OHM64

User Guide

Table Of Contents

Introduction

Thank you for purchasing the Ohm64 control surface from Livid Instruments. This instrument is built with all the familiar controls of a standard MIDI controller, instrument, or mixer. The Ohm64 features assignable LED backlighting, custom designed rubber buttons, and high quality sliders and knobs. The control layout was inspired by some of the most innovative and popular audio software programs that have revolutionized the way music is created and performed. Ohm64 users will find it to be highly intuitive and adaptable to a variety of music and video software and makes an ideal companion for hardware such as analog synthesizers, video mixers, DJ software, lighting equipment, and other musical instruments.

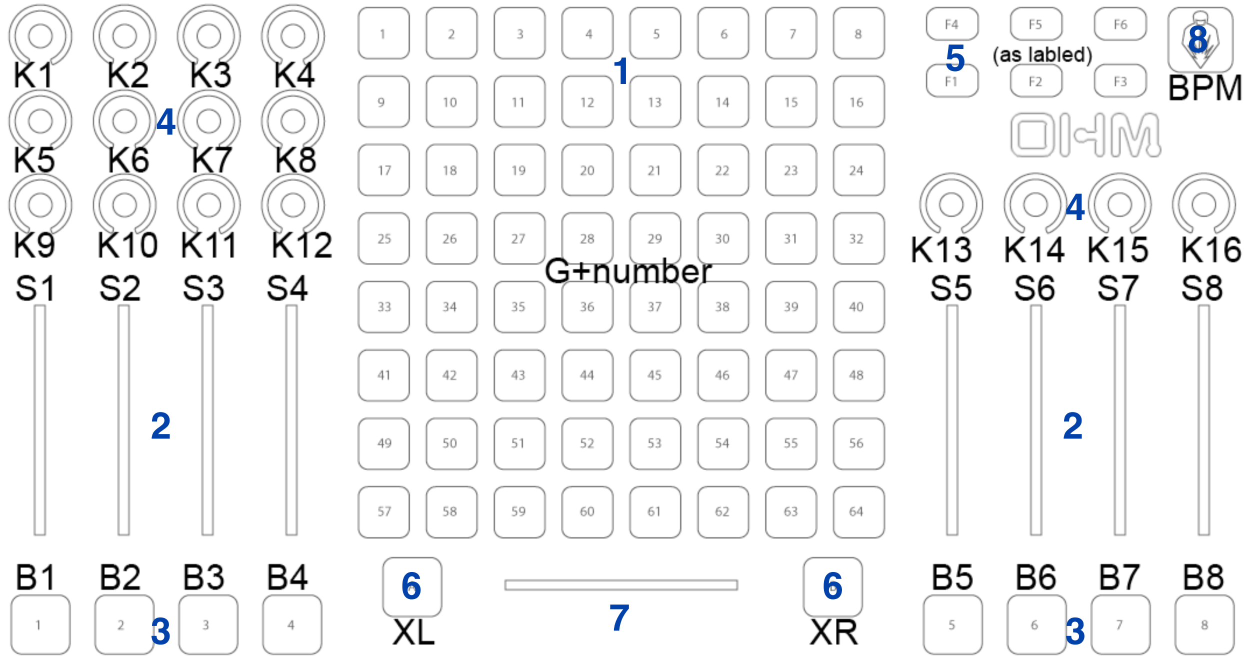
What's in the box

Livid Ohm64 Controller
USB Cable
Livid installer CD with drivers, software, and a user manual

Features

64 button clip bank
8 Faders
1 Crossfader
16 Rotary knobs
6 Function knobs
1 BPM/Tap button
8 Solo / Mute buttons
2 Transform buttons

Controls and Connections



While any of the controls on the Ohm64 can be used for a wide range of functionality, here are some basic terms and uses for the controls.

1. Programmable Clip Bank – The Clip Bank sends out data for activating files, triggering loops, sequences, or files. The programmable buttons on the Clip Bank can be assigned to MIDI notes or Continuous Controllers (CCs).

2. Programmable Sliders – These programmable vertical sliders can be used to adjust effects, mapped to pitch and volume, or used to manipulate data. The sliders can be programmed to send CC or Pitch Bend messages.

3. Programmable Triggers – These programmable buttons can be used to activate effects or mute channels. The buttons can be assigned to MIDI notes or CCs.

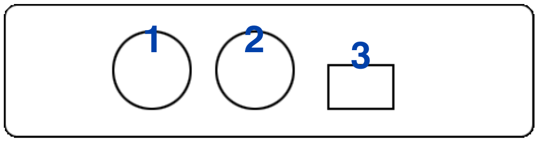
4. Programmable Knobs – These programmable knobs can be used to adjust effects, EQ settings, and control playback. The knobs can be programmed to send CC or Pitch Bend messages.

5. Programmable Function Buttons – These programmable buttons can be used to activate various functions within the software such as: switching between channels or banks, selecting tracks, or used to cue, play, pause, and rewind your media. The buttons can be assigned to MIDI notes, program changes, or CCs.

6. Programmable Transform Buttons – These programmable buttons can be used to transform, mix, switch, or fade between channels. The buttons can be assigned to MIDI notes, program changes, or CCs.

7. Programmable Crossfader – Use the crossfader to control the crossfade, mix, or transition between channels. The crossfader can be programmed to send increasing values to the left or to the right.

8. **Programmable Tap Button –** This programmable button is designed for tapping the BPM. This button can be assigned to MIDI notes, program changes, or CC's.



