THE COMPLETE SCI MIDI
First Edition

Contents

Sequential Circuits, Inc.
Nijverheidsweg 11c
3641 RP Mijdrecht, Netherlands
02979-6211
TELEX: 12721 SQNTL NL

Sequential Circuits, Inc.
3051 North First Street
San Jose, CA 95134-2093
408/946-5240
TELEX: 364412 INTR 706

THE COMPLETE SCI MIDI
First Edition

Edited by Stanley Jungleib

Document Number: MIDI-3
Issued: January, 1983
Revised:

Document No: MIDI-1
Issued: January, 1983
Revised:

MIDI HISTORY

Stanley Jungleib, SCI

Introduction

The Musical Instrument Digital Interface (MIDI) is a specification which enables manufacturers to design equipment that is basically compatible. This is most beneficial for the owner, whose equipment is thereby protected from obsolesence. As MIDI-compatible equipment is introduced, one will be able to freely choose keyboards, sequencers, and rhythm units from a variety of manufacturers with confidence that they will work together as one programmable system through which complete pieces can be composed and realized.

The problem of instrument compatibility is not new. It can be probably said of any two keyboards, that someone has desired if not actually tried to interconnect them. Keyboard couplers were developed for both pipe organs and harpsichords. In the heyday of electric organ technology this interest occasionally led to the installation of thick cables for wiring keyboards in parallel. The first synthesizers were easier to interface, because of the nature of modular equipment. However modules from different manufacturers might have incompatible control voltage, trigger, gate, and output levels or polarities. These differences have been promulgated in scores of synthesizer, keyboard, and effect devices, ultimately giving rise to an entire industry devoted to modifications and interfacing. And though they provide the best opportunity for interface so far, even microcomputer-based synthesizer equipment has been developed along independent, incompatible lines.

Like many other defacto "standards," the MIDI has arisen primarily from the activities of those concerned that the incompatibility of current equipment discourages wider availability of the kinds of complex systems which can be envisioned utilizing even current technology. (The S-100 microcomputer buss evolved for similar reasons.) It is more than anything else the advent of the home computer which has forced music manufacturers to finally address the issue of compatibility. For the musician, the keyboard interface to the computer terminal offers the possibility of multi-track sequencing and editing, score display and printing. In this light the usefulness and need for a standard computer keyboard interface is obvious. Only with some such standard can these musical tools be developed.

The following explains how the MIDI specification resulted from this industry-wide consensus. The MIDI specification neither possesses nor claims any authority over equipment design. Rather, it is merely an informal agreement on some simple interface circuitry and the "grammar" of a non-proprietary language which can carry meaningful information between instruments. The incorporation or support of the MIDI facility in a product remains entirely a decision for each manufacturer.

The SCI Digital Interface

SCI first became interested in microcomputer interfacing in conjunction with the design of the Prophet-10 polyphonic synthesizer and its internal polyphonic sequencer. The Prophet and its sequencer each were based on Z-80 microcomputers. To record, as notes were played, every few milliseconds (at a rate set by the sequencer clock), the Prophet would send its complete keyboard "status" to the sequencer. The sequencer had to figure out which notes were going on and off, and record these events in reference to the clock count. On playback, the sequencer computer also sent the complete keyboard status every clock pulse, with events as counted out by the clock. The Prophet would play these notes just as if they came from its own keyboard. Later, this sequencer was made available as an accessory for the Prophet-5. The Prophet-5 Remote Keyboard was also developed which used this interface. SCI published the data protocol upon which this interface was based, in the hopes that the programming public would be encouraged to develop their own interfaces for the Prophet-5.

This did not occur, apparently because in being conceived for a specific application, the interface was very fast but too clumsy for general-purpose use. It was criticized as requiring too much programming "overhead," in the constant transmission of meaningless keyboard information. As a result of this experience, SCI resolved to pursue a more streamlined interface that would be easier for programmers to work with.

The Universal Synthesizer Interface

In the meantime, occasional discussions between the presidents of Sequential Circuits (SCI), Oberheim Electronics, and Roland (Dave Smith, Tom Oberheim and Ikutaroo Kakehashi) also revealed a shared interest in the interface problem and development of an interface widely acceptable to the industry.

Smith then outlined a specification for a "Universal Synthesizer Interface" (USI). It was developed with the assistance of SCI's Chet Wood and presented at the Fall, 1981 convention of the Audio Engineering Society (AES).

The USI differed markedly from the earlier SCI Digital interface in that rather than being polled at the sequencer clock rate, information was only sent when an event actually occured--for example, a note going on or off. The USI was proposed to be serial, operating at 19.2 kBaud, with TTL levels, and connected through phone jacks.

After incorporating changes in response to comments from AES, Smith sent a questionnaire to all manufacturers and industry consultants he could find, asking for their suggestions and any special requirements. There was a strong response to this initiative; some saying, for example, that it would not be possible to do it serially, that a parallel interface was necessary. Others thought the proposed serial speed too fast for operation with home computers. Many other issues were raised.

All respondents were invited to a conference in coincidence with the January, 1982 Western National Association of Music Merchants (NAMM) convention in Anaheim. This meeting was attended by representatives from SCI, Roland, Oberheim, CBS/Rhodes, Yamaha, E-mu, Unicord (Korg), Music Technology Inc., Kawai, Octave Plateau, Passport Designs, and Syntauri. Other manufacturers seemed to be maintaining a "wait-and-see" policy.

At this meeting the chief changes which occured to the USI were to add optoisolation to prevent audio ground loops, and to increase the speed to 31.25 kBaud.

The Japanese Interface Proposal

Following the USI discussion at Anaheim, an alternative specification was presented by some of the Japanese companies which had grown out of their own research. Whereas the USI was basically content to specify note on/off codes, this new proposal went on to define many more complex operations. It also offered a different data structure, with status and data bytes being flagged by bit 7 (1=status, 0=data). This greatly simplified the protocol by eliminating all the checks which were otherwise needed to distinguish the data category. With the most significant bit now defined as a "flag," data is thereby limited to 7 bits, but this is sufficient for most synth data, and when not, can simply be sent as multiple 4-bit nibbles.

The MIDI

After the Anaheim meeting, Smith and Wood integrated the USI and Japanese proposals, forming the first MIDI specification. This was sent to all of the meeting participants but, curiously, provoked no further comment from this continent. The final document was therefore arrived at after several exchanges betwen SCI and Roland, which is serving as liason with Yamaha, Korg, and Kawai.

The development of MIDI was first made public by Robert Moog, in his October, 1982 column in KEYBOARD magazine.

In December of 1983, SCI began shipping the Prophet-600, the first commerically available instrument to include the MIDI.

Document No: MIDI-2
Issued: January, 1983
Revised:

MIDI FUNDAMENTALS

Stanley Jungleib, SCI

INTRODUCTION

The Musical Instrument Digital Interface (MIDI) specification was recently worked out as a cooperative effort by several synthesizer manufacturers (see "MIDI HISTORY"). Intended as an introduction, this article includes all information contained in the specification, surrounded by more context and explanation. Once you become familiar with MIDI, you may prefer working with the actual "MIDI SPECIFICATION."

The purpose of the specification is to enable the easy integration of synthesizers, other electronic keyboards, sequencers, drum boxes, and home computers from various manufacturers, into one programmable system. In being made compatible with forseeable microcomputer technology, the useful lifetime of the musician's equipment is thereby multiplied. The realization of complex electronic-assisted music hitherto reserved for well-financed professionals becomes more widely available. For example,

> Synthesizers can be easily configured "in parallel," with instruments played simultaneously or remotely.
>
> Entire compositions, consisting of monophonic and polyphonic sequences and rhythm, can be played at one touch.
>
> The computer terminal can be used for composing, sequence creation and editing.
>
> Graphic-quality printers can print the "hardcopy" manuscript of an improvisation or composition.
>
> Video synthesis can be integrated with music synthesis.
>
> Those parts of musical education requiring drill, e. g. learning to read music, scale recognition, and ear training, can be automated. This frees the teacher's time to concentrate on technique.

HARDWARE

To simplify cabling between instruments, the interface is serial. It operates at 31.25 kBaud (thousand-bits-per-second), asynchronous. This is considered a high speed for serial operation--in comparison to the typical RS-232 maximum of 19.2 kBaud--and was chosen to prevent objectionable delays between equipment. The 31.25 kHz clock can also be easily obtained from hardware, for example, by dividing 1 Mhz by 32. One serial data byte consists of a start bit, 8 data bits (DO to D7), and a stop bit--for a total of 10 bits transferred in 320 microseconds (us).

Physically, MIDI appears as two or three jacks on the instrument. See Figure M2-1, the hardware schematic. The connectors are DIN 5-pin (180 degree) female panel mount receptacles (SWITCHCRAFT 57GB5F or equivalent). DIN connectors were agreed to by U. S. manufacturers because it was felt that DIN connectors are now widely available here. However the specification does provide that a manufacturer can use XLR connectors, if the firm makes available all necessary conversion cables.

The two required jacks are MIDI OUT and MIDI IN. The transmitter data typically originates in the instrument's UART. The interface circuit is a 5-mA current loop, designed especially to prevent the formation of audio ground loops which often develop in complex systems. The output is normally meant to drive only one input. If transmit data is low (0), current flows from Vcc (+5V) through Ra, over pin 4 of both connectors, through the opto-isolator, returns over pin 5, then through Rc. The opto-isolator output is normally pulled high by Rd. However when current flows through the internal LED, the isolator output switch turns on, grounding Vo, thus sending a low to the receiver UART. When data is high, the LED does not light. The receiver UART therefore sees a high. D1 protects the opto-isolator from reverse-polarity currents which may result from transmitter anomalies.

Interconnect cables should not exceed fifty feet (15 meters), and must have a corresponding 5-pin DIN male plug (SWITCHCRAFT 05GM5M or equivalent.) The cable should be shielded twisted pair, with the shield connected to pin 2 at both ends. Notice that while the MIDI OUT jack is grounded to the instrument chassis, MIDI IN is not. This allows the cables to provide their shielding services without creating ground loops.

The optional third jack, MIDI THRU, provides a direct copy of data coming in MIDI IN. It is included when the manufacturer intends the instrument to operate in a "chain" or "loop" network, as opposed to a "star" network. This question provides a convenient seque into the topics of modes and channels.

MODES AND CHANNELS

The first thing to realize about MIDI is that the total control features available still depend on the design of each specific piece of equipment. MIDI does not magically transcend equipment limitations or differences. Rather it merely enables them to "communicate" at their "least common" level. For example, specific programmed sounds can't be transferred directly between different models of synthesizers because of inherent design differences, but keyboard information and program selections can be communicated.

