DIMENSIONAL SPACE REVERB
Model SRV-330

MIDI Implementation

Roland
Date : Jun. 18. 1993
Version :1.00

1.RECOGNIZED RECEIVE DATA

CHANNEL VOICE MESSAGE

Note On

Status	Second	Third
9nH	kkH	vvH

n = MIDI Channel	0H - FH (ch.1 - ch.16)
kk = Note Number	00H - 7FH (0 - 127)
vv = Velocity	01H - 7FH (1 - 127)

Real time control of parameter is possible by assigning a note number or velocity to that parameter

Control Change

Status	Second	Third
BnH	ccH	vvH

n = MIDI Channel	0H - FH (ch.1 - ch.16)
cc = Control Number	00H - 1FH,20H-3FH (0 - 31,32 - 63) 40H - 5FH (64 - 95)
vv = Control Value	00H - 7FH (0 - 127)

Real time control of parameter is possible by assigning a control changeto that parameter.

Program Change

Status	Second
CnH	ppH

n = MIDI Channel	0H - FH (ch.1 - ch.16)
pp = Program Number	00H - 7FH (prog.1 - prog.128)

The program number of internal memory, that corresponds to the received program number, can be called.

Channel pressure

Status	Second
DnH	vvH

n = MIDI Channel	0H - FH (ch.1 - ch.16)
vv = Pressure Value	00H - 7FH (0 - 127)

Real time control of a parameter is possible by assigning the channel pressure to that parameter.

Pitch Bend Change

Status	Second	Third
EnH	llH	mmH

n = MIDI Channel	0H - FH (ch.1 - ch.16)
ll = Data LSB	00H - 7FH (0 - 127)
mm = Data MSB	00H - 7FH (0 - 127)

Real time control of a parameter is possible by assigning the pitch bender to that parameter.

SYSTEM EXCLUSIVE MESSAGE

Status	
F0H	System Exclusive
F7H	EOX (End of Exclusive)

For more details, please refer to "3.EXCLUSIVE COMMUNICATIONS" and "Roland Exclusive Messages".

2.TRANSMITTED DATA

2.1.BYPASS MESSAGE

Whole messages bypassed, transmitted to MIDI OUT by setting System Parameter 'MIDI THRU' to 'ON'.

2.2.GENERATED MESSAGE

CHANNEL VOICE MESSAGE

Control Change

Status	Second	Third
BnH	ccH	vvH

n = MIDI Channel	0H - FH (ch.1 - ch.16)
cc = Control Number	00H - 1FH (0 - 31), 40H - 5FH (64 - 95)
vv = Control Value	00H - 7FH (0 - 127)

Transmit value of Pedal connecting EXP Pedal JACK with Control Number by setting System Parameter 'TX Exp. Pedal'.

SYSTEM EXCLUSIVE MESSAGE

Status	
F0H	System Exclusive
F7H	EOX (End of Exclusive)

For more details, please refer to "3.EXCLUSIVE COMMUNICATIONS" and "Roland Exclusive Messages".

3.EXCLUSIVE COMMUNICATION

Via Exclusive Messages, SRV-330 can send or receive parameter setting data in conjunction with external MIDI devices.
Exclusive communications of SRV-330 are always conducted under the following One way communication format (shown as the Roland Exclusive Format, type IV).
For more details, please refer to "Roland Exclusive Messages" in this Manual.

For the Model ID, SRV-330 accepts 59H.
For the Device ID, numerals one unit lower than Receive MIDI number are used.