1. **MIDI IN –** The MIDI Input port provides a standard 5 PIN MIDI jack for connecting other MIDI devices to the Ohm64. MIDI input data is then routed over USB to the host computer, and is available to your software on the second port named "Ohm64 External" (on Windows XP, "USB Audio Device[2]")

2. MIDI Out - The MIDI Output port provides a standard 5 PIN MIDI jack for outputting MIDI from your computer, over the USB cable, and to the MIDI jack. The Ohm64 will also send its data out the MIDI jack. This MIDI Output can be accessed on your computer via the port named "Ohm64 External" (on Windows XP, "USB Audio Device[2]")

3. USB – USB Port for powering the Ohm64 and sending MIDI message to and from the instrument.

Connecting

To make the Ohm64 controller compatible with industry standard sound and video equipment, the controller comes equipped with both a USB and MIDI out connection. The USB is designed to connect directly to a computer and the MIDI out can be used to connect to a MIDI interface, PC soundcard, external soundcard, or any of the hundreds of MIDI processors and synthesizers using a standard MIDI cable. The Ohm64 is completely USB powered and does not require an external power supply.

For Direct USB Connectivity

The Ohm64 is class-complaint and does not require drivers when using the USB connectivity.

1. Connect the USB cable from the computer to the USB port on the Ohm64.

2. When the logo on the Ohm64 is illuminated, the device is powered.

For MIDI Connectivity

Using a standard MIDI cable, connect the Ohm64 MIDI in and out to the MIDI in and out of your soundcard or other MIDI interface. The Ohm64 must be plugged into a USB host and have MIDI software running (even if it's something basic like MIDI Monitor (http://www.snoize.com) or MIDI OX (http://www.midiox.com) running. This is a by-product of USB-powered class-compliant devices with no external power source. However, once you have the Ohm64 connected to a computer and some simple MIDI application running, you can send the Ohm's MIDI out the MIDI output jack.

Troubleshooting

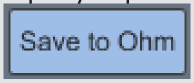
- Make sure your USB cable is properly connected. If you are using a USB hub, try connecting the Ohm64 USB cable directly to your computer. The lights should come on once the Ohm64 is connected.
- If you are using a Macintosh Computer running OSX, make sure you have Ohm64 selected as your device in the Audio MIDI setup. To access Audio MIDI go to Applications>Utilities>Audio MIDI Setup.
- Check to make sure your software is configured to use the Ohm64. Open your application's preferences to verify that the software is configured to receive the MIDI messages from the Ohm64 controller.
- Windows XP will show the port names as "USB Audio Device" and "USB Audio Device[2]", though the "New Hardware Found" popup will show the device as "Ohm64".
- If you are still having problems, confirm that you are receiving MIDI data using MidiMonitor (for OS X) at http://www.snoize.com/MIDIMonitor/ or MIDIOX (for Windows) http://www.midiox.com.

Ohm64 Editor

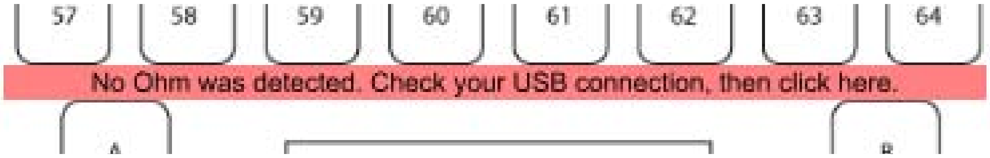
The Ohm64 Editor provides an interface for changing the MIDI assignments of the Ohm64 buttons, knobs, and sliders. Sliders and knobs can send out a variety of combinations of control and pitchbend data

Buttons send out note or controller messages. The lights behind all buttons except the F1-6 buttons respond to note messages to turn them on and off.

1. Make sure computer is connected to Ohm64 with USB cable.

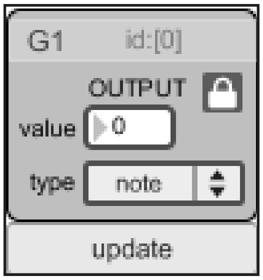
> It’s very important that any changes made to settings in the Editor must be sent to the Ohm64. This does NOT happen automatically. For example if you assign Note 10 to button ID 3, the Ohm64 will not register that change until you press the **Send** button:
>
>
>
>
> Equally important is the **Save To Ohm** button:
>
> 
>
>
> Once you press **Save to Ohm**, the settings currently on the Ohm64 will be recalled the next time you plug the Ohm64 into the USB port on your computer. If you do not press **Save**, your current settings will be lost once the Ohm64 loses power. You will not want to press **Save** numerous times in an editing session, however, as the flash memory has a finite lifespan of approximately 10,000 cycles. This is a lot, but excessive saving could exhaust

2. Open the Editor application. The port menu in the lower left corner should read "Ohm64" (Windows XP will only show “USB Audio Device” and, for the MIDI jack on the back panel, “USB Audio Device[2]”). If an Ohm64 is not found, you will get a very obvious message at the base of the grid:



No Ohm was detected. Check your USB connection, then click here.

Once properly connected, the Editor retreives the Ohm64's current settings from the hardware. You can then click or use the Ohm64 itself to select a knob, slider, or buton, and use the Inspector to change the values and types of messages.



