S1000 MIDI Exclusive Communication
S1000
S900

General Information About the S1000
The S1000 sampler is the successor to the S900 and bears some similarity to it. Program, keygroup and sample header data is stored in blocks all of the same size (about 150 bytes). There are about 480 blocks available.

A program consists of one block containing common data and then one block per keygroup, containing data unique to its key-span. Unlike the S900, each keygroup has four velocity zones which behave rather like four adjacent keygroups but in the velocity domain, each containing data unique to its velocity range. Each velocity zone names one sample.

A sample consists of a header which uses one block and the sample data itself which is stored in separate memory of minimum size 1 MB words and max 4 MB words. (Other samplers in the family may have 16 MB words max) Unlike the S900 the sample header contains information about several loops and the looping point (of each loop) is independent of the end of the sample. Samples data is presented in 16-bit straight binary form. (peak levels are 0 and FFFFh)

The S1000 has two sample inputs and can sample in stereo. At this stage it appears that this has few consequences for an external editor as the stereo concept ceases to exist once the sampling process has finished; the stereo pair of samples can be treated as two separate samples. However, there may be some internal stereo pair editing functions which operate if the sample names have the last two characters of their names as "-L" and "-R"

The S1000 is capable of running several programs simultaneously and this operates on the basis of allowing more than one program to share the same (MIDI) program number. MIDI play commands (note on/off, bend etc.) operate on all resident programs which have the same program number as the S1000's selected program number. The S1000 has 16 real-time allocated internal voices which are assigned by program data to one of 8 physical mono outputs and/or stereo outputs with left/right pan control and a mono echo send.

The S1000 uses custom LSI devices to perform sample playback, envelope control, channel routing etc. Only forward sample playback and looping is possible. Resident samples can be "resampled" at a new equivalent rate for retuning, the appropriate interpolation or decimation being carried out at high speed by special hardware.

The user interface consists of a 240x64 graphic screen, two S900-like rotary encoders and a similar set of function and numeric keys. In addition, there are 8 "soft" function keys under the screen. Where possible program and sample parameters are presented in logical groups on the same page and graphic images are used to aid programming (e.g. sample plots and ADSR graphs). There are about 35 screen "pages" in all which could perhaps be reduced to about 8 pages on a full CRT. Functions such as sample amplitude normalize, time reverse, cross-fade looping, visual loop point matching and real-time parameter editing are provided.

The Following MIDI Exclusive Functions Will Be Provided:
1) Standard MIDI Sample Dump

This conforms to the MIDI 1.0 Addendum Revision 3.3 and includes retransmission of data packets with checksum errors. Any legal number of sample data bits can be received (but will be truncated or padded to 16 bits). The sample NUMBER is treated as follows: on transmit, the number is simply that of the order in which the sample is stored in the S1000 at the time. On receive, the sample is stored as the next in memory.

It is given a default name of "MIDI nnnnn " where nnnnn is the received number in decimal form. If the name formed this way is the name of an existing sample, that sample will be replaced by the new sample. Note that the S1000 has only two possible sampling rates: 44100 and 22050 KHz. The S1000 sample header created on receipt of a standard sample dump has a tuning offset parameter which is adjusted to compensate for sampling rates other than these. Note also that up to 63 (dummy) sample words may be inserted at the beginning of the sample to conform with S1000 internal requirements.

2) Akai S1000/S1100 Exclusive Messages

The first 5 bytes of all S1000 messages have the form:
F0h Exclusive
47h Akai
cc Exclusive Channel (0-127)
ff Function Code
48h S1000

Unless stated, groups of bytes in messages represent concatenated 7-bit sections of a data word, LSB first

The functions provided are:

> denotes from S1000, < denotes to S1000
ff mnem. direction
00h RSTAT < request S1000 status
01h STAT > S1000 status report
02h RPLIST < request list of resident program names
03h PLIST > list of resident program names
04h RSLIST < request list of resident sample names
05h SLIST > list of resident sample names
06h RPDATA < request program common data
07h PDATA <> program common data
08h RKDATA < request keygroup data
09h KDATA <> keygroup data
0Ah RSDATA < request sample header data
0Bh SDATA <> sample header data
0Ch RSPACK < request sample data packet(s)
0Dh ASPACK < accept sample data packet(s)
0Eh RDDATA < request drum input settings
0Fh DDATA <> drum input settings
10h RMDATA < request miscellaneous data
11h MDATA <> miscellaneous data
12h DELP < delete program and its keygroups
13h DELK < delete keygroup
14h DELS < delete sample header and data
15h SETEX < set S1000 exclusive channel
16h REPLY > S1000 command reply (error or ok)
1Dh CASPACK < corrected ASPACK

