware via the **US-LINK** interface.



NOTE: Use either the front mounted or rear mounted Ethernet interface. Do not connect to both interfaces at the same time.

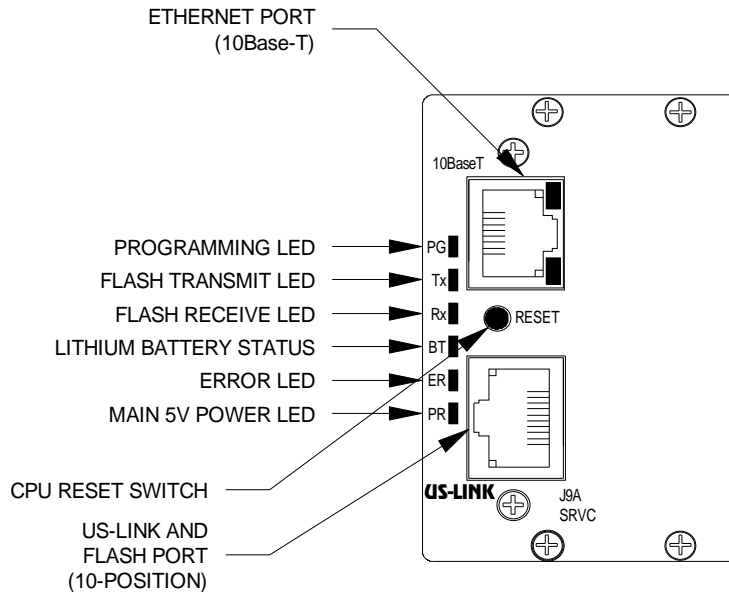


The XC710 plug-in assembly is designed specifically for the G2S400 mainframe and includes the following:

- Standard GPIB, Serial, and Ethernet interface ports
- LED status indicators
- Jumper selectable RS-232C, RS-422A or Multi-drop RS-485 serial settings
- Serial firmware upgrade port (J9A)
- Lithium-backed RAM
- Upgradeable FLASH program area
- Store/Recall complete switching configurations
- TCP/IP is included

9.5. C710-E10: Ethernet Interface

C710-E10 ETHERNET ONLY

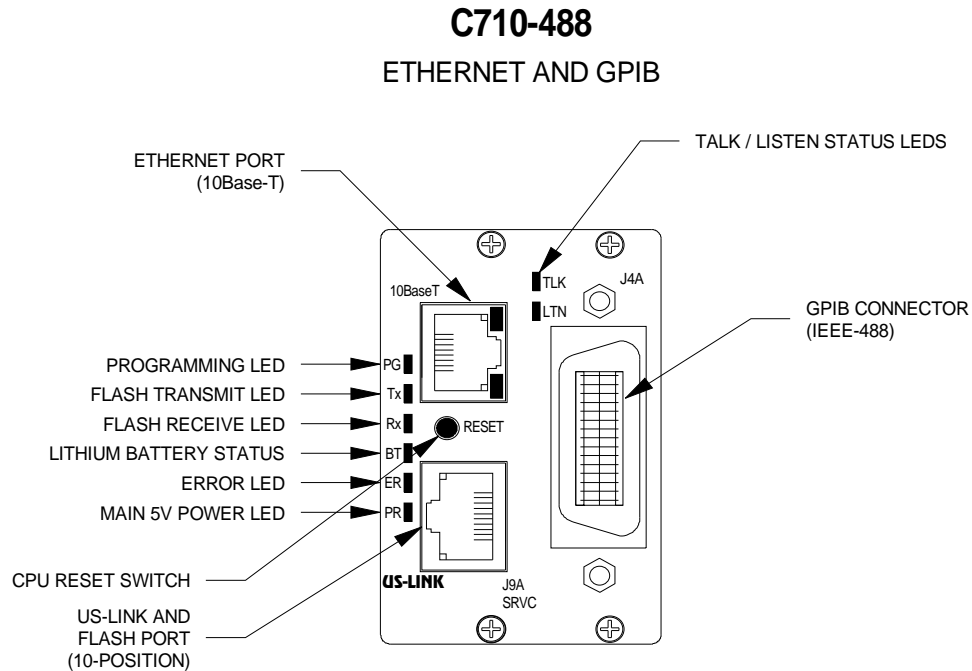


The C710-E10 provides only an Ethernet port (10-base-T) control port. The following sections explain how to change the functions of the plug-in CPU.

