4YDM

User Reference Manual

This manual covers the following CPU firmware versions:

CA
CB
CD
CE
DO

Document Version

A4

Notices

No patent liability is assumed with respect to the use of information contained herein.

While every precaution has been taken in the preparation of this publication, AutoPatch assumes no responsibility for error or omissions. No liability is assumed for damages resulting from the use of the information contained herein.

Further, this publication and features described herein are subject to change without notice.

The United States Federal Communications Commission (in 47CFR 15.838) has specified that the following notice be brought to the attention of the users of this product.

Federal Communication Commission Radio Frequency Interference Statement:

> "This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type-tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However there is no guarantee that interference will not occur in a particular installation. If this equipment causes interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
>
> » Re-orient the receiving antenna
>
> » Relocate the matrix with respect to the receiver
>
> » Move the matrix away from the receiver
>
> » Plug the matrix into a different outlet so that computer and receiver are on different branch circuits

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the booklet, *How to Identify and Resolve Radio-TV Interference Problems*, prepared by the Federal Communications Commission to be helpful."

This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock N. 004-000-00345-4.

Use shielded cables. To comply with FCC Class B requirement, all external data interface cables and adapters must be shielded.

AutoPatch Statement of Warranty

AutoPatch warrants that the products manufactured by it will, under normal use and service, be free of defects in materials and workmanship for a period of two (2) years from the date of shipment shown on the sales invoice or warranty registration on file. With respect to any product found to be defective during this warranty period, AutoPatch's liability and Buyer's remedies under this warranty shall be limited solely to repair, replacement, or credit, at AutoPatch's option. This warranty will not apply to any AutoPatch product that has been modified, repaired after leaving the factory, or improperly installed, used, or maintained.

This warranty is exclusive and in lieu of any other warranty, expressed or implied, including but not limited to any implied merchantability or fitness for a particular purpose.

AutoPatch shall not be liable under any circumstances for consequential or incidental damages including, but not limited to, labor costs or loss of profits arising in connection with the use of or inability to use AutoPatch products.

AutoPatch will not be responsible for items damaged during shipment to or from AutoPatch.

These warranty terms may not be modified except in writing by an authorized officer of AutoPatch.

The terms of this warranty are governed by the laws of the state of Washington; certain other states restrict warranty limitations. You may have rights that are not defined herein.

Warranty Extensions

Extended warranties providing the same benefits as the original two year warranty are offered on the following terms:

1. Warranty extensions will be allowed only on products for which we receive a warranty registration within 90 days of shipment.
2. All extension transactions must be completed before the first anniversary of the warranty registration date.
3. Extensions can be purchased in one year increments to a maximum of eight (8) years.
4. The price PER YEAR of the warranty extension will be calculated as one percent (1%) of the original invoice net price for the product.

See Appendix A for claims procedures

Table of Contents

Overview

This manual is designed to help you learn to fully operate the 4YDM. If you have little or no experience with distributed switching systems, it is advised you read the *Getting Started* section of this manual before advancing. *Getting Started* is designed to help you quickly begin using the 4YDM; it explains how to install the distribution matrix, switch an input to output devices, and query the status of the I/O configuration. The rest of this manual is more detailed to help you further understand how to operate the 4YDM.

The 4YDM is designed to fit in a broad range of audio/video environments and can be controlled from several different sources.

How to Read This Manual

Key References

When referring to the keys on the control panel, the name of the key will either be enclosed by < and > or referred to as the "name of key" key. For example, the key on the control panel entitled Special will be referred to as <Special>, or the Special key. Both cases refer to the same key, but because including text in < and > interferes with software-generated lists used to create this manual, the name of the key cannot always be included in < and >. All references to numeric keys on the control panel are specified by including the number of the key in < and >.

Breakdown of the Chapters

This manual is composed of two introductory sections, seven chapters, and five appendices and attempts to present information in both a task-oriented and feature-oriented manner. The Table of Contents lists the chapters by feature; Appendix E is a list of step-by-step tasks in this manual. Use these two sources to help you understand any aspect of the distribution matrix or quickly find any of the listed tasks.

Technical Support

AutoPatch provides technical support from 6 a.m. to 5 p.m. PST Monday - Friday. Before calling with a question, please consult this manual first. If the manual cannot, or does not fully answer your question, call (800) 622-0246.

Starting Point for Specific Users

If you are using a control panel to operate the matrix, begin reading from *Getting Started*. If you are using a PC or another type of external controlling device, begin reading at Chapter 5. You may be required to reference the first four chapters, but Chapter 5 instructs how to make serial connections from a PC to the matrix and describes various methods of controlling the matrix.

The following is a summary of the chapters and appendices.

Getting Started

This section is a quick introduction to your 4YDM and common technical terms used in this manual.

Chapter 1 - The Enclosure and Its Parts

This chapter describes the pieces of the enclosure, how to install and link the enclosures, how to remove and install input and output boards, and how to differentiate between the master and slave enclosures if the matrix has more than one enclosure.

Chapter 2 - Configuring the Inputs and Outputs

There are three keys on the control panel that can be used to change the I/O configuration. This chapter explains how to use each of these keys and their effect on the I/O configuration.

Chapter 3 - Status/Problem Solving

Because all output devices may not be in listening or viewing range of the matrix, you can use <Status> to track which input an output is receiving. Chapter 3 explains how to specify and track a signal. If an error should occur because of incorrect input or a faulty action, an error screen appears. Chapter 3 also explains the cause behind the error screens and the correct action to take.

Chapter 4 - Customizing the Distribution Matrix

So that the matrix can fit a variety of audio/video environments, the matrix has several adjustable features which make operation as user-specific as possible. These features include the function of <Special>, setting the BAUD rate, setting the SBC polling range, and the default command when a numbered key is pressed.

Chapter 5 - External Control

External control is the ability to control the matrix from a device other than the control panel; this control may be provided by a PC, dry contacts, or any 3rd party controlling device linked to the matrix via the serial port. Chapter Five explains how to make connections between the matrix and an external controller and how to make the external controller communicate with the matrix.

Chapter 6 - Configuration Files

Configuration files contain the input and output structure (I/O configuration) of the matrix, and specific commands. This chapter explains the five sections of a configuration file and how to upload it to the CPU.

Chapter 7 - Advanced Features

The advanced features of the matrix include literal operations, programming configuration words, resetting enclosures and refreshing the logical status table. Literal operations are one of the most powerful features of the matrix. Be sure that you understand chapters 1-5 before attempting the tasks described in this chapter.

Appendix A: Warranty and Terms

AutoPatch warranty and the claims procedures.

Appendix B: Vertical Interval Sync Expansion Board

AutoPatch offers a vertical interval sync expansion board. Appendix B details the specifications of the board and lists some of its uses. If your matrix does not have a sync board(s) and you would like to learn more about the sync board or purchase one, call *AutoPatch* at (800) 622-0246.

Appendix C: Single Bus Controller

Single bus controllers (SBC) are individual output device controllers that allow you to select the input going to an output device. SBC's are optional extras for use with the 4YDM. Appendix C explains how to install and use SBC's.

Appendix D: 4YDM Specifications

Appendix D lists the power, audio, and video specifications of the 4YDM.

Appendix E: APS Command Sample Program

A sample C program that was written to test various APS commands. Use this as an example when creating custom control software that uses APS commands.

Glossary

The glossary defines terms as they are used in this manual.

Index of Tasks

This Index lists the tasks in this manual and the page on which they begin.

Index

The index is listed by feature. The tasks that those features may be used to accomplish are listed below each feature.

Getting Started

The *AutoPatch* 4Y Distribution Matrix (4YDM), or matrix, is a signal switching device that can route audio, video, and data signals. The purpose of this section is to allow you to immediately begin using the matrix. This section covers how to assemble the 4YDM, switch signals, and query the status of the matrix.

Note: AutoPatch will not be responsible for damage to returned products incurred during shipping due to insufficient packaging. If requested, AutoPatch will supply a new shipping carton at cost.

Before assembling the matrix and beginning to make switches, read the following definitions. These terms are used throughout this manual.

Signal - A signal can be either audio, video, or data information. A signal traveling through a signal path from a source such as a video camera, microphone, or satellite dish and entering an input board in an enclosure (see Enclosure), is an input signal. When a signal travels from an output board in the enclosure through a signal path to a destination device such as a video monitor, VCR, or computer, that signal is an output signal.

Distribution Matrix (DM)- The distribution matrix, or matrix, is the environment of all signals and the hardware and software necessary to switch these signals. The terms, routing switcher, switching matrix, and routing matrix, are also used commonly in the audio/video industry to describe a distribution matrix. The distribution matrix facilitates the routing of an input signal to one, all, or any variation of outputs.

Enclosure - An enclosure is a metal chassis that houses input and output boards. An enclosure can hold up to eight input and eight output boards. Each input and output board can handle four signals. A matrix can have multiple enclosures which can be linked together allowing more input signals to be switched to more output signals.

Control Panel - The panel on the front of the enclosure that has keys for entering commands, and a screen for viewing messages about the matrix. The enclosure pictured in figure 1 has a control panel.

External Controller - An external controller is any device other than the control panel, that is used to change the input/output configuration. External controllers can be dry contacts, single bus controllers (SBC) or any device that can send and receive ASCII code via the serial port. Examples: *AutoPatch* single bus controller, third party control systems, or a PC that uses a serial communications program to communicate with the distribution matrix.

Logical Inputs and Outputs - Logical inputs and outputs are definitions of the physical input and output signals that are used when making a switch. Each definition can specify from one to sixteen signals. Logical inputs and outputs are especially useful for switching component signals (see Component Signal) or groups of signals that are normally switched at the same time.

Input/Output (I/O) Configuration - The input/output configuration is the status of all the input and output signals in the matrix (or enclosure, if so specified) at a given time. For example the status of an input signal would be the output signal to which it was routed and the status of an output signal would be the input signal it was receiving.

Component Signals - Component signals are groups of signals that are normally switched together because each signal provides a necessary component for intelligible output. Some examples of component signals are: RGBS video, RGsB video, Y-C video, and stereo audio. Because component signals are comprised of multiple signals, *AutoPatch* uses a grouping (see Grouping Table) to switch sets of component signals.

Grouping Table - A grouping table is a text file that contains the signal definitions for logical inputs and outputs. A grouping table is used to upload the logical signal definitions to the 4YDM's CPU.

Module - A module is a segment of code that controls a function of the matrix. The software modules provide the features.

Unpacking and Installing the Matrix

When opening the shipping boxes, check the condition of the boxes and immediately report any damage to the shipping agency. The shipping boxes contain enclosures or parts, a packing slip, documentation, a floppy disk, and other matrix products. Included with the instructions is a sheet labeled, ***AutoPatch* 4YDM Connector and Groupings Guide,** which shows the input and output connector (signal path) configuration of each enclosure.

The floppy disk contains executable programs and information files necessary for normal operation of the 4YDM. The configuration file contains a backup of the grouping table that was uploaded to the CPU. The DOS executable program is used to upload a configuration file to the CPU. *AutoPatch* advises you to make a copy of the disk using any standard DOS copy feature. In the unlikely event of a loss of memory, or to alter the information in the configuration file, you will need to upload the configuration file (grouping tables) again.

Enclosures are designed to be installed in a standard EIA 19" rack. All necessary cables for linking and powering the matrix are included with the enclosures. The matrix produces very little heat and is designed to adequately dissipate this heat under normal operating conditions. This ability is defeated by placing high heat-producing equipment directly below or above the matrix in a rack or any other mounting system.

Note: Please save the shipping box and materials in the event the products need to be shipped back to AutoPatch. AutoPatch will not be responsible for damages sustained in shipping.

Installing the Matrix:

1. Remove the four shipping screws.

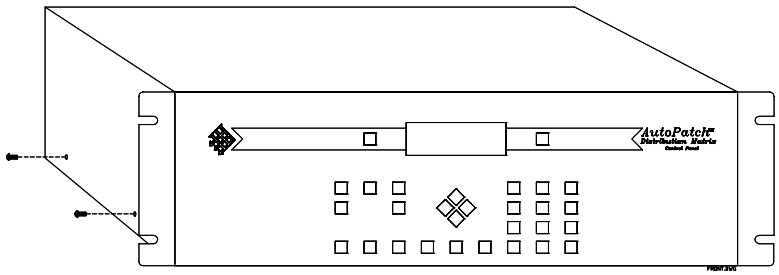


figure 1 Remove the shipping screws

Note: Removing the shipping screws during installation allows easier access to the inside of the matrix during maintenance.

2. Place each enclosure in the rack and attach front mounting screws to hold the enclosure in place. For ease of operation, install an enclosure with a control panel at

eye level in the rack. Viewing the LCD screen at extreme angles makes it hard to see the screen messages; the optimum viewing angle for the LCD screen is +15º.

3. If the matrix has more than one enclosure, all enclosures must be linked so control information can travel to and from each enclosure. Using the cables provided, link the enclosures as shown in figure 2. These cables allow up to six inches of spacing between each enclosure. Additional custom-built cables may be ordered; these cables allow enclosures to be placed up to 300 feet apart . If a CPU board contains a fourth jack, do not use the lowest jack for linking.

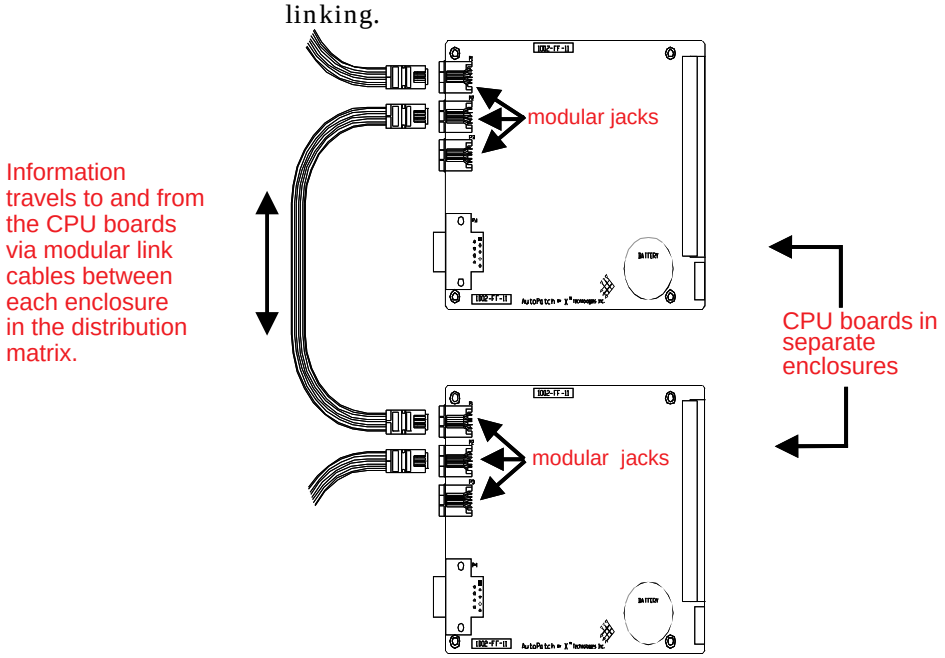


figure 2 Inter-enclosure cabling

4. After ensuring the voltage selector is in the correct position for the electrical supply, plug each enclosure into a power source. The voltage selector is identified in figure 3. Voltage requirements for each enclosure are 115v or 220v +/-10% . It may be helpful to plug each enclosure into a power strip and then plug the power strip into an outlet. **Power surge protection is recommended.**

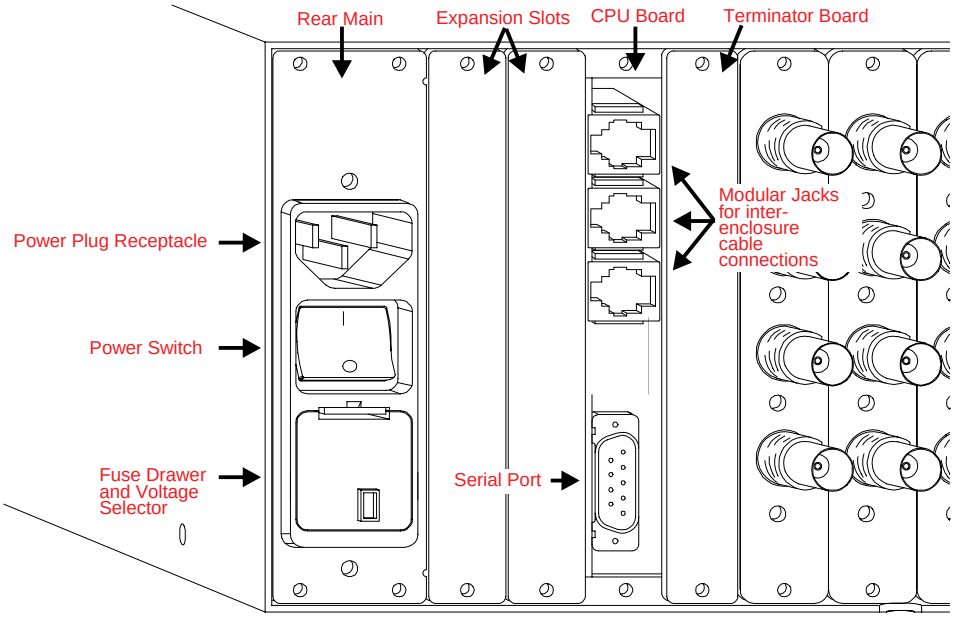


figure 3 Rear main and expansion slots

5. The screen on each control panel flashes start up information, displays the number of enclosures that are recognized, and indicate whether or not there are errors.

```
Acknowledged 2
Enclosure/s Online
   >Warning<
3 Errors Reported
```

figure 4a Two enclosures acknowledged and three errors

```
  Acknowledged 2
Enclosure/s Online

No Errors Reported
```

figure 4b Two enclosures acknowledged and no errors

6. Ensure that the number of enclosures acknowledged is the number of enclosures in the matrix. If the number is correct, installation was a success. If the number is not correct, one or more enclosures have not been recognized. Check the power source connection and the linking cables between the enclosures. If after checking the power source connections and linking cables the number of enclosures is still not correct, call *AutoPatch* Customer Service (800) 622-0246.
7. If installation was successful, the matrix is ready to make a switch. Press <Cancel>; the Command screen appears. The Command screen is the beginning point for any command. When executing a command, begin from the Command screen (see figure 5).

```
  AutoPatch 4YDM

Command:
```

figure 5 Command screen

Attaching the Input and Output Signals

Before making a switch, connect one input and one output so you can verify the result of a switch. A sheet labeled, "***AutoPatch* 4YDM Connector and Groupings Guide"** is included in the shipping boxes. This sheet defines the position of each input and output signal connection on the rear of each enclosure. When making the signal connections, follow the layout exactly; the matrix has been pre-programmed to work with the signal connections as indicated.

Before installing all the input and output signals, connect the first two inputs and outputs and switch those inputs to verify the matrix is in proper working order. The following section, "Making a Switch," leads you through the switch. The number of signals per input and output may vary. Be sure the input and output signal wires are properly attached to the respective input and output devices. After switching the first two input and output signals and verifying that they switched correctly, install the rest of the input and output signals.

Making a Switch

The CPU keeps track of the input signal each output signal is receiving, i.e., the status of the I/O configuration. The CPU is ready for you to make a switch and stores the effect of the switch even though the signal wires for the input and output signals may not be connected.

The following switch routes input 2 to output 1. Before making a switch, be sure the first two inputs and outputs are attached as shown on the *AutoPatch* 4YDM Connector and Groupings Guide.

1. On the control panel, press <Change>. The following screen appears:

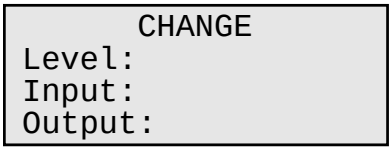


figure 6 Change screen

2. Press <Input>. '0' appears after the Level prompt which indicates the change will be made on all levels (i.e., the same change occurs in each enclosure) and the cursor appears after the Input prompt. Press <2>.

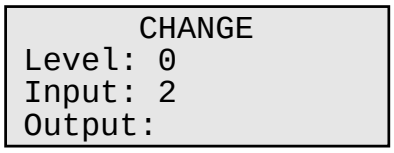


figure 7 Routing input 2

3. Press <Output>; the cursor appears after the Output prompt. Press <1>.

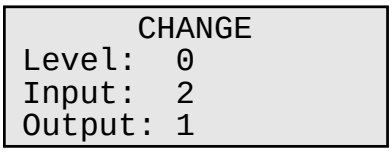


figure 8 Routing input 2 to output 1

4. Press <Take>; the Command screen appears. The change has been entered.

You have just switched logical input 2 to logical output 1. You should be able to see or hear logical input 2 on the output device for logical output 1. If the switch did not work, check all link and signal connections on the rear of the enclosures to make sure the matrix is in proper order and then attempt the switch again. If the switch still does not work, call *AutoPatch* at 1-800-622-0246.

Checking the Status of the Matrix

If you are not close enough to an output device to see or hear if a switch was made, check the status of the logical input or logical output in question. When checking the status of the matrix, specify a logical input or logical output; the CPU responds with the respective logical output or logical input. Since you switched logical input 2 to logical output 1, check the status of logical output 1.

1. Press <Status>; the Status screen appears.

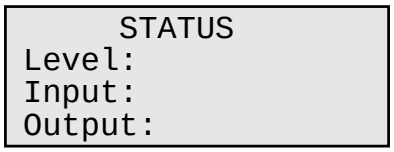


figure 9 Status screen

2. Press <Output>; the cursor appears on the Output prompt. Press <1>. '1' appears next to the Level and Output prompts.

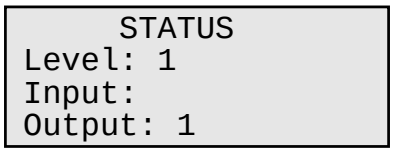


figure 10 Enter an output

3. Press <Take>. The screen shows the input signal that is routed to output 1. If you made the switch that was described in the previous section, input 2 should be routed to output 1.

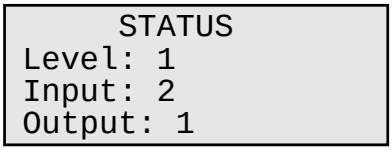


figure 11 Input for the specified output appears

4. Press <Cancel> to return to the Command screen.

Attach the wiring for the remainder of the inputs and outputs in the matrix. Remember to follow the ***AutoPatch* 4YDM Connector and Groupings Guide** when attaching the signal wires**.**

Master and Slave Enclosures

The master enclosure is the most important enclosure in the matrix because it keeps track of the slave enclosures and verifies they are all online. The master enclosure sends a message to each control panel and external controller screen that defines the number of enclosures recognized as online. To identify whether an enclosure is the master or slave, look at the Pld number. The number of the enclosure is also the level number.

To Identify the Master and Slave Enclosures:

1. At the Command screen, press <Program>; the Program menu appears.

figure 12 Program menu

2. Press <Special>; the Version screen appears on all enclosures. The information contained on the version screen differs depending on the matrix.

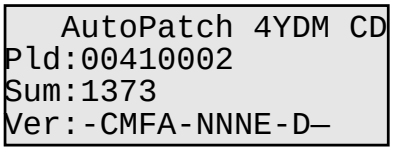


figure 13 Verification screen

3. Look at the number after the Pld prompt. The fourth digit from the left is the number of that enclosure. If the third digit from the left is a '4', that enclosure is the master. If the third digit from the left is not a '4,' that enclosure is a slave. The number after Sum is the checksum of the ROM; this number is important only if you want to change the modules in the ROM. Master and slave enclosures display the same checksum. The dashes and letters after `Ver:` are the modules that are present in the matrix (for more information on modules, see section 7.4, "Modules and Configuration Words").

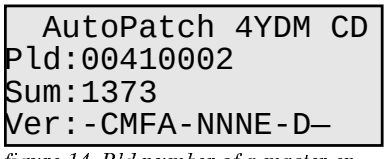


figure 14 Pld number of a master enclosure

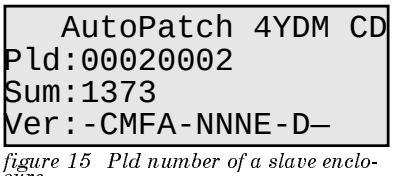


figure 15 Pld number of a slave enclosure

Chapter 1 - The Enclosure and Its Parts

A matrix can be comprised of 1 to 30 enclosures. An enclosure is a chassis which can hold up to eight input and eight output boards. From the rear of the enclosure, the output boards are located to the left, and the input boards are located to the right. Input and output boards handle either audio or video signals and each board can handle four signals. Audio boards handle only audio signals and video boards handle only video signals. To the left of the output boards is the terminator board slot and three expansion slots. The enclosure's CPU board fits in any of the expansion slots.

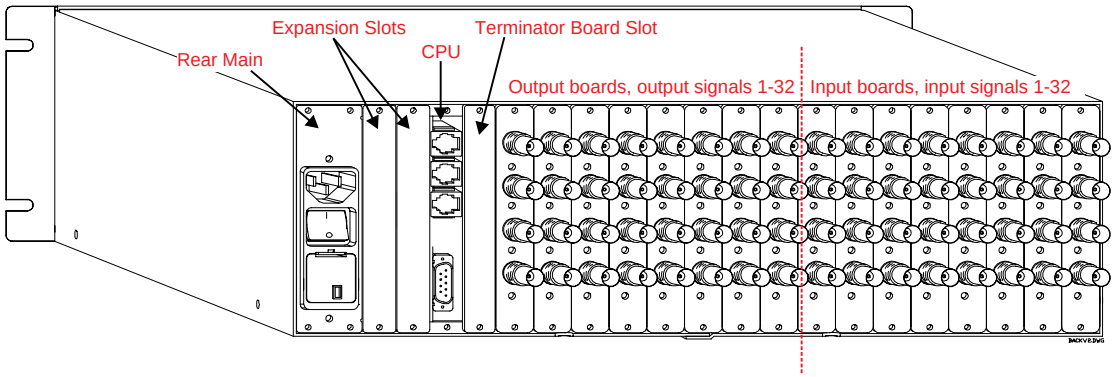


figure 1.1 Rear view of an enclosure

Inputs and outputs are numbered separately beginning with the top channel of the right-most board and down the board as seen in figure 1.1. The number of enclosures used in a matrix is the fewest number necessary to handle all input and output signals. For example, to route 32 composite video signals with stereo audio to 8 video monitors, three enclosures would be required. Enclosure #1, with 8 video input and 2 video output boards, would handle all video signals. Enclosure #2, with 8 audio input and 2 audio output boards, would handle the left audio signals. Enclosure #3, with 8 audio input and 2 audio output boards, would handle the right audio signals.

All enclosures in a matrix must be linked together via a link cable so that control information can be distributed through all enclosures.

All enclosures are basically the same. The characteristics that make enclosures different are: the programming of the CPU board, the type (audio or video) and number of boards an enclosure contains, and whether or not an enclosure has a control panel. See section 1.6, "Master and Slave Enclosures," to determine which enclosure in the matrix is the master. If the matrix contains only one enclosure, it is the master.

1.1 Installation Information for the Matrix

The shipping boxes for each order contain enclosures (or parts), a packing slip, documentation, a floppy disk, and other distribution matrix products. Also included in the boxes is a sheet labeled, ***AutoPatch* Connector and Groupings Guide**, which shows the input and output connector configuration of each enclosure (please note that this sheet substitutes the term 'cage' for the term 'enclosure'). Store this sheet in a safe place, you will need it when connecting the input and output signal wires to the rear of the enclosure.

The floppy disk contains executable programs and information files necessary for normal operation of the 4YDM. The information files have already been uploaded to the CPU, but *AutoPatch* advises you to keep the disk as a backup. Always retain a copy of the configuration file so that it can be uploaded in the unlikely event of a loss of memory, or used as a starting point if you wish to alter the information in the configuration files.

Enclosures are designed to be installed in a standard EIA 19" rack. All necessary cables for linking and powering the matrix are included with the enclosures. Placing high heat-producing equipment directly above or below the matrix is not recommended. The 4YDM produces very little heat and is designed to adequately dissipate this heat under normal operating conditions. In cases of a populated high-res 4YDM, it may be necessary to supply fans for cooling of multiple enclosures.

Note: Please save shipping box and materials in event product needs to be shipped back to AutoPatch. AutoPatch will not be responsible for damaged incurred through shipping.

Installing the Matrix:

1. Remove the four shipping screws.

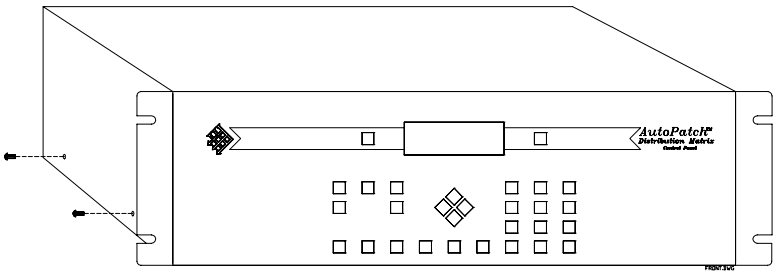


figure 1.2 Remove the shipping screws

N ote: Removing the shipping screws during installation allows easier access to the matrix during maintenance.

2. Place each enclosure in the rack and attach front-mounting screws to hold the enclosure firmly in place. For ease of use, mount an enclosure with a control panel in the rack at eye level. Viewing the LCD screen at extreme angles makes it hard to see the screen messages; the optimum viewing angle for the LCD screen is +15º.

3. If the matrix has more than one enclosure, all enclosures must be linked so control information can travel to and from each enclosure. Link the enclosures as shown in figure 1.3 using the cables provided. These cables allow up to six inches of spacing between each enclosure. (If a CPU board contains a fourth jack, do not use the lowest jack for linking.) Custom-built cables may be ordered to allow enclosures to be placed up to 300 feet apart.

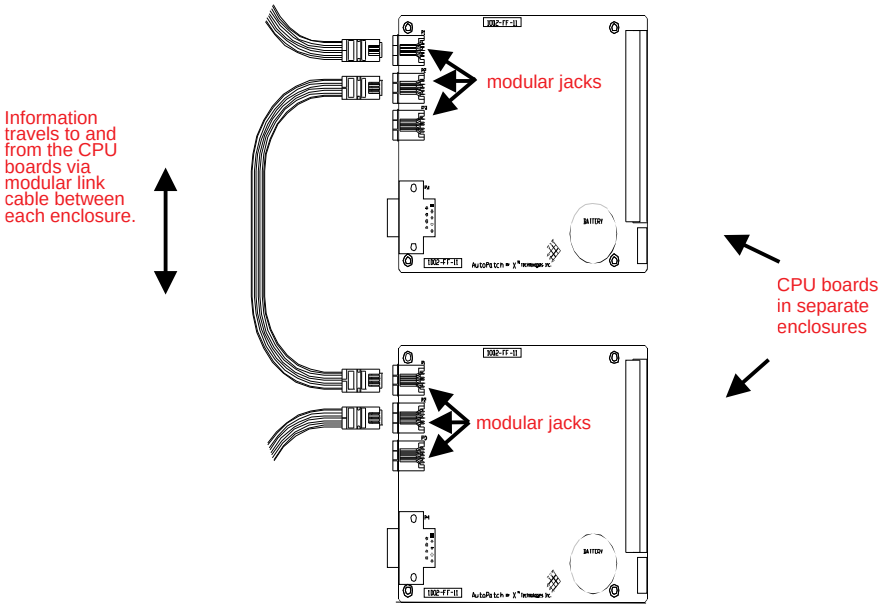


figure 1.4 Linking CPU boards

4. After ensuring the voltage selector is in the correct position, plug each enclosure into a power source.

   The voltage selector is identified in figure 1.12. Voltage requirements for each enclosure are 115v or 220v +/-10%. It may be helpful to plug each enclosure into a power strip and then plug the power strip into an outlet. **Power surge protection is recommended.**

   The screen on each control panel will flash start-up information, show the number of enclosures that are recognized, and indicate whether or not there are any errors.

```
Acknowledged 2
Enclosure/s Online
   >Warning<
3 Errors Reported
```

figure 1.3a Two enclosures acknowledged and three errors

```
Acknowledged 2
Enclosure/s Online

No Errors Reported
```

figure 1.4b Two enclosures acknowledged

5. Ensure the number of enclosures acknowledged is the number of enclosures in the matrix. If the number is correct, installation was a success. If the number is not correct, one or more enclosures have not been recognized. Check the power source connection and the linking cables between the enclosures. The control panel screen will report the new number of enclosures recognized when loose linking cables are secured.