Figure M2-1
HARDWARE SCHEMATIC

One of MIDI's design goals was to be simple enough so that you could connect any polyphonic synthesizer to any other, or to a sequencer, and at the very least the notes would be correctly played or stored. This would be possible with virtually no other action on the part of the user. Above this minimum, each instrument may or may not include further facilities for complex control options.

Each type of equipment has different minimum requirements. For synthesizers, minimal usefulness seems to include remote keyboard control and program switching. While polyphonic sequencers send and receive keyboard data, they may or may not be interested in program changes. Monophonic sequencers can only deal with individual lines, so keyboard data must somehow be different for them. Drum units don't usually care about specific keyboard notes, but may need to synchronize to their timing, or to the sequencer, and perhaps react to program changes as well.

While most of these requirements and useful control options can be foreseen, the number of possible interconnections cannot. Therefore while the specification says that each transmitter will drive one and only one receiver, provision has been made so that any specific instrument or synthesizer voice on the MIDI bus can be addressed, regardless of the interconnection scheme. This is accomplished by assigning up to 16 channels under increasingly powerful (and complex) modes.

Each unit connected to the MIDI bus has separate transmit and receive ports. There are three modes of operation for transmitters and receivers: Omni, Poly, and Mono. Omni mode is the most general level of operation, interfacing all units. Poly mode allows each unit (synth, sequencer, or drum box) to be addressed separately. Mono mode is the most specialized, allowing individual addressing of (for example) each synthesizer voice.

Normally, transmitters will periodically send out a Mode Select command for the most powerful mode to which they can be configured. However, the actual data transmitted will be in the mode to which a second transmitter may have switched the receiver. For example, Synth A by default transmits in Omni mode to Synth B. Synth B, being capable of Poly mode operation, periodically transmits Poly Mode Select codes to Synth C. But the data sent from Synth B to C will be in Omni format (because Synth B's receiver is constantly getting Omni Mode Select commands from Synth A). Synth C may or may not respond to the Poly Mode Select commands from Synth B, because if a receiver is capable of operating in the requested mode, it switches to that mode. Otherwise it ignores the Mode Select command. (By the way, the Mode Select commands double as "All Notes Off" commands, therefore can only be sent while all notes are off, or when it is desired to turn all notes off.)

Omni Mode

At power up or reset, all instruments default to Omni mode. See Figures M2-2 and M2-3. Regardless of the system configuration, Omni transmitters always send polyphonic data on Channel 1. Omni receivers respond to Note On/Off Events sent over any channel (1-16). These notes are handled according to the internal assignment scheme of the synthesizer. So this configuration allows any number of polyphonic synthesizers to play in parallel, as soon as they are interconnected.

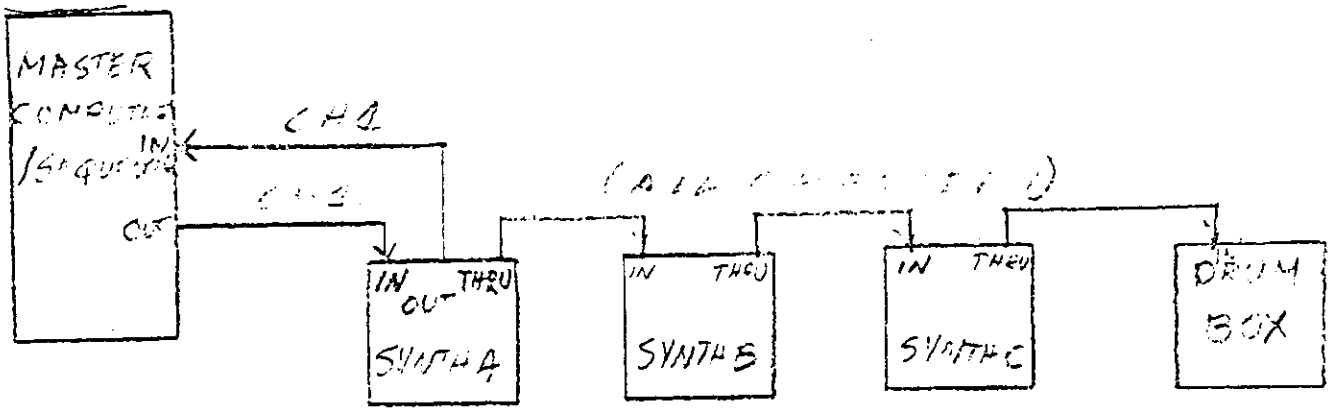


Figure M2-2
OMNI MODE CHAIN NETWORK

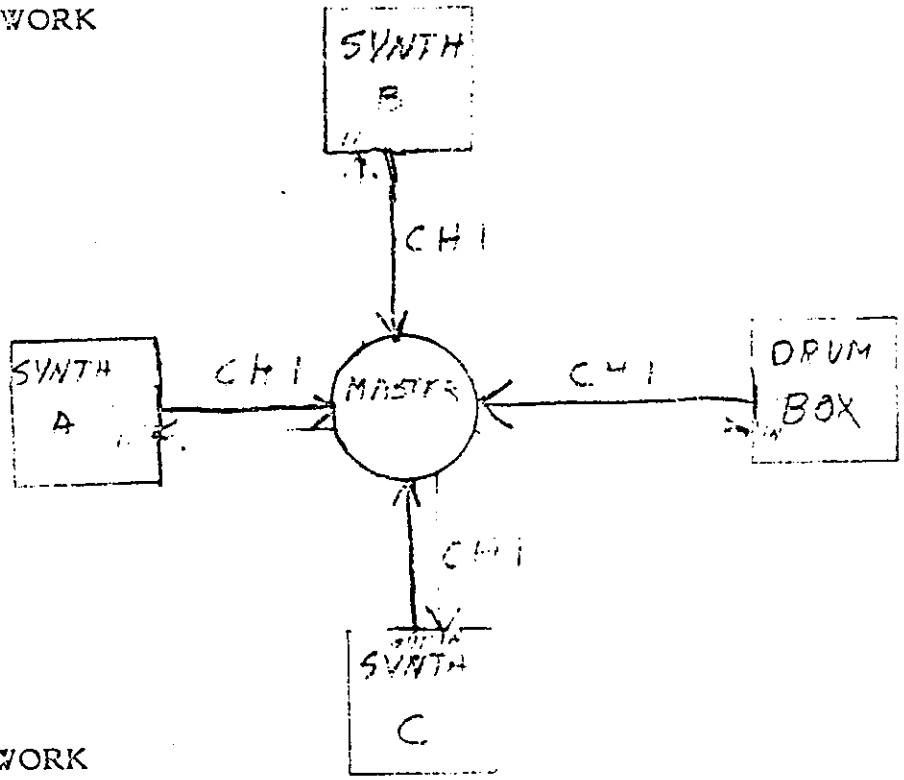


Figure M2-3
OMNI MODE STAR NETWORK

A receiver's mode can only be changed by a Mode Select command transmitted in the channel(s) to which it is currently assigned. If the receiver is not capable of operating in the requested mode, it ignores the Mode Select command. No unit may switch its own modes. Even though a receiver in Omni mode receives in all channels, it will respond to Mode Select commands in only one channel: the one to which it is assigned.

Receivers and transmitters without channel selection capability are always assigned by default to Channel 1.

Poly Mode

Omni mode addresses all units with the same data. Poly mode allows individual addressing of each unit. In other words, the master controller can send separate parts to each synth, whereas in Omni mode they all played the same part.

As shown in Figure M2-4, the master controller in the chained network sends all commands, which are encoded with their destination channel number, over one line. This requires each unit include an address selector switch to define its channel of operation.

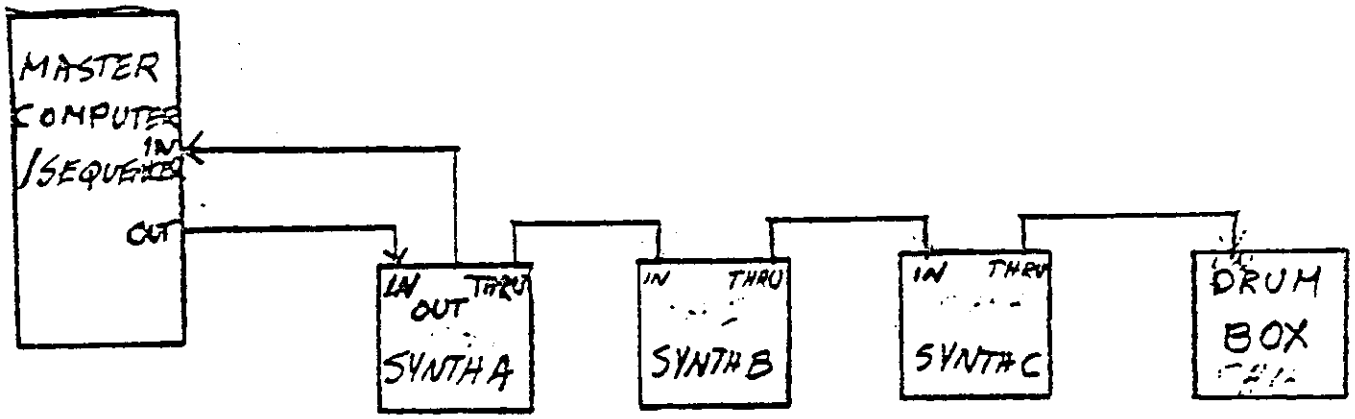


Figure M2-4
POLY MODE CHAIN NETWORK

The channel definitions having been made, the master controller must issue the command to the receiver on that channel to switch to Poly mode. Thereafter, the receiver listens for keyboard data encoded with its channel number. Any number of notes can be sent, to which, again, the polyphonic synth will respond according to its own priorities.

Poly mode will be useful for sequencing multi-part arrangements of standard synths, for example, which can't be done in Omni mode.

Mono Mode

When a synthesizer has Mono capability, and it receives a Mono Mode Select command, it configures itself to receive on the channel it is assigned to and above, up to the number of voices it has. For example, the Prophet-T8 in Mono mode will transmit and receive on Channels 1 - 8. (Future synthesizers could contain more elaborate channel selection capability.)

Channeling each voice provides fast transfer of individual pressure (also called "after touch") data for each key. It also makes true legato possible, because the note value (=voice pitch) can be changed without having to first turn the note off (as in Poly mode).

The data format for the specific codes which control the modes and channels can now be presented.

DATA FORMAT

There are five categories of MIDI data: Channel, System Common, System Real Time, System Exclusive, System Reset.

