**“Unlock Your Midi Creativity”**

**MidiXLate Midi Router**

**User Manual**

**Version 3.1 March 2023**



**Sierra Digital Audio Phone: 925-765-2883**

**13155 Falcon Point Place Website:** [**SierraDigitalAudio.com**](http://www.SierraDigitalAudio.com)

**Truckee, CA 96161 USA Support:** [**lscottae@gmail.com**](mailto:lscottae@gmail.com)

**Table of Contents**

1: [Important Safety Instructions](#B1)

2: [Legal Disclaimer](#B2)

3: [Limited Warranty](#B3)

4: [Product Registration](#B4) and Support

5: Specifications (fix numbering)

5: [Introduction to the MidiXLate Router](#B5)

6: [Requirements and Setup](#B6)

7: [Faceplate Overview](#B7)

8: [Rearplate Overview](#B8)

9: [MidiXLate Software](#B9) Reference

9.1: [Download, Install, Launch](#B91)

9.2 [Navigation, menus, and buttons](#B92)

9.3: [Creating a New Configuration](#B93)

9.4: [Midi Routing](#B94)

9.5: [Midi Translation](#B95)

9.6: [Midi Channel Locking](#B96)

9.7: [Saving a Configuration](#B97)

9.8: [Download Configuration to MidiXLate Router](#B98)

10: [Specifications](#B10)

11: [Technical Notes](#B11)

**Appendix A**: [MidiXLate Configuration File Format](#AppendixA)

**Appendix B**: [MidiXLate Logger Format](#AppendixB)

**Appendix C**: [MidiXLate Instructional Videos](#AppendixC)

**Appendix D:** [Midi Concepts from Roland](#AppendixD)

**Appendix E:** [Midi Concepts from McGill](#AppendixE)

**1: Important Safety Instructions**

Use only high-quality commercially-available midi cables, usb cables, and the provided power supply

All other installation or modifications should be performed only by qualified personnel.

Please read the following cautions and recommendations.

**Cautions**

* Do not remove the face plate or rear plate.
* To reduce the risk of fire or electric shock, do not expose this router to rain or moisture.
* No objects filled with liquids, such as vases, should not be placed on the router.
* Do not install the router near any heat sources such as radiators, heat registers, or amplifiers.
* Clean only with dry cloth.
* Do not block any ventilation openings.
* Read these instructions.
* Keep these instructions.
* Heed all warnings.
* Follow all instructions.
* All repairs need to be performed by Sierra Digital Audio

Power is delivered via the 9 volt external power supply (provided with the router). Make sure that the power supply is grounded at all times. For your own protection, you should never tamper with the grounding of the power supply. The power supply must always be connected to a 120 volt power outlet with a standard protective ground connection.

**2: Legal Disclaimer**

**2.1: Legal Disclaimer**

Technical specifications and appearance are subject to change without notice. The information contained in this manual is correct at the time of printing. All trademarks are the property of their respective owners.

Sierra Digital Audio accepts no liability for any loss which may be suffered by any person who relies either wholly or in part upon any description, photograph or statement contained herein. Colors and specifications may vary slightly from product.

Sierra Digital Audio products are only sold by Sierra Digital Audio. Distributors and dealers are not agents of Sierra Digital Audio and have absolutely no authority to bind Sierra Digital Audio to any express or implied undertaking or representation.

This manual is protected under an unpublished copyright. No part of this manual maybe reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording of any kind, for any purpose, without the express written permission of Sierra Digital Audio.

ALL RIGHTS RESERVED, 2023

Sierra Digital Audio

13155 Falcon Point Place

Truckee, CA 96161 USA

[www.SierraDigitalAudio.com](file:///C:\Users\LarryDell\AppData\Roaming\Microsoft\Word\www.SierraDigitalAudio.com)

[www.scottae.github.com](file:///C:\Users\LarryDell\AppData\Roaming\Microsoft\Word\www.scottae.github.com)

[LScottAE@GMail.com](mailto:LScottAE@GMail.com?subject=MidiXLate%20Router%20Inquiry)

**3: Limited Warranty**

**3.1: Terms and Conditions**

[1] The warranty is valid only if you purchased this product directly from Sierra Digital Audio.

[2] The warranty is not transferable.

[3] Electronic components of this product are guaranteed to be free of defects in material and workmanship if used under normal operating conditions for a period of one year from the original date of purchase.

[4] If the product shows any defects within the specified warranty period, Sierra Digital Audio shall, at its discretion, either replace or repair the router using suitable new or reconditioned parts. If Sierra Digital Audio decides to replace the router, the original limited warranty will apply to the replacement product for the remaining initial warranty period, i.e., one year from the date of purchase of the original product.