G1 id:[0]

OUTPUT

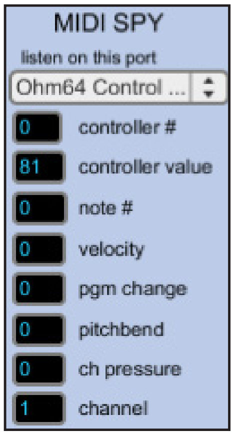
value 0

type note

The Inspector displays some fields to change values and types of controls an element will send/receive, and also displays the name and ID of the element.The IDs of buttons range from 0-80, and from 128-153 for sliders/knobs. For example, to change the MIDI message the second slider on the left to send on controller 25, click on the slider, then click on the **value** number box. Enter "25". Then, make sure the **type** menu is selected as "controller". To send these values to the Ohm64, press **update**, or just click on another control element. If you are warned that the note or cc number you chose is already assigned to something else, then chose a different value.

3. When the **Lock** button is set in the inspector (buttons only),this ensures that MIDI going to the Ohm64 to control LEDs is the same value that the button sends out. This is mostly for convenience, rather than necessity. If you press the Lock button, the inspector will expand to reveal settings for the LEDs, independent of the output messages.

4. You can confirm the Ohm64, or any other device, is sending the proper MIDI by clicking the **MIDI Spy** button to open a panel that displays incoming MIDI messages.

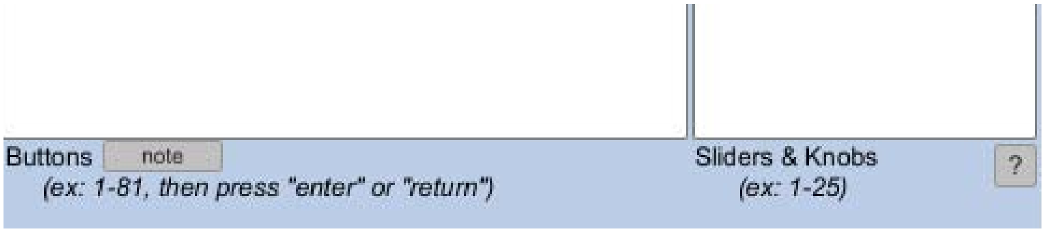


MIDI SPY

listen on this port: Ohm64 Control ...

0	controller #
81	controller value
0	note #
0	velocity
0	pgm change
0	pitchbend
0	ch pressure
1	channel

5. **QuickAssign** is a quick way to assign values to all the controls, just by typing in a few values.



Buttons note

(ex: 1-81, then press "enter" or "return")

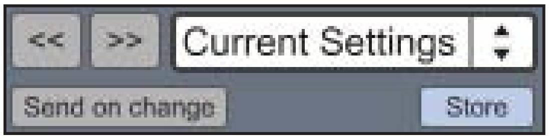
Sliders & Knobs

(ex: 1-25)

For example, you can make a list of numbers that will, in order of the ID number, assign notes to the buttons or slider/knobs. A list such as
32 33 34 35 36 37 38 39
when entered into the buttons field, would assign those notes to the top row of the grid (ID 0-7).
You can also enter ranges of numbers such as

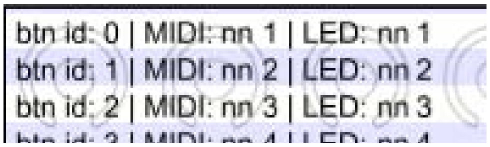
1-81
or even
10-17, 20-27, 30-37, 40-47, 50-57, 60-67, 70-77, 80-87, 100-109, 110-106
which would assign the eight rows of the grid, the slider and crossfade buttons (with 100-109), and the F1-F6 and BPM buttons with the notes defined by the ranges.

6. **Presets** are a convenient way of storing a variety of Ohm64 MIDI mappings on your computer, which can be quickly loaded onto the Ohm64. You can select presets with the arrow buttons, or by choosing from the menu. Presets can be renamed by clicking on the name, then entering in a new name. Press **Store** to save the current settings to a file on your computer. The toggle button **Send on Change** will send all the settings to the Ohm64 when the preset is selected. If it's off, then you will need to press the **Send** button to put a preset’s settings data onto the Ohm.

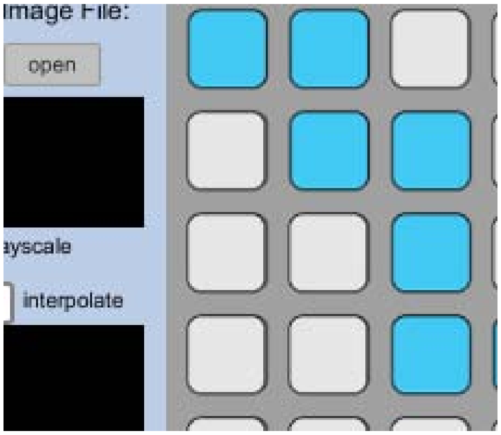


7. **Save to Ohm** is very important. When the Ohm64 is connected to power, it will load its settings from Flash memory. If you want the current settings to load when powered up, then you will have to press Save to Ohm.

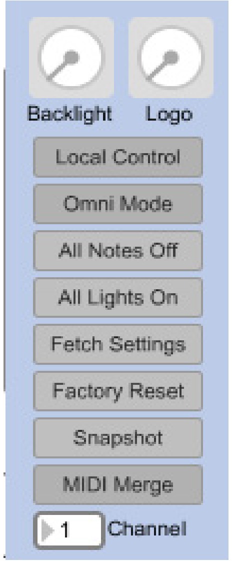
8. **Show Assignments** provides a spreadsheet-style overview for all the assignments for the controls. It is a convenient display for when you want a quick look at everything.



btn id: 0 | MIDI: nn 1 | LED: nn 1
btn id: 1 | MIDI: nn 2 | LED: nn 2
btn id: 2 | MIDI: nn 3 | LED: nn 3
btn id: 3 | MIDI: nn 4 | LED: nn 4



9.**LEDs** reveals a panel that gives control over lighting the Ohm64 LEDs in patterns. These patterns are instantly displayed on the attached Ohm64, and saved with the preset. Additionally, you can load still images (JPEG, PNG, TIFF, etc.) which will be reduced to a resolution of 8x8 pixels. There are controls that determine how the image is scaled and reduced to a 64 pixel, 1 bit image.