Request Data1 RQ1 (11H)

Byte	Description	
F0H	Exclusive status	
41H	Manufacturer ID (Roland)	
Dev	Device ID	(Dev:00H - 0FH)
59H	Model ID	(SRV-330)
11H	Command ID	(RQ1)
aaH	Address MSB	
bbH	Address	
ccH	Address LSB	
ssH	Size MSB	
ttH	Size	
uuH	Size LSB	
sum	Checksum	
F7H	EOX	(End of System Exclusive)

*SRV-330 dose not transmit this message

Data Set1 DT1 (12H)

Byte	Description	
F0H	Exclusive status	
41H	Manufacturer ID (Roland)	
Dev	Device ID	(Dev:00H - 0FH)
59H	Model ID	(SRV-330)
12H	Command ID	(DT1)
aaH	Address MSB	
bbH	Address	
ccH	Address LSB	
ddH	Data	
:	:	
eeH	Data	
sum	Checksum	
F7H	EOX	(End of System Exclusive)

4.ADDRESS MAPPING OF PARAMETERS

The address and size are displayed under 7-bit hexadecimal notation.

Address	MSB		LSB
Binary	0aaa aaaa	0bbb bbbb	0ccc cccc
7-bit Hexadecimal	AA	BB	CC

Address	MSB		LSB
Binary	0sss ssss	0ttt tttt	0uuu uuuu
7-bit Hexadecimal	SS	TT	UU

Parameter base address

The actual address of each parameter is the start address of the block plus the offset address.

There are two type of the SRV-330 exclusive message. One is an individual parameter communication, the other is a bulk dump communication.

In bulk dump communication, SRV-330 needs EOB(End Of Bulk Mark) end of communication.
In individual parameter communication, One system exclusive message "F0 ... F7" can only have one parameter.

You cannot use any address having "#" for the top address in a system exclusive message.

