

# CTK-4200/CTK-5300/LK-211/LK-280 WK-220/WK-225/WK-330/CDP-220R MIDI Implementation

CASIO COMPUTER CO., LTD.

## Contents

<b>I</b>	<b>MIDI Message Overview</b>	<b>5</b>
<b>1</b>	<b>Product Configuration as a MIDI Device</b>	<b>5</b>
1.1	System Section . . . . .	5
1.2	Performance Controller Section . . . . .	5
1.3	Sound Generator Section . . . . .	5
<b>2</b>	<b>Timbre Type Specific Operation</b>	<b>6</b>
<b>3</b>	<b>Conditions that Disable Message Send and Receive</b>	<b>7</b>
<b>II</b>	<b>Channel Message</b>	<b>8</b>
<b>4</b>	<b>Receive Channel</b>	<b>8</b>
<b>5</b>	<b>Send Channel</b>	<b>8</b>
<b>6</b>	<b>Note Off</b>	<b>8</b>
<b>7</b>	<b>Note On</b>	<b>8</b>
<b>8</b>	<b>Polyphonic Key Pressure</b>	<b>8</b>
<b>9</b>	<b>Control Change</b>	<b>9</b>
9.1	Bank Select (00H,20H) . . . . .	9
9.2	Modulation (01H) . . . . .	9
9.3	Data Entry (06H,26H) . . . . .	10
9.4	Volume (07H) . . . . .	10
9.5	Pan (0AH) . . . . .	10
9.6	Expression (0BH) . . . . .	10

9.7	Hold1 (40H) . . . . .	11
9.8	Sostenuto (42H) . . . . .	11
9.9	Soft (43H) . . . . .	11
9.10	Reverb Send (5BH) . . . . .	12
9.11	Chorus Send (5DH) . . . . .	12
9.12	NRPN (62H,63H) . . . . .	12
9.13	RPN (64H,65H) . . . . .	12
<b>10</b>	<b>Mode Message</b>	<b>14</b>
10.1	All Sound Off (78H) . . . . .	14
10.2	Reset All Controllers (79H) . . . . .	14
10.3	All Notes Off (7BH) . . . . .	14
10.4	Omni Off (7CH) . . . . .	14
10.5	Omni On (7DH) . . . . .	14
10.6	Mono (7EH) . . . . .	15
10.7	Poly (7FH) . . . . .	15
<b>11</b>	<b>Program Change</b>	<b>15</b>
<b>12</b>	<b>Channel After Touch</b>	<b>16</b>
<b>13</b>	<b>Pitch Bend</b>	<b>16</b>
<b>III</b>	<b>System Message</b>	<b>17</b>
<b>14</b>	<b>Timing Clock</b>	<b>17</b>
<b>15</b>	<b>Start</b>	<b>17</b>
<b>16</b>	<b>Stop</b>	<b>17</b>
<b>17</b>	<b>Active Sensing</b>	<b>17</b>
<b>18</b>	<b>System Exclusive Message</b>	<b>17</b>
18.1	Universal Real Time System Exclusive Message . . . . .	18
18.2	Instrument-Specific System Exclusive Message . . . . .	20
<b>IV</b>	<b>Instrument-Specific System Exclusive Messages</b>	<b>21</b>
<b>19</b>	<b>Format</b>	<b>21</b>
19.1	Message Classifications . . . . .	21
19.2	Basic Message Structure . . . . .	22

19.3	Format of Each Field . . . . .	22
<b>20</b>	<b>Individual Parameter Operations</b>	<b>29</b>
<b>21</b>	<b>Parameter Set Transfer</b>	<b>29</b>
21.1	Communication Modes . . . . .	29
21.2	One-way Mode Communication Flow . . . . .	30
21.3	Handshake Mode Communication Flow . . . . .	30
<b>V</b>	<b>Parameter List</b>	<b>34</b>
<b>22</b>	<b>Using the Parameter List</b>	<b>34</b>
<b>23</b>	<b>System Parameters</b>	<b>34</b>
23.1	System Information Parameter . . . . .	34
23.2	Data Management Parameter . . . . .	35
<b>24</b>	<b>Patch Parameter</b>	<b>36</b>
24.1	Master Tune Parameter . . . . .	36
24.2	Master Mixer Parameter . . . . .	36
24.3	Part Parameter . . . . .	36
<b>25</b>	<b>Scale Memory Parameter</b>	<b>36</b>
25.1	Scale Memory Octave . . . . .	36
<b>26</b>	<b>All Data Parameter</b>	<b>37</b>
<b>27</b>	<b>Sequence Parameter</b>	<b>37</b>
27.1	Sequence Directory Info . . . . .	37
<b>28</b>	<b>Registration Parameter</b>	<b>37</b>
28.1	Registration Directory Info . . . . .	37
<b>29</b>	<b>Rhythm Parameter</b>	<b>37</b>
29.1	Rhythm Directory Info . . . . .	37
<b>30</b>	<b>Song Parameter</b>	<b>37</b>
30.1	Song Directory Info . . . . .	37
<b>31</b>	<b>Lesson Rec Parameter</b>	<b>37</b>
31.1	Lesson Rec Directory Info . . . . .	37
<b>VI</b>	<b>Parameter Set List</b>	<b>38</b>

<b>32</b>	<b>Parameter Set Table</b>	<b>38</b>
<b>VII</b>	<b>Setting Values and Send/ Receive Values</b>	<b>40</b>
<b>33</b>	<b>Setting Value Tables</b>	<b>40</b>
33.1	Off/On Setting Value Table . . . . .	40
33.2	-64 - 0 - +63 Setting Value Table . . . . .	40
33.3	Pan Setting Value Table . . . . .	40
33.4	Fine Tune Setting Value Table . . . . .	40
33.5	Reverb Time Setting Value Table . . . . .	41
33.6	Chorus Type Setting Value Table . . . . .	41
<b>VIII</b>	<b>MIDI Implementation Notation</b>	<b>42</b>
<b>34</b>	<b>Value Notation</b>	<b>42</b>
34.1	Hexadecimal Notation . . . . .	42
34.2	Binary Notation . . . . .	42

## Part I

# MIDI Message Overview

## 1 Product Configuration as a MIDI Device

As a MIDI device, this Instrument consists of the System Section, Performance Controller Section, and Sound Generator Section described below. Each of these sections can send and receive specific MIDI Messages in accordance with its function.

### 1.1 System Section

The System Section manages the Instrument status and user data. A communication method known as a "bulk dump" can be used for two-way transfer of user data between the Instrument and a computer.

### 1.2 Performance Controller Section

The Performance Controller Section performs keyboard play and controller operations, and generates performance messages in accordance with auto play, etc. Basically, generated performance messages are sent to external destinations while also being transmitted to the Sound Generator Section. The channel number of the sent channel message is in accordance with the Instrument's instrument part number.

**MIDI Output Performance Information** The following describes the performance information that is output and is not output as MIDI signals.

- Output performance
  - Keyboard play and controller operations by the musician
  - Auto accompaniment
  - Recorded song playback
- Non-output performance
  - Demo Songs
  - Song playback
  - Lesson function
  - Music Challenge

### 1.3 Sound Generator Section

The Sound Generator Section mainly performs receive of performance information and sound source setting information. It consists of a common part that does not depend on the channel and a musical instrument part that is independent of each channel.

#### 1.3.1 Sound Generator Common Block

The common block consists of system effects, mixer master control, etc. These can be controlled by general universal system exclusive messages, and the Instrument's system exclusive messages.

### 1.3.2 Instrument Part Block

The instrument part section consists of a total of 32 instrument parts, divided into two groups, named Group A and Group B of 16 instruments each. Each part can perform operations and setting changes using channel messages and Instrument's system exclusive messages. Only Group B can be controlled by external channel messages.

As shown in the following table, there is a fixed relationship between channel message receive channel numbers and instrument parts.