10. The **More** button shows a panel with some other controls. The two dials give you control over the backlighting and logo brightness. These dials are merely sending CC values: refer to the MIDI documentation for more information.

When **Local Control** is off, the Ohm64's LEDs will only blink in response to incoming MIDI messages. When Local Control is on, the LED's will blink under the buttons that are pressed, and will ignore any incoming MIDI.

Omni Mode determines if the Ohm64 will listen for control data on all channels (Omni on) or just the specified channel (Omni off).

All Notes Off is a standard MIDI message that will turn off all LEDs (excepting the backlights). It is necessary to have Omni Mode off for this to work.

All Lights On will send a sysex command to turn all LEDs on.

Fetch Settings will request all settings data from the Ohm and populate the Editor with those settings.

Factory Reset will reset the Ohm to the standard settings it shipped with, as well as fetch the settings to populate the Editor.

Snapshot sends out all the button and knob values over MIDI, in the event you want your software values to match up with the hardware.

MIDI Merge, when on, will send MIDI data from the Ohm64 controls out the MIDI Out jack on the back panel, merging it with any data coming into the MIDI Input jack.

The send/receive channel can be changed with the channel number box. True to MIDI specification, the Ohm64 also respects Omni Mode. When on, the Ohm64 will respond to messages on ANY channel for LED control, when off, it will listen only on the specified channel. Omni mode does not affect outgoing MIDI, so the Ohm64 will send only on the specified channel.

11. The **Updates** button takes you to our website to keep you informed of all the new projects, software, and examples for the Ohm64. These will mostly be open-source programs so you will be free to modify them as desired.

12. There’s a few more things to know about the Inspector that are not entirely obvious. Buttons can be programmed for notes and controllers, but they can also be programmed to **send MIDI Machine Control (MMC)** messages “Start,” “Continue,” and “Stop.” To do this, select “control” as the type, and set the number to:

122 = Start
123 = Continue
124 = Stop

Buttons can only send controller numbers 0-120, so we used higher numbers to squeeze in MMC.

Sliders can be programmed to **send Pitch Bend data**, too. Select “cc/bend” as the type, and used values 96-111 to send Pitch Bend on channels 1-16. Values 1-31 will set the slider to send 14-bit style cc messsages with MSB and LSB, however, it is important to note the data is really only 7 bit.

Support

For questions or support please visit our forums at http://forums.lividinstruments.com. Here you will find answers to many of your questions and ask new ones. We will do our best to answer most of of your questions within 24 hours.

We are also always updating our site with video tutorials and examples here
lividinstruments.com/support_movies.php

Appendix

A. Ohm 64 Sysex

****
SYSTEM EXCLUSIVE COMMAND BYTES

The System Exclusive message includes the official MIDI Manufacturer ID for Livid, a Product number, and the Command number. A "wild card" Product number allows the same commands to be shared across multiple Livid products. These messages can be almost any number of bytes from 7 to (183) bytes.
Note: In this specification, all numbers are hexadecimal unless parenthesized or otherwise noted, in which case they are decimal.

F0 00 01 61 02 NN ... F7

The Product number is 02 for the Ohm64, and the Command numbers, NN, are described below.

****
COMMAND SUMMARY

00 : individual button map (not supported) *
01 : individual control map (not supported) *
02 : Save User Settings
03 : read presets (ignored) *
04 : Set all LED indicators
05 : Recall User Settings
06 : Factory Reset
07 : Request, multiple subtypes:
07.04 : Request all LED indicators
07.06 : Request Control Surface Snapshot
07.08 : Request MIDI Settings (Local Control, etc.)
07.09 : Request Single MIDI Input mapping
07.0A : Request Analog mapping
07.0B : Request Button mapping
07.0C : Request MIDI Basic Channel
07.0D : Request MIDI Output Merge setting
07.0F : Request crossfader Flip setting

08 : MIDI Settings Response
09 : Map Single LED Indicator
0A : Map Analog Inputs
0B : Map Buttons
0C : Set MIDI Basic Channel
0D : Set MIDI Output Merge
0E : Erase entire MIDI Input Map
0F : Set crossfader Flip
7E : NAK Negative Acknowledge
7F : ACK Positive Acknowledge

* Commands 0, 1, and 3 are Ohm "classic" System Exclusive commands. Even though the MIDI Manufacturer ID is different, and the command structure is different as well, these skipped command numbers pay tribute to what came before.

****
COMMAND DETAILS

02 : Save User Settings

F0 00 01 61 02 02 F7
All User Settings which can be remembered will be stored in Flash and automatically restored whenever the Ohm64 is powered up.
The Ohm64 responds with ACK when finished processing this command.

04 : Set all LED indicators

F0 00 01 61 02 04 LL HH LL HH LL HH LL HH LL HH LL HH F7
All (75) LED indicators can be set on or off with this command. There are (12) bytes, or 6 pairs of LL HH. Values must be 7F or less. The first LL HH pair represents column 1, with each subsequent LL HH pair representing the next column. Within each pair, LL represents the first 7 rows, and HH represents the remaining 6 rows. There are (14) bits total for rows, but bit (11) is unavailable and reserved for future use. A 1 bit turns the LED off, a 0 bit turns it on.