The C710-E10 includes the following:

- TCP/IP is included
- LED status indicators
- Serial firmware upgrade port (J9A)
- Lithium-backed RAM
- Upgradeable FLASH program area
- Standard **US-LINK** interface for remote control panels

9.6. C710-488: Ethernet and GPIB Interfaces



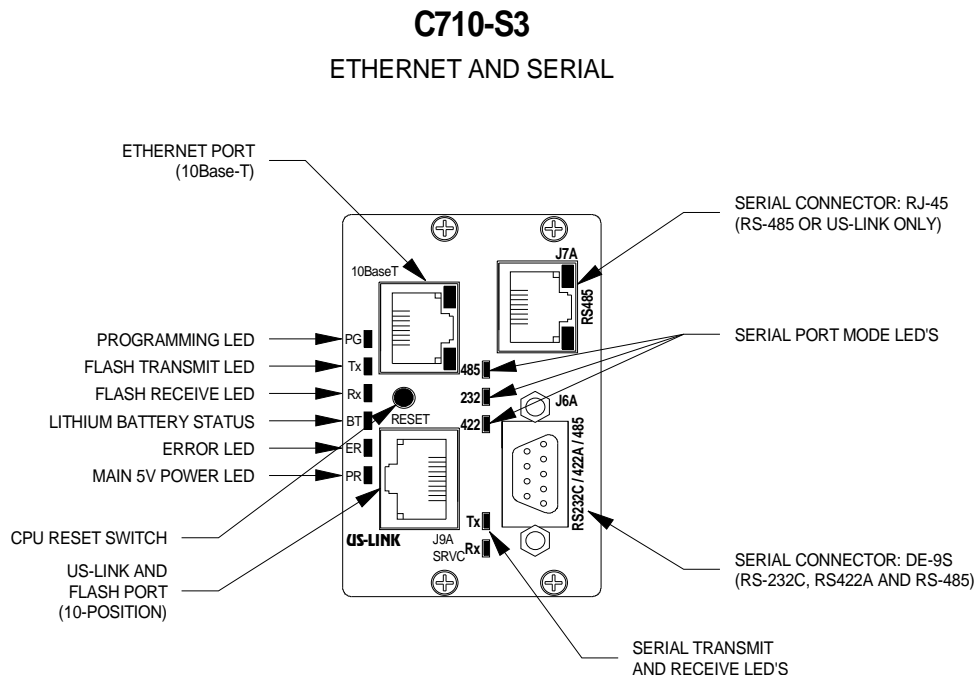
The C710-488 provides both an Ethernet port and a GPIB control port (IEEE-488). **The GPIB port is factory configured for an address of 20.**

The C710-488 includes the following features:

- TCP/IP is included
- Standard GPIB interface port (J4A)
- LED status indicators
- Serial firmware upgrade port (J9A)
- Lithium-backed RAM
- Upgradeable FLASH program area
- Standard **US-LINK** interface for remote control panels

9.7. C710-S3: Ethernet and Serial Interfaces

The C710-S3 provides an Ethernet control port (10-base-T) and either RS-232C, RS-422A or multi-drop RS-485 type interfacing. **The serial port is factory configured for 9600 Baud, 8 Data bits, 2 Stop bits, No Parity, and no flow control.** Changes to these settings are explained in the following sections. The J6A connector is used for RS-232C/422A/485 serial modes, and the J7A connector is used only for RS-485.

