Materials Engineering and Testing Services

2070 Controller

Introduction to the Controller

Table of Contents

I. Overview

A. Communications Structure
B. Motherboard Structure
C. OS-9 Operating System

II. 2070 Models

A. 2070 L (Lite)
B 2070 V (VME)
C. 2070 LC (Lite with 3C Front Panel)

III. 2070 Component Cards

A.	CPU Card (VME)	2070-1A
B.	CPU Card (LITE)	2070-1B
C.	Field I/O Card	2070-2A
D.	Field I/O Card (ITS Only)	2070-2B
E.	Front Panel Interface	2070-3A
F.	Front Panel Interface	2070-3B
G.	Front Panel Interface	2070-3C
H.	Power Supply	2070-4A and 4B
I.	VME Cage	2070-5
J.	Modem Card	2070-6B Async/Modem Serial Comm Module
K.	RS-232 Card	2070-7A EIA-232 Serial Port

List of Figures

Figure 1 2070 Architecture
Figure 2 Connecting Laptop Connected to 2070
Figure 3 Hyperterminal Terminal Shell
Figure 4 2070 V Unit Front View
Figure 5 2070 V Unit Back View
Figure 6 2070 V Unit Back – Top View With Additional 7A Card Installed
Figure 7 2070 L Unit Front View
Figure 8 2070 L Unit Back View
Figure 9 2070 L Unit Back - Top View With Additional 7A Card Installed
Figure 10 2070 LC Unit Front View
Figure 11 2070 LC Unit Back View
Figure 12 2070 LC Unit Back - Top View With Additional 7A Card Installed
Figure 13 2070-1A Transition Board Front and Side View
2070-1A CPU Located in VME Chassis
Figure 14 2070-1B Front and Side View
Figure 15 Pinout For C14S Ethernet Connector
Figure 16 Pinout For C13S EIA-485 Connector
Figure 17 2070-2A Front and Side View
Figure 18 Pinout For C12S EIA-485 Connector
Figure 19 2070-2B Front and Side View
Figure 20 Pinout For C12S EIA-485 Connector
Figure 21 2070-3A Front Panel
Figure 22 2070-3A Front Panel – Rear View
Figure 23 C50S EIA-232 Connector
Figure 24 2070-3B Front Panel
Figure 25 2070-3B Front Panel Rear View
Figure 26 C50S EIA-232 Connector
Figure 27 2070-3C Front Panel
Figure 28 2070-3C Front Panel Rear View
Figure 29 C50S EIA-232 Connector
Figure 30 C50S EIA-232 Connector
Figure 31 2070-4A Front and Back
Figure 32 Pinout for PS1 Connector
Figure 33 Pinout for PS2 Connector
Figure 34 VME Cage with CPU Installed in Spot 1
Figure 35 2070-6B Front and Back
Figure 36 Pinout for C2 and C20 Connector
Figure 37 2070-7A Front and Side View
Figure 38 Pinout for C21 and C22 Connector

I. Overview

The 2070 controller is a modular, multipurpose controller used for traffic control operations. Essentially, it is the "brains" of the majority of field traffic control devices, such as traffic signals, ramp metering, and changeable message signs.

A. Communications Structure

The communications structure of the 2070 is based on having every port number assigned to certain protocol standards so that it can be used by a variety of applications. Current assigned ports for serial communications are:

Serial Port 1 (/SP1)	Used for EIA-232 or 485 communications
Serial Port 2 (/SP2)	Used for EIA-232 or 485 communications
Serial Port 3 (/SP3)	Used for EIA-232 or 485 communications
Serial Port 4 (/SP4)	Used for EIA-232 or 485 communications Also used with the C50S connector on front panel
Serial Port 5 (/SP5)	Used for EIA-232 or 485 communications And converted for parallel uses for field I/O on 2070-2A
Serial Port 6 (/SP6)	Used to communicate with front panel
Serial Port 8 (/SP8)	Used for EIA-485 communications (2070L or 2070LC Only)
Ethernet	Used for 10baseT Ethernet communication (2070L or 2070 LC Only)

B. Motherboard Structure

The motherboard for the controller has 5 positions, which are referred to as slots, and are labeled A1 to A5. At each of these slots a 96-pin connectors attaches the 2070 card (ie. 2070-1B) to the motherboard, and allows it to communicate with other components within the controller. Current assignment of these slots is:

Slot A1	2070-7x or 2070-6x communication cards (/SP3 and /SP4)
Slot A2	2070-7x or 2070-6x communication cards (/SP1 and /SP2)
Slot A3	2070-2A or 2070-2B Field I/O cards
Slot A4	Covered by 2070-2A, and reserved for future use
Slot A5	2070-1A Transition Board or 2070-1B CPU card
VME Chassis	2070-1A CPU card (VME position 1)

Figure 1 shows how the ports are arranged on the motherboard and how the motherboard slots are arranged. Additionally notice that the front panel has Serial Port 6 and Serial Port 4 Assigned. Serial Port 6 is used to display information to the front panel and receive information from the front panel keyboard. Serial Port 4 is used with the C50S connector to allow someone to connect to the 2070 operating system using a laptop. A description of how to connect to the 2070 operating system is shown in section C.

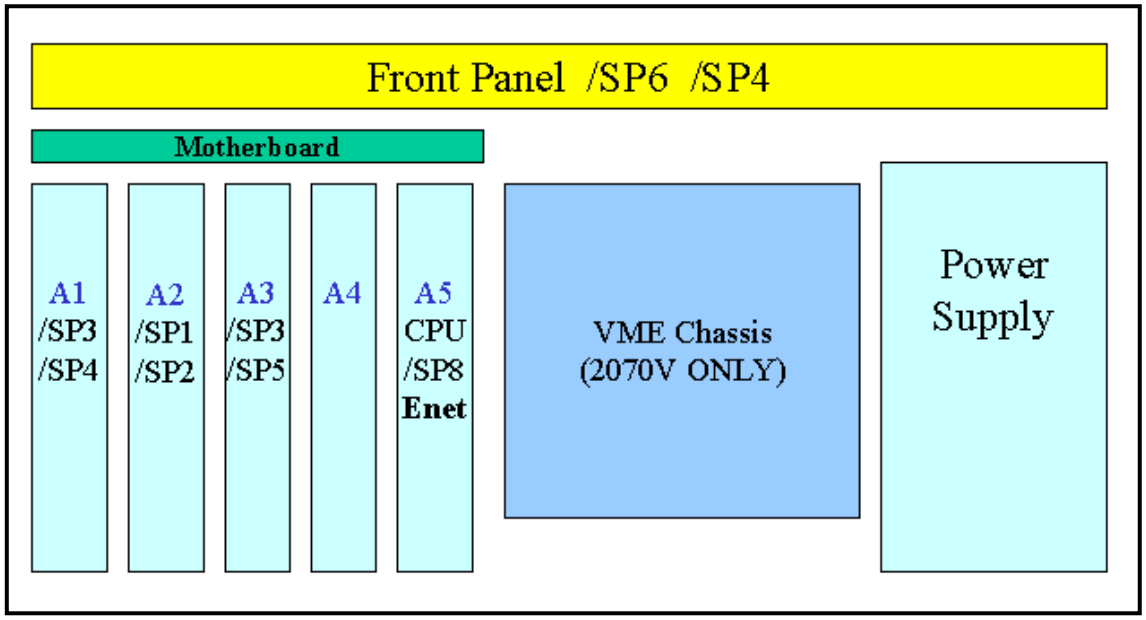


Figure 1 2070 Architecture

C. Operating System

The 2070 uses a real-time operating system embedded in to the controller called OS-9. This operating system, written by Microware (www.microware.com) is similar to MS-DOS, and has a text based command structure. Unlike your desktop computer, however, there is no QUERTY keyboard, and no monitor. While there is a keypad and LCD display on the front of the controller, this input and output (I/O) is used primarily for receiving using selections for program applications as well as applications the ability to display program menus for users.

General interface with the operating system is by using a *direct mode communication cable* (pinout shown in appendix B) running from a host computer on serial port 1 to the C50S port of the 2070 and using program terminal emulation software such as Hyperterminal. An explanation as to how to set up Hyperterminal is in appendix C for those that are not familiar with this program.

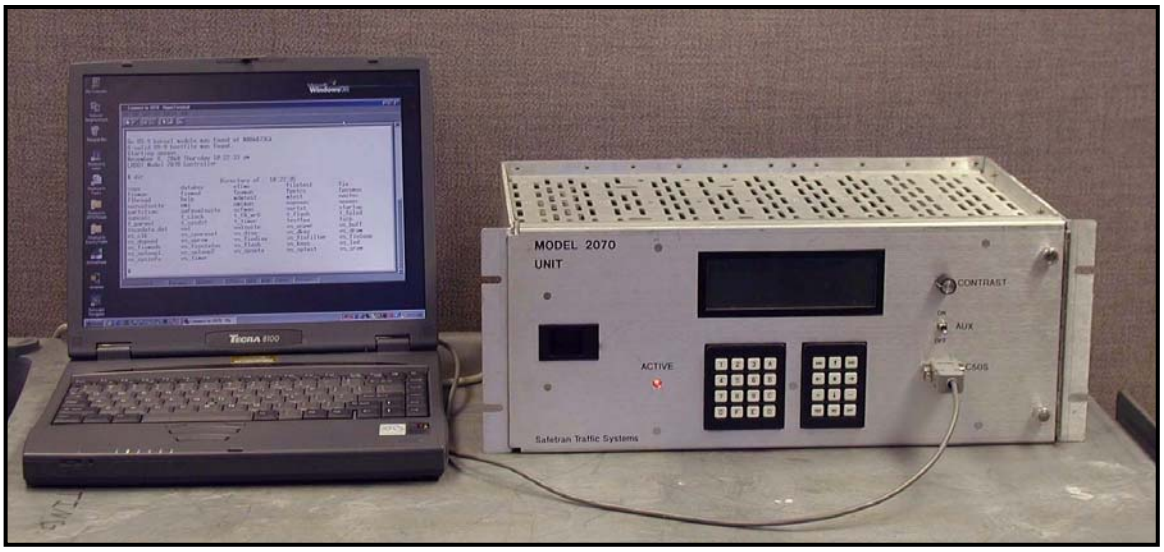


Figure 2 Connecting Laptop Connected to 2070

The Default Protocols to use for terminal emulation are:

Baud = 9600
Parity = None
Data Bits = 8
Stop Bits = 1
Flow Control = None

Once communication is established, commands can be entered in a format similar to MS-DOS. A summary of available commands is listed in appendix D.

Figure 3 shows a typical 2070 bootup as seen in the Hyperterminal terminal emulator. Normally, any OS-9 shell command can now be entered at the "$" prompt.