Number	Name	Channel	Assigned Function
00	A01	01	Keyboard (Main)
01	A02	02	Keyboard (Layer)
02	A03	03	Keyboard (Split)
03	A04	04	Keyboard/Recorder (Harmonize)
04	A05	05	Sampling Phrase
05	A06	06	Guide Sound
06	A07	07	Guide Voice
07	A08	08	Metronome
08	A09	09	Auto Accompaniment (Percussion)
09	A10	10	Auto Accompaniment (Drum)
10	A11	11	Auto Accompaniment (Bass)
11	A12	12	Auto Accompaniment (Chord 1)
12	A13	13	Auto Accompaniment (Chord 2)
13	A14	14	Auto Accompaniment (Chord 3)
14	A15	15	Auto Accompaniment (Chord 4)
15	A16	16	Auto Accompaniment (Chord 5)
16	B01	01	MIDI/Auto Performance Functions/Recorder (Keyboard Playback Main)
17	B02	02	MIDI/Auto Performance Functions/Recorder (Keyboard Playback Layer)
18	B03	03	MIDI/Auto Performance Functions/Recorder (Keyboard Playback Split)
19	B04	04	MIDI/Auto Performance Functions/Recorder (Solo 1)
20	B05	05	MIDI/Auto Performance Functions/Recorder (Solo 2)
21	B06	06	MIDI/Auto Performance Functions/Recorder (Solo 3)
22	B07	07	MIDI/Auto Performance Functions/Recorder (Solo 4)
23	B08	08	MIDI/Auto Performance Functions/Recorder (Solo 5)
24	B09	09	MIDI/Auto Performance Functions
25	B10	10	MIDI/Auto Performance Functions
26	B11	11	MIDI/Auto Performance Functions
27	B12	12	MIDI/Auto Performance Functions
28	B13	13	MIDI/Auto Performance Functions
29	B14	14	MIDI/Auto Performance Functions
30	B15	15	MIDI/Auto Performance Functions
31	B16	16	MIDI/Auto Performance Functions

## 2 Timbre Type Specific Operation

The sound source operation performed for a sound generator instrument receive message may depend on the value of the Timbre Type (see "About the Timbre Type" in "11 Program Change") of each part's

operation mode. For details, see the explanation for each message.

### **3 Conditions that Disable Message Send and Receive**

All MIDI message send and receive is temporarily disabled while any one of the following processes is in progress.

- SD memory card operation in progress(LK-280, LK-211, CTK-5300, WK-330, CDP-220R only)
- Recorder recording in progress
- Sampling operation in progress

## Part II

# Channel Message

## 4 Receive Channel

The channel number of the channel message received by each part is shown in the table under “1.3.2 Instrument Part Block”.

## 5 Send Channel

Basically, the MIDI channel of the channel message sent when the Instrument is played coincides with the MIDI channel of the part being played. Note, however, that the MIDI channel of the performance information that corresponds to the keyboard main part depends on the Keyboard Channel setting value.

## 6 Note Off

Message Format: 8nH kkH vvH  
9nH kkH 00H(receive only)

---

n: MIDI Channel Number  
kk: Key Number  
vv: Velocity(Send:40H, Receive:Ignored)

**Send** Sent when the keyboard is played and when play is performed using Auto Accompaniment. The key number changes in accordance with the Octave Shift function.

**Receive** Receipt stops a note being sounded by a note on message.

## 7 Note On

Message Format: 9nH kkH vvH

---

n: MIDI Channel Number  
kk: Key Number  
vv: Velocity

**Send** Sent when the keyboard is played and when play is performed using Auto Accompaniment. The key number changes in accordance with the Octave Shift function.

**Receive** Receipt sounds a note of the corresponding instrument part.

## 8 Polyphonic Key Pressure

Message Format: AnH kkH vvH

---

n: MIDI Channel Number  
kk: Key Number  
vv: Pressure Value



**Send** This message is not sent by this Instrument.

**Receive** This message is not received by this Instrument.

## 9 Control Change

Message Format: BnH cH vvH

---

n: MIDI Channel Number  
cc: Control Number  
vv: Value

**Send** Sent when the Instrument's pedal is operated or when Instrument settings are changed.

**Receive** Receipt changes the pedal and other performance conditions, and Instrument settings.

### 9.1 Bank Select (00H,20H)

Message Format: BnH 00H mmH (MSB)  
BnH 20H 11H (LSB)

---

n: MIDI Channel Number  
mm: MSB Value(Note1)  
11: LSB Value(Send:00H, Receive:Ignored)

Note1 : For details about the relationship between the MSB value and the tone, see the Tone List that comes with the Instrument.

**Send** Sent when a tone is selected.

**Receive** Receipt causes a change in the tone bank number stored in Instrument memory, but the tone is not actually changed until a Program Change message is received. For details, see "11 Program Change".

### 9.2 Modulation (01H)

Message Format: BnH 01H vvH

---

n: MIDI Channel Number  
vv: Value

**Send** This message is not sent by this Instrument.

**Receive** Receipt adds, to the tone being sounded, modulation of a depth specified by the value. In the case of a tone that already has modulation applied, receipt of this message increases the modulation depth. The modulation effect differs according to the tone being used.

### 9.3 Data Entry (06H,26H)

Message Format: BnH 06H mmH (MSB)  
BnH 26H llH (LSB)

---

n: MIDI Channel Number  
mm: MSB Value  
ll: LSB Value

**Send** Sent when there is a change to the parameter assigned to RPN. This Instrument does not have a parameter that corresponds to NRPN.

**Receive** Receipt changes the parameter assigned to RPN. This Instrument does not have a parameter that corresponds to NRPN.

### 9.4 Volume (07H)

Message Format: BnH 07H vvH

---

n: MIDI Channel Number  
vv: Value

**Send** Sent when auto accompaniment is used.

**Receive** Receipt changes the volume of the corresponding part.

### 9.5 Pan (0AH)

Message Format: BnH 0AH vvH

---

n: MIDI Channel Number  
vv: Value(Note1)

Note1 : For information about the relationship between setting values and send/receive values, see “33.3 Pan Setting Value Table” in “VII Setting Values and Send/Receive Values”.

**Send** Sent when auto accompaniment is used.

**Receive** Receipt changes the pan of the corresponding part.

### 9.6 Expression (0BH)

Message Format: BnH 0BH vvH

---

n: MIDI Channel Number  
vv: Value

**Send** Sent when auto accompaniment is used.

**Receive** Receipt changes the Expression value.

## 9.7 Hold1 (40H)

Message Format: BnH 40H vvH

---

n: MIDI Channel Number

vv: Value (Note1)

Note1 : For information about the relationship between setting values and send/receive values, see the “33.1 Off/On Setting Value Table” in “VII Setting Values and Send/Receive Values” of this document.

**Send** Sent when a pedal that has a sustain (damper) function is operated.

**Receive** Receipt performs an operation equivalent to a sustain pedal operation.

**Timbre Type Specific Operation** This operation differs in accordance with the Timbre Type (see “About the Timbre Type” in “11 Program Change”) setting.

- Timbre Type: Melody  
Sustain off/on control is performed in accordance with the value of the received message.
- Timbre Type: Drum  
The received message does not affect sound source operation.

## 9.8 Sostenuto (42H)

Message Format: BnH 42H vvH

---

n: MIDI Channel Number

vv: Value (Note1)

Note1 : For information about the relationship between setting values and send/receive values, see the “33.1 Off/On Setting Value Table” in “VII Setting Values and Send/Receive Values” of this document.

**Send** Sent when a pedal that has a sostenuto function is operated.

**Receive** Receipt performs an operation equivalent to a sostenuto pedal operation.

## 9.9 Soft (43H)

Message Format: BnH 43H vvH

---

n: MIDI Channel Number

vv: Value (Note1)

Note1 : For information about the relationship between setting values and send/receive values, see the “33.1 Off/On Setting Value Table” in “VII Setting Values and Send/Receive Values” of this document.