[5] Upon validation of the warranty claim, the repaired or replacement product will be returned to the user freight  
prepaid by Sierra Digital Audio.

[5] Warranty claims other than those indicated above are expressly excluded.

**PLEASE RETAIN YOUR SALES RECEIPT. IT IS YOUR PROOF OF PURCHASE COVERING YOUR LIMITED WARRANTY.  
THIS LIMITED WARRANTY IS VOID WITHOUT SUCH PROOF OF PURCHASE.**

**3.2: Warranty Registration**Please take a few minutes and fill out the registration card within 14 days of the date of purchase.

Register online at [www.SierraDigitalAudio.com](file:///C:\Users\LarryDell\AppData\Roaming\Microsoft\Word\www.SierraDigitalAudio.com)

The serial number needed for the registration is located on the rear of the unit.

Failure to register your product will void future warranty claims.

**3.3: Return materials authorization (RMA)**

If you need to return your router to Sierra Digital Audio for warranty service, please fill out the Warranty Request form at [www.SierraDigitalAudio.com](http://www.SierraDigitalAudio.com). All warranty claims must have a Sierra Digital Audio generated RMA BEFORE the router is returned by Sierra Digital Audio. Routers received by Sierra Digital Audio without an RMA will not be accepted.

Warranty inquiries must be made via our website [SierraDigitalAudio](http://www.SierraDigitalAudio) and must be accompanied by your contact information, a description of the problem, and the serial number of the router. After verifying the product’s warranty eligibility, Sierra Digital Audio will issue a Return Material Authorization (“RMA”) number. The router is shipped UPS Ground by the customer. The returned router is shipped UPS Ground be Sierra Digital Audio. Please see the [SierraDigitalAudio](http://www.SierraDigitalAudio) support page for detailed RMA instructions.

**4: Product Support**

**4.1: User Manual**  
This user manual is designed to give you both an overview of the router hardware and software, as well as detailed information on how to use them. If you need more specific information about the router hardware or software, please visit the support page on our website at www.SierraDigitalAudio.com, where you’ll find additional information on issues, resolution, and enhancements to MidiXLate router.

**4.2: Product Support**

Sierra Digital Audio will provide presales support and technical support for customers who register their routers. For product support, please send your support requests to [LScottAE@GMail.com](mailto:LScottAE@GMail.com?subject=MidiXLate%20Router%20Inquiry) . We will review your request and typically respond within 48 hours. For simple requests, we will most likely respond by email. For more difficult requests, we may coordinate a phone call or [zoom](https://zoom.us/signin#/login) meeting.

**5: Specifications**

**General specifications of the MidiXLate router.**

|  |  |
| --- | --- |
| Size: width, height, depth | 16.4” wide x 3.8 in high x 6.0” deep |
| Weight | 4 lbs |
| Power | 9V wall transformer, 1500 ma (supplied) |
| Processor | [Atmel 32 bit, Cortex-M3, 84MHz](http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf) |
| Ports In | 6 SPI |
| Ports Out | 6 SPI |
| Display | 3.5 inch color 320x480 pixel |
| Logger port | Micro USB |
| Configuration port | Micro USB |
| Data persistence | Micro SD simm card, removable |
| USB Device driver | None, future enhancement |

\*\* Logger port and configuration port share the same micro usb socket.

**5: Introduction to the MidiXLate Router**

This manual assumes the reader has a full grasp of Midi concepts, Midi message types (such as ‘program change’, ‘control change’, and ‘note on/off’) and Midi message formats (such as 2 byte ‘program change’ and 3 byte ‘control change’). As a reference, this manual provides an appendix containing a rudimentary [introduction to Midi](#B161), Midi message types, and Midi message formats.

**5.1: A TCP/IP router analogy**

**To understand the MidiXLate router, let’s first consider the functionality of your TCP/IP router in your home. Your TCP/IP router** enables every device connected to your router (wired or WiFi) to communicate with every other device on your home network and the public internet. Most devices on your home network, such as your cell phone, simply connect (via your router) to the internet, while some devices connect to other devices on your local home network. For example your PC connects to your network printer or you might ‘cast’ video from your cell phone to your ChromeCast TV. These devices are connected to your home router via (wired or WiFi connections) which allows the devices to ‘talk’ to each other.