Request S1000 Status

F0, 47, cc, RSTAT, 48, F7

S1000 status report

F0, 47, cc, STAT, 48,
vv,VV S1000 software version VV.vv
bb,bb maximum number of program, keygroup, sample header blocks

fb,fb number of blocks free
ww,ww,ww,ww maximum number of sample words
fw,fw,fw,fw number of words free
ec current exclusive channel setting
F7 eox

Request List of Resident Program Names

F0, 47, cc, RPLIST, 48, F7

List of Resident Program Names

F0, 47, cc, PLIST, 48,
pp,pp number of resident programs
12 bytes program 1 name (in non-ascii form - see below)
12 bytes program 2 name
... etc.
F7 eox

Request List of Resident Sample Names

F0, 47, cc, RSLIST, 48, F7

List of Resident Sample Names

F0, 47, cc, SLIST, 48,
ss,ss number of resident samples
12 bytes sample 1 name
12 bytes sample 2 name
...etc.
F7 eox

Request Program Common Data

F0, 47, cc, RPDATA, 48,
pp,pp program number
F7 eox
If the program number is higher than the highest program in the S1000, an error message will be given instead of data.

Program Common Data

F0, 47, cc, PDATA, 48,
pp,pp program number
ln,hn first byte of data in low/high nibble form (see below)
ln,hn second byte
...etc.
F7 eox
This a bidirectional message. In the case of transmitting to the S1000, if the program number is above the highest existing program number, a new program will be created (if sufficient blocks are free -one for the program common block and one for each keygroup as specified by the parameter GROUPS). The created program will have dummy keygroups with unspecified data; the appropriate number of keygroup data messages should be given immediately. If the program name in data is the same as that of any existing program, that program will be deleted first. If the program number is of an existing program, the existing data will be replaced but the parameter GROUPS must be correct. This allows complete freedom to change

parameters - the use of a duplicate program name should be avoided. If either error situation occurs, an error message will be given, otherwise an OK message will be given.

Request Keygroup Data

F0, 47, cc, RKDATA, 48,
pp,pp program number
kk keygroup number
F7 eox
If the keygroup number is higher than the highest keygroup in the program, an error message will be given instead of data.

Keygroup Data

F0, 47, cc, KDATA, 48,
pp,pp program number
kk keygroup number
ln,hn first byte of data in low/high nibble form (see below)
ln,hn second byte
...etc.
F7 eox
This a bidirectional message. In the case of transmitting to the S1000, if the keygroup number is above the highest existing keygroup number, a new keygroup will be created if a block is free, otherwise the existing keygroup will be replaced. The use of program number 255 is a special case where the keygroup data will be installed in program previously created. This avoids the need to read the program list to find out what number was allocated to that program. If there are no free blocks for a new keygroup, an error message will be given.

Request Sample Data Header

F0, 47, cc, RSDATA, 48,
ss,ss sample number
F7 eox
If the sample number is higher than the highest sample in the S1000, an error message will be given instead of data.

Sample Header Data

F0, 47, cc, SDATA, 48,
ss,ss sample number
ln,hn first byte of data in low/high nibble form (see below)
ln,hn second byte
...etc.
F7 eox
This a bidirectional message. In the case of transmitting to the S1000, if the sample number is above the highest existing sample number, a new sample will be created (if a blocks is free and the sample length specified in the header is not too great) and this should be followed by the appropriate sample data packets. If the name in the data is the same as that of an existing sample, it will be deleted first (preceeded by a standard dump WAIT message). If the sample number is of an existing sample, the existing data will be replaced but the sample length parameter SLNGTH must be correct. This allows complete freedom to change parameters - the use of a duplicate name should be avoided. If a new sample cannot be created, an error message will be given, otherwise an OK message will be given. On receipt of a sample header which creates a new sample, the S1000 will behave as if it has received a standard MIDI dump header and if standard MIDI dump is to be used, the data packets should follow immediately. Alternatively, the data can be delivered after an ACCEPT PACKETS command which gives greater flexibility.