DANGER

Never enter terminal commands while the controller is controlling an active intersection as serious damage or injury may result

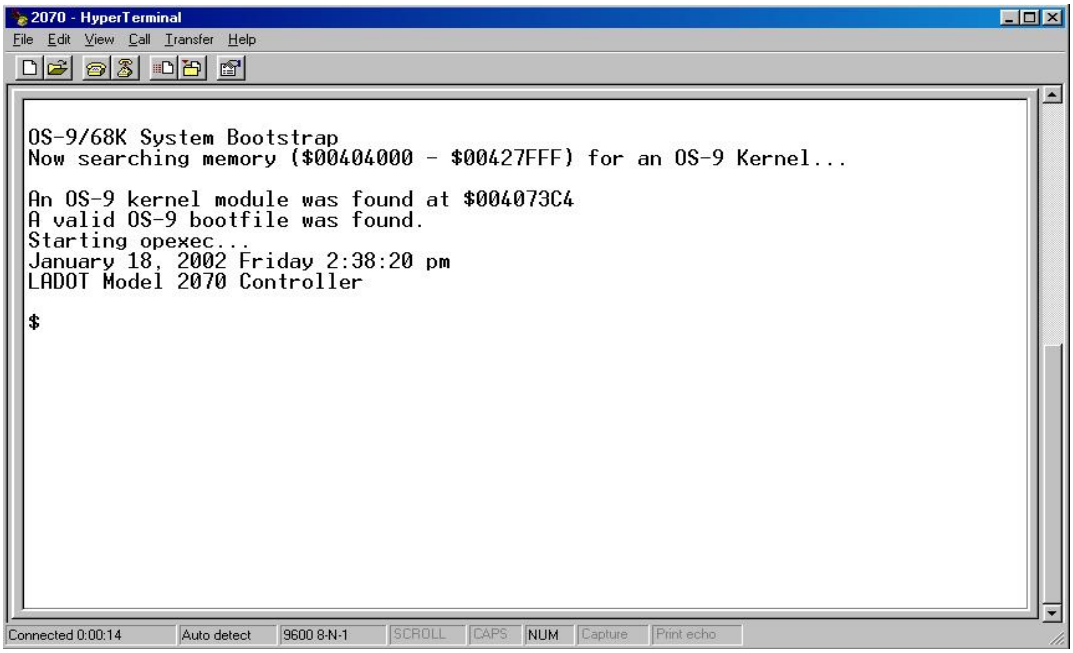


Figure 3 Hyperterminal Terminal Shell

II. 2070 Models

There are currently three standard versions of 2070. According to the copy of the Caltrans March 29, 2002 TEES these versions are:

2070 V unit Provides directly driven VME and mates to the 170 and ITS cabinets. It consists of :

Unit Chassis, 2070-1A Two Board CPU, 2070-2A Field I/O, 2070-3A Front Panel, 2070-4A Power Supply and 2070-5 VME Cage Assembly

Figure 4 2070 V Unit Front View

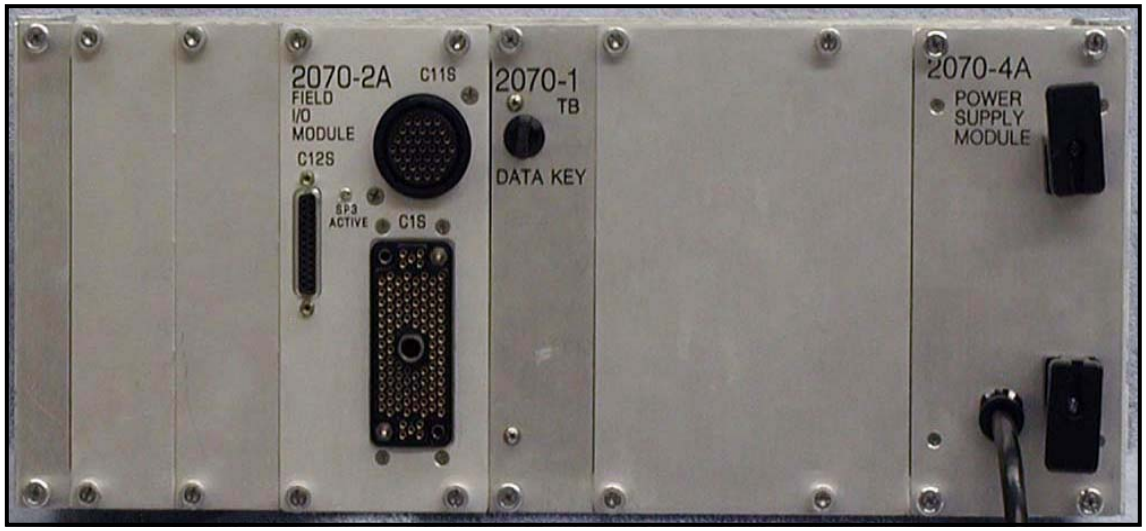


Figure 5 2070 V Unit Back View

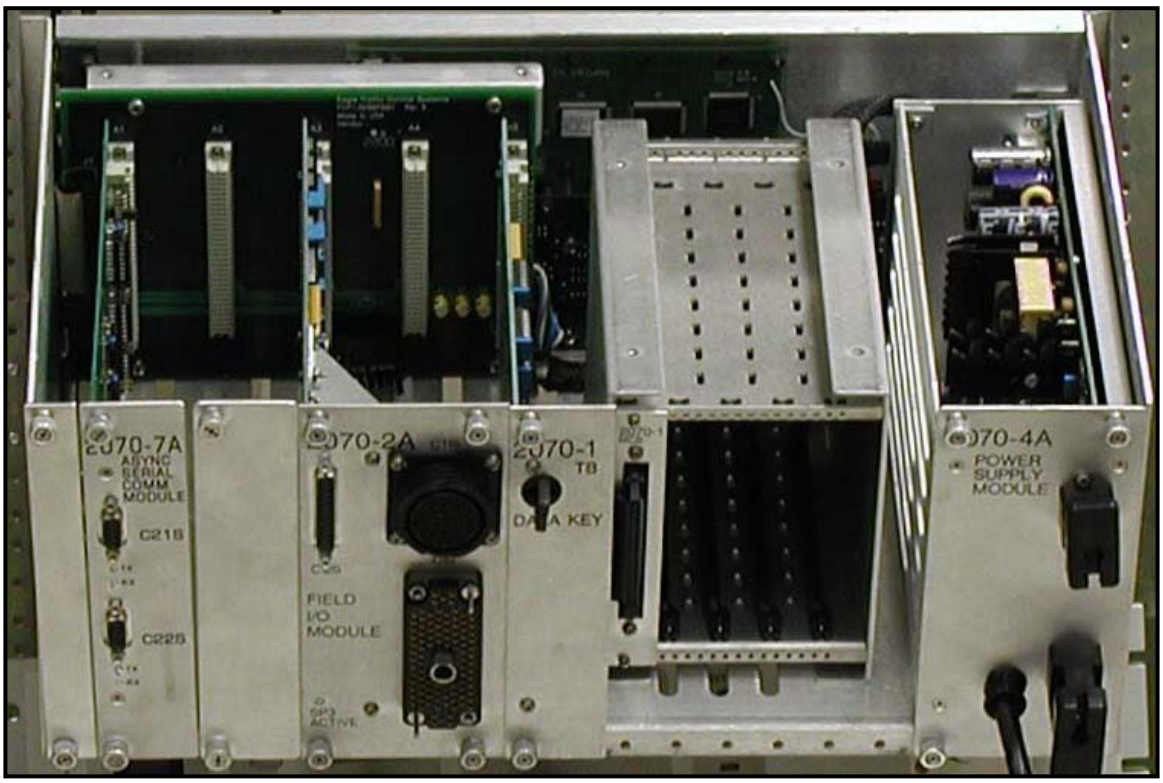


Figure 6 2070 V Unit Back – Top View With Additional 7A Card Installed
Top Cover of Chassis Removed

2070 L unit Lite unit mates to the 170 and ITS cabinets. It consists of:

Unit Chassis, 2070-1B CPU, 2070-2A field I/O for 170 cabinets or 2070-2B for ITS cabinets, 2070-3B Front Panel, and a 2070-4A or 2070-4B power Supply

Figure 7 2070 L Unit Front View

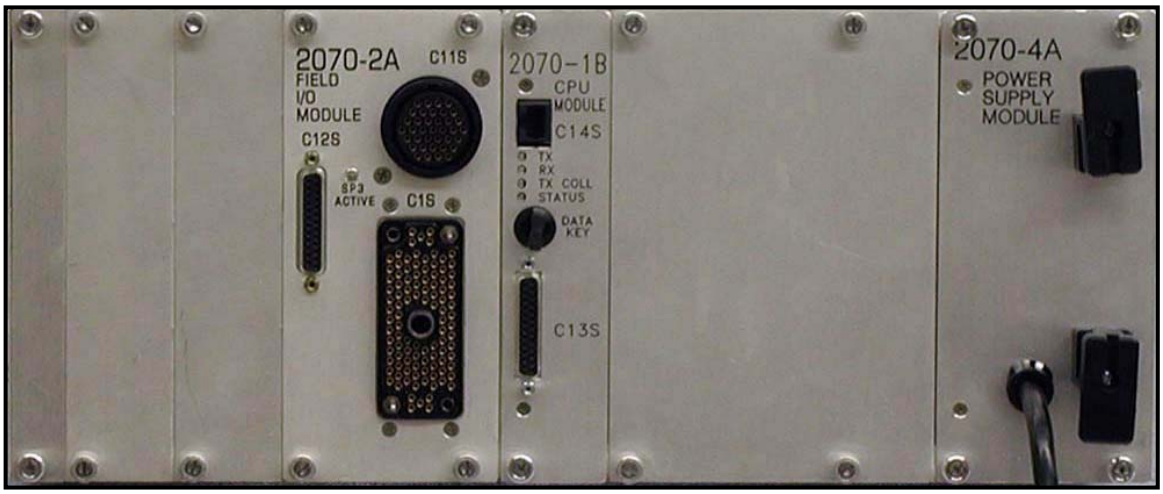


Figure 8 2070 L Unit Back View

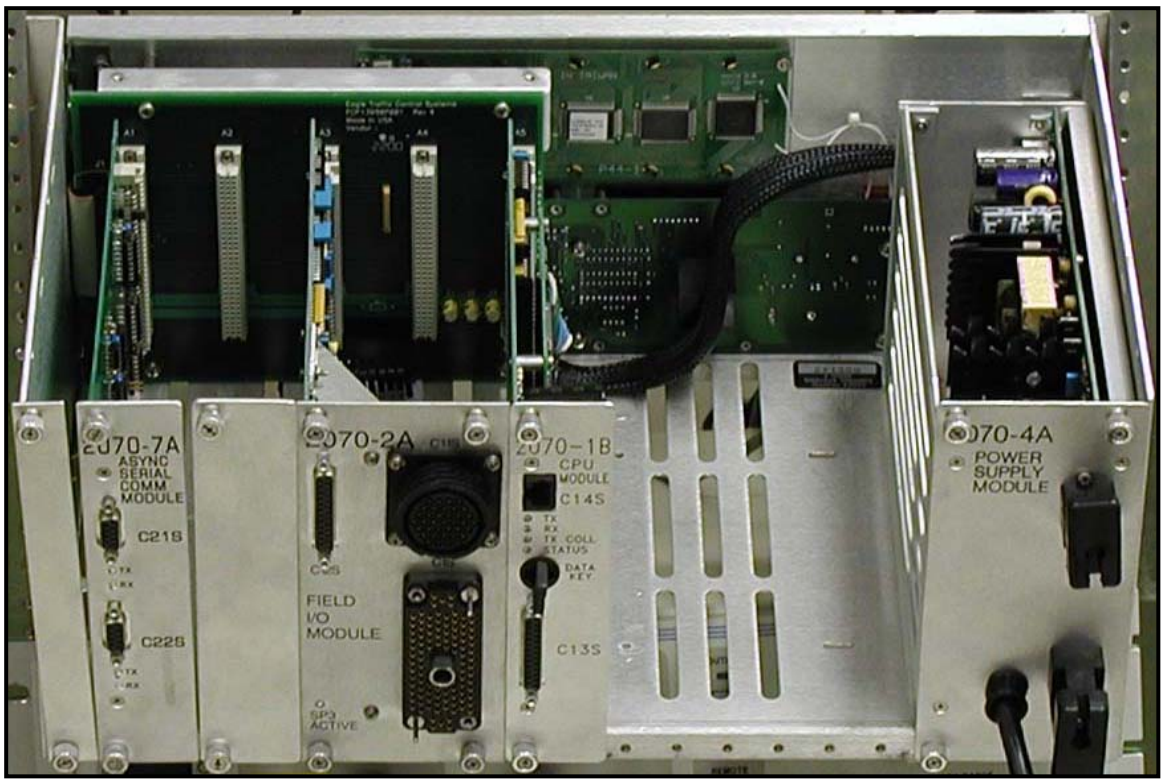


Figure 9 2070 L Unit Back - Top View With Additional 7A Card Installed Top Cover of Chassis Removed

2070 LC Unit Lite unit mates to the ITS cabinet only. It consists of:

Unit Chassis, 2070-1B CPU, 2070-2B for ITS cabinets, 2070-3B Front Panel, and a 2070-4A or 2070-4B power Supply

