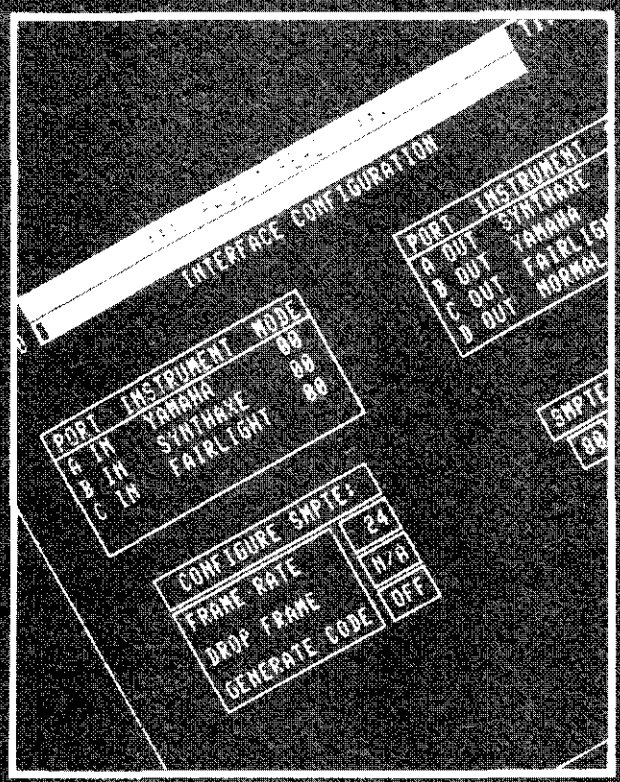


FAIRLIGHT

COMPUTER MUSICAL INSTRUMENT

GENERAL INTERFACE CARD

F A I R L I G H T

- C M I -

MANUAL FOR THE GENERAL INTERFACE

(MIDI/SMPTE) CARD AND SUPPORT UNIT

Revision 1.5

March 1986

by Tom Stewart and Peter Farleigh

The General Interface Card

A SMPTE/MIDI hardware option is available for the CMI. This is in the form of a circuit card which can be inserted into one of the spare expansion slots in the CMI and a connector box that attaches to rear of the CMI. The card contains its own 32-bit 68000 processor, memory and peripheral interfaces. This provides SMPTE time-code generator and reader, three MIDI input ports and four MIDI output ports.

Synchronizing with SMPTE - the industry standard

The CMI can be programmed by the user to play sequences of music or sound effects triggered by SMPTE time-code. SMPTE code can be produced and read by the CMI.

SMPTE time-code is an internationally recognized digital code recorded on an unused sound track of a multi-track tape. The code carries timing information in hours, minutes seconds and frames. By recording SMPTE time-code from the beginning to the end of the audio tape, it is possible to uniquely identify any place on tape by its time-code at that point.

Time code **00:00:00:00** would be the beginning of time-code.
Time code **01:28:45:19** would be 1 hour, 28 minutes, 45 seconds and 19 frames from the beginning.

A **frame** is a subdivision of a second that is used in film and video and can be one of the following values:

for film and cinema	- 24 frames per second
for video (not U.S.A.)	- 25 frames per second
for video	- 30 frames per second

The Fairlight SMPTE system features include 24, 25 or 30 frames per second, "drop frame" capability for 30 fps and balanced line input and output. Signals conform to international standards.

MIDI - Musical Instrument Digital Interface

MIDI is the industry standard digital keyboard communications interface that provides musical performance information in a standardised format. MIDI is compatible with a wide range of modern synthesizers, sequencers, drum machines, and keyboard (and other) controllers.

Fairlight's MIDI controller has three independent MIDI inputs and four independent MIDI outputs. A powerful 32-bit processor provides the fastest possible speed throughout while allowing independent data configurations to be selected for different manufacturers instruments which may be used simultaneously.

Please note that not all MIDI data sent by the CMI will be necessarily played by another synthesizer. For example, if a synthesizer does not have touch sensitivity, then it will not respond to touch sensitivity data sent from the CMI via MIDI.

MIDI Software is included on CMI system disk and is used whenever the system is started with the General Interface card (CMI-28) plugged in. MIDI is therefore active all the time.

The following information is subject to change - please check with your dealer when ordering to confirm latest specifications.

Fairlight Receiving MIDI Data

Fairlight CMI will respond to the following messages received via MIDI IN.

