Professional MIDI Management Systems

MIDIX

with
MIDI RECALL™

Versions 1.3 & 2.0
Operating Manual

June 1991

HINTON
INSTRUMENTS

CONTENTS

MIDIX Specification

MIDI Processing Rack Size:	19″ sub–rack or free standing case, 3U or 6U
Depth:	250mm behind panel, 50mm handle protruding
Power:	120/240V 50/60Hz, IEC fused filter inlet

All MIDI connectors professional grade metal DIN 41524 type
All opto–isolators high speed type
All ICs socketed

MIDIX 16 – MIDI Routing Matrix

Maximum MIDI Inputs:	16
Maximum MIDI Outputs:	24 (6U Case)
In to Out delay:	‹200ns
Cascade delay:	‹2us (16 channels)
Pulse width distortion:	‹0.7%

Solo, Stop and safe switching in hardware
Full LED indication of status

MIDIX Controller with MIDI Recall™

Software version:	1.3a November 1989

Manual, MIDI and RS232 control, footswitches & remote optional
Illegal Groups, Solo Groups, Stop, Panic Clear and Clear Event in software
Hex and English MIDI monitoring with Filtering
VT100 compatible paged displays

Number of Patches:	50 equatable to any MIDI Program
Number of Events:	50
Event List maximum size:	50 bytes
Chain maximum length:	50 Patches, any order
Maximum System Exclusive Block Recalled:	23054 bytes
Patch title size:	20 characters
Input/Output name size:	16 characters
Event name size:	16 characters
RS232 connector:	D25BM
RS232 signals:	TxD, RxD, Common
RS232 Protocol:	8 data bits, 2 stop bits, no parity, XON/XOFF flow control
Baud rates:	9600Bd standard Either 19k2 or 38k4 optional

MIDIZ – Long Haul Converter

Proprietary fully opto–isolated, balanced design

Cable type:	Audio grade screened twisted pair
Connector type:	XLR compatible
Maximum cable length:	› 1000 metres
Delay:	0.5us 100 metres 11us 1000 metres
Pulse width distortion:	‹1.0%

This specification and the information presented in this document may change without notice in the interests of continuing product improvement. While every effort is made to make this document accurate Hinton Instruments cannot accept any responsibility for the interpretation of the information provided or any liability for any injury, loss or damage, direct or consequential, caused by application or inability to use the equipment or information provided.

HINTON
INSTRUMENTS
Oldford, Nr. Frome, Somerset, BA11 2NN, England.

Mechanical

The MIDI Processing Rack takes 3U/6U of 19″ rack space and requires a mains power supply with IEC connector on the left hand side. The rack is shipped with the mains voltage set for the country of destinastion. Please take normal precautions for siting an electronic instrument, do not place on a source of direct heat, in direct sunlight or near another instrument that may interfere with either's operation. The rack will dissipate less than 40W and is cooled only by natural convection. If sited in a rack with vertical convection or fan cooling the plastic trim strips on the top extrusion may be removed to expose ventilation slots.

Earthing

The case is connected to Mains Earth and must be grounded for safety. The sole ground reference should be provided via the mains inlet and in countries that normally use a two pin mains connector a technical earth must be provided via an IEC lead with the mains supply.

Do not provide a ground via the rack mounting bars. Use insulating mounting kits if this is the practice.

Do not remove the ground from the mains connector. *Failure to observe this practice may result in hazard to life or damage to connected equipment.*

MIDI and MIDIZ Long Haul are both optically isolated systems and the cable screens are connected to the driving circuitry ground, but not to the receiver ground. Do not connect the cable screens to the plug shells as this will connect to the case ground via the receptacle chassis and may cause grounding related problems.

After siting the rack connect all permanent cabling:

MIDI

The metal DIN connectors used for MIDI Ins and Outs may be used with latching type plugs as well as the normal type. Any MIDI connectors on the front panel will normally be paralled to a corresponding connector on the rear panel, but should not be used at the same time. One MIDI driver should only be connected to one MIDI receiver, irrespective of whether it is powered on or not. Use only quality cables intended for MIDI use and not hi–fi cables. Cable lengths should not exceed 15 metres. Longer cables, including special long run types, cannot be guaranteed to work in all circumstances as they have no control over the opto–isolator type used in the receiving equipment, its sensitivity or its aging characteristic.

Long Haul

MIDIZ Long Haul connectors on the rear panel are XLR compatible types intended to be used with balanced audio distribution cabling. Outputs are male and inputs female. The electrical signals correspond to a high audio level (5V complimentary pulsetrains) equivalent to greater than +8dbm and are, of course, at the MIDI transmission rate of 31.25kHz. *This may cause damage if connected to audio equipment particularly monitor loudspeaker voicecoils* which are not designed to withstand continous high frequency tones and more care should be taken than when routing SMPTE timecode signals.

The proprietary MIDIZ circuitry is optically isolated, similar to the MIDI circuitry, and will drive over 1 kilometre of balanced screened cable with a large tolerance to cable and connector types i.e. it does not require a controlled impedance like many other transmission systems. The screen should be maintained throughout the run and should not be earthed along the way e.g. at patchfields. Failure to observe this practise may result in high frequency noise being picked up by audio equipment. Care should be taken to avoid connection to unbalanced cable systems that short one of the conductors to ground e.g. special patchfield cables or XLR adaptors.

Otherwise observe a practise similar to MIDI:

* One output drives one input only.

* Do not place phase reversers, attenuators, transformers or active circuitry in line.

* Avoid connecting two outputs together or to phantom power supplies, although the drive circuitry appears to be able to sustain these conditions it is not guaranteed.

RS232

The MIDIX Controller should be connected via RS232 to a computer to be used for full control and bulk storage. If the computer has a 25–way male D–type connector on its RS232 port then it is likely to be a standard RS232 DTE connection, otherwise suspect the computer manufacturer and consult their manuals carefully. IBM PCs and Atari STs conform, but the Apple Macintosh will require a conversion cable and other types could be practically anything! MIDIX is configured as a DTE so a "Null Modem" connection is required to connect two DTEs together. This is simply a 25–way D female to 25–way D female cable with pins 2 and 3 crossed over and a common:

TxD	2	←→	3	Rx
RxD	3	←→	2	TxD
Common	7	←→	7	Common

The computer or terminal should be configured as follows:

9600 Baud on Transmit and Receive
8 Data bits
2 stop bits

No parity
XON/XOFF handshaking ON
CTS/RTS handshaking OFF
ENQ/ACK handshaking OFF

Checking

When connected, power up the MIDI Processing Rack. The presence of internal ac power is indicated by an LED behind the PSU front panel, if not on check the mains power supply and the mains fuse within the IEC mains inlet. Each MIDIX module should have at least one red LED illuminated and all yellow LEDs off. Check that the yellow LEDs toggle on and off by pressing the Solo buttons and that the red 1–16 LEDs can be advanced with the Select button. The MIDIX Controller has two LEDs, a red Run and a green Busy, and in normal operation the red is permanently on and the green flickering slightly. If the red LED ever goes off or the green permanently on a fault has developed which may be cleared by pressing the shrouded reset button. There is no difference between switching on the mains power or pressing either front or back reset button, all execute a complete reset and load the last selected patch (not edit).

Start up your preferred communications software on the computer and when running press the MIDIX reset button. A sign–on message should appear on the screen similar to:

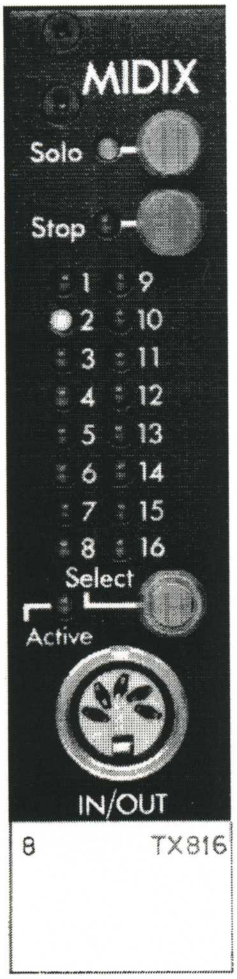
```
MIDIX-16 Controller with MIDI Recall (TM)
Vers:1.3a (c) 1989 Hinton Instruments

MIDIX Size:  16 Inputs into  24 Outputs
 50 Patches equatable to any MIDI Program number
 50 Event Lists of up to 50 MIDI bytes each
Maximum System Exclusive Block capture size: 23054 bytes

Memory Protect:   0
Base Channel:   0
Front Panel enable:  1

Patch 1: ....................

If using a terminal with VT100 capability:
 first enable this mode and then type:
        T2 <ret>
```

The sign on message will display the version and revision number of the software and information on the current configuration, options and hardware state. Check that this is what you expect it to be.