**Send** Sent when a pedal that has a soft function is operated.

**Receive** Receipt performs an operation equivalent to a soft pedal operation.

## 9.10 Reverb Send (5BH)

Message Format: BnH 5BH vvH

---

n: MIDI Channel Number  
vv: Value

**Send** Sent when auto accompaniment is used.

**Receive** Receipt changes the reverb send of the corresponding part.

## 9.11 Chorus Send (5DH)

Message Format: BnH 5DH vvH

---

n: MIDI Channel Number  
vv: Value

**Send** Sent when auto accompaniment is used and when a Chorus on/off operation is performed.

**Receive** Receipt changes the chorus send of the corresponding part.

## 9.12 NRPN (62H,63H)

Message Format: BnH 62H 11H (LSB)  
BnH 63H mmH (MSB)

---

n: MIDI Channel Number  
11: LSB Value  
mm: MSB Value

**Send** This message is not sent by this Instrument.

**Receive** This Instrument does not have a corresponding NRPN message.

## 9.13 RPN (64H,65H)

Message Format: BnH 64H 11H (LSB)  
BnH 65H mmH (MSB)

---

n: MIDI Channel Number  
11: LSB Value  
mm: MSB Value

### 9.13.1 Pitch Bend Sensitivity

Message Format: BnH 64H 00H  
BnH 65H 00H  
BnH 06H mmH  
BnH 26H 11H

---

n: MIDI Channel Number  
mm: MSB Value(00H - 0CH)  
ll: LSB Value(Send:00H, Receive:Ignored)

**Send** Sent when Bend Range is changed (CTK-5300, WK-330, CDP-220R only).

**Receive** Receipt changes Bend Range.

### 9.13.2 Fine Tune

Message Format: BnH 64H 01H  
BnH 65H 00H  
BnH 06H mmH  
BnH 26H 11H

---

n: MIDI Channel Number  
mm: MSB Value  
ll: LSB Value

**Send** This message is not sent by this Instrument.

**Receive** Receipt changes the fine tune of the corresponding part.

### 9.13.3 Coarse Tune

Message Format: BnH 64H 02H  
BnH 65H 00H  
BnH 06H mmH  
BnH 26H 11H

---

n: MIDI Channel Number  
mm: MSB Value  
ll: LSB Value

**Send** This message is not sent by this Instrument.

**Receive** Receipt changes the coarse tune of the corresponding part. Does not affect sound source operation when the Timbre Type (see "About the Timbre Type" in "11 Program Change") is Drum.

### 9.13.4 Null

Message Format: BnH 64H 7FH  
BnH 65H 7FH

---

n: MIDI Channel Number

**Send** Sent when an RPN message send operation is performed.

**Receive** Receipt de-selects RPN.

## 10 Mode Message

### 10.1 All Sound Off (78H)

Message Format: BnH 78H 00H

---

n: MIDI Channel Number

**Send** Sent when the local control is set to OFF on the Instrument.

**Receive** Receipt stops all voices that are sounding.

### 10.2 Reset All Controllers (79H)

Message Format: BnH 79H 00H

---

n: MIDI Channel Number

**Send** Sent when MIDI send related settings are changed.

**Receive** Receipt initializes each performance controller.

### 10.3 All Notes Off (7BH)

Message Format: BnH 7BH 00H

---

n: MIDI Channel Number

**Send** Sent when MIDI send related settings are changed, or when auto play is stopped, etc.

**Receive** Receipt releases (key release) all voices that are sounding.

### 10.4 Omni Off (7CH)

Message Format: BnH 7CH 00H

---

n: MIDI Channel Number

**Send** This message is not sent by this Instrument.

**Receive** Receipt performs the same operation as when All Notes Off is received.

### 10.5 Omni On (7DH)

Message Format: BnH 7DH 00H

---

n: MIDI Channel Number

**Send** This message is not sent by this Instrument.

**Receive** Receipt performs the same operation as when All Notes Off is received.

## 10.6 Mono (7EH)

Message Format: BnH 7EH 00H

---

n: MIDI Channel Number

**Send** This message is not sent by this Instrument.

**Receive** Receipt performs the same operation as when All Notes Off is received.

## 10.7 Poly (7FH)

Message Format: BnH 7FH 00H

---

n: MIDI Channel Number

**Send** This message is not sent by this Instrument.

**Receive** Receipt performs the same operation as when All Notes Off is received.

# 11 Program Change

Message Format: CnH ppH

---

n: MIDI Channel Number

pp: Program Number (Note1)

Note1 : For details about the relationship between the program number and the tone, see the Tone List that comes with the Instrument.

**Send** Sent when a tone is selected.

**Receive** Receipt changes the tone of the corresponding part. The selected tone is determined by the program value of this message and the Bank Select message value received prior to this message. Also note that receipt of this message also may change the Timbre Type that corresponds to the selected tone. For more information, see “About the Timbre Type” below.

**About the Timbre Type** Tones that are selected by each Instrument part have an attribute that depends on the sound source operation type. This attribute is called the “timbre type,” which is one of the types described below.

- Melody

This timbre type optimizes for normal melody tones.

- Drum

This setting optimizes for drum sounds. The damper pedal does not function. The Hold1, Channel Coarse Tune, and Master Coarse Tune messages are ignored if they are received.

## 12 Channel After Touch

Message Format: DnH vvH

---

n: MIDI Channel Number  
vv: Value

**Send** This message is not sent by this Instrument.

**Receive** Receipt adds, to the tone being sounded, modulation of a depth specified by the value. In the case of a tone that already has modulation applied, receipt of this message increases the modulation depth. The modulation effect differs according to the tone being used.

## 13 Pitch Bend

Message Format: EnH llH mmH

---

n: MIDI Channel Number  
ll: Value LSB  
mm: Value MSB

**Send** Sent when the bender is operated (CTK-5300, WK-330, CDP-220R only).

**Receive** Receipt changes the pitch of the currently sounding note. The range of the pitch change depends on the Bend Range value setting.



## Part III

# System Message

## 14 Timing Clock

Message Format: F8H

**Send** Sent when auto accompaniment is used.

**Receive** This message is not received by this Instrument.

## 15 Start

Message Format: FAH

**Send** Sent when auto accompaniment is used.

**Receive** This message is not received by this Instrument.

## 16 Stop

Message Format: FCH

**Send** Sent when auto accompaniment is used.

**Receive** This message is not received by this Instrument.

## 17 Active Sensing

Message Format: FEH

**Send** This message is not sent by this Instrument.

**Receive** Once this message is received, the Active Sensing mode is entered. If no MIDI message is received for a specified amount of time, voices being sounded by this Instrument's sound source are released, the controller is reset, and the Active Sensing mode is exited.

## 18 System Exclusive Message

Message Format: F0H...F7H

The Instrument sends and receives standard universal system exclusive messages, and system exclusive messages that have Instrument-specific formats.

## 18.1 Universal Real Time System Exclusive Message

Message Format: F0H 7FH...F7H

### 18.1.1 Master Volume

Message Format: F0H 7FH 7FH 04H 01H 11H mmH F7H

---

11: LSB Value(Send:00H, Receive:Ignored)

mm: MSB Value

**Send** This message is not sent by this Instrument.

**Receive** Receipt changes the Master Volume.

### 18.1.2 Master Fine Tuning

Message Format: F0H 7FH 7FH 04H 03H 11H mmH F7H

---

11: LSB Value(Note1)

mm: MSB Value(Note1)

Note1 : For information about the relationship between setting values and send/receive values, see “33.4 Fine Tune Setting Value Table” in “VII Setting Values and Send/Receive Values” of this document.

**Send** This message is sent when the tuning setting is changed.

**Receive** Receipt changes the tuning setting.

### 18.1.3 Master Coarse Tuning

Message Format: F0H 7FH 7FH 04H 04H 11H mmH F7H

---

11: LSB Value(Send:00H, Receive:Ignored)

mm: MSB Value

**Send** Sent when Transpose is changed.

**Receive** Receipt changes the Transpose parameter. Does not affect sound source operation when the Timbre Type (see “About the Timbre Type” in “11 Program Change”) is Drum.

### 18.1.4 Reverb Time

Message Format: F0H 7FH 7FH 04H 05H 01H 01H 01H 01H 01H 01H vvH F7H

---

vv: Value(Note1)

Note1 : For information about the relationship between setting values and send/receive values, see “33.5 Reverb Time Setting Value Table” in “VII Setting Values and Send/Receive Values” of this document.