Voice messages:

- Note Off event
- Note On event
- Polyphonic key pressure
- Control and switch change
- Channel pressure
- Pitch bend change

Any Fairlight effect on Page 7 may be patched to any CONTROL (or SWITCH) and thus be controlled by any other synthesizer via MIDI.

Mode messages:
All Channel Mode messages are accepted but ignored.
The Fairlight operates in MIDI Mode 4.
Within the Fairlight, keyboards can be assigned to any combination of REGISTERS/VOICES/CHANNELS.

Fairlight Transmitting MIDI Data

The following data is transmitted via MIDI OUT.

Voice Messages:

- Note Off event
- Note On event
- Polyphonic key pressure
- Control and switch change
- Channel pressure
- Pitch Bend change

PAGE I - General Interface Page

This page operates in conjunction with an optional hardware circuit card, the General Interface Card (or SMPTE/MIDI card). The card is inserted into slot 2 in the CMI and wired to a junction box attached to the rear of the CMI (or as a remote box). See diagram for external connection.

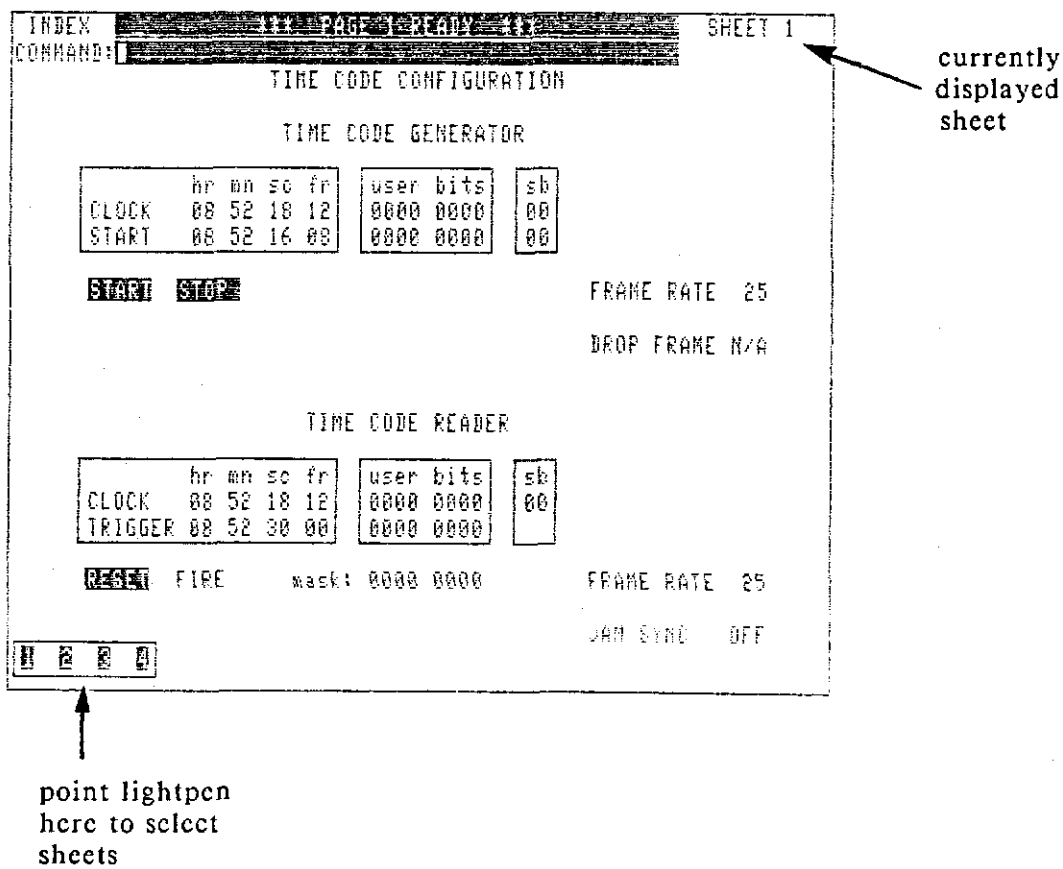
The CMI may be rapidly configured as a MIDI instrument, in a variety of transmit and receive modes.

MIDI uses sixteen channels to send or receive music information. Switch and control fader information is also sent or received.