If nothing appears type Control–Q (hold down Control and then type Q) which is the XON code and may be needed if the computer has transmitted an XOFF. If there is still no response check all the RS232 connections carefully. If random characters appear the computer has not been set to the correct baud rate. Type H a few times, this should generate several pages of the help message which will scroll up and off the screen. If characters or words are missing then the XON/XOFF flow control has not been enabled.

If you experience any difficulty with the above operations, please contact Hinton Instruments for advice.

MIDIX Routing Module



Each MIDIX module comprises a MIDI Input which may drive one of sixteen buses, a bus selector and a MIDI Output. If there are more than sixteen modules in a system the MIDI Input section is not used in the remainder. There are various optional settings for each module: the front panel MIDI socket may be set to be an Input, Output or not used by plugging an internal lead to one of three labelled positions. Each module may be set to Controller or Manual Only operation by a link (so that it is not affected by patch changes) and Soloing may be disabled within Manual mode by another link. A Rack is normally supplied configured to suit an application, but may be reconfigured after experience or change of use dictate. Please refer to the Service Manual for full details.

The MIDIX module front panel is an *Output Selector* and the current routing state is displayed on the red **Select** LEDS. The small grey Select pushbutton advances the Input bus selected from 1 to 16 in manual mode and through the actual legal number of inputs present in the rack when under software control.

The red **Stop** LED indicates that the module output will not transmit even if there is data on the selected bus. The Stop state is toggled by the large red pushbutton.

The green **Active** LED shows that data is actually being output from that module and the brightness is proportional to the data density. Active sensing or slow tempo clocks show as a dim flicker and MIDI Controller movements and System Exclusive blocks as bright bursts. Individual bytes may be seen as a brief flash and a solidly on bright light is probably a MIDI feedback loop.

The yellow **Solo** LED is only set or cleared by its associated large grey pushbutton and will stop MIDI output on all other non–Soloed modulesin the same *Solo Group*. The Solo condition is not part of of any patch and is not stored by the Controller. The Stop state overides Solo and MIDIX Controller transmissions override Stop. The indicator really shows a Solo *request* that will be observed if there are no overiding conditions.

The MIDIX modules employ a *Safe Switching* circuit that only allows the routing to change in the gaps between bytes to prevent the MIDI data from being corrupted. If MIDI data were corrupted, because of its inherent structure, receiving equipment tends to interpret the damaged codes as bytes numerically towards the higher end which has been allocated to all the System Real Time commands i.e. the worst possible to receive by mistake! A delay of the order of 1 millisecond prevents data bytes from being separated from a preceding status byte. Patch and Routing Changes may sometimes be delayed until current data activity has ceased.

MIDIZ Dual Long Haul Converter

Each MIDIZ module comprises two MIDI to Long Haul converters and two Long Haul to MIDI converters. Each pair is designated "A" and "B", but there is no other connection between the pairs and they may be used completely independently.

The front panel has two green LED indicators marked **Receive A** and **Receive B** which display incoming Long Haul to MIDI data. These displays can show individual MIDI bytes as a brief flash and other types of MIDI data produce characteristic flickering or various intensity glows. With practise different types of MIDI data may be readily identified – remember that MIDI transmits a Note Off code when keys are released which creates flashes on the LEDs too.

If an LED stays on at full intensity it indicates that a MIDI feedback loop has been created, usual causes of this problem are routing the Output of a merger or sequencer back to its Input.

Any MIDI source may be connected to the MIDIZ MIDI In, converted to Long Haul, transmitted over audio tie lines up to 1 kilometre in length, connected to another MIDIZ and converted back to MIDI. The limiting factor with the length of transmission is the dc resistance of the cable conductors which should be less than 100 ohms per conductor (most audio twisted pair cabling is about 80 to 90 ohms per conductor per kilometre).

There is a delay in the signal corresponding to the cable length and is of the order of 11 microseconds for a kilometre. As the MIDIZ receiver reconstitutes the MIDI data this is a pure delay and not a data distortion like that normally associated with MIDI Thrus using slow speed optocouplers. Most MIDI processing devices delay MIDI data in excess of 400 microseconds so the delays due to Long Haul transmission will not be perceived or cause any adverse effect.

MIDIZ converters may be cascaded on a point to point basis to send MIDI data between several MIDI Processing Racks distributed over a wide area.

Routing of MIDI or Long Haul data should not be changed while data is active. MIDIX Routing modules employing intelligent switching techniques are recommended to alleviate this problem.

The MIDIZ Long Haul circuitry is a proprietary design to a high specification. It is *not* RS422 as has been used for similar purposes with mixed results. If you wish to use MIDIZ in conjunction with any other long haul converter, please consult Hinton Instruments first regarding compatibility. It *might* be possible to modify the other equipment.

Setting Up

Introduction

The MIDIX MIDI Routing Matrix is a third generation sophisticated digital switching system for the interconnection of large numbers of MIDI devices in a professional environment. MIDIX is not a simple device and it has many advanced features, but it has been designed to be easy to use, to be reliable and to enhance rather than obstruct.

All the MIDI equipment is connected to the MIDIX matrix where they are inter-routed under the direction of the MIDIX Controller which is connected both to the matrix via MIDI and to an external computer via RS232. All settings are stored in the MIDIX Controller's non-volatile memory and may also be loaded and dumped by the external computer. MIDIX may be operated manually by its front panel controls without the computer connected at all, but initial setting up and full operation and displays require the computer.

Please refer to the installation section for full information on setting up the MIDIX hardware and connecting to the computer. This section is intended to familiarise an operator with the basic concepts and day to day use and assumes that you have powered up the rack and have a Terminal Emulator running on the computer. More detailed descriptions of the commands are given in the Command and Reference sections.

The MIDIX-16 matrix has up to sixteen input buses and may have more outputs. One of these, usually the last, is the Controller. Each MIDIX module contains a MIDI Input stage that feeds one of the buses and a selector that routes from one of the buses to a MIDI Output stage. If there are more than sixteen modules in a system the MIDI Inputs above sixteen are redundant, all modules are identical to facilitate rapid servicing.

A MIDIX module is required for every piece of equipment to be interconnected. Most equipment has a MIDI In and a MIDI Out so will logically be associated with one module. Some devices have unequal numbers of Ins and Outs so some thought is needed to allocate them to the matrix in a managable fashion. Equipment with no Outs may be connected in a chain to one module by their Thru outputs. If some equipment only uses its MIDI Output for bulk dumps it could be merged with others to condense the matrix usage. This subject would normally have been covered before purchase, but may need to be revised from time to time as setups are changed or expanded. If you require advice please contact Hinton Instruments' Technical Support.

Memory Protect

The first task to be done with a new system or after any changes in the equipment is to name all the matrix inputs and outputs and before that may be done *Memory Protect* must be set to off by typing:

```
M 0 ␍
```

when all the changes have been made this may be set back on by typing:

```
M 1 ␍
```

Naming Inputs and Outputs

Each connection may have up to a sixteen character long name which will be used on all displays giving routing information. There is no extra knowledge associated with the name, like identifying particular models of equipment, it is simply a label for the operator's convenience. All commands associated with naming use the quotation mark (") to start the name and a carriage return (CR) to end it, extra characters are ignored.

To name an Input type:

```
I n "sixteencharacter ␍
```

where n is the input number between 1 and 16, or to name an output type:

```
O n "sixteencharacter ␍
```

Names may be changed at any time, but like all settings it is recommended that a file is kept on disc as a backup. To check what all the names currently are the *Dump Names* command may be used. Typing `D"` will give a list of all the Inputs and Outputs in the system.

Naming and Defining Events

Each MIDIX Output may have an *Event* associated with it that will be transmitted whenever it becomes disconnected from an Input. An Event is an arbitary user definable list of MIDI codes up to fifty bytes in length. MIDIX is supplied with several common and useful ones already defined in its memory, but this may be redefined at any time. Events are referred to by the number sign (#) and a number from 1 to 50, they also have a sixteen character name.

To define an Event type:

```
# n < hh, hh, hh, . . . hh> ␍
```

where n is the event number and hh are the MIDI codes in hex and to name it type:

```
# n "sixteencharacter ␍
```

To check an event type:

```
# n ␍
```

and the name followed by the hex list will be given. All the events may be dumped with the *Dump Events* command, `D#`. This is quite a lot of information and will be more than a screen full and is really intended for saving to disc.