The LED columns and rows do not conveniently match up with the physical layout of the Button columns and rows. Using codes as C.R to specify column and row, the LED codes are as follows (all codes are decimal in this table, and all columns and rows are numbered starting at 1 - the first line is the first Button column from first to last row, etc.):

64 grid:
1.1, 3.2, 5.3, 1.5, 3.6, 5.7, 1.9, 3.10,
2.1, 4.2, 6.3, 2.5, 4.6, 6.7, 2.9, 4.10,
3.1, 5.2, 1.4, 3.5, 5.6, 1.8, 3.9, 5.10,
4.1, 6.2, 2.4, 4.5, 6.6, 2.8, 4.9, 6.10,
5.1, 1.3, 3.4, 5.5, 1.7, 3.8, 5.9, 1.12,
6.1, 2.3, 4.4, 6.5, 2.7, 4.8, 6.9, 2.12,
1.2, 3.3, 5.4, 1.6, 3.7, 5.8, 1.10, 3.12,
2.2, 4.3, 6.4, 2.6, 4.7, 6.8, 2.10, 4.12
Left/Right crossfader:
5.12, 6.12
1-8 Slider 'mute':
1.13, 3.13, 5.13, 1.14, 2.13, 4.13, 6.13, 2.14
BPM:

3.14

The Ohm64 responds with ACK when finished processing this command.

05 : Recall User Settings

F0 00 01 61 02 05 F7
Recall User Settings from storage, just like what happens on power up.
The Ohm64 responds with ACK if the User Settings are valid, or NAK when no setting have been saved or the memory is unreadable.

06 : Factory Reset

F0 00 01 61 02 06 F7
This will erase the User Settings so that the Ohm64 is factory fresh.
The Ohm64 responds with ACK when finished processing this command.

07 : Request

F0 00 01 61 02 07 RR .. F7
This command allows several types of information to be requested. Each request type, RR, will be detailed below. Some request messages require additional information bytes to specific which particular item is being requested.
The Ohm64 responds with the specified data.

07.04 : Request all LED indicators

F0 00 01 61 02 07 04 F7
Request state of all LED indicators. The Ohm64 will respond with Command 04, and the bits are interpreted identically.

07.06 : Request Control Surface Snapshot

F0 00 01 61 02 07 06 F7
Request values for all Analog inputs. The Ohm64 will re-scan every analog input and report the value with the usual MIDI Channel Message, even if it has not changed.

07.08 : Request MIDI state (Channel Mode, etc)

F0 00 01 61 02 07 08 F7
Request state of settings that are controlled by MIDI messages, such as Local Control, Omni, Mono/Poly Mode. The Ohm64 will respond with Response Command 08.

07.09 : Request Single MIDI Input mapping

F0 00 01 61 02 07 09 LL HH F7
Request the current mapping of incoming MIDI messages. LL HH is the 8-bit map code defined under Command 09 below. Both values must not exceed 7F. LL holds the lower 7 bits, HH holds the most significant bit. Values of HH greater than 1 are reserved. The Ohm64 will respond with Command 09, which repeats LL and HH followed by the LED code as CR.

07.0A : Request Analog mapping

F0 00 01 61 02 07 0A F7

Request MIDI map for all Analog inputs. The Ohm64 will respond with Command 0A, and the bits are arranged identically.

07.0B : Request Button mapping

F0 00 01 61 02 07 0B F7
Request MIDI map for all Buttons. The Ohm64 will respond with Command 0B, and the bits are formatted identically.

07.0C : Request MIDI Basic Channel

F0 00 01 61 02 07 0C F7
Request MIDI Basic Channel. The Ohm64 will respond with Command 0C.

07.0D : Request MIDI Output Merge setting

F0 00 01 61 02 07 0D F7
Request MIDI Output Merge setting. The Ohm64 will respond with Command 0D.

07.0F : Request crossfader Flip setting

F0 00 01 61 02 07 0F F7
Request crossfader Flip setting. The Ohm64 will respond with Command 0F.

08 : MIDI Settings Response

F0 00 01 61 02 08 XX F7
Individual MIDI settings, such as the Channel Mode flags, are combined into the data of this message, XX. Bit 1 is Omni Mode Off/On. Bit 3 is Local Control Off/On. All other bits, including 0 and 2, are Reserved. They may reflect other settings, but they may change in future releases. Note that this is a response command only, and it will be ignored if received by the Ohm64. Use the official MIDI messages to change these settings.

09 : Map Single LED Indicator

F0 00 01 61 02 09 LL HH CR F7
This command updates the MIDI Input map for a single entry. If HH is 00, then LL specifies the Note number. If HH is 01, then LL specifies the Control number, but only valid control numbers are accepted from 00 to 78. Values of HH above 1 are reserved. CR is the LED indicator Column and Row (see Command 04 for details), or 7F to clear the mapping for the specified entry.
The Ohm64 responds with a Single MIDI Input mapping for the specified message.

0A : Map Analog Inputs

F0 00 01 61 02 0A (25)*[LL HH] F7
This command updates the MIDI map for all (25) Analog inputs. If HH is 00, then LL specifies the 7-bit Control number, but only valid control numbers 00 to 78 are accepted. If HH is 01, then LL selects between 14-bit Control numbers and Pitch Bend. In the latter case, LL between 60 and 6F specifies a Pitch Bend message on Channel (1) through (16), respectively. Otherwise, LL selects a 14-bit Control number, of which the only valid control numbers are 00 to 1F. All other values for LL, 20 through 5F and 70 through 7F are reserved for future use. Values of HH above 1 are similarly reserved.