**Send** Sent when the Reverb duration is changed.

**Receive** Receipt changes the Reverb duration.

### 18.1.5 Chorus Type

Message Format: F0H 7FH 7FH 04H 05H 01H 01H 01H 01H 02H 00H vvH F7H

---

vv: Value(Note1)

Note1 : For information about the relationship between setting values and send/receive values, see “33.6 Chorus Type Setting Value Table” in “VII Setting Values and Send/Receive Values” of this document.

**Send** Sent when the Chorus Type is changed.

**Receive** Receipt changes the Chorus Type.

### 18.1.6 Modulation Rate

Message Format: F0H 7FH 7FH 04H 05H 01H 01H 01H 01H 02H 01H vvH F7H

---

vv: Value

**Send** This message is not sent by this Instrument.

**Receive** Receipt changes the Chorus Rate.

### 18.1.7 GM System On

Message Format: F0H 7EH 7FH 09H 01H F7H

**Send** This message is not sent by this Instrument.

**Receive** Receipt puts the sound source into a GM sound source mode.

### 18.1.8 GM System Off

Message Format: F0H 7EH 7FH 09H 02H F7H

**Send** This message is not sent by this Instrument.

**Receive** Receipt changes the sound source setting to the Instrument presetting.

### 18.1.9 GM2 System On

Message Format: F0H 7EH 7FH 09H 03H F7H

**Send** This message is not sent by this Instrument.

**Receive** Though the Instrument does not support GM2, receipt of the GM2 System On message has the same result as receipt of the GM System On message.

### 18.1.10 GS Message

Message Format: F0H 41H ddH 42H 12H 40H 00H 7FH 00H 41H F7H

---

dd: Device ID(Ignored)

**Send** This message is not sent by this Instrument.

**Receive** Receipt performs the same operation as when the GM System On message is received.

## 18.2 Instrument-Specific System Exclusive Message

Message Format: F0H 44H 16H 02H ... F7H

This message can be used to send the Instrument memory status, for two-way transfer of special operation commands and user data, to perform sound source parameter operations, etc. For more information, see “IV Instrument-Specific System Exclusive Messages”.

## Part IV

# Instrument-Specific System Exclusive Messages

## 19 Format

This section explains the format of the Instrument-specific System Exclusive Messages. See “V Parameter List” and “VI Parameter Set List” for information about how parameter sets actually are transferred.

### 19.1 Message Classifications

Basically, the operation that corresponds to Instrument-specific system exclusive messages is parameter data transfer.

The following operations can be performed from an external device using this parameter transfer message.

- Modification of an individual Instrument parameter
- Batch modification of a particular Instrument parameter set
- Import of an individual Instrument parameter value
- Batch import of a particular Instrument parameter set

In addition to parameters being used as device setting values, some parameters act as commands when received by the Instrument and as device status information when sent from the Instrument.

The following table shows the parameter category for each type of transfer.

Function Section	Parameter Category	Description
System	System	Commands to the Instrument, Instrument stats
	All	All user data
Performance Controller	Song	User Song
	Scale Memory	User Scale Memory
	Sequence	Song Sequencer
	Registration	Registration
	Lesson Rec	Recorder Song(Play-Along)
	Rhythm	User Rhythm
Sound Generator	Patch	Sound source common settings (system effects, master settings, etc.), Instrument part settings (tone selection, mixer channel setting, tuning, etc.)
	Tone	Sampled sound (melody/drum) selected by the user
	Drum	Sampled sound (drum) referenced from “Tone” category parameters
	Instrument	Individual instrument setting referenced from “Drum” category parameters
	Wave Parameter	Sampled sound Wave setting
	Wave Data	Waveform actual data referenced from “Wave Parameter” category parameters

## 19.2 Basic Message Structure

Instrument-specific system exclusive message operation can be broadly divided between two methods: Individual Parameter Transfer (single parameter send/receive) and Bulk Parameter Set Transfer (batch parameter send/receive). Each method includes a number of different messages.

The field in the SysEx message that specifies the message type is the action (*act*) field. The format of the *body* part of the message depends on the *act* value.

The table below shows the body format for each action of Instrument-specific system exclusive messages. An actual message consists of the items indicated by Y, from left to right.

	<- body (Depends on act) ->																
	SX	MAN	MOD	<i>dev</i>	<i>act</i>	<i>cat</i>	<i>mem</i>	<i>pset</i>	<i>blk</i>	<i>pkt</i>	<i>prm</i>	<i>idx</i>	<i>len</i>	<i>data</i>	<i>img</i>	<i>sum</i>	EOX
NOP	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	Y
IPR	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	-	-	-	Y
IPS	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	-	-	Y
OBR	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
OBS	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	-	-	Y	-	Y	Y	Y
HBR	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
HBS	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	-	-	Y	-	Y	Y	Y
ACK	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
BSY	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
RJC	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
EOD	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
EOS	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y
ERR	Y	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	Y

## 19.3 Format of Each Field

### 19.3.1 SX : System Exclusive Message Status

Format: 11110000B (F0H)

This is the System Exclusive Message status byte established by the MIDI standard.

### 19.3.2 MAN : Manufacturer's ID

Format: 01000100B (CASIO = 44H)

Indicates this Instrument's manufacturer ID.

### 19.3.3 MOD : Model ID

Format: MSB 00010110B (16H)

LSB 00000001B (01H)

These two successive bytes (MSB, LSB) indicate the CTK-4200, LK-280, LK-211, WK-220, WK-225, CTK-5300, WK-330, CDP-220R model ID.

### 19.3.4 *dev* : MIDI Device ID 00H - 7FH

Format: 0oddddddB

The contents of this field in a received message are compared with the Model's MIDI Device ID, and receipt of the incoming message is allowed only when the two IDs match. When a message containing 7FH is received, receipt of the message is always allowed, regardless of the Instrument's ID setting. Note, however, that the Instrument does not have a specific Device ID, so use only 7FH for send and receive.

### 19.3.5 *act* : Action

Format: 0aaaaaaaaB

This field indicates the operation of the Instrument-specific System Exclusive Message.

aaaaaaaaB	Action	Function
00H	NOP	No Operation
01H	IPR	Individual Parameter Request
02H	IPS	Individual Parameter Send
03H	OBR	One-way Bulk Parameter Set Request
04H	OBS	One-way Bulk Parameter Set Send
05H	HBR	Handshake Bulk Parameter Set Request
06H	HBS	Handshake Bulk Parameter Set Send
0AH	ACK	Acknowledge
0BH	BSY	Busy
0CH	RJC	Reject
0DH	EOD	End of Data
0EH	EOS	End of Session
0FH	ERR	Error

**NOP:No Operation** No operation is performed when this action is received.

**IPR:Individual Parameter Request** Indicates an individual parameter value send request message. When the Instrument receives this action, it uses an IPS message to return the specified parameter value.

**IPS:Individual Parameter Send** Indicates an individual parameter value send message. When the Instrument receives this action, it rewrites the value specified by the *data* field with the specified parameter value.

**OBR:One-way Bulk Parameter Set Request** Indicates a send request message using parameter set image one-way mode. When the Instrument receives this action, it uses an OBS message to return the specified parameter set.

**OBS:One-way Bulk Parameter Set Bulk Send** Indicates a parameter set image send message using one-way mode. The parameter set to be transferred is divided into multiple packets when it is greater than a prescribed size. At this time, packets are transferred at intervals of the prescribed time (20 msec) or greater.

**HBR:Handshake Bulk Parameter Set Request** Indicates a send request message using the parameter set image handshake mode. When the Instrument receives this action, it uses an HBS message to return the specified parameter set.

**HBS:Handshake Bulk Parameter Set Send** Indicates a parameter set image send message using handshake mode. The parameter set to be transferred is divided into multiple packets when it is greater than a prescribed size. The packets are transferred in accordance with handshake mode.

**ACK:Acknowledge** Indicates a message used by the receiver during parameter set handshake mode transfer to convey to the sender that it is ready for send of the next packet. The *cat*, *mem*, and *pset* fields indicate the value carried by the last received message.