The General Interface card has three input ports, A to C and four output ports, A to D. Any port may carry any MIDI channel. Some synths only operate on MIDI channel 1. By connecting these synths to different CMI MIDI ports, the problem of MIDI channel contention is avoided.

Present MIDI features of Page I allow the patching of any MIDI port/channel to any of the eight Fairlight keyboards for input and output. As well, MIDI switches/control faders may be patched to Fairlight controls/switches for input and output.

Page I has four sheets. Select sheets by typing the sheet number; **1, 2, 3 or 4** followed by **<return>**. Alternatively, sheets can be selected by pointing the light-pen at the sheet selection box in the bottom left hand corner of the screen. The CMI will remember which sheet you are working on at present and returning to Page I from another page will land you on the sheet last used.



SHEET 1 - SMPTE TIME CODE CONFIGURATION

This sheet is divided into time code **Generating** and time code **Reading**.
PAGE I can read or generate SMPTE code in different modes. These modes are described below.

TIME CODE GENERATOR

Connect a cable between the SMPTE OUT socket on the rear of the CMI to a line level input on your tape recorder.

Record time code onto high quality multi-track tape at levels between -10db to -20db, rather than 0db. This will minimise bleed-through to adjacent tracks. Avoid recording onto outside tracks (e.g., track 1 or track 24 on 24 track tape) as these are the first tracks to fray if the tape is extensively used. Also avoid recording instruments with heavy percussive bass content (e.g., bass drum) on an adjacent track as this may modulate the SMPTE track.

To generate time code for recording onto tape, firstly set up the START time by tabbing down to the numbers opposite START.

These numbers are: **hr**(hour), **mn**(minute), **sc**(second), **fr**(frame)

Set the START time by typing **number<set>**.

Time code can now be generated by either typing **S<return>** or pointing the lightpen at the **START** box.

To stop the time code generator, either type **S<return>** or point the lightpen at the **STOP** box.

Note that all time code readers need a few seconds to lock onto the time code, so make START time slightly early.

EXAMPLE Record time code onto tape with a START time of **00** hours **59** minutes **55** seconds **00** frames. This will allow sufficient time to trigger a sequence of music at SMPTE time **01 00 00 00**.

The CLOCK area above the START area will display the SMPTE code as it is generated. Similarly the CLOCK in the Time Code reader section will display the code as it is read in. You may verify that CMI SMPTE is functioning by directly connecting SMPTE-OUT to SMPTE-IN. Both clocks should display the same time if running.

FRAME RATE can be set to **24, 25** or **30** frames per second for either generating or reading time code. When generating code at 30 fps, there is a **Drop Frame** option. This is used mainly with NTSC colour television applications.

JAM SYNC is used to re-stripe time code onto tape if the original time code has dropouts (discontinuities) on it. Feed the original time code from tape into the SMPTE input. Record the SMPTE output to a spare track on tape. By pointing the lightpen at **JAM SYNC**, the CMI will generate the same time code as that being fed into it. If time code on tape drops out, the CMI SMPTE generator will continue to generate time code at the correct rate. When time code from tape reappears, the CMI will use that as its reference.

TIME CODE READER

The CMI SMPTE time code reader needs a minimum level of around **1 volt p-p.**

TRIGGER time is the point where music sequences may be started. It is set up in a similar way to START TIME, described previously.

The CMI has three sequencers, PAGE 9 - Keyboard Sequencer, PAGE C - Music Composition Language and PAGE R - Real Time Composer. These sequencers may be triggered to start at a particular SMPTE time code. Source of the time code may be from tape, another SMPTE generator, or from the CMI SMPTE time code generator itself.

At present these sequencers will not lock with SMPTE on tape so that any variations in tape speed will not be reflected in the speed of Page 9, C or R. If this is a problem, record a sync tone on another track of the tape so that any tape speed variations can be taken into account. However, sequence overdubs triggered by SMPTE can "free run" (SYNC=INT) on a reasonable tape recorder for around five minutes before any noticeable delays occur.

Set up your SMPTE/SYNC equipment as shown in the connection diagram.

TRIGGER TIME may be setup on Page I by tabbing down to the numbers opposite TRIGGER.

These numbers are: **hr**(hour), **mn**(minute), **sc**(second), **fr**(frame)

Set the TRIGGER time by typing **number<set>.**

Using the time code reader with PAGE R

Set the trigger time either on Page I as previously described or on Page R, in the bottom left-hand corner.