   If after checking the power source connections and linking cables, the number of enclosures is still not correct, call *AutoPatch* Customer Service.

6. If installation was successful, the matrix is ready to make a switch. Press <Cancel>; the Command screen (as pictured below) appears. The Command screen is the beginning point for any command.

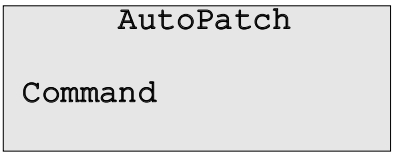


figure 1.5 Command screen

1.2 Input and Output Signals

The sheet labeled, "***AutoPatch* 4YDM Connector and Groupings Guide,**" that was included in the shipping boxes shows where to attach the signal wires on the rear of each enclosure. Follow the sheet exactly; the matrix has been designed and pre-programmed to work only as indicated.

Attaching the Input and Output Signals:

If you followed the *Getting Started* section in the front of this manual, you should have installed the first two logical inputs and outputs listed on the I/O connector position sheet. Remember, logical inputs and outputs can contain more than one physical signal. Using this sheet, you should be able to see the signals that comprise the logical inputs and outputs. Install all inputs and outputs listed on the sheet.

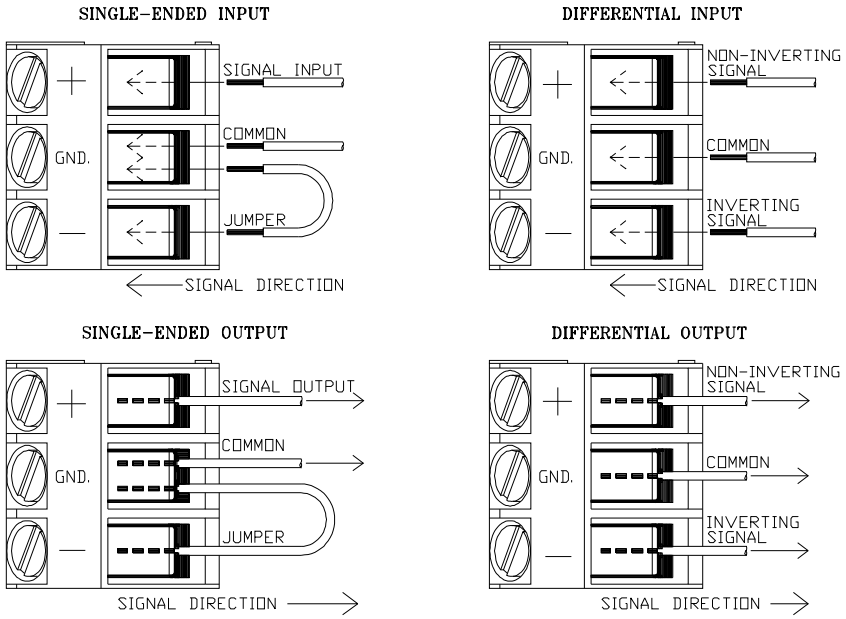


figure 1.6 Single-ended and differential connections

To attach video inputs and outputs, insert the video cable onto the proper input or output. To attach audio inputs and outputs: unscrew the clamps on the audio connector, insert the proper wire, and screw the wire back in so the clamp is tight and the audio wire makes a proper connection. Figure 1.6 shows examples of single-ended and differential audio inputs and outputs.

Note: Connect the shield only at the receive end to alleviate low frequency noise problems.

1.3 Input and Output Boards

The number of boards in an enclosure depends on the number of input and output signals. In an enclosure, signals are either sent or received through a channel. Each board has four channel receptors which can be seen on the connector assembly (as seen in figure 1.7).

Removing an Input or Output Board:

1. Unplug the power source to the enclosure.
2. If you can easily reach the rear of the enclosure, do not remove the enclosure from the rack. If you cannot easily reach the rear of the enclosure, unplug any inter-enclosure cabling and remove the enclosure from the rack.
3. Remove the screws holding the connector assembly.
4. Disconnect the connector assembly from the board by pulling the connector assembly out from the board until the pins on the back clear. The connector assembly may be difficult to remove; pull firmly.

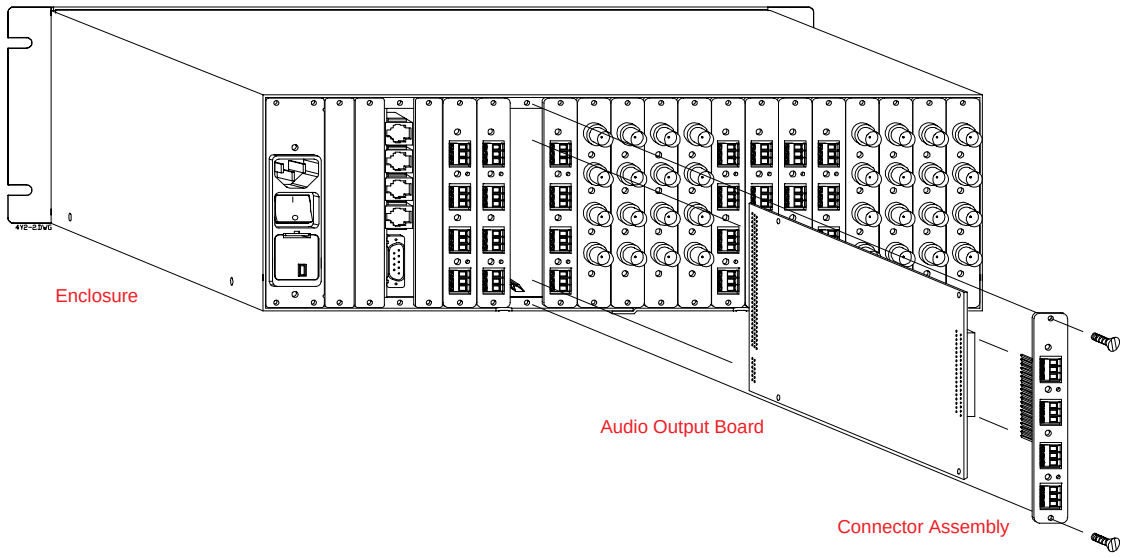


figure 1.7 Connector Assembly, Board and Enclosure

Warning: Before touching any board, touch a grounded enclosure to limit possible EMI or shock to the circuit board. Do not pull the connector assembly at an angle; you could damage the pins.

5. Grasp the bottom of the board with a pair of pliers or board remover and pull. If the board is seated snugly, it may help to wiggle the board up and down as you pull.

Inserting an Input or Output Board:

1. Ensure the board is being inserted in the correct slot. An audio board fits into a video slot, and vice versa, but the matrix will not work properly.
2. Line the board up with the board guides on the top and bottom of the board slot. Carefully push the board all the way into place. It may be difficult to snap the board into place; push firmly, but exercise caution (you can damage the back plane or board). If the board resists going in, the pins may not be lined up with the receptor on the rear of the slot. Realign the pins and try again.
3. Once the board has been snapped into place, plug the connector assembly into the board (see figure 1.7). If the pins are offset, the screw holes on the connector assembly and the enclosure do not line up.
4. Insert the screws into the holes on the connector assembly and tighten the screws until snug.
5. Apply power to the enclosure.

1.4 Expansion Slots

Each enclosure provides additional slots for the CPU and other expansion boards. The only expansion board currently available is the 4YDM video sync board which provides vertical interval switching for systems with gen-locked video. For more information on the video sync board, see Appendix B or call *AutoPatch* (800) 622-0246. When adding an expansion

board to an enclosure, remove the cover plate on the slot to be filled and follow the instructions for inserting a board (see section 1.3, “Input and Output Boards").

1.5 CPU Board

The CPU provides a DB-9 serial port for external controller access, and contains modular jack connectors for linking enclosures. The CPU board may be placed in any of the three expansion slots (see figure 1.1). If the CPU board has four modular jacks, do not use the lowest jack for linking enclosures. Removal and insertion of the CPU board is identical to other boards, see section 1.3, “Input and Output Boards.”

Warning: The CPU board is always powered. Component damage may occur if the board is set on a conductive surface.

When plugging a serial cable into the DB-9 connector, make sure the pins are lined up with the receptor on the connector assembly and push firmly until snug.

A battery backup is provided on each CPU board to ensure the items stored in the CPU’s memory are not lost when power is removed from the matrix either intentionally or due to a power failure. The RAM does not draw power from the battery as long as power is present in the matrix. The active life of this battery is rated at 10 years.

1.6 Master and Slave Enclosures

The master enclosure keeps track of the slave enclosures and verifies they are all online. The master enclosure sends the message to the control panel or external controller screen that identifies number of enclosures online. To identify whether an enclosure is the master or slave, find the number of the enclosure. The number of the enclosure is also the level number.

Identifying the Master and Slave Enclosures:

1. At the Command screen, press <Program>; the Program menu appears.

figure 1.8 Program menu

2. Press <Special>; the Version screen appears. The information contained on the version screen may differ depending on your matrix.

```
 AutoPatch 4YDM CD
Pld:00410002
Sum:1373
Ver:-CMFA-NNNE-D—-
```

figure 1.11 Version screen

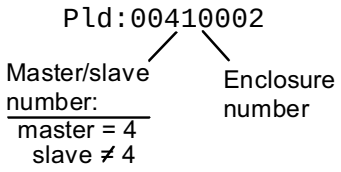
Look at the number after the `Pld:` prompt. The fourth digit from the left is the number of that enclosure. If the third digit from the left is a '4', that enclosure is the master. If the third digit from the left is not a '4' that enclosure is a slave. The number after `Sum:` is the checksum of the ROM; this number is important only if you want to change the software modules in the ROM. The dashes and letters after `Ver:` are the modules present in the matrix (for more information on software modules, see section 7.4, "Modules and Configuration Words").

```
 AutoPatch 4YDM CD
Pld:00410002
Sum:1373
Ver:-CMFA-NNNE-D—-
```

figure 1.9 Pld number of a master enclosure

```
 AutoPatch 4YDM CD
Pld:00020002
Sum:1373
Ver:-CMFA-NNNE-D—-
```

figure 1.10 Pld number of a slave enclosure



1.7 Voltage Selector

The voltage selector has two settings: 110V and 220V. The selector is located inside the fuse drawer on the 4YDM rear main. To change the setting on the selector, insert a flathead screwdriver below the slot on the fuse drawer and flip the tab up to release the drawer. Pull the fuse drawer out; the voltage selector is still in the fuse drawer slot. Place your flathead screwdriver underneath the voltage selector to pull it out. To change the voltage selection rotate the selector 180 degrees and replace it in the fuse drawer slot. Replace the fuse drawer.

Warning: Damage can occur during operation if the voltage selector is set incorrectly.

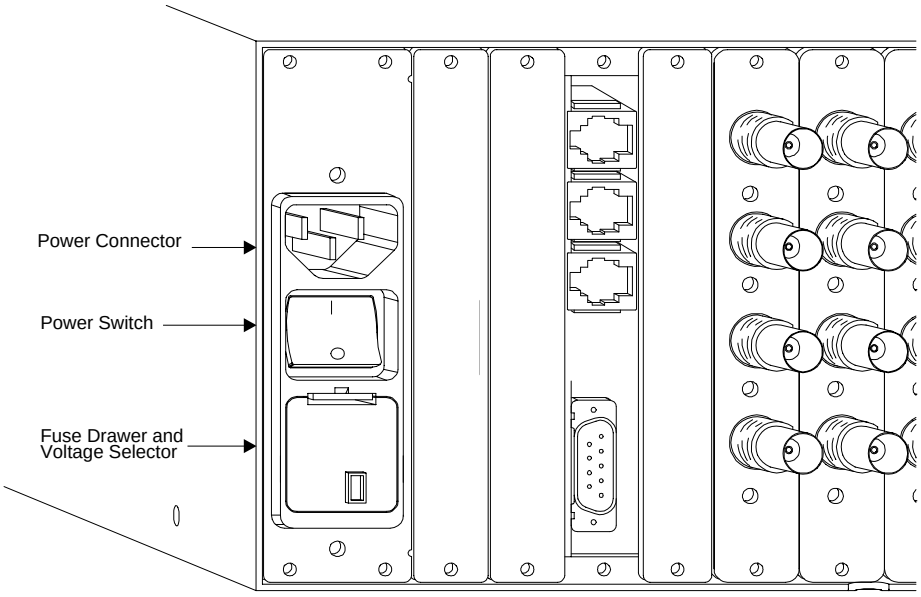


figure 1.12 Voltage selector, fuse, and power connector

1.8 1.8 Fuse

Every enclosure has a fuse drawer located just below the power switch. Enclosures containing standard video and audio boards will use a 1 AMP MDL Slow Blow fuse. See figure 1.12 for the location of the fuse drawer. To release the fuse insert a flathead screwdriver into the slot below the fuse drawer tab and lift the tab with the screwdriver. To install a new fuse, remove the fuse drawer and place the fuse in the correct fuse slot in the drawer, then replace the drawer.

1.9 1.9 Terminating Board

The fourth expansion board slot from the left contains a terminating board hidden behind the blank plate. This board properly terminates the incoming signal lines. The fourth expansion slot is the only slot designed to accept a terminating board. Do not remove this board unless it is being replaced.

1.10 Control Panel and Its Keys

At least one enclosure in the matrix should have a control panel, although it is not absolutely necessary. It is important to know which enclosure is the master because it "identifies" the other enclosures (all enclosures other than the master are slave enclosures). To determine whether an enclosure is a master or slave, see section 1.6, "Master and Slave Enclosures." Because an enclosure can only identify its own enclosure number, a control panel is always placed on the master (unless specified otherwise). If your 4YDM contains only one enclosure, it is the master.

The following sections briefly introduce you to the keys on the control panel. Further instructions for using these keys are located in Chapter 2, "Configuring the Inputs and Outputs" and Chapter 4, "Customizing the Distribution Matrix."

1.10.1 Change

<Change> is used to make a switch in the I/O configuration. When <Change> is pressed, the Change menu appears prompting you for information about the desired switch. For more information on altering the I/O configuration or <Change>, see Chapter 2, "Configuring the Inputs and Outputs."

1.10.2 Status

<Status> allows you to query the CPU to find out the specific logical input being routed to a logical output or the logical output(s) receiving a specific logical input. For more information on Status, see Chapter 3, "Status/Problem Solving."

1.10.3 Cancel

<Cancel> is used to delete a command that has not been completely entered. The 4YDM does not have a delete or erase key. If a mistake is made while entering a command, press <Cancel> to remove the partially entered command from the Command screen and re-enter the desired command. <Cancel> cannot be used to undo a completed operation.

1.10.4 Special

<Special> is used to execute Preset #1 or a user-specified command string. For more information about <Special>, see Chapter 2, “Configuring the Inputs and Outputs” and Chapter 4, “Customizing the Distribution Matrix.”

1.10.5 Preset

<Preset> allows you to store and recall up to 32 I/O configurations. For an explanation on setting and recalling presets, see Chapter 2, “Configuring the Inputs and Outputs.”

1.10.6 Backlight

The Backlight key illuminates the screen. This option is helpful in viewing the screen when there is little light. For further discussion about the backlight, see Chapter 4, “Customizing the Distribution Matrix.”

1.10.7 Space

The Space key is used to separate multiple outputs when entering a command; it is used only for outputs because no other prompt accepts multiple entries when entering a command.

1.10.8 Undo and Comma

<Undo> and <,> are not implemented at this time.

1.10.9 Period

<Period> is used to enter a literal operation. For more information on <.> and literal operations, see Chapter 7, “Advanced Features.”

1.10.10 Arrow Keys

The left, right, up and down arrow keys are used to move the cursor along a scrollable list that extends off the screen. When a list extends off the screen, an arrow appears pointing the way the list extends off the screen. Use the respective arrow key to scroll the extended list.

Chapter 2 - Configuring the Inputs and Outputs

This chapter explains the various ways to change the I/O configuration from the control panel. There are three keys that can be used to configure the inputs and outputs:

- Change
- Preset
- Special

Each key allows you to perform a different task in altering the I/O configuration; the difference in their use is the number of input and output signals they control. For example, if you are monitoring an output device and want to change the input to the output device you are monitoring, use <Change>. Use <Preset> to save or restore a known I/O configuration. Use <Special> to quickly execute redundant commands.

The Change screen, which appears when you press <Change>, allows you to specify the enclosure(s) that contains the inputs and outputs used in the switch. The Level prompt is used only for literal switches, which are discussed in section 7.1, "Literal Operations." All operations described in this manual prior to Chapter 7, "Advanced Features," are logical operations because they use logical inputs and outputs. Ignore the Level prompt until you are ready to do literal operations. A '0' often appears next to the Level prompt during logical operations. This is because the enclosure, or level, that contains the input and output signals is specified in the logical input and output definitions in the grouping table.

Note: If you are using a vertical interval sync signal and the signal is missing or somehow becomes detached, the next time you perform a switch a sync timeout error appears on the control panel screen. To continue operation, check the vertical interval sync connector and press <Cancel>. Your switches are still executed without the sync signal; however, the switch will be made without synchronization. For more information, see section B.4, "Sync Timeout."

2.1 Change Key

<Change> is used to alter the I/O configuration of the matrix by switching specified inputs and outputs. <Change> allows you to route an input to one or more outputs. To route more than one input at a time, see section 2.2, “Preset.”

Routing a Logical Input to a New Logical Output(s):

1. At the Command screen, press <Change>; the Change screen appears.

```
      CHANGE
Level
Input
Output
```

figure 2.1 Change screen

2. Press <Input>; ‘0’ appears after the `Level:` prompt and the cursor appears after the `Input:` prompt. Ignore the `Level:` prompt; levels pertain to literal switches only. The logical input and output level is specified in the logical definition of your configuration file. Enter the number of the input you wish to route.

```
      CHANGE
Level: 0
Input: 17
Output
```

figure 2.2 Enter an input

3. Press <Output>; the cursor appears after the `Output:` prompt. Enter the outputs that will receive the specified input. You may enter up to 32 outputs; be sure to place a space between multiple outputs. If you enter more outputs than can fit on the control panel screen, the `Output:` prompt slides to the left so the most recent output entered fits on the screen. If, at any time, you enter an incorrect number, press <Cancel> and enter the entire sequence of commands again.

```
      CHANGE
Level: 0
Input: 17
Output: 1 5 23 29
```

figure 2.3 Enter an output(s)

4. Press <Take>; the Command screen appears. The input is routed to the specified output(s).

Note: If you are using a vertical interval sync signal and the signal is missing or somehow becomes detached, the next time you perform a switch a sync timeout error appears on the control panel screen. To continue operation, check the vertical interval sync connector and press <Cancel>. Your switches are still executed without the sync signal; however, the switch will be made without synchronization. For more information, see section B.4, “Sync Timeout.”

Connecting Logical Outputs to a New Input:

1. Press <Change>; the Change screen appears.

```
      CHANGE
Level
Input
Output
```

figure 2.6 Change screen

2. Press <Output>; the cursor appears after the `Output:` prompt. 0' appears after the `Level:` prompt. Ignore the `Level:` prompt; levels pertain to literal switches only. Enter the assigned number of the output to receive a new input. The logical input and output level is specified in the logical definition of your configuration file.

```
      CHANGE
Level: 0
Input
Output: 15
```

figure 2.4 Routing a new input to output #15

3. Press <Input>; the cursor appears after the `Input:` prompt.
4. Enter the input that will be routed to the specified output. Only one input may be routed to an output.

```
      CHANGE
Level: 0
Input: 6
Output: 15
```

figure 2.5 Routing input #6 to output #15

5. Press <Take>; the Command screen appears. The output receives the specified input.

Note: If you are using a vertical interval sync signal and the signal is missing or somehow becomes detached, the next time you perform a switch a sync timeout error appears on the control panel screen. To continue operation, check the vertical interval sync connector and press <Cancel>. Your switches are still executed without the sync signal; however, the switch will be made without synchronization. For more information, see section B.4, "Sync Timeout."

2.2 Preset

Setting or "programming" a preset is similar to taking a picture of the current audio or video routings (I/O configuration). Presets are used to quickly restore a known I/O configuration. If a configuration is used often, set it as a preset. When a preset is set, the I/O configuration is assigned a specific preset number. When you wish to restore an I/O configuration, enter the preset number at the Preset Execute

screen. The enclosure is returned to the stored preset configuration. The preset will only restore the literal routings, it will not reset the logical tables. The CPU can store up to 32 presets.

Upon shipment, the 4YDM has no defined presets. The following instructions describe how to define and execute a preset. To understand how a preset works, define a preset and then make some switches. Execute the preset you defined and notice how the I/O configuration returns to the state it was at before the switches were made. If it is possible to distinguish a most often used or most important I/O configuration, define that I/O configuration as preset #1. Preset #1 is the most flexible of all presets because it can be executed with the Special key (see section 2.3.1, “Preset #1”).

Defining a Preset:

1. Using logical switches, set the I/O configuration.
2. At the Command screen, press <Preset>; the Preset Execute screen appears.

```
      PRESET

 Execute:
(Cancel)     (Take)
```

figure 2.7 Preset Execute screen

3. Press <Preset> again; the Preset Program screen appears.

```
      PRESET

 Program:
(Cancel)     (Take)
```

figure 2.8 Preset Program screen

4. Assign a number between 1 and 32 to the I/O configuration (the example screen assigns “12” to the current I/O configuration).

   Do not enter a number larger than 32; it will cause an error. If you enter a number assigned to an existing preset, the new preset definition will overwrite the existing preset. *AutoPatch* advises you to keep track of the preset definitions so you will not inadvertently overwrite a preset. Using the matrix, there is no way to check the numbers of existing presets or their I/O configuration.

```
      PRESET

 Program: 12_
(Cancel)     (Take)
```

figure 2.9 Define the current I/O configuration

5. Press <Take>; the specified preset number is defined as the current I/O configuration.

Executing a Preset:

1. At the Command screen, press <Preset>; the Preset Execute screen appears.

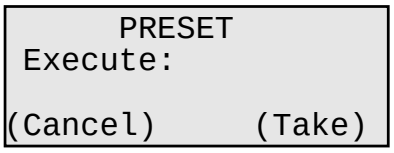


figure 2.10 Preset Execute screen

2. Enter the number of a preset.

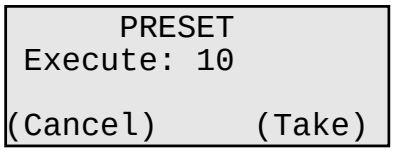


figure 2.11 Execute preset #10

3. Press <Take>; the defined I/O configuration is restored.

Note: When a preset is executed, the logical status table is not updated, which means that when you check the status of a logical input or output, the response on the status table may be incorrect. The literal status table is always correct. To check the literal status table, see Chapter 7, “Advanced Features.”

<Special> can be used to execute preset #1; for more information, see section 2.3, “Special.”

2.3 Special

<Special> is a programmable key which can be used to execute either preset #1 or a user-defined command string. <Special> is designed to be a time-saving feature; use it to quickly accomplish either of the programmable tasks. When the 4YDM is shipped, <Special> is set to execute preset #1.

<Special> also contains a “hot” feature. When the Hot Special Key option is on, <Special> immediately executes its programmed task. When the Hot Special Key option is off, a verification screen appears when <Special> is pressed. The verification screen is a safeguard against unknowingly changing the I/O configuration, because it allows you to verify that you are executing a programmed task. To execute the programmed task from the verification screen, press <Take>. When the 4YDM is shipped, the Hot Special Key option is turned off.

Making the Special Key Hot:

1. At the Command screen, press <Program>; the Program menu appears.

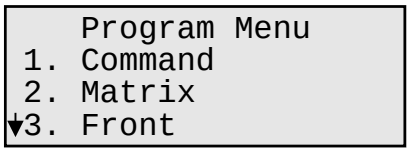


figure 2.12 Program Menu

2. At the Program menu, press <3>; the Front Config menu appears.

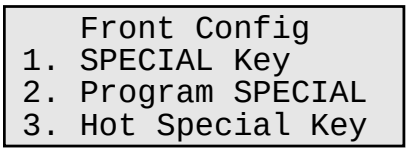


figure 2.13 Front Config menu

3. Press <3>; the Hot Special Key menu appears.

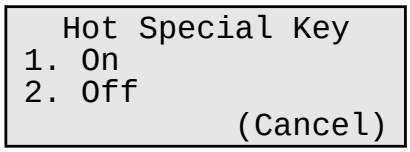


figure 2.14 Hot Special Key on/off screen

4. To turn the Hot Special Key option on, press <1>. To turn the Hot Special Key option off, press <2>. After setting the Hot Special Key option, the Command screen appears.

2.3.1 Preset #1

Setting <Special> to execute preset #1 can be a reset or safety feature. As an example of a reset feature, define preset #1 to be an often-used I/O configuration, so when preset #1 is executed, you know the matrix is in proper working order.

Use preset #1 as a safety feature by setting it to an I/O configuration that could be used to communicate information in times of an emergency. Whenever an emergency arises, press <Special> and communicate the necessary information. Upon shipment, the 4YDM has no presets defined unless specifically ordered. If <Special> is set to execute preset #1, be sure to define preset #1. For information on setting presets, see section 2.2, “Preset.”

Setting the Special Key Config to Execute Preset #1:

1. At the Command screen, press <Program>; the Program menu appears.

figure 2.15 Program Menu

2. At the Program menu, press <3>; the Front Config menu appears.

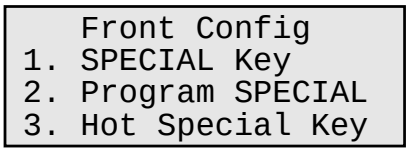


figure 2.16 Front Config menu

3. Press <1>; the Special Key Config menu appears.

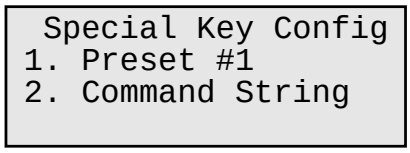


figure 2.17 Special Key Config menu

4. To set <Special> to execute preset #1, press <1>; the Command screen appears

Executing Preset #1 Using the Special Key:

1. <Special> must be set to execute preset #1. At the Command screen, press <Special>; the Special verification screen appears. If the Special verification screen does not appear, the Hot Special Key option is on and the Special Command has been executed (see Setting the Hot Special Key Option). The verification screen does not show the task it has been programmed to do; you are required to know the task.

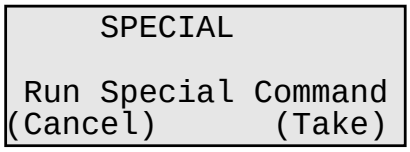


figure 2.18 Special Verification screen

2. To execute preset #1, press <Take>; the task executes and the Command screen appears.

2.3.2 Command String

You can set <Special> to execute any control panel command that affects the I/O configuration. Some examples are: checking the status of a particular output, checking for system errors, or executing a preset *and* then executing a change command. The command string can execute as many commands as will completely fit in the 31-digit area following the command entry prompt.

Setting the Special Key Config to Execute a Command String:

1. At the Command screen, press <Program>; the Program menu appears.

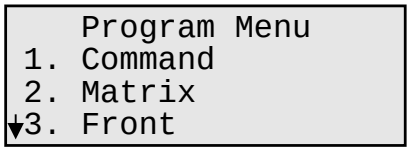


figure 2.19 Program Menu

2. At the Program menu, press <3>; the Front Config menu appears.

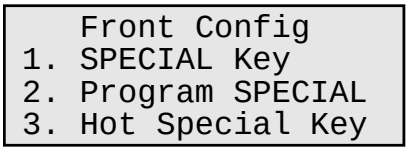


figure 2.20 Front Config menu

3. Press <1>; the Special Key Config menu appears.

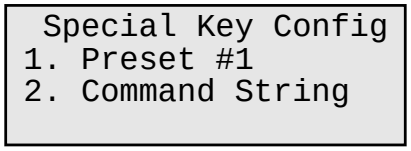


figure 2.21 Special Key Config menu

4. To set <Special> to execute a command string, press <2>; the Command screen appears. If you have not done so already, define a command string.

Defining the Command String:

1. At the Command screen, press <Program>; the Program menu appears.

```
   Program Menu
1. Command
2. Matrix
3. Front
```

figure 2.22 Program Menu

2. At the Program menu, press <3>; the Front Config menu appears.

```
   Front Config
1. SPECIAL Key
2. Program SPECIAL
3. Hot Special Key
```

figure 2.23 Front Config menu

3. Press <2>; the Program Special Key screen appears. If a command appears after the > prompt, it is the current command.

```
Program Special Key
>

(Cancel)   (Special)
```

figure 2.24 Program to execute a command

4. Enter a complete command string.

   The command string can be up to 31 characters long and must include all proper key presses. As you enter the command string, BCS commands appear on the command line (see section 5.3.1, "BCS," for more information). For example, the command string to execute Preset 17 would look like this:

```
Program Special Key
>R17T

(Cancel)   (Special)
```

figure 2.25 Execute preset #17

5. Press <Special>; the command string is stored and Command screen appears.

Executing a Command String Using the Special Key:

1. <Special> must be set to execute a command string. At the Command screen, press <Special>; the Special verification screen appears. If the Special verification screen does not appear, the Hot Special Key option is on and the Special Command was executed (see section 2.3, “Special”). The verification screen does not show the task it has been programmed to do; you are required to know the task.

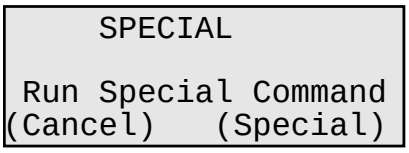


figure 2.26 Special verification screen

2. Press <Take>; the command string is executed.

Chapter 3 - Status and Problem Solving

The 4YDM monitors itself for internal errors and tries to prohibit any user actions that may cause errors to occur. However, the 4YDM cannot check for user-discrepancy errors.

You can monitor the I/O configuration through the Status screen by checking a specific logical input or a specific logical output. It is possible to find the status for all logical inputs and outputs by using the Advanced Packet Structure (APS) command, #A4 (request logical dump). For more information on this command, see section 5.3.2, "APS."

This chapter contains the steps for checking the status of the I/O configuration, how to use the matrice's problem-solving features, and what to do when certain error messages appear on the screen.

3.1 Logical Input/Output Status

Checking the status of a logical input or output is useful for problem solving. The CPU contains a logical status table which keeps track of the logical input and output connections. The logical status table is updated each time a logical switch is made so that the current logical input or output is returned each time the logical status is checked.