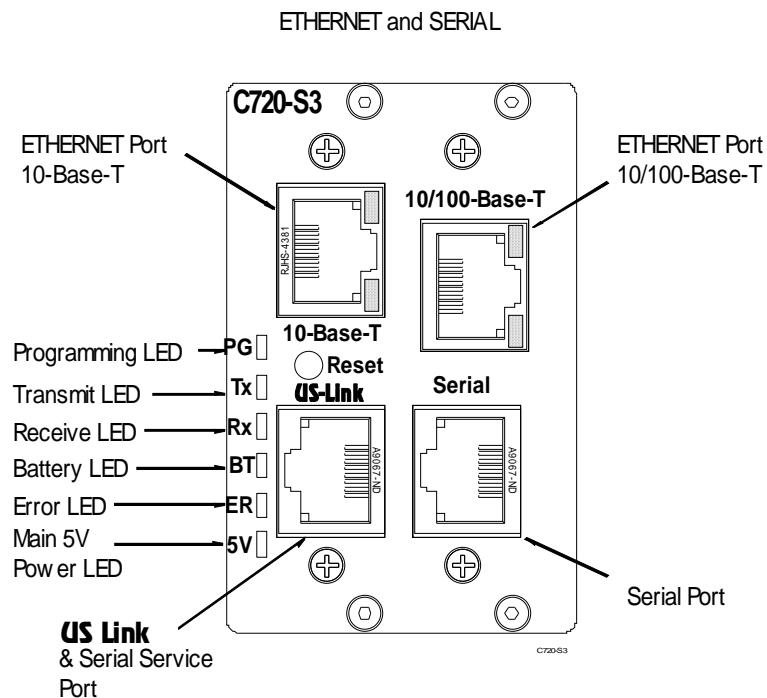


The C710-S3 includes the following features:

- TCP/IP is included
- Standard serial interface port (J6A and J7A)
- LED status indicators
- Jumper selectable RS-232C, RS-422A or Multi-drop RS-485
- Serial firmware upgrade port (J9A)
- Lithium-backed RAM
- Upgradeable FLASH program area
- Standard **US-LINK** interface for remote control panels

9.8. C720-S3: Ethernet and Serial Interfaces

The C720-S3 provides Ethernet control ports (10-Base-T and 10/100-Base-T), a **US-Link** port that doubles as a Serial Service firmware upgrade interface, and a Serial Port for RS-232, RS-422 or multi-drop RS-485 type interfacing. The Serial Port is factory configured for 9600 Baud, 8 Data bits, 2 Stop bits, No Parity, and no flow control.



The C720-S3 includes the following features:

- TCP/IP (10-Base-T and 10/100-Base-T)
- Serial interface port
- LED status indicators
- Jumper selectable RS-232C, RS-422A or Multi-drop RS-485
- Firmware upgrade serial service port (shared with **US-LINK**)
- Lithium-backed RAM
- Standard **US-LINK** interface for remote control panels

9.9. Remote Control Interfaces

The following explains the various remote control interfaces. Each type of CPU contains an Ethernet (10-base-T) and US-Link port. Additional ports are featured on specific CPU models.

9.9.1. The Ethernet (10-Base-T & 10/100-Base-T) Port Pin Assignment

The Ethernet port is designed per standard 10-Base-T and 10/100-Base-T specifications, and has standard pin assignments. For reference, the table below describes the pin assignments.

Pin	Signal Assignment
1	Tx +
2	Tx -
3	Rx +
4	No Connect
5	No Connect
6	Rx -
7	No Connect
8	No Connect

9.9.2. Default IP Address

When a unit leaves the factory it is configured with the following:

- IP address: 10.100.1.49
- Port: 7145 (10BaseT)
- Port 5000 (10/100BaseT)



NOTE: The 10-base-T and 10/100-base-T Ethernet have the same IP address and should not be on the same network.

9.9.2.1. Changing the Ethernet Settings

To change the factory default IP address, you can use the "GET?" and "SET" commands with parameters 33 through 48 to read, change and verify the IP address. See the REMOTE CONTROL COMMANDS section of this manual.


9.9.2.2. Changing the Ethernet Settings from the Front Panel

See the FRONT PANEL section of this manual for instruction to change the Ethernet Setting from the front panel.

9.9.3. The **US-Link** Interface and Service Port (C710, C720, XC710)

The CPU/interface is equipped with a **US-Link**/serial service connector used to link systems with multiple chassis/mainframes and to flash firmware to the CPU. Both control and status reporting are passed to/from the remote units through the **US-Link** port. The **US-Link**/serial service connector also shares the J9A port that is used for uploading new firmware to the CPU.

The **US-Link**/serial service connector used on the C710, C720, and XC710 CPUs is a 10-position RJ-50 type jack. See the table below for signal assignments. The port utilizes half-duplex RS-232 type of serial transmission. A cable assembly P/N CA350-008 is provided to connect the **US-Link**/serial service port to a computer or other serial control device.

 **NOTE:** The CA350-008 cable must be used to connect a computer to the **US-Link**/serial service port otherwise damage to the unit may occur.