To have PAGE R start from the trigger point type **P;S<return>.**

PAGE R will go into a wait state (similar to the P;W option) until the clock time reaches the trigger time.

This can be over-ridden by pressing any key which will start the sequence.

Once the sequence has started, PAGE R will update time code as each new pattern is displayed on the screen.

Using the time code reader with PAGE 9

Page 9 will start at the TRIGGER time setup on Page I.

Type **REP;S<return>** to replay, waiting for SMPTE trigger.
Type **REC;S<return>** to record, waiting for SMPTE trigger.
Type **MER;S<return>** to merge, waiting for SMPTE trigger.

Using the time code reader with PAGE C

Type **SMPTE=ON<return>**

This replaces the ;S option as used on Page 9 and Page R. The PLAY command is unchanged.

Page C sequence will TRIGGER from times inserted in to the sequence file. Times may be absolute or relative to the BASE time. Triggering is similar to the WAIT command, in that MCL waits for something to happen before playing. See the MCL manual.

SMPTE COMMAND - given from the command line (not the editor).

Used to enable or disable all SMPTE triggering. This command must be used before attempting any SMPTE triggering.

Type:

SMPTE=ON<return>	to use embedded trigger times.
SMPTE=OFF<return>	to ignore embedded trigger times.
SMPTE<return>	to see the state of SMPTE.

If no SMPTE card is installed, then a message will be displayed . . .

SMPTE= NO CARD INSTALLED

BASE COMMAND - given from the command line (not the editor).

This stores an absolute time value in hours, minutes, seconds and frames. BASE is setup by the user to contain the required start time for synchronizing PAGE C with external devices. BASE time is usually used as a zero reference point.

Type **BASE=<hours>:<minutes>:<seconds>:<frames><return>**

Spaces may be used instead of colons. A parameter may be skipped over by typing a colon twice.

Some valid examples are:

BASE=12:13:14:15<return>
BASE 12 13 14 15<return> same as above
BASE 11: : :14<return> change only the hours and frames

Type **BASE<return>** to see current stored base time.
Alternatively, type **Q<return>** to see the total state of Page C.
A typical display . . .

```
SPEED=10000
SYNC=INT
CLICK=OFF : 48,4
X=OFF
TIME=OFF
WAIT=OFF
WARN=OFF
PROMPT=OFF
FREE MEMORY = $3C00/15630
SMPTE=ON  01:15:25:00
BASE= 01:15:31:20                    <== look here
```

TRIGGER COMMAND - this is embedded within a sequence file (.SS).
It starts and ends with the @ character. It's function is to specify either an absolute time value or an offset value relative to the BASE register to allow SMPTE synchronizing within an MCL composition.

When editing a .SS file, type . . .

@<A or R><hours>:<minutes>:<seconds>:<frames>@<return>

The command must start and end with a @ character. Similar, in a way, to PROMPT MESSAGES which start and end with [and]. Spaces may replace the colon. The letter A or R before the hour setting is optional and has the following effect . . .

A = absolute trigger time - the time is the actual trigger time for SMPTE. Base time has no effect.
R = relative trigger time - the time is relative to BASE time. Think of BASE time as a local zero reference time.

If the A or R option is not typed then default is A (absolute).

EXAMPLES

`@A 12:55:59:23@`	- absolute
`@A12 55 59:23@`	- same as above
`@R 00:04:16:22@`	- relative
`@4:45:01:19@`	- absolute (careful!)

Page I display sheets 2 and 3 are used to configure various aspects of MIDI data and the CMI MIDI input and output ports. These configuration settings are saveable to disk as a .MC file.

SHEET 2 - INTERFACE CONFIGURATION

Other brands of MIDI instruments have subtle differences when transferring MIDI data. These differences only become apparent when connecting different manufacturers' MIDI equipment. INTERFACE CONFIGURATION sets up the CMI MIDI ports to be compatible with other MIDI equipment.

To configure the MIDI ports, tab to the INSTRUMENT field and type the first letter of the manufacturer's product name using that port. Then type <SET>.

Valid letters are F - Fairlight
Y - Yamaha
S - Synthaxe
C - Casio
R - Roland
and N - Normal

The MODE field enables you to specify options to the manufacturers' equipment in which the MIDI port is to operate.
A hexadecimal code number is assigned to the appropriate port to select special modes. Valid hexadecimal numbers are 0 to 9 and A to F.