Logical operations do not include executing presets or any literal operation. Any time a preset is executed or a literal switch is made, the logical status table becomes invalid because it does not reflect the current I/O configuration. Literal switches and presets do not use the signal definitions that make up the grouping table. (for more information on literal operations, see section 7.1, "Literal Operations").

Checking the Logical Input Status:

1. At the Command screen, press <Status>; the Status screen appears.

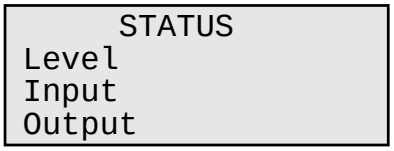


figure 3.1 Status screen

2. Press <Input>; the cursor appears after the `Input:` prompt and '1' appears after the `Level:` prompt. Ignore the number that appears after the `Level:` prompt. Levels are used strictly for literal operations.

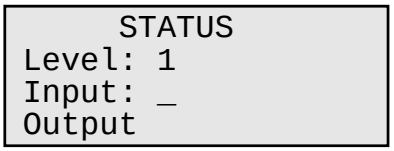


figure 3.2 Status screen

3. Enter the desired input and press <Take>. The output(s) that receives the specified input appears.

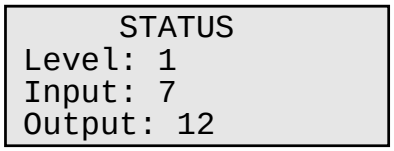


figure 3.3 Enter an input

4. To exit the Status screen, press <Cancel>; the Command screen appears.

Determining the Logical Output Status:

1. At the Command screen, press <Status>; the Status screen appears.

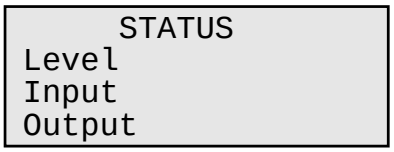


figure 3.4 Status screen

2. Press <Output>; the cursor appears after the `Output:` prompt and '1' appears after the `Level:` prompt. Ignore the number that appears after the `Level:` prompt.

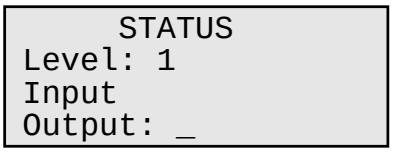


figure 3.5 Status screen

3. Enter the desired logical output number and press <Take>. The logical input that is routed to the specified logical output appears.

```
     STATUS
Level: 1
Input 7
Output: 12
```

figure 3.6 Enter the output number

4. To exit the Status screen, press <Cancel>; the Command screen appears.

3.2 Systems Online

When the matrix is powered, the master enclosure attempts to identify and establish communications with all slave enclosures that are linked together. The Systems Online screen refers to the number of slave enclosures that the master enclosure has recognized and with which it has established communications. For a discussion on master and slave enclosures, see section 1.6, “Master and Slave Enclosures.” If an enclosure is not responding or if there is a question as to whether or not the enclosure is online, use the following procedure.

Determining the Enclosures Online:

1. At the Command screen, press <Program>; the Program menu appears.

```
   Program Menu
1. Command
2. Matrix
▼3. Front
```

figure 3.7 Program Menu

2. Press <1>; the Command Config menu appears.

```
  Command Config
1. Reset Enclosure
2. System Errors
3. Systems Online
```

figure 3.8 Command configuration menu

3. Press <3>; the Systems Online screen appears listing the numbers of the enclosures that the CPU recognizes.

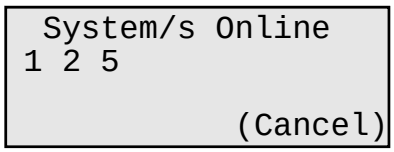


figure 3.9 Number of enclosures online

4. To exit the Systems Online screen, press <Cancel>.

3.3 Error Reporting

The 4YDM versions C5 and later provide five levels of error reporting. The following table lists the errors and information related to each. The Recoverable column lists whether or not the matrix can be repaired after suffering the respective error. The Startup and Runtime columns indicate when the error is reported. The Reported Globally column lists whether or not the error is reported on all control panel screens and external controllers of linked 4YDMs. If the error is not reported globally, it is reported only on the control panel screen of the 4YDM with the problem.

Error Name	Recoverable	Startup	Runtime	Reported Globally
Critical	No	Yes	Yes	No
Configuration	Yes	Yes	No	Yes
Syntax	Yes	No	Yes	No
Link	Yes	No	Yes	No
Operational	Yes	No	Yes	Yes

3.3.1 Critical Errors

If the master and slave enclosures are connected properly and the CPU hardware is undamaged, a critical error should never occur. If this error is encountered, copy down the numbers on the screen and call *AutoPatch* Customer Service immediately. Figure 3.10 is an example of a critical error screen.

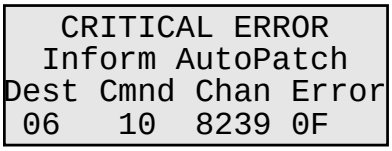


figure 3.10 Critical Error screen

3.3.2 System (Configuration) Errors

After applying power to the 4YDM, if the screen reports that there are System Errors, check those errors before continuing with normal operations. Even if the matrix appears to be operational, do not attempt normal operations before checking the errors. Press <Cancel> to return to the Command screen.

Determining the System Errors:

1. At the Command screen, press <Program>; the Program screen appears.

```
   Program Menu
1. Command
2. Matrix
3. Front
```

figure 3.11 Program menu

2. Press <1>; the Command Config menu appears.

```
  Command Config
1. Reset Enclosure
2. System Errors
3. Systems Online
```

figure 3.12 Command configuration menu

3. Press <2>; the System Errors screen appears. If the distribution matrix has any errors, they appear as shown in the example in figure 3.13. To repair system errors, follow the instructions in the section “Repairing the System Errors.”

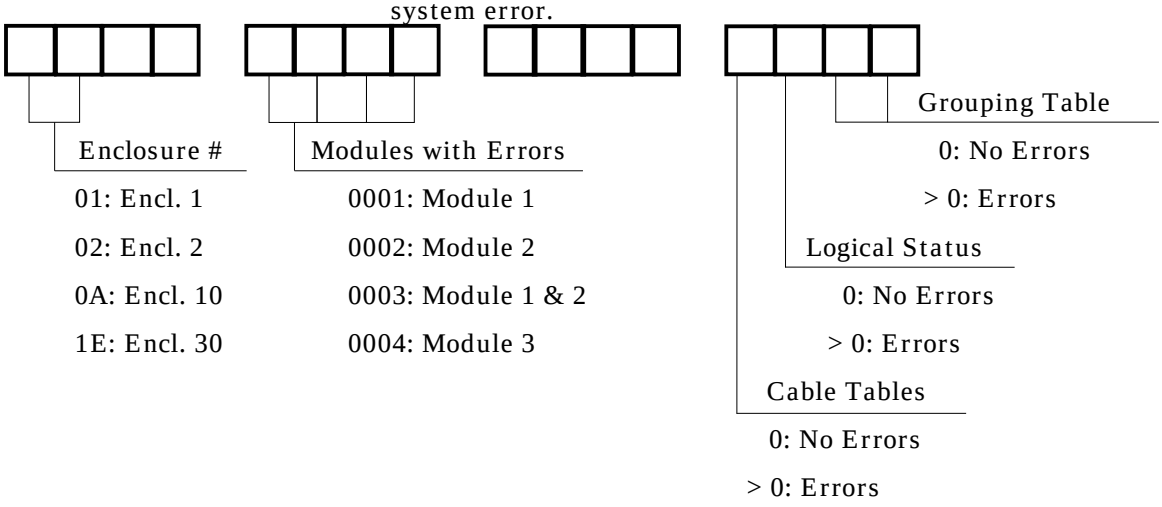
Note: If the error screen reports a recurring error and you understand how to transpose hexadecimal nibbles into bits, you can determine the software module(s) in which the error occurs by referring to “Identifying the Software Modules with Errors,” on page 3-6. After determining the software modules with errors, call *AutoPatch* for further instructions on solving the error(s). If you recognize an error that keeps recurring and you do not know how to transpose hexadecimal, note the System Error number(s) and call *AutoPatch*.

```
  System Errors
01000400 0000003E
0C000008 00000000
```

figure 3.13 System Errors screen

Identifying the Software Modules with Errors:

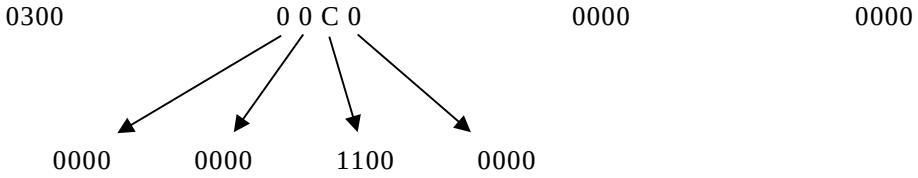
To determine the cause of a system error, use the table below to breakdown the error number. The system error number is comprised of 16 bits, or 4 nibbles. Using the second group of four nibbles from the left, you can identify the software modules with errors. To identify the module(s) with errors, break the four nibbles into 16 bits. The 16 bits represent how the CPU identifies the modules and creates that section of the system error.



1. After breaking the nibbles into bits, invert the order of the modules that appear in the startup screen and match them against the error number. This example uses the system error number:

   `0300  00C0 0000 0000`

2. The numbers in bold are the nibbles that identify the modules with errors. When these nibbles are broken into bits they are:



3. Now invert the entire order of the modules listed on the version screen and match them against the bits (see section 7.4, “Modules and Configuration Words,” to view the modules listing for the distribution matrix).

-	-	-	D	-	E	N	N	N	-	A	F	M	C	-
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

The software modules that are matched up with a ‘1’, contain an error. In the example above, the two network modules contain errors. After you discover the software modules in which the errors keep recurring, *AutoPatch* will be able to

quickly help you solve the problem. If it is not convenient to call *AutoPatch*, fix the errors using the instructions below and call when it is convenient.

Repairing the System Errors:

To fix the systems errors, upload the configuration file that reflects the way you want the distribution matrix to operate. The configuration file restores the distribution matrice's grouping tables to factory installed defaults. If you have not been updating a configuration file to the way you want the distribution matrix to work, upload the original configuration file sent with the distribution matrix. If errors still appear after uploading the configuration file, call *AutoPatch*.

Uploading the Configuration File:

1. To upload the configuration file, make a serial connection between an IBM or compatible PC, and the 4YDM.
2. If you have saved the contents of the floppy disk that was sent with the 4YDM to the PC's hard drive, at the DOS prompt, type the name of the configuration file and the communications port. Use the following format:

   ```
   C:\YTOOLS [file name] /com1 <or> /com2
   ```

   If you have not saved the contents of the floppy disk to the hard drive, insert the floppy disk into a disk drive and type the above command while substituting the correct disk drive on the command line.

For more information about uploading a configuration file, consult section 6.8, "Uploading a Configuration File."

Resetting the Configuration Words to the Factory Defaults:

User-defined settings can be lost as a result of a power surge or improper treatment of a CPU board. For example, removing the CPU board and setting it on a metal object causes the battery to short out which results in loss of information. This may cause system configuration errors in various modules. You can reset the configuration words to factory defaults through the control panel screen menus.

Note: Resetting configuration words to factory settings will clear any presets that may have been set previously.

When you reset the factory defaults, all module configuration words are set to zeroes (see section 7.4, "Modules and Configuration Words," for information on configuration words) which means the 4YDM's adjustable settings are restored to:

- If a numbered key is pressed first, the Status screen appears
- Literal changes are allowed
- <Special> executes Preset #1
- <Special> Hot Key function is turned off
- The serial filter for all serial ports is turned on
- The SBC polling range for all serial ports is #1
- SBC polling for all serial ports is turned off
- BAUD Rate for all serial ports is set to 9600

1. At the Command screen, press <Program>; the Program menu appears.

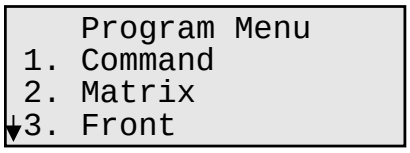


figure 3.14 Program menu

2. Press <1>; the Command Config menu appears.

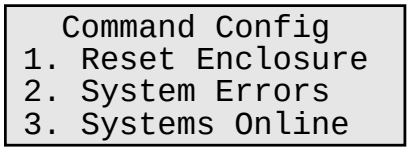


figure 3.15 Command configuration

3. Press the down arrow key until the screen scrolls down to the Default Config option (see below).

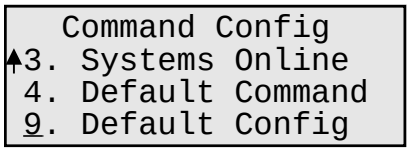


figure 3.16 Command configuration

Warning: If you reset all configuration words to their factory defaults, any changes you have made to the way the matrix operates will be lost. Polling will be turned off and any SBCs connected to any enclosure will not function until polling is re-established. See the table on page 3-6 to understand the settings of the matrix.

4. Press <9>; the verification screen appears.

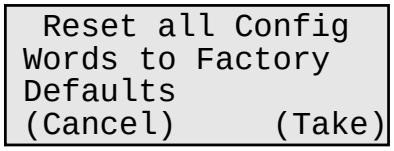


figure 3.17 Reset all configuration words

5. Press <Take>; the 4YDM's settings return to the factory defaults. If the configuration file does not reflect the desired settings of the 4YDM, you must manually set the adjustable features of the 4YDM to the desired setting after resetting the configuration words to the factory defaults.

3.3.3 Syntax Errors

There are three possible error screens that appear if incorrect syntax is used when entering a command. These error screens are listed in the following sections. To correct a syntax error, press <Cancel> and enter the command again.

3.3.3.1 Range Error

When a number that is out of range for a specific operation is entered, one of the following error screens appears. Logical inputs and outputs can range from 1 - 128. The range for Presets, inputs, and outputs that are used for literal operations is from 1 - 32 (see section 7.1, “Literal Operations”).

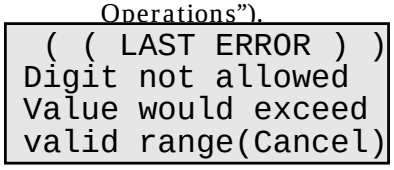


figure 3.18 Out of range error screen

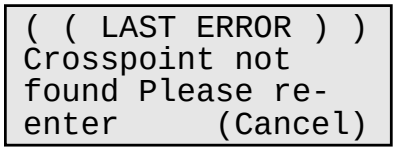


figure 3.19 Out of range error screen

3.3.3.2 Multiple Command Error

Only one command can be entered at a time. Any time the distribution matrix is requested to execute more than one command at a time, the following error screen appears.

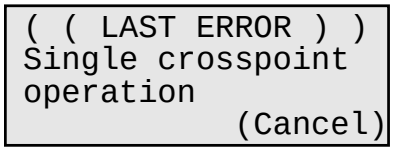


figure 3.20 Multiple command error screen

3.3.4 Link Errors

Link errors should not be encountered during normal operations. A link error is caused because the communication link from one enclosure to another has been broken, jarred, or removed. A link error screen appears as soon as the break in communications occurs. To fix the link error, press <Cancel>; if the communication error is internally resolved, the Command screen appears. If the error persists, check the cable listed on the screen.

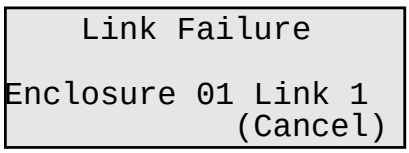


figure 3.21 Link errors

In this example, the first enclosure has a break in the cable leaving it’s first link port. Link ports are numbered from the top to the bottom, starting at 1.

3.3.5 Operational Errors

Operational errors are very similar to syntax errors because they are caused by attempting an operation that cannot be done. The difference between an operational and syntax error is that when attempting a task that causes an operational error, the CPU checks the validity of the operation. An operational error screen alerts you that the command that was entered could not be executed for a variety of reasons, including: no such logical input or output exists, or the incorrect security password was entered (the security password is associated with the optional Security module). To fix an operational error, press <Cancel> and enter the command again.

3.4 Common Installation/Maintenance Errors

The following sections describe some common errors that occur during installation or maintenance. The following errors are listed by the result. These results can appear on the screen or can be apparent when viewing the matrix.

Screen Shows Two Dark Lines

CPU board is inoperative, but power is applied to the distribution matrix. Check to make sure the CPU is seated properly.

Screen is Blank and Backlight is Illuminated

The control panel signal cable may be reversed.

Screen is Blank on Power Up

Verify that each enclosure has power properly supplied and the enclosure's fuse is not blown.

Additional Enclosures Not Responding

Check that the link cables between enclosures are properly connected. If all link cables seem in order, reboot the distribution matrix and check the numbers of the enclosures that the master enclosure recognizes. Check the fuses of the enclosures the master enclosure does not recognize.

Inputs Do Not End Up on the Proper Output Device

Be sure the input and output signal wires are attached exactly as they are shown on the ***AutoPatch* 4YDM Connector and Groupings Guide.** If any input or output boards were removed, switched, or added, ensure that they were correctly replaced in the enclosure (see section 1.3, “Input and Output Boards”).

Sync Timeout Error Appears

This error appears during the matrix's startup sequence if the sync signal was disconnected or missing while making a logical switch before power was cycled. Although your switches are still executed, the lack of synchronization may create a noticeable roll in video output devices.

Check the sync connector, fix the Matrix module's configuration word (by entering '0000' at the Matrix Config screen; see section 7.4.2, "Matrix Module" for more information on this configuration word), and re-start the matrix. If the error reappears, your sync signal generator may not be working properly.

If this error appears and you are switching a signal that should not require sync, check its entry in your configuration file. See sections 6.3, "Groupings," and B.4, "Sync Timeout."

Chapter 4 - Customizing the Distribution Matrix

To make the 4YDM as user-friendly as possible and speed up operation, you can set several features of the 4YDM to work the way you want. This chapter describes the menu-driven adjustable features and how to set each of them. These features are:

- Default command screen when a numbered key press begins a command string
- Making <Special> hot
- Command executed when <Special> is pressed
- BAUD Rate setting
- Setting Async (serial) filters on or off
- Setting the SBC polling range

4.1 Default Command Screen

The Default Command Screen is an option designed to help you speed the time it takes to execute an action from the Status, Change, or Execute Preset screens. Instead of opening the Status, Change, or Preset screens and specifying input, output, or preset number(s) for that operation, at the Command screen, you can press a numbered key which opens the Default Command Screen and enters the number in that screen.

When the 4YDM is shipped, the Default Command Screen is set to the Status screen. If you press <7> from the Command screen, the Status screen opens and '7' appears after the Input prompt. Press <Take> and the logical outputs receiving input 7 appear after the Output prompt.

When the Default Command Screen is set to the Status or Change screens, the numbers that were entered appear next to the Input prompt. The numbers cannot be set to appear next to the Output prompt. When the Default Command Screen is set to the Preset screen, only the Execute Preset screen can appear.

Setting the Default Command Screen:

1. At the Command screen, press <Program>; the Program menu appears.

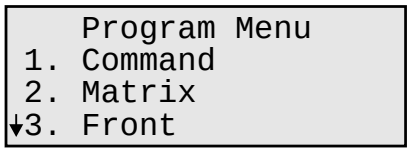


figure 4.1 Program menu

2. At the Program menu, press <1>; the Command menu appears.

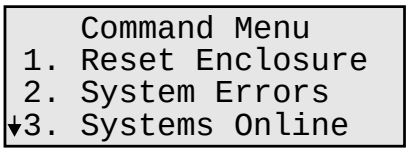


figure 4.2 Command menu

3. Using the down arrow key, scroll down to the Default Command option.

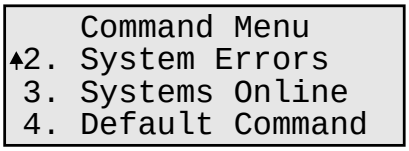


figure 4.3 Scroll down to the Default Command

4. Press <Take>; the Default Command screen appears.

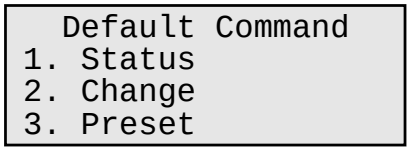


figure 4.4 Default command menu

5. Arrow down to the desired default command option and press <Take>. The selected default command is set and the Command screen appears.

Executing a Command Using the Default Command Screen:

1. At the Command screen, press the number of the desired logical input or preset. If you are entering a two-digit number, the default command screen appears after the first digit is entered. The number entered appears next to the `Input:` prompt in a Status or Change screen, and appears next to the `Execute:` prompt in a Preset screen.

```
      STATUS
Level: 1
Input: 12
Output
```

figure 4.5 Pressing #12 activates the default command

Note: If you are entering a Change command, you must also enter a logical output.

2. Press <Take>; the command is executed. If you execute a Status command, the logical output(s) that receives the specified logical input appears. If you execute a Change command, the switch is made and the Command screen appears. If you execute a Preset, the switches are made and the Command screen appears.

```
      STATUS
Level: 1
Input: 12
Output: 7
```

figure 4.6 Output 7 receives input 12

4.2 "Hot" Special Key Option

When <Special> is "hot", it executes its programmed task as soon as it is pressed (for information on programming <Special>, see section 4.3, "Function of the Special Key"). If <Special> is not "hot", when it is pressed, the Special verification screen appears.

```
     SPECIAL

 Run Special Command
(Cancel)      (Take)
```

figure 4.7 Special Verification screen

Making the Special Key Hot:

1. At the Command screen, press <Program>; the Program menu appears.

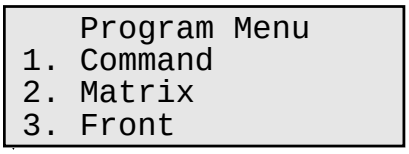


figure 4.8 Program Menu

2. At the Program menu, press <3>; the Front Config menu appears.

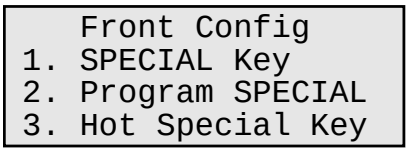


figure 4.9 Front Config menu

3. Press <3>; the Hot Special Key menu appears.

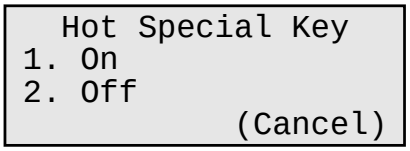


figure 4.10 Hot Special Key on/off screen

4. To turn the Hot Special Key option on, press <1>. To turn the Hot Special Key option off, press <2>. After setting the Hot Special Key option, the Command screen appears.

4.3 Function of the Special Key

<Special> is a key used for shortcuts. Use <Special> to quickly execute preset #1 or a specified command string. See section 2.3, “Special,” for instructions for setting the task and operation of the key.

4.4 BAUD Rate Setting

The 4YDM responds to 1200, 2400, 9600, and 19200 BAUD. You can set the matrix to any of these BAUD rates, although *AutoPatch* recommends you use 9600 BAUD. For instructions on serial communications and attaching an external controller, see section 5.1, "Attaching an External Controller."

Note: If you are using single bus controllers (SBC), they are set at the factory for 9600 BAUD communications, although SBCs that respond to a different BAUD rate can be specifically ordered. BAUD rate changes cannot be made while enclosures are polling their SBCs. Be sure to turn off polling before changing BAUD rate and then return polling to on. For more information on SBC's refer to Appendix C, "Single Bus Controllers."

Setting the BAUD Rate for the 4YDM:

1. Press <Program>. The Program menu appears.

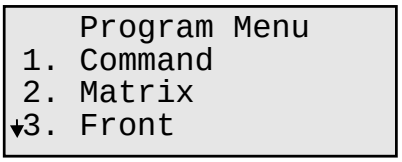


figure 4.11 Scroll to the async module(s)

2. Press the down arrow until the screen scrolls to the Async1 module.

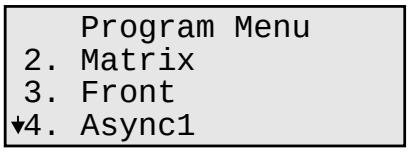


figure 4.12 Scroll to the async module(s)

3. When the cursor is on the Async1 module; press <Take>. The Async1 Config screen appears.

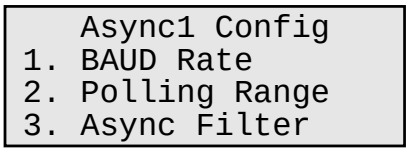


figure 4.13 Async config menu

4. Press <1>. The BAUD Rate Selection screen appears.

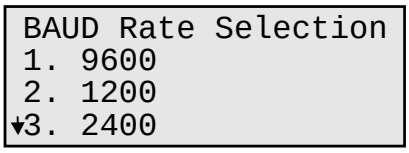


figure 4.14 BAUD Rate Selection menu

5. Press the down arrow until the cursor scrolls to the desired BAUD rate. Press <Take>; the Command screen appears. The BAUD rate has been set.

4.5 Async Filter

The async filter translates the communications going out the serial port into printable character codes in ASCII format. The CPU responds to ASCII codes in hexadecimal format; these codes are always filtered into printable codes before they are displayed on the control panel screen. If the async filter is turned off, the communications going out the serial port are not filtered and remain in hexadecimal APS format (for more information, see section 5.3.2, “APS”). These codes can be used in various programs that interact with the 4YDM. When the async filter is turned on, the communications going out the serial port are filtered into printable ASCII codes. Unless you have specific needs for sending unfiltered communication out the serial port, *AutoPatch* recommends you keep the async filter turned on.

Turning the Async Filter On and Off.

1. At the Command screen, press <Program>.

```
   Program Menu
1. Command
2. Matrix
3. Front
```

figure 4.15 Program menu

2. Each enclosure with a CPU has one async software module, which will be either the fourth or fifth choice on the Program menu. At the Program menu, press either <4> or <5> to set the async filter for the serial port. (The following example sets the async filter.)

```
   Async1 Menu
1. BAUD Rate
2. Polling Range
3. Async1 Filter
```

figure 4.16 Async menu

3. At the Async1 menu, press <3>; the Async Filter screen appears.

```
  Async1 Filter
1. On
2. Off
          (Cancel)
```

figure 4.17 Async1 filter on/off screen

4. At the Async Filter screen, press <1> to turn the async filter on; press <2> to turn the async filter off.

4.6 SBC Polling Range

Polling is the process of the CPU servicing each Single Bus Controller (SBC). When the CPU services a SBC, it checks for change or routing function requests.

If SBCs are used with the matrix, it is necessary to set the SBC polling range so that the matrix polls the correct number of SBCs. The polling range is controlled by the Async (or serial port) module.

4.6.1 Polling

Set the polling range in the Async module.

Setting the SBC Polling Range:

1. At the Command screen, press <Program>. The Program menu appears.

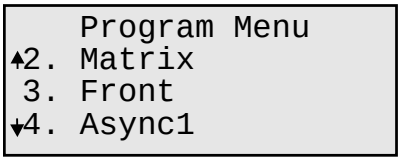


figure 4.18 Program menu

2. Arrow down to Async1 and press <Take>. The Async Config screen appears.

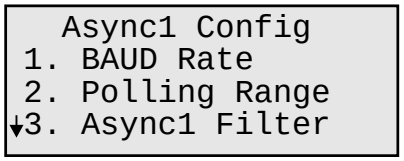


figure 4.19 Async1 Configuration menu

3. To select the Polling Range option, press <2>. The Enclosure Specification screen appears.

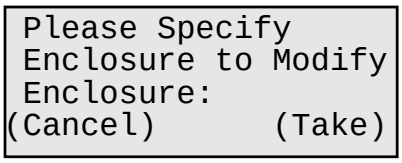


figure 4.20 Enclosure Specification screen

4. The cursor appears at the 'Enclosure:' prompt. Enter the number of the enclosure you would like to modify. It is not possible to select 0 or 1 from this screen (refer to "Polling a Master"). Press <Take>. The Enter Polling Range screen appears.

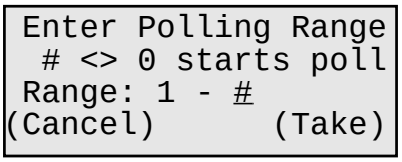


figure 4.21 Async1 Configuration menu

Note: Entering any number other than 0 for the polling range establishes an RS 485 interface on the serial port (Async1). Zero converts it back to an RS 232 interface and turns polling off.

5. The cursor appears on the '#'. Enter the number of the highest numbered SBC and press <Take>. The polling range is set and the Command screen appears. The highest numbered SBC does not refer to the physical number of SBCs being polled; the highest numbered SBC corresponds to the highest CPU assignment on the rear of the SBCs. For more information see appendix C, section C.2, "CPU Assignments."

4.6.2 Polling on a Master

Polling should occur on a slave cage (if your system has slave cages) rather than the master, reserving the master for serial communication from a PC or other RS 232 device; however, the master serial configuration can be set to enable polling and RS 485 communication as follows:

Setting the Master to Poll:

1. At the Command screen, press <Program>. The Program menu appears.

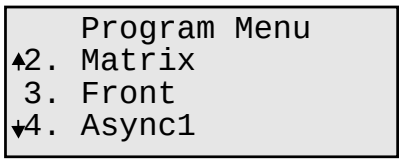


figure 4.22 Program menu

2. Arrow down to Async1 and press <Take>. The Async Config screen appears.

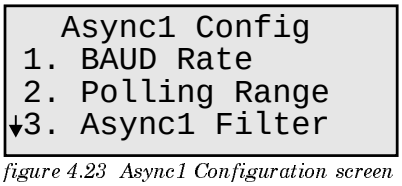


figure 4.23 Async1 Configuration screen

3. Press <Special> (or serially, enter P4*). The Async1 Config # screen appears.

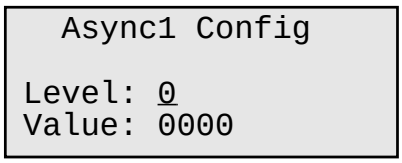


figure 4.24 Async1 Configuration # screen

4. Enter '1' at the Level prompt.
5. Enter '0771' (polling range is 32 SBCs) at the Value prompt. Refer to section 7.4.4, "Async Module," for other values.
6. Press <Take> to store the value.

A single cage system should have a control panel if it will be polling. The control panel is the only way to communicate with the system other than using SBCs.

4.7 Backlight

The backlight allows you to illuminate the control panel screen. To turn the backlight on, press <Backlight>; to turn the backlight off, press <Backlight> again. The backlight turns off automatically after 20 seconds of control panel inactivity. The backlight cannot be set to stay on continuously during periods of inactivity. *AutoPatch* recommends you use the backlight only when needed. The lifetime of the backlight is rated at 2000 hours.

4.8 Power Backup

The 4YDM is equipped with a battery backup so, in cases of power failure or if you wish to unplug the 4YDM, the items in memory will not be lost. The life expectancy of the battery is 10 years.

AutoPatch recommends you use a surge protector.

Chapter 5 - External Control

Your distribution matrix accommodates several control options, including an X/Y Control Panel, serial controllers, and dry contacts. An X/Y Control Panel may be either local or remote, but is not considered a form of external control. An external controller can be a serial controller or dry contacts. A serial controller can be a Single Bus Controller (SBC) or any device that can communicate with your matrix via the serial port. Dry contacts are single-direction controllers wired directly to the CPU board, and are used to execute a specific preset. Each type of external controller is specialized for different types of applications, but they all allow you to access your distribution matrix from a remote location.