Address Block Map

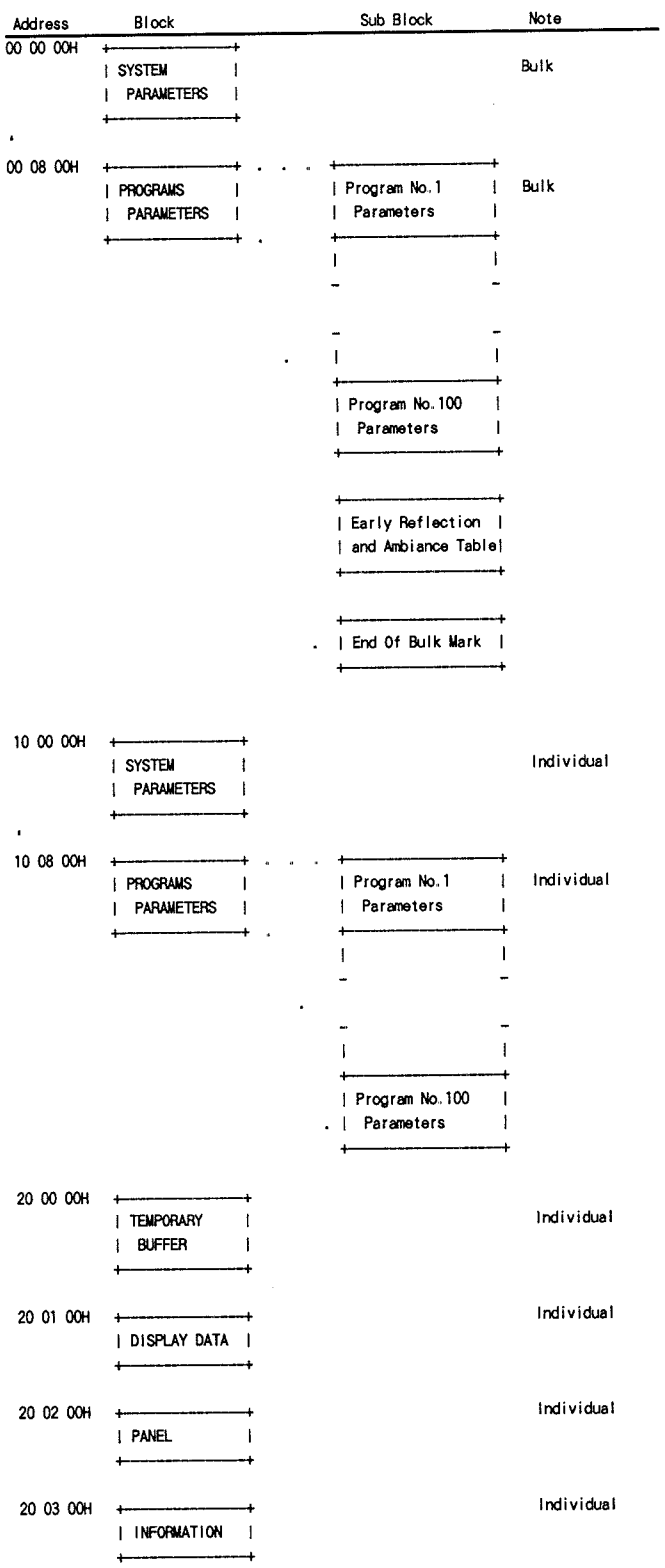


Table-1. SYSTEM PARAMETERS

Offset	Data	Description	
00 00 00H	0000_000mB	OMNI MODE	0:OMNI OFF 1:OMNI ON
00 00 01H	0000_000pB	RX Program Change	0:No 1:Yes
00 00 02H	0ppp_ppppB	TX Exp. Pedal	p: 00H-1FH(0 - 32),40H-5FH(64 - 95) 7FH(OFF)
00 00 03H	0000_000mB	BYPASS Key Def	0:BYPASS 1:MUTE
00 00 04H	0vvv_vvvvB	LCD Contrast	v:00H-64H (0 - 100)
00 01 00H	0000_ppppB	Program Change Map No.1 MSB (0 - 399)	
00 01 01H#	0000_ppppB	Program Change Map No.1	
00 01 02H#	0000_ppppB	Program Change Map No.1	
00 01 03H#	0000_ppppB	Program Change Map No.1 LSB	
:	:		
00 04 7CH	0000_ppppB	Program Change Map No.128 MSB (0 - 399)	
00 04 7DH	0000_ppppB	Program Change Map No.128	
00 04 7EH	0000_ppppB	Program Change Map No.128	
00 04 7FH#	0000_ppppB	Program Change Map No.128 LSB	

Table-2.1. PROGRAMS PARAMETERS

Offset	Description
00 00 00H	Program No.1 Parameters
00 08 00H	Program No.2 Parameters
:	
06 18 00H	Program No.100 Parameters

Table-2.2. Program Parameters

Offset	Data	Description	
00 00 00H	0000_aaaaB	1st. Character of Program Name (ASCII)	LSB
00 00 01H#	0000_aaaaB		MSB
00 00 02H#	0000_aaaaB	2st. Character of Program Name (ASCII)	LSB
00 00 03H#	0000_aaaaB		MSB
: #			
00 00 22H#	0000_0000B		
00 00 23H#	0000_0000B	End of Program Name (FIXED)	
00 00 24H	0000_nnnnB	Algorithm Number t:00H-39H (0 - 57)	LSB
00 00 25H#	0000_nnnnB		MSB
00 00 26H#	0000_0000B	Reserved/Undefined	LSB
00 00 27H#	0000_0000B		MSB
00 00 28H	0000_nnnnB	1st. Parameter Number	LSB
00 00 29H#	0000_nnnnB	1st. Parameter Number	MSB
00 00 2AH#	0000_vvvvB	1st. Parameter Value	LSB
00 00 2BH#	0000_vvvvB	1st. Parameter Value	
00 00 2CH#	0000_vvvvB	1st. Parameter Value	
00 00 2DH#	0000_vvvvB	1st. Parameter Value	MSB
00 00 2EH	0000_nnnnB	2nd. Parameter Number	LSB
00 00 2FH#	0000_nnnnB	2nd. Parameter Number	MSB
00 00 30H#	0000_vvvvB	2nd. Parameter Value	LSB
00 00 31H#	0000_vvvvB	2nd. Parameter Value	
00 00 32H#	0000_vvvvB	2nd. Parameter Value	
00 00 33H#	0000_vvvvB	2nd. Parameter Value	MSB
: :			
00 06 4CH	0000_nnnnB	135th Parameter Number	LSB
00 06 4DH#	0000_nnnnB	135th Parameter Number	MSB
00 06 4EH#	0000_vvvvB	135th Parameter Value	LSB
00 06 4FH#	0000_vvvvB	135th Parameter Value	
00 06 50H#	0000_vvvvB	135th Parameter Value	
00 06 51H#	0000_vvvvB	135th Parameter Value	MSB