9.9.3.1. The US-Link/Service Port Pin Assignment

Pin	Signal Assignment (US-Link)	Signal Assignment Serial Service (J9A)
1		Program (ground to activate programming)
2		RxD (used for programming only)
3		TxD (used for programming only)
4	GND	
5	GND	
6	- RS485 (US-Link)	
7	+ RS485 (US-Link)	
8		DSR (used for programming only)
9		DTR (used for programming only)
10		GND

 **NOTE:** The **US-Link**/serial service port is located on the front mounted XC710 CPU on the G2S400 mainframe.

9.9.4. **US-Link** Rear Panel G2S400 Mainframe Interface Port (XC710)

The XC710 controller CPU rear panel interface is equipped with a **US-Link** port used to link multiple systems. Both control and status reporting are passed to/from the remote units through the **US-Link** port.

The **US-Link** connector on the rear panel of a G2S400 mainframe is a 8-position RJ-45 type jack. The table below describes the pin assignments.

 **NOTE:** The G2S400 mainframe rear panel **US-Link** port is used to link multiple systems. It is **NOT** a serial service port.

9.9.4.1. The **US-Link** Port Pin Assignment

Pin	Signal Assignment (US-Link)
1	
2	
3	GND
4	GND
5	+ RS485 (US-Link)
6	- RS485 (US-Link)
7	
8	

9.9.5. GPIB Port Pin Assignment (C710-488, XC710)

The GPIB port is designed per the IEEE-488.2 specification and has standard pin assignments. For reference, the table below describes the pin assignment.

Pin	Signal Assignment (J4A)
1	ID 1
2	ID 2
3	ID 3
4	ID 4
5	EOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	SHIELD
13	ID 5
14	ID 6
15	ID 7
16	ID 8
17	REN
18	GND (6)
19	GND (7)
20	GND (8)
21	GND (9)
22	GND (10)
23	GND (11)
24	LOGIC GROUND

9.9.6. Setting the GPIB Address (C710-488, XC710)

When a C710-488 unit leaves the factory, it is preconfigured with GPIB address 20. The address and SRQ settings may be altered in the field by the user from the front panel controls.

9.9.6.1. Changing the GPIB and SRQ Settings from the Front Panel

The GPIB port setting and SRQ setting can be modified from the front panel control. A different GPIB address may be selected or the SRQ setting may be disabled on the interface.

To change the GPIB interface parameters from the front panel, the following sequence must be observed (any remote control activity is automatically void and the interface is reset once the alterations are done).

- Press the MNU (menu) key on the control pad repeatedly until the "Interface Menu", is shown.
- Use the arrow keys (up/down) to move the cursor to the interface (CPU1-CPU2) that you desire to modify, and press ENTER. When depressed, the firmware version of the interface card is shown.
- Use the (left/right) arrow keys to select a parameter to be modified, then use the enter key to alter the value.

When all desired modifications are completed and have been stored; press "CNL" to return to the main screen and resume operation, or press "MNU" again to display the next setup menu.

9.9.7. Serial Port Pin Assignment: J6A (C710-S3)

The J6A serial port (DE-9S type) has RS-232C, RS-422A and RS-485 capability. The serial type is jumper selectable. Verify that only the desired signal wires are used (RS232C, RS422A or RS-485) in your cable, or damage could occur. Pins labeled as "Not Used" may contain active circuits and should not be connected.

Pin Number	RS-232C Mode	RS-422A Mode	RS-485 Mode
1	Not Used	Transmit Data - (out)	Not Used
2	Transmit Data (out)	Transmit Data + (out)	Not Used
3	Receive Data (in)	Receive Data + (in)	Data +
4	Not Used	Receive Data - (in)	Data -
5	Ground	Ground	Ground
6	Not Used	Clear To Send -	Not Used
7	Clear To Send	Clear To Send +	Not Used
8	Ready To Send	Ready To Send +	Not Used
9	Not Used	Ready To Send -	Not Used

9.9.8. RS-485 Serial Port Pin Assignment: J7A (C710-S3)

The J7A serial port (RJ-45 type) on the C710-S3 is used only for RS-485 multi-drop serial control and is a standard RJ-45 type connector. It offers the system engineer a connection alternative to the J6A D-Sub connector.