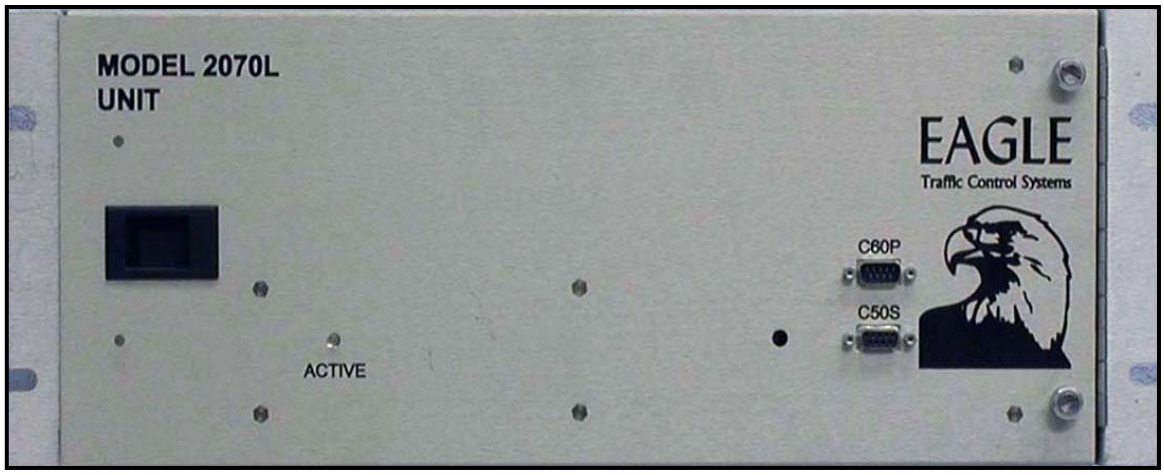


Figure 10 2070 LC Unit Front View

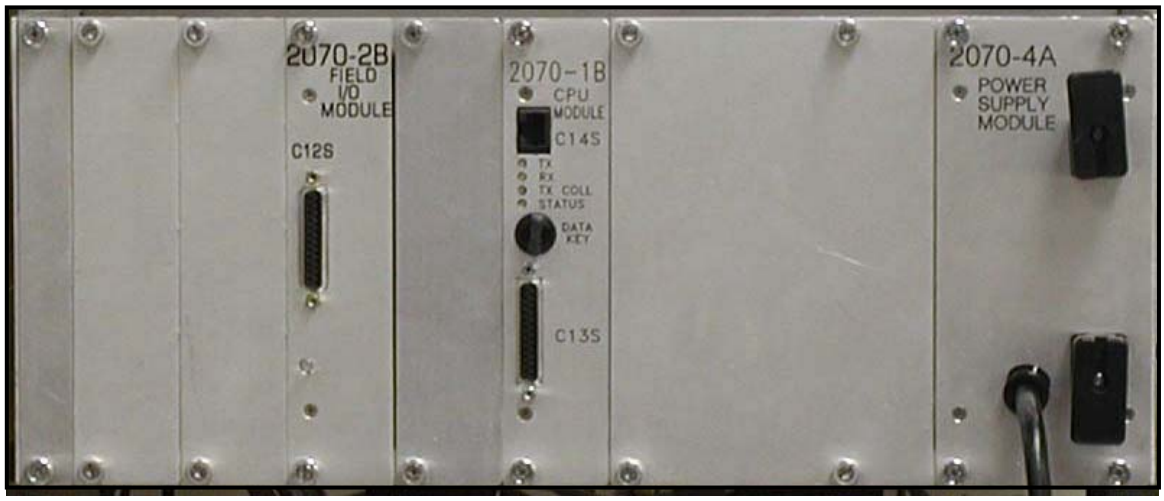


Figure 11 2070 LC Unit Back View

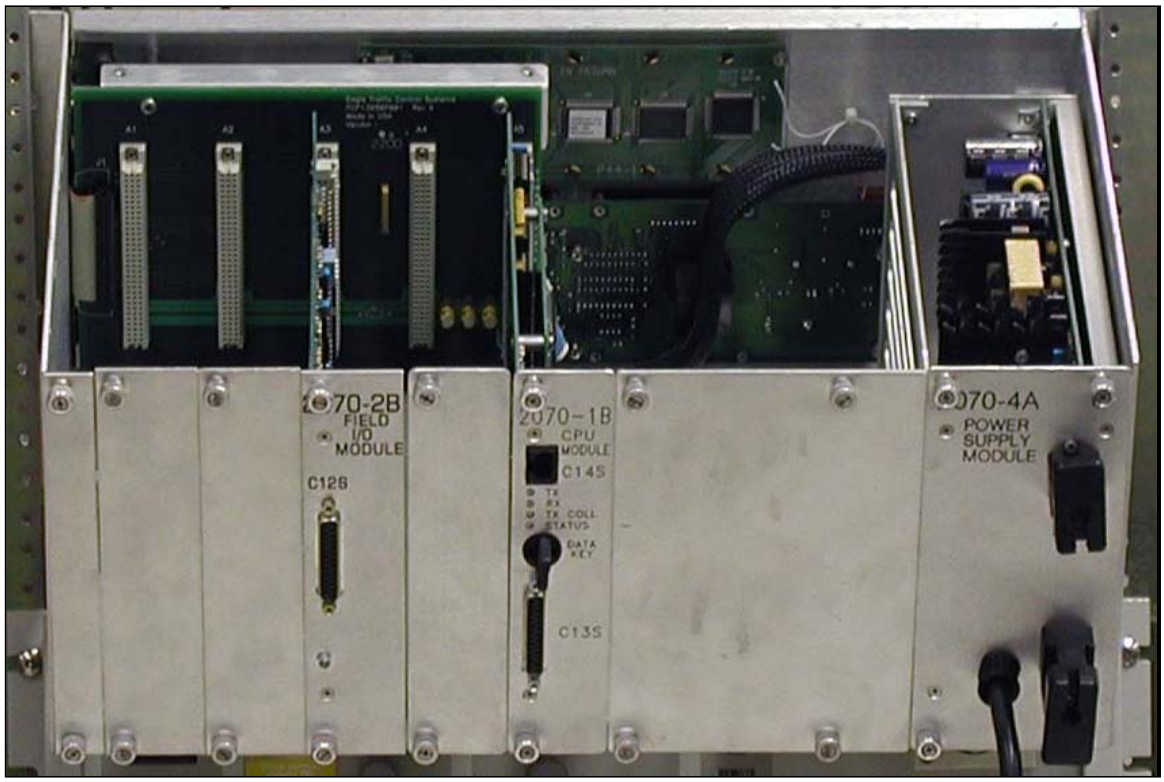


Figure 12 2070 LC Unit Back - Top View
Top Cover of Chassis Removed

III. 2070 Component Cards

All of component cards which can be installed in a 2070 are completely interchangeable. This means that you can add a RS-232 interface to you controller by merely adding an 2070-7A interface card. You can even turn a 2070V into a 2070L be merely replacing the 2070-1A CPU board with a 2070-1B CPU board. If, for example, the Field I/O module goes bad you only need to replace that unit and your controller is up and running perfectly again.

A. CPU card 2070-1A

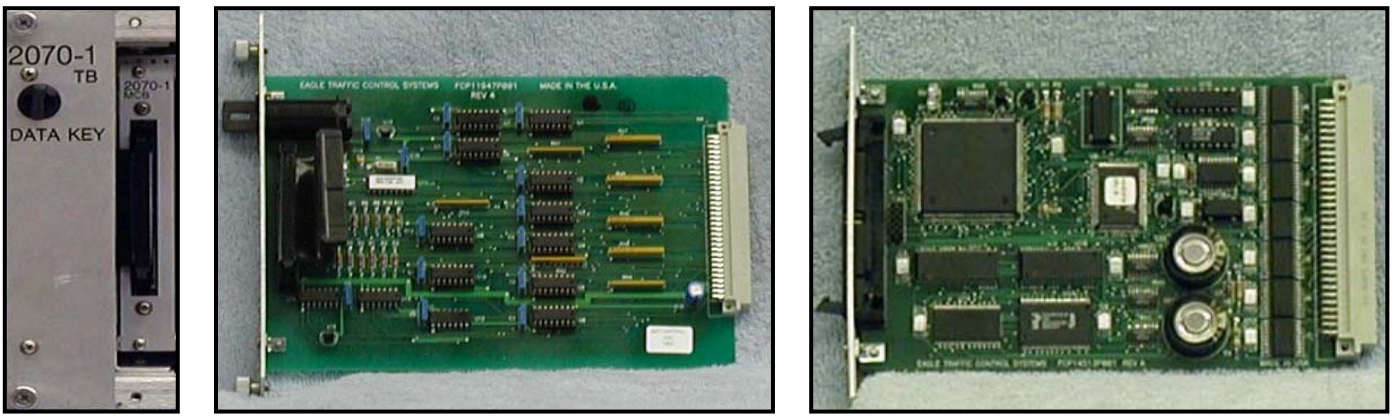


Figure 13 2070 – 1A Transition Board Front (Left) and Side View (Middle)
2070 – 1A CPU Located in VME Chassis (Right)

The 2070 CPU card houses the central processing unit for the controller. It is located in the VME chassis and is where the operating system and application programs reside. Its functions are very similar to you standard PC computer, in that it has an operating system, drive directories, and similar command structure. In addition, there is a transition board, located in slot A5, which houses the data key, and allows communication with other devices located in slot A1 thru A4.

Features:

Motorola 68360 Processor running at 25 Mhz
4 MB DRAM Memory
512 KB SRAM Memory – Battery Backed for 30 Days
4 MB Flash Memory
No Eproms, All user programs downloaded into controller
Real Time Clock

B. CPU card 2070-1B

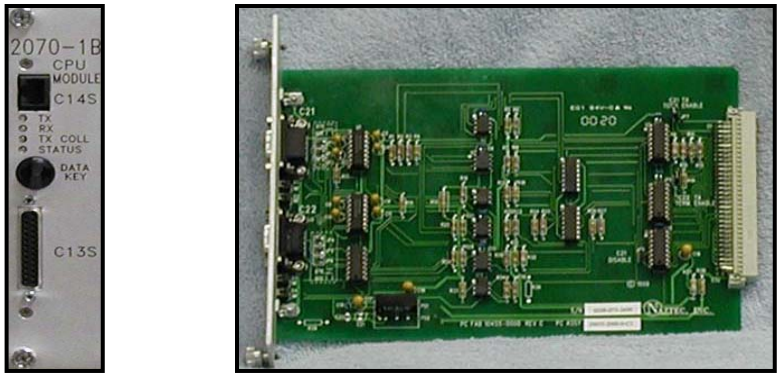


Figure 14 2070 – 1B Front and Side View

Located in slot A5, the 2070 CPU card houses the central processing unit for the controller. It is where the operating system and application programs reside. Its functions are very similar to you standard PC computer, in that it has an operating system, drive directories, and similar command structure.