This chapter covers the following items related to external control:

- Attaching a Serial Controller
- Echoing Serial Commands
- Control Languages
- External Control Software
- Dry Contacts

5.1 Attaching a Serial Controller

Some specific types of serial controllers are SBCs, PC or Macintosh computers, and third party control devices. A distribution matrix can support only one type of serial controller at a time, although it can have a combination of a serial controller and dry contacts.

A PC or Macintosh computer can be used to control your distribution matrix with any program that can transmit at 1200, 2400, 9600, or 19200 BAUD (9600 is the default) and send BCS and/or APS commands. AutoPatch has the following software available to help you control your matrix: YTOOLS (DOS & Windows versions), 4YROUTE, AutoRoute, WinRoute, and ScanPatch.

To attach a serial controller:

To attach a serial controller to your distribution matrix, connect an RS-232, -422, or -485 null modem serial cable between the controller and the serial port on an enclosure (preferably the master enclosure). Transmit, receive, and ground must be run from the controller to the distribution matrix. Make the serial connections as seen in figures 5.1 - 5.5.

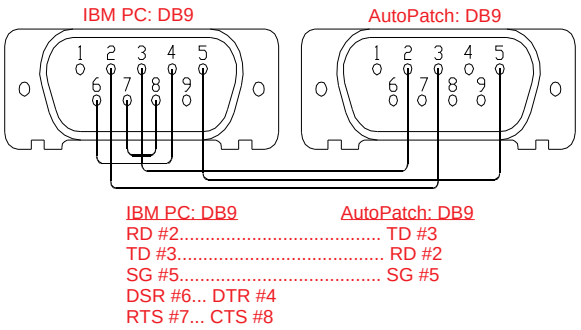


figure 5.1 RS232 communications for an IBM PC

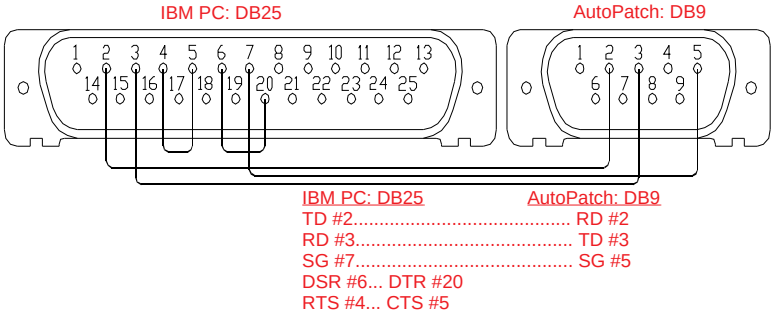


figure 5.2 RS232 communications for a 25pin connector

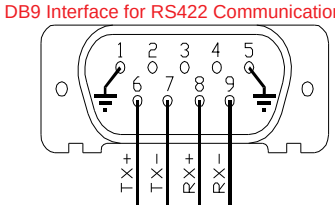


figure 5.3 RS422 communications

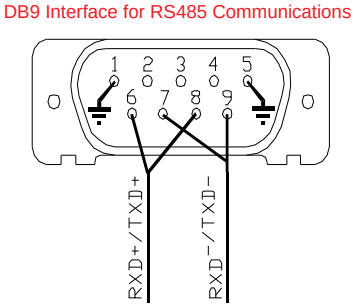


figure 5.4 RS485 communications

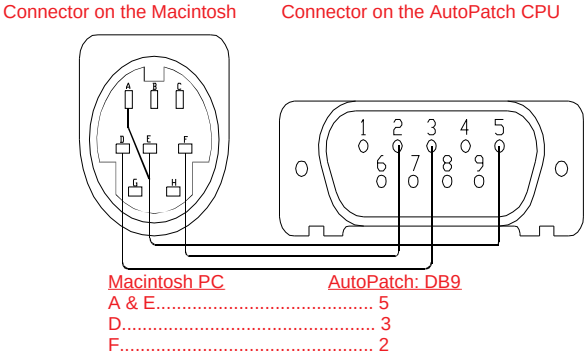


figure 5.5 RS232 communications for a Macintosh

Figure 5.1 shows an RS-232 standard wiring diagram for a DB9 port. In most situations, connecting pin 2 of the PC end to pin 3 of the AutoPatch end, pin 3 of the PC end to pin 2 of the AutoPatch end, and pin 5 to ground, is all that is required to operate your 4YDM. Figure 5.4 shows an RS-485 standard wiring diagram. RXD+/TXD+ to pin 8 and RXD-/TXD- to pin 9 with 5 to ground will also work with the 4YDM.

Before attaching an external controller to an enclosure, make sure the BAUD rate settings for the external controller and the enclosure correspond. Also make sure the cable and the communication settings are the same. The factory default BAUD rate is 9600 and communication protocol is eight bits, no parity, and one stop bit. For instructions on setting the BAUD rate, see section 4.4, “BAUD Rate Setting.”

Note: Normally SBCs respond only to commands given at 9600 BAUD, although SBCs that respond to a different BAUD rate can be specifically ordered. BAUD rate changes cannot be made while enclosures are polling their SBCs. Be sure to turn off polling before changing the BAUD rate and return polling status after changing the BAUD rate.

5.2 Echoing Serial Commands (Control Panel Echo)

The Control Panel Echo command allows matrix menus and commands to be echoed on the matrix’s X/Y Control Panel while the commands are entered from an external controller. Use this command for troubleshooting an external controller and for redundant control. The Control Panel Echo command is turned on and off in the Async configuration word, not from a menu prompt.

The Async configuration word is a hexadecimal number that tells the matrix the BAUD rate setting, the number of SBCs to poll, and whether or not to echo serial commands on the control panel. Change the settings in this configuration word by entering an appropriate four-digit hexadecimal number in the Async1 Config # screen. See section 7.4, “Modules and Configuration Words,” for more information on configuration words. The instructions below are specific to the Async configuration word and its settings.

Changing the Async Configuration Word to Echo Serial Commands on the X/Y Control Panel:

1. At the Command screen, press <Program>. The Program menu appears.

```
   Program Menu
 1. Command
 2. Matrix
↓3. Front
```

figure 5.6 Program menu

2. Using the down arrow, scroll down to the Async command and press <Take>. The Async Config screen appears.

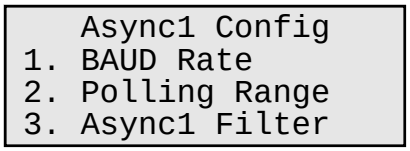


figure 5.7 Async1 configuration menu

3. Press <Special>. The Async Config # screen appears.

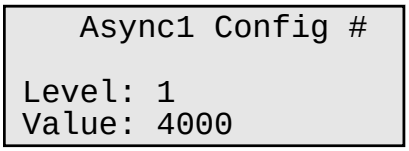


figure 5.8 Async1 configuration number entry screen

4. Enter the appropriate four-digit hexadecimal number for the settings you desire and press <Take>. The Command screen appears and your changes are made.

   The following table shows some sample Async Configuration words in binary and hexadecimal and an explanation of each configuration word. For more details see section 7.4.4, “Async Module.”

Binary	Decimal	Effect
0000 0010 0101 0100	254	Echo off, BAUD rate 9600, Polling off, Polling range 22, Serial Filter off, Serial Mode RS 232
0110 0111 0111 0001	6771	Echo on, BAUD rate 2400, Polling on, Polling range 32, Serial Filter on, Serial Mode RS 485

5.3 Control Languages

Your 4Y distribution matrix can receive commands from an external serial controller that uses either Basic Control Structure (BCS) or Advanced Packet Structure (APS) commands. Both languages were developed by AutoPatch for controlling AutoPatch matrices. BCS is a set of alphanumeric characters that simulate entering commands from an X/Y Control Panel. APS is a set of more complex binary commands used to gain greater matrix control than BCS commands alone.

5.3.1 BCS

Note: AutoPatch *strongly* suggests implementing software handshaking as defined in section 5.3.1.2, "BCS Software Handshaking."

BCS is a set of alphanumeric characters that allow a PC or a controlling device to simulate control panel commands. If you use a PC to control your 4Y, you must use serial communications software. The control panel LCD displays the commands as they are entered at the PC. The following table shows the PC key, the simulated control panel function, and a short description of the function.

Key	Function	Description
X	Cancel	Cancels the previous incomplete command
T	Take	Executes command
S	Status	Allows you to check the status of the 4YDM's input and output locations
L	Level	Level refers to a particular enclosure. Levels are used for literal operations.
C	Change	Change command
I	Input	Input number entry
O	Output	Output number entry
R	Preset	Executes the preset configuration
*	Special	Executes the programmed feature. See Chapter 4, "Customizing the Distribution Matrix," for an explanation
RR	Define Preset	Allows you to define a preset configuration
' '	Space	A delimiter for separating multiple input and output entries
.	Period	Executes literal commands and preset definitions. See Chapter 7, "Advanced Features" for information on literal switches
'0' - '9'	Number	Digits used to define inputs and outputs
:	Number	':' serves as the number 10 because two-digit numbers cannot be entered at the Program menu
;	Number	';' serves as the number 11
<	Number	'<' serves as the number 12
=	Number	'=' serves as the number 13
>	Number	'>' serves as the number 14
?	Number	'?' serves as the number 15
P	Program	Opens the Program menu

5.3.1.1 Entering BCS commands

Entering BCS commands from an external controller is similar to entering commands from the control panel. Each serial controller has its own user interface that sends BCS commands to the matrix. For example, on a PC you can use the terminal emulation utility in YTOOLS for Windows as your user interface for sending BCS commands.

One advantage of using BCS commands is the ability to enter multiple commands in the same string. For example, you can recall preset 10 and then switch input 1 to outputs 6, 7, and 8, all with one command string:

R10TCI1O6 7 8T

To enter BCS commands:

Enter BCS commands in the same sequence you would enter commands from the control panel. All BCS entries must be complete commands. Terminate any incomplete or erroneous commands with an 'X' (Cancel) and then re-type the command(s). Typing a space does not cancel a previous command.

The following table shows examples of commands.

Entry	Description
CI2O23T	Change input 2 to output 23.
RR20T	Define the current I/O configuration as Preset #20.
CO1 2 3 4 5 6I20T	Change input 20 to outputs 1,2,3,4,5, and 6.
R17T	Execute Preset #17.
CI5XCI6O4T	Change input 6 to output 4. Notice that the command to change input 5 was canceled.
R1XR7T	Recall and execute Preset #7. Notice that the command to execute Preset #1 was canceled.
P1*00001	Opens the Program menu and selects the Command module. The '*' opens the configuration word menu for the Command module. The first '0' means that the following configuration word will work for all enclosures. The four remaining numbers set the Command configuration word.

5.3.1.2 BCS Software Handshaking

BCS software handshaking prevents serial buffer overflow when using a serial external controller and BCS commands.

Each character the CPU reads is returned through the serial port to the external controller. When a logical switch is made, the BCS command 'T' is sent from the matrix to the controller software. The software should wait until it receives the echoed 'T' back from the CPU before it sends the next command.

5.3.2 APS

Note: AutoPatch *strongly* suggests implementing software handshaking as defined in section 5.3.2.17, "APS Software Handshaking."

APS commands give you access to operations not available in any other control structure. Other benefits of APS are as follows, it:

- is packetized (each packet has an associated length and checksum)
- can be verified
- is 2-3 times faster than BCS commands
- expands your capabilities beyond BCS commands and control panel operations

When creating custom control software, use APS commands to make switches, request information, and display messages on the control panel LCD. APS commands are non-readable ASCII codes that can be sent to the matrix via software that supports APS structure or in the packet section of the configuration file. Because they are non-readable ASCII codes, APS commands cannot be entered from a text editor or terminal emulation program. A good understanding of hexadecimal numbers and byte structure will help you use APS commands. Please note, if you are using or creating custom control software, APS commands are the preferred method of communicating with the 4Y.

One method of sending APS commands to the CPU, is to include them in the packet section of a configuration file. Section 6.6, "Packets," describes how to include an APS command in a configuration file and section 6.8, "Uploading a Configuration File," describes how to send the configuration file to the CPU.

The following table lists the APS commands and their descriptions. (This manual uses '#' to identify hexadecimal numbers.)

Command	Description	Domain
#A0	Request enclosure config errors	Direct
#A1	Answer enclosure config errors	n/a
#A2	Request literal dump	Direct
#A3	Answer literal dump	n/a
#A4	Request logical dump	Local
#A5	Answer logical dump	n/a
#A6	Publish Input Status	Global
#A7	Publish Output Status	Global
#AF	Command String	Local

#B0	Publish Text String	Global
#B1	Background Switch	Direct
#B4	Request enclosures online	Direct
#B5	Answer enclosures online	n/a
#B6	Request version	Direct
#B7	Answer version	n/a
#B8	Background Switch	Direct

Construct each APS command string using the following command template:

Command	Length	Data	Checksum

The **Command** section contains the hexadecimal name of the command you want to execute. For example, if you want to request a literal dump, the command section would contain #A2 (the hexadecimal number for that command).

The **Length** section identifies how many bytes are contained in the data section of the command string. For example, if you are sending command #A2 (Request for Literal Dump), the length section will contain the value #01 because there is only one byte in the data section of that command.

The **Data** section contains specific information regarding the command you wish to execute. For example, in the #A2 command, the data section contains the number of the enclosure you are requesting a literal dump from. APS commands hold up to 64 bytes of information in the data section.

The **Checksum** is the least significant byte of the sum of all bytes from the command to the end of the data section. For example, if the command section (up to the checksum) looked like this: B8 06 00 0A 0B 00 1E 1F, the sum would be the value #0110. Since we use only the least significant byte, the value we would use is #10. Checksums are used to help validate that what was intended to be sent to the enclosure is what the enclosure received.

After sending an APS command to an enclosure the enclosure will return the checksum it calculated from your packet. If the checksum received matches the checksum you calculated, then the packet was received correctly. If they differ,

a.) your command was not executed, and

b.) an error would appears on the front panel, and finally

c.) you will have to resend the packet.

APS commands can be integrated into software written in any programming language. The following are sample sections of C code utilizing the Background Switch commands (#B1 & #B8), Request Logical Dump(#A4), and Answer Logical Dump (#A5) commands. Appendix E, “APS Command Sample Program,” has a sample C program that utilizes several APS commands.

Example 1 is sending APS command #B1 (Background Switch Command). Please note that this command has been replaced by #B8 but is still available for backward compatibility. Please use command #B8 in new software.

Example #1

```
// Sending APS B1 Background Switch Command.

publish_test_msg( “APS B1 Cmnd” );
 printf(“\nPerforming APS, B1 Background Switch Command, Logical 8-4.”);

c_putc(p0,0xB1);
c_putc(p0,0x02);
c_putc(p0,0xC3);
c_putc(p0,0x83);
c_putc(p0,0xF9);

_bios_timeofday(0,&st);
err=EOF;
while ((ct<24)&&(err!=0xF9))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}

if (err==EOF)
{
printf(“\nFailure Sending APS B1 Command.\n”);
printf(“\nResponse from CPU was not received within allotted time frame.”);
exit(1);
}
```

Example 2 is sending APS command #B8 (Background Switch Command). This command replaces the #B1 (Background Switch Command). Please use this command in new software.

Example #2

```
// Sending APS B8 Background Switch Command.

publish_test_msg( “APS B8 Log Cmnd” );
printf(”\nPerforming APS, B8 Background Switch Command, Logical 3-5, 4-6.”);

c_putc(p0,0xB8);
c_putc(p0,0x06); // Two Switches
c_putc(p0,0x00);
c_putc(p0,0x02); // Input 3
c_putc(p0,0x04); // Output 5
c_putc(p0,0x00);
c_putc(p0,0x03); // Input 4
c_putc(p0,0x05); // Output 6
c_putc(p0,0xCC);

_bios_timeofday(0,&st);
err=EOF;
while ((ct<48)&&(err!=0xCC))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}

if (err==EOF)
{
 printf(“\nFailure Sending APS B8 Command, Logical Test.\n”);
 printf(“\nResponse from CPU was not received within allotted time frame.”);
 exit(1);
}
```

Example 3 uses the #A4 (Request Logical Dump) command to request the logical status of the logical inputs and outputs of an enclosure, then uses the #A5 (Answer Logical Dump) command to receive the logical status list from the enclosure.

Example #3

```
// Download Logical Packets
c_putc(p0,0xA4);
c_putc(p0,0x01);
c_putc(p0,0x01);
c_putc(p0,0xA6);

_bios_timeofday(0,&st);
err=EOF;
while ((ct<24)&&(err!=0xA6))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}
if (err==EOF)
{
 printf("\nFailure Verifying BCS Packet. Performing Req Logical Download.");
 printf("\nResponse from CPU was not received within allotted time frame.");
 exit(1);
}
if (err != 0xA6)
{
 printf("\nFailure Verifying BCS Packet. Performing Req Logical Download.");
 printf("\nChecksum was not recieved with allotted time frame.");
 exit(1);
}

// Receive Logical Download
_bios_timeofday(0,&st);
err=EOF;
while ((ct<24)&&(err!=0xA5))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}
if (err==EOF)
{
 printf("\nFailure Retrieving Logical Download Packet.");
 printf("\nResponse from CPU was not received within allotted time frame.");
 exit(1);
}
if (err != 0xA5)
{
 printf("\nFailure Retrieving Logical Download Packet. No return command received.");
 printf("\nChecksum was not recieved with allotted time frame.");
 exit(1);
}
_bios_timeofday(0,&st);
err=EOF;
while ((ct<24)&&(err!=0x41))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}
if (err==EOF)
{
 printf("\nFailure Retrieving Logical Download Packet. Length not found");
 printf("\nResponse from CPU was not received within allotted time frame.");
 exit(1);
}
if (err != 0x41)
{
 printf("\nFailure Retrieving Logical Download Packet. Invalid Packet Length.");
 printf("\nChecksum was not recieved with allotted time frame.");
 exit(1);
}

// Get Enclosure Number
_bios_timeofday(0,&st);
err=EOF;
while ((ct<24)&&(err!=1))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}
if (err==EOF)
{
 printf("\nFailure Retrieving Logical Download Packet. Expecting Enclosure Number.");
 printf("\nResponse from CPU was not received within allotted time frame.");
 exit(1);
}
if (err != 1)
```

```
{
 printf("\nFailure Retrieving Logical Download Packet. Bad Enclosure Number.");
 printf("\nChecksum was not recieved with allotted time frame.");
 exit(1);
}

// Get 32 of the first 64 Outputs
for (j = 0; j<32; j++)
{
 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<24)&&(err==EOF))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
 }
 if (err==EOF)
 {
  printf("\nFailure Retrieving Logical Download Packet. Expecting Output Number.");
  printf("\nResponse from CPU was not received within allotted time frame.");
  exit(1);
 }
 return_outputs[j] = (int) err;
}

flush_buffer();     // Dont need the rest of the packet structure(s).
```

This section describes how to create APS commands and how the matrix responds to them. The following sections are more detailed descriptions of each APS command.

5.3.2.1 Request Enclosure Errors-#A0

Use this command to request a list of all configuration errors from a specific enclosure. In the command syntax tables below, the error list is being requested from enclosure number two. The enclosure number and checksum values differ depending on the enclosure number.

Command	Length	Enclosure No.	Checksum
#A0	#01	#02	#A3

5.3.2.2 Answer Enclosures Errors-#A1

This command contains the error list requested by the #A0 command. #A1 data section always has an 8 byte data word identifying error types and the modules where they reside. To interpret the error list, refer to section 3.3.2, “System (Configuration) Errors.”

After each command syntax table (below) is an interpretation of the configuration error word.

Command	Length	Configuration Errors	Checksum
#A1	#08	#01 #00 #00 #06 #00 #00 #00 #FF	#AF

In the configuration error word above, the matrix and command modules of enclosure one are having difficulty interpreting the grouping table.

Command	**Length**	**Configuration Errors**	**Checksum**
#A1	#08	#01 #00 #00 #04 #00 #00 #0B #FE	#67

In this configuration error word, the matrix module of enclosure one is having difficulty interpreting the logical status table and the grouping table.

5.3.2.3 Request Literal Dump-#A2

Use this command to request a list of the literal signal routes (literal I/O configuration) in a specific enclosure. In the syntax tables below, the literal configuration list is being requested from enclosure two. The enclosure number and checksum values vary depending on the enclosure number.

Command	Length	Enclosure No.	Checksum
#A2	#01	#02	#A5

5.3.2.4 Answer Literal Dump-#A3

This command contains the literal configuration list requested by the #A2 command. The literal status contains 32 input numbers; the first input listed is connected to output one, the second input listed is connected to output two, etc. all the way up to output 32. The command and length values are constant, but the enclosure number, literal status, and checksum values are variable.

After each syntax table (below) is an explanation of the literal status in decimal.

Command	Length	Enclosure No.	Literal Status	Checksum
#A3	#21	#03	#0C #0B #09 #08 ... #18	–

The literal status for enclosure three *in decimal* is:

output 1 - input 12

output 2 - input 11

output 3 - input 9

output 4 - input 8

Command	Length	Enclosure No.	Literal Status	Checksum
#A3	#21	#06	#18 #1A #0D #02 #03 #13 #09 #10	#3A

The literal status of enclosure six in decimal is:

output 1 - input 24

output 2 - input 26

output 3 - input 13

output 4 - input 2

output 5 - input 3

output 6 - input 19

output 7 - input 9

output 8 - input 16

5.3.2.5 Request Logical Dump-#A4

Use this command to request a list of the logical signal routes (logical I/O configuration) in a specific enclosure. Even though the logical status should be the same in all enclosures, it is necessary to specify an enclosure in this command (specify the enclosure that is serially linked to the external controller). For more information about the logical I/O configuration, refer to section 2.1, “Change Key.” In the syntax table below, the logical configuration list is being requested from enclosure two.

Command	Length	Enclosure No.	Checksum
#A4	#01	#02	#A7

5.3.2.6 Answer Logical Dump-#A5

This command contains the logical configuration list requested by the #A4 command. Because the 4Y supports up to 128 logical inputs and outputs, this command is sent as two command packets. The first packet includes outputs 1-64, the second packet includes outputs 65-128. The logical status segment of this command lists logical input numbers; the first input listed is connected to the first logical output, the second input is connected to the second output, etc. all the way up to 64 outputs in the first packet. The second packet can be distinguished from the first because the most significant bit of the enclosure number byte is set high. In this command the command and length values are constant, but the enclosure number, logical status, and checksum values are variable.

The following syntax tables show the two packets returned for enclosure three after a #A4 command.

Command	Length	Enclosure No.	Logical Status	
#A5	#41	#03	#04 #09 #07 #20 #08 ... #67	–

Command	Length	Enclosure No.	Logical Status	Checksum
#A5	#41	#83	#0D #0C #09 #21 #20 ... #08	–

5.3.2.7 Publish Input Status-#A6

This command contains a list of the outputs a specific input is routed to. #A6 is the result of a BCS input status request while the serial filter in the matrix is turned off. This command is either literal or logical, depending on your request. #A6 is used to simplify parsing of returned values. The command section always contains the value #A6. The length, input status, and checksum values are variable.

After each syntax table (below) is an explanation of the input status in decimal.

Command	Length	Input Status	Checksum
#A6	#03	#02 #04 #1A	#C9

The input is routed to outputs 3, 5, and 27.

Command	Length	Input Status	Checksum
#A6	#05	#10 #20 #0C #05 #12	#FE

The input is routed to outputs 17, 33, 13, 6, and 19.

5.3.2.8 Publish Output Status-#A7

This command contains the input routed to a specific output. #A7 is the result of a BCS output status request while the serial filter in the matrix is turned off. This command is either literal or logical, depending on your request. #A7 is used to simplify parsing of returned values. The command and length sections always contain the same values. The output status and checksum values are variable.

After each syntax table (below) is an explanation of the output status in decimal.

Command	Length	Output Status	
#A7	#01	#19	#C1

The input routed to the specified output is 26.

Command	Length	Output Status	Checksum
#A7	#01	#F0	#67

The input routed to the specified output is 16.

5.3.2.9 Command String-#AF

Use this command to verify that the correct BCS command(s) is being received by the matrix CPU. The BCS command(s) section of this command can hold up to 64 BCS commands. In the following examples, we have converted all ASCII values to hexadecimal. Each hexadecimal number is one byte. The command and modifier sections always contain the same values. The length, BCS command(s), and checksum values are variable. The length is the number of BCS commands plus one for the modifier byte.

Note: This command places the BCS command into the matrix where it is treated as a control panel entry. This command echoes on the control panel regardless of the echo bit setting (see section 5.2, “Echoing Serial Commands (Control Panel Echo)”).

After each syntax table (below) is an explanation of the BCS command(s) sent.

Command	Length	Modifier Byte	BCS Command(s)	Checksum
#AF	#09	#00	#43 #4C #31 #49 #31 #4F #31 #54	#C6

The BCS command is CL1I1O1T (change level 1 input 1 to output 1).

Command	Length	Modifier Byte	BCS Command(s)	Checksum
#AF	#11	#00	#43#4C#20#49#32#4F#33#54#53#4C#20#49#32#54	#4E

Note: #43 is ASCII code for ‘C’, #4C is ASCII code for ‘L’, #31 is ASCII code for ‘1’, #49 is ASCII code for ‘I’, #4F is ASCII code for ‘O’, #54 is ASCII code for ‘T’.

The BCS commands are CL0I2O4TSL0I2T (change level 0 input 2 to output 4 and check the status of level 0 input 2).

5.3.2.10 Publish Text String-#B0

Use this command to display a text string on all control panel LCDs in your distribution matrix. Enter the text string as the hexadecimal equivalent of its ASCII values. Certain hexadecimal command bytes can be included with the text string for better control over the LCD(s). Up to 64 characters (bytes) can be included in the text section of this command. The command section always contains the value #B0. The length, text, and checksum sections are variable. The following table lists the APS commands that can be part of the Text section of the #B0 command.

Command	Explanation
#C8	Clear screen
#E0	Save cursor location to internal memory
#E1	Restore cursor to location saved in internal memory
#E7	Turn cursor on
#E8	Turn cursor off
#EE	position cursor at start of first line
#EF	position cursor at start of second line
#EC	position cursor at start of third line
#ED	position cursor at start of fourth line

After each syntax table (below) is an explanation of the text area in English.

Command	Length	Text	Checksum
#B0	#05	#C8 #44 #4F #4E #45	#03

#C8 clears the LCD(s) for the message: DONE.

Command	Length	Text	Checksum
#B0	#07	#C8#EF#45#52#52#4F#52	#F8

#C8 clears the LCD(s), #EF places the cursor at the start of the second line for printing the message: ERROR.

5.3.2.11 Background Switch-#B1

Note: This command has been replaced by #B8. Please use #B8 in all new applications.

#B1 sends commands to be executed by the CPU without affecting on-going control panel operations. Background switches are the preferred method of executing commands from an external controller. The commands used in the operation section may include logical and literal changes, and executing and defining presets. Each operation must be

entered as a hexadecimal byte. The command section always contains the value #B1. The length, operation, and checksum values are variable. The operation section can contain up to 64 bytes of commands and the values are base zero.

After each syntax table (below) is an explanation of the commands sent to the matrix CPU.

Command	Length	Operation	Checksum
#B1	#02	#10 #05	#C8

This command string sets preset 6 as the current I/O configuration.

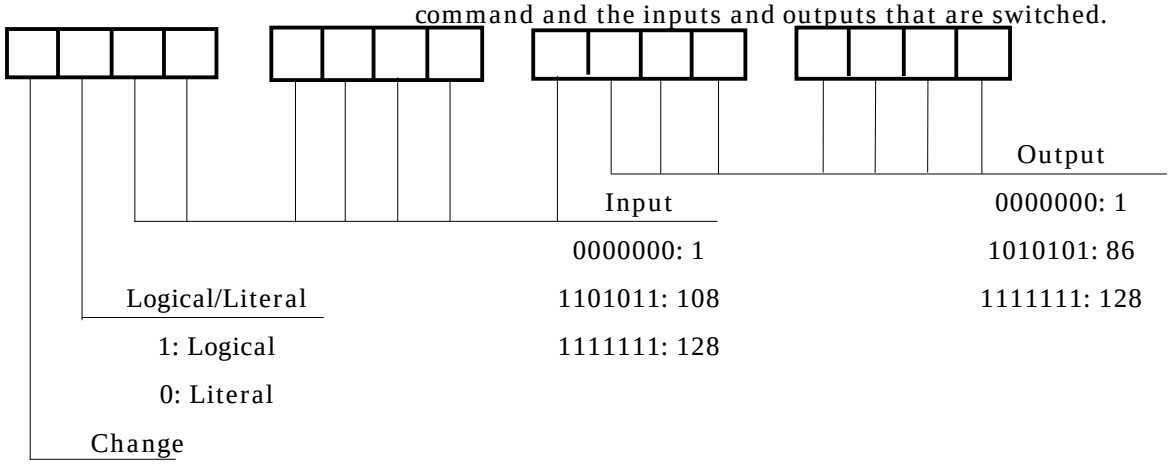
Command	Length	Operation	Checksum
#B1	#04	#C9#18#10#09	#AF

This command string makes the logical switch of input 20 to output 9 and then sets preset 10 as the current I/O configuration.

The following sections give a binary breakdown of the hexadecimal numbers for all background switch commands.

Logical and Literal Switches

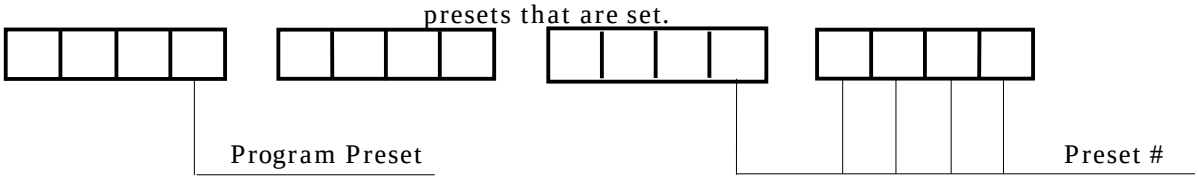
A logical switch routes a logical input to a logical output. A literal switch routes a literal input to a literal output. Set the Change and Logical bits as indicated below. Define the Input and Output signals using the indicated bits. There are seven bits for each signal, which allow you to create input and output numbers up to 128. Below the command format is a table that gives some examples of the logical change command and the inputs and outputs that are switched.



Binary	Hex	Effect
1100 1111 0111 0000	CF70	Switches input 31 to output 118
1110 1001 1001 0011	E994	Switches input 84 to output 20
1100 0011 1000 0011	C383	Switches input 8 to output 4

Program Preset

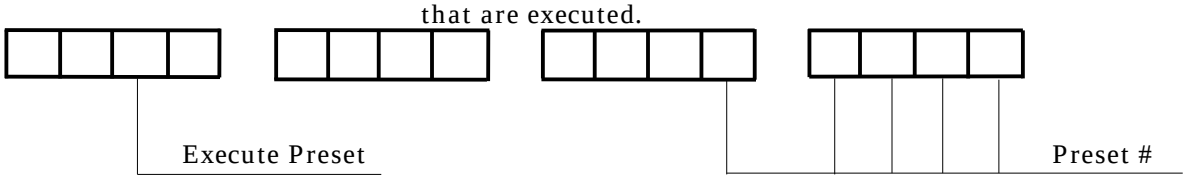
A background switch can define a preset using the current input/output configuration. When the command is sent, the current I/O configuration is defined as the specified preset number. If there is an existing preset with the same number as the one specified in the packet, the current preset is overwritten by the new preset. Below the command format is a table that gives some examples of the commands and the presets that are set.