There are (25) sets of LL HH value pairs in this message, each corresponding to the index of an Analog input. These indices do not conveniently match up with the physical layout. The index codes are arranged as follows (all indices are decimal in this table, and start with 0):

crossfader:
    24
eight faders, from left to right:
    23, 22, 15, 14, 5, 7, 6, 4
upper left knobs:
    17, 16, 9, 8
    19, 18, 11, 10
    21, 20, 13, 12
right knobs:
    3, 1, 0, 2

The Ohm64 responds with ACK when finished processing this command.

0B : Map Buttons

F0 00 01 61 02 0B (88)*[LL HH] F7
This command updates the MIDI map for all (81) Buttons. If HH is 00, then LL specifies the Note number. If HH is 01, then LL specifies the Control number, but only valid control numbers are accepted from 00 to 78. Larger values of LL and HH are reserved, with 7A (start), 7B (continue), and 7C (stop) used for MMC.

There are (88) sets of LL HH value pairs in this message, each corresponding to the index of a Button. These indices do not conveniently match up with the physical layout, and there are seven missing codes near the end which are reserved. The index codes are arranged as follows (all indices are decimal in this table, and start with 0):

grid buttons are:
    0, 8, 16, 24, 32, 40, 48, 56
    1, 9, 17, 25, 33, 41, 49, 57
    2, 10, 18, 26, 34, 42, 50, 58
    3, 11, 19, 27, 35, 43, 51, 59
    4, 12, 20, 28, 36, 44, 52, 60
    5, 13, 21, 29, 37, 45, 53, 61
    6, 14, 22, 30, 38, 46, 54, 62
    7, 15, 23, 31, 39, 47, 55, 63
crossfader
    64, 72
1-8 Slider 'mute' buttons, from left to right, are:
    65, 73, 66, 74, 67, 75, 68, 76
function 1-6:
    77, 78, 79, 69, 70, 71
BPM:
    87

The Ohm64 responds with ACK when finished processing this command.

0C : Set MIDI Basic Channel

F0 00 01 61 02 0C NN F7

This command sets the MIDI Basic Channel to NN, where 00 through 0F are valid and the rest are reserved. During operation, incoming Channel messages are ignored if Omni Mode is Off and the Channel does not match. If Omni Mode in On, then the incoming Channel is not checked, and all messages are interpreted. For outgoing MIDI messages, the Basic Channel is always used unless an Analog input is mapped to Pitch Bend, in which case the Channel number is specified by the mapping entry.
The Ohm64 responds with ACK when finished processing this command.

0D : Set MIDI Output Merge

F0 00 01 61 02 0D NN F7
This command allows the Control Surface to merge its messages onto the MIDI output jack, in addition to USB. NN can be 00 (off) or 01 (on). Other values are reserved.
The Ohm64 responds with ACK when finished processing this command.

0E : Erase entire MIDI Input Map

F0 00 01 61 02 0E F7
This command erases all MIDI Input message mappings. These affect the remote control of LED indicators, so no remote control will be available immediately after the map is erased. Because there are as many as (256) possible MIDI Input messages that can be mapped to an LED, and because multiple assignments to the same LED can cause conflicts, it is best to start the LED indicator mapping process by first erasing all entries before filling in the desired mappings, no more than (75), preferably.
The Ohm64 responds with ACK when finished processing this command.

0F : Set crossfader Flip (left or right as minimum)

F0 00 01 61 02 0F NN F7
This command allows the crossfader response to be flipped. NN can be 00 or 01. Other values are reserved.
The Ohm64 responds with ACK when finished processing this command.

7E : NAK Negative Acknowledge

F0 00 01 61 02 7E F7
This is intended as a way for the Ohm64 to indicate a problem with the previous message. Unfortunately, when messages are sent too fast, it becomes impossible for the Ohm64 to reply with anything reliably, even NAK. The best approach is to wait for ACK (see below) after sending each message, rather than depend on NAK to indicate an error. The one exception is Recall User Settings, where the Ohm64 will reply with NAK if the contents of User memory are empty or otherwise corrupted.

7F : ACK Positive Acknowledge

F0 00 01 61 02 7F F7
ACK means the previous message was handled correctly. Some messages (e.g. Map Single LED Indicator) simply reply with a specific command instead of ACK, when more context is needed than a simple acknowledgment.

****
MAPPING SUMMARY

Analog to CC; 121 control numbers; 7-bit value
Analog to CC; 32 control numbers; 14-bit value
Analog to Pitch Bend; 16 channels; 14-bit value

Button to Note; 128 note numbers; 1-bit value
Button to CC; 121 control numbers; 1-bit value
Button to MMC; 3 types; no value
LED from Note; 128 note numbers; 1-bit value
LED from CC; 121 control numbers; 1-bit value

****

MIDI IMPLEMENTATION

The following standard MIDI messages are implemented.

System Real Time:
MIDI System Reset

Channel Mode Messages:
Reset All Controllers
Local Control (connects buttons to corresponding LED indicators)
Poly On / Mono On (both set Omni Off)
Omni On/Off
All Notes Off (turns off all LED indicators)

Controller Numbers:
1B : backlight, 7-bit dimmer
1F : logo, 14-bit (10-bit effective) dimmer

****

FRONT PANEL CONTROL

The following key combinations allow front panel control of certain settings.