Clear Events

To associate an Event with an Output type:

```
CL n  # nn ␍
```

where n is the output number and nn is the Event number. Notice how the command is expanded when you type it and the current setting shown, if you do not wish to change it simply type {CR}. The complete *Clear List* may be obtained by using the *Dump Clear List* command, **DC**.

If no Event is required to clear an Output, enter 0 for the Event number. Events may also be used for other purposes which we will cover later.

Solo Groups

Each MIDIX module has a Solo button on it that has a definable action. You have probably already played with them and wondered why they did not seem to work! In a large MIDI system it is often desirable to solo certain instruments quickly, but it would be disasterous if, for example, the clock was cut off to the sequencer at the same time. Each module may have an independent *Solo Group* defined as a list of what other modules are Stopped when Solo is set.

Say we have three synthesizer expanders on Outputs 3, 4, and 5 and a drum machine on Output 6. If each modules' Solo Group were defined similarly as:

```
G n (3, 4, 5) ␍
```

where n = 3, 4, 5 and 6 in turn, we could then solo each expander and still hear the drums, but stop all the expanders when we solo the drums. Think about that one and wonder why mixing desks don't work like that.

It does not matter that the module's own number appears within the solo definition as its own Solo button overides that, it just makes it easier to identify the action when looking at lists of numbers. If no Soloing action is required on any module it must be given a null group:

```
G n () ␍
```

A Solo Group may be shown by typing:

```
G n ␍
```

or all of them may be saved with the *Dump Solo Group* command, **DG**.

Illegal Groups

In a similar style each module output can have a defined *Illegal Group*. This is a list of all the Inputs that cannot be connected to that Output. Certain MIDI devices such as sequencers and mergers can lock up if they have their output connected to their input forming a *MIDI Feedback Loop* and this is the first line of defence. To define an Illegal Group type:

```
X n ( n, . . . n ) ␍
```

where the first n is the Output number and the ns in the list are

Input numbers that should not be connected to that Output. Of course, this cannot prevent two MIDI devices being connected in a loop, but please refer to the *Hold* command for use in that eventuality.

An Illegal Group may be cleared by typing:

```
X n () CR
```

inspected by typing:

```
X n CR
```

and saved with the Dump Illegal Groups command, **DX**.

Everyday Use

Status Lock

Now we have defined Names, some Clear Events, Solo Groups and Illegal Groups we can use MIDIX to make some connections. MIDIX is always in edit mode and routing may be altered at any time by using the front panel *Select* and *Stop* buttons or with corresponding commands. There is a Status Lock feature to guard against accidentally knocking the panel buttons, this may be set by the *Front Panel Enable* command or by an optional panel switch.

```
F 1 CR
```

enables the front panel buttons and takes Status Lock off.

Routing

There are several Output commands, we have already used the Output Name (O"). All routing is accomplished with the Output Equals command (O=):

```
O n = m CR
```

where n is the Output number and m is the Input number or 0 for off. After entering the {CR} MIDIX will report on the connection made using the already defined Input and Output names. To see what an Output is set to type:

```
O n CR
```

and to see where an input is connected to (there may be more than one connection) type:

```
I n CR
```

The complete current routing may be shown with the *Dump Current Status* command, **D CR**.

Routing is accomplished on the front panel in either of two ways, One Finger mode or Two Finger mode.

One Finger Mode

To select an Output routing in One Finger mode press the Select button on that module. The Input bus indicator LEDs will advance on the *release* of the button. Sometimes it will skip over one or more Inputs – this is because they have been defined as an Illegal Group. There may also be brief flashing to the last Input and a flash on the green Active LED – this is the MIDIX Controller sending out the Clear Event for that Output. If there is MIDI flash on the green Active LED – this is the MIDIX Controller sending out the Clear Event for that Output. If there is MIDI activity such as sequencers playing it is advisable to set the module in Stop before using Select to prevent sending snatches of MIDI codes not intended for that Output.

Two Finger Mode

Two Finger mode is more direct to use. Press the Select button on the module number corresponding to the Input source and keep the button held. Now press the Select button(s) on any Output module(s) that you wish to route TO. The routing will change on the DOWNSTROKE after sending any defined Clear Event. Finally, release the Input Select button. If you attempt to make a connection defined as Illegal the Stop LED will come on and transmission on that Output will be inhibited.

It is not possible to route an Input to an Output on the same module in Two Finger mode, use One Finger mode. Usually this would be undesirable if both are connected to the same equipment.

If you select an Input and then change your mind the release will constitute a One Finger type advance. To prevent this, route to an Output that is already set i.e. no change or to an Output not being used.

Patches

Once a routing configuration has been set up it may be stored in one of fifty Patch Memories for later recall. The current setting is kept in an Edit Buffer and it is possible to toggle between this and the last patch with a compare/recall action like synthesizer program memories.

To STORE a Patch type:

`S nn` CR

where nn is the memory number between 1 and 50 and defaults to the last recalled memory. To RECALL a Patch type:

`P nn` CR

and to toggle between the Edit Buffer and this Patch after subsequent editing simply type:

`P` CR

Patches may also by recalled by MIDI PROGRAM CHANGES. Please refer to the PATCH EQUATE and BASE CHANNEL commands if you wish to use this facility.

Naming Patches

Each Patch and the Edit Buffer may have up to a twenty character name. The Edit Buffer may be named by typing:

`"twentycharactername` CR

and this will be transferred to a memory together with the patching information with the STORE command.

Patches may also be named directly by typing

`P nn "twentycharactername` CR

Chains

Patches may be joined into a CHAIN to create a cycle for working e.g. Record into Sequencer / Edit Synth 1 / Edit Synth 2 / etc. A Chain is a list of Patch numbers in sequence, a number may appear more than once, up to a total of fifty steps and is defined by typing:

CH [nn, nn, nn, . . . nn] CR

A Chain is entered by typing open bracket (**[**), advanced with slash (**/**), retarded with backslash (****) and exited with close bracket (**]**). An optional footswitch may also be used to advance through the Chain.

To show the current Chain List type:

CH CR

Dumps

All the information that we have set up may be dumped separately or altogether with the Dump commands. We have already introduced Dump Names, Dump Events, Dump Clear List, Dump Solo Groups, Dump Illegal Groups and Dump Current Status. The remaining ones are Dump Patch (**Dnn**), Dump All Patches (**DP**), Dump MIDI Equates (**D=**), Dump Chain List (**D[**), Dump Filters (**D%**) and Dump Everything (**D&**). Most of these commands are executed immediately the second character is typed without waiting for a CR or allowing a correction.

All the dumps are in MIDIX command format so that all that is necessary to save and restore a complete setup is to keep the information as a disc file with a communications program that provides disc filing facilities.

Dumping Setups to Disc

To Save any MIDIX setup first enable your communications program to receive data for filing, then type the appropriate dump command or commands and watch the information on the screen. When it has finished close reception and save the file.

All the information is in printable ASCII, so may be sent to any printer or edited with any word processor. It is better to design Event Lists and MIDI Program Equates "offline" with a word processor than to type it into MIDIX directly.

Reloading Setups from Disc

To Restore any setup or partial setup select the file for transmission with the communications program and send it. You will see all the information on the screen confirming that MIDIX has accepted it. Notice that the information has been expanded from the minimum condensed form used for dumping.

MIDIX is completely independent of computer model types and disc formats and sizes. When we started designing MIDIX popular opinion considered BBC Bs and Commodore 64s as serious personal computers. It is now obvious that they were jokes. In a few years time Atari STs, IBM PCs and Macintoshes may be equally obsolete and unsupported, in fact the first versions of these already are, but MIDIX will still be performing the same role with the same data regardless of where you store it.

There are two special commands to deal quickly with problem situations, these are PANIC and HOLD.

Panic !

The major fear in any MIDI setup especially in live work is getting notes stuck on. The Clear List and Safe Switching features of MIDIX go a long way towards preventing this, but if it does happen a Panic Sequence may be sent out to all equipment simultaneously by simply typing **!** or, if the optional front panel Panic Switch is installed, by pressing that. The sequence comprises Stop, All Notes Off on all Channels, then Centre all Pitchbend Wheels, Zero all Modulation Wheels and Reset All Controllers on all Channels and finally every Note Off on all Channels. There are pauses within this massive stream of data to avoid overflowing the buffers in all the receiving equipment.

There is also an OUTPUT PANIC command that sends the same sequence to one specified Output, type:

```
O n ! CR
```

where n is the Output number. Don't forget the carriage return as this is not a single character command.