Binary	Hex	Effect
0001 0000 0001 1001	1019	Sets preset #26
0001 0000 0000 1111	100F	Sets preset #16
0001 0000 0001 0011	1013	Sets preset #20

Execute Preset

A background switch can execute a preset when the CPU receives the packet. Use the indicated bits to specify the preset to be executed. Below the command format is a table that gives some examples of the commands and the presets that are executed.



Binary	Hex	Effect
0010 0000 0001 1011	101B	Executes preset #28
0010 0000 0000 1111	100F	Executes preset #16
0010 0000 0000 0011	1003	Executes preset #4

5.3.2.12 Request Enclosures Online-#B4

Use this command to verify the enclosures that are linked to and recognized by the master enclosure. Since only one enclosure can be queried, specify the master enclosure (#01) in the enclosure number section of this command.

The following syntax table requests the list of enclosures from the master enclosure.

Command	Length	Enclosure No.	Checksum
#B4	#01	#01	#B6

5.3.2.13 Answer Enclosures Online-#B5

This command contains a list of the enclosures linked to and recognized by the master enclosure, it also identifies any enclosures experiencing errors. The master enclosure lists itself in the enclosure number section of this command. Any enclosure that is experiencing an error is listed with its most significant bit set high. The command sections always contains the value #B5. The length, enclosure number, and checksum values are variable.

The syntax tables (below) are followed by an explanation of the enclosure number values.

Command	Length	Enclosure No.	Checksum
#B5	#04	#01 #82 #03 #04	#43

Enclosures 1, 2, 3, and 4 are linked and recognized, but enclosure 2 is experiencing an error.

Command	Length	Enclosure No.	
#B5	#05	#01 #02 #03 #04 #05	#C9

Enclosures 1, 2, 3, 4, and 5 are linked and recognized and none of them are experiencing an error.

5.3.2.14 Request Version-#B6

Use this command to request the 4Y version number from a specific enclosure's CPU. Enclosures with different CPU versions should not be linked together. You can request the version number from any enclosure in the distribution matrix, one enclosure at a time. The command and length values are always the same. The enclosure number and checksum values are variable.

After each syntax table (on the next page) is an explanation of the enclosure number section values.

Command	Length	Enclosure No.	
#B6	#01	#02	#B9

Enclosure 2 is being queried.

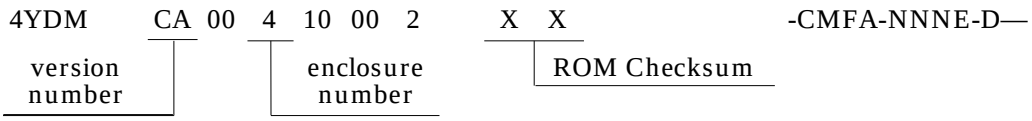
Command	Length	Enclosure No.	
#B6	#01	#08	#BF

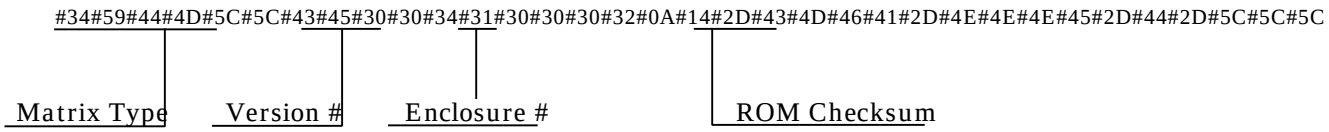
Enclosure 8 is being queried.

5.3.2.15 Answer Version-#B7

This command contains the CPU version, ROM checksum, and a list of currently installed software modules for the enclosure specified in the #B6 command. The command and length sections of this command always contain the same values. The CPU Version and checksum values are variable. The CPU Version section of the command will look similar to this:

The software modules are all listed by a code. The following tables list the software module and what code identifies them within the CPU Version section of the #B7 command. The second table displays the entire command in its ASCII hexadecimal value.





After the syntax table (below) is an explanation of the CPU Version section value.

Command	Length	CPU Version	Checksum
#B7	#22	4YDM CD 4 10 00 2 0C 14 -CMFA-NNNE-D-	

Enclosure 4 is a 4YDM with CPU version CD with the following modules installed:

- ❑ Command
- ❑ Matrix
- ❑ Front
- ❑ Async
- ❑ Link port
- ❑ Link port
- ❑ Link port
- ❑ Event
- ❑ Dry Contacts

5.3.2.16 Background Switch Commands-#B8

Use #B8 to send operation commands to be executed by the 4Y CPU without affecting on-going control panel operations. Background switches are the preferred method of executing commands from an external controller. The commands used in the operation(s) section may include logical and literal switches and executing and defining presets. Each operations must be entered as three sequential hexadecimal bytes. The first byte defines the operation and enclosure. The second byte defines the affected input and the third byte defines the affected output. The command section always contains the value #B8. The length, operation(s), and checksum values are variable. The operation section can contain up to 64 bytes of commands. When including inputs and outputs in the operation section, remember that an input, output, or preset specified as 0 is actually input, output, or preset 1 (base zero).

After each syntax table (below) is an explanation of the command(s) sent to the CPU.

Command	Length	Operation(s)	Checksum
#B8	#03	#00 #02 #03	#C0

A logical switch is made from input 3 to output 4.

Command	Length	Operation(s)	Checksum
#B8	#06	#21 #0A #0B #24 #1E #1F	#55

Two literal switches are performed. The first switch is input 11 to output 12 on enclosure 1. The second switch is input 31 to output 32 on enclosure 4.

Command	Length	Operation(s)	Checksum
#B8	#03	#41 #00 #11	#0D

The current I/O configuration of enclosure 1 is stored as preset 18.

Command	Length	Operation(s)	Checksum
#B8	#03	#61 #00 #12	#2E

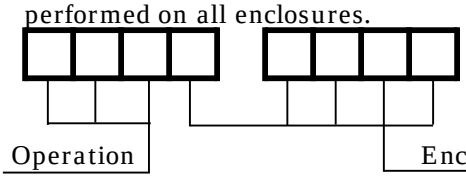
Preset 19 is restored on enclosure 1.

The following sections give a binary breakdown of the hexadecimal bytes in the operation(s) section and what parts of the operation command they apply to.

Operation and Enclosure Number (byte 1)

Byte one defines the operation to be performed and assigns the operation to an enclosure. The first three bits of this byte identify the operation. The last five bits assign the enclosure number. When performing logical switches, assign enclosure zero in this byte since the enclosure is ignored. The following diagram explains the bits of byte one.

Note: When performing a literal or any preset operation, if the enclosure is not specified (enclosure zero is specified instead), then the operation requested is performed on all enclosures.



000: Logical Switch	00001: enclosure 1
001: Literal Switch	00010: enclosure 2
010: Program Preset	00011: enclosure 3
011: Execute Preset	11111: enclosure 31

So if you want to perform a literal switch on enclosure three, the following would be the correct first byte.

binary: 0010 0011

hexadecimal: #23

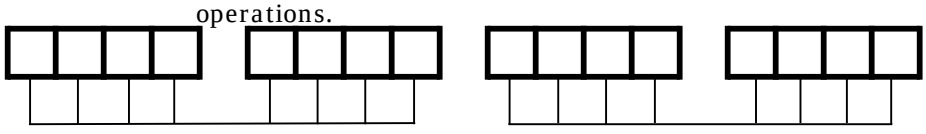
decimal: 35

Operation Specific Data (bytes 2 and 3)

Byte two defines the affected input during switches (logical and literal). Byte three defines the affected output during switches and during preset operations it identifies the affected preset. Both bytes are base zero.

Logical and Literal Switches

Byte two is the input port. Byte three is the output port. Both values are base zero. The following diagram explains the binary breakdown of bytes one and two during switch operations.



0000 0000: input 1	0000 0000: output 1
0000 0001: input 2	0000 0001: output 2
1111 1111: input 256	1111 1111: output 256

So if you want to execute a switch that routes input 12 to output 25, the following would be the second and third bytes.

binary: 0000 1011 0001 1000

hexadecimal: #0B #18

decimal: 12 25

Preset Operations

During preset operations Byte two must be set to #00. Byte three is the value of the Preset. You cannot set a preset on one enclosure and then apply it to another. The value of Byte three is base zero.

So if you want to set the current configuration of one of the matrices as preset 27, the following would be the third byte.

binary: 0001 1010

hexadecimal: #1A

decimal: 26

5.3.2.17 APS Software Handshaking

After receiving an APS command, the 4Y CPU sends the checksum back to the external controller. The external controller verifies the returned checksum by comparing it with the calculated checksum. If they are the same, the command has not been contaminated. The CPU will ignore the command if the checksum is different.

5.4 Software

Control software allows a PC to control a distribution matrix via the serial port(s). AutoPatch offers several software packages that fall into three categories:

Manual Execution–implements commands as they are entered

Automatic Execution–implements commands at the times they are scheduled to be executed

Configuration File Manipulation–allows you to edit your matrix's configuration files

This section discusses the following AutoPatch software:

- 4YRoute (manual execution)
- YRoute (manual execution)
- WinRoute (manual execution)
- AutoRoute Event Scheduler (automatic execution)
- ScanPatch (automatic execution)
- YTOOLS for DOS (configuration file manipulation)
- YTOOLS for Windows (configuration file manipulation and manual execution)

5.4.1 4YRoute

4YRoute is a DOS software program created by AutoPatch for changing the I/O configuration from a PC. 4YRoute allows you to simulate all I/O configuration operations except check the status of an input or output. It is suggested you familiarize yourself with the operation from the control panel before using 4YRoute. See chapter 2, “Configuration the Inputs and the Outputs.”

4YRoute can only be run on a PC that uses RS-232 communications. After starting the program, set the serial port that the commands are sent through by using 4YRoute’s com porT feature; see the com porT section on page 5-25. If the serial communications cable is not properly connected, when you enter a command at the Option prompt, the message, “Communications Error,” appears on the PC screen.

Running 4YRoute:

1. Connect the matrix to a PC using RS-232 communications.
2. Using any copy feature, copy 4YRoute from the DOS directory that contains 4YRoute, type 4YROUTE, and press the Enter or Return key on the PC keyboard. The following screen appears.

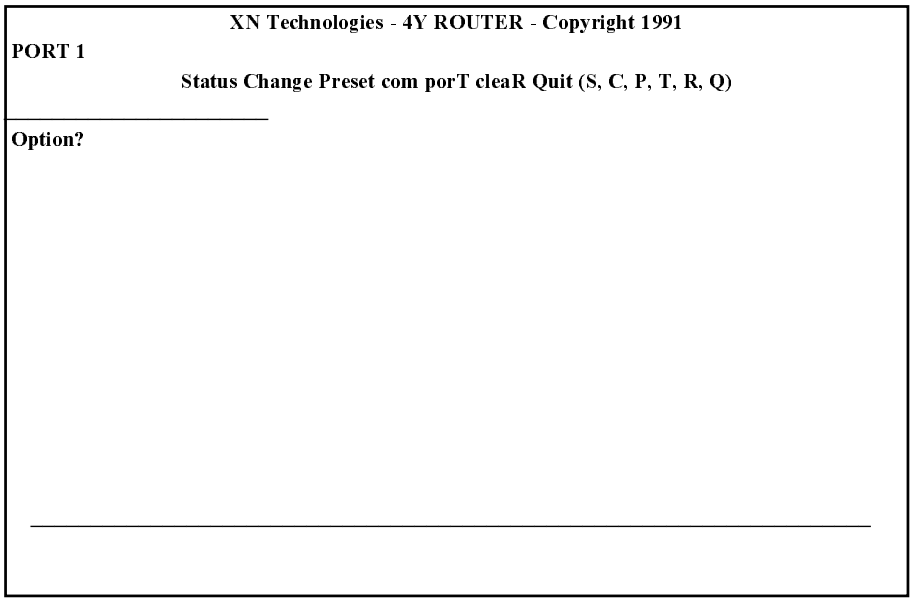


figure 5.9 4YROUTE screen

5.4.1.1 4YRoute Commands

At the top of the 4YRoute screen are the commands that may be entered at the Option? prompt. The capitalized letter in each of the commands is the key that activates that command. Three of the commands (Change, Preset, and com porT) are interactive. When using any of these commands, 4YRoute prompts you for the next logical piece of information in creating a complete command. For example, if you pressed <C> to begin a Change, 4YRoute would prompt you for: the Level in which the change would be executed, the Input, and then the output. To enter the command, press either <.> to specify it as a literal command or <K> to specify it as a logical command; the command executes and the Option? prompt returns. For more information on literal commands, see section 7.1, “Literal Operations.” Use 4YRoute exactly as you would enter commands from the control panel. After entering a complete command, the Options? prompt returns.

Status

4YRoute does not support serial status at this time. Pressing <S> returns the message, “Serial Status not Supported.”

Change

Enter a change using 4YRoute just as you would from the control panel. Press <C> to open the Change menu and enter the numbers for the logical input and output(s). To enter the command by pressing either <.> (literal change) or <K> (logical change). 4YRoute will not display an error message if an incorrect or out of range value is entered; instead it rejects the value and prompts for a different value.

Preset

4YRoute can execute and define presets. When <P> is pressed, 4YRoute prompts you to either define or execute a preset. If you press <X>, it prompts you for the number of the preset you wish to execute. If you press <D>, it prompts you for a number that will define the current I/O configuration.

com porT

The com porT option allows you to choose the serial port that 4YRoute uses to communicate with the matrix. Press <T>; 4YRoute prompts you to define whether you wish to use serial port 1, serial port 2, or both. Enter either 1,2, or B for both.

cleaR

The cleaR options serves the same function as <Cancel> on the control panel. Press <R> to cancel any incomplete commands; the Option prompt returns.

Quit

Press <Q> to exit 4YRoute; you will return to the DOS prompt.

5.4.2 YRoute

YRoute is a DOS-based software package that simulates all configuration operations, except checking the status of an input or output. Operations are implemented by entering the commands (from the top of the screen) at the Option? prompt. Before using YRoute, familiarize yourself with the operation of your distribution matrix from the master enclosure Local X/Y Control Panel. See Chapter 2, “Configuring the Inputs and the Outputs.” YRoute can only be run on a PC that uses RS-232 communications.

Note: Using YRoute requires an RS-232 cable with the handshaking pins shorted. See figures 5.1-5.5 in section 5.1, “Attaching a Serial Controller.”

5.4.3 WinRoute

WinRoute is a Windows based software package with a customizable graphical interface. Each input and output is represented by an icon, and connections are represented by lines between inputs and outputs. The icons can be placed in any pattern you desire inside the WinRoute window. For example, the icons can be placed to mimic the physical locations of devices around your office.

Note: WinRoute can only be run on a PC that uses RS-232 communications and runs Windows 3.1 or higher, including Windows 95.

5.4.4 AutoRoute Event Scheduler

AutoRoute Event Scheduler is a DOS-based control program that schedules events for matrix processing. Command lines must be entered in the string file, which is scheduled in the event file. The event file is processed by AutoRoute Event Scheduler. Processing the event file allows the software to changed the distribution matrix configuration according to user defined parameters, including command and event timing. AutoRoute Event Scheduler can handle an event file containing up to 100 events and a string file containing up to 100 strings. The strings can recur in intervals of one second to one year.

Note: AutoRoute Event Scheduler can only be run on a PC that uses RS-232 communications; see figures 5.1-5.5 in section 5.1, “Attaching a Serial Controller.”

5.4.5 ScanPatch

ScanPatch is a DOS-based control program that cycles I/O configurations. Each output has a set of assigned inputs that are cycled through it. Each ScanPatch session can handle up to 128 outputs, with a set of inputs for each output. Each set can contain no more than 40 inputs. The length of a ScanPatch session can be pre-defined or you can let it run until you need or want to terminate it.

Note: ScanPatch can only be run on a PC that uses RS-232 communications; see figures 5.1-5.5 in section 5.1, "Attaching a Serial Controller."

5.4.6 YTOOLS for DOS

YTOOLS for DOS is a DOS-based program that allows you to upload and manipulate configuration files for your 4Y. YTOOLS can be run on a PC with DOS version 3.3 or higher.

5.4.7 YTOOLS for Windows

YTOOLS for Windows is the Windows version of YTOOLS for DOS with a few added features. YTOOLS for Windows uses a graphical user interface and includes a text editor, a terminal emulation package, and an icon tool bar. YTOOLS for Windows can be run on a 386 or higher that is running Windows version 3.1 or higher. It requires at least 2 megabytes of available disk space.

5.5 Dry Contacts

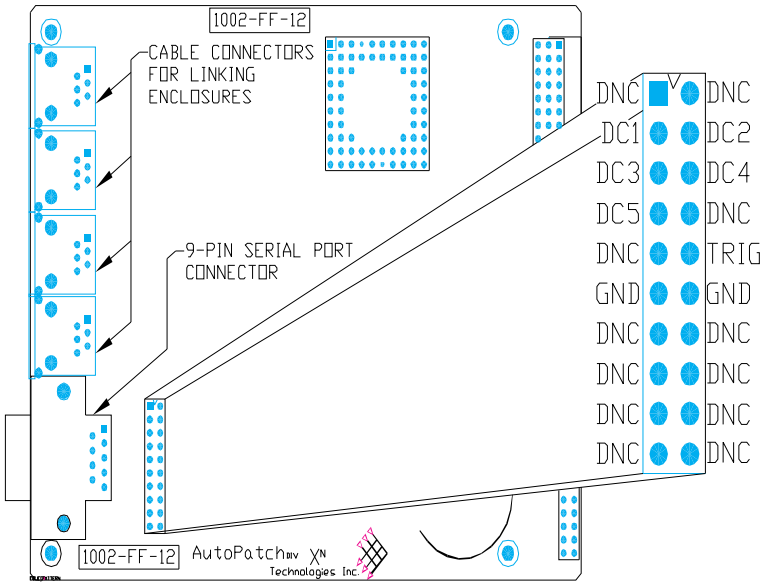
The 4YDM supports dry contacts which can be used to execute any defined preset. The dry contact should be made to the external control header on the CPU board (see figure 5.10).

DC lines provide the binary pattern for any desired preset number from 1 to 32. The CPU pulls up the pin settings. Define the desired preset by shorting the DC1 to DC5 pins to ground in a binary format. The table below shows examples of the pin settings and the presets that would be executed. Notice the pin settings in the table below go from DC1 to DC5, where DC1 is one and DC5 is 16.

DC1	DC2	DC3	DC4	DC5	Preset
0	0	0	0	0	1
1	0	0	0	0	2
0	0	1	1	0	13
1	1	1	0	1	24
1	1	1	1	1	32

0 = ground 1 = pulled up

All DC lines are pulled to +5V; activate any DC line by shorting the GND on the connector. Activate the TRiG line by shorting to the GND on the connector. Data must remain in the trigger .01 seconds and the trigger must be held for .001 seconds.



DNC = Do not connect; DC = Dry contact

figure 5.10 Pinout for a dry contact

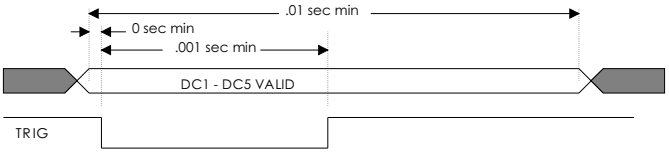


figure 5.11 TRiG line for a dry contact

Chapter 6 - Configuration Files

A configuration file is a text file that is used to upload information to the matrices CPU. Configuration files were originally designed (under the C-4 version of the 4YDM) only to upload groupings, but now configuration files may include five information sections: groupings, presets, packets, strings, and cables. The following table briefly describes each information.

Note: AutoPatch strongly suggests you make a backup of your configuration disk in the event the original disk gets damaged.

Groupings	a data table that contains the physical signal definitions for the inputs or outputs used in the matrix. Each grouping defines the signal(s) that are switched when a logical input or output is switched (see Grouping Table definition in the *Getting Started* manual).
Presets	a data table containing an image of the I/O configuration for an enclosure. Each preset included in the configuration file specifies every input and output signal connection for an enclosure.
Cables	a data table which describes the wiring scheme used to connect multiple enclosures.
Packets	are Advanced Packet Structure commands that are executed as the configuration file is read. Packets will allow access to future commands as they become available.
Strings	are Basic Control Structure (BCS) commands that are executed as the configuration file is read.

This chapter explains how to create each of the information sections and how to upload a configuration file. To create a new configuration file, open a file in a text editor that can save a file in text only (.txt) format and follow the instructions in sections 6.1 - 6.7. Section 6.8 explains how to upload a configuration file.

The configuration file is important because it is the only way that you can add or change groupings. But equally important is the fact that you can send commands to the matrix to set its features and make switches to the I/O configuration. Keeping your configuration file up to date allows you to restore the matrix to your desired method of operation in case of accidental memory loss on the matrix CPU.

6.1 The Configuration File Sent with the Matrix

Before the matrix is shipped, a configuration file is uploaded to the CPU. This configuration file contains relevant sections, including groupings and packet sections. The groupings section contains the basic groupings specified when the matrix was ordered. The packet section contains a few APS commands which prepare the matrix to accept the first switch described in the Getting Started section. The floppy disk sent with the matrix contains a copy of the configuration file and an executable file, YTOOLS.EXE and/or YTOOLS for Windows, which is used to upload the configuration file. The steps for using YTOOLS.EXE or YTOOLS for Windows to upload a configuration file are in section 6.8, “Uploading a Configuration File.”

The configuration file is included on the floppy disk so that if the groupings file is accidentally lost because of an error or power surge, you can upload the groupings file again and continue with normal operations. Another reason the configuration file is included on floppy disk is that having a disk copy of the configuration file makes it easier to update or create a new configuration file. As you will learn later in this chapter, the configuration file can be used to set the way that the matrix operates. It is advised that you always have a backup copy of the configuration file on floppy disk.

6.2 Updating the Configuration File

The configuration file is used to upload and store pertinent information in the CPU. Once stored, the information remains in the CPU unless it is overwritten by a newer section of a configuration file. A configuration file can be uploaded with one to five sections. When a new configuration file is uploaded, each section of the new configuration file overwrites the existing section in the CPU. If the new configuration file does not contain all five sections, the sections missing from the new file are not changed in the CPU. For example, if a new configuration file contains a groupings and packet section, when the new configuration file is uploaded only the groupings and packet sections in the CPU are overwritten. The strings presets, and cables sections remain the same.

It is recommended that you maintain a configuration file containing the relevant sections for programming the matrix to work as desired. This is a safeguard against an unexpected power surge or another unforeseen event which could cause the CPU to loose memory. In such a case, all that is required to restore the matrix to proper working order is to load the configuration file. For more information on uploading a configuration file, see section 6.8, “Uploading a Configuration File.”

To use the configuration file for maintaining the desired matrix operating procedure, create a strings section which uses BCS commands to set the configuration words. For more information on updating the performance of the matrix via the configuration file, see section 5.3.1, “BCS” and section 6.7, “Strings.” Additions or changes to a configuration file can be made in any text editor that can save a file in text only (.txt) format.

To create a new configuration file, open a new file in a text editor and follow the instructions in sections 6.3 - 6.5. Section 6.8 explains how to upload a configuration file.

6.3 Groupings

Groupings define the physical, or literal, signals used when switching a logical input or output. The 4YDM is shipped with logical groupings for the inputs and outputs already defined and uploaded to the CPU. The following table is an example of a groupings section, or table, in a configuration file.

```
[grouping]
Input 01=(01:01) ; composite video from camera in Room #26
Input 02=(02:01m 02:02m 02:03m 02:04w) ; RGBS from Room #26
Input 03=(01:02) ; composite video from camera in Room #27
Input 04=(02:05m 02:06m 02:07m 02:08w) ; RGBS from Room #27
Output 01=(01:01) ; video monitor at front desk
Output 02=(02:01 02:02 02:03 02:04) ; RGBS monitor at front desk - sync
Output 03=(01:02) ; video monitor at front desk
Output 04=(02:05 02:06 02:07 02:08) ; RGBS monitor at front desk - sync
```

The text enclosed in brackets on the first line of the file is the section header; the section header must be the first line in the section. Input and output groupings can be in any order as long as they begin on the line below the section header.

The groupings section of a configuration file can contain up to 128 input and 128 output groupings. Each input and output grouping can contain up to sixteen literal signals. When defining the groupings, list signals by enclosure number and signal number; separate the enclosure and signal numbers with a colon. The enclosure number is the number of the enclosure that houses that physical signal; the signal number is the input or output number of that signal on the rear of its enclosure. Comments, which help identify the groupings, may be included after a grouping. Comments must begin with a semicolon and end with a hard return or line break. If a comment cannot fit on the same line as its grouping, place the comment by itself on a separate line.

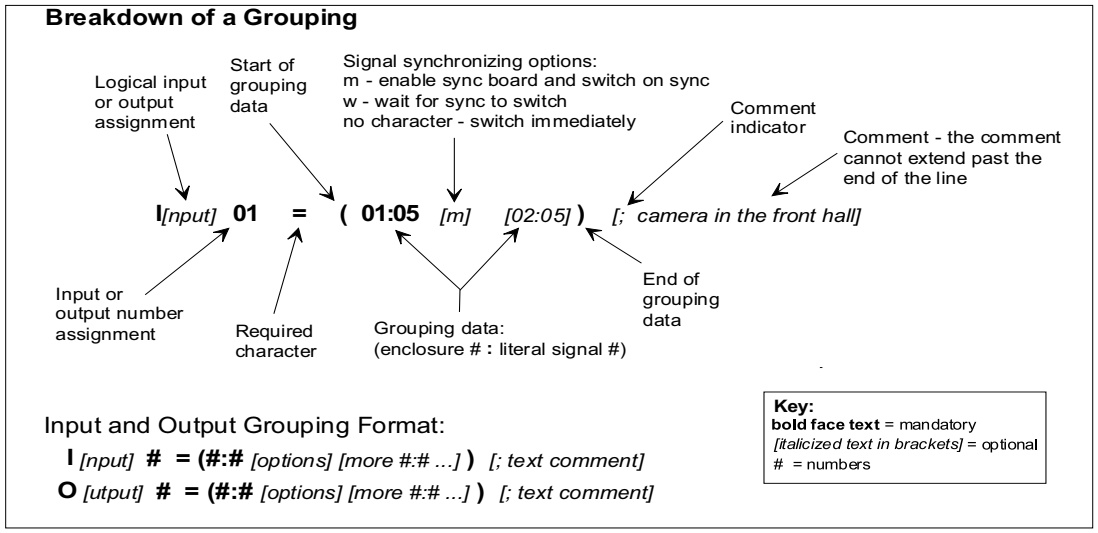


figure 6.1 Input and Output groupings format

When creating groupings, your logical input definitions should contain the same number, order, and kind of signals as the specific logical output definitions they can be routed to. For example, when the R, G, B, and s signals of a video input are defined as a logical input, the logical output receiving that logical input must be defined with the same signal sequence (R, G, B, s).

Editing a Grouping Section:

Note: When a groupings section is uploaded, the previous groupings section is overwritten. Unless you wish to create the entire groupings section, always edit a disk copy of the existing groupings section.

1. Make a copy of the configuration file that was sent with the matrix or the most current configuration file.
2. Open the copy of the configuration file in a text editor that can save text in text only (.txt) format.
3. Find the groupings section in the file.
4. To add input or output groupings, type the new groupings on any new line in the groupings section. Either the current groupings in the groupings section or the groupings in figure 6.1 may be used as examples. If more than one input or output grouping has the same grouping number, YTOOLS displays a warning and then overwrites the previously defined grouping(s) when the configuration file is uploaded.
5. To remove groupings, delete the entire grouping and any comments.
6. After completing all editing, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the program will search for a file name with a .cfg extension.

Creating a Groupings Section for a New Configuration File:

1. To create a new groupings section, open a file in a text editor that can save in text only (.txt) format.
2. In brackets, type “grouping” and enter a line break.

   ```
   [grouping]
   ```
3. Following the example in figure 6.1, create all input and output groupings.
4. After creating the desired groupings, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the program will search for a file name with a .cfg extension.

Creating an Input Grouping for Vertical Interval Switching:

If the matrix contains a vertical interval sync board you can synchronize signals in a grouping, such as RGBs, to switch during the next vertical retrace interval. To make this occur, specify when the signal should switch by setting the option for that signal (see figure 6.2). There are three options that can be attached to the signals in a grouping: m, w, or no option. If the matrix does not have a sync board, all signals must have no option; no option is indicated by a space after the signal. The ‘w’ option tells the signal to queue up and wait for the sync signal from the sync board to execute. Like the ‘w’ option, the ‘m’ option tells the signal to queue up and wait for the sync; however, ‘m’ also sends an enable signal to the sync board to send a sync signal. After the sync board receives the enable signal, the sync board strobes the ***system*** at the next vertical sync pulse and all signals that have been queued up are executed. Only input groupings can be synchronized, output groupings cannot contain sync signal options.

1. Follow the first three instructions in the Editing a Grouping Section.
2. To add input or output groupings, type the new groupings on any new line in the groupings section. When editing the grouping, insert the desired option immediately following the signal. Insert at least one space between the option and the next signal. (See the example grouping in figure 6.2)

Note: If more than one input or output grouping has the same grouping number, YTOOLS displays a warning and then overwrites the previously defined grouping(s) when the configuration file is uploaded.

3. After completing all editing, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing (uploading) the file. If you do not specify an extension when executing a file, the compiler will search for a file name with a .cfg extension.

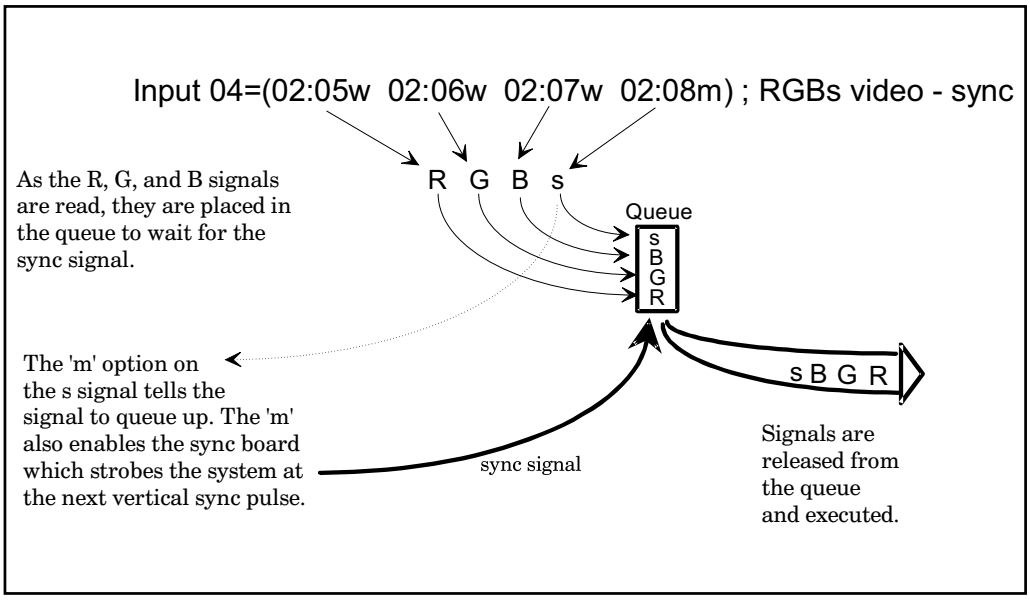


figure 6.2 signals queue up and are released on sync

Creating Output Groupings for Multiple Enclosure Systems:

Output grouping definitions for multiple enclosure systems with more than 32 inputs require an asterisk (*) on the end of the definition. The asterisk notifies the master enclosure's CPU that the signal will go through several stages to complete a path. The asterisk is not for input groupings, it is only needed for output groupings. Figure 6.3 shows an example of an output grouping definition for a multiple enclosure system with more than 32 inputs.