Features:

Motorola 68360 Processor running at 25 Mhz
4 MB DRAM Memory
512 KB SRAM Memory – Battery Backed for 30 Days
4 MB Flash Memory
Ethernet Port (Standard TCP/IP Proticols)
EIA-485 Port (/SP8)
No Eproms, All user programs downloaded into controller
Real Time Clock

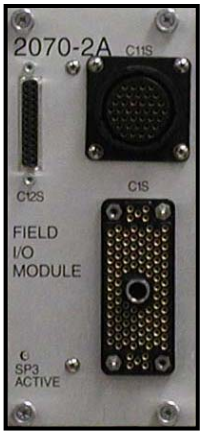
C14S Pin Assignments	
Pin	Function
1	TX +
2	TX -
3	RX +
4	NA
5	NA
6	RX -
7	NA
8	NA

Figure 15
Pinout For C14S Ethernet Connector

C13S Pin Assignments			
Pin	Function	Pin	Function
1	SP8 TX +	14	SP8 TX -
2	SP8 RX +	15	SP8 TX -
3	SP8 TXC +	16	SP8 TXC -
4	SP8 RXC +	17	SP8 RXC -
5	SP8 RTS +	18	SP8 RTS -
6	SP8 CTS +	19	SP8 CTS -
7	SP8 DCD +	20	SP8 DCD -
8	NA	21	NA
9	LINESYNC +	22	LINESYNC -
10	NRESET +	23	NRESET -
11	POWERDOWN +	24	POWERDOWN -
12	+5 VDC	25	EQUIP GND
13	DC DND #2		

Figure 16
Pinout For C13S EIA-485 Connector

C. Field I/O Card 2070-2A



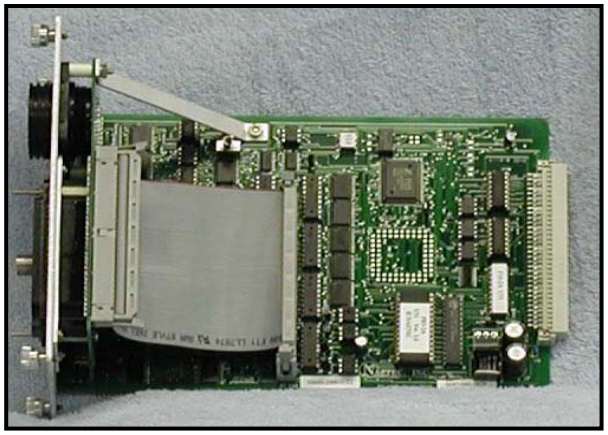


Figure 17 2070 – 2A Front and Side View

Located in slot A3 and A4, the 2070 – 2A Field I/O card translates the CPU output into parallel outputs, which allows the 2070 to be compatible with the 332 cabinets. Additionally this component card allows EIA-485 communication for ports SP3 and SP5 through the C12S connector.

Additionally, the C12S connector provides a 485 outputs for serial ports 3 and 5 to allow control of an ITS cabinet.

It is recommended to shut off SP3 though the internal card switches before using SP3 on either a 2070-6A or 2070-7A card (Card in Slot A1)

Features:

64 Parallel Inputs and Outputs
EIA – 485 Outputs for /SP3 and /SP5
Optically isolated from other components

C12S Pin Assignments			
Pin	Function	Pin	Function
1	SP5 TX +	14	SP5 TX -
2	SP5 RX +	15	SP5 TX -
3	SP5 TXC +	16	SP5 TXC -
4	SP5 RXC +	17	SP5 RXC -
5	SP3 TX +	18	SP3 TX -
6	SP3 RX +	19	SP3 RX -
7	SP3 TXC +	20	SP3 TXC -
8	SP3 RXC +	21	SP3 RXC -
9	LINESYNC +	22	LINESYNC -
10	NRESET +	23	NRESET -
11	POWERDOWN +	24	POWERDOWN -
12	+5 VDC	25	EQUIP GND
13	DC DND #2		

Figure 18
Pinout For C12S EIA-485 Connector

D. Field I/O Card 2070-2B

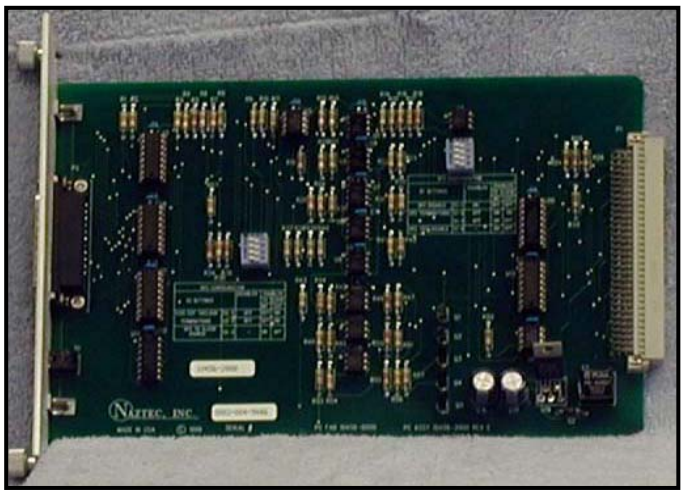


Figure 19 2070 – 2B Front and Side View

Located in slot A3, the 2070 – 2B Field I/O card provides a 485 outputs for serial ports 3 and 5 to allow control of an ITS cabinet.

C12S Pin Assignments			
Pin	Function	Pin	Function
1	SP5 TX +	14	SP5 TX -
2	SP5 RX +	15	SP5 TX -
3	SP5 TXC +	16	SP5 TXC -
4	SP5 RXC +	17	SP5 RXC -
5	SP3 TX +	18	SP3 TX -
6	SP3 RX +	19	SP3 RX -
7	SP3 TXC +	20	SP3 TXC -
8	SP3 RXC +	21	SP3 RXC -
9	LINESYNC +	22	LINESYNC -
10	NRESET +	23	NRESET -
11	POWERDOWN +	24	POWERDOWN -
12	+5 VDC	25	EQUIP GND
13	DC DND #2		

Figure 20
Pinout For C12S EIA-485 Connector

E. 2070 – 3A Front Panel

Figure 21 2070-3A Front Panel

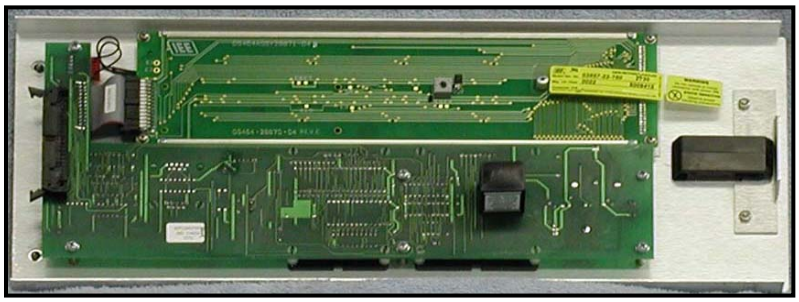


Figure 22 2070-3A Front Panel – Rear View

The 2070 Front Panel Interface allows users to communicate with software applications.

Features:

- Large 4x40 character LCD
- Backlight Display
- 4x4 Keypad for alphanumeric entry
- 4x3 Keypad for Cursor control and Symbol Entry
- C50S Connector for software installation and management (/SP4)
- Auxiliary Switch
- CPU Active LED
- Bell

C50 S Connector Outputs	
Pin	Function
1	C-50 Enable
2	SP4 RX
3	SP4 TXD
4	NA
5	DC GND #1
6	NA
7	NA
8	NA
9	NA

Figure 23
C50S EIA-232 Connector

F. Front Panel Interface 2070–3B

Figure 24 2070-3B Front Panel

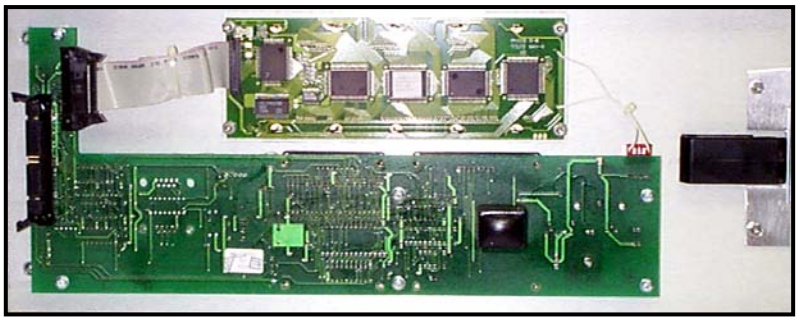


Figure 25 2070 – 3B Front Panel Rear View

The 2070 Front Panel Interface allows users to communicate with software applications.

Features:

¼ inch 8x40 LCD
Backlight Display
4x4 Keypad for alphanumeric entry
4x3 Keypad for Cursor control and Symbol Entry
C50S Connector for software installation and management (/SP4)
Auxiliary Switch
CPU Active LED
Bell

C50 S Connector Outputs	
Pin	Function
1	C-50 Enable
2	SP4 RX
3	SP4 TXD
4	NA
5	DC GND #1
6	NA
7	NA
8	NA
9	NA

Figure 26

C50S EIA-232 Connector

G. 2070 – 3C Front Panel

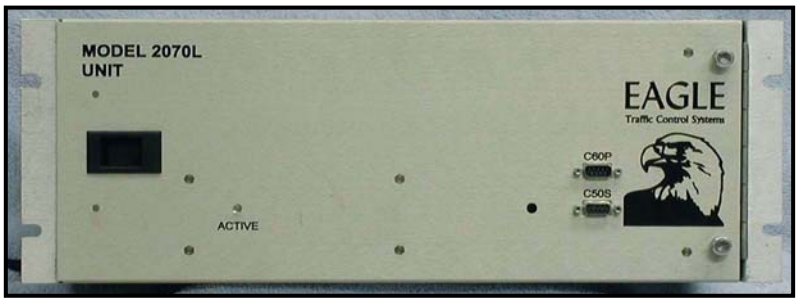


Figure 27 2070 – 3C Front Panel

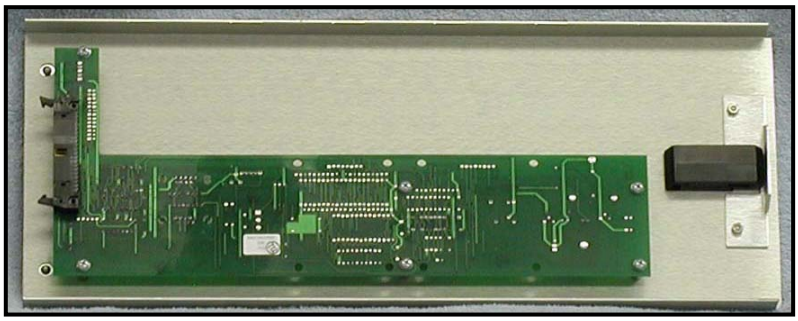


Figure 28 2070 – 3C Front Panel Rear View

The 2070 Front Panel Interface allows users to communicate with software applications using a laptop connected to the C60S connector. By eliminating the keypad and the LCD displays, the cost of the front panel can therefore be reduced significantly.

C50 S Connector Outputs	
Pin	Function
1	C-50 Enable
2	SP4 RX
3	SP4 TX
4	NA
5	DC GND #1
6	NA
7	NA
8	NA
9	NA

Figure 29
C50S EIA-232 Connector

C60 S Connector Outputs	
Pin	Function
1	NA
2	SP6 RX
3	SP6 TX
4	NA
5	DC GND #1
6	NA
7	CPU Reset
8	NA
9	NA

Figure 30
C50S EIA-232 Connector

H. Power Supply 2070-4A and 2070-4B (4A Shown)

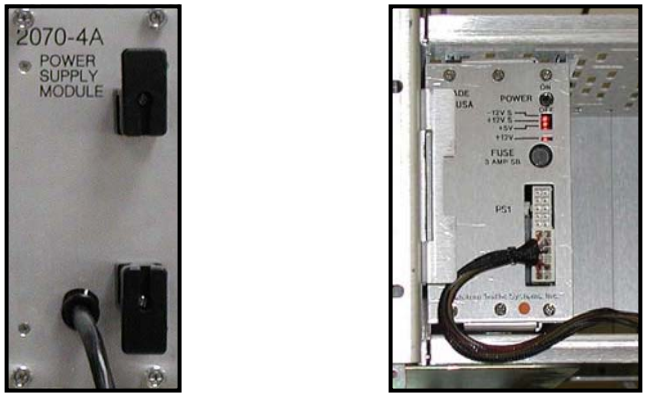


Figure 31 2070 – 4A Front and Back

The 2070 power supply is a 4-channel power supply, which supplies power to the controller, as well as timing signals for synchronization, power failure, and system reset.

Features:

½ Second Controller Holdup, to keep system operating during short outages
PS1 Connector used for VME – Not used on 2070L
LED indication of Normal Power Output
Short Out Protection
60 Hz Linesync

2070–4B Power Supply Maximum Power Ratings

Voltage	Maximum Current Load
+5 VDC	3.5 Amps
+12 VDC Serial	0.5 Amps
-12 VDC Serial	0.5 Amps
+12 VDC Isolated	1.0 Amps

Note:
A 2070-4A power supply is rated at 75 watts and may be substituted in place of a 4B unit.