Hold

The other problem area is MIDI Feedback Loops formed by connecting the output of a merger or sequencer back to its own input. Nothing will happen until a MIDI code is transmitted and then it will circulate endlessly causing the MIDIX Matrix to appear locked. This is because MIDIX has Safe Switching which only changes routings or patches in the gaps between MIDI codes so that data is not corrupted. When a feedback loop occurs there are no gaps. The Illegal Group definitions prevent many direct loops, but several devices might get connected in a loop or equipment external to the matrix, but providing an input, might be feeding back. In this occurance press the optional Hold Switch or type the Emergency Hold command:

```
E
```

all MIDI Inputs will be forced off and the routing may now be changed to a safe state. Press the Hold Switch again or type CR to continue normal operation.

Advanced Use

Messages

There is one final Output command that is, perhaps, the most powerful of them all: OUTPUT MESSAGE. The format is:

```
O n <hh, hh, . . . .hh> CR
```

where n is the Output number and hh are MIDI data in hex. From the start of the opening bracket (<) to the closing bracket (>) the selected Output is switched over to the MIDIX Controller instead of whatever it was connected to. Any amount of data may be sent and when finished the patch is restored.

It is hardly likely that anyone would ever attempt to type anything more than a Program Change or two by hand, so where is all this data originated?

Monitoring

The MIDIX Controller can function in two modes of aquiring MIDI data. It can Monitor any MIDI source and translate the data stream into either HEX or ENGLISH, subject to selectable FILTERING, or it can Capture specific manufacturers' System Exclusive blocks.

All available RAM not used for parameter and system storage is used as a buffer for MIDI input and in the current version is just over 23,000 bytes long. This is equivalent to over 7 seconds worth of solid head to tail MIDI transmission and should be adequate for most situations.

Monitoring is started by typing:

```
I n H
```

for Hex format, where n is the source Input number, or

```
I n E
```

for English. Note that this command starts imediately on the H or E and stops on the next character typed so cannot be given from a file. In Hex format the data is bracketed with < and > ready for saving and converting into an EVENT DEFINE or an OUTPUT MESSAGE type command.

Filters

Every different type of MIDI data may be independently filtered to eliminate unwanted data such as Aftertouch, Clocks, MIDI Time Code or Active Sensing. The filters operate before placing data into the buffer and may be set or cleared by typing:

```
% <hh> n CR
```

where hh is the MIDI Status and n is 0 for pass and 1 for remove. All filters may be cleared by typing:

```
% <0> CR
```

The filters are only removing data from the MIDIX Controller input for selective display. If you need processing of data within the matrix you will require a MIDIC Processor or MIDIY Merger module.

In Hex mode the Input Monitor display is an exact translation of the incoming data showing every nuance of Running Status and Undefined data types. This makes it possible to diagnose obscure fault conditions of equipment incompatibility due to manufacturers either misunderstanding the MIDI specification or simply being incompetent. The display has a new line inserted at every Status change to improve readabilty, but is otherwise in the exact form received. If you are receiving large amounts of data at high rates, like MTC, eventually the buffer will be overloaded and the data lost, but this will appear obvious. Start the Input Monitor command again to reset the buffer.

In English monitor mode only Pitchbend and Song Position Pointer, which are both two byte 14 bit data types, are INTERPRETED into a single decimal number. All other data is still a strict translation and is in Decimal except for System Exclusive and MIDI Time Code data which is more readable in Hex.

System Exclusive Blocks may be captured with the Input Exclusive command or the MIDI Recall comands.

Input Exclusive

The Input Exclusive command is intended for equipment where the bulk dumps may only be initiated manually. To enable capture type

```
I n <hh>
```

where n is the source Input number and hh the manufacturer's ID in hex. All current IDs are listed in the Reference section, but you may use 0 for don't care or to accomodate the new three byte IDs. The MIDIX Controller will switch over the matrix and wait for the next block with the specified ID. All front panel controls and commands are inoperative until the block is completed to allow time to move to the source equipment and manually persuade it to transmit its System Exclusive data. This command may be aborted with ESC and so should not be driven by a file as it will read all following characters looking for an ESC.

MIDI Recall™

The two Recall commands are similar, one may use an Event to request a System Exclusive Block and the other uses a Hex list. To execute an automatic Capture type:

```
        R n <hh> m <hh, hh, . . . hh>
or      R n <hh> m #nn
```

where n is the source Input number, hh is the ID, m the Output number to transmit the request on and ‹hh etc› is the request in hex or #nn the Event number containing the request. Neither command may be aborted once completed, but they will time out if a reply is not received within about 30 seconds.

The next block captured will be formatted as an Output Message command using Output m. All that is necessary to do to save and restore a complete setup is to have one file containing all the Recall commands for each piece of equipment and transmit it with reception enabled. The resulting data is already in the correct format for retransmitting as Output Message commands.

The saved file may be edited with a word processor and different dumps cut and pasted together. The System Exclusive data within the < > brackets should not usually be altered without careful consultation of the manufacturers documentation as the data often contains CHECKSUMS that require a hexadecimal calculator to compute.

The necessary information to construct System Exclusive requests is usually, and should be, provided within the equipment manual. If this is lacking in any way Hinton Instruments will obtain and provide this information ready for use upon request.

Comments

Now that all major operations performed by MIDIX have been covered and the general style and scope of the command set introduced it should be possible to perform a complete system setup of MIDIX and all connected equipment from one disc file. To make such archived data more understandable MIDIX has a COMMENT command, any line starting with a semicolon (;) is ignored until the CR. Descriptions may be freely inserted within the files provided that they begin with a semicolon. Tabs, spaces, carriage returns and line feeds are also ignored.

Help

On line help may be obtained by typing H and the summary of commands in this manual will be paged onto the screen. This is intended as an aide memoire rather than a complete built in manual. Section 3 gives a detailed description of each command individually and Section 4 provides a reference section on MIDI protocol.

Terminal Modes

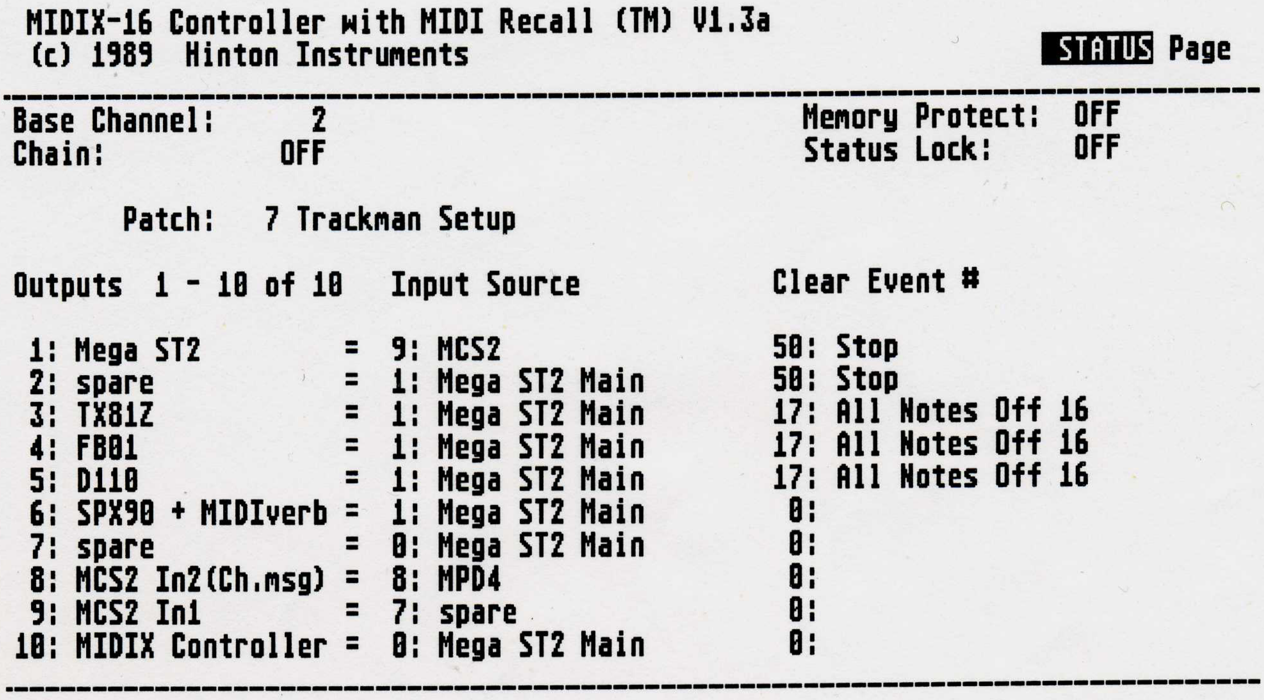
Everything described so far in this section has assumed that MIDIX was connected to a DUMB TERMINAL as this is the power up default state and all the command dialogue has been conducted on a scrolling screen simulating a paper roll. This may be altered with the Terminal command which has two other settings. Typing:

```
T 0 CR
```