Each data category encompasses a number of "status bytes" which define specific commands under that category, and which precede data bytes which specify the exact operation. Status bytes are distinguished from data bytes according to whether the most-significant (MS) bit is set (1=status) or reset (0=data). The status bytes under each category are defined below. Note that any data sets (e.g. Note On event data) which are sent successively under the same status, can be sent without a status byte until a different status byte is needed.

Channel information performs most of the routine work. Commands are addressed to specific channels by a 4-bit number which is encoded into the status byte. The associated data bytes can identify keys going down (on) and up (off), their on or off velocities, and pressure or "after-touch" (on keyboards so equipped).

System Common, Real Time, and Reset information is intended for all channels in a system. System Common information identifies song selections and measure numbers for all units. Real Time information is used for synchronizing everything (perhaps to a master sequencer). Therefore, Channel and System Common information is interruptible by System Real Time information.

System Exclusive information allows the exchange of data which can be formatted as the manufacturer wishes. Only devices which recognize the manufacturer's format will attend the exchange.

Reset simply initializes all equipment to power-on condition.

The five categories are ordered below according to their utility.

Channel

The most significant four bits of each Channel status byte define the command, while the least significant four bits identify the effective channel.

9xH NOTE ON EVENT
3 bytes: 1001 nnnn + 0kkk kkkk + 0vvv vvvv

nnnn
Channel code, 0-15. Corresponds to channel numbers 1-16.

kkk kkkk
Key number, 0-127.
For all keyboards, middle C=60. All C key numbers are multiples of 12.
The standard five-octave synth keyboard ranges 36-96.
The 88-note piano keyboard ranges 21-108.

vvv vvvv
Key On velocity, 0-127.
With no velocity sensors, default to 64.
With velocity, 1=ppp (softest), 127=fff (loudest).
Key On velocity=0, turns note off.

8xH NOTE OFF EVENT
3 bytes: 1000 nnnn + 0kkk kkkk + 0vvv vvvv

vvv vvvv
Key Off (release) velocity.
Implemented on Prophet-T8.

AxH POLYPHONIC KEY PRESSURE
3 bytes: 1010 nnnn + 0kkk kkkk + 0vvv vvvv

vvv vvvv
Pressure/After-touch value, 0-127.
Used in Omni mode. (Compare code DxH, Mono mode)

BxH CONTROL CHANGE
3 bytes: 1011 nnnn + 0ccc cccc + 0vvv vvvv

ccc cccc
Control address, 0-127.
Except for the Pitch Bender (0), the controllers are not specifically defined. A manufacturer can assign the logical controllers to physical ones as necessary. The controller allocation table must be provided in the user's operation manual. Continuous controllers (including the Pitch bender) are divided into Most and Least Significant Bytes. If only 7 bits of resolution are needed for a specific controller, only the MSB is sent. It is not necessary to send the LSB. If more resolution is needed, then both are sent, first the MSB, then the LSB. If only the LSB has changed in value, the LSB may be sent without re-sending the MSB.

0	Pitch bender MSB
1	Controller 1 MSB
2	Controller 2 MSB
3	Controller 3 MSB
4-31	Continuous controllers 4-31 MSB
32	Pitch bender LSB
33	Controller 1 LSB
34	Controller 2 LSB
35	Controller 3 LSB
36-63	Continuous controllers 4-31 LSB
64-95	Switches (on/off)
96-123	Undefined
124	Local/Remote Keyboard Control (toggle)
125	Omni Mode Select/All notes off.
126	Mono Mode Select/All notes off.
127	Poly Mode Select/All notes off.

If c=125, 126, or 127, v (see below) must be 0.

vvv vvvv
Control value, 0-127.
For mode selections (c=125, 126, or 127), vvv vvvv must be 0.
Pitch benders should range from 0-127, with 64 being center (no pitch bend).
Other controllers will range from 0=minimum to 127=maximum.
Switches are defined 0=off, 127=on.

CxH PROGRAM CHANGE
2 bytes: 1100nnnn + 0ppp pppp

ppp pppp
Program number, 0-127

DxH CHANNEL PRESSURE
2 bytes: 1101nnnn + 0vvv vvvv

vvv vvvv
Channel pressure/after-touch amount, 0-127.
For Mono mode: channel (rather than key) is identified.

ExH UNDEFINED
(SCI uses this status for Pitch Wheel change in the Prophet-600. For further information, see the Prophet-600 MIDI specification.)

System Exclusive

A format has been defined for System Exclusive information, consisting of a two-byte preamble, the data itself, and a one-byte end code. The purpose of this format is to provide for the transmission of data which may be useful to any two instruments from one manufacturer but uninterpretable to other MIDI-bussed devices. For example, SCI uses this protocol for loading and dumping program data. System Exclusive information can only be interrupted by a System Reset command.

Format: F0H + 0iii iiii + data + F7H

F0H
Status byte. Must be followed by manufacturers ID#

0iii iiii
Manufacturers ID#.
iii iiii can be 0-127
Current ID numbers are:

Sequential Circuits	01H
Kawai	40H
Roland	41H
Korg	42H
Yamaha	43H

Receivers which do not recognize the ID# ignore the ensuing system exclusive data.

data
Any number of bytes.
MSB must be reset. (Otherwise will signal a new status byte.) Data can range 0-127.
Data is intended for all channels.

F7H
An END-OF-BLOCK code which terminates System Exclusive status.
SYSTEM RESET will also terminate System Exclusive status.

In no case should other data or status codes be interleaved with System Exclusive data, regardless of whether or not the ID code is recognized.

Under "data", SCI uses the following protocol to code program transfers (see also the MIDI specification for each instrument):

Byte 3	Byte 4			
00H	PP	F7H		PROGRAM SEND REQUEST
01H	PP	dd...dd	F7H	PROPHET-5 PROGRAMS
02H	PP	dd...dd	F7H	PROPHET-600 PROGRAMS
03H	PP	dd...dd	F7H	PROPHET-T8 PROGRAMS
04H	PP	dd...dd	F7H	PROPHET-10 PROGRAMS

PP=program number
dd= data in four-bit nibbles, LS nibble first, right justified.

System Real Time

The System Real Time codes control the entire system in real time. They are used for synchronizing sequencers and rhythm units.

To maintain timing precision, these codes can be sent between any System Common or Channel data sets which consist of two or more bytes. However, the codes may not be interleaved with System Exclusive data.

System Real Time statuses are intended for all channels and recognized by all units using the interface. If the functions specified are not implemented, they are simply ignored.

F8H TIMING-CLOCK-IN-PLAY
This clock is sent while the transmitter is in Play mode. The system is synchronized with this clock which is sent at a rate of 24 clocks/quarter note.

F9H MEASURE-END
The MEASURE-END is transmitted instead of the TIMING-CLOCK-IN-PLAY at the end of each measure.

FAH START-FROM-1st-MEASURE
This code is immediately sent when the PLAY button on the master (e.g. sequencer or rhythm unit) is hit. The first TIMING-CLOCK-IN-PLAY must follow within 5 ms after this code.

FBH CONTINUE START
This is sent when the CONTINUE button (on the master) is hit. A sequence will restart from the point where the sequence stopped on the last TIMING-CLOCK-IN-PLAY. The next TIMING-CLOCK-IN-PLAY must be sent within 5 ms after this code.

FC TIMING-CLOCK-IN-STOP
This code is clocked in Stop mode, to synchronize a Phase-Locked Loop (PLL) which is used (during Stop) for interpolating the timing clock.

System Common

System Common information is intended for all channels in a system.

F1H Undefined

F2H MEASURE INFORMATION
3 bytes: F2H + 0mmm mmmm (MS) + 0mmm mmmm (LS)
The two data bytes code the 14-bit measure number.

F3H SONG SELECT
2 bytes: F3H + 0sss ssss
The data byte codes the 7-bit song number.

F4H Undefined

F5H Undefined

F6H TUNE REQUEST
Initiates synthesizer tune routines.

System Reset

There is one system reset code. It initializes the entire system to the condition of just having power switched on.

FFH SYSTEM RESET

System Reset should be used sparingly, preferably under manual command only. In particular, it should not be sent automatically on power up. This could cause two units connected together to endlessly reset each other.

CONCLUSION

This concludes this introduction to the MIDI specification. Practical applications will be covered in separate articles which discuss MIDI implementation for each SCI instrument.

Finally, I can't help but observe that MIDI really does present some astounding new opportunities for electronic musicians. It should stimulate those whose enthusiasm may have waned because of the general incompatibilities and rapid obsolescence of equipment manufactured over the past few years.

Document No: MIDI-0
Issued: January, 1983
Revised:

MIDI SPECIFICATION-12/82
MIDI Committee

MUSICAL INSTRUMENT DIGITAL INTERFACE

The Musical Instrument Digital Interface, or MIDI, is a specification designed to enable interconnecting synthesizers, sequencers, home computers, rhythm machines, etc. with a standard interface.

HARDWARE

The interface is serial, operating at 31.25 Kbaud, asynchronous, with a start bit, 8 data bits (D0 to D7), and stop bit. This makes a total of 10 bits (320 micro sec).

Circuit: 5 ma current loop type. Logical 0 is current ON. One output shall drive one and only one input. The receiver shall be opto-isolated and require less than 5ma to turn on.

Connectors: DIN 5 pin (180 degree) female panel mount receptacle. An example is the SWITCHCRAFT 57GB5F. The connectors shall be labelled "MIDI IN" and "MIDI OUT".

Cables shall have a maximum length of fifty feet (15 meters), and shall be terminated on each end by a corresponding 5-pin DIN male plug, such as the SWITCHCRAFT 05GM5M. The cable shall be shielded twisted pair, with the shield connected to pin 2 at both ends. (See Figure 1)

XLR connectors are acceptable as an alternate standard, providing manufacturers using them make available all necessary conversion cables.

A "MIDI THRU" output may be provided if needed, which provides a direct copy of data coming in MIDI IN.