The selection of the RS-485 mode is jumper selectable. An LED indicator identifies the serial control mode. The additional LED indicators built-into the RJ-45 connector are always on (YELLOW and GREEN).

Verify that only the desired signal wires are used or damage could occur. Pins labeled "Not Used" may contain active circuits and should not be connected.

Pin Number	RS-485 Mode
1	Not Used
2	Not Used
3	GND
4	GND
5	Data +
6	Data -
7	Not Used
8	Not Used



NOTE: Pins labeled as "Not Used" may contain active circuits and should not be connected.

9.9.9. Setting the RS-485 Serial Port: J7A (C710-S3)

When the C710-S3 leaves the factory, the RS-485 serial port is pre-configured for 9600 baud, 8 bits per character, no parity, 2 stop bits, no handshaking, and an RS-485 address of ASCII "A".

These settings may be altered in the field by means of the front panel controls with exception of the RS-485 address. The RS-485 multi-drop address is changed only by dipswitch settings on the CPU main board. To change this, the CPU must be removed from the unit.



NOTE: If the C710-S3 is at the end of the RS-485 string, jumper JP8 must be installed in the 1-2 position to provide proper termination.


9.9.10. Serial Port Pin Assignment (C720-S3)

The C720-S3 serial port is a RJ-50 10-pin connector configured per the following table.

Pin Number	RS-232 Mode	RS-485 Mode	RS-422 Mode
1	Not Used	Not Used	Tx-
2	Not Used	Not Used	CTS-
3	Tx	Not Used	Tx+
4	CTS	Not Used	CTS+
5	Rx	Data +	Rx+
6	RTS	Not Used	RTS+
7	Not Used	Data -	Rx-
8	Not Used	Not Used	RTS-
9	SGND	SGND	SGND
10	SGND	SGND	SGND

The mode is jumper selectable. An LED indicator identifies the serial control mode. The additional LED indicators built-into the RJ-50 connector are always on (YELLOW and GREEN).

Verify that only the desired signal wires are used (RS232C, RS422A or RS-485) in your cable, or damage could occur. Pins labeled as "Not Used" may contain active circuits and should not be connected.

 **NOTE: If the C720-S3 is at the end of the RS-485 string, jumper JP7 must be installed in the 1-2 position to provide termination.**

9.9.11. Serial Port Pin Assignment: DE-9S (XC710)

The XC710 DE-9S serial port has RS-232C, RS-422A and RS-485 capability. The mode is jumper selectable. Verify that only the desired signal wires are used (RS232C, RS422A or RS-485) in your cable, or damage could occur.

Pin Number	RS-232C Mode	RS-422A Mode	RS-485 Mode
1	Not Used	Transmit Data - (out)	Not Used
2	Transmit Data (out)	Transmit Data + (out)	Not Used
3	Receive Data (in)	Receive Data + (in)	Data +
4	Not Used	Receive Data - (in)	Data -
5	Ground	Ground	Ground
6	Not Used	Clear To Send -	Not Used
7	Clear To Send	Clear To Send +	Not Used
8	Ready To Send	Ready To Send +	Not Used
9	Not Used	Ready To Send -	Not Used



NOTE: If the XC710 is at the end of the RS-485 string, jumper JP7 must be installed in the 1-2 position to provide termination.

9.9.12. Changing the Serial Settings from the Front Panel

After delivery from the factory, many of the serial control parameters can be changed from the front panel controls. However, the RS-485 address and the control mode (RS-232C, RS-422A or RS-485) must be set by changing jumper settings and/or dipswitches on the CPU assembly.

To change the other serial interface parameters from the front panel, the following sequence must be observed (any remote control activity is automatically void and the interface is reset once the alterations are done).

Press the MNU (menu) key on the control pad repeatedly to move to the "Interface Menu". This is the second screen (press MNU twice).

Use the arrow keys (up/down) to move the cursor to the interface that you desire to modify, and press ENTER. After ENT is pressed, the firmware version of the interface card is displayed.

Use the (left/right) arrow keys to select a parameter to be modified, and then use the (up/down) arrows to alter the value.

When all desired modifications are completed and have been stored; press "CNL" to return to the main screen and resume operation, or press "MNU" again to display the next setup menu.