PS 1 Connector Pinout	
Pin	Function
1	+5 VDC
2	+12 VDC Serial
3	-12 VDC Serial
4	DC Ground #1 (+5 & 12 Serial)
5	+5 VDC Standby
6	+5 VDC Sense
7	DC Ground Sense
8	AC Fail (VME)
9	SYSRESET (VME)
10	N/A

Figure 32
Pinout for PS1 Connector

PS 2 Connector Pinout	
Pin	Function
1	+5 VDC
2	+12 VDC Serial
3	-12 VDC Serial
4	DC Ground #1 (+5 & 12 Serial)
5	+5 VDC Standby
6	+12 VDC (Isolated)
7	DC Ground (+12 VDC Isolated Only)
8	Power Down / AC Fail
9	Power Up / SYSRESET
10	Equipment Ground
11	Linesync
12	NA

Figure 33
Pinout for PS2 Connector

I. 2070 – 5 VME Cage

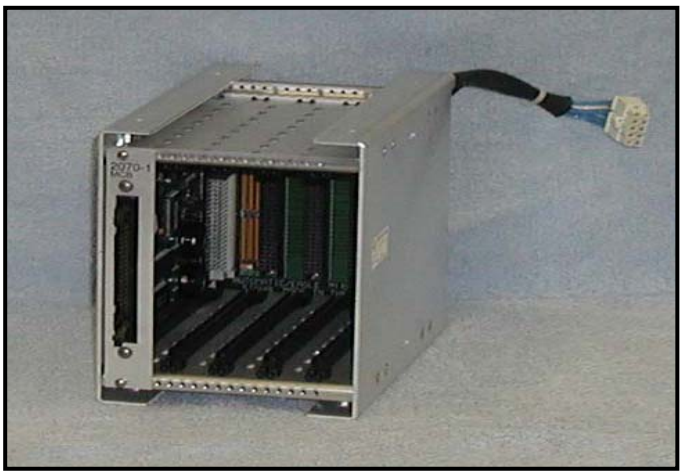


Figure 34 VME Cage with CPU Installed in Spot 1

The 2070 VME (Versa Module Europe) cage is used to allow the 2070 to be compatible with the worldwide standard of hardware available for hardened computers used by the military, petrochemical and robotic industries.

Features:

Expansion via parallel VME bus
3U half-height with 96 pin connector
2070-1A CPU occupies one slot, 4 spare expansion slots
Hundreds of standard VME modules from multiple vendors

J. 2070-6B Async/Modem Serial Comm Module

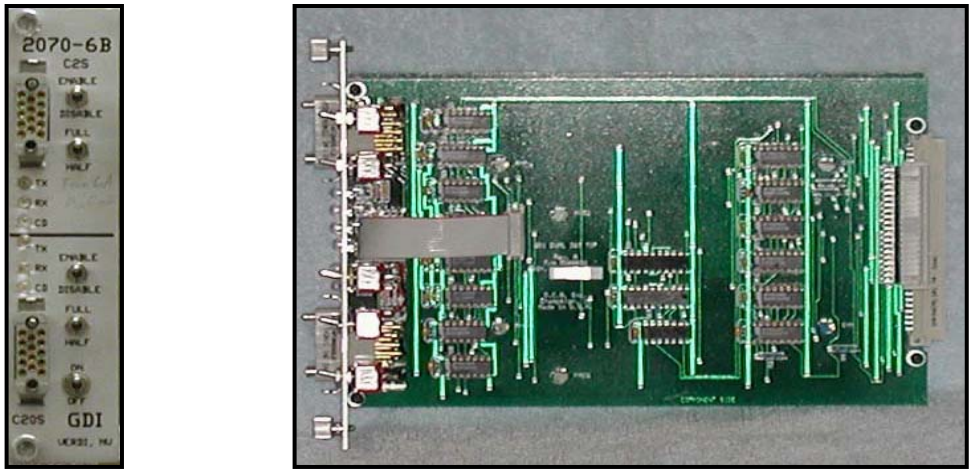


Figure 35 2070-6B Front and Side View

Located in slot A1 or A2, the 2070-6B provides two 9600-baud modem outputs though the C2 and C20 Connectors

Features:

Two 9600 Baud Modems
Front Panel Switches for Full/Half Duplex for each Modem
Each Modem section can be individually enabled/disabled from front Panel switches
The entire Modem can be powered down for live insertion/removal
170 compatible M14 connectors on the front panel (C2 and C20)
Front Panel mounted status indicators
Compatible with the GDI 496 Modems
Switch selectable Anti-streaming
All Line Drivers and Receivers are mounted in sockets

C2 and C20 Pin Assignments			
Pin	Function	Pin	Function
A	AUDIO IN	J	RTS
B	AUDIO IN	K	DATA IN
C	AUDIO OUT	L	DATA OUT
D	ISO +5 VDC	M	CTS
E	AUDIO OUT	M	ISO DC GROUND
F	NA	P	NA
H	CD	R	NA

Figure 36
Pinout for C2 and C20 Connector

K. 2070 – 7A EIA-232 Serial Interface Card

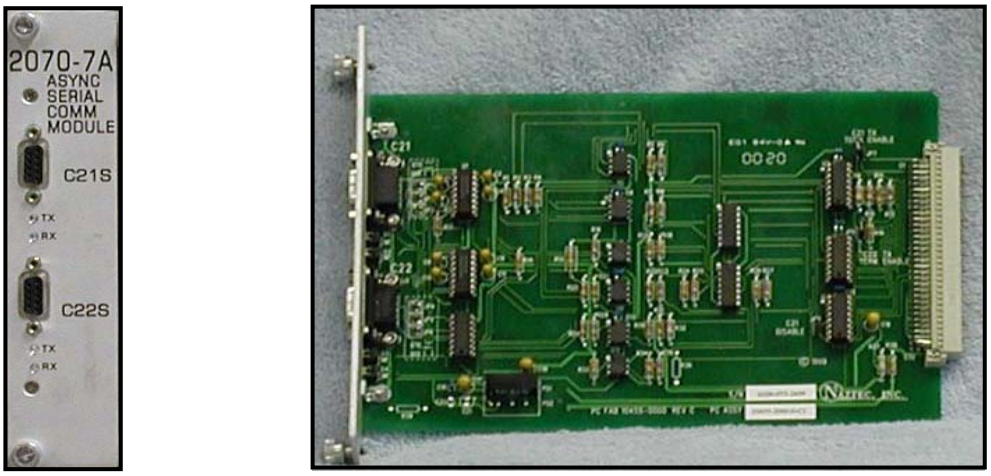


Figure 37 2070 – 7A Front and Side View

Located in slot A1 or A2, the 2070-7A provides two EIA-232 serial communications ports for asynchronous communication with serial devices such as a modem, or a computer.

Features:

Baud Rates – 1200,2400, 4800, 9600, 19200, and 38400
Optical Isolation
Works as Serial Ports 1 and 2 when in Slot A1
Works as Serial Ports 3 and 4 when in Slot A2
Uses Standard 9 Pin D Connector

2070 – 7A (DB-9M)	
C21 and C22 Connector Pinout	
Pin	Function
1	DCD
2	RXD
3	TXD
4	NA
5	ISO DC GND
6	NA
7	RTS
8	CTS
9	NA

Figure 38
Pinout for C21 and C22 Connector

Appendix

A. Glossary

Operating System:

An operating program is the supervisory program what allows applications programs to run. Operating systems perform basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers

Programming Applications:

An executable program, which is executed by the operating system

Real Time Operating System:

A real-time operating system (RTOS) is an operating system that guarantees a certain capability within a specified time constraint. Essentially, this means that the operating system will always respond to an input condition. A non-real time operating system such as Windows 98, on the other hand, does have this requirement, and is prioritized for operating system processes which can prevent the computer from responding to an input condition (i.e. keyboard locking up)