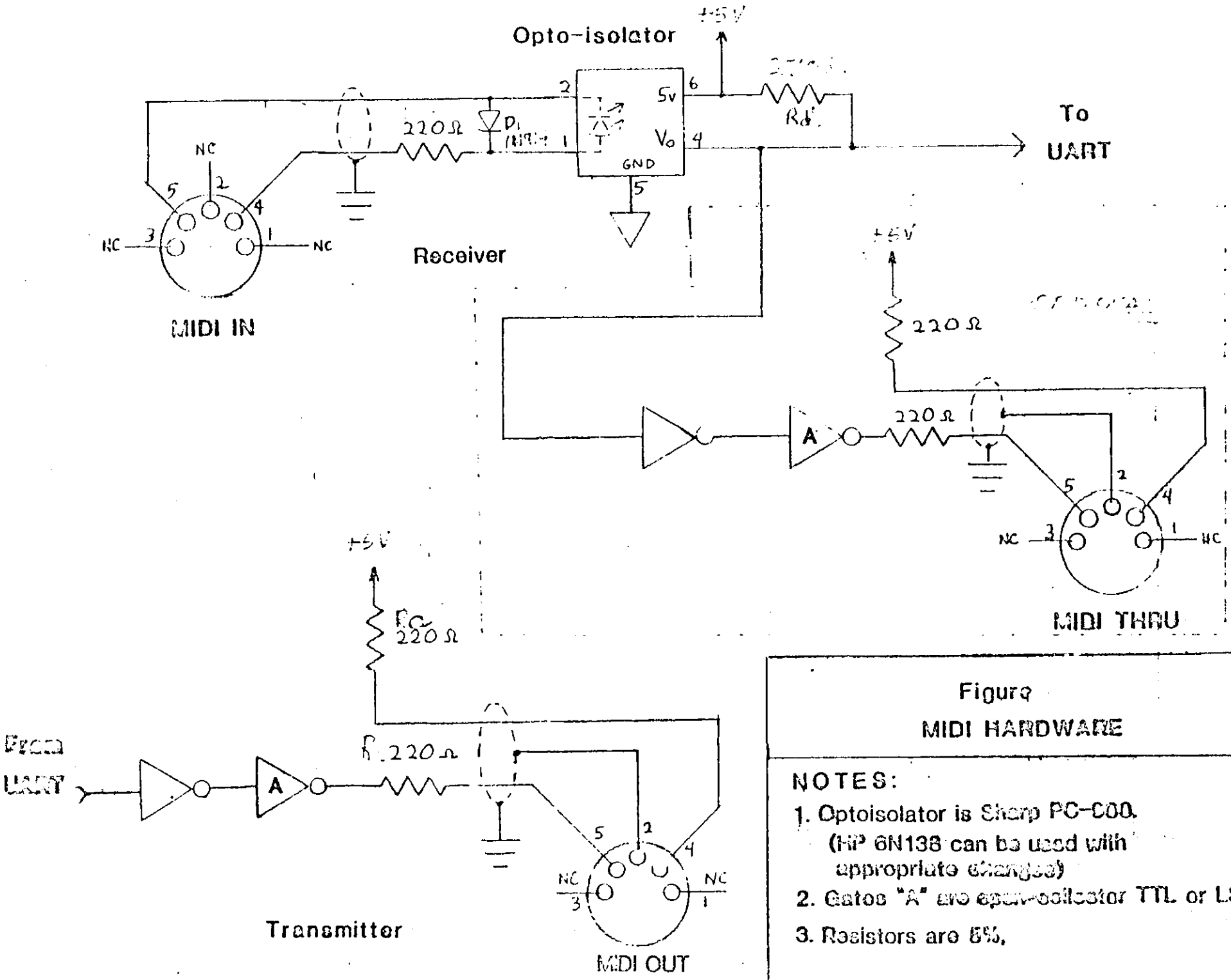


Figure
MIDI HARDWARE

NOTES:

1. Optoisolator is Sharp PC-000.
   (HP 6N138 can be used with appropriate changes)
2. Gates "A" are open-collector TTL or LS.
3. Resistors are 5%.

DATA FORMAT

Data are in two main categories: CHANNEL INFORMATION, which is addressed to a specific channel and contains a 4-bit channel number in the status byte, and SYSTEM INFORMATION, which is intended for all channels in a system and can be COMMON, EXCLUSIVE, or REAL TIME.

CHANNEL, SYSTEM REAL TIME and SYSTEM COMMON information are recognized by all units using the interface, and if the functions specified are not implemented, they are simply ignored. SYSTEM EXCLUSIVE information is sent between a MANUFACTURER's ID NUMBER and an END OF BLOCK status byte, can be in any format (providing the most significant bit is always reset), and is ignored by units not recognizing the ID code.

The most significant bit is a flag which is set in all status bytes and reset in all data bytes, limiting all data to 7-bit length.

The information is prioritized from highest to lowest according to the following scheme:

1. SYSTEM RESET command
2. SYSTEM EXCLUSIVE INFORMATION
3. SYSTEM REAL TIME INFORMATION (except SYSTEM RESET)
4. COMMON and CHANNEL INFORMATION

Data transmissions can be interrupted only by information of a higher priority.

There are three modes of operation for receivers and transmitters: OMNI, POLY, and MONO. Receivers are configured upon command of a transmitter, but need not be capable of working in all three modes. Receivers and transmitters without channel selection capability are always assigned by default to channel one. (nnnn=0000)

In MONO mode, only one voice per channel is sent. Legato can be done in this mode by sending a new NOTE ON, for a different note value, without having sent a NOTE OFF . When a receiver switches to MONO mode, it will receive on the channel it is assigned to and above, up to the number of voices it has. For example, an eight voice without channel selection capability would receive and send in channels 1-8.

In POLY mode, any number of voices may be sent in a single channel. Receivers assign the incoming notes to voices according to normal internal priority.

In OMNI mode, receivers respond to NOTE EVENT data coming in on any channel. All these incoming notes are assigned to voices according to normal internal priority.

At power up reset, a receiving device must default to OMNI mode until receiving a MODE SELECT code in the channel assigned. If it is a POLY code, the receiver switches to POLY mode, & acts only on data coming in on that channel. If it is a MONO code, & the receiver does not have the capability to handle MONO data, it remains in OMNI mode. If the receiver can handle MONO data, it switches to MONO mode. It is possible for a unit which is processing and sending data in OMNI mode to send out a MONO or POLY command, requesting another unit to switch to that mode. However, it may not itself switch modes until it receives a command.

SYSTEM REAL TIME INFORMATION codes are for synchronizing sequencers, rhythm units, etc. To maintain timing precision, these code can be sent at any time, even between data bytes sent under another status code, except during SYSTEM EXCLUSIVE data blocks.

The SYSTEM RESET code should be used sparingly, preferably under manual command only. In particular, it should not be sent automatically on power up. This could lead two units connected together to endlessly reset each other.

OMNI, POLY and MONO MODE select codes serve as "ALL NOTES OFF" commands in the channel(s) they control.

TABLE I

SUMMARY OF STATUS BYTES

Status D7----D0	# of Bytes Following	Description
Channel Information		
1000nnnn	2	Note OFF event
1001nnnn	2	Note ON event (velocity=0: Note OFF)
1010nnnn	2	Polyphonic key pressure/after touch
1011nnnn	2	Control change
1100nnnn	1	Program change
1101nnnn	1	Channel pressure/after touch
1110xxxx		Undefined
System Information		
11110000	*****	System Exclusive Information
11110sss	0 to x	System Common Information
11111ttt	0	System Real Time Information

NOTES:

#:	Number of following bytes
nnnn:	Channel #, where 0000 is channel one. 0001 is channel two. . . . 1111 is channel sixteen.
*****:	0iiiiiii, data ... 11110111 (EOB)
iiiiiii:	Identification
sss:	1 to 7
ttt:	0 to 7

TABLE II

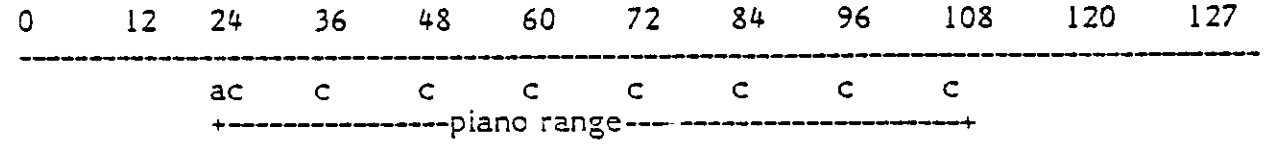
CHANNEL INFORMATION

STATUS	FOLLOWING BYTES	DESCRIPTION
1000nnnn	0kkkkkkk 0vvvvvvv	Note OFF event (see notes 1,2,3) vvvvvvv: note off velocity
1001nnnn	0kkkkkkk 0vvvvvvv	Note ON event vvvvvvv not =0: velocity vvvvvvv= 0: note OFF
1010nnnn	Okkkkkkk Ovvvvvvv	Polyphonic key pressure/after touch vvvvvvv: After-touch value.
1011nnnn	0ccccccc 0vvvvvvv	Control change ccccccc: control # (0-125)(notes 4,5,6,&7) vvvvvvv: control value ccccccc= 125: OMNI Mode /ALL NOTES OFF vvvvvvv= 0: (see Notes 4,10) ccccccc= 126: MONO Mode/ALL NOTES OFF vvvvvvv= 0 (see Notes 4,10) ccccccc= 127: POLY Mode /ALL NOTES OFF vvvvvvv= 0 (see Notes 4,10)
1100nnnn	0ppppppp	Program change ppppppp: Program# (0-127)
1101nnnn	0vvvvvvv	Channel pressure/after-touch vvvvvvv: after-touch value

NOTES:

1. nnnn: channel# (1-16, coded as defined in Table I)

2. kkkkkkk: note# (0 - 127)
   kkkkkkk=60: Middle C of keyboard



3. vvvvvvv: key velocity
   A logarithmic scale would be advisable.

```
0        1                          64                                   127
+----------------------------------------------------------------------------
off      ppp     pp     p     mp     mf     f              ff          fff
```

   vvvvvvv=64: in case of no velocity sensors
   vvvvvvv= 0: NOTE OFF

4. ccccccc: control number

ccccccc	description
0	Pitch bender MSB
1	Controller 1 MSB
2	Controller 2 MSB
3	Controller 3 MSB
4-31	Continuous controllers 4-31 MSB
32	Pitch bender LSB
33	Controller 1 LSB
34	Controller 2 LSB
35	Controller 3 LSB
36-63	Controllers 4-31 LSB
64-95	Switches (on/off)
96-124	Undefined
125	All notes OFF
126	Mono mode select
127	Poly mode select

5. The controllers except the PITCH BENDER (code = 0) are not specifically defined . A manufacturer can assign the logical controllers to physical ones as necessary. The controller allocation table must be provided in the user's operation manual.

6. Continuous controllers (including the Pitch bender) are divided into Most Significant and Least Significant Bytes. If only 7 bits of resolution are needed for any particular controllers, only the MSB is sent. It is not necessary to send the LSB. If more resolution is needed, then both are sent, first the MSB, then the LSB. If only the LSB has changed in value, the LSB may be sent without re-sending the MSB.