will turn off all expansion of commands and is intended for driving MIDIX with dedicated computer software. If your terminal emulator has VT100 capability type:

```
T 2 CR
```

and the screen will change to . . .



```
 MIDIX-16 Controller with MIDI Recall (TM) V1.3a
 (c) 1989  Hinton Instruments                                      STATUS Page
--------------------------------------------------------------------------------
Base Channel:       2                            Memory Protect:  OFF
Chain:            OFF                            Status Lock:     OFF

       Patch:   7 Trackman Setup

Outputs  1 - 10 of 10   Input Source           Clear Event #

 1: Mega ST2          =  9: MCS2                50: Stop
 2: spare             =  1: Mega ST2 Main       50: Stop
 3: TX81Z             =  1: Mega ST2 Main       17: All Notes Off 16
 4: FB01              =  1: Mega ST2 Main       17: All Notes Off 16
 5: D110              =  1: Mega ST2 Main       17: All Notes Off 16
 6: SPX90 + MIDIverb  =  1: Mega ST2 Main        0:
 7: spare             =  0: Mega ST2 Main        0:
 8: MCS2 In2(Ch.msg)  =  8: MPD4                 0:
 9: MCS2 In1          =  7: spare                0:
10: MIDIX Controller  =  0: Mega ST2 Main        0:
--------------------------------------------------------------------------------
```

VT100 Mode

All information on the current state of MIDIX is shown as a series of tabular displays starting on the MAIN STATUS page. The screen is divided into three parts: the HEADER which shows the page name in the top right corner, the DISPLAY WINDOW taking up most of the screen and the COMMAND WINDOW at the bottom. The Command Window behaves exactly as the Dumb Terminal mode and all commands may still be entered here.

When MIDIX is altered by command, MIDI Program Changes or front panel button operation the displays will update to always show the correct current state.

The CURRENT PARAMETER is always shown in reverse video and may be moved around the screen with the Cursor Keys and the Parameter VALUE may be decreased with the minus (–) key or increased with the plus (+) or comma (,) keys. (This makes sense when looking at the application keypad of a VT100 which has a comma instead of a plus.)

When the current parameter is the PAGE NAME the page may be incremented or decremented. To abort any command type ESC twice, if no command has been started this will redraw the page.

MIDIX Commands

Conventions:	{symbol}	= a key on the computer/terminal keyboard
	{CR}	= Carriage Return (Line Feeds are ignored)
	n	= small decimal number, usually 0 to 16
	nn	= decimal number, 0 to 255
	nnnn	= large decimal number
	hh	= hexadecimal number, 00 to FF
	‹ ... ›	= a byte list in hex
	(...)	= a number list in decimal
	" ... "	= an alphanumeric string in ASCII

{BS} and {DEL} both delete backwards

{ESC} aborts the current command

Commands may be entered manually or transmitted from an ASCII file. All Dumps and Recalls are in the correct format for saving and retransmitting with Bulletin Board type communications software. Command files may be edited with a Word Processor.

In VT100 mode the cursor keys move the current parameter selection which is shown in reverse. {+} or {,} increments the selected parameter and {–} decrements it. {.} toggles STOP when the parameter is Input Source.

Single Character Commands:

!	– Global Panic Clear
?	– Show Status
^	– Interogate Version/Revision
[	– Enter Chain
]	– Exit Chain
/	– Advance Chain
\	– Retard Chain
E	– Emergency Hold
H	– Help

Double Character Commands:

DP	– dump all patches
DC	– dump Clear List
DG	– dump Solo Groups
DX	– dump Illegal Groups
D#	– dump Event Lists
D=	– dump MIDI Program equates
D"	– dump Input/Output names
D[	– dump Chain List
D&	– dump everything

Commands terminated with {CR}:

D	– dump current status	
Dnn	– dump single patch nn	
Bn	– set Base Channel	(n: 1 – 16, 0 = Omni On)
Mn	– Memory Protect	(n: 0 = off, 1 = on)
Fn	– Front Panel enable	(n: 0 = off, 1 = on)
Tn	– Terminal type	(n: 0 = off, 1 = Dumb, 2 = VT100)
Wnn	– Wait nn x 100ms delay	
" ...	– Name edit title	
; ...	– Comment	

MIDIX Commands

PATCH CONTROL COMMANDS:

S	– Store current edit in current Patch
Snn	– Store current edit in Patch nn
Lnn(n,n, ... n)	– Load patch nn
P	– Recall/Compare patch
Pnn	– Recall Patch nn
Pnn" ...	– Name Patch nn
PRmm = nn	– equate Patch nn with MIDI Program mm (0 = off)
CH	– show current Chain List
CH[nn,nn, ... nn]	– define Chain List
Gnn	– show Solo Group for Output nn
Gnn(nn,nn, ... nn)	– define Solo Group
Xnn	– show Illegal Group for Output nn
Xnn(nn,nn, ... nn)	– define Illegal Group

EVENT COMMANDS:

#nn	– show Event List
#nn " ...	– name Event List
# nn ‹hh, hh, ... hh›	– define Event List

OUTPUT COMANDS:

Onn	– show input connection
Onn = mm	– select input mm, 0 = stop
Onn " ...	– name output
Onn !	– panic clear output
Onn #mm	– send Event List mm to output
Onn ‹hh, hh, ... hh›	– send MIDI message to output
CLnn #mm	– clear output nn with Event List mm

INPUT COMMANDS:

Inn	– show input name and connections
Inn " ...	– name input
Inn ‹hh›	– show next System Exclusive block with ID = hh, 0 = any
Inn H	– monitor MIDI data in Hex
Inn E	– monitor MIDI data in English
% ‹hh› n	– filter MIDI Status ‹hh› (n: 0 = pass, 1 = filter)

MIDI RECALL™ COMMANDS:

Rnn ‹ID› mm ‹hh, ... hh›	
	– Recall from Input nn the next System Exclusive block after transmitting message on Output mm
Rnn ‹ID› mm #nn	– Recall from Input nn the next System Exclusive block after transmitting Event List nn on Output mm

The MIDIX Controller and Routing matrix may be completely controlled from the RS232 port by either a terminal (dumb or smart) or a computer either emulating a terminal or running a program specific to MIDIX. The commands are given as readable lines of ASCII characters and are structured for entry by hand typing or by transmitting from a pre–prepared text file.

There is one command per line determined by the first character after a Carriage Return {CR}, Line Feeds are ignored, and if this is illegal an error message is given and the terminal bell rung {Control–G}.

There are three types of command:

Single Character commands that take immediate effect
Double Character commands
and Compound Commands ending with a Carriage Return

Apart from the single character commands all commands may be aborted by sending the ASCII Escape character {ESC, 1Bh}, twice if in VT100 mode. It is only necesssary to type the minimum number of characters to make a command unambiguous, e.g. CH for CHain define and CL for CLear output, any following characters will be ignored until a valid argument is entered.

Single Character Commands:

!	– Global Panic Clear
?	– Show Status
^	– Interogate Vers/Revs
[	– Enter Chain
]	– Exit Chain
/	– Advance Chain
\	– Retard Chain
E	– Emergency Hold
H	– Help

The single character commands are all operations requiring a single keystroke for fast manual operation.

! – Global Panic Clear

will immediately switch all modules to the MIDIX Controller and transmit a MIDI Panic sequence comprising All Notes Off, centre Pitchbend, zero Modulation Wheel and Reset All Controllers and 128 Note Offs for all sixteen MIDI Channels to guarantee clearing any stuck notes in any equipment. The sequence is sent as a series of blocks to avoid overflowing the input buffers of the connected equipment and when completed the matrix will revert to its previous routing.

```
        Panic Clear
```

will appear on the screen and the front panel leds will show the operation in progress. This operation may also be triggered by a footswitch or panel button Option.

E – Emergency Hold

is to allow breaking of MIDI feedback loops. This command should only be used to get out of the lockup condition caused by a MIDI data processing unit accidentally having its output connected to its input and continuously recirculating the data. As the MIDIX modules employ a safe switching technique so that the routing is only changed between MIDI data bytes, if there are no longer any gaps then any routing changes will never be executed. If this condition arises the Active LED will come solidly on and other panel or command operations will appear to be inoperative.