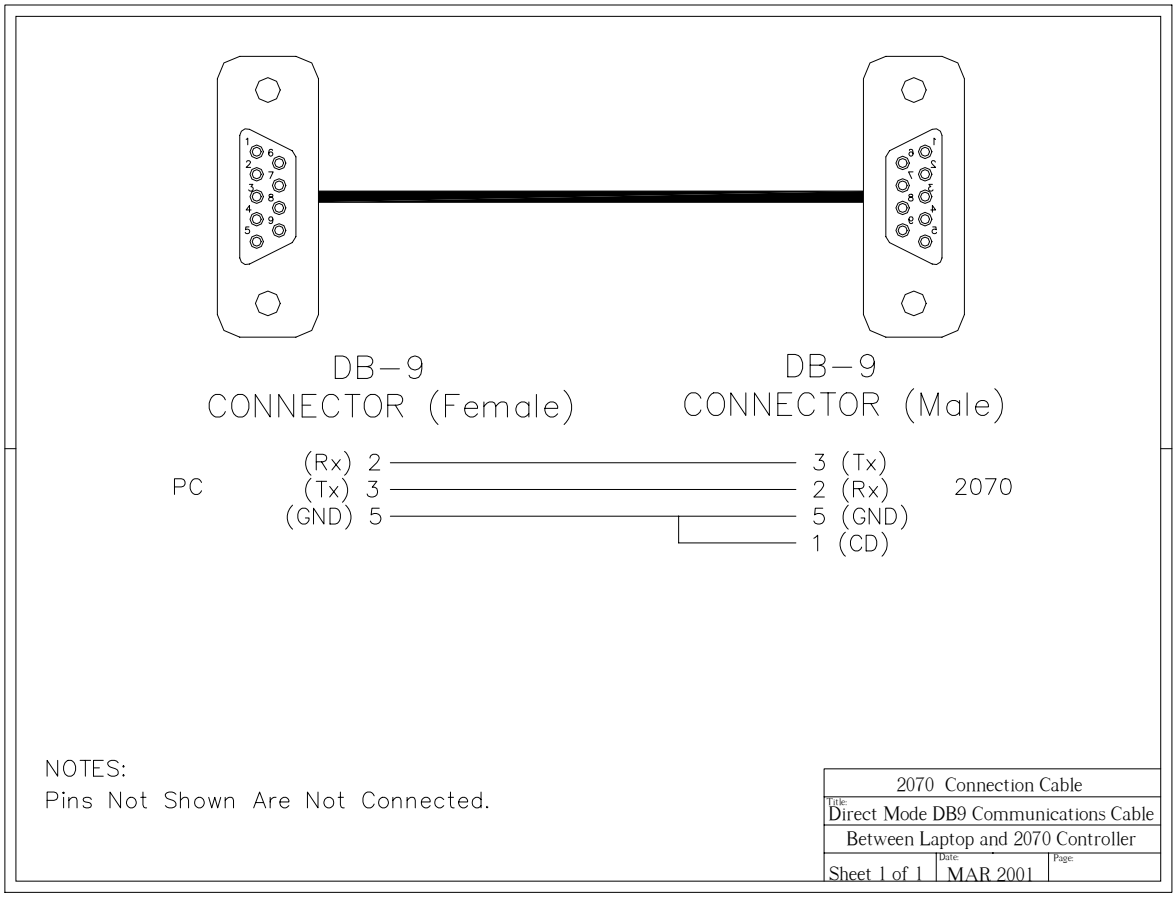
/SP1

Serial Port 1

/SPx

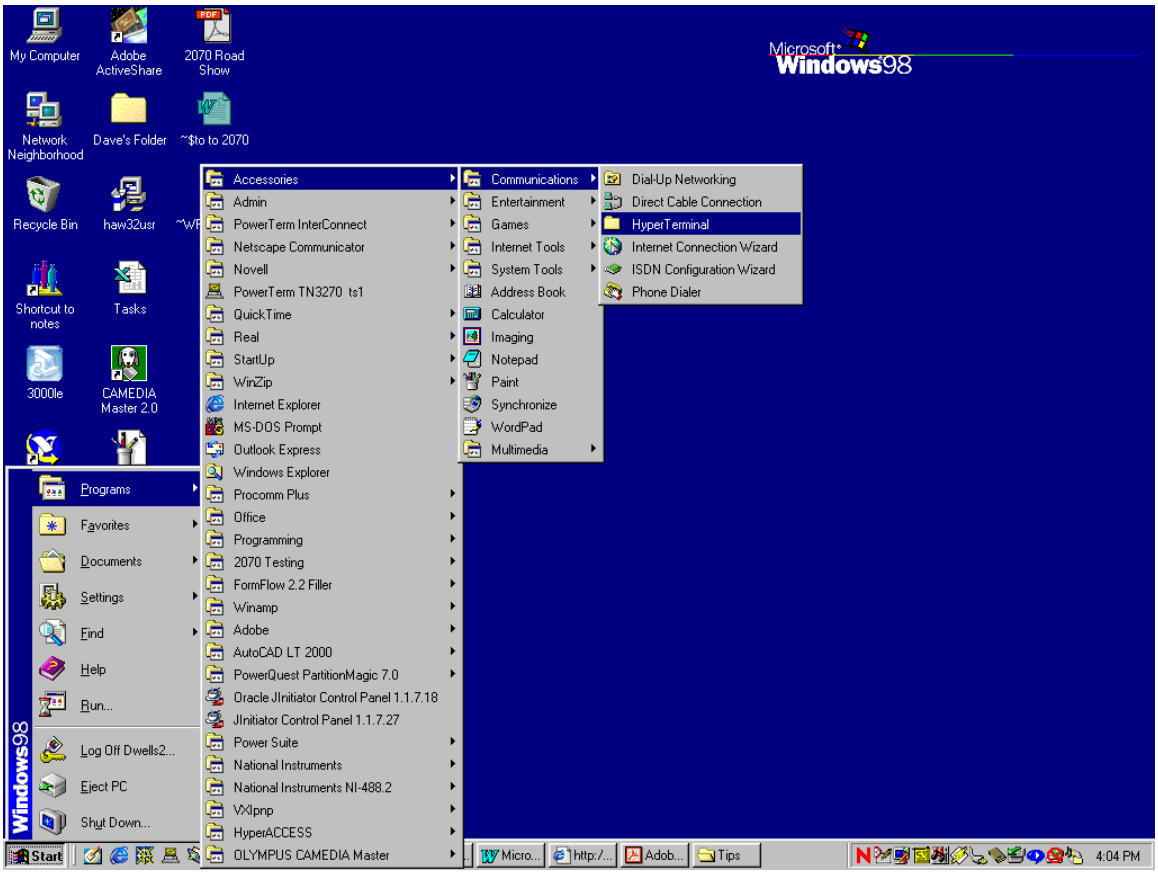
Serial Port x

B. 2070 to PC Connection Wiring Diagram

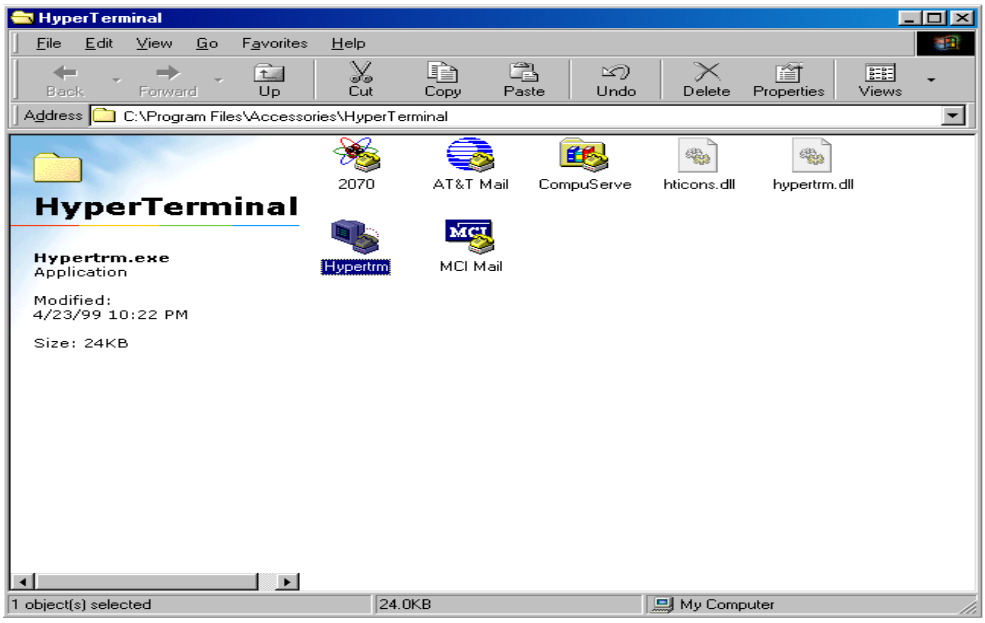


C. Setting Up Hyperterminal To Communicate With A 2070 Controller

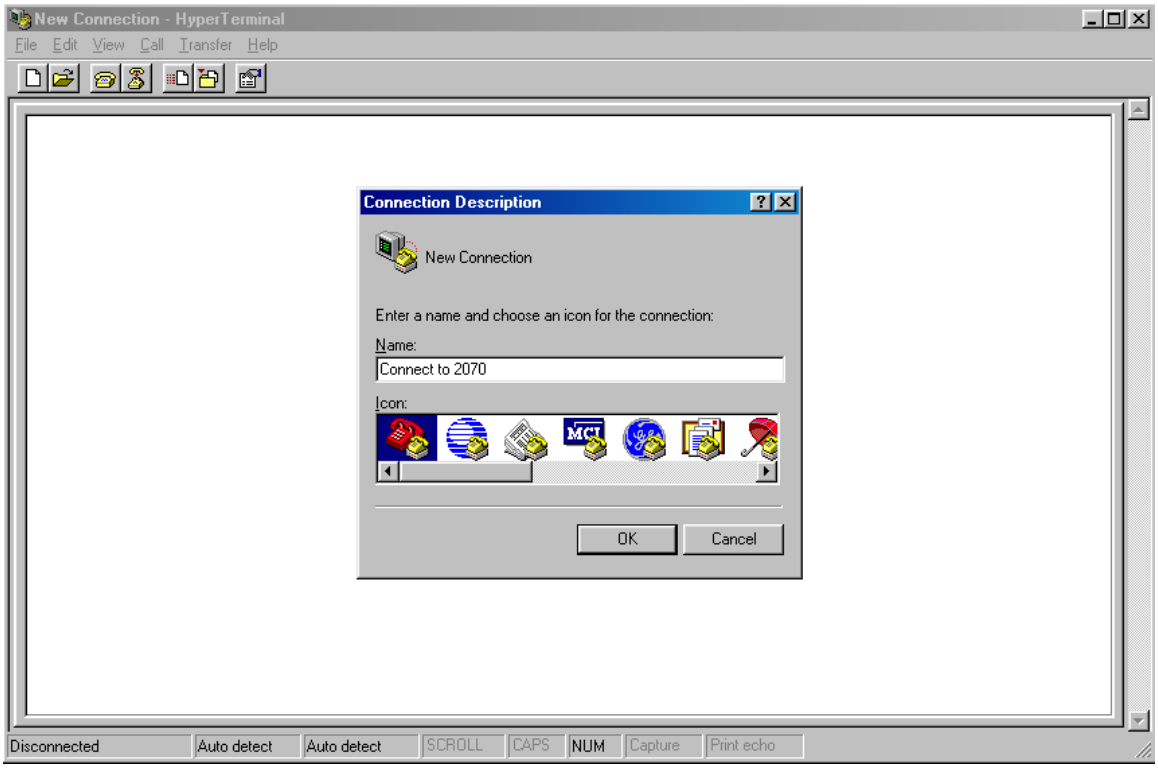
1. Using the start menu open the hyperterminal program



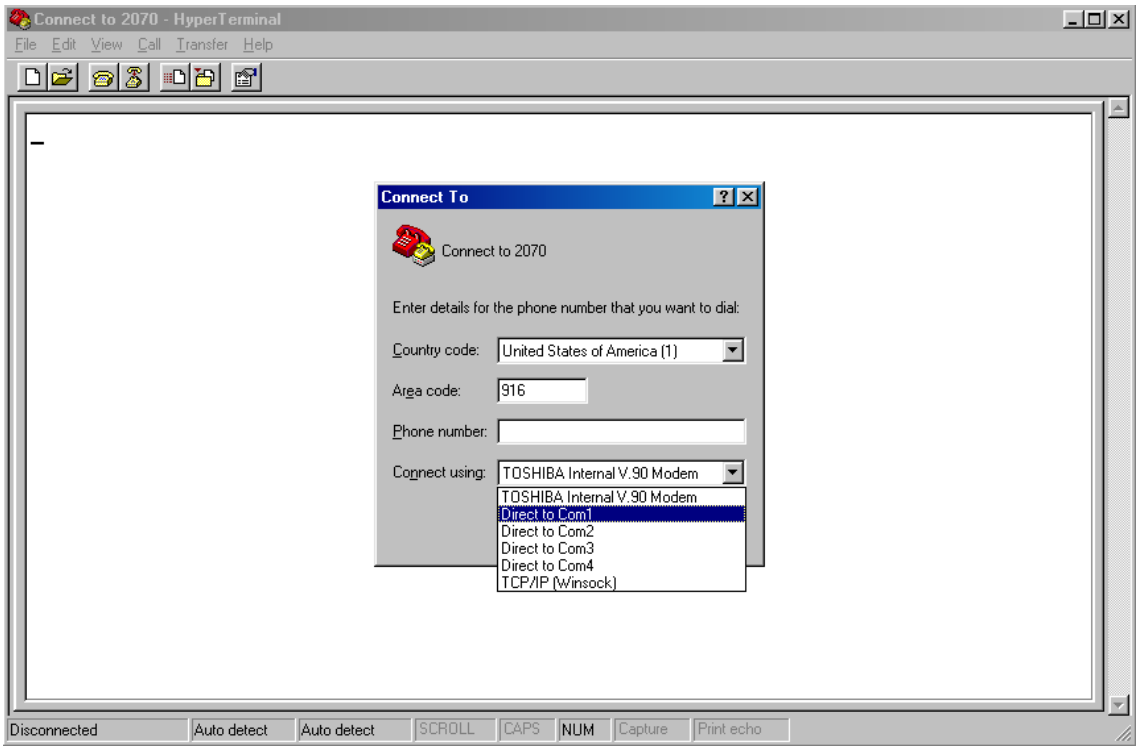
2. Select the Hyperterm icon



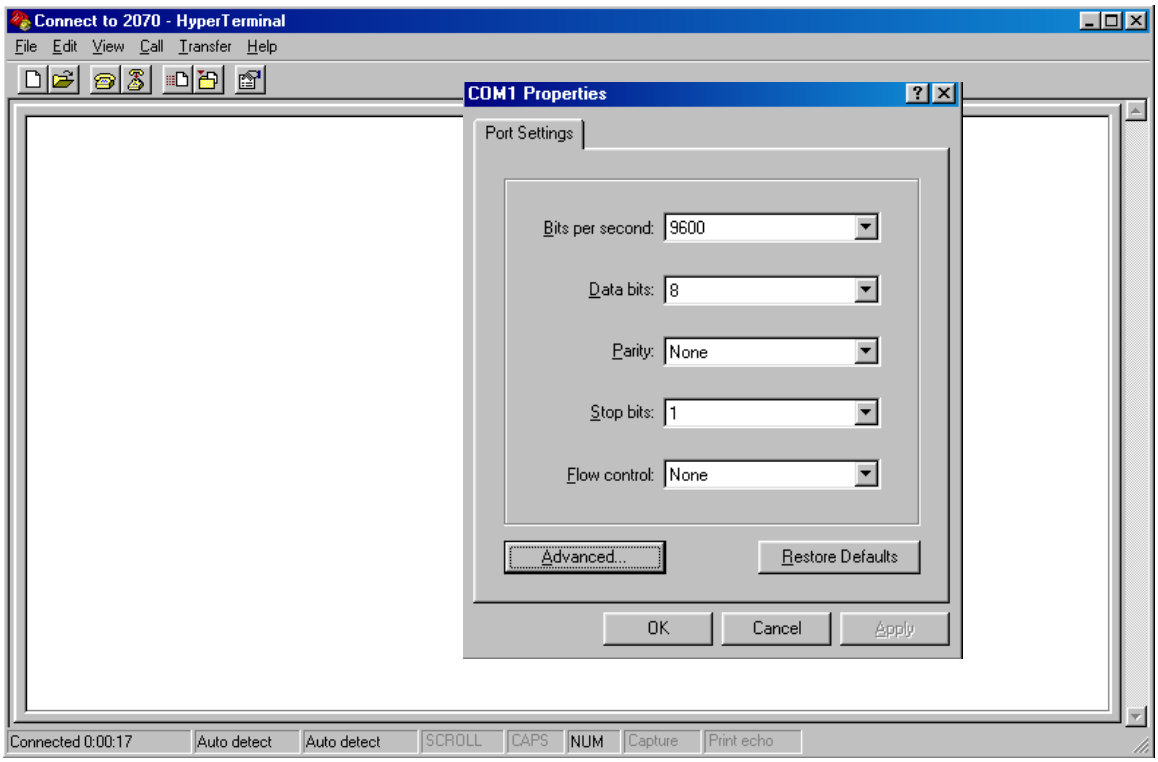
3. Create the name for your connection, and select the icon that you want to represent that connection.