Request Sample Data Packet(s)

F0, 47, cc, RSPACK, 48,
ss,ss sample number
oo,oo,oo,oo address offset from start of sample
nn,nn,nn,nn number of samples required
ii interval between samples
if interval function:- 0=single sample
1=average
2=peak
F7 eox
If the sample number is higher than the highest sample in the S1000, an error message will be given instead of data. Otherwise data packet transmission will commence and continue as per standard MIDI dump. If the interval number is greater than 1, the samples will be the function of groups of that number of samples, according to the interval function. The number of samples transmitted will be n/i.

Accept Sample Data Packet(s)

F0, 47, cc, ASPACK, 48,
ss,ss sample number
oo,oo,oo,oo address offset from start of sample
nn,nn,nn,nn number of samples to be delivered
F7 eox
If the sample number is higher than the highest sample in the S1000, an error message will be given. Otherwise a standard MIDI dump ACK message will be given and data packet transmission can commence and continue as per standard MIDI dump.

Request Drum Input Data

F0, 47, cc, RDDATA, 48, F7

Drum Input Data

F0, 47, cc, DDATA, 48,
ln,hn first byte of data in low/high nibble form (see below)
ln,hn second byte
...etc.
F7 eox
This is a bidirectional message.

Request Miscellaneous Data

F0, 47, cc, RMDATA, 48, F7

Miscellaneous Data

F0, 47, cc, MDATA, 48,
ln,hn first byte of data in low/high nibble form (see below)
ln,hn second byte
...etc.
F7 eox
This is a bidirectional message.

Delete Program and its Keygroups

F0, 47, cc, DELP, 48,

pp,pp program number
F7 eox

Delete Keygroup

F0, 47, cc, DELK, 48,
pp,pp program number
kk keygroup number
F7 eox

Delete Sample Header and Data

F0, 47, cc, DELS, 48,
ss,ss sample number
F7 eox
If the argument in any of the delete commands exceeds the maximum, an error message will be given.

Set S1000 Exclusive Channel

F0, 47, cc, SETEX, 48, F7
The S1000 exclusive channel will be set to cc.

S1000 Command Reply

F0, 47, cc, REPLY, 48,
mm reply message: -0=ok 1=error
F7 eox

The following are extracts from assembler files showing the data block structures of program, keygroup, sample-header, drum and miscellaneous files. All bytes are transmitted in LOW-NIBBLE/HIGH-NIBBLE form. Note that names (always 12 characters) in the S1000, including PLIST and SLIST are NOT in ASCII form; they are coded thus:

byte	ASCII equiv
0-9	"0"-"9"
10	" "
11-36	"A"-"Z"
37	"#"
38	"+"
39	"-"
40	"."

Bytes in name fields must be limited to this range.