The following are valid configuration modes . . .

Disable/enable output ports

To disable an output port, tab to corresponding MODE. Type FF<SET>.

If a CMI keyboard is output to multiple ports then the disable port mode will only apply to the first (lowest) port in the set. For example if KBD.1 is sent to ports A, B and C, channel 1 and port A is disabled then no frames are sent on ports A, B or C. If port A is enabled (mode not set to FF) then frames will be transmitted on ports A, B and C regardless of the mode settings of ports B and C.

Example 1 - Port disabling

PORT	INSTRUMENT	MODE
A IN	YAMAHA	00
B IN	NORMAL	00
C IN	NORMAL	00

PORT	INSTRUMENT	MODE	
A OUT	YAMAHA	FF	<== *Port A disabled*
B OUT	NORMAL	00	
C OUT	NORMAL	00	
D OUT	NORMAL	00	

MODE	PORT	CHAN	KEYBRD	
P	A+C+	1	KBD.1	<== *Port C wont transmit KBD.1 data*
O P	A+++	2	KBD.2	*because port A disabled*
U P	A+++	3	KBD.3	
T P	++C+	1	KBD.4	<== *Port C transmits normally for KBD.4*
P P	A+++	5	KBD.5	
U P	A+++	6	KBD.6	
T P	A+++	7	KBD.7	
P	A+++	8	KBD.8	

Example 2 - Port enabling

```
PORT  INSTRUMENT  MODE        PORT  INSTRUMENT  MODE
A IN  YAMAHA      00          A OUT YAMAHA      00  <== Port A enabled (not FF)
B IN  NORMAL      00          B OUT NORMAL      FF  <== Ports B,C and D disabled
C IN  NORMAL      00          C OUT NORMAL      FF
                              D OUT NORMAL      FF

 MODE PORT CHAN   KEYBRD
   P  ABCD   1    KBD.1  <== Ports B,C,D transmit normally because port A enabled
O  P  A+++  2    KBD.2
U  P  +BCD  1    KBD.3  <== Ports B,C,D wont transmit
T  P  A+++  4    KBD.4
P  P  A+++  5    KBD.5
U  P  A+++  6    KBD.6
T  P  A++D  7    KBD.7  <== Port D transmits normally because port A enabled
   P  ++++  8    KBD.8
```

Yamaha DX-7

Transmit aftertouch for **Yamaha DX-7** only transmits as control 3 if port mode set to D7 (instead of 00). This should only be used on original DX-7s. If a DX-7 has had a retrofit, do not change the port mode.

Synthaxe

The **Synthaxe** guitar synthesizer is connected to an input port only, and has the following modes . . .

With the low digit = 0, the Synthaxe vibrato arm affects control no. 1;
if it is = 1, the vibrato arm does pitch bend.
With the high digit = 0, all strings damp together;
if it is = 1, independent string damping.

Here is a possible setup . . .

```
PORT  INSTRUMENT  MODE        PORT  INSTRUMENT  MODE
A IN  SYNTHAXE    01          A OUT FAIRLIGHT   00  <= Synthaxe vibrato
B IN  YAMAHA      00          B OUT ROLAND      00      arm does pitch bend
C IN  CASIO       00          C OUT NORMAL      FF  <= Output port disabled
                              D OUT YAMAHA      D7  <= Old DX-7s only
```

The lower half of SHEET 2 display has port/channel/FAIRLIGHT keyboards assigment, ECHO, and output MODE.

Any port/MIDI channel combination may be patched to any of the 8 FAIRLIGHT keyboards and vice versa. This allows some rather unique patching.

It is possible to receive MIDI from one or more ports on the same MIDI channel and route the data to different CMI keyboards. This is mixing and filtering MIDI data.

Example

```
 ECHO PORT CHAN   KEYBRD
   OFF A++   1    KBD.1     Channel 1 MIDI data from ports A,B,C
I  OFF +B+   1    KBD.2     routed to CMI keyboards 1 to 3
N  OFF ++C   1    KBD.3
P  OFF +++   4    KBD.4     No MIDI goes to keyboards 4 to 7
U  OFF +++   5    KBD.5
T  OFF +++   6    KBD.6
   OFF +++   7    KBD.7
   OFF ABC  12    KBD.8     Channel 12 MIDI from all input ports
                            goes to keyboard 8
```

When outputting MIDI, data from one CMI keyboard may be sent to one or more ports simultaneously.