9.9.13. Changing the Serial Mode



NOTE: To make changes to the jumpers or DIP switches, the CPU must be removed from the unit.

An 8-position dipswitch is located on the plug-in CPU. Also located on this assembly are a number of configuration jumpers. These jumpers and dipswitch settings allow the user to select between the three (3) different serial modes available on the C710-S3, C720-S3 and XC710 CPU assemblies

9.9.13.1. Setting the Jumpers for C710-S3

The jumpers are moved to different positions to properly connect drivers and receivers for the different serial modes on the C710-S3. The table below indicates how to set the jumpers for the different serial modes.

Mode	JP1	JP2	JP3	JP4	JP5	JP6	JP7	JP8	JP9	JP10	JP11	JP12
RS232	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3	1-2	2-3	2-3	1-2
RS485	1-2	1-2	2-3	1-2	1-2	1-2	1-2	1-2	1-2	2-3	2-3	1-2
RS422	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	2-3	2-3	2-3	2-3

9.9.13.2. Setting the Jumpers for C720-S3

The jumpers are moved to different positions to properly connect drivers and receivers for the different serial modes on the C720-S3. The table below indicates how to set the jumpers for the different serial modes.

Mode	JP1	JP2	JP3	JP4	JP5	JP6	JP7	JP8	JP9	JP10	JP11	JP12
RS232	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3	1-2	1-2	2-3	2-3
RS485	1-2	1-2	2-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	2-3	2-3
RS422	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	2-3	2-3	2-3	2-3

9.9.13.3. Setting the Jumpers for XC710

The jumpers are moved to different positions to properly connect drivers and receivers for the different serial modes on the XC710. The table below indicates how to set the jumpers for the different serial modes.

Mode	JP1	JP2	JP3	JP4	JP5	JP6	JP7	JP8	JP9	JP10	JP11	JP12
RS232	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3	1-2	1-2	2-3	2-3
RS485	1-2	1-2	2-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	2-3
RS422	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	2-3	2-3	2-3	2-3

9.9.14. Setting the Serial Mode SW2 DIP Switch (C710, C720)


 **NOTE: Switch SW1 is not used.**

The SW2 DIP switch positions 7 and 8 must be set according to the serial mode. The following table illustrates the DIP switch settings.

Only positions 7 and 8 are set to select the serial mode.

Mode	1	2	3	4	5	6	7	8
RS232	-	-	-	-	-	--	ON	OFF
RS485	1(LSB)	2	4	8	16	64(MSB)	ON	ON
RS422	-	-	-	-	-	-	OFF	ON

9.9.15. Setting the Serial Control Mode SW1 DIP Switch (XC710)

 **NOTE: Switch SW2 is not used.**

The table below defines the DIP switch SW1 settings. Only switch 7 and 8 are changed to assign the serial control mode.

Mode	1	2	3	4	5	6	7	8
RS232	-	-	-	-	-	--	ON	OFF
RS485	1(LSB)	2	4	8	16	64(MSB)	ON	ON
RS422	-	-	-	-	-	-	OFF	ON

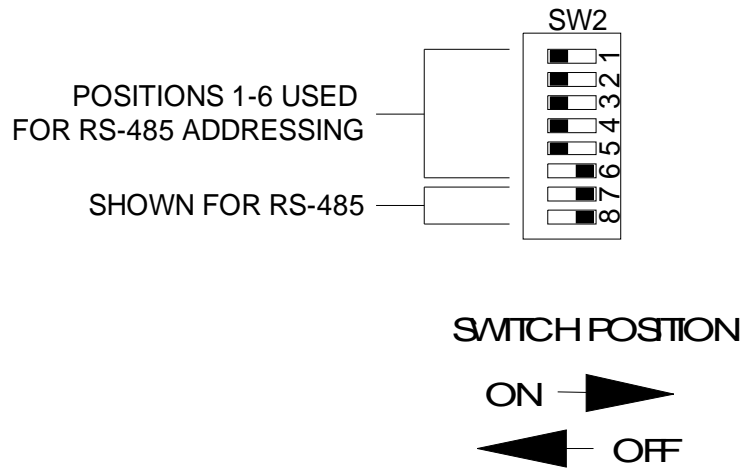
9.9.16. Setting the RS-485 Address

In the RS-485 serial mode, the unit needs to have an address. **The factory address default is uppercase ASCII "A"**. Positions 1 through 6 of the DIP SW2 switch are used to define the address of the port.