Program Common Header Block (PDATA)

```
PRIDENT DB 1 ;1=PROGRAM HEADER BLOCK IDENTIFIER
KGRP1@ DW ? ;1st KEYGROUP BLOCK ADDRESS (INTERNAL USE)
PRNAME DB 12 DUP (?) ;NAME
PRGNUM DB ? ;MIDI PROGRAM NUMBER (0-127)
PMCHAN DB ? ;MIDI CHANNEL (0-15. FFh=OMNI)
POLYPH DB ? ;POLYPHONY (1-16)
PRIORT DB ? ;PRIORITY (0=LOW 1=NORMAL 2=HIGH 3=HOLD)
PLAYLO DB ? ;PLAY-RANGE LOW (24-127 = C0-G8)
PLAYHI DB ? ;PLAY-RANGE HIGH (24-127 = C0-G8)
OSHIFT DB ? ;PLAY OCTAVE (KEYBOARD) SHIFT (+/-2)
```

```
OUTPUT DB ? ;OUTPUT NUMBER (0-7. FFh=OFF)
STEREO DB ? ;LEFT AND RIGHT LEVEL (0-99)
PANPOS DB ? ;LEFT/RIGHT BALANCE (+/-50)
PRLOUD DB ? ;BASIC LOUDNESS (0-99)
V_LOUD DB ? ;VELOCITY>LOUDNESS (+/-50)
K_LOUD DB ? ;KEY>LOUDNESS (+/-50)
P_LOUD DB ? ;PRESSURE>LOUDNESS (+/-50)
PANRAT DB ? ;PAN LFO RATE (0-99)
PANDEP DB ? ;PAN DEPTH (0-99)
PANDEL DB ? ;PAN LFO DELAY (0-99)
K_PANP DB ? ;KEY>PAN POSITION (+/-50)
LFORAT DB ? ;LFO SPEED (0-99)
LFODEP DB ? ;LFO FIXED DEPTH (0-99)
LFODEL DB ? ;LFO DELAY (0-99)
MWLDEP DB ? ;MODWHEEL>LFO DEPTH (0-99)
PRSDEP DB ? ;PRESSURE>LFO DEPTH (0-99)
VELDEP DB ? ;VELOCITY>LFO DEPTH (0-99)
B_PTCH DB ? ;BENDWHEEL>PITCH (0-12 SEMITONES)
P_PTCH DB ? ;PRESSURE>PITCH (+/-12 SEMITONES)
KXFADE DB ? ;KEYGROUP CROSSFADE (0=OFF 1=ON)
GROUPS DB ? ;NUMBER OF KEYGROUPS (1-99)
TPNUM DB ? ;TEMPORARY PROGRAM NUMBER (INTERNAL USE)
TEMPER DB 12 DUP (?) ;KEY TEMPERAMENT (+/-25 CENTS) C,C#,D,D# etc
ECHOUT DB ? ;ECHO OUTPUT LEVEL (0=OFF 1=ON)
MW_PAN DB ? ;MOD WHEEL PAN AMOUNT (+/-50)
COHERE DB ? ;SAMPLE START COHERENCE (0=OFF 1=ON)
DESYNC DB ? ;LFO DE-SYNC (0=OFF 1=ON)
PLAW DB ? ;PITCH LAW (0=LINEAR)
VASSOQ DB ? ;VOICE ASSIGN ALGORITHM (0=OLDEST 1=QUIETEST)
SPLOUD DB ? ;SOFT PEDAL LOUDNESS REDUCTION (0-99)
SPATT DB ? ;SOFT PEDAL ATTACK STRETCH (0-99)
SPFILT DB ? ;SOFT PEDAL FILTER CLOSE (0-99)
PTUNO DW ? ;TUNE OFFSET CENT:SEMI (+/-50.00 fraction is binary)
K_LRAT DB ? ;KEY>LFO RATE (+/-50)
K_LDEP DB ? ;KEY>LFO DEPTH (+/-50)
K_LDEL DB ? ;KEY>LFO DELAY (+/-50)
VOSCL DB ? ;VOICE OUTPUT SCALE. (0=-6DB, 1=0DB, 2=+12DB)
VSSCL DB ? ;STEREO OUTPUT SCALE. (0=0DB, 1=+6DB)
```