Example

```
 MODE  PORT  CHAN   KEYBRD
   P    ABCD    1    KBD.1   <== Keyboard 1 MIDI sent out all four ports
O  P    A+++    2    KBD.2
U  P    +B+D   16    KBD.3   <== Keyboard 3 MIDI sent out channel 16, port B and D
T  P    A+++    4    KBD.4
P  P    A+++    5    KBD.5
U  P    A+++    6    KBD.6
T  P    A+++    7    KBD.7
   P    ++++    8    KBD.8   <== No MIDI data is sent from Keyboard 8
```

ECHO (on input) means key data received will be transmitted to output unaffected. Note that data goes via the relevant CMI keyboard. This allows MIDI data to be received on one channel and be transmitted out another channel . . .

```
 ECHO  PORT  CHAN   KEYBRD   MODE  PORT  CHAN   KEYBRD
   ON   A++     1    KBD.1     P    ++C+   16    KBD.1  <== data received on
I  OFF  A++     2    KBD.2   O P    A+++    2    KBD.2      channel 1, port A
N  OFF  A++     3    KBD.3   U P    A+++    3    KBD.3      echoed to
P  OFF  A++     4    KBD.4   T P    A+++    4    KBD.4      Channel 16, port C
U  OFF  A++     5    KBD.5   P P    A+++    5    KBD.5
T  OFF  A++     6    KBD.6   U M    A+++    6    KBD.6  <== keyboard 6 is in
   OFF  A++     7    KBD.7   T P    A+++    7    KBD.7      MONO mode
   OFF  A++     8    KBD.8     P    A+++    8    KBD.8
```

Note that playing the music keyboard and using the master keyboard command Kn<return> or slave keyboard command KSn<return> (where n = 1 to 8) can transmit MIDI out different CMI keyboards. For example, type K7<return> on any display page. Playing the music keyboard will transmit MIDI out the channel and ports patched to KBD.7. This function is for convenience and saves having to re-patch on Page I. Also affected is the way received MIDI data is played by the CMI. MIDI data patched to CMI KBD.1 will be played by whichever keyboard the master keyboard is switched to on Page 3. Similarly, MIDI data patched to KBD.2 plays whichever keyboard the slave keyboard is switched to on Page 3.

MODE (on output) can be POLY(phonic i.e., chords) or MONO(phonic i.e., one note at a time). You must set all 8 keyboards to Mono when using Page R - Real Time Composer. This prevents notes from "sticking on" in MIDI equipment that does not recognize MIDI note OFF data.

Note that POLY/MONO mode here is not the same thing as the MIDI specification OMNI/POLY/MONO mode.

To **PATCH** the following parameters tab to the appropriate field . . .

To switch individual ECHO on or off, type ON or OFF<SET>.
To patch any port to a CMI keyboard, type <port letters><SET>.
To disconnect a port from a CMI keyboard, type X<SET>.
To patch a MIDI channel to a CMI keyboard type <channel><SET>.
To set MODE type P or M<SET>. For PAGE R set all keyboards to Mono.
. . . where channel = 1-16, port letters = A-C for input, A-D for output.

Because patching is extremely flexible, care must be taken otherwise unexpected results will occur ! !

Stopping Page R

When stopping a Page R song, it is preferable to type <CTRL-ESC> rather than S<return>. Typing <CTRL-ESC> will transmit "notes-off" for all current "notes-on". Also transmitted are "all-notes-off" frames to all ports and channels. This prevents notes hanging on.

SHEET 3 - MIDI/FAIRLIGHT CONTROL/SWITCH NUMBER CONFIGURATION

The CMI's 64 controls and switches are split into groups of eight. Port and MIDI channel numbers can be patched to banks of CMI controls and switches. Patching is the same as previously described. Remember that if outputting control/switch data from the CMI to other MIDI devices, controls 7 to 64 and switches 6 to 64 are only variable from Page C - Music Composition Language.

Note that MIDI pitchbend is received/transmitted as separate data (not a control) so it is unaffected by control/switch patching.

SHEET 4 - DEBUG

Used for diagnostic purposes. No musical use. You may see the actual MIDI data, represented as hexadecimal numbers (base 16), received by the CMI in the FRAME BUFFER box.