Table-2.2.1. Algorithm Number

Algorithm Name	Number
Reverb 1 (Type A)	0
Reverb 1 (Type B)	1
Reverb 1 (Type C)	2
Reverb 1 (Type E)	3
Reverb 2 (Type A, Type B)	4
Reverb 2 (Type A, Type C)	5
Reverb 2 (Type A, Type E)	6
Reverb 2 (Type B, Type A)	7
Reverb 2 (Type B, Type C)	8
Reverb 2 (Type B, Type E)	9
Reverb 3 (Type A)	10
Reverb 3 (Type B)	11
Reverb 3 (Type C)	12
Reverb 3 (Type E)	13
Reverb 4 (Type A)	14
Reverb 4 (Type B)	15
Reverb 4 (Type C)	16
Reverb 4 (Type E)	17
2 x Reverb (Type C)	18
Plate (Type A)	19
Plate (Type B)	20
Reverb (Type A) + Plate (Type A)	21
Reverb (Type A) + Plate (Type B)	22

Reverb (Type B) + Plate (Type A)	23
Reverb (Type B) + Plate (Type B)	24
Reverb (Type C) + Plate (Type A)	25
Reverb (Type C) + Plate (Type B)	26
Reverb (Type E) + Plate (Type A)	27
Reverb (Type E) + Plate (Type B)	28
Reverb (Type A) + Delay	29
Reverb (Type B) + Delay	30
Reverb (Type A) + Plate (Type A) + Delay	31
Delayed Reverb (Type B)	32
Chorus Reverb (Type A)	33
Chorus Reverb (Type B)	34
Chorus Reverb (Type C)	35
Chorus Reverb (Type E)	36
Chorus Reverb (Type D)	37
Chorus Reverb (Type D)	38
Chorus Reverb (Type D)	39
2 x Chorus Reverb (Type C)	40
Phaser Reverb (Type A)	41
Phaser Reverb (Type B)	42
Phaser Reverb (Type C)	43
Phaser Reverb (Type E)	44
Direct Mix Chorus Reverb (Type B)	45
Delay Chorus	46
Phaser Delay	47
3D Early Reflection + Reverb (Type A)	48
3D Early Reflection + Reverb (Type B)	49
3D Early Reflection + Reverb (Type C)	50
3D Early Reflection + Reverb (Type E)	51
3D Early Reflection + Plate (Type A)	52
3D Early Reflection + Plate (Type B)	53
3D Ambience	54
3D Non-linear	55
3D Non-linear Reverb (Type B)	56
Reverb (Type B) 3D Non-linear	57

Note: ‘Reverb (Type A - E)’ shows type of Reverb,
‘Plate (Type A, B)’ shows type of Plate.

Table-2.2.2. Parameter Number

Name	Number	Name	Number
ERLvl	0	EnT3B	37
ERTim	1	EnT4B	38
ERDif	2	EnL1B	39
ERDmp	3	EnL2B	40
RevTm	4	EnL3B	41
PrDly	5	LowFq	42
Dnsty	6	LowGn	43
LoDFq	7	Low_Q	44
LoDmp	8	MidFq	45
HiDFq	9	MidGn	46
HiDmp	10	Mid_Q	47
Depth	11	HigFq	48
Brill	12	HigGn	49
Width	13	Hig_Q	50
RSize	14	DyTmA	51
HFDmp	15	DyFbA	52
TypeA	16	DyLvA	53
Elv1A	17	DyTmB	54
Azm1A	18	DyFbB	55
Elv2A	19	DyLvB	56
Azm2A	20	ChoPD	57
PrDyA	21	ChoRt	58
EnT1A	22	ChoDp	59
EnT2A	23	ChoPh	60
EnT3A	24	GtMod	61
EnT4A	25	GtBal	62
EnL1A	26	GtThr	63
EnL2A	27	GtATm	64
EnL3A	28	GtHTm	65
TypeB	29	GtRTm	66
Elv1B	30	OutLv	67
Azm1B	31	SwMod	68
Elv2B	32	C1Src	69
Azm2B	33	C1Trg	70
PrDyB	34	C1Min	71
EnT1B	35	C1Max	72
EnT2B	36	C2Src	73
C2Trg	74	C5Src	85
C2Min	75	C5Trg	86
C2Max	76	C5Min	87
C3Src	77	C5Max	88
C3Trg	78	[CTL]	89
C3Min	79	[NAM]	90
C3Max	80	[PRM]	91
C4Src	81	BYPAS	92
C4Trg	82		
C4Min	83	EOPL	255 End Of Parameter List
C4Max	84		