7. vvvvvvv: control value (Most Significant Byte)
   (for pitch benders)

```
0                    64                          127
+-------------------------------------------------------------+
      low                          center                         high
```

   (for controllers)

```
0                                                127
+-------------------------------------------------------------+
      min                                                         max
```

   (for switches)

```
0                                                          127
+    -    -    -    -    -    -    -    -                    +
off                                                         on
```

8. Any data sets (e.g. NOTE ON event data), which are sent successively under the same status, can be sent without a status byte until a different status byte is needed.

9. NOTE OFF, NOTE ON, and Control change status bytes always have 2 data bytes following.

10. "ALL NOTES OFF/MODE SELECT" codes have a dummy byte for making the data set the same length as others, for easier detection in the receiver.

Example:

```
10110011
01111111
00000000    All OFF in chan 3  (POLY mode)
* * * *     interval                ch3 silent
10010011    note ON event in channel 3
00111100    note = c4
01000000    vel = 64
(10010011)  note ON event (status byte not necessary)
00111110    note = d4
01100000    vel = 96
* * * *     interval                ch3 c4, d4 on
(10010011)  note ON event (status byte not necessary)
00111100    note = c4
00000000    vel = 0 (note OFF)
(10010011)  note ON event (status byte not necessary)
* * * *     interval                ch3 d4 on
01000101    note = a4
01001000    vel = 16
* * * *     interval                ch3 d4, a4 on
10110111
01111110
00000000    All Notes Off/Mono mode select (ch 7)
* * * *     interval                ch7 silent
10010111    note ON event in ch 7
00111100    note = c4
00100000    vel = 32
* * * *     interval                ch7 c4 on
(10010111)  note ON event (not necessary)
00111110    note = d4 (legato)
(11111000)  timing clock in PLAY (can occur any time)
00110011    vel = 51
* * * *     interval                ch7 d4 on
10000111    note OFF event
00111110    note = d4
00010000    vel = 16 (for release)
10110111                            ch7 silent
01111110
00000000    All OFF in ch7 (mono)
* * * *     interval                ch7 silent
                                    ch3 d4, a4 on
```

TABLE III

SYSTEM COMMON INFORMATION
(WITHOUT ID)

Status	Following Bytes	Description
11110001		Undefined
11110010	0mmmmmmm 0mmmmmmm	Measure information mmmmmmm: measure # (Most significant) mmmmmmm: measure # (Least significant)
11110011	0sssssss	Song select sssssss: Song #
11110100		Undefined
11110101		Undefined
11110110		Tune request
11110111		End of block (EOB)

NOTE: EOB - End of Block is used only to terminate SYSTEM EXCLUSIVE Information.

TABLE IV

SYSTEM REAL TIME INFORMATION

Status	Following Bytes	Description
11111000		Timing clock in Play
11111001		Timing clock with measure end
11111010		Start at 1st meas
11111011		Continue start
11111100		Timing clock in Stop
11111101		undefined
11111110		undefined
11111111		System reset

NOTES:
*The SYSTEM REAL TIME INFORMATION codes are transmitted for controlling all of the system in real time.

*The SYSTEM RESET CODE has the highest priority in the interface, and can be sent at any time. The rest of these codes have priority over all transmissions except for SYSTEM EXCLUSIVE INFORMATION. Any CHANNEL and SYSTEM COMMON data sets which consist of 2 or more bytes may be split to insert REAL TIME INFO.

*TIMING-CLOCK-IN-PLAY $F8
This clock is sent while the transmitter is in PLAY mode. The system is synchronized with this clock which is sent at a rate of 24 clocks/quarter note.

*MEASURE-END $F9
The MEASURE-END is transmitted instead of the TIMING-CLOCK-IN-PLAY at the end of each measure.

*START-FROM-1st-MEASURE $FA
This code is immediately sent when the PLAY button on the master (e.g. sequencer or rhythm) is hit. The first TIMING-CLOCK-IN-PLAY must follow within 5 ms after this code.

*CONTINUE START $FB
This is sent when the CONT button is hit. A sequence will restart from the point where the sequence stopped on the last TIMING-CLOCK-IN-PLAY. Following TIMING-CLOCK-IN-PLAY must be sent within 5 ms after this code.

*TIMING-CLOCK-IN-STOP $FC
This clock is sent in STOP mode, and is used to synchronize the PLL (used for interpolating the timing clock) during STOP mode. It also can be used for fading out and so on.

*SYSTEM-RESET $FF
This code initializes all of the system to the condition of just having turned on power.

TABLE V

SYSTEM EXCLUSIVE INFORMATION
(WITH ID)

Status	Following Bytes	Description
11110000	0iiiiiii (0*******) . . . (0*******) 11110111 (EOB)	Bulk dump etc. iiiiiii: identification Any number of bytes may be sent here, for any purpose, as long as they all have a zero in the most significant bit.

NOTES:
*iiiiiii: identification ID (0-127)

*All bytes between the System Exclusive Status byte and the EOB must have zeroes in the most significant bit.

*The format and ID number can be obtained from the MIDI committee before using it. This information will be held in confidence by the committee if so requested by the manufacturer.

*These transmissions have the highest priority in the system, except for the SYSTEM RESET command. In no case should other data or status codes be interleaved with SYSTEM EXCLUSIVE INFORMATION, regardless of whether or not the ID code is recognized.

*Either EOB or SYSTEM RESET will terminate a SYSTEM EXCLUSIVE Transmission.

Document No: MIDI-6
Issued: January, 1983
Revised:

PROPHET-5 MIDI IMPLEMENTATION

Chet Wood, SCI

Unless otherwise specified, status/data bytes are given in binary, while numbers in descriptions are in decimal.

TRANSMITTED DATA

First Byte	Second	Third	Description
1000 nnnn	0kkk kkkk	0vvv vvvv	Note off. Key # = 36(C0) - 96(C5).
1001 nnnn	0kkk kkkk	0vvv vvvv	Note on. Key # = 36(C0) - 96(C5).
1011 0000	0111 1111	0000 0000	Poly mode select (sent periodically)
1100 0000	0ppp pppp	---	Prog change from front panel (0-39) or (64-103) See Note 1.
1111 0000	01H, 01H, 0ppp pppp, data (48)..., F7H		Prog data dump (sent upon request) (24 bytes)
	Note: Data in program data dumps is sent in four-bit nibbles, right justified, least significant nibble sent first.		
1111 0110	---	---	Tune

RECOGNIZED RECEIVE DATA

1000 nnnn	0kkk kkkk	0vvv vvvv	Note off
1001 nnnn	0kkk kkkk	0vvv vvvv	Note on (if vel=o, turn off) with default, veloc=64)
1011 0000	125, 126, or 127	---	Turns off all notes that MIDI has turned on.
1100 0000	0ppp pppp	---	Prog change (simulates prog# change from front panel) (0-39 or 64-103--see Note 1.)

First Byte	Second	Third	Description
1111 0000	01H, 00H,	0ppp pppp, F7H	Prog dump request (ignores if ID wrong)
1111 0000	01H, 01H, 0ppp pppp, data (48)..., F7H		Prog data dump (ignores if ID is wrong) (24 bytes)
	Note: Data in program data dumps is sent in four-bit nibbles, right justified, least significant nibble sent first.		
1111 0110	---	---	Tune request

NOTES

1. If the front panel is in Scale Mode, 64 is added to the Program number before it is sent. Conversely, if bit 6 of a received program number is set, the program is loaded as if the front panel were in Scale Mode.

2. When retrofitted with MIDI, the Prophet-5 MONO SEQUENCER* interface is disabled. However the standard DIGITAL interface will operate normally.

MODE

The Prophet-5 is always in Omni mode.

FRONT PANEL CODED FUNCTIONS

When the RECORD switch is held down, pressing a PROGRAM SELECT switch will activate the following functions:

Switch	Function
1	Enables/Disables program change receive only. Program changes are always transmitted.
2	Dump current program. When dumping programs, be sure to disable RECORD (with back panel switch). Otherwise you can inadvertently copy one program to another by hitting program changes (with the RECORD light on).

PROPHET-5 PROGRAM BIT MAP

	Switch Bit (7)	Pot Bits (0-6)
Byte 0	OSC A PULSE	FILT ATK
Byte 1	OSC A SAW	FILT DEC
Byte 2	OSC A SYNC	FILT SUS
Byte 3	OSC B SAW	FILT REL
Byte 4	OSC B TRI	AMP ATK
Byte 5	OSC B PULSE	AMP DEC
Byte 6	OSC B KBD	AMP SUS
Byte 7	UNISON	AMP REL
Byte 8	POLY-MOD FREQ A	FILTER CUTOFF
Byte 9	POLY-MOD PW A	FILT ENV AMT
Byte 10	POLY-MOD FILT	MIX OSC B
Byte 11	LFO SAW	OSC B PW
Byte 12	LFO TRI	MIX OSC A
Byte 13	LFO SQUARE	OSC A PW
Byte 14	FILT KBD	MIX NOISE
Byte 15	RELEASE	FILT RESONANCE
Byte 16	W-MOD FREQ A	GLIDE
Byte 17	W-MOD FREQ B	LFO FREQ
Byte 18	W-MOD PW A	W-MOD SOURCE MIX
Byte 19	W-MOD PW B	P-MOD OSC B
Byte 20	W-MOD FILT	P-MOD FILT ENV
Byte 21	OSC B LO FREQ	OSC A FREQ
Byte 22	X	OSC B FREQ
Byte 23	X	OSC B FINE

Document No: MIDI-4
Issued: January, 1983
Revised:

PROPHET-600 MIDI IMPLEMENTATION
Dave Smith, SCI

Unless otherwise specified, status/data bytes are given in binary, while numbers in descriptions are in decimal.

TRANSMITTED DATA

First Byte	Second	Third	Description
1000 0000	0kkk kkkk	40H	Note off (key # = 36(CØ) - 96(C5).)
1001 0000	0kkk kkkk	40H	Note on (key # = 36(CØ)) - 96(C5).)
1100 0000	0ppp pppp	---	Prog change from front panel (00-99), if enabled.
1111 0000	01H (SCI ID)	02H, 0ppp pppp, data (32)..., F7 (EOB)	Prog data dump (sent upon request) (32 nibbles, 16 bytes)

Note: Data in program data dumps is sent in four-bit nibbles, right justified, least significant nibble sent first.