Keygroup Block (KDATA)

```
;KEYGROUP COMMON
KGIDENT DB 2 ;2=KEYGROUP BLOCK IDENTIFIER
NXTKG@ DW ? ;NEXT KEYGROUP BLOCK ADDRESS (INTERNAL USE)
LONOTE DB ? ;KEYRANGE LOW (24-127 = C0-G8)
HINOTE DB ? ;KEYRANGE HIGH (24-127 = C0-G8)
KGTUNO DW ? ;TUNE OFFSET CENT:SEMI (+/-50.00 fraction is binary)
FILFRQ DB ? ;BASIC FILTER FREQUENCY (0-99)
K_FREQ DB ? ;KEY>FILTER FREQ (+/-24 SEMI-TONES/OCTAVE)
V_FREQ DB ? ;VELOCITY>FILTER FREQ (+/-50)
P_FREQ DB ? ;PRESSURE>FILTER FREQ (+/-50)
E_FREQ DB ? ;ENVELOPE>FILTER FREQ (+/-50)
ATTAK1 DB ? ;AMPLITUDE ATTACK (0-99)
DECAY1 DB ? ;AMPLITUDE DECAY (0-99)
SUSTN1 DB ? ;AMPLITUDE SUSTAIN LEVEL (0-99)
RELSE1 DB ? ;AMPLITUDE RELEASE (0-99)
V_ATT1 DB ? ;VELOCITY>AMP ATTACK (+/-50)
```

```
V_REL1 DB ? ;VELOCITY>AMP RELEASE (+/-50)
O_REL1 DB ? ;OFF VEL.>AMP RELEASE (+/-50)
K_DAR1 DB ? ;KEY>DECAY & RELEASE (+/-50)
ATTAK2 DB ? ;FILTER ATTACK (0-99)
DECAY2 DB ? ;FILTER DECAY (0-99)
SUSTN2 DB ? ;FILTER SUSTAIN LEVEL (0-99)
RELSE2 DB ? ;FILTER RELEASE (0-99)
V_ATT2 DB ? ;VELOCITY>FILTER ATTACK (+/-50)
V_REL2 DB ? ;VELOCITY>FILTER RELEASE (+/-50)
O_REL2 DB ? ;OFF VEL.>FILTER RELEASE (+/-50)
K_DAR2 DB ? ;KEY>DECAY & RELEASE (+/-50)
V_ENV2 DB ? ;VELOCITY>FILTER ENVELOPE OUTPUT (+/-50)
E_PTCH DB ? ;ENVELOPE>PITCH (+/-50)
VXFADE DB ? ;VELOCITY ZONE CROSSFADE (0=OFF 1=ON)
VZONES DB ? ;NUMBER OF VELOCITY ZONES IN USE (NOT USED)
LKXF DB ? ;CALCULATED LEFT KEY CROSSFADE FACTOR (INTERNAL)
RKXF DB ? ;CALCULATED RIGHT KEY CROSSFADE FACTOR (INTERNAL)

;VELOCITY ZONE 1
SNAME DB 12 DUP (@) ;SAMPLE NAME
LOVEL DB ? ;VELOCITY RANGE LOW (0-127)
HIVEL DB ? ;VELOCITY RANGE HIGH (0-127)
VTUNO DW ? ;TUNE OFFSET (+/-50.00 fraction is in binary form)
VLOUD DB ? ;LOUDNESS OFFSET (+/-50)
VFREQ DB ? ;FILTER FREQUENCY OFFSET (+/-50)
VPANO DB ? ;PAN OFFSET (+/-50)
ZPLAY DB ? ;LOOP IN RELEASE (0=as sample 1-4 see below)
LVXF DB ? ;LOW VELOCITY CROSSFADE FACTOR (INTERNAL USE)
HVXF DB ? ;HIGH VELOCITY CROSSFADE FACTOR (INTERNAL USE)
SBADD DW ? ;CALCULATED SAMPLE HEADER BLOCK ADDRESS (INTERNAL)

ZBYTES EQU $-SNAME ;BYTES PER ZONE
;VELOCITY ZONES 2-4
SNAME2 DB ZBYTES*3 DUP (?) ;IDENTICAL TO ZONE 1

;MORE KEYGROUP COMMON
KBEAT DB ? ;FIXED RATE DETUNE (BYTE)
AHOLD DB ? ;ATTACK HOLD UNTIL LOOP
;MORE VELOCITY ZONE ITEMS
CP1 DB ?
CP2 DB ?
CP3 DB ?
CP4 DB ? ;CONSTANT PITCH FOR EACH VELOCITY ZONE (0=TRACK 1=CONST)
VZOUT1 DB ?
VZOUT2 DB ?
VZOUT3 DB ?
VZOUT4 DB ? ;OUTPUT NUMBER OFFSET FOR EACH VELOCITY ZONE (0-7)
VSS1 DW ?
VSS2 DW ?
VSS3 DW ?
VSS4 DW ? ;VELOCITY>SAMPLE START (+/-9999)

;MORE KEYGROUP COMMON
KV_LO DB ? ;VELOCITY>LOUDNESS OFFSET(+/-50)

;ZPLAY:- TYPE OF SAMPLE PLAYBACK VALUES:-
```

```
; 0 - AS DEFINED BY SAMPLE HEADER
; 1 - NORMAL LOOPING
; 2 - LOOP UNTIL RELEASE
; 3 - NO LOOPING
; 4 - PLAY TO SAMPLE END
```