SAVING MIDI CONFIGURATION FILES

Page I MIDI setup can be saved to disk as a .MC file allowing instant recall of complex MIDI patches. Some SMPTE parameters are saved also. These are:

time code reader frame rate
time code generator frame rate
drop frame switch rate
jam sync switch rate.

To save a MIDI setup, type

```
SAVE FILENAME.MC<return>
```

When the file has been saved, its name will appear in the **CFG:** box at the bottom left-hand of screen. Note that you must type the **.MC** part of the filename. Otherwise, it will be assumed that **filename** is a voice and will probably display the error message **FILENAME NOT LOADED.**

To load a previously saved MIDI file, type

```
LOAD FILENAME.MC<return>
```

The filename will appear in the **CFG:** box as previously described. Whenever you load a .MC file, it will turn off the time code reader and generator.

To reset Page I, save the default (normal) setting as a file. Then, later on, load that same file when you want to reset Page I.

Page I MIDI configuration files (ending in .MC) can only be loaded or saved from Page I.

Page 2 and Page L display files with the .MC suffix.

APPENDIX A - CMI customization of MIDI devices

ROLAND ignores ALL NOTES OFF data received

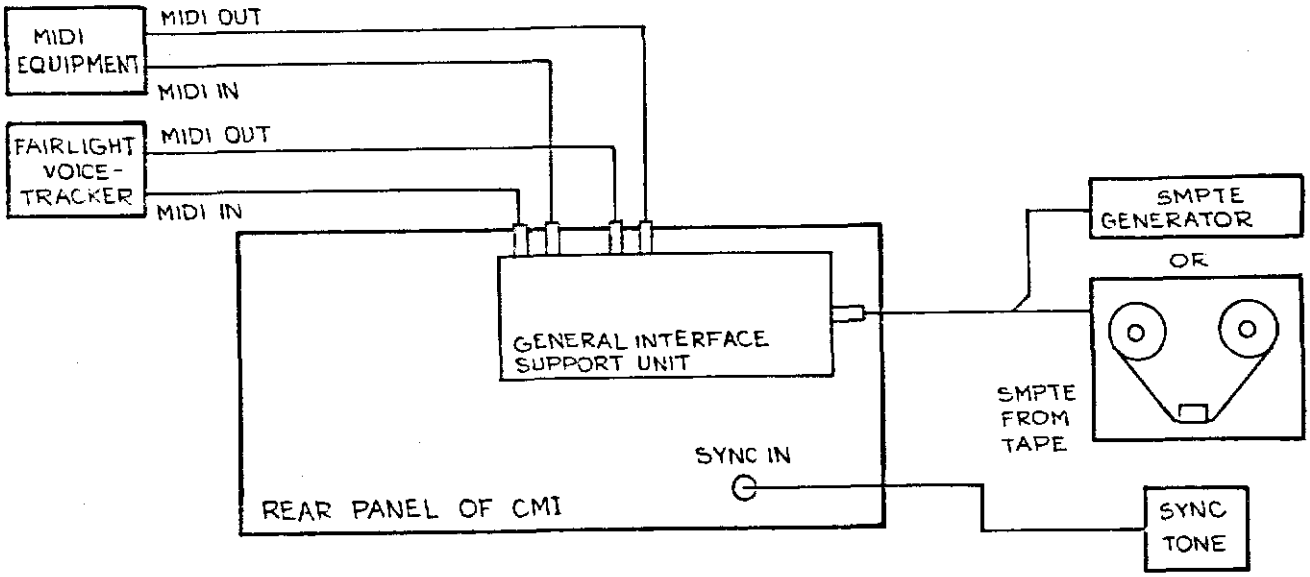
YAMAHA accepts control 3 as aftertouch data. Also accepts normal after touch data

transmits aftertouch as control 3 only if PORT MODE = D7

CASIO transmits LOCAL CONTROL ON instead of ALL NOTES OFF

NORMAL no customization

GENERAL INTERFACE CONNECTION DIAGRAM



AT PRESELECTED SMPTE CLOCK TIME PAGE R WILL START AT A TEMPO DETERMINED BY EITHER

i PAGE R SPEED SETTING
ii EXTERNAL SYNC TONE FROM OSCILLATOR
iii SYNC TONE ALREADY RECORDED ONTO TAPE