Table-2.2.3. Parameter Value

Name	Value	
ERLvl	0 - 100	
ERTim	1 - 100	*Ratio(%)
ERDif	0 - 100	
ERDmp	-360 - 0	*Gain(dB) x 10
RevTm	1 - 10000	*Reverb Time(sec) x 100
PrDly	0 - 800	*Delay Time(ms)
Dnsty	0 - 100	
LoDFq	50 - 4000	*Frequency(Hz)
LoDmp	-360 - 0	*Gain(dB) x 10
HiDFq	4000 - 20000	*Frequency(Hz)
HiDmp	-360 - 0	*Gain(dB) x 10
Depth	0 - 100	
Brill	0 - 100	
Width	0 - 100	
RSize	1 - 100	*Ratio(%)
HFDmp	-360 - 0	*Gain(dB) x 10
TypeA	0:1<-2, 1:NORM, 2:1->2	
Elv1A	-60 - +60	
Azm1A	-180(L180) - +180(R180)	
Elv2A	-60 - +60	
Azm2A	-180(L180) - +180(R180)	
PrDyA	0 - 800	*Delay Time(ms)
EnT1A	1 - 1200(ms)	
EnT2A	1 - 1200(ms)	
EnT3A	1 - 1200(ms)	
EnT4A	1 - 1200(ms)	
EnL1A	0 - 100	
EnL2A	0 - 100	
EnL3A	0 - 100	
TypeB	0:1<-2, 1:NORM, 2:1->2	
Elv1B	-60 - +60	
Azm1B	-180(L180) - +180(R180)	
Elv2B	-60 - +60	
Azm2B	-180(L180) - +180(R180)	
PrDyB	0 - 800	*Delay Time(ms)
EnT1B	1 - 1200(ms)	
EnT2B	1 - 1200(ms)	
EnT3B	1 - 1200(ms)	
EnT4B	1 - 1200(ms)	
EnL1B	0 - 100	
EnL2B	0 - 100	
EnL3B	0 - 100	
LowFq	20 - 2000	*Frequency(Hz)
LowGn	-120 - 120	*Gain(dB) x 10
Low_Q	2:SHELV, 3(0.3) - 100(10.0)	
MidFq	200 - 8000	*Frequency(Hz)
MidGn	-120 - 120	*Gain(dB) x 10
Mid_Q	3(0.3) - 100(10.0)	
HigFq	1500 - 20000	*Frequency(Hz)
HigGn	-120 - 120	*Gain(dB) x 10
Hig_Q	2:SHELV, 3(0.3) - 100(10.0)	
DyTmA	0 - 400	*Delay Time(ms)
DyFbA	-100 - 100	
DyLvA	-100 - 100	
DyTmB	0 - 400	*Delay Time(ms)
DyFbB	-100 - 100	
DyLvB	-100 - 100	
ChoPD	0 - 40	*Delay Time(ms)
ChoRt	1 - 1000	*Rate(Hz) x 100
ChoDp	0 - 100	
ChoPh	0 - 180	*Phase(deg)
GtMod	0:THRU, 1:GATE, 2:DUCK	
GtBal	0 - 100	
GtThr	-300 - 100	*Threshold Level(dB) x 10
GtATm	0 - 10000	*Attack Time(sec) x 1000
GtHTm	0 - 10000	*Hold Time(sec) x 1000