F4+XL Crossfader Flip off (value increase towards the left, default)
F1+XR Crossfader Flip on (value increases towards the right)
F1+M2 Local Control off (LED indicators under external control only)
F1+M4 Local Control on (LED indicators controlled by buttons and external)
F1+M6 Omni off (messages received on Basic Channel only)
F1+M8 Omni on (messages received on all channels)
F2+M6 MIDI out merge off (Control Surface sends on USB only)
F2+M8 MIDI out merge on (Control Surface sends to USB & MIDI out, default)

XL and XR are the Left and Right crossfader bump buttons, respectively.
M2 and M4 are the Mute buttons below faders 2 and 4, respectively.
M6 and M8 are the Mute buttons below faders 6 and 8, respectively.

B. LED Control Column/Row Codes

Controlling LEDs with MIDI: LED Control Column/Row codes

- ⬜ column 1
- 🟦 column 2
- 🟨 column 3
- 🟩 column 4
- 🟫 column 5
- ◻ column 6

Here's why the column/row numbers used in the sysex strings for creating LED MIDI control assignments are weird.
There are 6 columns, rather than 8, of LEDs to keep brightness at a high level. The wiring makes it so these "coumns" don't align visually. However, if you follow the colors, you can get a good idea of the pattern. For example, to assign note# 10 to the LED on note 10, you send the sysex string:
F0h 00h 1h 61h 02h 09h Ah 0h 41h F7h
(decimal: 240 0 1 97 2 9 10 0 65 247)
In general:
F0h 00h 1h 61h 02h 09h LL HH CR F7h
Where LL is note ro CC number, HH is a 0 for note, 1 for CC, and CR is the Column/Row number

10, 26, 42, 58, 74, 80 belong to row 11. which is skipped

1 00 00h	2 16 10h	3 32 20h	4 48 30h	5 64 40h	6 80 50h	7 01 01h	8 17 11h
9 33 21h	10 49 31h	11 65 41h	12 81 51h	13 02 02h	14 18 12h	15 34 22h	16 50 32h
17 66 42h	18 82 52h	19 03 03h	20 19 13h	21 35 23h	22 51 33h	23 67 43h	24 83 53h
25 04 04h	26 20 14h	27 36 24h	28 52 34h	29 68 44h	30 84 54h	31 05 05h	32 21 15h
33 37 25h	34 53 35h	35 69 45h	36 85 55h	37 06 06h	38 22 16h	39 38 26h	40 54 36h
41 70 46h	42 86 56h	43 07 07h	44 23 17h	45 39 27h	46 55 37h	47 71 47h	48 87 57h
49 08 08h	50 24 18h	51 40 28h	52 56 38h	53 72 48h	54 88 58h	55 09 09h	56 25 19h
57 41 29h	58 57 39h	59 73 49h	60 89 59h	61 11 0Bh	62 27 1Bh	63 43 2Bh	64 59 3Bh

45 2dh

BPM

12 0Ch	28 1Ch	44 2Ch	60 3Ch

slider buttons 1-4

75 4bh

xfade L

91 5bh

xfade R

76 4Ch	92 5Ch	13 0Dh	29 1Dh

slider buttons 5-8

C. Assigning Buttons MIDI Values with SySex

Assigning Buttons MIDI Values with SySex: a very long string.

The buttons are assigned MIDI values with a very long sysex string of 183 bytes. There is no command to set a single button, instead all buttons are set at once. This may seem inconvenient to program, it is, however, efficient for the hardware. The string follows this pattern:
F0 00 01 61 02 0B (88)*[LL HH] F7
This command updates the MIDI map for all (81) Buttons. If HH is 00, then LL specifies the Note number. If HH is 01, then LL specifies the Control number, but only valid control numbers are accepted from 00 to 78. Larger values of LL and HH are reserved.The positions of the buttons in the string are mapped on this graphic, where position "0" in the string corresponds to button 1, and position "7" of the LL HH pairs is button 57.

1 00	*2* 08	*3* 16	*4* 24	*5* 32	*6* 40	*7* 48	*8* 56
9 01	*10* 09	*11* 17	*12* 25	*13* 33	*14* 41	*15* 49	*16* 57
17 02	*18* 10	*19* 18	*20* 26	*21* 34	*22* 42	*23* 50	*24* 58
25 03	*26* 11	*27* 19	*28* 27	*29* 35	*30* 43	*31* 51	*32* 59
33 04	*34* 12	*35* 20	*36* 28	*37* 36	*38* 44	*39* 52	*40* 60
41 05	*42* 13	*43* 21	*44* 29	*45* 37	*46* 45	*47* 53	*48* 61
49 06	*50* 14	*51* 22	*52* 30	*53* 38	*54* 46	*55* 54	*56* 62
57 07	*58* 15	*59* 23	*60* 31	*61* 39	*62* 47	*63* 55	*64* 63