On your home network, most of your devices (example cell phone) connect to your router via DHCP (Dynamic Host Configuration Protocol) and WiFi, meaning your router recognizes the device’s attempt to connect to the router, your router’s DHCP service provides a TCP/IP address to your device (cell phone), and your device (cell phone) is connected to your network and the public internet via WIFI. Alternatively, you can physically connect devices (such as PCs and printers) to your router via a CAT5 network cables. The DHCP process performs the same function (of TCP/IP address provisioning) regardless if the connection is WiFi or wired.

In addition to configuring and connecting TCP/IP devices, your TCP/IP router can also perform ‘Network Address Translation’ or NAT. NAT occurs when a PC on your network makes a request to the public internet. When you search Google for your favorite restaurant, your PC (which has a local DHCP IP address) sends a request to your router where NAT translates your PC’s local DHCP IP address to your router’s public IP address before sending the search request to Google (on the public internet). Google’s search engine / server then services the request (builds a list of restaurants) and returns the list to your router, reverse NAT (public IP to local network translation) is performed, and the router sends the restaurant list from the router to your local PC where the restaurant list is displayed.

Now that we have a basic understanding of TCP/IP routing, DHCP, and NAT, let’s look at MidiXLate.

The MidiXLate router provides similar functionality to the TCP/IP functionality just described, except on a Midi network (not a TCP/IP network). On a midi network there is no DHCP. Midi devices are identified by the ‘device driver’ provided with the Midi device, for example your Midi keyboard ships with a CD installation disk that tells your PC how to recognize and ‘talk to’ your keyboard. MidiXLate currently requires Midi devices to be connected to the MidiXLate router via a hardwired Midi cable. Wireless (Wifi) Midi will be considered in a future release.

MidiXLate configuration software (Windows only) is downloaded from the [Sierra Digital Audio](http://SierraDigitalAudio.com) website. Once installed, you configure MidiXLate’s ‘routing’ (required), ‘midi translation’ (optional), and ‘midi channel locking’ (optional) functionality (discussed below). The configuration can be saved to your PC desktop (for a library of router configurations) or downloaded directly MidiXLate’s microSD card.**5: Introduction to the MidiXLate Router (cont)**

**5.2: Why MidiXLate: A Case study**

Many music lovers have experienced Midi frustration of trying to connect dissimilar devices that just don’t want to talk to each other. For example, consider you want to use controllers (buttons, sliders, and knobs) on your keyboard control a DAW (Reason, Reaper, Pro Tools, …) functions on your PC. There are many potential issues that can leave you shaking your head.

For example, most Midi configurations will have you configure each midi device to ‘listen’ for Midi commands on 1 of 16 available channels. Thus, you assign a Midi channel to each device in your rack. Now, along comes a program like Reason (a PC virtual rack of synthesizer). In Reason, each virtual rack synth is controlled on it’s own channel. Now, rather than having Midi channels assigned to your physical devices (pedals), you need all 16 channels assigned to you PC Reason (virtual) devices! MidiXLate allows you to overcome this physical synth **/ virtual synth conundrum using MidiXLate ‘routing’ ??????**

Next, very often keyboards will be not allow one controller to send commands on one channel and an adjacent controller to send commands on a different channel. For example, assume you want keyboard slider 1 to control the volume on my PC’s rack synth 1 (assigned to channel 1) and keyboard slider 2 to control the volume on my PC rack synth (assigned to channel 2), but your keyboard does not support issuing midi commands on multiple channels simultaneously. Using MidiXLate, you have two options to solve this problem. You can either instruct MidiXLate to intercept a Midi command (program change, controller, or note on/off channel) and ‘**translate**’ it to your desired Midi (program change, controller, or note on/off channel) command. Your second option is to use the buttons on your keyboard to ‘**channel** lock’ all midi commands sent from your keyboard to a particular channel. For example, by pressing button 1 on your keyboard, your keyboard now sends all keyboard (note, and control) data to PC (Reason Malstrom) virtual synth. By pressing button 2 on your keyboard, your keyboard now sends all keyboard (note, and control) data to PC (Reason NN19) virtual synth. This provides a very easy method to switch between PC virtual synth devices.

Now, for the icing on the cake: The Midi **routing** function. With Midi ‘routing’ (required configuration), you specify which midi devices are connected to which other midi devices. For example, let’s return to the example above where your keyboard is connected to the PC synth. By configuring MidiXlate to ‘route’ all keyboard commands to only your PC synth, you can now use all 16 midi channels to control just your PC synth.

Additionally, you can simultaneously route your keyboard data to a ‘stand alone’ tone generator.