**BSY:Busy** Indicates a message to convey to the requester that that the Instrument cannot send back data after a parameter set one-way mode or handshake mode send request is received. The *cat*, *mem*, and *pset* fields indicate the value carried by the last received message.

**RJC:Reject** Indicates a message to convey to the other side that an ongoing parameter set one-way mode or handshake mode send or receive session was interrupted. The *cat*, *mem*, and *pset* fields indicate the value carried by the last received message.

**EOD:End of Data** Indicates a message to convey to the receiver that a one-way mode or handshake mode serial packet transfer for sending a sub-session (one parameter set) is complete. The *cat*, *mem*, and *pset* fields indicate the value carried by the last received message.

**EOS:End of Session** Indicates a message to convey to the receiver that a one-way mode or handshake mode serial parameter set transfer session send, which was launched by some operation, is complete. The *cat*, *mem*, and *pset* fields indicate the value carried by the last received message.

**ERR:Error** Indicates a message to convey to the sender that checksum or message format of the previous packet was not correct when receiving a parameter set with handshake mode. The *cat*, *mem*, and *pset* fields indicate the value carried by the last received message.

### 19.3.6 *cat* : Category

Format: 0ccccccB

The category indicates the categories of data handled by the System Exclusive Message. The ID number (ID) of the Category is indicated on the left, while the communication operation (Action) is indicated on the right.



Category		Transfer		
ID (c)	Parameter Set	Individual Parameter	One-way Bulk	Handshake Bulk
00H	System	A	-	-
02H	Patch	A	-	-
03H	Tone	A	A	A
06H	Drum	A	A	A
0DH	Instrument	A	A	A
0EH	Wave Parameter	A	A	A
0FH	Wave Data	F	A	A
12H	Scale Memory	A	A	A
1FH	All	F	A	A
20H	Song	F	A	A
21H	Sequence	F	A	A
22H	Registration	F	A	A
23H	Lesson Rec	F	A	A
24H	Rhythm	F	A	A

A ... Available (Also including when only some parameters are available.)

F ... File Information (Not the data itself. Name, size, and other file information only.)

- ... Not Available

### 19.3.7 *mem* : Memory Area ID

Format: 0mmmmmmB

Specifies the memory area that is the object of the parameter transfer. The following are defined for this Instrument.

<i>mem</i>	Data Type	Meaning
0	User area	Read/write enabled
1	Preset area	Read/write disabled

### 19.3.8 *pset* : Parameter Set Number

Format: LSB 0nnnnnnnB  
MSB 0mmmmmmB

This field is a 2-byte (LSB, MSB) value indicating the number of the parameter set (mmmmmmnnnnnnB, Binary) being transferred.

### 19.3.9 *blk* Block Number

The block number is a supplementary number that specifies which block parameter is to be accessed when there are multiple blocks (instrument parts, etc.) that include parameters with the same ID within a single parameter set.

Format: LSB 0iiiiiiiB  
          0jjjjjjjB  
MSB 0kkkkkkkB

**Block Bit Field Division** When the parameter block has a multi-dimensional array structure, bit 21 of the block number is divided into prescribed bit fields based on the rules explained below.

- Case 1

When an array has three or fewer nesting levels and the number of arrays in each dimension is 128 or less, they are assigned below the three 7-bit fields. Unused regions are filled will zeros.

– Example:

parameter [A][B][C]

With a 3-dimensional array parameter that consists of  $A = 8$  (3 bits),  $B = 5$  (3 bits) and  $C = 10$  (4 bits), the block bit fields are allocated as:

Block = 0000aaa 0000bbb 000cccc (Binary).

- Case 2

When Case 1 conditions are not satisfied, the minimal number of fields required for each number of arrays is reserved from the lower bit of the block. Unused regions are filled will zeros. (This case does not apply with this Instrument.)

– Example 1:

parameter [A][B][C][D]

With a 4-dimensional array parameter that consists of  $A = 3$  (2 bits),  $B = 4$  (3 bits),  $C = 3$  (2 bits), and  $D = 4$  (3 bits) like the one shown above, the block bit fields are allocated as:

Block = 0000000 0000aab bbccddd (Binary).

– Example 2:

parameter [A][B]

With a 2-dimensional array parameter that consists of  $A = 3$  (2 bits) and  $B = 200$  (8 bits) like the one shown above, the block bit fields are allocated as:

Block = 0000000 0000aab bbbbbbb (Binary).

### 19.3.10 *pkt* : Packet Number

Format:    LSB    0iiiiiiiB  
                  0jjjjjjjB  
                  MSB    0kkkkkkkB

This is the divided packet number `kkkkkkkjjjjjjiiiiiiB` (Binary) for transferring a single parameter set.

### 19.3.11 *prm* : Parameter ID

Format:    LSB    0pppppppB  
                  MSB    0qqqqqqqB

The Parameter ID indicates the parameter type. When transferring parameters (see “V Parameter List” below) individually (as opposed to bulk transfer), this field is used to identify the parameter being transferred by its parameter ID.

### 19.3.12 *idx* : Data Index Number

Format:    LSB    0iiiiiiiB  
                  MSB    0jjjjjjjB

The data index number indicates the first array number of the array from which transfer starts.

### 19.3.13 *len* : Data Length

Format:    LSB    01111111B  
               MSB    0mmmmmmmmB

As shown below, the meaning of this field differs depending on whether an individual transfer or a bulk parameter set transfer is being performed.

**Individual Parameter Transfer** Data length indicates the length of the array being transferred minus 1 when the parameter contains a character string or other similar array structure.

**Bulk Parameter Set Transfer** Data length indicates the number of bytes of data included within a packet. When this value is zero, it means the data itself does not exist.

### 19.3.14 *data* : Parameter Data

#### Individual Parameter Transfer

Format:    index0    0dddddddB (0eeeeeeeB) (0fffffffB) (0gggggggB) (0hhhhhhhB)  
               index1    0dddddddB (0eeeeeeeB) (0fffffffB) (0gggggggB) (0hhhhhhhB)  
               index2    0dddddddB (0eeeeeeeB) (0fffffffB) (0gggggggB) (0hhhhhhhB)  
               :           :  
               indexN    0dddddddB (0eeeeeeeB) (0fffffffB) (0gggggggB) (0hhhhhhhB)

Parameter data indicates the parameter value. Data is repeatedly placed in an array of the size equivalent to len+1. For the structure of one data item, the length depends on the data bit width(Parameter List Size), as shown below.

Size	Number of Data
1 - 7	1
8 - 14	2
15 - 21	3
22 - 28	4
29 - 32	5

Each block of data is packed from the lowest order byte first. In the case of multiple-byte data, the lowest weighted bit is the least significant digit of the first data byte, and the highest weighted bit is the most significant digit of the final data byte. The following shows an example of how data would be divided for transfer in the case of 32-bit data.

	7	6	5	4	3	2	1	0
data0:	0	[bit06]	[bit05]	[bit04]	[bit03]	[bit02]	[bit01]	[bit00]
data1:	0	[bit13]	[bit12]	[bit11]	[bit10]	[bit09]	[bit08]	[bit07]
data2:	0	[bit20]	[bit19]	[bit18]	[bit17]	[bit16]	[bit15]	[bit14]
data3:	0	[bit27]	[bit26]	[bit25]	[bit24]	[bit23]	[bit22]	[bit21]
data4:	0	0	0	0	[bit31]	[bit30]	[bit29]	[bit28]

### 19.3.15 Single Parameter Data Size Limit

With the Instrument's system exclusive message format, the size of a single message cannot exceed 256 bytes. The data size and the array size, however, can cause a packet to exceed 256 bytes when transferring a single parameter array. In this case, the IPS and IPR message data length and data index number values can be modified to enable division of a single parameter value into multiple messages so it can be sent that way.

### 19.3.16 *img* : Parameter Set Image

Format:   LSB   0dddddddB  
              0cccccccB  
          MSB   000000abB

For a bulk data transfer operation, the parameter set data to be transferred is read sequentially in 16-bit units starting from the top address. Read values are divided into 3-byte segments as shown below, and then sent in sequence.

The following is the conversion format, which is the same as the individual parameter 16-bit transfer detailed above.