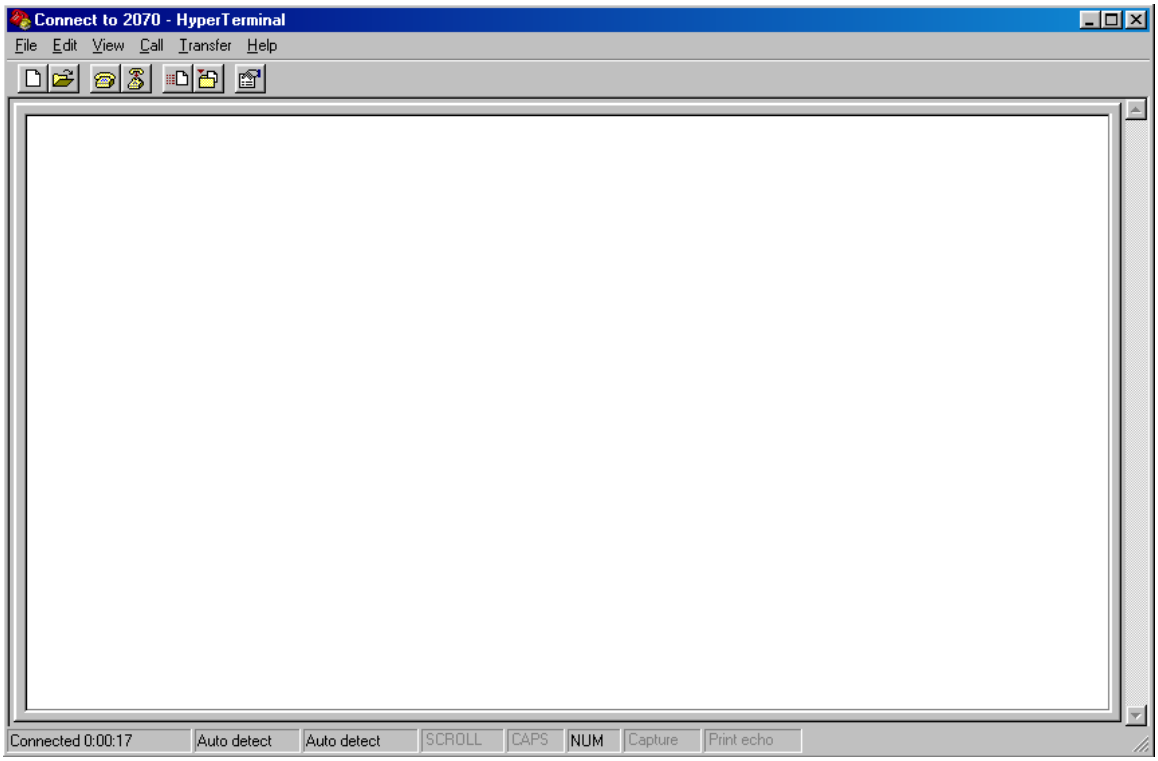
4. Under the “connect using” selector, select “Direct to Com1”.



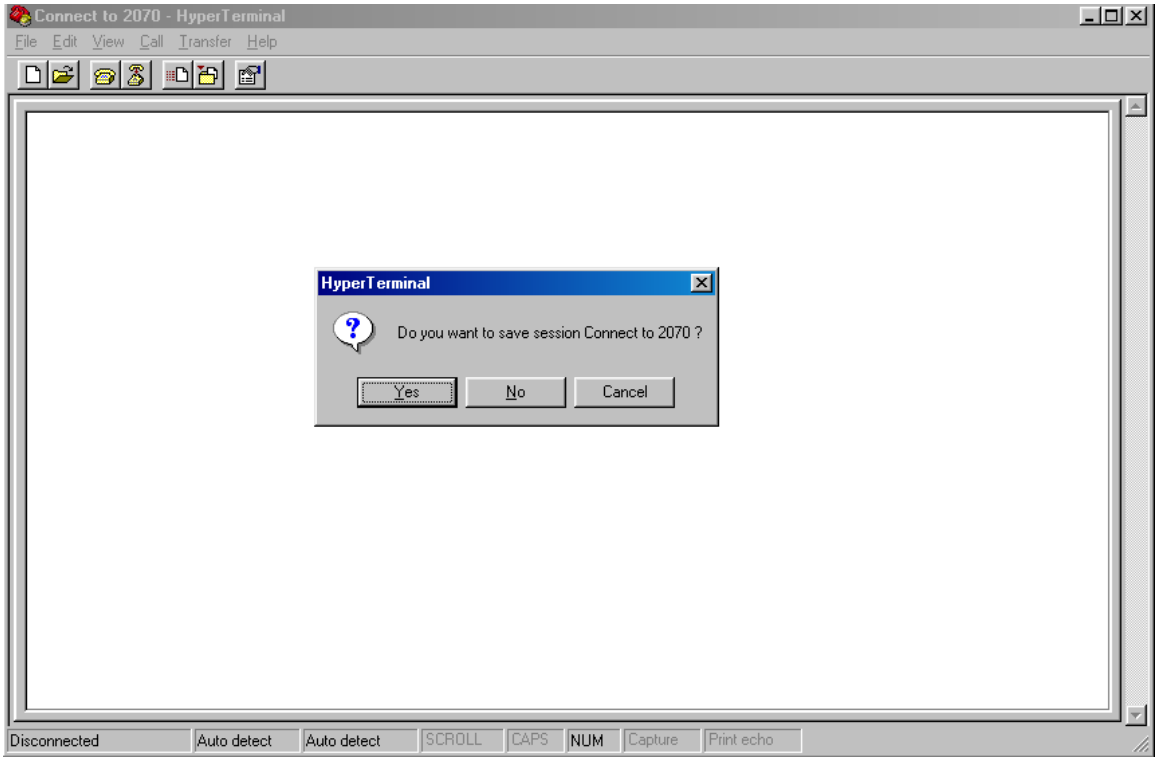
5. Select the baud, data bits, parity, stop bits and flow control as shown.



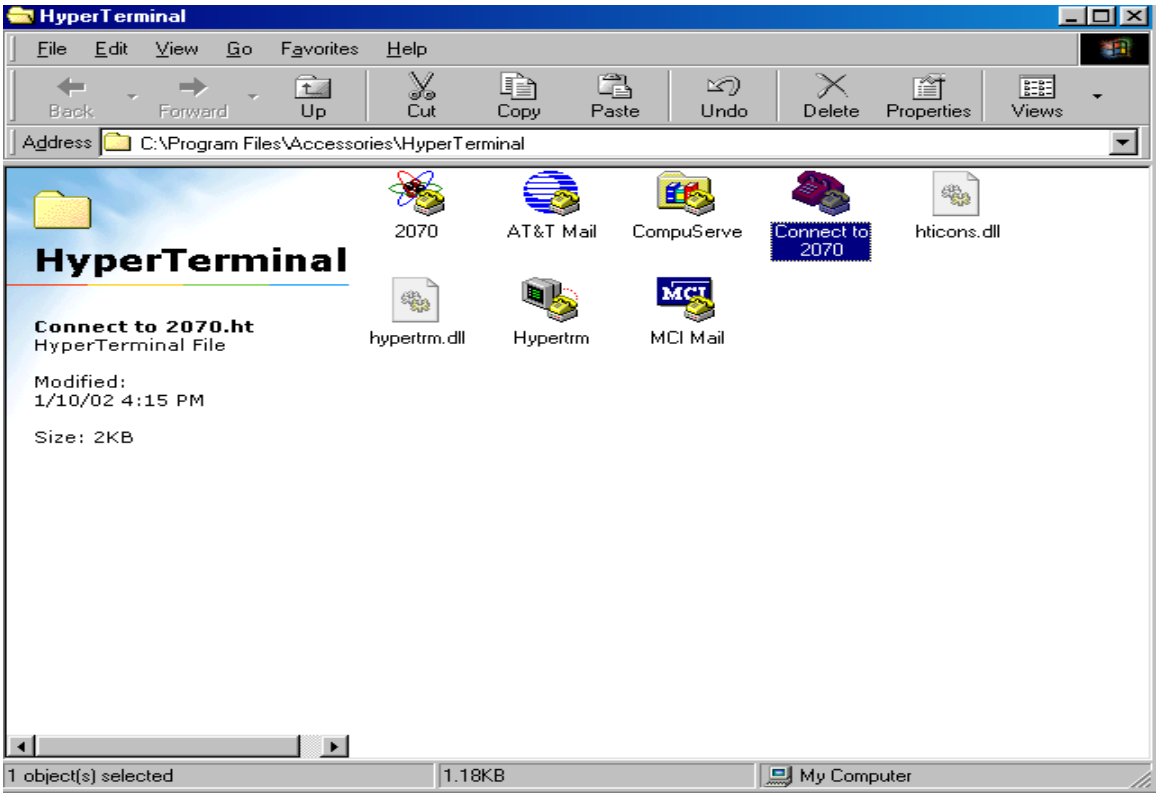
6. You should now be connected to the 2070. Note the lower left hand of the screen shows “connected”.



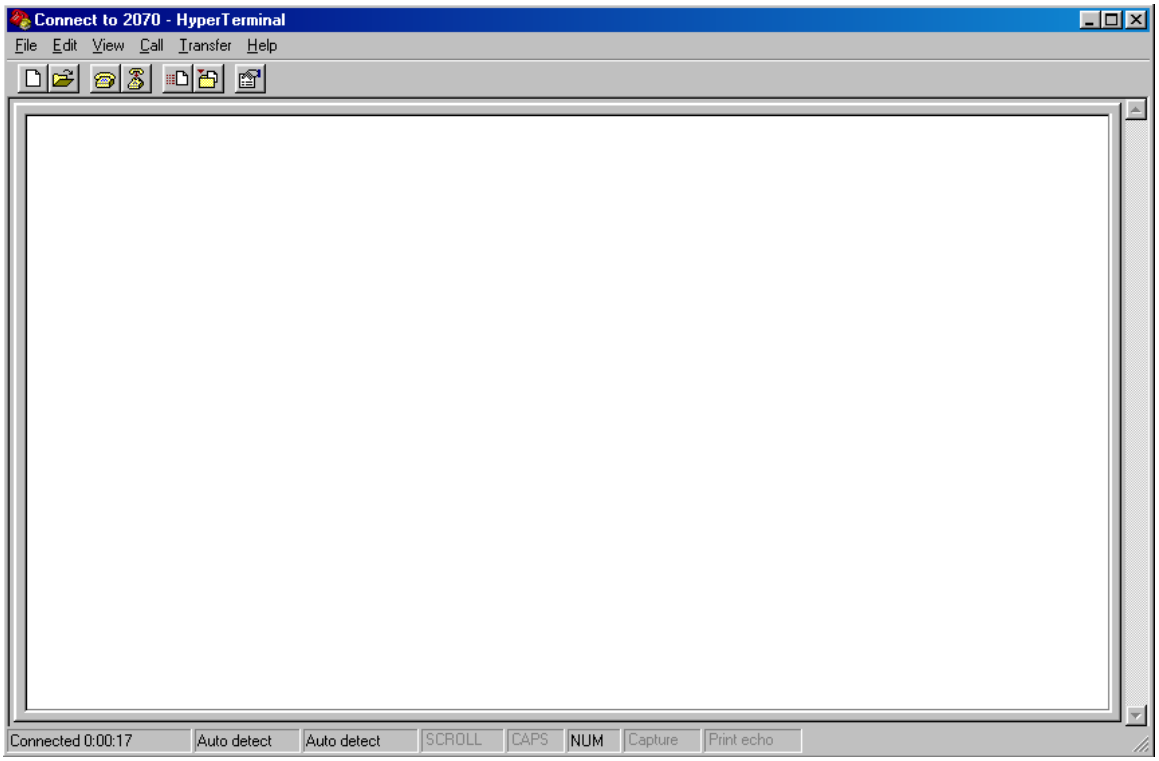
7. Upon exiting the program, select yes when you are asked if you want to save the session. This will allow you to preserve your settings for future sessions.