When E is typed all matrix routing is disconnected and the cause of the feedback should be dealt with before typing {CR} to resume normal operation. This operation may also be activated by a panel button Option

^– Interogate Vers/Revs

This character returns the program title and software version and revision number followed by the serial number. It is intended to be used by a computer program to check that it is connected to the correct equipment with the correct firmware, but is useful if sharing one terminal between several Controllers.

? – Show Status

Also envoked at power on/reset shows the title, software version and revision, matrix size, number of patches and events, event size and block capture size, memory protection and MIDI receive status and the currently selected patch and title.

Chain Control:

A chain of patch selections set by the Chain Define command may be sequenced by four commands or a footswitch Option.

[– Enter Chain

selects the first patch defined in the chain and enables advance/

retard action. If the first patch is already currently selected then it will be unchanged. The patch number and title appearing on the screen shows that the patch has been changed.

] – Exit Chain

disables further advancing or retarding the chain selection.

/ – Advance Chain & \ – Retard Chain

selects the next patch, forwards or backwards, in the Chain. The chain may be cycled indefinitely until the Exit command.

H – Help

displays several pages of concise descriptions of every available command. Use {ESC} to return to normal operation (twice in VT100 mode) or any other key to continue. The information is duplicated in the summary of commands at the beginning of this section.

Double Character Commands:

D{CR}	– dump current status
D{nn}	– dump patch number {nn}
DP	– dump all patches
DC	– dump Clear List
DG	– dump Solo Groups
DX	– dump Illegal Groups
D#	– Dump Event Lists
D=	– dump MIDI Program equates
D″	– dump Input/Output names
D[	– dump Chain List
D%	– Dump MIDI Filters
D&	– dump everything

All the above commands envoke a partial or complete dump in readable ASCII format. The commands may be given to view the current setups or to save to disc via a computer. When using a terminal emulator with disc filing, enable capture reception before issuing the dump command and then save the capture buffer to disc when the dump (seen on the screen) has finished.

All dumps are in MIDIX command format so that simply retransmitting the file or files back will restore that state. As the dumps are in printable ASCII they may be edited with any word processor and cut and pasted to make new setup files. Comments may be added to the files for future reference provided they are preceded by a semicolon on each line, e.g.

```
;patch dump on Tuesday afternoon{CR}
```

All D commands cause the message

```
;Dump
```

to appear on the screen to acknowledge that the command has been started. The message is preceded by a semicolon to turn it into a safe comment in case the dump is being saved to disc, otherwise it would be interpreted as a command when retransmitted. This line may be deleted when editing files.

When shipped the MIDIX Controller contains "factory presets" of useful events. It is recommended that a complete dump is saved to disc and used as a template to produce new load files.

D{CR} – dump current status

produces a dump of the current matrix connection as a series of

Output commands (see O Command section) with the names of the connected equipment as a comment.

eg
```
O  1 =  7 ;CZ101 to Fairlight III
```
etc.

When reloaded this dump will update and restore the matrix routing and the edit buffer, but will not alter any stored patches.

D{nn} – dump patch number {nn}

requests a specific patch to be dumped in the form of a Patch title command and a Load command (see P and L sections). If the number {nn} is out of range an error message/bell ring will be given and the command ignored.

eg.
```
D9
```
might dump:
```
Patch:   9 "Magic hit formula
L  9 (  1,  2,  3,  4,  5,  6,  7,  8,)
```
By editing the number 9 in the save file this patch could be reloaded in a different number patch location. When reloaded only the patch memory will be updated, the current status will be unaltered unless that actual patch number is currently selected.

DP – dump all patches

as above, but dumps all patches as Name Patch and Load Patch commands. Again, the dumped file may be edited and the patch numbers changed to produce new setups offline. If there are repeated patch numbers in a file the last one will take precedence.

DC – dump Clear List

Each MIDIX Output module may have an Event List associated with it that will be transmitted on that output whenever it becomes disconnected from an input. The Clear List is a set of commands to establish the correspondence of module with Event List.

DG – dump Solo Groups

dumps the Solo Group for each matrix Output as a Solo Group Define command.

DX – dump Illegal Groups

dumps the Illegal patching Group for each Output as a define

Illegal Group command

D# – dump Event Lists

gives all the events as a series of Name Event and Define Event commands.

D= – dump MIDI Program equates

gives 128 Program Equate commands to set up any MIDI controlled patch selection by Program Change.

D″ – dump Input/Output names

will output a list of Name Input and Name Output commands for the matrix size.

D[– dump Chain List

will output a Chain Define command. If there is no chain set up a clear chain command is sent.

i.e. `Chain: []`

D% – Dump MIDI filters

dumps the current state of the filters used by the MIDI Monitor as a series of MIDI Status filter definitions.

D& – dump everything

is equivalent to all of the above dump commands in turn.

Commands terminated with {CR}:

Bn	– set Base Channel
Mn	– Memory Protect
Fn	– Front Panel Enable
Tn	– Terminal type
Wn	– Wait
;	– Comment
"	– Name edit title

Bn – set Base Channel

n may be 1 to 16 to set the MIDI receive channel with Omni off or 0 to set Omni on. Like all commands with a numeric argument once the command is unambiguous all further characters are ignored until either numbers, {ESC}, {TAB} (Control–I) or {CR} is entered. On typing B the screen will show:

```
Base Channel:  n
```

where n is the current setting. If a number is now typed in it will replace the current setting and Delete and Backspace may be used to correct mistakes. When entered, by terminating with a non-numeric character, the number will be checked for being in the valid range. If out of range the terminal bell will ring and any further entry on that command line will be treated as comment. All argument entries follow this style.

Simply typing B{CR} will show the current Base Chanel setting without altering it.

Mn – Memory Protect

n may be 0 to turn protection off or 1 to turn it on. {CR} shows the current state. When set On all commands that alter the stored memory will be treated as errors.

Fn – Front Panel Enable

n may be 0 to disable the front panel manual controls or 1 to enable them. The default state upon reset is enabled. If an optional Status Lock pushbutton is fitted this will light when in the disabled state.

Tn – Terminal type

n may be 0 for off, 1 for on (dumb terminal) and 2 for intelligent terminal (VT100) and is always 1 on power up or reset. When on the MIDIX Controller assumes that it is communicating with a dumb terminal and pads out all responses to make them readable to a human user. When off responses are minimal and terse and intended for communication with a computer program that does not require superfluous information.

The dumb terminal mode is guaranteed to work with any terminal or emulator on any computer and uses only a single scrolling screen.

The intelligent terminal mode uses a subset of VT100 commands that will work on any hardware or software emulation of a VT100 i.e. it does not use dim, underline or double size attributes that are often unavailable. This mode features a number of paged displays for better presentation to a human operator. If going from VT100 mode to another, a VT100 reset terminal command is sent which will clear the screen and may cause some emulators to reload from disc.

Wnn – Wait

Wait for nn 100ms delays, where nn = 0 to 255. This command may be inserted between Output data commands in setup files when using with equipment that has a dead time after just receiving a block of data. If inserted in a Request file it will be transfered to the correct place in the Dump file.

; – Comment

All characters following a semicolon are treated as comments until an {ESC} or {CR}. This may be used to leave notes on the screen

e.g. `;DON'T TOUCH - gone to see a man about a dog`

or to anotate commands within a text file for future reference.

" – Name edit title

Up to twenty characters may be entered after the quote to name the edit buffer. When the current edit is stored (S command) this will become the patch title.

Patch Control commands:

S	– Store patch
L	– Load patch
P	– Recall/Compare patch
Pnn "	– name patch
PRmm = nn	– equate patch
CH [nn,...nn]	– define Chain List
Gnn(nn, ... nn)	– define Solo Group
Xnn(nn, ... nn)	– define Illegal Group

S – Store patch

loads the current matrix routing in the edit buffer and the edit title to the specified patch memory. If no number is specified the current patch selected will be overwritten with any edits. Any changes made by manual front panel controls or with the O= command since the last patch selection will be in the the edit buffer, but the current Solo status (including consequential Stops) is not stored as part of the patch. An error will occur if the patch number is out of range or Memory Protect has been set on.
On typing S

```
        Store:  nn
```

will appear in the screen, where {nn} is the current patch number. A new patch number may be typed in terminated by any non–numeric character, usually {CR}.

L – Load patch

directly loads a patch memory without effecting the current routing or edit buffer. This is a quicker way of loading a set of memories as it does involve setting every individual module in the edit buffer and then storing into a memory.