#### 16-bit Memory Image

MSB: abccccccB  
LSB: cdddddddB

data0: 0dddddddB  
data1: 0cccccccB  
data2: 000000abB

If the length of the data begin transferred is an odd number of bytes, the final bit field is filled in with zero.

**Parameter Set Packet Division** The maximum image size of the parameter set that can be sent with one packet is 128 bytes. When data that exceeds 128 bytes is transferred, data is always divided into 128-byte units, except for the final packet. Because of this, the data in received packets can be stored in an areas the size of  $[ParameterSetStartAddress] + [PacketNumber] \times 128$ .

For bulk parameter set transfer, the 16-bit image is divided in 3-byte groups, so the *img* field length is the value produced by multiplying the *len* value ( $len + 1$  when *len* is an odd number) by  $3/2$ . Note, however, that a parameter set of 128 bytes or less can be sent using a single packet, and anything greater than 128 bytes is divided among multiple packets. This means that the maximum length "data" field is  $128 \times 3/2 = 192$  bytes.

Only one parameter set can be transferred per session, and data from different parameter sets cannot be mixed within a single packet, even when sending multiple parameter sets. Different parameter sets are always divided into separate packets.

### 19.3.17 *sum* : Check Sum

Format:   0sssssssB

In this case, this "sum" field contains a value, which, when added to the total value of the "img" field, makes the lower seven bits 0. The receiving side checks if this is true, and performs error handling (re-request, etc.) if it is not.

### 19.3.18 EOX : End of System Exclusive Message

Format: 11110111B

This is the End of System Exclusive Message status byte established by the MIDI standard.

## 20 Individual Parameter Operations

There are two parameter unit operations: Individual Parameter Transfer and Individual Parameter Request. For one session, in response to an IPR (Individual Parameter Request) from an external device, this Instrument returns an IPS (Individual Parameter Send) or the session is concluded when the external device or this Instrument spontaneously sends an IPS. If this Instrument received an IPS, the value of the applicable parameter is changed. Depending on the function of a parameter, Individual Parameter Send may be used to issue a command to the Instrument and Individual Parameter Request may be used to check Instrument status information.

Data Receiver	Data Sender	Operation
IPR		Send Request(Optional)
	IPS	Data Transfer

## 21 Parameter Set Transfer

### 21.1 Communication Modes

#### 21.1.1 One-way and Handshake

In order to ensure maximum speed for bulk dumping of Parameter Sets, the data format is different from the data format used for Individual Parameter Send. Data is transferred as-is, using the Model's memory image. Parameter Sets can be transferred by bulk dump using the message exchange types described below.

- One-way mode Parameter Set send/receive
- One-way mode Parameter Set send request send/receive
- Handshake mode Parameter Set send/receive
- Handshake mode Parameter Set send request, receive rejected, error notification send/receive

With the one-way mode, the sending device sends data and ends the session without regard to the response of the receiving device. This mode is best for one-way transfers from a sequencer or similar device.

With the handshake mode, the sending device sends the data and then waits for a response from the receiving device before advancing to the next session. This is a high-speed mode in which there is no time wasted waiting.

See "VI Parameter Set List" for details about Instrument parameter sets.

#### 21.1.2 Session and Subsession

**Subsession** "One subsession" refers to transfer of one parameter set. A subsession transfers one parameter set or a parameter set that has been divided into multiple packets for transfer, with EOD (End of Data) at the end to terminate the send. Division of a parameter set into multiple packets is used when the size of the parameter set is greater than a prescribed size. The packet number in the packet index field indicates the sequential position of a packet relative to the other packets. A single packet cannot be used to transfer multiple small parameter sets. A parameter set delimiter always must be transferred as a packet delimiter.

**Session** “One session” refers to a series of processes that occur for one user operation. One subsession or multiple subsessions make up a session. The sender sends EOS (End of Session) to end a session. Regardless of whether there is a single parameter set or multiple parameter sets being transferred, a bulk dump always takes the form of a session, never a subsession only.

## 21.2 One-way Mode Communication Flow

A session starts with the receiving device sending a request using an OBR, or with the sending device sending OBS data. The session ends after transfer of all the data in the parameter set being transferred by the sending device is complete.

The transferred messages of a single parameter set cannot exceed 256 bytes. When 256 bytes are exceeded, data is divided into multiple packets of 256 bytes or less each and transferred at fixed intervals (20 msec).

A final EOD informs the receiving device when the session is ended.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
OBR	----->		Send Request (Optional)
	<-----	OBS (20 msec or greater interval)	Data Transfer
	<-----	OBS (20 msec or greater interval)	Data Transfer
	<-----	OBS (20 msec or greater interval)	Data Transfer
	:		
	:		
	<-----	EOD	End of Data
	:		
	Other Sub Session		
	:		
	<-----	EOS	End of Session

## 21.3 Handshake Mode Communication Flow

A session starts with the receiving device sending a request using a HBR, or with the sending device sending HBS data. The sending device does not send the next packet until it receives an ACK from the receiving device. The maximum wait time of at least 2000 msec is reserved. Failure of a response to arrive within the wait time is treated as a timeout error, and data communication is terminated. The receiving device returns ERR (error) when there is mismatch in the received data checksum or when an incompatible data format is discovered. When the sending device receives ERR (error) from the receiving device, it resends the last data sent. If an error repeats a number of times (three times or more for this Instrument), either the sending device or the receiving device sends an RJC to terminate the session. A session ends after the sending device transfers the volume of data determined by the number of parameter sets being transferred, and sends EOD and EOS in response to an ACK from the sending device.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request (Optional)
	<-----	HBS	Data Send
ACK	----->		Acknowledge
	<-----	HBS	Data Send
ACK	----->		Acknowledge
	:		
	:		
ACK	----->		Acknowledge
	<-----	EOD	End of Data
	:		
	Other Sub Session		
	:		
	<-----	EOS	End of Session

The packet with the same packet number is resent when a checksum mismatch or incompatible data format error is detected.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request (Optional)
	<-----	HBS	Data Send
ACK	----->		Acknowledge
	<-----	HBS	Data Send 1
ERR	----->		Error
	<-----	HBS	Data Send 2 (Retry 1)
ERR	----->		Error
	<-----	HBS	Data Send 3 (Retry 2)
ACK	----->		Acknowledge
	:		
	:		
ACK	----->		Acknowledge
	<-----	EOD	End of Data
	:		
	Other Sub Session		
	:		
	<-----	EOS	End of Session

Session terminates if ERR is detected a number of times.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request (Optional)
	<-----	HBS	Data Send
ACK	----->		Acknowledge
	<-----	HBS	Data Send
	(Error generated)		
ERR	----->		Error
	<-----	HBS	Data Send (Retry 1)
	(Error generated)		
ERR	----->		Error
	<-----	HBS	Data Send (Retry 2)
	(Error generated)		
ERR	----->		Error
	<-----	HBS	Data Send (Retry 3)
	(Error generated)		
RJC	----->		Session stopped

RJC is sent to terminate the session in case ACK cannot be recognized.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request (Optional)
	<-----	HBS	Data Send
	:		
	(Fixed amount of time elapses)		
RJC	<-----		Timeout error

The session can be canceled for any reason by sending an RJC. RJC can be sent by the sending device or the receiving device. The bulk dump session is terminated immediately upon receipt of an RJC.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request (Optional)
	<-----	HBS	Data Send
ACK	----->		Acknowledge
	<-----	HBS	Data Send
	:		
	:		
	(Terminate operation)		
RJC	----->		Data Receive Canceled



Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request (Optional)
ACK	----->		Acknowledge
	<-----	HBS	Data Send
ACK	----->		Acknowledge
	<-----	HBS	Data Send
	:		
	:		
	(Terminate operation)		
	<-----	RJC	Data Send Canceled

BSY is returned to the external device when OBS, OBR, HBS, or HBR is sent while the Instrument mode is not suitable to perform bulk dump, etc. After BSY is received, the external device should wait until the instrument enters a mode in which the session is enabled.