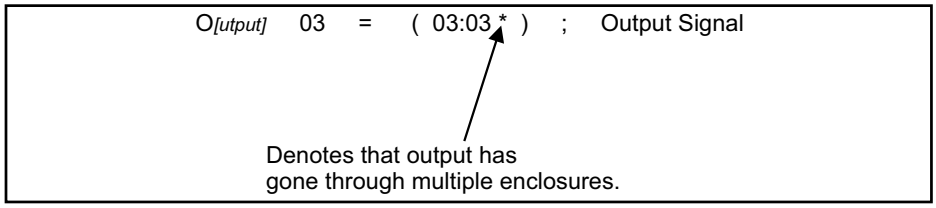


figure 6.3 Output grouping for a multiple enclosure system

6.4 Presets

A preset is a time-saving feature for quickly recalling an I/O configuration (see section 2.2, “Preset”). Presets can be loaded from a configuration file, or they can be defined and executed from the control panel or an external controller. Both types of presets store the same I/O configuration information. When you execute a preset, the configuration information stored in the preset is implemented.

To create presets in the preset section of a configuration file, type the preset and number assignment and then the I/O configuration. When the configuration file is uploaded, the preset section overwrites any existing presets in the CPU.

Defining presets in a packet section of a configuration file is most advantageous for matrices that have one enclosure, or matrices that switch single-signal inputs and outputs or audio follow video (AFV) inputs and outputs. Each matrix CPU can store up to 32 presets.

The table below shows an example of some presets that can be uploaded in a configuration file.

Note: All inputs and outputs are literal. Presets cannot be defined with logical inputs and outputs.

In this example, presets 01, 02, and 04 route the same input to every output in the matrix. In Preset 03, inputs are routed to specified outputs. The numbers specified within the parenthesis are the inputs; the position of each number indicates the output to which that input is routed. For example, in preset 03, input 10 is routed to output 1, input 9 is routed to output 2, input 8 is routed to output 3, ..., input 5 is routed to outputs 10, 15, and 28.

```
[preset]
Preset 01=(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)
Preset 02=(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)
Preset 03=(10 9 8 7 6 1 2 3 4 5 1 2 3 4 5 20 19 18 17 16 30 31 32 1 2 3 4 5)
Preset 04=(5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5)
```

Editing Presets in a Configuration File:

Note: When a preset section is uploaded, the previous preset section is overwritten. Unless you wish to create the entire preset section, always edit a disk copy of the existing preset section.

1. Make a copy of the configuration file that was sent with the matrix, or the most current configuration file.
2. Open the copy of the configuration file in a text editor that can save text in text only (.txt) format.
3. Find the preset section in the file.
4. To add presets, type the new presets on a new line in the preset section. If more than one preset has the same preset number, when the configuration file is uploaded, YTOOLS displays a warning and then overwrites the previously defined preset(s).
5. To remove presets, delete the entire preset and any comments.
6. After completing all editing, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the program will search for a file name with a .cfg extension.

Creating a New Preset Section in a Configuration File:

1. To create a new preset section, open a configuration file in a text editor that can save in text only (.txt) format.
2. In brackets, type “preset” and enter a line break.

   ```
   [preset]
   ```
3. Following the example on the previous page, create all presets.
4. After creating the desired presets, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the program will search for a file name with a .cfg extension.

6.5 Cables

The cables section of a configuration file contains a table of all input and output cable connections. The table is referenced by the master enclosure's CPU when finding signal paths in a multiple enclosure system with more than 32 inputs. The cable table only affects multiple enclosure systems.

AutoPatch includes a cable table in the original configuration file sent with your matrices. The table can only be modified in a text editor. If you need assistance modifying your cable connections and your cable table, contact AutoPatch Customer Service at (800) 622-0246.

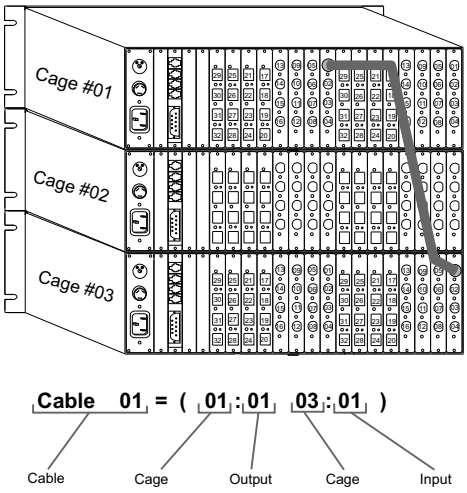


figure 6.4 Cable connecting output 1 of cage 1 to the first input of the third cage.

6.6 Packets

The packet section of a configuration file is used to send APS commands to the CPU. For information on packet commands and format, see the Advanced Packet Structure (APS) Commands section; it instructs how to place the APS commands in a configuration file.

Since a configuration file cannot receive information, only two APS commands, #B0 Publish Text String and #B1 Background Switch, are useful when sent from a configuration file. The following table is an example of APS commands in a packet section; notice each packet has a following comment that explains what the packet does.

[packet]

#B1#02#C0#84#F7 ; background switch of input 1 to output 5

#B1#02#C0#85#F8 ; background switch of input 1 to output 6

#B0#04#44#6F#6E#65#3A ; prints “Done” on the control panel screen

Editing Packets in a Configuration File:

1. Open the configuration file in a text editor that can save a file in text only (.txt) format
2. Locate the packet section of the configuration file.
3. To add packets to a configuration file, on a new line in the configuration file, type the new packets.
4. To remove packets, delete the entire packet and any comments.
5. After adding the desired groupings, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the program will search for a file name with a .cfg extension.

Creating a Packet Section in a New Configuration File:

1. To create a new packet section, open a configuration file in a text editor that can save in text only (.txt) format.
2. In brackets, type “packet” and enter a return.

   ```
   [packet]
   ```
3. Following the format given in the table on the previous page, type all desired packets.
4. After adding the desired groupings, save the file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the program will search for a file name with a .cfg extension.

6.7 Strings

Strings are commands in BCS format. A string section can contain up to 128 strings which are executed when the configuration file is uploaded. A string section can be useful for redundant testing because each time you test the 4YDM, you only have to upload the configuration file instead of typing in every command. String sections are easier to understand than packet commands.

To create a string, label the string with a number and define a complete BCS command. The CPU reads the string section and executes the commands in that section each time the configuration file is uploaded. The following commands are strings from a string section.

String	Description
string 01="CL0I1O1T"	Switch logical input 1 to logical output 1.
string 02="RR20T"	Define the current input/output configuration as preset #20.
string 03="CL0I2O23T"	Switch logical input 2 to logical output 23.
string 04="P1*00001"	Sets the Command configuration word. The Default Command screen opens to the Change screen.

For more information on BCS commands, see section 5.3.1, “BCS.”

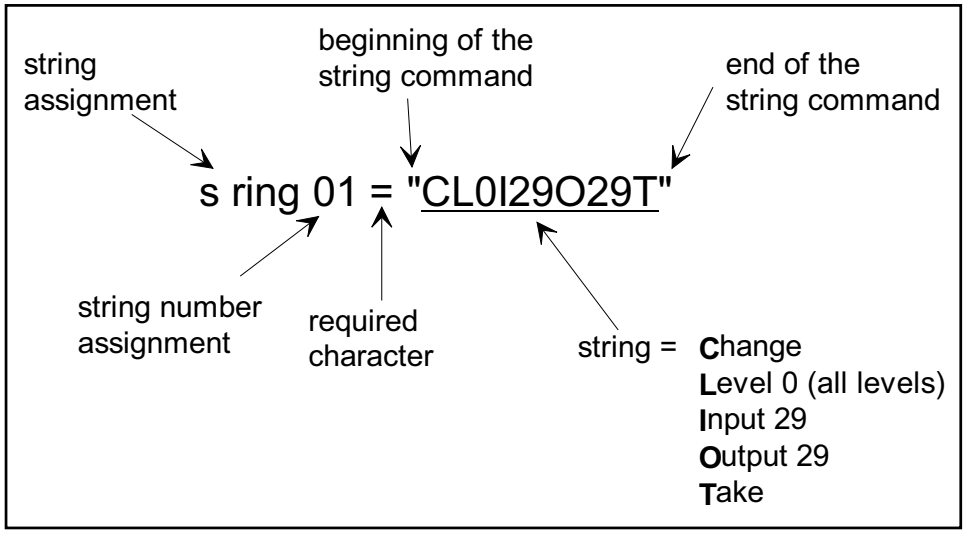


figure 6.5 format for a string command

Editing a String Section:

1. Open a configuration file in a text editor that can save a file in text only (.txt) format
2. If there is not an existing string section, create one by typing “string” in brackets. If a string section exists, go to step 3.

```
[string]
```

3. Using the format in figure 6.5, add the new strings to the string section. Each string should begin on a new line. A string section may contain up to 128 strings.
4. If you are adding strings to an existing string section of a configuration file, be sure to delete any unwanted strings.
5. After editing the string section, save the configuration file in text only (.txt) format. If you give the file name an extension, be sure to include that extension when executing the file. If you do not specify an extension when executing a file, the compiler will search for a file name with a .cfg extension.

6.8 Uploading a Configuration File

You can upload a configuration file from any IBM PC, or clone, that is serially connected to the 4YDM. AutoPatch has two software packages available for uploading configuration files, they are YTOOLS for DOS and YTOOLS for Windows. For more information on these software packages, please refer to section 5.4, “Software.”

6.8.1 Uploading Using YTOOLS for DOS

When uploading the configuration file, include four items on the command line:

- YTOOLS (the upload software)
- Name of the configuration file
- Communications port (if you are not using com1)
- Any options

The command string for uploading a configuration file must be entered in the following format. All options must be entered with square brackets and a space must be included between each component.

```
C:\YTOOLS [file name] /[com port] /[options]
```

For example, to upload the grouping and preset sections of a configuration file named sample.cfg using com port 2, you would enter the following command string at the C:\ prompt:

```
YTOOLS sample.cfg /com2 /[grouping] /[preset]
```

6.8.1.1 Configuration File

The configuration file is the information file created and saved in ASCII format. Be sure to include any file name extension when typing the configuration file name. Unless the

extension is specified, YTOOLS looks in the working directory for a configuration file with a .cfg extension.

6.8.1.2 Com Port

Com port is the communications port through which the configuration file is uploaded. The default com port is /com1. Specify the com port you want to use, if you want to use a com port other than /com1. The YTOOLS program recognizes three other com ports: /com2, /com3, and /com4. If you do not specify a com port, the CPU defaults to com port 1.

6.8.1.3 Options

Options allow you to specify the sections of the configuration file you want to upload, if you do not want to upload the entire configuration file. The default option is all sections of the configuration file. To specify section(s) of the configuration file to be uploaded, type that section's name in the options area of the command line. To specify a section, type:

```
/[section name]
```

The following table shows examples of command line directives and an explanation of each.

Command	Explanation
`ytools jan.cfg /com2 /[grouping] /[string]`	The grouping and string sections of the jan.cfg configuration file are uploaded through communications port 2.
`ytools joe.cfg /[grouping]`	The grouping section of the joe.cfg configuration file is uploaded through communications port 1.
`ytools dri.cfg /com4`	All sections of the dri.cfg configuration file are uploaded through communications port 4.
`ytools klf.cfg /com3 /[preset]`	The preset section of the klf.cfg configuration file is uploaded through communications port 3.
`ytools`	The help screen and the full options list are displayed on your monitor screen.

Uploading a Configuration File:

1. Ensure the desired configuration file is saved in ASCII format and resides in the working directory.
2. Using the following format, include the proper filename, communications port, and all necessary options.

```
C:\YTOOLS file name /com port /options
```

3. Press <Enter>. If the file that was uploaded contained all sections, the PC screen will display the following as the sections are uploaded.

   YTOOLS communicating with version CD

   [grouping]

   .....

   [cable]

   ....

   [preset]

   .....

   [string]

   .....

   [packet]

   .....

After the sections have been uploaded, the front panel of the matrix displays the following message:

```
      System
   Configured
  Successfully
File: <filename>.cfg _
```

figure 6.6 System Configured Successfully screen

6.8.2 Uploading Using YTOOLS for Windows

Within the YTOOLS for Windows software, a configuration file is referred to as a grouping file. This file type is still identified by the .cfg extension in the filename. You must open a file before you can upload it. YTOOLS allows you to have more than one file open at a time, but only the active file can be uploaded. If you are unsure how to open a .cfg file, refer to the instructions below. The following page contains instructions for uploading a .cfg file.

To open a .cfg file:

1. Choose the Open command from the File menu.

   The File Open dialog box appears (figure 6.7)

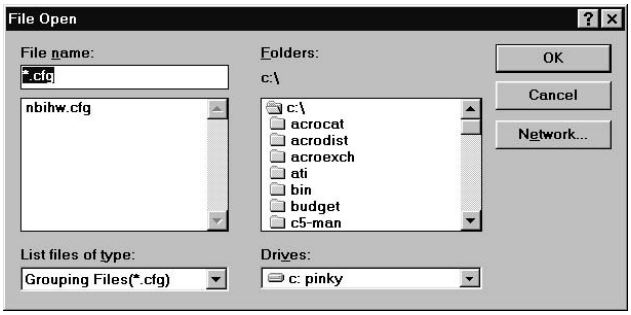


figure 6.7 File Open dialog box

2. Select the desired .cfg file from the list and click on the OK button, or use the Folders: box on the right side of the dialog box to locate the desired .cfg file and click on the OK button.

   The File Open dialog box disappears and the .cfg file is open.

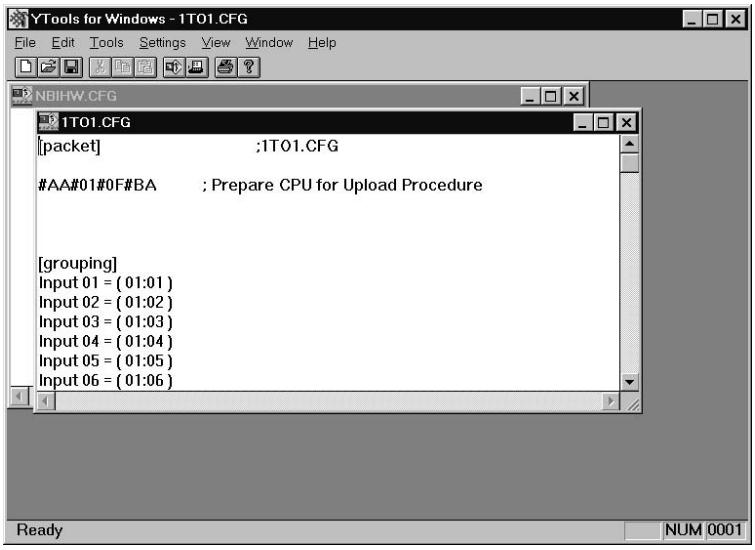


figure 6.8 open .cfg file

6.8.2.1 Uploading

Before uploading a .cfg file, make sure the communication settings are correct for your matrix. The default communication settings are: com1, 9600 BAUD, 8 data bits, no parity bit, 1 stop bit, no hardware flow control, and TTY autowrap.

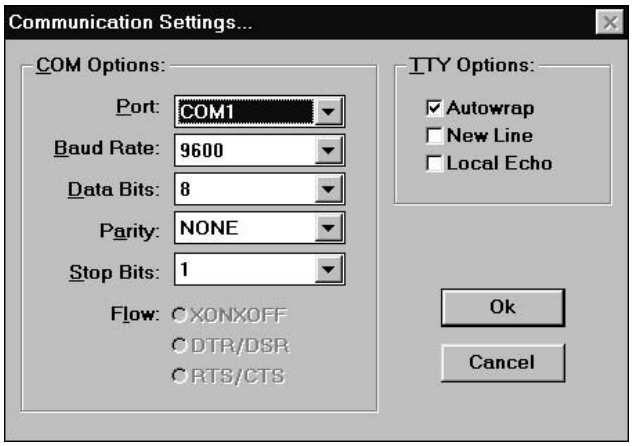


figure 6.9 Communications Settings dialog box

To upload a .cfg file:

Choose the Upload command from the File menu. A status bar appears and the active .cfg file is uploaded to your matrix. The transfer usually takes between 2 and 20 seconds.

figure 6.10 Progress bar for uploading file

Chapter 7 - Advanced Features

This chapter discusses the advanced features of the 4YDM. It is a good idea to understand the operations of the 4YDM discussed in the preceding chapters before reading about the advanced features. The features in this chapter are:

- Literal operations
- Reset Enclosures
- Refresh Logicals
- Explanations of the configuration words and their settings
- Key press shortcuts

7.1 Literal Operations

Literal operations refer to switches and status checks on individual physical signals on the rear of an enclosure. Every logical input and output is defined in the grouping table; see section 6.3, “Groupings.” Logical inputs and outputs are defined so all signals in the grouping switch when a logical switch is done. Using literal operations, it is possible to switch or check the status of any signal even if it is included as part of a logical input or output. Literal operations can be used for troubleshooting and greater control of the matrix.

When using literal operations, signals must be specified by their enclosure (level) and signal number. For example, to specify a literal input when doing a status check, enter the level number, which is the enclosure that houses the signal, and the input signal number.

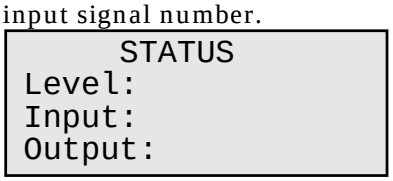


figure 7.1 Status screen

Even though the enclosure (level) is specified, what makes this operation a literal operation is the finishing action. Logical operations are entered by pressing <Take>; literal operations are entered by pressing <.>.

Warning: Literal switching invalidates the logical status table which is used to keep track of the logical inputs and outputs. You may, however, use the literal status table to check the status of the individual physical signals. After making a literal switch, it is a good idea to refresh logicals, see section 7.3, “Refresh Logicals.”

When defining a literal Change operation, the order that the input and output signals are specified does not matter. In the following example, the input signal is specified first, but the output signals could have been specified first and the operation would have the same outcome.

Making a Literal Change:

1. At the Command screen, press <Change>; the Change screen appears.

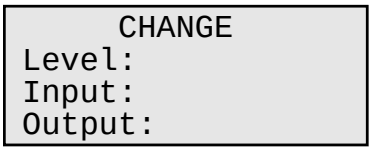


figure 7.2 Change screen

2. Press <Level> and enter the number of the enclosure in which the change occurs.

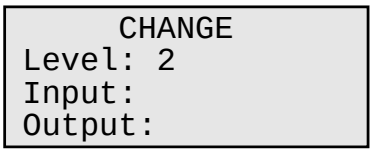


figure 7.3 Specify the enclosures

3. Press <Input>; the cursor appears after the `Input:` prompt. Enter an input signal number.

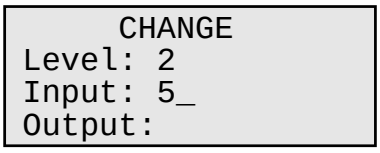


figure 7.4 Enter an input

4. Press <Output>; the cursor appears after the `Output:` prompt. Enter the output signal number that will receive the input signal. Up to 32 output signals may be specified to receive the input signal.

```
     CHANGE
Level: 2
Input: 5_
Output: 1 4 16
```

figure 7.5 Specify the output signals

5. Press <.>; the change occurs and the Command screen appears.

In this example, input signal 5 was switched to output signals 1, 4, and 16 in enclosure 2.

Checking the Literal Status:

When checking the status, the order that the input and output signals are specified does not matter. In the following example, the input signal is specified first, but the output signals could have been specified first and the operation would have the same outcome.

1. Press <Status>; the Status screen appears.

```
     STATUS
Level:
Input:
Output:
```

figure 7.6 Status screen

2. Press <Level> and enter the number of the enclosure that contains the input or output signal to be checked.

```
     STATUS
Level: 1
Input:
Output:
```

figure 7.7 Status screen

3. Press <Input>; the cursor appears after the Input prompt. Enter an input signal number.

```
     STATUS
Level: 1
Input: 3
Output:
```

figure 7.8 Status screen

4. Press <.>; all output signals that receive the specified input signal are displayed.
5. To return to the Command screen, press <Cancel>.

This status check revealed input signal 3 is routed to output signals 3, 7, 11, and 15.

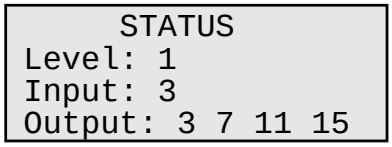


figure 7.12 Status screen

7.2 Reset Enclosures

Resetting the enclosures allows you to reboot all enclosures in the matrix. *AutoPatch* suggests you reset the enclosures any time you need to cycle the power. You can also reset one enclosure by holding down <Cancel> and pressing <Take> on that enclosure's control panel. This will only reset that enclosure. The screen on the control panel will display the startup screens.

Resetting All Enclosures:

1. At the Command screen, press <Program>; the Program menu appears.

figure 7.9 Program menu

2. Press <1>; the Command Config menu appears.

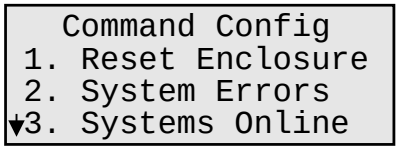


figure 7.10 Program menu

3. Press <1>; the Reset Enclosure/s screen appears.

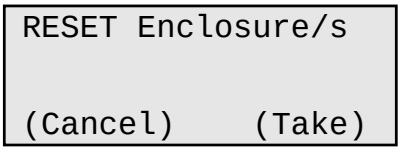


figure 7.11 Reset enclosure/s verification screen

4. Press <Take>; the matrix goes through the startup sequence: the Startup Sequence screen appears (an hourglass may appear in the upper right-hand corner of the screen)...

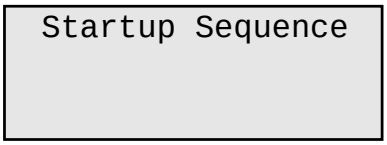


figure 7.13 Beginning of the startup sequence

then the matrix verification screen appears (the information in this screen may differ from the information that appears on your screen)...

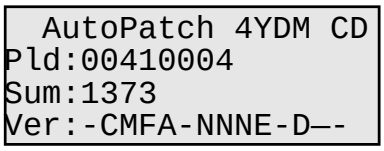


figure 7.14 Matrix verification screen

then the Command screen appears...

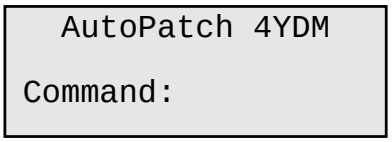


figure 7.15 Command screen

followed by the Acknowledged screen. (The information on this screen may differ from the screen you see when you reset the enclosure(s). If errors are reported, see section 3.3, “Error Reporting.”

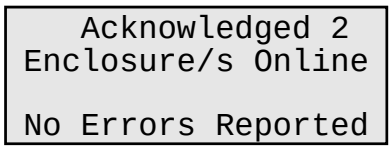


figure 7.16 Number of enclosures online; error reporting

5. Press <Cancel> to return to the Command screen. You are now ready to begin normal operations.

7.3 Refresh Logicals

Logicals refer to the logical inputs and outputs that are defined by the grouping table. Whenever a logical status is requested, the CPU checks the logical status table to find the status of the requested logical input or output. The logical status table is built as logical changes are made to the matrix. Whenever a logical switch is made, the CPU updates the logical status table. When a literal switch is made or a preset is executed (when presets are executed, they are made using literal switches), the logical status table is *not* updated; this causes the status of those signals in the logical status table to be incorrect. If the logical status of the recently switched signals were requested, the logical status would report the status of the matrix prior to the literal switches. To keep the matrix up-to-date with the logical status table, refresh the logicals after doing a literal switch or executing a preset. Refreshing the logicals changes the I/O configuration to reflect the logical status table. In effect, refreshing the logicals will "undo" any literal switch or execution of a preset.

Refreshing the Logicals:

1. At the Command screen, press <Program>; the Program screen appears.

figure 7.17 Program menu

2. Press <2>; the Matrix Config menu appears.

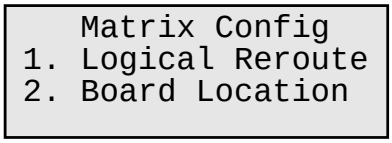


figure 7.18 Matrix config menu

3. Press <1>; the Refresh Logicals screen appears.

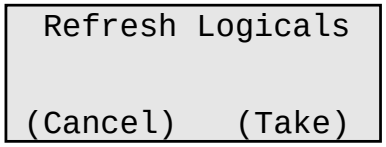


figure 7.19 Refresh logicals screen

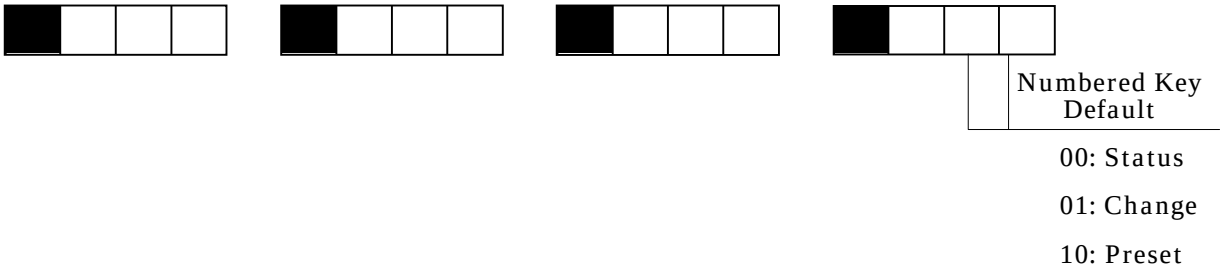
4. Press <Take>; the I/O configuration reflects the logical status table. No effects of literal operations exist in the I/O configuration. The Command screen appears.

7.4 Modules and Configuration Words

Software modules are sections of programs in the CPU that control the operation and retain the settings for the adjustable features of the 4YDM. The Program Menu lists the software modules present in the matrix; all matrices do not contain the same modules. Additional modules are available if you wish to exchange features used in the matrix. Please note additional software modules can be incorporated only if current modules are dropped. Current software modules must be dropped because there is not enough space in the ROM to incorporate additional features. The configuration word is the part of the module that retains the settings of the features. When the features of the matrix are adjusted, the configuration words change to reflect the settings. The adjustable features of the matrix can be set from the control panel or an external controller; it is also possible to adjust the features of the matrix by directly changing the configuration word. There are two advantages of setting the features of the matrix via the configuration word: it is faster than setting each feature by menu and there are features which can be set from the configuration word that do not appear on the screen menus. Each configuration word is comprised of 16 bits. The following sections show the 16 bits of each configuration word and explain how to set the user-adjustable features. Each section also has a table that contains examples of the binary settings of the configuration word, the hexadecimal equivalent of the binary settings, and the effect of the configuration word.

7.4.1 Command Module

The Command module is responsible for the screen that appears when a numbered key press begins a command. There are three options: Status screen, Change screen, or Preset screen.



Binary	Hex	Effect
0000 0000 0000 0000	0000	Status is the default
0000 0000 0000 0001	0001	Change is the default
0000 0000 0000 0010	0002	Preset is the default

7.4.2 Matrix Module

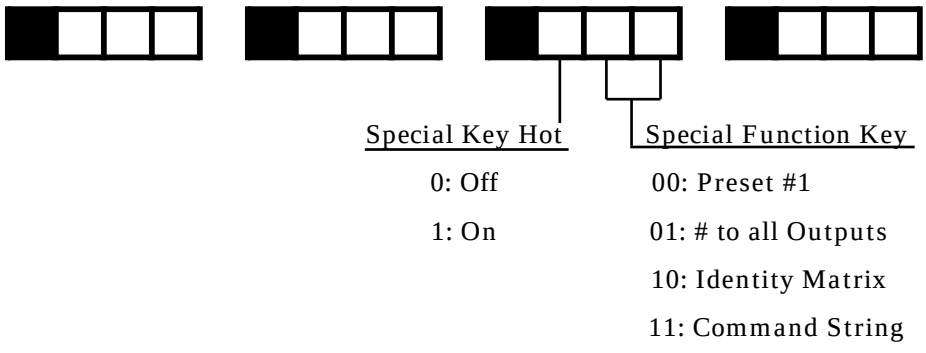
Currently the Matrix module allows or disallows literal switching and acts as a flag for sync failure (version D0 and above). Literal changes can be disallowed by setting the appropriate bit to '1'. When literal changes are disallowed, the <.> key does not function in the Change screen. Pressing <.> when literal switches are disallowed results in an operational error (see section 3.3.5, "Operational Errors"). Literal status may still be checked even if literal switches are disallowed.

If you are correcting a Sync Timeout error (version D0 and above), check the sync connector, enter '0000' as the Matrix configuration word, and restart your matrix. Sync failure errors occur during the matrix's startup sequence if the sync signal was disconnected or missing while making a logical switch before power was cycled. For instructions on fixing this error, refer to section 3.4, "Common Installation/Maintenance Errors."

Binary	Hex	Effect
0000 0000 0000 0000	0000	Literal Change Allowed
0100 0000 0000 0000	4000	Literal Change Disallowed

7.4.3 Front Module

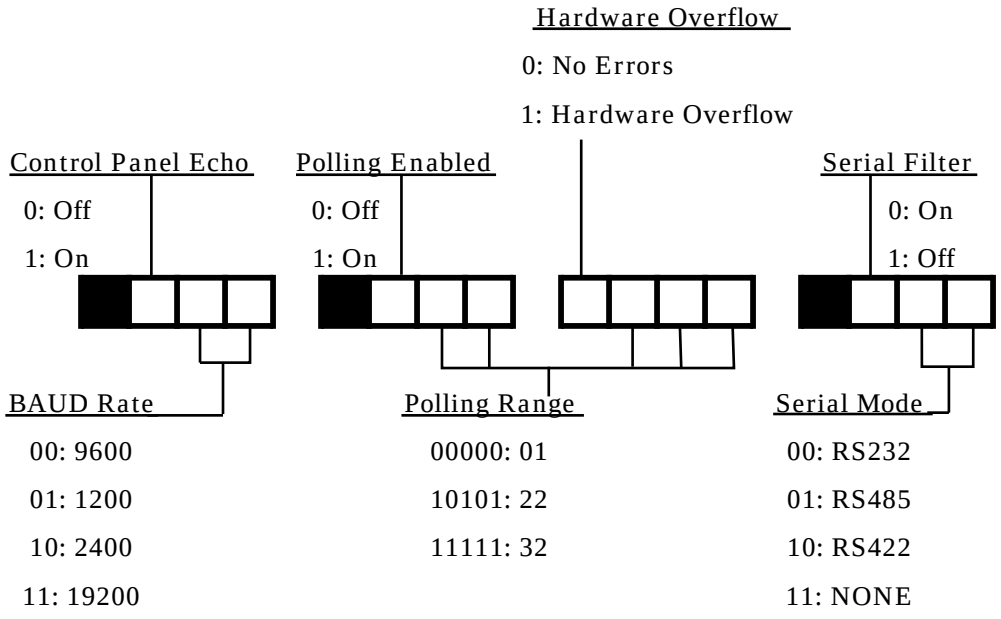
The Front module controls two features, the function of <Special> and making <Special> hot. When <Special> is hot, the function of <Special> is executed immediately. If <Special> is not hot, a verification screen appears when <Special> is pressed. For more information on the functions of <Special>, see section 2.3 "Special."