Since the RS-485 serial format provides for multiple units to share a common link, the addressing allows for individual control by preceding control data with an address. If the address byte matches, the unit responds. Only one device on the RS-485 can have a given address.

Only positions 1 through 6 are set to select the RS-485 address.

The diagram below shows the dipswitch on the CPU in the factory default for most RS-485 type shipments. The DIP switch positions (1 through 6) indicate an address of uppercase "A". Positions 7 and 8 are set to ON indicating RS-485.



 **NOTE:** Switch SW2 is used for the C710 and C720; SW1 is used for the XC710.

9.10. Upgrading the CPU's Firmware Driver

The CPU's firmware driver can be upgraded by the user while in the field. This is achieved by uploading the new firmware to the **US-Link**/serial service port.



NOTE: The G2S400 mainframe rear panel *US-Link* port is used to link multiple systems. It is NOT a serial service port.

The upgrading procedure for the CPU's Firmware Driver explained hereafter is provided for the following mainframes:

The G2S400CE (2RU high) Mainframe Series.

The G2S600CE (3RU high) Mainframe Series.

The G2S1200CE (6RU high) Mainframe Series.

The G2S1600CE (8RU high) Mainframe Series.

The S2084 (5RU high) Mainframe Series.

The S2560 (5RU high) Mainframe Series.

The S2561 (5RU high) Mainframe Series.

The S2562 (5RU high) Mainframe Series.

The S6025CE (9RU high) Mainframe Series.



NOTE: Except for the G2S400CE and G2S600CE mainframe, which use a single CPU, the rest of mainframes listed above are designed for optional dual CPU installation.

9.10.1. The Firmware Upgrade Package

A complete firmware upgrade package is provided in a CD-ROM along with a cable to connect the PC system or other serial control device into the CPU's **US-Link**/serial service port. Also, the upgrade package can be delivered via e-mail.

This upgrade package contains all the files required to upgrade the CPU's firmware driver. The following section is a step by step procedure on how to upgrade the CPU's firmware driver.

9.10.1.1. Upgrading Procedure:

Note: All firmware updates are available directly from the USWI.COM website. Open the website, and follow the link for Support/Software, Firmware, Drivers and Configuration Tools/Firmware Packages/Firmware.

1. Ensure that the system is powered ON.
2. Install the cable into the **US-Link** serial service port of the CPU faceplate. The programming LED (PG) illuminates.
3. If the programming LED does not illuminate; verify that the cable is connected to the **US-Link** serial service port, and NOT to an Ethernet or serial port.
4. Load the CD-ROM into the PC or other serial control device.
5. Open the CD. There are two (2) folders. One is labeled FALLBACK and is a copy of the delivered firmware of your system. The other is folder labeled KIT and contains the upgrade files and procedure.
6. Confirm that the system number on the upgrade (KIT) folder is the same as the system number of the system to be upgraded.
7. Drag the KIT folder onto the Desktop.
8. Open the KIT folder.
9. The KIT folder contains multiple files, including a FSP-C710 file containing firmware upgrade instructions, and a BIN file with the system model number.
10. Drag the file with the system model number onto the serial1, serial2 or serial3 file (according to your serial port using the CA350-008 cable).
11. A window opens. Follow the instructions on the screen to initiate the procedure.
12. After the procedure is completed, remove the cable from the **US-Link** serial service port and press the CPU reset switch to reset the system.
13. Verify the accuracy of the firmware version and revision code. The verification can be done from the front panel after resetting the system or from a response to an *idn? Query.
14. Finally, it is very important to remove the upgrade folder from your PC system desktop or drive to prevent interference with futures updates.



NOTE: The serial ports on Dell Inspiron laptop computers have been known to cause problems. If you intend to use this type of computer, a USB to serial converter is recommended.

10. Record of Changes

This section only applies to revised documents. The table below indicates the revision level entered and a brief description of the change(s).

Revision	Description of Change	Date
A	Revised jumper settings for RS485 at 9.7.11 per ECR 2450. Added XC710 configuration at 9.3, 9.4. Added serial port pin assignment table at 9.9.11. Added jumper table at 9.9.13.3. Added serial DIP switch SW1 at 9.9.15.	27 July 2009