On typing L the screen will show

```
        Load   1
```

where 1 is the default memory number. A new number may be entered and then all characters are ignored until an open parenthesis "(" followed by a list of module input selections in strict module order ending with a close parenthesis ")".

e.g.

```
        Load   1 ( 1,  2,  0,  0,  0,  6,  7,  8,)
```

"0" means that a module is set to Stop and is the default condition.

An error occurs if a number is out of range. All patch dumps output in this format, but if a dump from a larger matrix is loaded only the relevant number of modules will be accepted and the remainder of the line treated as a comment.

Pnn – Recall patch nn

Sets the matrix routing and the edit buffer to the patched stored in memory nn. On typing P a P will appear on the screen as this is an ambiguous command (see PR command below) until qualified by either an R or a number. As soon as a number is entered, say 4, the screen will show:

```
        Patch:   4
```

which may be further entered or deleted, upon typing {CR} the screen will display on the following line:

```
        Patch:   4 "Patch 4 title
```

The first line, entering the command, is a request to select patch 4. The second line is an acknowledgement that the patch has actually been updated. If the same patch were already selected there would be no response to the request. Patch acknowledgements will also appear on the screen when the patch is changed from different source:, MIDI Program Changes, optional front panel/remote manual control or by Chaining.

P – Recall/Compare patch

If a patch has been edited, either manually or by using the O= command, entering P with no argument and typing {CR} will cause toggling between the routing in the edit buffer and the memory store. Extending the previous example, typing "P{CR}" three times after editing would produce on the screen:

```
        Patch:   4
        Recall Patch:   4 "Patch 4 title
        Patch:   4
        Edit Patch:   4 "Patch 4 title
        Patch:   4
        Recall Patch:   4 "Patch 4 title
```

This action will also result from MIDI or Chained patch changes if the edit buffer has been altered. Starting editing when in the Recall state will loose the previous edit.

Pnn " – name patch

Directly sets the title of the specified patch number to be up to twenty characters following the quote terminated by a {CR}. If more than twenty characters are entered they will be ignored. To produce the title in the above examples enter:

```
        P4 "Patch 4 title{CR}
```

the screen will actually show:

```
        Patch:  4 "Patch 4 title
```

PRmm = nn – equate patch

Sets the equivalence of MIDI Program Change {mm}, numbered 1 to 128 and an internal patch number {nn} with 0 meaning no action. After typing the = the current setting will be shown and entering a {CR} at this point will make no change otherwise a patch number or 0 may be entered.

MIDI Program mapping should be used with caution as selecting some patches may disconnect the source of Program Changes from the Controller MIDIX module. When shipped all Program equates are set to 0 for this reason.

MIDI Program Changes are recognised on any module input selected by the Controller module on the Base Channel or all MIDI Channels as set up by the B command.

CH [nn,...nn] – define Chain List

C is an ambiguous command (see CL below), qualifying with H will produce on the screen:

```
Chain define:
```

all characters are then ignored until an open bracket ([) and then a sequence of patch numbers may be entered terminated by a close bracket (]). The patch numbers may be in any order up to a maximum chain length of fifty. The patches in the chain will be sequenced when using the [,], SPACE and BACKSPACE commands detailed above. Tabbing will automatically increment the patch numbers in the chain. eg. entering:

```
CH[6{TAB}{TAB}{TAB}14{TAB}{TAB}]{CR}
```

will result in:

```
Chain define: [  6,  7,  8, 14, 15,]
```

Note that the first {TAB} each time delimits the patch number. A chain may be cleared by entering the close bracket as the first step i.e. `CH[]`

Gnn(nn, ... nn) – define Solo Group

A Solo Group is a list of all the module numbers that will be Stopped when a certain module's SOLO button is pressed. The soloing action for each MIDIX module may be defined independently.

On typing G the response will be:

```
Solo Group: 1
```

the argument is the MIDIX Output module number which may be changed at this point. To show the current group type {CR}.

```
Solo Group: 1 ()
```

shows that no soloing is associated with module 1, whereas

```
Solo Group: 1 (  1,  3, 10, 11)
```

shows that three other modules are associated in a group of four.

Module 1 appears in its own solo group, but this will have no effect it simply makes setting up groups easier to understand.

To define a solo group enter within parentheses a list of the module numbers, not necessarily in numeric order. An empty or null list disables the Solo button on a particular module.

Xnn(nn, ... nn) – define Illegal Group

An Illegal Group is a list of all Inputs that cannot be connected to a certain Output. On typing X the screen will show

```
        Illegal Group:  1
```

and, like all commands, the argument may be altered and the current state shown or redefined.

A null list enables a module to be connected to any input to the MIDIX matrix:

```
        Illegal Group: 1 ()
```

MIDI devices that should always be prevented from being connected to themselves include Sequencers and Mergers. A less obvious example is an Expander Module that may get confused if it receives a System Exclusive block that it is transmitting.

It is also good practise to limit MIDI Clocks and MIDI Time Code to only those devices that actually require them. Although the MIDI specification states that all devices should ignore all MIDI codes that are not used, most common equipment has poor handling of this type of data. If a SMPTE timecode to MIDI Clock/MTC converter is being used, include the Input number in every Illegal Group list except the computer or sequencer and any automation devices.

Event Commands:

#nn {CR} – show Event List
#nn " – name Event List
#nn ‹hh,...hh› – define Event List

#nn {CR} – show Event List

Typing "#" will produce on the screen:

```
        Event #:    1
```

the event number may now be entered terminated with a {CR} and if within range the event title will be shown following on the same line and the event list data on the next line.

```
Event #:   1 "All Notes Off 16
#   1 <B0,7B,00,B1,7B,00,B2,7B,00,B3,7B,00,B4,7B,00,B5,7B,00,B6,7B,00,B7,
      7B,00,B8,7B,00,B9,7B,00,BA,7B,00,BB,7B,00,BC,7B,00,BD,7B,00,BE,7B,
      00,BF,7B,00,>
```

This command is intended for use by an operator to check the contents of an Event List, the first line should not be saved to disc as it could contend with the E command. Use D# to dump events, see above.

If the event number is not delimited with a {CR} then the event title or the event data may be entered with the following commands:

#nn " – name Event List

Up to sixteen printable characters may be entered following the quote, further characters or trailing comments will be ignored. Delete and Backspace both delete backwards and the string may be aborted by typing {ESC} or entered by typing {CR}.

#nn ‹hh,...hh› – define Event List

After typing the **<** character up to fifty hexadecimal codes may be entered that will constitute that Event List. {CR} will not end this command until the hex entry is ended by typing **>**. Leading Spaces, {TAB}s and {CR}s will be ignored so that an event list may go over more than one line in a file.

The data defining an event list will be used whenever that event is invoked for clearing on patch changes, presetting outputs or recalling system exclusive blocks. The validity of the MIDI protocol within the event list is not checked either on entry or use so the onus is on the user. The Controller is shipped with a variety of useful events as "factory presets" which may be dumped to disc and used as the basis for editing into new events. All MIDI equipment should include all the MIDI codes it recognises in its manual if you experience any difficulty translating this data into events lists or require any information not given by the manufacturer please contact Hinton Instruments. The Reference section gives a synopsis of MIDI codes.

O – Output Commands:

Onn	– show input
Onn = mm	– select input/stop
Onn "	– name output
Onn !	– panic clear output
Onn #mm	– send Event List mm to output
Onn ‹hh,...hh›	– send MIDI message to output

Onn – show input

Typing O will produce on the screen:

```
Output:   nn
```

where {nn} is the default of the last MIDIX module addresed by a command. A different number may be entered or if the default is correct simply type {CR} and the source and destination names will be shown following as a comment. e.g. typing

```
01 CR
```

might result in

```
Output:  1 =  7 ;CZ101 to Fairlight III
```

if Output 1 were named "Fairlight III" and Input 7 were named "CZ101. If that module were in the Stop state

```
Output:  1 =  0 ;CZ101 to Fairlight III
```

would result with the names showing the connection that would be made when Stop is changed.

Onn = mm – select input/stop

This command is equivalent to the use of the Select and Stop front panel controls. After entering the output module number, if = is typed the current input selected will be shown. This may be overtyped with the new input number (1 to 16) or 0 to Stop and on entering {CR} the new connection will be shown. e.g. typing

```
07 = 1 CR
```

might result in

```
Output:  7 =  1 ;Fairlight III to CZ101
```

Onn " – name output

Up to sixteen printable characters may be entered following the quote, further characters or trailing comments will be ignored. Delete and Backspace both delete backwards and the string may be aborted by typing {ESC} or entered by typing {CR}. This name will be used in displaying all connections with this output.