Binary	Hex	Effect
0000 0000 0111 0000	0070	<Special> is Hot, <Special> executes a command string
0000 0000 0010 0000	0020	<Special> is not Hot, <Special> executes an identity matrix
0000 0000 0100 0000	0040	<Special> is Hot, <Special> executes Preset #1

7.4.4 Async Module

The 4YDM can support one or two serial software modules. A serial module controls all adjustable features for a serial port. The adjustable features for a serial port are: BAUD Rate, polling the single bus controllers, the number of single bus controllers polled, turning the status filter on or off, and turning the control panel echo on or off.



Note: If Polling Enabled is off, the BCS commands P, R, and * do not work when entered from an external controller.

Binary	Hex	Effect
0001 0001 0111 0000	1170	BAUD Rate is 1200, Polling is off, Polling range is 16, Serial filter is on, Communication via RS 232
0011 0111 0111 0100	3774	BAUD Rate is 19200, Polling is on, Polling Range is 32, Serial filter is off, Communication via RS 232
0000 0110 0110 0100	0664	BAUD Rate is 9600, Polling is on, Polling Range is 23, Serial filter is off, Communication via RS 232

7.4.5 7.4.5 Setting a Configuration Word

1. Press <Program>; the Program menu appears.

figure 7.22 Program menu

2. Arrow down to the desired module and press <Take>; the menu for that module appears. (The following example sets the Async1 module.)

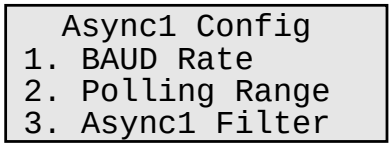


figure 7.21 Async1 Configuration

3. Once the menu for the desired module appears, press <Special>; the configuration word screen appears. The number of the enclosure from which the command was entered appears after the `Level:` prompt. The current module configuration word appears after the `Value:` prompt.

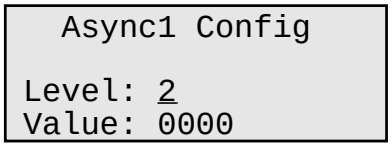


figure 7.20 Enter a configuration

4. Enter the number of the enclosure whose configuration word you wish to set. If you wish to set that configuration word for all enclosures, enter '0' at the `Level:` prompt. You must enter a number after the `Level:` prompt, even if it is the same number that currently appears. After entering a number on the `Level:` prompt, the cursor moves to the first digit after the `Value:` prompt. Enter a configuration word; the configuration word must be entered in four hexadecimal digits. See sections 7.4.1 - 7.4.4 for instructions on creating a configuration word.

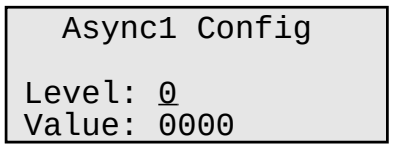


figure 7.23 Enter a configuration word

5. Press <Take> to enter the configuration word.

7.5 Key Press Shortcuts

There are a few key press shortcuts from the control panel which can make operation of the matrix faster. To quickly access any of the modules in the Program menu from the Program menu, press the number of that software module. The specified module opens and displays the module screen, if it is user-interactive. The modules numbered 10 - 16 cannot be accessed by entering the number of that module, because they require a two-digit key press. To access modules 10 - 16, use the key press listed in the table below. Please note that all of these software modules may not be populated. To access the desired module, press and hold <Program> then press the other key; let both keys up simultaneously.

Module No. #	Key Press Shortcut
10	<Program> + <Special>
11	<Program> + <Preset>
12	<Program> + <Undo>
13	<Program> + <Space>
14	<Program> + <Comma>
15	<Program> + <0>
16	<Program> + <.>

There is also a key press shortcut to reboot an enclosure. To reboot the software for an enclosure, press <Cancel> and <Take> simultaneously. The 4YDM can be set to open to either the Change, Status, or Preset execution screens when a numbered key press begins a command; see section 4.1, “Default Command Screen.”

To view the verification screen, press <Program> and then <Special>; do not press them at the same time.

AutoPatch Warranty / Returns Policy

A.1 Return Authorizations

Except for warranty claims, merchandise will not be accepted for return or exchange after the first thirty (30) days following the invoice date.

Merchandise will not be accepted for any reason without a Return Materials Authorization (RMA) number. Returned items must be shipped prepaid, insured, with the RMA number clearly stated on the outside of each carton and, if possible, in original packing container(s).

Products and parts returned or exchanged for any reason other than warranty purposes are subject to a restocking fee not greater than twenty percent (20%) of the invoiced price, if returned in unused condition.

A.2 Claims for Shipping Damages

Unless otherwise specified, merchandise is normally shipped by Federal Express Economy service but AutoPatch reserves the right to select the final method and carrier for any shipment.

Although we take special care to ensure the safe arrival of all orders, shipping accidents and damage can occur. Shipments are transferred to the appointed carrier in good condition and AutoPatch's liability for the product ceases when the transfer to the carrier is complete. Therefore, claims for damages and shortages must be filed with the transporting company by the receiving company within fifteen (15) days of receipt. Visible damage and shortages must be noted on the freight bill; packaging and contents must be retained for inspection.

A.3 Replacement Policies and Procedures

During the warranty period:

1. Describe the problem to an *AutoPatch* dealer, regional representative, or the *AutoPatch* customer service department.
2. Upon verification of a problem that requires factory repairs, an *AutoPatch* customer service representative will issue a Return Materials Authorization (RMA) number and we will, at no cost, repair or replace the part(s) returned to the factory and return the part(s) to the sending party. If conditions do not permit this

procedure, we will invoice new or reconditioned (at AutoPatch's option) replacement part(s) to the dealer and ship the part(s) to the dealer or to the consumer if so directed by written order from the dealer. Unless otherwise instructed in writing by an *AutoPatch* customer service representative, part(s) replaced under this warranty must be returned to the factory:

a) within thirty (30) days;

b) with shipping and insurance costs prepaid;

c) with the RMA number clearly indicated on the outside of each container;

d) if possible, in the original shipping container(s)

e) with a written description of problem.

If the replaced part(s) are returned within thirty (30) days, we will apply credit to the dealer's account for the total value of part(s) determined defective, plus return shipping costs. Any part(s) received after thirty (30) days or otherwise not in compliance with these requirements may be refused and credit will not be issued.

3. Repaired or replaced part(s) will be warranted for the remainder of the original system warranty period, for the first thirty (30) days following the invoice date, or we will extend the original warranty period by the period of verifiable downtime, whichever provides the greatest benefit.

Following warranty expiration:

1. Call your *AutoPatch* dealer, area representative, or the *AutoPatch* customer service department with a description of the problem.
2. Upon verification of a problem that requires factory repairs, an *AutoPatch* customer service representative will issue a Return Materials Authorization (RMA) number. We will, at nominal cost, invoice the sending party, repair or replace the part(s) returned to the factory and return those part(s) to the sending party. If conditions do not permit this procedure, we will invoice and ship new or reconditioned (at AutoPatch's option) replacement part(s) to the dealer or to the consumer if so directed by written order from the dealer.
3. Post warranty repairs and replacements are warranted for the first thirty (30) days following invoice date.

A.4 Special Notice

AutoPatch reserves the right to modify or discontinue designs, specifications, warranties, and policies without notice. All data with regard to model numbers series, specifications, and prices in our literature have been thoroughly reviewed and edited. Although we cannot assume responsibility for inadvertent omissions or errors, we sincerely apologize if misunderstandings occur and we will appreciate your criticism, corrections, and suggestions.

Appendix B - Vertical Interval Sync Expansion Board

The Vertical Interval Sync Expansion Board, or sync board, is designed to provide the *AutoPatch* Y series with a complete vertical interval synchronization switching capability. The sync board can either utilize a station master sync signal or provide a master synchronization signal by separating the sync from an input signal such as a composite video input. A sync board can also provide switching synchronization between enclosures in a distribution matrix.

Note: The sync board does not synchronize all signals that are switched. The sync board can only synchronize signals that are switched using a logical switch and only if those signals are specified in the grouping definition. This appendix explains the specifics of the sync board. For information on synchronizing signals in a grouping, see section 6.3, “Groupings.”

B.1 External Connections

Five external BNC-type connectors are provided. The BNC-type connectors align the left side of the board and are as follows:

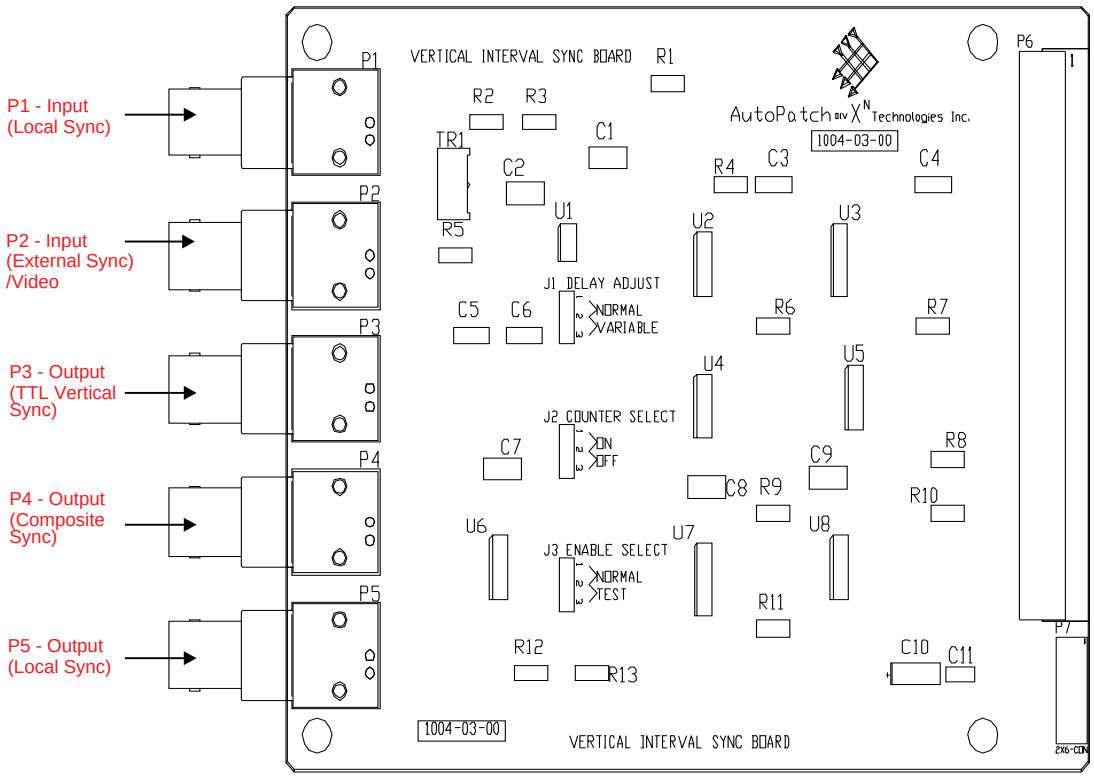


figure B.1 Vertical Interval Sync Expansion Board

P1 - Input (Local Sync)

The local sync input is intended to be connected to the local sync output from another sync board. The local sync input allows the enclosure to be switched on command from the master enclosure. The enclosure switches immediately upon reception of an active low TTL level signal and the signal is echoed out the local sync output connector. A master enclosure can sequence the switching of an entire distribution matrix by either driving in parallel all of the local sync inputs of the slave enclosures, or connecting them input to output in a daisy-chain fashion.

P2 - Input (External Sync)

The external sync input accepts any one of the following signal types:

- ❑ NTSC/PAL/SECAM/Composite video signal
- ❑ Composite sync type of synchronization signal (for example, Blackburst)
- ❑ Station master synchronization signal
- ❑ Member of an RGBS group that contains sync information

The vertical synchronization information is stripped from the signal and used to generate a local sync command which is sent to the enclosure at the appropriate time. The local sync command is also provided to the local sync output. TTL level composite sync output is also generated, depending upon the type of external input signal received.

P3 - Output (TTL Vertical Sync)

A TTL level vertical interval sync signal, derived from the external sync input, P2, is output.

P4 - Output (Composite Sync)

A TTL level composite sync signal is derived from the external sync input if possible. If the external input is a composite NTSC signal, the video portion is removed and the sync portion is output to this connector. Other external sync input signal types are handled in a similar manner, if possible. For example, a TTL vertical interval sync input signal does not contain a composite sync structure.

P5 - Output (Local Sync)

If the board is enabled, the switch sync command signal is echoed out this connector regardless of whether it came from the local sync input, the external input, or was generated internally by the processor.

B.2 On Board Jumper Settings

Three 3-position jumpers (J1, J2, and J3) are used on the board and are factory set. Pins 1 and 2 are connected for normal operation; the other positions allow adjustments to be made for very high frequencies, stand alone operation, or testing.

J1 - Delay Adjust

The standard setting (with the jumper connecting pins 1 and 2) is best for normal operation with NTSC or moderate frequency RGB signals. If the standard delay is not sufficient, instead of using a fixed resistor, move the jumper to connect pins 2 and 3 which selects the potentiometer, TR1. The delay can then be adjusted with TR1.

J2 - Counter Select

The standard selection is to connect pins 1 and 2, which enables the counter section. In this mode the counter section is used to determine the switch timing. It should be used with any signal that contains horizontal sync pulses such as NTSC, RGB with sync on Green, and similar signals. If the sync input signal does not have a horizontal sync component, then the counter will have no effect. In the event that a manually selectable time delay is desired (use J1 and TR-1 to

select and adjust the time period), the counter section can be disabled by jumpering pins 1 and 2 together.

J3 - Enable Control

The normal position, with the jumper between pins 1 and 2, allows the processor to control board enable. Placing the jumper between pins 2 and 3 disconnects the processor control line and manually enables the board. Removing the jumper completely disables the board. This jumper is used primarily for testing, but there may be some applications which require external board enable control. The board is enabled if pin 2 is high and disabled if pin 2 is low. Since pin 2 is pulled to ground with a resistor, a switch connected between pins 2 and 3 controls board enable mechanically, or TTL signal can be placed on pin 2 to control enable electronically.

B.3 Suggested Applications

The following tasks can be accomplished using the vertical interval sync board.

Note: If you are using a vertical interval sync signal and the signal is missing or somehow becomes detached, the next time you perform a switch a sync timeout error appears on the control panel screen. To continue operation, check the vertical interval sync connector and press <Cancel>. Your switches are still executed without the sync signal; however, the switch will be made without synchronization. For more information, see section B.4, “Sync Timeout.”

B.3.1 Extracting a Master Sync Signal

A system master sync signal can be extracted from any video input signal which contains vertical, or vertical and horizontal sync information. A video signal provided to the horizontal sync input (P2) will be decoded and a TTL vertical sync pulse containing horizontal sync information, such as a standard composite video signal, a composite sync signal (actually removed) will be provided on the composite sync output (P4). The video input signal is selectable if one of the video outputs is connected to the external sync input (P2). Any suitable video input can then be routed to the sync board and used to provide sync information.

B.3.2 Synchronizing a multi-enclosure switch

When the board is enabled, the next switching pulse decoded is routed to the processor board as a switch execution command. The sequence of actions by the matrix is for the master to issue a switching command to all enclosures which is to be executed upon receipt of the sync pulse. The master then enables the board, and the next sync pulse decoded by

the board commands switch execution. When enabled, the sync pulse is also provided to the local sync output. The sync pulse can then be provided to all slave enclosures so all enclosures switch at the same time. The master enclosure's local sync output can be used to drive the local sync inputs of several slave enclosures, or they may be daisy-chained by connecting the master to the local sync input (P1) of the next slave enclosure and so on.

All signals do not respond to the sync pulse. To set the signals so that they respond to the sync pulse, set the correct option for that signal in the grouping definition. For more information see section 6.3, "Groupings."

B.3.3 External Control of Board Ensemble

Jumper J1 is configured so the processor enable control is present on pin 1. Pin 2 is connected to the internal enable circuitry and is pulled to ground by a resistor. Pin 3 is tied up to +5V. Normal operation is set with a jumper connecting pins 1 and 2, which allows the processor to control the board. Leaving the jumper off disables the board. A normally open switch connected to pins 2 and 3 provides manual control of the board. Alternatively, a TTL signal connected to pin 2 provides digital control of the board from an external system. A 1 enables the board and a 0 disables it.

B.4 Sync Timeout

A sync timeout error occurs during the matrix's startup sequence if the sync signal was disconnected or missing while making a logical switch before power was cycled. Although your switches are still executed, the lack of synchronization may create a noticeable roll in video output devices. For more information see section 3.4, "Common Installation and Maintenance Errors."

Appendix C - Single Bus Controllers

A single bus controller (SBC) is a device that allows you to control an input routed to a particular output device. The output device that the SBC controls must be specified on the dipswitches on the rear of the SBC face panel. *AutoPatch* offers three models of SBCs (see figures C.1a - C.1c); the only difference between the three are the faceplate and the method of selecting an input. The CSB1 has a 3-key face plate, the CSB2 has a 12-key face plate and the CSB3 has a 16-key face plate. SBCs can be used with the 4XDM, 1YDM, and 4YDM.

Note: SBCs used with a 4YDM will only switch logical inputs.

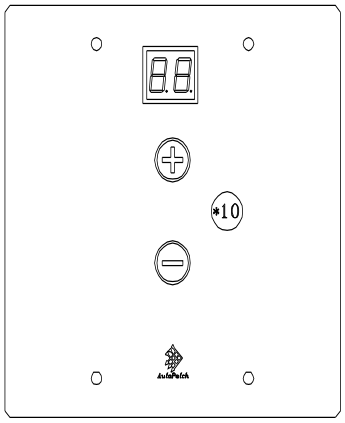


figure C.1a CSB1 model SBC

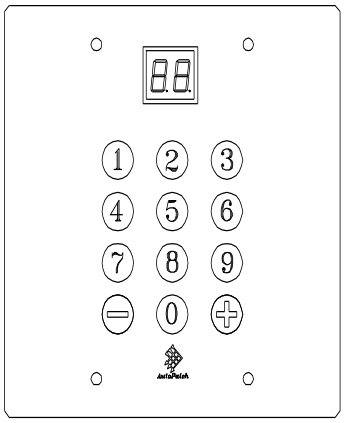


figure C.1b CSB2 model SBC

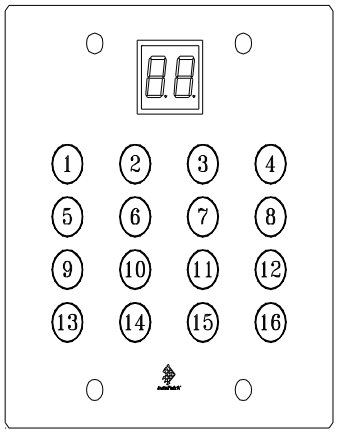


figure C.1c CSB3 model SBC

Each enclosure in a distribution matrix can handle up to 32 SBCs. The SBCs for each enclosure are daisy chained together and attached to the enclosure through a serial port. SBCs allow remote users greater control of the distribution matrix because they can switch inputs to that output device without accessing the control panel of an enclosure.

This Appendix covers:

- ❑ Installation
- ❑ CPU Assignments
- ❑ Network Communications Interface
- ❑ Power Requirements and Connections
- ❑ Mounting SBCs
- ❑ Operation of each model of SBC

C.1 Installation

Ready each enclosure in the distribution matrix that will receive a chain of SBCs:

1. Set the enclosure's polling range equal to the highest SBC identification number.
2. Set the enclosure's BAUD rate to 9600 BAUD.
3. Wire the enclosure serial port for RS-485 standards; see figure C.2.

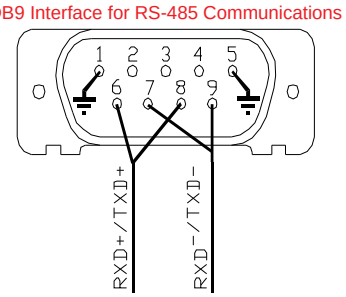


figure C.2 485 communications

Note: Figure C.2 shows an RS-485 standard wiring diagram. RXD+/TXD+ to pin 8 and RXD-/TXD- to pin 9 with 5 to ground will also work with the 4YDM.

4. Attach power. For more details, see section C.4, "Power Requirements."
5. SBCs are designed to be installed in two-gang electrical boxes. Be sure the SBC face plate properly fits the (4"x4"x2+") two-gang electrical box. The blue circles for the screw holes on the face plate should match up with the tabs in the two-gang electrical box.
6. Be sure each SBC has a distinct identification number. Each output device can have only one SBC that controls its input. See section C.2, "CPU Assignments," for more information on assigning SBC identification numbers.

C.2 CPU Assignments

Each SBC must have its own identification number. Set the identification number on the CPU Assignment Dipswitch; it is located on the rear of the SBC control panel (see figure C.3). The distribution matrix CPU continually monitors, or polls, all SBCs for change in the input channel selection.

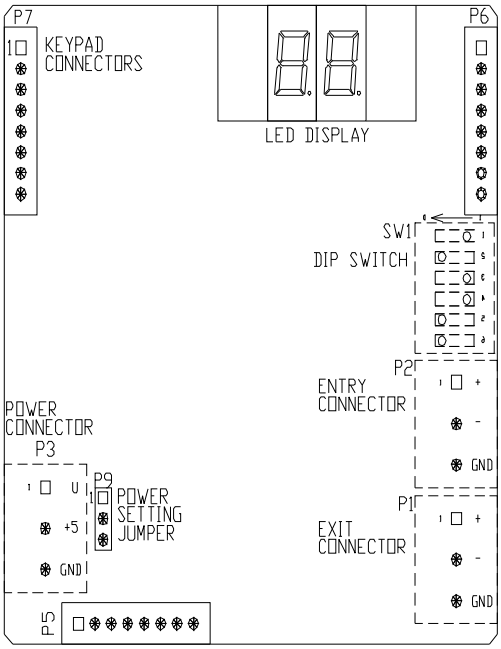


figure C.3 Rear view of SBC control panel

The CPU Assignment Dipswitch is a series of six binary switches. These switches allow you to set the identification number anywhere from 1-32. If you set the dipswitch to 1, that SBC's identity is 1; if you set the dipswitch to 32, that SBC's identity is 32.

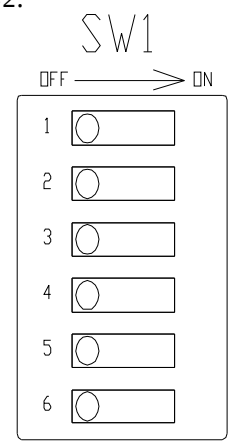


figure C.4 Binary Dipswitch

The SBC identification number is the total of all the binary switches. In the examples in figure C.5, the three controllers' identification numbers are set to 29, 13, and 9. When the distribution matrix is powered up, each SBC displays its identification number on the LED indicator. Double check the LED indicator against the setting on the dipswitch.

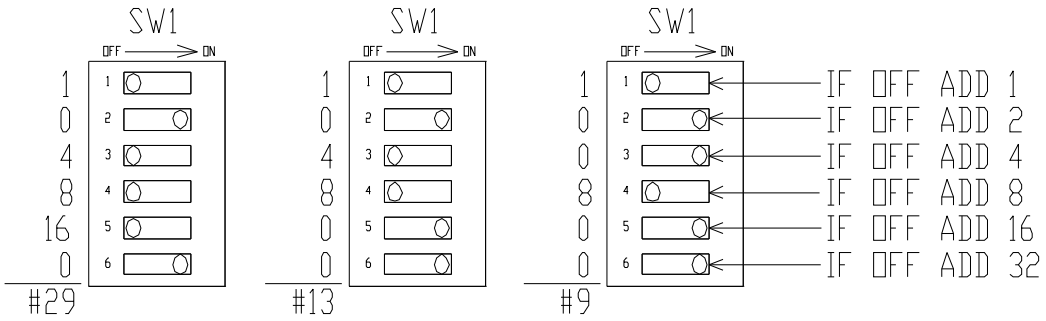


figure C.5 Binary dipswitch settings for #29, #13, and #9

C.3 Network Communications Interface

All SBCs that are going to be used with the same enclosure must be linked together in a daisy chain fashion. Use figure C.6 as a guide to link all SBCs, except the final SBC, in the chain.

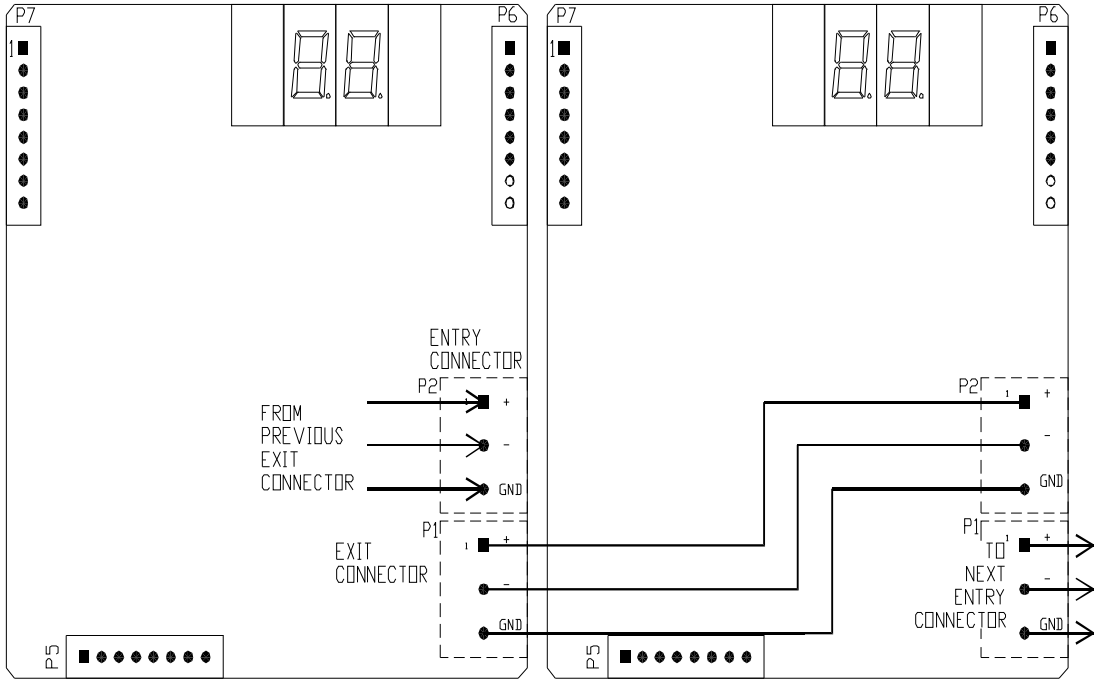


figure C.6 First and all intermediate connectors

The final SBC in the chain should be terminated with a 120 ohm resistor as shown in Figure C.7. In some cases, if the chain of channel selectors is small enough, a terminating resistor is not required and will actually disrupt communications.

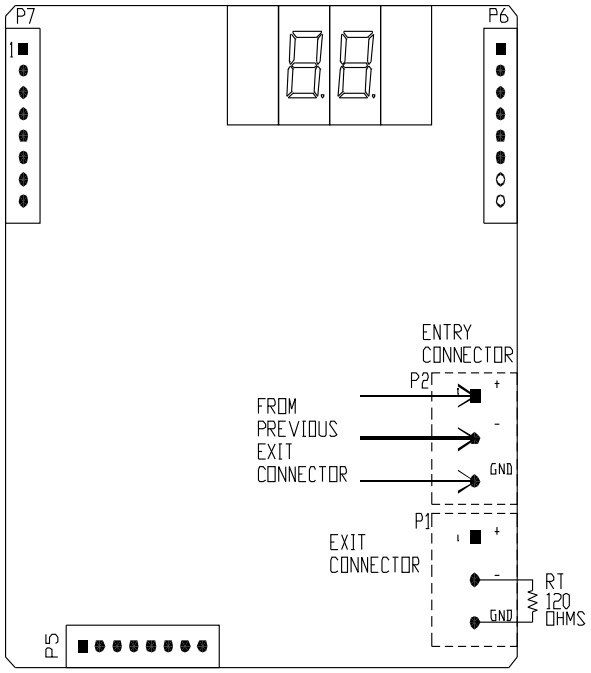


figure C.7 Final SBC in the chain

The order of the SBC identification numbers in the daisy chain does not matter. However, the first SBC in the chain must be linked to the enclosure's serial port via DB-9 interface.

Note: Recommended for network installations out to 300 feet: low capacitance cable for EIA RS 485 standard; 2 pair twisted 24 gauge/stranded conductors (7x32) e.g.: Belden #1419A=NEC: CM PCC FT 1.

C.4 Power Requirements and Connections

The recommended power for SBCs is 7-12v unregulated DC or 5v regulated DC. At the factory, the power regulation jumper is set to accept 7-12v DC. To connect unregulated DC power to an SBC, unscrew the top and bottom screws on the connector and insert the wires as shown in figure C.8. To use 5v DC, switch the power regulation jumper before attaching power to the SBC; see figure C.9. Once the jumper has been set for regulated power, unscrew the middle and bottom screws and insert the wires.

Each SBC requires 160 milliamps of power. A 9v unregulated power transformer is offered as an option.

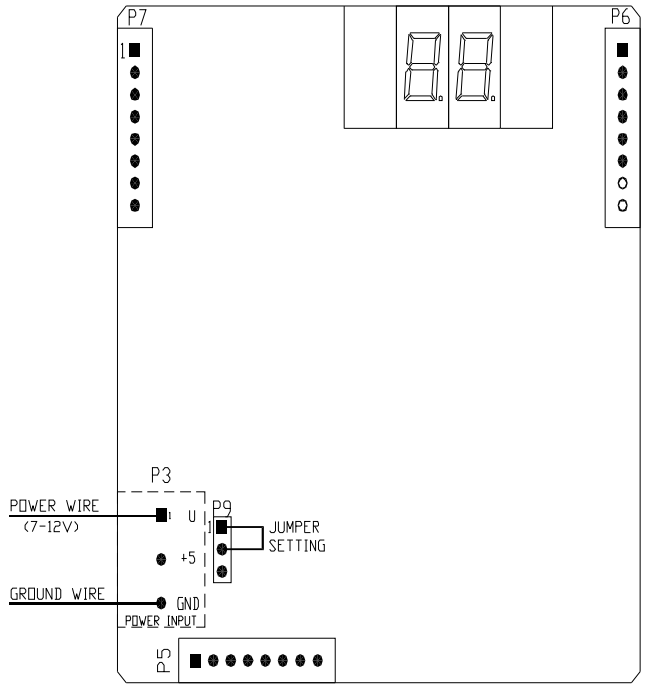


figure C.8 Jumper setting for unregulated power

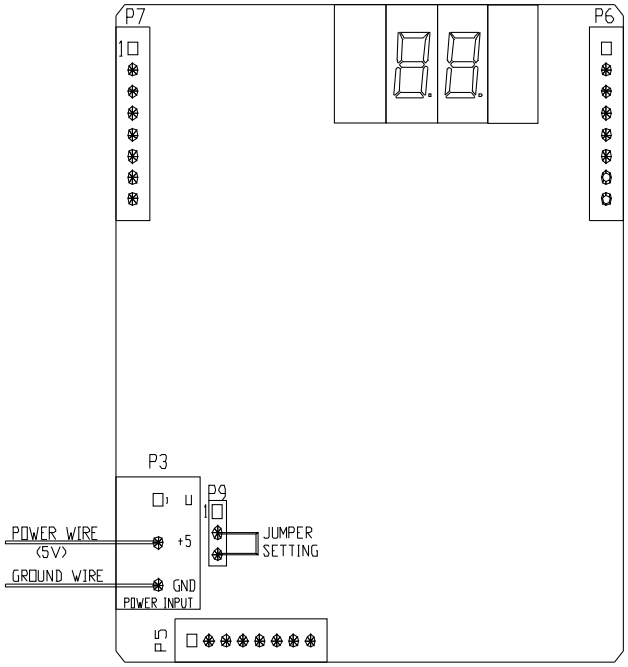


figure C.9 Jumper setting for Regulated power

C.5 Mounting the SBC

SBCs are designed to fit into a container the size of a (4"x4"x2+") two-gang electrical box. When mounting the SBC in a gang box, it may be necessary to tilt the SBC at a slight angle to get the circuit board past the tabs.