1110 0000	0 p6 p5 p4 p3 p2 p1 p0 (LS 7 bits)	0 p13 p12 p11 p10 p9 p8 p7 (MS 7 bits)	Note: Pitch wheel change (when enabled). 14-bit signed 2's complement pitch wheel value. 8-bit accuracy (Bit p0=0, bits p9-p13 are sign bits). If both bytes=0, no pitch change (wheel centered).
1011 0000	1	000m mmmm	Mod wheel amount data (if enabled).

Note: Wheel values are only sent whenever a change in position is detected.

RECOGNIZED RECEIVE DATA

1000 xxxx	0kkk kkkk	0vvv vvvv	Note off. Velocity ignored.
1001 xxxx	0kkk kkkk	0vvv vvvv	Note on (if vel=0, turn note off. Otherwise velocity ignored.)

Note: When the note-on code is sent once (1001 xxxx), new notes can be played without using a new note on or off status byte, by using velocity=0 for note off.

First Byte	Second	Third	Description
1100 xxxx	0ppp pppp	---	Prog change (If enabled, simulates prog# change from front panel.)
1111 xxxx	01H (SCI ID)	00H, 0ppp pppp, F7 (EOB)	Prog dump request (ignores if ID wrong)
1111 xxxx	01H (SCI ID)	02H, 0ppp pppp, data (32)..., F7 (EOB)	Prog data dump (ignores if ID is wrong) (32 nibbles, 16 bytes.)
1110 xxxx	0 p6 p5 p4 p3 p2 p1 p0 (LS 7 bits) 0 p13 p12 p11 p10 p9 p8 p7 (MS 7 bits)		Note: Pitch wheel change (when enabled). The 14 bits are shifted, after receipt, as follows: p13 p13 p12 p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 p1 p0 0 That is, p13 is assumed to be the sign bit and hence is moved into the MSB. The LS byte is shifted to put the bits in the correct order. For reference, bit p6 = 1 semitone in the Prophet-600. Also note that this pitch value does not go through auto-tune, and as such should not range beyond +/- a perfect 5th, for optimum tuning.
1011 xxxx	1	000m mmmm	External modulation amount (if enabled). Note: This amount is added to MOD wheel and INITIAL MOD AMOUNT pots to establish total amount of modulation.

MODE

The Prophet-600 is always in Omni mode.

FRONT PANEL CODED FUNCTIONS:

When the RECORD switch is held down, pressing a PROGRAM SELECT switch will activate the following functions:

Switch	Function
1	Enable/Disable program change, both transmit and receive (toggles). On power up, MIDI program change is disabled.
2	Dump current program. Sends the 16 bytes of the stored Non-Volatile program that is displayed in the PROGRAM display, regardless of mode (Preset, Manual, Edit, etc.).
3	Reserved for machine service function (pitch wheel deadband centering).
4	Enable/Disable Pitch and Mod wheel control transmit and receive. Disabled on power up.

Note that if two -600's are connected together, to work correctly, both switch function 1 and 4 must be done on both instruments.

PROPHET-600 PROGRAM BIT MAP

16 bytes of program data

BYTE	MS BIT						LS BIT	
0	B0	A6	A5	A4	A3	A2	A1	A0
1	D0	C3	C2	C1	C0	B3	B2	B1
2	E1	E0	D6	D5	D4	D3	D2	D1
3	F4	F3	F2	F1	F0	E4	E3	E2
4	H0	G5	G4	G3	G2	G1	G0	F5
5	I1	I0	H6	H5	H4	H3	H2	H1
6	J3	J2	J1	J0	I5	I4	I3	I2
7	K4	K3	K2	K1	K0	J6	J5	J4
8	M2	M1	M0	L3	L2	L1	L0	K5
9	O2	O1	O0	N3	N2	N1	N0	M3
A	Q2	Q1	Q0	P3	P2	P1	P0	O3
B	S2	S1	S0	R3	R2	R1	R0	Q3
C	U2	U1	U0	T3	T2	T1	T0	S3
D	V6	V5	V4	V3	V2	V1	V0	U3
E	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
F	ZF	ZE	ZD	ZC	ZB	ZA	Z9	Z8

POT BITS
A=OSC A PULSE WIDTH (7)
B=PMOD FIL ENV AMT (4)
C=LFO FREQ (4)
D=PMOD OSC B AMT (7)
E=LFO AMT (5)
F=OSC B FREQ (6)
G=OSC A FREQ (6)
H=OSC B FINE (7)
I=MIXER (6)
J=FILTER CUTOFF (7)
K=RESONANCE (6)
L=FIL ENV AMT (4)
M=FIL REL (4)
N=FIL SUS (4)
O=FIL DEC (4)
P=FIL ATK (4)
Q=AMP REL (4)
R=AMP SUS (4)
S=AMP DEC (4)
T=AMP ATK (4)
U=GLIDE (4)
V=OSC B PULSE WIDTH (7)

SWITCH BITS
Z0=OSC A PULSE
Z1=OSC B PULSE
Z2=FIL KBD FULL (only
Z3=FIL KBD ½ 1 on)
Z4=LFO SHAPE (1=TRI)
Z5=LFO FREQ AB
Z6=LFO PW AB
Z7=LFO FIL
Z8=OSC A SAW
Z9=OSC A TRI
ZA=OSC A SYNC
ZB=OSC B SAW
ZC=OSC B TRI
ZD=PMOD FREQ A
ZE=PMOD FIL
ZF=UNISON

Document No: MIDI-5
Issued: January, 1983
Revised:

USING THE PROPHET-600 MIDI

Stanley Jungleib, SCI

INTRODUCTION

The Prophet-600 is the first commercial synthesizer available with the Musical Instrument Digital Interface (MIDI). This section explains first briefly, then in more detail how to use MIDI and how it is implemented on the Prophet-600. Programmers should also consult the MIDI specification itself and "MIDI Fundamentals" available c/o Sequential Circuits, Inc.

BASIC OPERATION

1. Switch power off on all equipment to be interconnected.

2. Connect Synth A MIDI OUT to Synth B MIDI IN jack.

3. Switch power on. After TUNE, notes played on Synth A will be played simultaneously on Synth B.

4. To enable Synth A program selections to simultaneously select Synth B programs, hold RECORD and press PROGRAM SELECT 1, on both units.

5. To enable the Synth A MOD and PITCH wheels to control Synth B modulation and pitch, hold RECORD and press PROGRAM SELECT 4, on both units.

6. To reprogram Synth B with a specific sound from Synth A, select the Synth A program, then (on Synth A) hold RECORD and press PROGRAM SELECT 2.

7. Steps 4 and 5 must be performed (if desired) each time power is switched on.

CONNECTION/INITIALIZATION

The simplest application is to tie two Prophet-600s together, gaining the sonic power of simultaneous programs (Double Mode). MIDI OUT on the "master" is connected to MIDI IN on the "slave" (Figure M5-0). If it is desired to use either keyboard to control the other, a second cable can be added (Figure M5-1). The -600s are smart enough to distinguish information which arises from their keyboard from that which comes in through the MIDI. Each will send what is played on its keyboard or by its sequencer, but they do not "echo" the MIDI IN info over MIDI OUT. This prevents an infinite loop from forming from the slight delays this all takes.

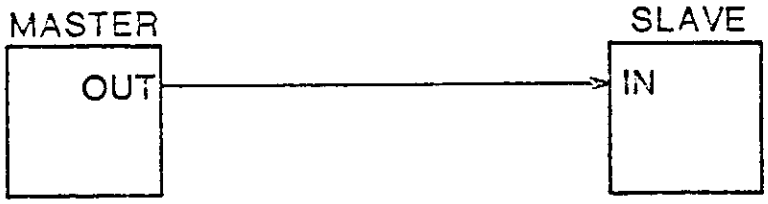


Figure M5-0
SIMPLE CONTROL

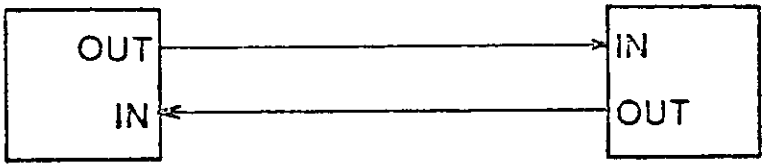


Figure M5-1
DUAL CONTROL

Returning to the simple installation for explanatory purposes, when power is first turned on, both instruments TUNE, then initialize to Omni mode. This means that the master will always be transmitting keyboard and control information on Channel 1, while the slave will be receiving on all channels (even though in this installation Channels 2 - 16 aren't being used).

Both -600s remain in Omni mode, since they are not equipped with Poly or Mono modes.

KEYBOARD INFORMATION

At this point, any key played on the master will be played simultaneously on the slave. Specifically, suppose middle C on the master is struck. This is a Note On event, transmitted to the slave as a three-byte package where the first byte codes the command and channel numbers, the second byte is the key number, and the third byte is the key velocity. For example: 90H-3CH-40H.

90H 9H= Note On status
0H= Channel 1. Range: 0-FH (Channels 1-16)

3CH = key number 60, middle C. Range: 24-5FH (keys 36/C0 - 96/C5)

40H = velocity 64, the default value since velocity is not implemented.

Because the MSB is set (1), the slave sees the first byte as a status byte. This flag tells the receiver to decode this byte as a command, and prepare for the key number and velocity data bytes which follow.

When the instruction has been completely received, the slave plays middle C and will hold it until one of two kinds of commands are received which turn that specific note off (release).

One way to turn the note off is with a Note On command with velocity set to 0, e.g., 90H-3CH-00H. Actually, the slave receiver has a convenient feature which allows a transmitter to delete unnecessary status bytes. The transmitter can leave out the status byte whenever the status doesn't change. So turning this note off in this way really requires only two bytes: 3CH-00H.

The second way to turn the note off is with the Note Off command, which has a different status, and therefore takes three bytes, whenever it follows a Note On status, e.g, 80H-3CH-40H. The Prophet-600 transmitters always send Note Off commands (rather than Note On/Velocity 0 commands). (Note Off is not redundant. It is needed to define the release velocities of, for example, the Prophet-T8.)

As multiple notes are turned on, the slave will assign its voices just as if the notes were coming from its keyboard. In fact its keyboard can be played normally. It will simply play along with the MIDI input, and "steal" voices if a total of more than six are played.

PROGRAM SELECT

On power-up, the Prophet-600s select program 00 for themselves. For simplicity, when they are simply interconnected, master and slave program selections are not linked. Instead, one independently selects programs on both the master and slave.

However a coded control function is provided to enable the slave to follow master program changes. To enable MIDI program changes, while holding down the RECORD then press PROGRAM SELECT 1. This must be done on both units (after power-up).