GtRTm	0 - 10000	*Release Time(sec) x 1000
OutLv	0 - 200	
SwMod	0:NORM, 1:TOGGL	
C1Src	-1:OFF, 0:ExPdl, 1:CtlSW, 2:PtBnd, 3:AfTch, 4:Note, 5:Veloc, 6(CC#0) - 101(CC#95)	
C1Trg		*Parameter Number
C1Min		*Depend on 'C1Trg'
C1Max		*Depend on 'C1Trg'
C2Src	-1:OFF, 0:ExPdl, 1:CtlSW, 2:PtBnd, 3:AfTch, 4:Note, 5:Veloc, 6(CC#0) - 101(CC#95)	
C2Trg		*Parameter Number
C2Min		*Depend on 'C2Trg'
C2Max		*Depend on 'C2Trg'
C3Src	-1:OFF, 0:ExPdl, 1:CtlSW, 2:PtBnd, 3:AfTch, 4:Note, 5:Veloc, 6(CC#0) - 101(CC#95)	
C3Trg		*Parameter Number
C3Min		*Depend on 'C3Trg'
C3Max		*Depend on 'C3Trg'
C4Src	-1:OFF, 0:ExPdl, 1:CtlSW, 2:PtBnd, 3:AfTch, 4:Note, 5:Veloc, 6(CC#0) - 101(CC#95)	
C4Trg		*Parameter Number
C4Min		*Depend on 'C4Trg'
C4Max		*Depend on 'C4Trg'
C5Src	-1:OFF, 0:ExPdl, 1:CtlSW, 2:PtBnd, 3:AfTch, 4:Note, 5:Veloc, 6(CC#0) - 101(CC#95)	
C5Trg		*Parameter Number
C5Min		*Depend on 'C5Trg'
C5Max		*Depend on 'C5Trg'
[CTL]		*Value is ignored
[NAM]		*Value is ignored
[PRM]		*Parameter Number
BYPAS	0:OFF, 1:ON	

Note: Refer to "ALGORITHM GUIDE".

Table-2.3. Early Reflection and Ambience Table

Offset	Data	Description
0E 00 00H	0000_eeeeB	ER3D
: #		:
0E 18 3FH#	0000_eeeeB	ER3D
0E 20 00H	0000_aaaaB	AM3D
: #		:
0E 4B 7FH#	0000_aaaaB	AM3D

Table-2.4. End Of Bulk Mark

Offset	Data	Description
0F 7F 7FH	0000_0000B	End Of Bulk Mark

Table-3. TEMPORARY BUFFER

Offset	Data	Description	
00 00 00H	0000_nnnnH	Parameter Number	LSB
00 00 01H#	0000_nnnnH	n:00H-FFH(0-255)	MSB
00 00 02H#	0000_vvvvH	Parameter Value LSB	
00 00 03H#	0000_vvvvH	Parameter Value	
00 00 04H#	0000_vvvvH	Parameter Value	
00 00 05H#	0000_vvvvH	Parameter Value MSB	
		v:0000H-FFFFH(-32768 - 32767)	

Table-4. DISPLAY DATA

Offset	Data	Description
00 00 03H	20H-7FH	1st. Line Displayed Character (ASCII)
00 00 04H#		
00 00 05H#		
: #		
00 00 13H#		
00 00 40H	20H-7FH	Program Number Area Displayed Character (ASCII)
00 00 41H#		
00 00 42H#		
00 00 43H	20H-7FH	2nd. Line Displayed Character (ASCII)
00 00 44H#		
: #		
00 00 53H#		

Table-5. PANEL CONTROL (Write Only)