Four mounting holes, which are indicated by blue circles, are provided in every SBC control panel. The surface is not tapped unless specified when ordered. Bezels are supplied with all tapped units and can be ordered for other units.

C.6 Operation of the CSB1

After linking the SBCs to the enclosure, apply power to the distribution matrix. An LED test of all the segments in the display runs. Then the LED on every SBC briefly displays its identification number. To change the input signal, press either <+>, <->, or <*10>. Pressing <+> or <-> increments or decrements the input selection by one. Holding down <+> or <-> sequentially increments or decrements the input selection. To increment or decrement the input selection quickly, hold down <*10> while pressing either <+> or <->. This moves the input selection in intervals of 10.

Note: The input signal on the output device does not change until the keys are released.

The input number selection is circularly sequential. If the input selection is incremented above the maximum existing input number, the input selection returns to 1, and if the input number is reduced below 1, the input number proceeds to the maximum existing input number.

C.7 Operation of the CSB2

After linking the SBCs to the enclosure, apply power to the distribution matrix. The LED on every SBC briefly displays its identification number. There are two ways to change the input signal on a CSB2 model SBC: sequentially and direct access.

C.7.1 Sequentially

Use <+> or <-> to increment or decrement the input selection by one; hold down <+> or <-> to continuously increment or decrement the input selection.

The input selection is circularly sequential. If the input selection is incremented past the maximum existing input number, the input selection returns to 1. If the input number is reduced below 1, the input number returns to the maximum existing input number.

Note: The input signal on the output device does not change until the keys are released.

C.7.2 Direct Access

To directly access an input, enter the number using <1> - <9>. All entries must be double digit; enter '0' before any inputs lower than 10. The input signal on the output device changes upon entry of the second digit. Single digit entries and any entry larger than the highest available input number are ignored. To cancel an incorrect single digit entry, press <+> or <->; the input selection returns to the previous selection.

C.7.3 Quick View

CSB2 can toggle back and forth between two channels. To use quick view:

1. Enter an input number.
2. Enter a second input number.
3. Press <0>,<0> to go back to the first input.
4. Press <0>,<0> to return to the second input.

Quick view works only for the two most recent inputs.

C.8 Operation of the CSB3

The faceplate of the CSB3 model SBC has keys for inputs 1-16. This model is for use in a system that does not have more than 16 input sources. To choose an input, press a key on the SBC faceplate. The requested input number appears in the LCD display and that input is routed to the output device.

C.9 Cleaning the SBCs

SBCs require only topical cleaning. Do not use abrasive cleaners to clean the face plate; use a dry cloth and a mild glass cleaner to clean the surface of the SBC.

Appendix D - Specifications

D.1 Documentation and Reliability Information

All equipment will be delivered with one document set consisting of an Installation Manual and a Reference Manual. Additional copies of any manuals may be ordered at nominal cost per copy. Since AutoPatch provides board level repair support, schematics and logic diagrams are not normally provided, however, they may be provided under special circumstances following receipt of a non-disclosure agreement.

D.1.1 Reliability

The expected mean time between failures (MTBF) will exceed 200,000 hours for the entire system. MTBF for each Line Replaceable Unit (LRU) will exceed 350,000 hours.

D.1.2 Maintainability

Due to the modularity of the system, any subassembly can be replaced in 15 minutes or less. Failures can be diagnosed in 20 minutes or less by a competent technician with standard test equipment. The worst case time-to-repair is 45 minutes and the expected mean time to repair (MTTR) is 30 minutes.

D.1.3 Useful Life

It is the policy of *AutoPatch* to provide full support of each product for ten years after the end of production.

D.1.4 Environment

All equipment is designed to operate over a temperature range of 0°- 43° Centigrade (32° - 110° Fahrenheit) in relative humidity conditions from 0 - 90%, non-condensing, and at altitudes up to 10,000 feet.

Dissimilar metal contact is avoided as much as possible. All ferrous metal components are anodized and/or painted to minimize corrosion.

All equipment meets or exceeds the requirements of FCC class A computing devices requirements in accordance with FCC OST 62.

D.1.5 UL Listing

All 4YDM equipment is tested to UL-1419, the standard for Professional Video and Audio Equipment by ETL Laboratories as presented on page 620 of the Directory of ETL listed products, dated July 1993.

D.2 Power

115v or 220v	.7 A maximum
30 - 81 watts (depending on configuration of the matrix)	total dissipation

D.3 Audio

This appendix contains a partial listing of the specifications for the 4YDM input and output boards. For a complete listing of the specifications for all 4YDM input and output boards, see the AutoPatch Technical Specifications Manual.

D.3.1 Distortion and Noise Performance

CMRR	100 dB - DC to 1 KHz 70 dB - 20 KHz
PSRR	105 dB
THD	< 0.006 % over full range < 0.004 % DC to 10 KHz
DIM	< 0.002 % over full range

D.3.2 Frequency Performance

Gain Error	< 0.1 %
Input Impedance	25 K ohms
Output Impedance	50 ohms

D.3.3 Total Audio Specifications (end-to-end)

Input level	1 to +/- 12 V (balanced or SE)
Crosstalk	< -100 dB DC to 20 KHz
CMRR	100 dB DC to 20 KHz 70 dB - 20 KHz
THD	< 0.01 % over full range
DIM	< 0.015 % over full range
Voltage Noise	+/- 10 microvolts or > 120 dB at max. output
Input Impedance	25 K ohms
Output Impedance	50 ohms
Frequency Response	+/- 0.1 dB DC to 20 KHz +/- 3 dB DC to MHz
Adjustable Gain	-3 to + 10 dB
Drive Capacity	10 Vrms into 600 ohm load minimum 60 ma max. drive capacity (short circuit protected)
Maximum Cable length (to remain within THD specs)	500 feet (Belden 8451 or equivalent)

D.4 Video

This appendix contains a partial listing of the specifications for the input and output boards used in the 4YDM. For a complete listing, see the AutoPatch Technical Specifications Manual.

D.4.1 Overall Specifications Standard (formerly Model 40/32 Board)

Input Impedance	Hi-Z (47K) or 75 ohms
Bandwidth	40 MHz
	Note: bandwidth is reduced by driving multiple outputs
CMRR	80 dB
Differential Phase Error	< 0.2º
Differential Gain Error	< 0.5 %
Output Impedance	75 ohms
Input capacitance	7 pf
Crosstalk (Adjacent Input)	< -75 dB at 10 MHz
(Single Channel)	< -85 dB at 10 MHz
Equivalent Noise Floor	-154 dBm
Equivalent SNR	> 78 SNR
Max. output voltage swing	+/- 2.1 V
	+/- 3.2 V (typical)

Appendix E - APS Command Sample Program

The following program was written in C to test various APS commands with the 4YDM. For more information about APS commands, please refer to section 5.3.2, “APS.”

```
// APSTEST                Version A0.0           01/29/96
//
// This program tests various APS commands.
//
//
//
// Ver  Date   Whom  Description of Modifications
// ---------------------------------------------
// A0.0 01/29/96 PAH   Initial Version.
//


#include <dos.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <bios.h>
#include <time.h>

#include “comm.h”

// Structures

typedef struct STR_PTR
{
 unsigned char  *t;
 long int     len;
 struct STR_PTR *next;
} STR_PTR;

// Global Variables

COMM_PORT   *p0;

// Function Prototypes

void      delay(int);
void      publish_test_msg( char * );
void      flush_buffer( void );
int       MakeRnd(int,int);


// Main Function

int   main(int argc, char *argv[])
{
 FILE      *datafile;

 long       baud = 9600L;
 unsigned short  com = 1;
 int        options = 0;
 int        _init = 1;
 int        irqline;
 int        download = 1;
 int        size = 10;
 int        batch = 0;
 int        batch_mode = ‘N’;
 int        flood = 1;
 int        speed = 1;
 int        sendnconf = 1;

 char       *filename = “YTEST.TMP”;
 char       rom_version[10];

 struct STR_PTR *gened_packet;
 short      err;
 long       ct = 0,
   st,
   ft;
 int        i,j,k,l,m,match;
 short      ver1,ver2;
 char       *argcpy, *pdest;

 int        num_groups = 32;
 int        inputs[32] =
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32};
```

```
 int        outputs[32] =
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32};
 unsigned char  packet_cmd;
 unsigned char  packet_len;
 unsigned char  packet_loc;
 char       checksum;
 int       minin, maxin, minout, maxout;      /* Input & Output ranges */
 char       SendStr[15];
 char       ReturnStr[15];
 int       insw, outsw;
 int       return_outputs[32];
 clock_t     start, finish;
 double     duration, elapse;


// Set Seed

srand( (unsigned ) time(NULL) );

// Begin

printf("\nAPSTEST - XN Technologies, AutoPatch Division - Version A0.0, 01/29/96\n");
printf("\nThis program is used to test various APS commands.\n");
printf("\n");

if (argc > 0)              // Parse Arguments
{
 for   (j = 1; j < argc; j++)
 {
  argcpy = _strupr(_strdup( argv[j] ));
  match = 0;
  if (stricmp(argcpy, "/PORT=COM1") == 0)
  {
   com = 1;
   match = 1;
  }
  if (stricmp(argcpy, "/PORT=COM2") == 0)
  {
   com = 2;
   match = 1;
  }
  if (stricmp(argcpy, "/PORT=COM3") == 0)
  {
   com = 3;
   match = 1;
  }
  if (stricmp(argcpy, "/PORT=COM4") == 0)
  {
   com = 4;
   match = 1;
  }
  if (stricmp(argcpy, "/BAUD=1200") == 0)
  {
   baud = 1200L;
   match = 1;
  }
  if (stricmp(argcpy, "/BAUD=2400") == 0)
  {
   baud = 2400L;
   match = 1;
  }
  if (stricmp(argcpy, "/BAUD=9600") == 0)
  {
   baud = 9600L;
   match = 1;
  }
  if (stricmp(argcpy, "/BAUD=19200") == 0)
  {
   baud = 19200L;
   match = 1;
  }
  if (!match)
  {
   printf("\nError: Invalid argument in command line found %s", argv[j]);
   exit(1);
  }
 }
}

printf("\nCurrent Settings      Command Line    Current Value\n");
printf("\nCommunication Port     /PORT=       COM%d",com);
printf("\nBaud Rate          /BAUD=       %d",(int) baud);

printf("\n\n");

if ((datafile = fopen(filename, "w")) == NULL)
{
 printf("Error:  Could not open temporary file: %s\n", filename);
 exit(2);
}

if (com == 1)
{
 com = COM1;
 irqline = 4;
} else if (com == 2)
{
```

```
 com = COM2;
 irqline = 3;
} else if (com == 3)
{
 com = COM3;
 irqline = 4;
} else
{
 com = COM4;
 irqline = 3;
}

p0 = u8250_init(com, baud, DATABITS8, PARITY_NONE, STOPBITS1);
if (p0 == NULL)
{
 printf("\nFailure to intializing port");
 printf("\n\nCom port was initialized inproperly. Please check your connection,");
 printf("\ncabling, and communication baud rates.");
 exit(2);
}

install_ipr(p0, RECEIVE, NULL, 10);
install_ipr(p0, TRANSMIT, NULL, 100);
install_isr(p0, irqline, (PIHANDLER) NULL);

set_tx_xlat(p0, REMOTE_ECHO, OFF);
set_rx_xlat(p0, REMOTE_ECHO, OFF);
set_rx_xlat(p0, LOCAL_ECHO, OFF);
set_tx_xlat(p0, LOCAL_ECHO, OFF);

if (_init)
{
 c_putc(p0, 'X');

 _bios_timeofday(0, &st);
 err = EOF;
 while ((ct < 3) && (err != 'X'))
 {
  _bios_timeofday(0, &ft);
  ct = ft - st;
  err = c_inchar(p0);
 }

 if (err == EOF)
 {
  printf("\nCommunication not established with matrix.\n");
  printf("\nAcknowlegment was not recieved back from CPU. Please check the CPU");
  printf("\nand serial communication settings. The port attached to this computer");
  printf("\nwas initialized properly.\n");
  exit(1);
 }


 publish_test_msg( "Getting Version" );

 // Send packet to request make and version of matrix.

 gened_packet = ((STR_PTR *) malloc(sizeof(STR_PTR)));
 gened_packet->t = ((char *) malloc(4));
 gened_packet->t[0] = 0XB6;
 gened_packet->t[1] = 0X01;
 gened_packet->t[2] = 0X01;
 gened_packet->t[3] = 0XB8;
 gened_packet->len = 4;
 gened_packet->next = NULL;

 set_tx_xlat(p0, REMOTE_ECHO, OFF);
 set_rx_xlat(p0, REMOTE_ECHO, OFF);
 set_rx_xlat(p0, LOCAL_ECHO, OFF);
 set_tx_xlat(p0, LOCAL_ECHO, OFF);

 // Send packet
 for(k=0;k<gened_packet->len;k++)
  c_putc(p0,gened_packet->t[k]);

 // Wait for return status

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<10)&&(err!=gened_packet->t[k-1]))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
 }
 if(err==EOF)
 {
  printf("\nVersion request was not answered. Checksum not received.\n");
  printf("\nResponse from CPU was not received within allotted time frame. This is");
  printf("\na normal error message when communication with 4YDM Version C4 and below.\n");
 }

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<200)&&(err!=0XB7))
 {
  err=c_inchar(p0);
  _bios_timeofday(0,&ft);
```

```
 ct=ft-st;
}

if (err==EOF)
{
 printf("\nVersion request was not answered. Second attempt.\n");
 printf("\nResponse from CPU was not received within allotted time frame. This is");
 printf("\na normal error message when communication with 4YDM Version C4 and below.");
 exit(1);
}
else
{
 for (m=0;m<7;m++)
 {
  _bios_timeofday(0,&st);
  err=EOF;
  while ((ct<3)&&(err==EOF))
  {
   _bios_timeofday(0,&ft);
   ct=ft-st;
   err=c_inchar(p0);
  }
 }

 _bios_timeofday(0,&st);
 ver1=EOF;
 while ((ct<3)&&(ver1==EOF))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  ver1=c_inchar(p0);
 }

 _bios_timeofday(0,&st);
 ver2=EOF;
 while ((ct<3)&&(ver2==EOF))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  ver2=c_inchar(p0);
 }

}


sprintf(rom_version,"%c%c",ver1,ver2);
printf("\nCommunication Established with Matrix CPU Version %s\n", rom_version);

//in here eventually for B6 capture length and waste c_inchar for length
k=0;
while (k < 20000)
{
 err=c_inchar(p0);
 if(err!=EOF)
  k=0;
  k+=1;
}

_bios_timeofday(0,&st);
while (ct<6)
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
}

}


if (1)
{
 // Sending APS B1 Background Switch Command.

 publish_test_msg( "APS B1 Cmnd" );
 printf("\nPerforming APS, B1 Background Switch Command, Logical 8-4.");

 c_putc(p0,0xB1);
 c_putc(p0,0x02);
 c_putc(p0,0xC3);
 c_putc(p0,0x83);
 c_putc(p0,0xF9);

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<24)&&(err!=0xF9))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
 }

 if (err==EOF)
 {
  printf("\nFailure Sending APS B1 Command.\n");
  printf("\nResponse from CPU was not received within allotted time frame.");
  exit(1);
 }
```

```
// Sending APS B1 Background Switch Command, Multiple Logicals.

publish_test_msg( “APS B1 Cmnd, Mult” );
 printf(“\nPerforming APS, B1 Background Switch Command, Logical 10-12, 13-13.”);

c_putc(p0,0xB1);
c_putc(p0,0x04);
c_putc(p0,0xC4);
c_putc(p0,0x8B);
c_putc(p0,0xC6);
c_putc(p0,0x0C);
c_putc(p0,0xD6);

_bios_timeofday(0,&st);
err=EOF;
while ((ct<24)&&(err!=0xD6))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
 }

 if (err==EOF)
 {
  printf(“\nFailure Sending APS B1 Command.\n”);
  printf(“\nResponse from CPU was not received within allotted time frame.”);
  exit(1);
}

 // Sending APS B8 Background Switch Command.

publish_test_msg( “APS B8 Log Cmnd” );
 printf(“\nPerforming APS, B8 Background Switch Command, Logical 3-5, 4-6.”);

c_putc(p0,0xB8);
c_putc(p0,0x06); // Two Switches
c_putc(p0,0x00);
c_putc(p0,0x02); // Input 3
c_putc(p0,0x04); // Output 5
c_putc(p0,0x00);
c_putc(p0,0x03); // Input 4
c_putc(p0,0x05); // Output 6
c_putc(p0,0xCC);

_bios_timeofday(0,&st);
err=EOF;
while ((ct<48)&&(err!=0xCC))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}

if (err==EOF)
{
 printf(“\nFailure Sending APS B8 Command, Logical Test.\n”);
 printf(“\nResponse from CPU was not received within allotted time frame.”);
 exit(1);
}


// Sending APS B8 Background Switch Command, Literals.

publish_test_msg( “APS B8 Lit Cmnd” );
printf(”\nPerforming APS, B8 Background Switch Command, Literal 15-17.”);

c_putc(p0,0xB8);
c_putc(p0,0x03); // One Switch
c_putc(p0,0x21); // Literal, Cage 1
c_putc(p0,0x0E); // Input 15
c_putc(p0,0x10); // Output 17
c_putc(p0,0xFA);

_bios_timeofday(0,&st);
err=EOF;
while ((ct<48)&&(err!=0xFA))
{
 _bios_timeofday(0,&ft);
 ct=ft-st;
 err=c_inchar(p0);
}

if (err==EOF)
{
 printf(“\nFailure Sending APS B8 Command, Literal Test.\n”);
 printf(“\nResponse from CPU was not received within allotted time frame.”);
 exit(1);
}

// Sending APS B8 Background Switch Command, Program Preset.

publish_test_msg( “APS B8 Prg Preset” );
printf(”\nPerforming APS, B8 Background Switch Command, Program Preset 10");

c_putc(p0,0xB8);
c_putc(p0,0x03); // One Data Structure.
c_putc(p0,0x41); // Operation Code, Program Preset, Enclosure 1.
c_putc(p0,0x00); // Element Ignored.
```

```
 c_putc(p0,0x09); // Preset 10.
 c_putc(p0,0x05);

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<48)&&(err!=0x05))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
 }

 if (err==EOF)
 {
  printf("\nFailure Sending APS B8 Command, Program Preset.\n");
  printf("\nResponse from CPU was not received within allotted time frame.");
  exit(1);
 }

 // Sending APS B8 Background Switch Command, Literals, Just to Change the Config.

 publish_test_msg( "APS B8 Lit Cmnd" );
 printf("\nPerforming APS, B8 Background Switch Command, Literal 16-17.");

 c_putc(p0,0xB8);
 c_putc(p0,0x03); // One Switch
 c_putc(p0,0x21); // Literal, Cage 1
 c_putc(p0,0x0F); // Input 16
 c_putc(p0,0x10); // Output 17
 c_putc(p0,0xFB);

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<48)&&(err!=0xFB))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
 }

 // Sending APS B8 Background Switch Command, Execute Preset

 publish_test_msg( "APS B8 Exe Preset" );
 printf("\nPerforming APS, B8 Background Switch Command, Execute Preset 10");

 c_putc(p0,0xB8);
 c_putc(p0,0x03); // One Data Structure.
 c_putc(p0,0x61); // Operation Code, Program Preset, Enclosure 1.
 c_putc(p0,0x00); // Element Ignored.
 c_putc(p0,0x09); // Preset 10.
 c_putc(p0,0x25);

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<48)&&(err!=0x25))
 {
   _bios_timeofday(0,&ft);
   ct=ft-st;
   err=c_inchar(p0);
 }

 if (err==EOF)
 {
   printf("\nFailure Sending APS B8 Command, Execute Preset.\n");
   printf("\nResponse from CPU was not received within allotted time frame.");
   exit(1);
 }

 if (err==EOF)
 {
   printf("\nFailure Sending APS B8 Command, Literal Test.\n");
   printf("\nResponse from CPU was not received within allotted time frame.");
   exit(1);
 }
}

publish_test_msg( "Tests Completed." );

return (0);
}


/*  Delay - Causes a pause for a certain amount of clock ticks.  Clock ticks
 should be somewhat constant independent of the CPU this program will
 run on.  Phil Hale  29-Sep-94
*/
void delay( int t )
{
 long ct=0;
 long st, ft;

 _bios_timeofday(0,&st);
 while (ct<t)
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
 }
}
```

```
/*  Publish Test Message - Clears screen, displays "Test in Process" on the first line
 and on the third line displays the msg string sent from calling program.
*/

void      publish_test_msg( char *msg )
{
 int       _len;
 int       i;
 int       checksum = 0;
 short      err;
 long       ct = 0,
   st,
   ft;

 _len = strlen(msg);

//   printf("\nString Publishing = [%s], Len = [%d]",msg,_len);
 c_putc(p0,0xB0);
 c_putc(p0,(unsigned char) (_len + 18));
 c_putc(p0,0xC8);     // Clear Screen
 c_putc(p0,0xEE);     // Position Cursor at Beginning of First Line
 c_putc(p0,'T');
 c_putc(p0,'e');
 c_putc(p0,'s');
 c_putc(p0,'t');
 c_putc(p0,' ');
 c_putc(p0,'I');
 c_putc(p0,'n');
 c_putc(p0,' ');
 c_putc(p0,'P');
 c_putc(p0,'r');
 c_putc(p0,'o');
 c_putc(p0,'c');
 c_putc(p0,'e');
 c_putc(p0,'s');
 c_putc(p0,'s');
 c_putc(p0,0xEC);     // Position Cursor at Beginning of Third Line

 checksum += (0xB0 + _len + 18 + 0xC8 + 0xEE + 'T' + 'e' + 's' + 't' + ' ' + 'I' + 'n' + ' ');
 checksum += ('P' + 'r' + 'o' + 'c' + 'e' + 's' + 's' + 0xEC);

 for( i=0; i<_len; i++)
 {
  c_putc(p0,(unsigned char) msg[i]);
 checksum += msg[i];
 }

 checksum = (checksum & 0X00FF);
 c_putc(p0,(unsigned char) checksum);

 _bios_timeofday(0,&st);
 err=EOF;
 while ((ct<24)&&(err==EOF))
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
 }
 if (err==EOF)
 {
  printf("\nFailure Publishing Text.");
  printf("\nResponse from CPU was not received within allotted time frame.");
  exit(1);
 }
 if (err != (unsigned char) checksum)
 {
  printf("\nFailure Publishing Text.");
  printf("\nPacket not downloaded.  Invalid Returned Checksum.  Expected: %x, Received:
%x",(unsigned char) checksum,err);
  exit(1);
 }

 flush_buffer();

 return;
}


void  flush_buffer( void )
{
 short      err;
 long       ct = 0,
   st,
   ft;

//   printf("\nFlushing Buffer: [");
 _bios_timeofday(0,&st);
 err=EOF;
 while (ct<12)
 {
  _bios_timeofday(0,&ft);
  ct=ft-st;
  err=c_inchar(p0);
//        if (err != EOF)
//            printf("%c",err);
 }
//   printf("]\nBuffer Flushed.");
```

```
 return;
}

/*   MakeRnd - Returns a random integer between two numbers.
*/

int   MakeRnd(int rmin, int rmax)
{
 int t=0;
 int rnddiv;

 rnddiv = rmax-rmin;

 while ((t<rmin) || (t>rmax))
 {
  t=rand();
  t = (( t / rnddiv) +1 );
 }
 return(t);
}
```

Glossary

4YROUTE - An AutoPatch software program used to control the 4YDM from an IBM or compatible PC. 4Y-ROUTE is implemented in DOS format menus.

Advanced Packet Structure (APS) Commands - A group of hexidecimal, packetized commands created by Auto-Patch for use with the 4YDM. APS commands are uploaded to the matrix in the packet section of a configuration file and are used to control the matrix and request information.

backlight - The light that illuminates the LED screen of the control panel on the enclosure. The backlight is a control panel key.

Basic Control Structure (BCS) - A string of alphanumeric characters used to serially control the matrix from a PC keyboard. BCS commands can be used to execute any command that can be keyed-in from the control panel.

BAUD rate - The speed that communications travel through the serial port. A 4YDM can send and receive communications at 2400, 4800, 9600, and 19200 BAUD.

check sum - A crude way of error-checking the information sent in a packet. When a packet is uploaded to the matrix, the check sum is the total value of the information in the packet. The program that reads the information in the packet verifies that all the information it received adds up to the check sum. If the check sum equals the sum of the information, all information was received; if the check sum does not equal the sum of the information, something happened when the information was uploaded and the information needs to be uploaded again.

com port - Refers to the communications port, or serial port. The term com port is used in several of the software packages and devices used to control the matrix. There are two com ports on every matrix enclosure.

component signals - Groups of signals that are switched together because each signal carries a necessary component for that group of signals to be recognizable. Some examples of component signals are: RGBS video, RGsB video, Y-c video, and stereo audio. Each signal in a component signal travels through a separate wire or cable, but all signals are switched together so that they arrive at the output device at the same or vitually the same time.

configuration file - A text file used to upload information to the CPU. The configuration file can contain four different kinds of information: a groupings table, a string section, a packet section, and a preset section. Each section either contains information that is used by the matrix, or contains commands that control the matrix functions.

connector assembly - An adaptor that allows the audio, video, or data signals to pass from a source device to the input board, or from the output board to the destination device.

control panel - The panel on the front of an enclosure that has buttons and a screen for viewing messages about the distribution matrix. Almost all distribution matrices will have at least one enclosure with a control panel.

(distribution) matrix - The environment of all signals and the hardware and software necessary to switch these signals. Distribution matrices are also known as routing switchers, routing matrices, and switching matrices.

dry contacts - Dry contacts are a non-electrical method of control that can be used with a 4YDM to execute presets. The dry contact must have switches that provide the hexidecimal value of the preset to be executed. The dry contact module polls the switches and uses the value provided to execute the desired preset.

echo serial command - A feature of the matrix that when turned on, echoes the commands sent to the matrix from an external controller on the control panel screen.

enclosure - An enclosure is a metal chasis which holds input and output boards, a CPU board, and a power supply. Each enclosure in the matrix can contain up to eight input and eight output boards providing a total capacity of up to 32 input and 32 output signal paths.

external controller - An external controller can be either dry contacts or any device that can be used to control the matrix via the serial port.

grouping table - A grouping table defines the signals that comprise the component and non-component signals input to and output from a matrix. Each definition is specified as an input or an output and is given a specific input or output number (these input and output definitions are logical inputs and outputs). A grouping table can hold 128 input and 128 output definitions. The input and output numbers given in the grouping table are used to specify particular input and output definitions.

input/output (I/O) configuration - The input/output configuration is the status of all the input and output signals in the matrix (or enclosure, if so specified) at a given time. For example the status of an input signal would be the output signal to which it was routed and the status of an output signal would be the input signal it was receiving.

input board - A circuit board that receives video, audio or data signals from outside sources.

level - The 4YDM uses the term level to refer to an enclosure. An enclosure is selected by entering its enclosure number after the Level prompt in a Change or Status operation. Each enclosure has a distinct enclosure number; to specify all enclosures, enter ‘0’.

literal inputs and outputs - Literal inputs and outputs are the physical connector locations on the rear of an enclosure. Each enclosure holds a maximum of 32 literal inputs and 32 literal outputs.

logical inputs and outputs - Logical inputs and outputs are definitions of groups of physical input and output signals. These definitions are specified in the grouping section of the configuration file. Logical input and output definitions can specify from one to eight signals. When a logical input or output is switched, all signals listed in the definition are switched. There can be 128 logical inputs and 128 outputs defined for use in the matrix.

logical status table - A table in the CPU that keeps track of all logical inputs and outputs. This table is updated whenever a logical switch is made and is the source when the status of a logical input or output is checked.

master enclosure - An enclosure that has its EPLD chip set to 01. This setting, which is done at the factory, identifies this enclosure as the master. Only one enclosure per matrix can be the master enclosure. The master enclosure checks the slave enclosures and initializes them on startup.

matrix verification - Upon startup, or whenever a user requests it, the master enclosure checks the links with its slave enclosures. This verifies that all enclosures are properly linked and that the matrix can act as a single entity.

module - Also known as a software module. A module is a chunk of software code that allows the matrix to accomplish specific tasks.

input/output configuration - The status of all the input and output signals in the matrix at a given time. For example, the status of an input signal is the output signal to which it is routed and the status of an output signal is the input signal that it receives.

Preset key - The Preset key is used to store the I/O configuration of the matrix. Using the Preset key, up to 32 different I/O configurations can be saved. The Preset key is also used to restore the I/O configurations. The entire matrix can be reconfigured at any time, exactly as it was saved, by recalling an I/O configuration.

Program key - The Program key is used to arrange the enclosures of the matrix in a desired fashion, check configuration errors, reroute logicals, program the general configuration word and set the BAUD rate and polling range of the serial configuration word.

output board - A circuit board that routes input signals to specified destinations.

output status - A function of the matrix used to find out the input being routed to the specified output.

RGBS - A four input signal video signal, R = Red, G = Green, B = Blue and S = sync. This signal requires four literal inputs to relay it to an output device.

serial configuration word - The serial word is a hexidecimal number that tells the matrix how to: set the BAUD rate, choose the number of SBC's to poll, and whether or not to echo serial commands on the control panel.

serial filter - When turned on, the serial filter changes hexidecimal packets of information, traveling from the CPU of the matrix to an external controller, into more readable ASCII characters.

single bus controllers (SBC) - SBC's are remote controller devices used to control the input to a specified output device.

slave enclosure - Any enclosure in a matrix other than the master enclosure.

software module - A software module is a chunk of software code that allows the matrix to accomplish specific tasks.

strings - Basic Command Structure (BCS) commands are referred to as strings.

Special key - The Special key can hold 3 distinct functions which will occur when the key is pressed. The key can: execute the Preset #1, cause the associated literal input to be switched to all outputs on all levels, and execute an identity matrix to occur in all enclosures.

switch - Any time that the Change key is used alter the input/output configuration, a switch was done. Switches can be made from the control panel, an external controller, and the packet and string sections of a configuration file. A switch does not include presets.

YTOOLS.EXE - The executable program used to compile and upload a configuration file to matrice's CPU. YTOOLS.EXE is included on the floppy disk sent with the matrix.

Index of Tasks

4YDM Reference Manual Index

T

U

V

W

Y