Data Receiver (External Device)		Data Sender (This Instrument)	Operation
OBR	----->		Send Request
	<-----	BSY	Send Rejected
Data Receiver (External Device)		Data Sender (This Instrument)	Operation
	<-----	OBS	Data Send
BSY	----->		Receive Rejected
Data Receiver (External Device)		Data Sender (This Instrument)	Operation
HBR	----->		Send Request
	<-----	BSY	Send Rejected
Data Receiver (External Device)		Data Sender (This Instrument)	Operation
	<-----	HBS	Data Send
BSY	----->		Receive Rejected

## Part V

# Parameter List

This section explains the parameters that actually can be transferred by the Instrument.

## 22 Using the Parameter List

- Parameter field  
Shows the parameter name.
- ID field  
Shows the parameter ID as a hexadecimal number.
- R/W field  
Shows “R” to indicate that an IPR (Individual Parameter Request) read operation (Read) is possible or “W” to indicate that an IPS (Individual Parameter Send) write operation is possible.
- Block field  
Shows the bit field allocation of the block number. The bit field position is shown as a decimal format number.
- Size field  
Shows the parameter bit width as a decimal format value.
- Array field  
Shows the parameter array size as a hexadecimal value.
- Min-Def-Max field  
Shows the minimum value, default value, and maximum value for parameter acquisition as a hexadecimal value.
- Description field  
Explains the meaning of parameter values. Unless otherwise specified, setting values are all indicated in decimal format.

## 23 System Parameters

These parameters make it possible for an external device to check the status of the Instrument and for an external device to command some operation of the Instrument.

### 23.1 System Information Parameter

This parameter is a container for system information.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Model Name	0000	R	000000	7	08	00-20-7F	Ascii Character CTK-4200....."CTK-4200" CTK-5300....."CTK-5300" LK-211....."LK-211 " LK-280....."LK-280 " WK-220/WK-225..."WK-220 " WK-330....."WK-330 " CDP-220R....."CDP-220R"
Program Revision	0001	R		7	08	00-20-7F	Ascii Character Ex. " 12345"
General Register	0002	R/W		8	01	00-00-FF	General-purpose register for communication test

## 23.2 Data Management Parameter

These are information acquisition and operation command parameters for this Instrument's Data Manager PC application.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Ps Category	0005	W	00000000	7	01	00-00-7F	Specifies the category ID of the parameter set that corresponds to an operation.
Ps Memory	0006	W		7	01	00-00-7F	Specifies the memory ID of the parameter set that corresponds to an operation.
Ps Number	0007	W		14	01	0000-0001-3FFF	Specifies the number of the parameter set that corresponds to an operation.
Ps Data Type	0008	R		8	01	00-00-FF	This the data type of the parameter set in the specified category. Compatibility is provided between models with the same type.
Current Ps Existence	0009	R		1	01	00-00-01	Whether a parameter set exists in the specified category. 0...No 1...Yes
Current Ps Protect	000A	R		1	01	00-00-01	Protect status of a parameter set data in the specified category. 0...Off 1...On
Current Ps Size	000B	R		32	01	00000000-00000000-FFFFFFFF	Size of a parameter set in the specified category/number (bytes).
Current Sub Ps Size	000C	R		32	01	00000000-00000000-FFFFFFFF	Total size of a parameter sets under the specified category (bytes).
Current Ps Name	000D	R		8	08	00-20-7F	Name of a parameter set in the specified category (ASCII characters).
Max Ps Size	000E	R		32	01	00000000-00000000-FFFFFFFF	Maximum size of parameter sets in the specified category (bytes).
Max Ps Number	000F	R		14	01	0000-0000-FFFF	Maximum number parameter sets in the specified category (bytes).
Current Free Size	0010	R		32	01	00000000-00000000-FFFFFFFF	Size of current parameter set free space in the specified category (byte).
Max Free Size	0011	R		32	01	00000000-00000000-FFFFFFFF	Maximum size of parameter free space in the specified category (bytes).
Delete Ps	0012	W		1	01	00-00-01	Deletes parameter sets in the specified category. Values are ignored.

## 24 Patch Parameter

The main function of patch parameters is to configure the settings of the sound source of a device.

### 24.1 Master Tune Parameter

These parameters configure the Master Tuning settings.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Master Fine Tune	0000	R/W	000000	10	01	0000-0200-03FF	-512*100/512 ~ 0*100/512 ~ +511*100/512(cent)
Master Coarse Tune	0001	R/W		7	01	00-40-7F	-24 ~ 0 ~ +24(semitone)

### 24.2 Master Mixer Parameter

These parameters configure the Master settings of the mixer.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Master Volume	0002	R/W	000000	7	01	00-7F-7F	0 ~ 127

### 24.3 Part Parameter

Part parameters configure the settings of each musical instrument part.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Part Enable	0068	R/W	4-0: Part #	1	01	00-01-01	0...Off 1...On
Scaletulan Enable	0069	R/W		1	01	00-01-01	0...Disable 1...Enable
Tone Num	006A	R/W		14	01	0000-0000-3FFF	0 ~ 16383
Fine Tune	006B	R/W		7	01	00-40-7F	-100 ~ 0 ~ +99(cent)
Coarse Tune	006C	R/W		7	01	28-40-58	-24 ~ 0 ~ +24(semitone)
Volume	006D	R/W		7	01	00-64-7F	0 ~ 127
Acmp Volume	006E	R/W		7	01	00-7F-7F	0 ~ 127
Pan	006F	R/W		7	01	00-40-7F	-64 ~ 00 ~ +63
Cho Send	0070	R/W		7	01	00-00-7F	0 ~ 127
Rev Send	0071	R/W		7	01	00-28-7F	0 ~ 127
Bend Range	0072	R/W		7	01	00-02-18	0 ~ 24

## 25 Scale Memory Parameter

### 25.1 Scale Memory Octave

Scale memory parameters store scale memory data.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Note	0000	R/W	000000	8	0C	00-80-FF	-128 ~ 0 ~ 127 Array : Note 0...C 1...C# 2...D 3...D# 4...E 5...F 6...F# 7...G 8...G# 9...A 10...A# 11...B
A Key Mode	0001	R/W		1	01	00-00-01	0...Relative 1...Absolute

## 26 All Data Parameter

There is no parameter for storing all data directory information.

## 27 Sequence Parameter

### 27.1 Sequence Directory Info

The sequence parameter stores recorder song directory information.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Size	0001	R	000000	32	01	00000000-00000000-00FFFFFFF	0 ~ 0xFFFFFFFF

## 28 Registration Parameter

### 28.1 Registration Directory Info

The registration parameter stores registration data directory information.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Size	0001	R	000000	32	01	00000000-00000000-00FFFFFFF	0 ~ 0xFFFFFFFF

## 29 Rhythm Parameter

### 29.1 Rhythm Directory Info

The rhythm parameter stores rhythm directory information.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Name	0000	R	000000	7	08	20-20-7F	Ascii Character
Size	0002	R		32	01	00000000-00000000-00FFFFFFF	0 ~ 0xFFFFFFFF

## 30 Song Parameter

### 30.1 Song Directory Info

The song parameter stores song data directory information.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Name	0000	R	000000	7	08	20-20-7F	Ascii Character
Size	0002	R		32	01	00000000-00000000-00FFFFFFF	0 ~ 0xFFFFFFFF

## 31 Lesson Rec Parameter

### 31.1 Lesson Rec Directory Info

The lesson rec parameter stores recorder song (play-along) directory information.

Parameter	ID	R/W	Block	Size	Array	Min-Def-Max	Description
Size	0001	R	000000	32	01	00000000-00000000-00FFFFFFF	0 ~ 0xFFFFFFFF

## Part VI

# Parameter Set List

This section explains actually how parameter sets can be transferred by the Instrument with bulk dump.