Now whenever the master program is changed, it will transmit the new program number to the slave in two bytes. The status byte again defines the command and channel numbers, while the second byte contains the program number. For example, C0H-62H.

C0H CH= Program Change status
0H= Channel 1

62H = program number 98. Range: 00-63H (programs 00-99)

When the slave receives this code, it switches to its program 98 and plays in whatever sound is stored there.

To disable MIDI program changes, again hold RECORD and press 1.

MOD WHEEL

Likewise, PITCH and MOD wheel information is also not linked by default, but can be enabled and disabled with a coded function performed on both units (and which links the PITCH wheel as well, see below). Hold RECORD and press 4, on both units. Now whenever a change of master MOD wheel position is detected, a three-byte code will be sent which codes the command, the control number, and the control value. For

example, when the wheel is raised from its off (down) position, the first code sent will be B0H-01H-01H:

B0H BH= Control Change status
0H= Channel 1

01H MOD wheel control address. Range: 00-01H (selects PITCH or MOD)

01H Control value. Range: 00-20H (0-31)

The control value 01H, of course, is the first increment of MOD wheel increase above 00. In the slave, this amount is added to the current values of the slave's MOD wheel and INITIAL MOD AMOUNT pots to establish the total depth of modulation.

PITCH WHEEL

Like the MOD wheel, master PITCH wheel control over the slave is also disabled on power up, but controlled by the RECORD/PROGRAM SELECT 4 operation. Because it affects pitch, this control needs 8-bits of resolution, therefore two data bytes following the status byte. The pitch wheel value is formatted as a 14-bit signed two's-complement number. For example, when the master PITCH wheel is moved up one increment from center, the code will be E0H-02H-00H:

E0H EH=Pitch Change status
0H=Channel 1

02H Increment of LS byte: 0 p6 p5 p4 p3 p2 p1 p0 (LS 7 bits)

00H No change in MS byte: 0 p13 p12 p11 p10 p9 p8 p7 (MS 7 bits)

Bit p0=0 and bits p9-p13 are sign bits when transmitted by a -600. In the receiver, the 14 bits are shifted as follows:

p13 p13 p12 p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 p1 p0 0

That is, p13 is assumed to be the sign bit and hence is moved into the MS bit. The LS byte is shifted to put the bits in the correct order. For reference, bit p6 equals 1 semitone in the -600. Also note that this pitch value does not go through the auto-tune and as such should not range beyond +/- a 5th, for optimum tuning.

PROGRAM DUMP

The master -600 can dump programs to the slave, reprogramming the same location by another coded switch function. For example, if program 33 is selected on the master, hold RECORD and hit PROGRAM SELECT 2. The slave's program 33 will be replaced with the master's program. This will happen regardless of the state of either instrument (Preset, Manual, Edit, etc.). Also note that the actual stored program value is sent, even if it has been edited. This means that edited programs must be recorded before they can be sent.

The program dump occurs within the System Exclusive data format. In this example, the code would be F0H-01H-02H-21H-data-F7H:

F0H System Exclusive status

01H SCI's Manufacturer's ID number

02H Defines program dump (Prophet-600).

21H Program number 33. Range 00-63H (programs 00-99).

data 16 bytes of program data, formatted according to Table 1.
Sent as 32 4-bit nibbles, right justified, LS nibble sent first.

F7H End-of-Block code terminates System Exclusive status.

Although there is no way to transmit this code from the -600, it will also respond to requests for specific program data. The program dump request takes the form:

F0H System Exclusive status

01H SCI's Manufacturer's ID number

00H Defines program dump request

21H Program number 33. Range 00-63 (programs 00-99).

F7H End-of-Block code terminates System Exclusive status.

When this is received, the -600 will transmit the requested program in the format of Table 1, again regardless of the state of the instrument.

If the receiver sees an incorrect ID number, it will ignore the dump request.

Table 1
PROPHET-600 PROGRAM BIT MAP

16 bytes of program data

BYTE	MS BIT						LS BIT	
0	B0	A6	A5	A4	A3	A2	A1	A0
1	D0	C3	C2	C1	C0	B3	B2	B1
2	E1	E0	D6	D5	D4	D3	D2	D1
3	F4	F3	F2	F1	F0	E4	E3	E2
4	H0	G5	G4	G3	G2	G1	G0	F5
5	I1	I0	H6	H5	H4	H3	H2	H1
6	J3	J2	J1	J0	I5	I4	I3	I2
7	K4	K3	K2	K1	K0	J6	J5	J4
8	M2	M1	M0	L3	L2	L1	L0	K5
9	O2	O1	O0	N3	N2	N1	N0	M3
A	Q2	Q1	Q0	P3	P2	P1	P0	O3
B	S2	S1	S0	R3	R2	R1	R0	Q3
C	U2	U1	U0	T3	T2	T1	T0	S3
D	V6	V5	V4	V3	V2	V1	V0	U3
E	Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0
F	ZF	ZE	ZD	ZC	ZB	ZA	Z9	Z8

POT BITS
A=OSC A PULSE WIDTH (7)
B=PMOD FIL ENV AMT (4)
C=LFO FREQ (4)
D=PMOD OSC B AMT (7)
E=LFO AMT (5)
F=OSC B FREQ (6)
G=OSC A FREQ (6)
H=OSC B FINE (7)
I=MIXER (6)
J=FILTER CUTOFF (7)
K=RESONANCE (6)
L=FIL ENV AMT (4)
M=FIL REL (4)
N=FIL SUS (4)
O=FIL DEC (4)
P=FIL ATK (4)
Q=AMP REL (4)
R=AMP SUS (4)
S=AMP DEC (4)
T=AMP ATK (4)
U=GLIDE (4)
V=OSC B PULSE WIDTH (7)

SWITCH BITS
Z0=OSC A PULSE
Z1=OSC B PULSE
Z2=FIL KBD FULL (only
Z3=FIL KBD ½ 1 on)
Z4=LFO SHAPE (1=TRI)
Z5=LFO FREQ AB
Z6=LFO PW AB
Z7=LFO FIL
Z8=OSC A SAW
Z9=OSC A TRI
ZA=OSC A SYNC
ZB=OSC B SAW
ZC=OSC B TRI
ZD=PMOD FREQ A
ZE=PMOD FIL
ZF=UNISON

Document No: MIDI-9
Issued: January, 1983
Revised:

PROPHET-T8 MIDI IMPLEMENTATION

Donna Murray, SCI

Unless otherwise specified, status/data bytes are given in binary, while numbers in descriptions are in decimal.

TRANSMITTED DATA

First Byte	Second	Third	Description
1000 nnnn	0kkk kkkk	0vvv vvvv	Note off. Key # = 21(A0) - 96(C6).
1001 nnnn	0kkk kkkk	0vvv vvvv	Note on. Key # = 21(A0) - 96(C6).
1011 0000	0111 1110	0000 0000	Mono mode select (sent periodically)
1100 0000	0ppp pppp	---	Prog change from front panel (0 - 63 (left) 64-127 (right))
1101 nnnn	0vvv vvvv	---	Pressure change (0-127)
1111 0000	01H, 03H, 0ppp pppp, data (64)..., F7H		Prog data dump sent upon request, or REC + #3 control function (32 bytes)

Note: Data in program data dumps is sent in four-bit nibbles, right justified, least significant nibble sent first.

1111 0110	-		Tune

RECOGNIZED RECEIVE DATA

1000 nnnn	0kkk kkkk	0vvv vvvv	Note off
1001 nnnn	0kkk kkkk	0vvv vvvv	Note on (if vel=o, turn off, with default veloc=64)
1011 0000	124 (decimal)	---	KBD control connect--toggle

Note: When KBD local control is disabled, the KBD will not control the synthesizer. KBD signals will be sent out on MIDI, and only MIDI IN signals will control the synthesizer sound-generation circuitry.

First Byte	Second	Third	Description
1011 0000	125	---	Omni mode select
1011 0000	126	---	Mono mode select
1011 0000	127	---	Poly mode select
1100 0000	0ppp pppp	---	Prog change (simulates prog# change from front panel)
1101 nnnn	0vvv vvvv	---	Pressure
1111 0000	01H, 00H, 0ppp pppp, F7H		Prog dump request (ignores if ID wrong)
1111 0000	01H, 03H, 0ppp pppp, data (64)..., F7H		Prog data dump (ignores if ID is wrong) (32 bytes)
1111 0110			Tune request

MODE

The Prophet-T8 defaults to Omni mode on power-up.

In Poly mode, the -T8 reads only channel 1.

In Mono mode, the -T8 reads 8 mono channels. Assigns key on, key off, and pressure by voice #nnnn.

In either Omni or Poly modes, regular internal voice assignment is used and pressure is ignored.

FRONT PANEL CODED FUNCTIONS

When the RECORD switch is held down, pressing a PROGRAM SELECT switch will activate the following functions:

Switch	Function
1	Enable/Disable MIDI program change. (If disabled, any program changes received from MIDI are ignored. Disabled on powerup.)
2	Dumps present program to MIDI. (If in upper, dumps upper program. See Note 1)
3	Machine service function.