Sample Header Block (SDATA)

```
SHIDENT DB 3 ;3=SAMPLE HEADER BLOCK IDENTIFIER
SBANDW DB ? ;BANDWIDTH (0=10KHZ 1=20 KHZ)
SPITCH DB ? ;ORIGINAL PITCH (24-127 = C0-G8)
SHNAME DB 12 DUP (?) ;NAME (SAME POSITION AS PROGRAM)
SSRVLD DB ? ;SAMPLE RATE SSRATE VALID (80H=YES)
SLOOPS DB ? ;NUMBER OF LOOPS (INTERNAL USE)
SALOOP DB ? ;FIRST ACTIVE LOOP (INTERNAL USE)
DB ? ;SPARE BYTE
SPTYPE DB ? ;PLAYBACK TYPE (SEE BELOW)
STUNO DW ? ;TUNE OFFSET CENT:SEMI (+/-50.00)
SLOCAT DW ?,? ;DATA ABSOLUTE START ADDRESS
SLNGTH DW ?,? ;DATA LENGTH (NUMBER OF SAMPLES)
SSTART DW ?,? ;PLAY RELATIVE START ADDRESS
SMPEND DW ?,? ;PLAY RELATIVE END ADDRESS

;FIRST LOOP
LOOPAT DW ?,? ;RELATIVE LOOP POINT (BITS 0-5 ARE TREATED AS 1)
LLNGTH DW ?,?,? ;LOOP LENGTH (BINARY)FRACTION:INT.LOW:INT. HIGH
LDWELL DW ? ;DWELL TIME (0=NO LOOP 1-9998=mSEC 9999=HOLD)

LBYTES EQU $-LOOPAT ;BYTES PER LOOP

;LOOPS 2-8
LOOP2 DW LBYTES*7 DUP (0) ;SAME AS LOOP 1

;MORE SAMPLE COMMON
SSPARE DB ?,? ;SPARE BYTES USED INTERNALLY
SSPAIR DW ? ;ADDRESS OF STEREO PARTNER (INTERNAL USE)
SSRATE DW ? ;SAMPLE RATE IN HZ
SHLTO DB ? ;HOLD LOOP TUNE OFFSET (+/-50 cents)

;TYPE OF PLAYBACK VALUES:-
; 0 - NORMAL LOOPING
; 1 - LOOP UNTIL RELEASE
; 2 - NO LOOPING
; 3 - PLAY TO SAMPLE END
```

DRUM TRIGGER UNIT BLOCK (DATA IS FOR 2 UNITS) (DDATA)

```
;UNIT 1

D1OPER DB ? ;UNIT 1 IN OPERATION (0=OFF 1=ON)
D1EXCH DB ? ;UNIT 1 EXCLUSIVE CHANNEL (0-15)
D1THRU DB ? ;UNIT 1 MIDI THRU ENABLE (0=OFF 1=ON)
DRNAME DB 12 DUP (?) ;NAME IN SAME PLACE AS PROGS/SAMPS

;INPUT 1 OF UNIT 1
DCHAN DB ? ;DRUM MIDI CHANNEL (0-15)
DNOTE DB ? ;DRUM MIDI NOTE (24-127 = C0-G8)
DSENS DB ? ;DRUM SENSITIVITY (0-127)
```

```
DTRIG DB ? ;DRUM TRIGGER THRESHOLD (0-127)
DVCRV DB ? ;DRUM VELOCITY CURVE (0-7)
DCATP DB ? ;DRUM CAPTURE TIME (0-20 mS)
DRCVR DB ? ;DRUM RECOVERY TIME (0-20 mS)
DONTM DW ? ;DRUM ON-TIME (0-999 mS)

DRBYTES EQU $-DU1TAB ;BYTES PER INPUT

;INPUTS 2-8
DB DRBYTES*7 DUP (?) ;SAME AS INPUT 1

DUBYTES EQU $-D1OPER ;BYTES PER UNIT

;UNIT 2
DB DUBYTES DUP (?) ;SAME AS UNIT 1
```

Miscellaneous Data Block (MDATA)

```
BMCHAN DB ? ;BASIC MIDI CHANNEL (0-15) FOR MIDI PROGRAM SELECT
BMOMNI DB ? ;BASIC CHANNEL OMNI (0=OFF 1=ON)
PSELEN DB ? ;MIDI PROGRAM SELECT ENABLE (0=OFF 1=ON)
SELPNM DB ? ;SELECTED PROGRAM NUMBER (0-127)
OMNOVR DB ? ;MIDI PLAY COMMANDS OMNI OVERRIDE (0=OFF 1=ON)
EXCHAN DB ? ;MIDI EXCLUSIVE CHANNEL (0-127)
```

[Reference Section]