Offset	Data	Description	
00 00 00H	00ab_cdefB	SW Push	a:FUNC2 b:FUNC3 c:DOWN d:PAGE e:MEMORY f:BYPASS
00 00 01H#	00ab_cdefB	SW Push	a:PROGRAM b:FUNC1 c:UP d:EDIT e:SYSTEM f:EXIT
00 00 02H	000d_vvvvB	Rotary Encoder	d: 0=CW 1=CCW v: Step
00 00 03H	0000_000bB	MUTE ON/OFF	b: 0=OFF 1=ON
00 00 04H	0000_000cB	CONTROL SW	c: 0=OFF 1=ON
00 00 10H	0000_000mB	PANEL LOCK MODE	m: 0=Exit 1=Enter

Table-6. INFORMATION (READ ONLY)

Offset	Data	Description		
00 00 00H	07H (7)	Number of MODEL NAME Character		
00 00 01H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 02H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 03H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 04H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 05H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 06H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 07H#	20H-7FH (32-127)	Character of MODEL NAME (ASCII)		
00 00 08H				
:		Reserved/Undefined		
00 00 0FH				
00 00 10H	20H-7FH (32-127)	Software Version Number (ASCII)		
00 00 11H#	20H-7FH (32-127)	Software Version Number (ASCII)		
:				
00 00 17H#	20H-7FH (32-127)	Software Version Number (ASCII)		
00 00 18H				
:		Reserved/Undefined		
00 00 1FH				
00 00 20H	0abc_defgB	Lch Level Meter LED	a:OVER	b:+6dB
00 00 21H	0abc_defgB	Rch Level Meter LED	c: 0dB	d:-3dB
			e:-6dB	f:-10dB
			g:-20dB	

ROLAND EXCLUSIVE MESSAGES

1. Data Format for Exclusive Messages

Roland's MIDI implementation uses the following data format for all Exclusive messages (type IV):

Byte	Description
F0H	Exclusive Status
41H	Manufacturer ID (Roland)
DEV	Device ID
MDL	Model ID
CMD	Command ID
[BODY]	Main data
F7H	End of exclusive

•MIDI status: F0H, F7H

An Exclusive message must be flanked by a pair of status codes, starting with a Manufacturer ID immediately after F0H (MIDI version 1.0).

•Manufacturer ID: 41H

The Manufacturer ID identifies the manufacturer of a MIDI instrument that sends an Exclusive message. Value 41H represents Roland's Manufacturer ID.

•Device ID: DEV

The Device ID contains a unique value that identifies individual devices in the implementation of several MIDI instruments. It is usually set to 00H–0FH, a value smaller by one than that of a basic channel, but value 00H–1FH may be used for a device with several basic channels.

•Model ID: MDL

The Model ID contains a value that identifies one model from another. Different models, however, may share an identical Model ID if they handle similar data.

The Model ID format may contain 00H in one or more places to provide an extended data field. The following are examples of valid Model IDs, each representing a unique model:

01H
02H
03H
00H, 01H
00H, 02H
00H, 00H, 01H

•Command ID: CMD

The Command ID indicates the function of an Exclusive message. The Command ID format may contain 00H in one or more places to provide an extended data field. The following are examples of valid Command IDs, each representing a unique function:

01H
02H
03H
00H, 01H
00H, 02H
00H, 00H, 01H

•Main data: BODY

This field contains a message to be exchanged across an interface. The exact data size and content will vary with the Model ID and Command ID.

2. Address-mapped Data Transfer

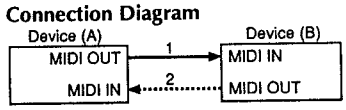
Address mapping is a technique for transferring messages conforming to the data format given in Section 1. It assigns a series of memory-resident records—waveform and tone data, switch status, and parameters, for example, to specific locations in a machine-dependent address space, thereby allowing access to data residing at the address a message specifies.

Address-mapped data transfer is therefore independent of models and data categories. This technique allows use of two different transfer procedures: one-way transfer and handshake transfer.

•One-way transfer procedure (See Section 3 for details.)

This procedure is suited to the transfer of a small amount of data. It sends out an Exclusive message completely independent of the receiving device's status.