## 32 Parameter Set Table

Parameter Set Category	<i>cat</i>	<i>mem</i>	<i>pset</i>	description
Tone	03H	00H	0000H - 0004H 0005H - 0007H	(Melody Sampling Tone 0 - 4) (Drum Sampling Tone 0 - 2)
Drum	06H	00H	0000H - 0002H	(Drum Sampling Tone 0 - 2)
Inst	0DH	00H	0000H - 007FH 0080H - 00FFH 0100H - 017FH	(Drum Sampling Tone 0 / Inst 0 - 127) (Drum Sampling Tone 1 / Inst 0 - 127) (Drum Sampling Tone 2 / Inst 0 - 127)
Wave Param	0EH	00H	0000H - 0004H 0005H - 000CH 000DH - 0014H 0015H - 001CH	(Melody Sampling Tone 0 - 4) (Drum Sampling Tone 0 / Wave Param 0 - 7) (Drum Sampling Tone 1 / Wave Param 0 - 7) (Drum Sampling Tone 2 / Wave Param 0 - 7)
Wave Data	0FH	00H	0000H - 0004H 0005H - 000CH 000DH - 0014H 0015H - 001CH	(Melody Sampling Tone 0 - 4) (Drum Sampling Tone 0 / Wave Param 0 - 7) (Drum Sampling Tone 1 / Wave Param 0 - 7) (Drum Sampling Tone 2 / Wave Param 0 - 7)
Scale Memory	12H	00H	0003H - 0006H	(User 1 - 4)
All	1FH	00H	0000H - 003CH	
Song	20H	00H	0000H - 0009H	(Song Bank Song 0 - 9)
Sequence	21H	00H	0000H - 0004H	(Recorder Song 0 - 4)
Registration	22H	00H	0000H - 0007H	(Bank 0 - 8)
Lesson Rec	23H	00H	0000H	
Rhythm	24H	00H	0000H - 0009H	(User Rhythm 0 - 9)

### Field Contents

- *cat* field  
Shows the category value.(Note1)
- *mem* field  
Shows the memory area ID value.(Note1)

- *pset* field

Shows the parameter set number value. Applicable parameter set numbers are those in the user area where the top number is zero, and are not the same numbers as those displayed by the Instrument.  
(Note1)

Note1 : Operation is not guaranteed for values other than those noted here.

## Part VII

# Setting Values and Send/ Receive Values

## 33 Setting Value Tables

### 33.1 Off/On Setting Value Table

Transmit Value	Receive Value	Parameter
00H	00H - 3FH	Off
7FH	40H - 7FH	On

### 33.2 -64 - 0 - +63 Setting Value Table

Transmit Value	Receive Value	Parameter
00H	00H	-64
:	:	:
40H	40H	0
:	:	:
7FH	7FH	+63

### 33.3 Pan Setting Value Table

Transmit Value	Receive Value	Parameter
00H	00H	Left
:	:	:
40H	40H	Center
:	:	:
7FH	7FH	Right

### 33.4 Fine Tune Setting Value Table

Transmit Value	Receive Value	Parameter
(LSB, MSB)	(LSB, MSB) - (LSB, MSB)	
(43H, 00H)	(00H, 00H) - (5FH, 00H)	415.5 Hz
(65H, 00H)	(60H, 00H) - (7FH, 00H)	415.6 Hz
(07H, 01H)	(00H, 01H) - (1FH, 01H)	415.7 Hz
(29H, 01H)	(20H, 01H) - (3FH, 01H)	415.8 Hz
:	:	:
(40H, 3FH)	(30H, 3FH) - (4FH, 3FH)	439.8 Hz
(60H, 3FH)	(50H, 3FH) - (6FH, 3FH)	439.9 Hz
(00H, 40H)	(70H, 3FH) - (1FH, 40H)	440.0 Hz
(20H, 40H)	(20H, 40H) - (3FH, 40H)	440.1 Hz
(40H, 40H)	(40H, 40H) - (5FH, 40H)	440.2 Hz
:	:	:
(54H, 7EH)	(50H, 7EH) - (6FH, 7EH)	465.6 Hz
(73H, 7EH)	(70H, 7EH) - (0FH, 7FH)	465.7 Hz
(11H, 7FH)	(10H, 7FH) - (2FH, 7FH)	465.8 Hz
(30H, 7FH)	(30H, 7FH) - (7FH, 7FH)	465.9 Hz



### 33.5 Reverb Time Setting Value Table

Transmit Value	Receive Value	Parameter
00H	00H - 0BH	Off
0CH	0CH - 17H	1
18H	18H - 23H	2
24H	24H - 2FH	3
30H	30H - 3BH	4
3CH	3CH - 47H	5
48H	48H - 53H	6
54H	54H - 5FH	7
60H	60H - 6BH	8
6CH	6CH - 77H	9
78H	78H - 7FH	10

### 33.6 Chorus Type Setting Value Table

Transmit Value	Receive Value	Parameter
00H	00H	Chorus 1
01H	01H	Chorus 2
02H	02H	Chorus 3
03H	03H	Chorus 4
04H	04H	Chorus 5

## Part VIII

# MIDI Implementation Notation

## 34 Value Notation

### 34.1 Hexadecimal Notation

MIDI implementation sometimes requires that data be expressed in hexadecimal format. Hexadecimal values are indicated by the letter “H” after the value. The hexadecimal equivalents of decimal values 10 through 15 are expressed as the letters A through F.

The table below shows the hexadecimal equivalents for decimal values 0 through 127, which are often used in MIDI messages.

Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal
0	00H	32	20H	64	40H	96	60H
1	01H	33	21H	65	41H	97	61H
2	02H	34	22H	66	42H	98	62H
3	03H	35	23H	67	43H	99	63H
4	04H	36	24H	68	44H	100	64H
5	05H	37	25H	69	45H	101	65H
6	06H	38	26H	70	46H	102	66H
7	07H	39	27H	71	47H	103	67H
8	08H	40	28H	72	48H	104	68H
9	09H	41	29H	73	49H	105	69H
10	0AH	42	2AH	74	4AH	106	6AH
11	0BH	43	2BH	75	4BH	107	6BH
12	0CH	44	2CH	76	4CH	108	6CH
13	0DH	45	2DH	77	4DH	109	6DH
14	0EH	46	2EH	78	4EH	110	6EH
15	0FH	47	2FH	79	4FH	111	6FH
16	10H	48	30H	80	50H	112	70H
17	11H	49	31H	81	51H	113	71H
18	12H	50	32H	82	52H	114	72H
19	13H	51	33H	83	53H	115	73H
20	14H	52	34H	84	54H	116	74H
21	15H	53	35H	85	55H	117	75H
22	16H	54	36H	86	56H	118	76H
23	17H	55	37H	87	57H	119	77H
24	18H	56	38H	88	58H	120	78H
25	19H	57	39H	89	59H	121	79H
26	1AH	58	3AH	90	5AH	122	7AH
27	1BH	59	3BH	91	5BH	123	7BH
28	1CH	60	3CH	92	5CH	124	7CH
29	1DH	61	3DH	93	5DH	125	7DH
30	1EH	62	3EH	94	5EH	126	7EH
31	1FH	63	3FH	95	5FH	127	7FH

### 34.2 Binary Notation

When a MIDI implementation data value is expressed in binary, the letter “B” (for “binary”) is affixed at the end of the value. The table below shows the binary equivalents for the decimal values 0 through 127, which are often used for settings.

Decimal	Hexadecimal	Binary
0	00H	00000000B
1	01H	00000001B
2	02H	00000010B
3	03H	00000011B
4	04H	00000100B
5	05H	00000101B
6	06H	00000110B
7	07H	00000111B
8	08H	00001000B
9	09H	00001001B
10	0AH	00001010B
11	0BH	00001011B
12	0CH	00001100B
13	0DH	00001101B
14	0EH	00001110B
15	0FH	00001111B
16	10H	00010000B
:	:	
125	7DH	01111101B
126	7EH	01111110B
127	7FH	01111111B

## Free Manuals Download Website

<http://myh66.com>

<http://usermanuals.us>

<http://www.somanuals.com>

<http://www.4manuals.cc>

<http://www.manual-lib.com>

<http://www.404manual.com>

<http://www.luxmanual.com>

<http://aubethermostatmanual.com>

Golf course search by state

<http://golfingnear.com>

Email search by domain

<http://emailbydomain.com>

Auto manuals search

<http://auto.somanuals.com>

TV manuals search

<http://tv.somanuals.com>