The remaining Output commands all transmit MIDI data from the specified MIDIX module. After typing the qualifying character (**!**, **#** or **<**) following the Output number as above, the Output will be switched to the MIDIX Controller, normally the last module in a system. Any other modules already connected to the Controller will be put into the Stop state before any MIDI data is sent. At the completion of the command the patch will be restored.

Onn ! – panic clear output

Transmits a Panic Sequence as in the Global Panic Clear command (!), but only on the specified Output. This command needs a {CR} before entering the next instruction.

Onn #mm – send Event List mm to Output nn

Transmits an Event List to the current Output. This may be used for a specific initialisation, request or preset.

Onn ‹hh,...hh› – send MIDI message to output

A free format version of the above command. Each MIDI byte is transmitted when delimited so that {Delete} and {Backspace} only function within the currently entered byte. {CR} will not end this command until the hex entry is ended by typing ›. Leading Spaces, {TAB}s and {CR}s will be ignored so that data may go over more than one line in a file. The patch is restored at the closing ›.

e.g. `05 <90,3C,40,3C,0,>` CR

will transmit a middle c Note On and Off to a unit connected to Output 5.

`03 <C0,0,C2,5>` CR

will transmit Program Change 1, MIDI Channel 1 and Program Change 6, MIDI Channel 3 to two pieces of equipment connected to Output 3, one THRU the other.

There is no limit to the number of bytes transmitted with this command so it may be used to send large ammounts of system exclusive data. All Recall commands (see below) save data in this format. A series of this command in a file for all modules in a system can completely (or partially) configure all connected MIDI equipment in a setup.

CLnn #mm – clear output {nn} with events list {mm}

Each MIDI Output channel may have an event associated with it that will be sent to the connected equipment before any patch changes that would result in that equipment becoming disconnected from a source take place, i.e. selecting a new Input, Stopping or Soloing another module.

On typing CL (C is ambiguous) the screen will show

```
        Clear output:  nn
```

where {nn} is the last module addressed. On entering or defaulting the module number the screen will show

```
        Clear output:  nn with Event #: mm
```

where {mm} is the event number already set up for that output channel, if mm = 0 no event will be used. A new event number may be entered or {CR} will leave the setup intact.

This action is not only required by keyboards and expanders to prevent droning notes. Effects units and MIDI data processors that map different controllers to various functions may also require setting to a safe state.

I – Input commands:

Inn	– show name and connections
Inn "	– name input
Inn ‹ID›	– show next System Exclusive block from input nn
Inn H	– monitor MIDI data in Hex
Inn E	– monitor MIDI data in English
% ‹hh› n	–filter MIDI Status ‹hh›

Inn – show name and connections

Typing I will produce on the screen:

```
        Input:  nn
```

where {nn} is the default of the last MIDIX module addresed by a command. A different number may be entered or if the default is correct simply type {CR} and the source and destination names will be shown following as a comment. e.g. typing

```
        I7 CR
```

might result in

```
        Input:  7 ;CZ101 not connected
```

or

```
Input:  7 ;CZ101
;         to  1 Fairlight III
```

with any further connections shown on following lines.

Inn "   – name input

Up to sixteen printable characters may be entered following the quote, further characters or trailing comments will be ignored. Delete and Backspace both delete backwards and the string may be aborted by typing {ESC} or entered by typing {CR}. This name will be used in displaying all connections with this input.

Inn ‹ID› – show next System Exclusive block from input {nn}

This command really belongs in the next group, the MIDI Recall commands, but is a special case for capturing System Exclusive dumps from older equipment where the dump could only be manually initiated, the most obvious example being the original DX7. After qualifying the input number with a < the manufacturers ID should be entered in hex (a list is included in the reference section). An ID of "00" may be used for any ID.

The next single system exclusive block received from the specified Input Channel will be captured (up to the maximum capture block size) and then shown formatted on the screen in ASCII hex within " < " and " > ". This data may be saved to disc, but will require an "O{nn}" entered before the " < " to convert it into a send MIDI message command or a "#{mm}" to convert it into a define event command.

Once started only receiving the block or typing {ESC} to abort will end this command, {CR} will be ignored.

MIDI Monitor

Inn E – monitor MIDI data in English
Inn H – monitor MIDI data in Hex
‹hh› n – filter MIDI Status ‹hh›

The MIDIX Controller can monitor any MIDIX Input and interpret the MIDI data into either hexadecimal or English on the terminal screen. Any MIDI Status and its Data may be selectively filtered to make the display easier to comprehend.

The commands function slightly differently between the dumb and intelligent terminal modes as there are also dedicated VT100 pages for monitoring and filtering.

Inn E – monitor MIDI data in English

If an E is typed after setting an Input the screen will show:

```
Input:  1 English Monitor
```

and any MIDI codes received on that Input will be displayed until another key is pressed.

If a keyboard is connected to the MIDIX input and a program change, slight pitchbend movement and slight modulation wheel movement are made the screen should show something similar to:

```
Ch1:Program:  8,
Ch1:Pitchbend:+  260,
Ch1:Pitchbend:    0,
Ch1:Controller:  1/Modulation,  1,  1/Modulation,  0,
```

showing that the data is transmitted on MIDI Channel 1 and that the Pitchbend is transmitted with Running Status off while the Controller has Running Status on.

Pitchbend and Song Position Pointer which both have two byte, 14 bit data are shown with a single decimal number, with Pitchbend centre as zero. All other data types are shown as transmitted. Please refer to the Reference Section for details of MIDI data structure.

Inn H – monitor MIDI data in Hex

If an H is typed after setting an Input the screen will show

```
Input:  1 Hex Monitor
<
```

and any MIDI codes received on that input will be displayed as hex bytes until another key is pressed. Note that the translation is preceeded with a **<** and terminated with a **>**, this is for use in defining Events and Output Messages by recording actual input data.

Using the previous example, if the same data were sent the screen would now show:

```
C0,08,
E0,04,42,
E0,00,40,
B0,01,01,01,00,
```

which is exactly the bytes transnmitted. A newline is inserted at every Status change to improve readability. Every nuance of the MIDI protocol is displayed as it is often that violations of this cause compatibility problems between equipment.

% ‹hh› n – filter MIDI Status ‹hh›

R – MIDI Recall™ commands

Rnn ‹ID› mm ‹hh,...hh›
Rnn ‹ID› mm #nn

Rnn ‹ID› mm ‹hh,...hh›

Recall from Input {nn} the next System Exclusive block with ID after sending on Output {mm} the MIDI message ‹hh,...hh›. On typing R the screen will show

```
;Recall from Input:  nn
```

where {nn} is the last module addressed. The command has been preceded with a semicolon turning the line into a comment so that it is safe to store to disc without being interpreted as a command again when reloaded. The input channel may now be entered for the equipment that the dump is expected on. The system exclusive ID is then entered in hex (a list is included in the reference section). An ID of ″00″ may be used for any ID. The screen may then show:

```
;Recall from Input: nn <ID = 0> after sending Output: nn
```

The output channel will default to the same number as the input channel, but if the equipment connected to the MIDIX routing matrix is not on a one to one basis a different number may be entered.

The command may now be qualified with a free format ASCII hex string containing the system exclusive dump request data. Files containing complete requests for various MIDI equipments are available from Hinton Instruments either on Atari ST 3.5″ discs or via modem.

Any outputs connected to the Controller will be placed in the Stop state before sending the MIDI message. Any outputs connected to the dumping equipment will not be Stopped to allow programmed transfers of data between different units using the matrix.

The next single system exclusive block received from the specified Input Channel will be captured (up to the maximum capture block size) and then shown formatted on the screen in ASCII hex within ″ **<** ″ and ″ **>** ″ preceded by an ″O{mm}″ to make it into a send MIDI message command. It is assumed that the block will be sent back to the same output as the request was transmitted on, but may obviously be edited off line for different use.

NB system exclusive blocks nearly always include checksums so are not suitable for editing in ASCII format. As long as everything between an "F0" and an "F7" is kept intact different system exclusive blocks may be cut and pasted in the stored files.

Once the MIDI message is terminated with a " > " the screen will show

```
... > Waiting ...
```

and will time out after approximately 30 seconds if nothing is received and ring the terminal bell. This is the one situation where typing {ESC} will not abort a command which is necessary to ensure correct operation from Recall commands prepared in files.

Rnn ‹ID› mm #nn

This is very similar to the previous command except that one of the defined Events Lists may be used instead of the MIDI message. Some useful dump requests are included amongst the "factory preset" Event Lists.