Connection Diagram

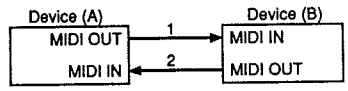


Connection at point 2 is essential for "Request data" procedures. (See Section 3.)

•Handshake-transfer procedure

(This device does not use this procedure)
This procedure initiates a predetermined transfer sequence (handshaking) across the interface before data transfer takes place. Handshaking ensures that reliability and transfer speed are high enough to handle a large amount of data.

Connection Diagram



Connection at points 1 and 2 is essential.

Notes on the above procedures

* There are separate Command IDs for different transfer procedures.

* Devices A and B cannot exchange data unless they use the same transfer procedure, share identical Device ID and Model ID, and are ready for communication.

3. One-way Transfer Procedure

This procedure sends out data until it has all been sent and is used when the messages are so short that answerbacks need not be checked.
For longer messages, however, the receiving device must acquire each message in time with the transfer sequence, which inserts 20 milliseconds intervals.

Types of Messages

Message	Command ID
Request data 1	RQ1 (11H)
Data set 1	DT1 (12H)

•Request data #1: RQ1 (11H)

This message is sent out when there is a need to acquire data from a device at the other end of the interface. It contains data for the address and size that specify designation and length, respectively, of data required.
On receiving an RQ1 message, the remote device checks its memory for the data address and size that satisfy the request.
If it finds them and is ready for communication, the device will transmit a "Data set 1 (DT1)" message, which contains the requested data. Otherwise, the device won't send out anything.

Byte	Description
F0H	Exclusive Status
41H	Manufacturer ID (Roland)
DEV	Device ID
MDL	Model ID
11H	Command ID
aaH	Address MSB
⋮	⋮
⋮	⋮
	LSB
ssH	Size MSB
⋮	⋮
⋮	⋮
	LSB
sum	Check sum
F7H	End of exclusive

* The size of the requested data does not indicate the number of bytes that will make up a DT1 message, but represents the address fields where the requested data resides.

* Some models are subject to limitations in data format used for a single transaction. Requested data, for example, may have a limit in length or must be divided into predetermined address fields before it is exchanged across the interface.

* The same number of bytes comprises address and size data, which, however, vary with the Model ID.

* The error-checking process uses a checksum that provides a bit pattern where the last 7 bits are zero when values for an address, size, and that checksum are summed.

•Data set 1: DT1 (12H)

This message corresponds to the actual data transfer process. Because every byte in the data is assigned a unique address, a DT1 message can convey the starting address of one or more bits of data as well as a series of data formatted in an address-dependent order.

The MIDI standards inhibit non real-time messages from interrupting an Exclusive one. This fact is inconvenient for devices that support a "soft-thru" function. To maintain compatibility with such devices, Roland has limited the DT1 to 256 bytes so that an excessively long message is sent out in separate 'segments'.

Byte	Description
F0H	Exclusive Status
41H	Manufacturer ID (Roland)
DEV	Device ID
MDL	Model ID
12H	Command ID
aaH	Address MSB
⋮	⋮
⋮	⋮
	LSB
ddH	Data MSB
⋮	⋮
⋮	⋮
	LSB
sum	Check sum
F7H	End of exclusive

* A DT1 message is capable of providing only the valid data among those specified by an RQ1 message.

* Some models are subject to limitations in data format used for a single transaction. Requested data, for example, may have a limit in length or must be divided into predetermined address fields before it is exchanged across the interface.

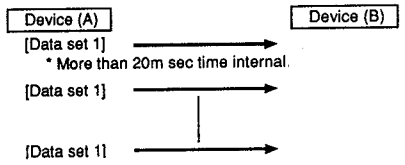
* The number of bytes comprising address data varies from one Model ID to another.

* The error-checking process uses a checksum that provides a bit pattern where the last 7 bits are zero when values for an address, size, and that checksum are summed.

•Example of Message Transactions

• Device A sending data to Device B

Transfer of a DT1 message is all that takes place.



• Device B requesting data from Device A

Device B sends an RQ1 message to Device A.
Checking the message, Device A sends a DT1 message back to Device B.