PROPHET-T8 PROGRAM BIT MAP

	Switch Bit (7)	Control Bit (6)	Aux Bit (5)	Pot Bits (0-4, 0-5, or 0-6)
Byte 0	PR LFO FREQ			FILT RES (0-6)
Byte 1	PR LFO AMT	S7		MIX NOISE (0-5)
Byte 2	PR AMP	S6		MIX OSC B (0-5)
Byte 3	PR FILT	S5		MIX OSC A (0-5)
Byte 4	PR PW			P-MOD OSC B (0-6)
Byte 5	PR FREQ B			PRESS AMT (0-6)
Byte 6	PR FREQ A	S4	x	LFO FREQ (0-4)
Byte 7	ENA WHEEL			P-MOD FILT ENV (0-6)
Byte 8	P-MOD FILT	S3		OSC A FREQ (0-5)
Byte 9	P-MOD PW A			OSC A PW (0-6)
Byte A	P-MOD FR A			LFO-MOD INIT AMT (0-6)
Byte B	OSC A TRI	S2		OSC B FREQ (0-5)
Byte C	OSC A SAW			OSC B PW (0-6)
Byte D	OSC A SYNC			OSC B FINE (0-6)
Byte E	LFO FILT			FILT CTF (0-6)
Byte F	LFO PW			FILT KBD AMT (0-6)
Byte 10	LFO FREQ B	S1	x	FILT REL (0-4)
Byte 11	LFO FREQ A			FILT SUS (0-6)
Byte 12	LFO SQUARE	S0	x	FILT DEC (0-4)
Byte 13	LFO TRI	x	x	FILT ATK (0-4)
Byte 14	LFO SAW			FILT ENV AMT (0-6)
Byte 15	OSC A PULSE	P3		REL ENV RATE (0-5)
Byte 16	OSC B SAW	P2		FILT ENV PEAK (0-5)
Byte 17	OSC B KBD	P1		AMP ENV PEAK (0-5)
Byte 18	OSC B LO	P0		ATK/DEC ENV RATE (0-5)
Byte 19	OSC B PULSE	L5	x	AMP REL (0-4)
Byte 1A	OSC B TRI			AMP SUS (0-6)
Byte 1B	ADR	L4	x	AMP DEC (0-4)
Byte 1C	DOUBLE	L3	x	AMP ATK (0-4)
Byte 1D	SPLIT	L2	x	GLIDE (0-4)
Byte 1E	SINGLE	L1	x	2ND FILT RELEASE (0-4)
Byte 1F	UNISON	L0	x	2ND AMP RELEASE (0-4)

x=not used
S0-S7= Split key number
P0-P3= PROG VOLUME
L0-L5= Link program number

Document No: MIDI-7
Issued: January, 1983
Revised:

PROPHET-10 MIDI IMPLEMENTATION

Chet Wood, SCI

Unless otherwise specified, status/data bytes are given in binary, while numbers in descriptions are in decimal.

TRANSMITTED DATA

First Byte	Second	Third	Description
1000 nnnn	0kkk kkkk	0vvv vvvv	Note off. Key # = 36(C0) - 96(C5).
1001 nnnn	0kkk kkkk	0vvv vvvv	Note on. Key # = 36(C0) - 96(C5).
1011 0000	0111 1111	0000 0000	Poly mode select (sent periodically)
1100 0000	0ppp pppp	---	Prog change from front panel (0-31)
1111 0000	01H, 04H, 0ppp pppp, data (64)..., F7H		Prog data dump (sent upon request) (32 bytes). See note 1.
	Note: Data in program data dumps is sent in four-bit nibbles, right justified, least significant nibble sent first.		
1111 0110	---	---	Tune

RECOGNIZED RECEIVE DATA:

1000 nnnn	0kkk kkkk	0vvv vvvv	Note off
1001 nnnn	0kkk kkkk	0vvv vvvv	Note on (if vel=o, turn off with default veloc=64)
1011 0000	125	---	Omni mode select
1011 0000	127	---	Poly mode select
1100 0000	0ppp pppp	---	Prog change (simulates prog# change from front panel)
1111 0000	01H, 09H, 0ppp pppp, F7H		Prog dump request (ignores if ID wrong) (Note 3)

First Byte	Second	Third	Description
1111 0000	01H, 04H, 0ppp pppp, data (64)..., F7H		Prog data dump (ignores if ID is wrong) (32 bytes) (note 1)
1111 0110			Tune request

NOTES

1. Program dump/request is numbered 0-31 for lower and 32-63 for Upper, and is always sent and received in Channel 1.

2. When the Prophet-10 is retrofitted with MIDI the MONO SEQUENCER interface is disabled.

MODE

The keyboard and mode functions are enabled and disabled by the front panel coded functions described on the next page.

See Figure 1.

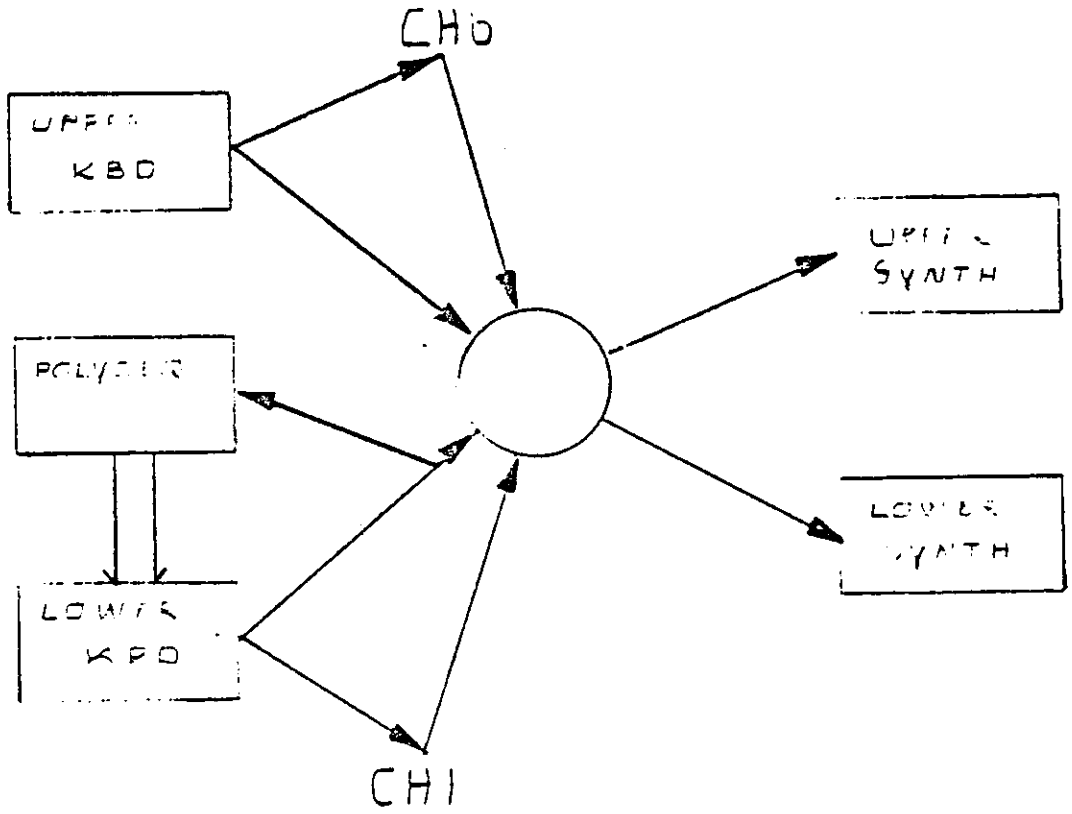


Figure 1
PROPHET-10 CHANNEL FLOW

On power up, both the Upper and Lower synths (if enabled) will receive and transmit in Omni mode. Receiving encompasses any channel: In Omni mode all received data is sent to Lower and/or Upper synths, and Polysequencer. Also if enabled, Upper and Lower and Polysequencer keyboard data will all be sent out on Channel 1.

The Prophet-10 transmits a Poly Mode Select/All keys off command continually.

The Lower keyboard listens on Channel 1, while the Upper keyboard listens on Channel 6 for Mode Select codes.

When it receives a Poly Mode Select command in Channel 1, it will switch to Poly mode: transmitting and receiving Lower keyboard (and internal Polysequencer) data (if enabled) in Channel 1, and Upper keyboard data (if enabled) in Channel 6. If the Crossover switch is activated (see below), Upper will go over Channel 1, and Lower, over Channel 6.

FRONT PANEL CODED FUNCTIONS

When the RECORD switch is held down, a MIDI mode select function is enabled, and the present state will be displayed on the LEDs. When an LED is lit, it means the corresponding function is enabled for both Transmit and Receive. NOTE: To prevent accidental program erasure, switch the back panel RECORD switch to DISABLE.

SWITCH	FUNCTION
UPPER	Upper Keyboard data
LOWER	Lower Keyboard data
NORMAL	Upper Program Changes
SINGLE	Lower Program Changes
A-440	Crossover. When in the Crossover state, the Upper data (if enabled) will be sent in Channel 1, and Lower/polysequencer data (if enabled) in Channel 6.
TUNE	Program send. 'Program send' is not a state, but a command which sends out the program currently selected: either the Upper one if Upper keyboard data is enabled, or the Lower one if the Lower keyboard is enabled and the Upper is not, or neither if neither is enabled.

PROPHET-10 BIT MAP

BYTE(H)	MS BIT	LS 7 BITS
0	C7	FILT ATK
1	C6	FILT DEC
2	C5	FILT SUS
3	C4	FILT REL
4	C3	AMP ATK
5	C2	AMP DEC
6	C1	AMP SUS
7	C0	AMP REL
8	B7	GLIDE
9	B6	OSC A PW
A	B5	OSC B PW
B	B4	MIX OSC A
C	B3	MIX OSC B
D	B2	MIX NOISE
E	B1	FILT RES
F	B0	FILT ENV AMT
10	A7	LFO FREQ
11	A6	LFO AMT
12	A5	FILTER CUTOFF
13	A4	PROG VOLUME
14	A3	P-MOD ENV AMT
15	A2	P-MOD OSC B AMT
16	A1	OSC A FREQ
17	A0	OSC B FREQ
18	—	OSC B FINE
19	--	TUNE

C= HI EQ
B= MID EQ
A= LO EQ

Note: EQ bytes for Prog: are stored under Prog:

18	28
28	38
38	48
48	18

1A	--	--	Z5	Z4	Z3	Z2	Z1	Z0
1B	--	--	Z15	Z14	Z13	Z12	Z11	Z10
1C	--	--	Z25	Z24	Z23	Z22	Z21	Z20
1D	--	--	Z35	Z34	Z33	Z32	Z31	Z30
1E	--	--	--	Z44	Z43	Z42	Z41	Z40
1F	--	--	Z55	Z54	Z53	Z52	Z51	Z50

—= not used

SWITCH BITS

Z5 LFO SQUARE	Z4 LFO TRI	Z3 LFO SAW	Z2 P-MOD FILT	Z1 P-MOD PW A	Z0 P-MOD FREQ A
Z15 UNISON	Z14 M-MOD FILT	Z13 M-MOD PW B	Z12 M-MOD PW A	Z11 M-MOD FREQ B	Z10 M-MOD FREQ A
Z25 OSC B PULSE	Z24 OSC B TRI	Z23 OSC B SAW	Z22 OSC A SYNC	Z21 OSC A PULSE	Z20 OSC A SAW
Z35 OSC B KBD	Z34 OSC B LO	Z33 PED 1 FILT	Z32 PED 1 F B	Z31 PED 1 F A	Z30 FILT KBD
--	Z44 LFO UL MIX	Z43 PED 2 AMP	Z42 PED 2 FILT	Z41 PED 1 M-MOD	Z40 PED 1 AMP
Z55 ALTERNATE	Z54 DOUBLE	Z53 SINGLE	Z52 NORMAL	Z51 DRONE	Z50 RELEASE