45

BPM

10, 26, 42, 58, 74, 80 belong to row 11. which is skipped

65	73	66	74

slider buttons 1-4

64

xfade L

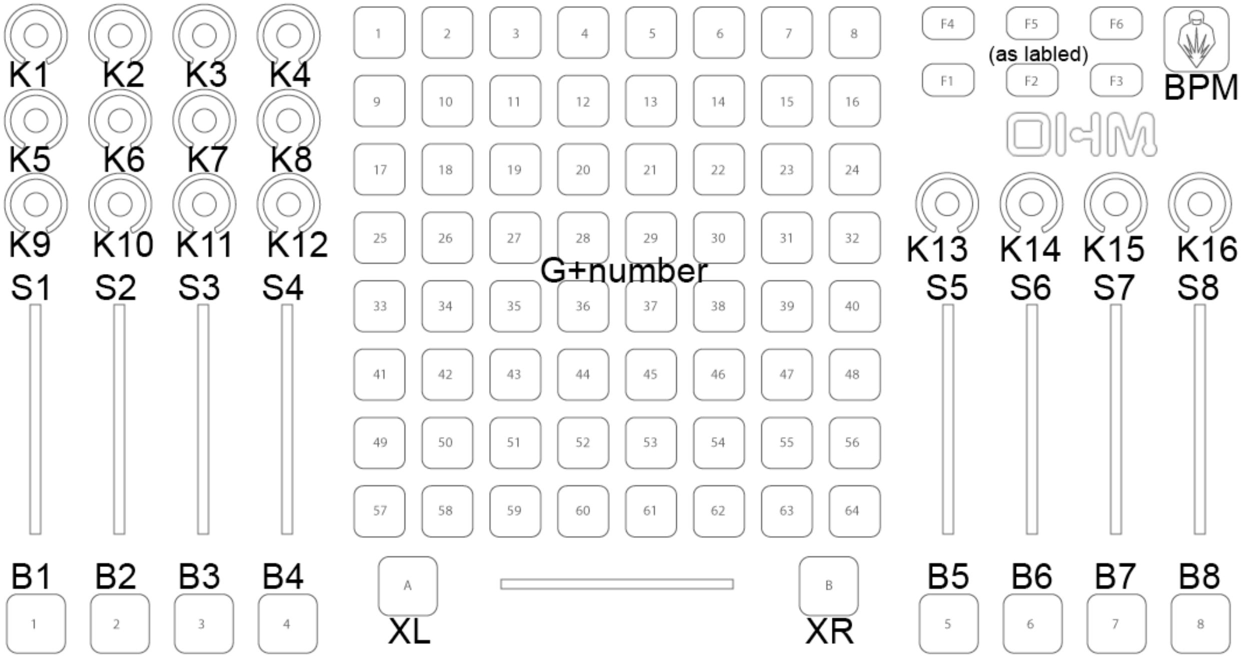
72

xfade R

67	75	68	76

slider buttons 5-8

D. Controls Naming Convention



Warranty

Ohm64 WARRANTY AND RETURN POLICY

Damage
Your Instrument should be inspected immediately upon receiving it. If there are any damages they should be reported immediately. All of our instruments are fully tested just prior to leaving our plant.

Return Policy
All sales of Livid Instruments hardware are given a fifteen (15) day return policy. A controller may be returned within fifteen (15) days of purchase. We will refund the purchase price of the product, less shipping, immediately upon receipt of the returned item in as good of shape as it was shipped in. Any damage will be the responsibility of the customer. We will do our best to determine if damage happened during shipping and will help the customer by allowing carrier inspection of any damaged items. Software cannot be returned. Any open software packaging is not returnable or available for any refund.

Warranty
Livid Instruments warrants that this product is free of defects in materials and workmanship under normal conditions for one year from the purchase date, so long as the product is: owned by the original purchaser; the original purchaser has proof of purchase from Livid Instruments or an authorized dealer. Warranty does not include repairs needed due to misuse or abuse.

In the event that Livid Instruments receives written notice of defects in materials or workmanship from such an original purchaser, Livid Instruments will either replace the product, repair the product, or refund the purchase price at its option. In the event any repair is required, shipment to and from Livid Instruments and a nominal handling charge shall be born by the purchaser. In the event that repair is required, a Return Authorization (RMA) number must be obtained from Livid Instruments. After the RMA number is obtained, the unit should be shipped back to Livid Instruments in a protective package with a description of the problem and the Return Authorization (RMA) clearly written on the package.

In the event that Livid Instruments determines that the product requires repair because of user misuse or regular wear, it will assess a fair repair or replacement fee. The customer will have the option to pay this fee and have the unit repaired and returned, or not pay this fee and have the unit returned un-repaired.

THE REMEDY FOR BREACH OF THIS LIMITED WARRANTY SHALL NOT INCLUDE ANY OTHER DAMAGES. LIVID INSTRUMENTS WILL NOT BE LIABLE FOR CONSEQUENTIAL, SPECIAL, INDIRECT, OR SIMILAR DAMAGES OR CLAIMS INCLUDING LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGE, EVEN IF ITS AGENTS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IN NO EVENT WILL LIVID INSTRUMENTS' LIABILITY FOR ANY DAMAGES TO THE PURCHASER OR ANY OTHER PERSON EXCEED THE PRICE PAID FOR THE PRODUCT, REGARDLESS OF ANY FORM OF THE CLAIM. LIVID INSTRUMENTS SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED. SPECIFICALLY, LIVID INSTRUMENTS MAKES NO WARRANTY THAT THE PRODUCT IS FIT FOR ANY PARTICULAR PURPOSE.

This warranty shall be construed, interpreted, and governed by the laws of the state of New Jersey. If any provision of this warranty is found void, invalid or unenforceable, it will not affect the validity of the balance of the warranty, which shall remain valid and enforceable according to its terms. In the event any remedy hereunder is determined to have failed of its essential purpose, all limitations of liability and exclusion of damages set forth herein shall remain in full force and effect.

Livid Instruments, Viditar Inc.

The Ohm64 is handcrafted and assembled in The United States of America.