8. If you need to start another session of Hyperterminal you can select the icon with the name you have previously selected.



9. Upon restarting a saved session, you will be automatically connected with the 2070.



C. Summary of 2070 OS-9 Commands

attr
Syntax: attr [<opts>] {<path> [<opts>] <permissions>}
Function: display or change file attributes
Attributes: d s pe pw pr e w r
'-' turns attribute on '-n' turns attribute off
Options:

-a	do not print attributes after changes
-x	directory to search is execution directory
-z	get list of file names from standard input
-z=<path>	get list of file names from <path>

break
Syntax: break
Function: invoke the system level debugger (or reset system)
Options: none
(You must be the super user)

build
Syntax: build <path>
Function: build a short text file

copy
Syntax: copy [<opts>] <srcpath> [<dstpath>] [<opts>]
Function: copy data from one path to another
Options:

-a	abort on error
-b=<size>	buffer size
-f	rewrite destination files with no write permission
-p	don't print file names copied (with -w option only)
-r	rewrite destination
-v	verify integrity of files written
-w=<dir name>	wild card copy to <dir name>
-x	look in execution directory for source
-z	get list of file names from standard input
-z=<path>	get list of file names from <path

date
Syntax: date [<opts>]
Function: display system date and time
Options:
-j print day, seconds past midnight in julian time
-m print hour:minute:sec in military format

dcheck
Syntax: dcheck [<opts>] <devnam>
Function: check directory/file integrity
Options:
-d=<num> print path to dir <num> deep
-r rebuild allocation map from file structure
-y answer "y" to all questions in repair mode

deiniz
Syntax: deiniz [<opts>] {<devname> [<opts>]}
Function: detach devices
Options:
-z get list of device names from standard input
-z=<path> get list of device names from <path>

del
Syntax: del [<opts>] {<file> [<opts>]}
Function: delete files
Options:
-e erases the disk space that the file occupied
-f delete files with no write permission
-p show file name and ask before deleting
-x delete files from execution directory
-z get list of file names from standard input
-z=<path> get list of file names from <path>

deldir
Syntax: deldir [<opts>] {<dir> [<opts>]}
Function: delete a directory
Options:
-q delete directories without asking questions
-f delete files with no write permission
-z get list of directory names from standard input
-z=<path> get list of directory names from <path>

devs
Syntax: devs [<opts>]
Function: print system device table
Options:
(none available)

dir
Syntax: dir [<opts>] {<dir names> [<opts>]}
Function: display directory contents
Options:

-a	show all files
-d	show directories with a slash
-e	extended dir listing
-n	treat dirs like files
-r	recursive dir listings
-r=<num>	recursive dir listing to depth <num>
-s	unsorted dir listing
-u	unformatted listing
-x	directory is execution dir
-z	get list of dir names from standard input
-z=<path>	get list of dir names from <path>

dump
Syntax: dump [<opts>] <path> [<starting offset>] [<opts>]
Function: formatted display of contents of a device
Options:

-c	don't compress duplicate lines
-m	dump from a memory resident module
-s	interpret starting offset as sector number
-x	path implies execution directory

echo
Syntax: echo [<opts>] [<text>] [<opts>]
Function: echo text to output path and convert hex to ASCII
Options:

-n	separate text with carriage returns
-r	don't send out a return on exit
-z	get text from standard input
-z=<file>	get text from <file>

edt
Syntax: edt [<opts>] <path> [<opts>]
Function: line editor
Options:
-b=<size> use larger buffer (default is 2k larger than file size)
Commands:

<cr>	move line pointer down one line
<num>	move to line <num>
+[<num>]	move line pointer down <num> lines(default: 1)
-[<num>]	move line pointer up <num> lines(default: 1)
c	search/replace from current line
d[<num>]	delete lines(default: 1)
l[<num>]	list <num> lines(default: 1)
s	search from current line
<space>	insert line
<esc> or q	write file and return to shell

events
Syntax: events
Function: display active events
Options:
(none)

free
Syntax: free [<opts>] {<device> [<opts>]}
Function: report free space on disk
Options:

-b=<size>	buffer size

ident
Syntax: ident [<opts>] {<modname> [<opts>]}
Function: display module information
Options:

-m	ident module in memory
-q	quick mode, only one line per module
-s	silent mode: quick, but only displays bad crcs
-x	ident module in execution directory
-z	get list of module names from standard input
-z=<file>	get list of module names from <file>

iniz
Syntax: iniz [<opts>] {<devname> [<opts>]}
Function: attach devices
Options:

-z	get list of device names from standard input
-z=<path>	get list of device names from <path>

irqs
Syntax: irqs [<opts>]
Function: display entries on system IRQ polling table
Options:
(none available)

link
Syntax: link [<opts>] {<modname> [<opts>]}
Function: link a module in memory
Options:
-z get list of module names from standard input.
-z=<path> get list of module names from <path>

list
Syntax: list [<opts>] {<path> [<opts>]}
Function: list a file
Options:
-z get list of file names from standard input
-z=<path> get list of file names from <path>

kermit
OS-9 Kermit Version 1 Release 6 (pcb)
Usage: kermit c[8le line esc.char] (connect mode)
or: kermit s[dx8ifl line] file ... (send mode)
or: kermit r[dx8iflk line] (receive mode)
or: kermit h[dx8ifl line] (Host server mode)
or: kermit g[dx8iflk line] file ... (get file from server)
or: kermit q[dx8iful line] (Quit remote Host server)

lmm
Syntax: lmm [<opts>] <path>
Function: load a memory module from <path>
Options:
-q exit without error regardless of outcome

load
Syntax: load [<opts>] {<module> [<opts>]}
Function: load a module into memory
Options:
-d load file from data directory
-l print pathlist of file loaded
-z get list of module names from standard input
-z=<file> get list of module names from <file>

makdir
Syntax: makdir [<opts>] {<dir name> [<opts>]}
Function: create a directory
Options:

-x	create directory in execution directory
-z	get list of dir names from standard input
-z=<file>	get list of dir names from <file>

mdir
Syntax: mdir [<opts>] [<mod names>] [<opts>]
Function: display module directory
Options:

-a	print language instead of type
-e	print extended directory listing
-t=<type>	list modules only of type <type>
-u	print unformatted listing

mfree
Syntax: mfree [<opt>]
Function: display system memory information
Options:

-e	extended free memory description

pd
Syntax: pd [<opt>]
Function: print path to data or execution directory
Options:

-x	display execution directory path

procs
Syntax: procs [<opts>]
Function: display user processes
Options:

-a	display alternate data
-b	display both normal and alternate data
-e	display every valid process

rename
Syntax: Rename [<opts>] <path> <name> [<opts>]
Function: rename a file or directory
Options:

-x	path starts from execution dir

save
Syntax: save [<opts>] {<modname> [<opts>]}
Function: save given modules to files
Options:

-f=<path>	save list of modules to <path>
-r	rewrite files
-x	create output in execution directory
-z	get list of modules from standard input
-z=<path>	get list of modules from <path>

setime
Syntax: setime [<opt>] [yy mm dd hh mm ss] [am/pm]
Function: set system date and time
Options:

-d	don't display time
-s	setime for battery backed-up clocks

shell
Shell options:

-e[=<path>]	print error explanations.
-ne	don't print error explanations.
-p[=<prompt>]	print prompt.
-np	don't print prompt.
-t	echo input lines.
-nt	don't echo input lines.
-x	exit on error.
-nx	don't exit on error.
-l	require "logout" to logout.
-nl	<eof> on input will logout.
-v	print attempts to execute command.
-nv	don't print attempts to execute command.

sleep
Syntax: sleep [<opts>] [<count>] [<opts>]
Function: suspend process for ticks/seconds/until signalled
Options:

-s	count represents seconds

Note: Count defaults to 0 if not specified

tmode
Syntax: tmode [<opts>] {<parameter>} [<opts>]
Function: display and/or change terminal operating characteristics
Options:

-w=<#path>	specify path to perform operations on
[no]upc	- upper case only
[no]bsb	- erase on backspace
[no]bsl	- backspace over line
[no]echo	- echo input to terminal
[no]lf	- auto line feed
[no]pause	- screen pause
null=n	- number of null chars after CR
pag=n	- display page length
bsp=h	- input backspace char (in hex)
bse=h	- output backspace char (in hex)
del=h	- input delete char (in hex)
eor=h	- end of record char (in hex)
eof=h	- end of file char (in hex)
reprint=h	- reprint line character (in hex)
dup=h	- duplicate last line char (in hex)
psc=h	- pause char (in hex)
abort=h	- abort char (in hex)
quit=h	- quit char (in hex)
bell=h	- bell output char (in hex)
type=h	- ACIA initialization value (in hex)
par=s	- set parity (s=odd,even,none)
cs=n	- character size in bits (n=8,7,6,5)
stop=n	- number of stop bits (n=1,1.5,2)
baud=n	- software settable baud rate (in decimal)
xon=h	- DC1 resume output char (in hex)
xoff=h	- DC2 suspend output char (in hex)
tabc=h	- tab char (in hex)
tabs=d	- number of spaces per tab
normal	- return to default values

unlink
Syntax: unlink [<opts>] {<modname> [<opts>]}
Function: unlink modules from memory
Options:

-z	get list of module names from standard input
-z=<path>	get list of module names from <path>

xmode

Syntax: xmode [<opts>] <descriptor name> {<desc. name> [<opts>]}

Function: display/change options section of SCF/GFM/UCM device descriptor

Options:

-z	- get list of device names from standard input
-z=<path>	- get list of device names from <path>
[no]upc	- upper case only
[no]bsb	- erase on backspace
[no]bsl	- backspace over line
[no]echo	- echo input to terminal
[no]lf	- auto line feed
[no]pause	- screen pause
null=n	- number of null chars after CR
pag=n	- display page length
bsp=h	- input backspace char (in hex)
bse=h	- output backspace char (in hex)
del=h	- input delete char (in hex)
eor=h	- end of record char (in hex)
eof=h	- end of file char (in hex)
reprint=h	- reprint line character (in hex)
dup=h	- duplicate last line char (in hex)
psc=h	- pause char (in hex)
abort=h	- abort char (in hex)
quit=h	- quit char (in hex)
bell=h	- bell output char (in hex)
type=h	- ACIA initialization value (in hex)
par=s	- set parity (s=odd,even,none)
cs=n	- character size in bits (n=8,7,6,5)
stop=n	- number of stop bits (n=1,1.5,2)
baud=n	- software settable baud rate (in decimal)
xon=h	- DC1 resume output char (in hex)
xoff=h	- DC2 suspend output char (in hex)
tabc=h	- tab char (in hex)
tabs=d	- number of spaces per tab
normal	- return to default values