The routing point that is often overlooked is; On a Midi network, you configure devices to listen on a channel and quite often, you need to configure other devices to not listen on a channel thus reducing unwanted side effects. With MidiXLate, each midi device is listening on a segment of your Midi network, not the entire midi network. This provides you with a much more hardened, bulletproof, robust, and secure network.

The MidiXLate router is designed to provide enhanced Midi functionality and control over your Midi network. MidiXLate provides discrete Midi routing, granular Midi translation, and simple Midi channel locking features. Before explaining the benefits of the MidiXLate router, let’s start with a brief mention of the Midi issues that MidiXLate endeavors to solve.

**The limitation of USB to Midi converter cables**

This solution is a simple point-to-point midi connection. For example, assume you connect a keyboard (midi connector) to your PC DAW (USB connector). This is a Midi point to point connection. Point to point midi is very limited because you have to work solely within the Midi constrains of your keyboard and PC. Foe a simple example, assume your keyboard only operates over one Midi channel but your PC DAW requires control over multiple channels. There may be no elegant solution. Lastly, point to point Midi does not allow simultaneous control of multiple Midi devices. Using Midi ‘thru ports’ is typically less than ideal.

**The problems with Midi Hubs**

Midi hubs can be a nightmare in large Midi networks because Midi hubs typically broadcasts Midi data recived from one port to all Midi devices connected to the hub. Managing this Midi network is very difficult and fought with unwanted side effects. Not only do you need to manage the Midi tasks you want to control, you also need to manage the Midi tasks you don’t want (ie: unwanted side effects). Hubs are dumb devices that generate excess Midi network traffic, increase network latency, and provide no software control, no configurable routing capability, and no software configurable midi translation

The following three sections will explain how MidiXLate is a better solution.

[5.3: Midi Routing](#B53)

[5.4: Midi Translation](#B54)

[5.5 Midi Channel Locking](#B55)

**5: Introduction to the MidiXLate Router (cont)**

MidiXLate has three modes of operation:

* Run Mode: The router is processing incoming and outgoing Midi Data
* Logger Mode: The router is dumping midi activity to a terminal session to troubleshoot data.
* Configure Mode: The router is receiving configuration information for a Windows PC

MidiXLate provides three types of Midi Processing:

* Routing: The router sends Midi data to one or many ports
* Translation: The router translates Midi data from one midi format to another midi format
* Channel Locking: Locks midi outbound data to a specific channel regardless of inbound channel.

Most music equipment provides a ‘Midi In’ port and ‘Midi Out’ port (and sometimes a Midi Thru port) to typically control ‘patch’ (preset) selection, control changes, and note on / note off commands. Midi devices are connected to each other using Midi cables connecting device Midi Input, Midi Out, and Midi Thru ports.

MidiXLate has 6 input ports and 6 output ports

MidiXLate requires physical connections of music equipment to the MidiXLate router via a Midi cable. Once the physical midi cable connections are made to the router, a Windows PC program (free download from the Sierra Digital Audio website) is used to describe / configure the MidiXLate router ports, any address translation, and any channel locking functions you require.

As of this MidiXLate Version 3.3, MidiXLate connections to a Windows PC DAW or Synth requires a Midi to USB conversion cable such as this [Amazon Midi USB cable](https://www.amazon.com/FORE-Interface-Converter-Adapter-Laptop/dp/B0719V8MX1/ref=sr_1_4?crid=114Q7BC21NVS3&keywords=usb+to+midi&qid=1673642815&sprefix=usb+to+m%2Caps%2C1127&sr=8-4). If you do not intend to connect MidiXLate to a Windows PC DAW or Synth (such as PropellerHead Reason), then you do not need this cable.

MidiXLate requires a USB 2.0 type A to Micro USB type B cable such as this [Amazon USB 2.0 type A to Micro USB type B](https://www.amazon.com/AmazonBasics-Male-Micro-Cable-Black/dp/B07232M876/ref=sr_1_4?crid=373CBX19WDAUO&keywords=usb%2Bmicro%2BB&qid=1673643665&sprefix=usb%2Bmic%2Caps%2C2882&sr=8-4&th=1) to download PC configurations to MidiXLate. This cable is required.

More detail is provided below.

**5: Introduction to the MidiXLate Router (cont)**

**5.2: Midi Connections**

MidiXLate currently supports physical midi cable connections to the router. In the future, USB connections will be provided. For now, USB to MIDI conversion cables provide a good solution where USB Midi connections are required (for example, your router to a PC)

**5.3: Midi Routing**

Midi Routing is the process of forwarding midi traffic from a Midi In port to one or many Midi Out port(s).

MidiXLate has six input ports and six output ports. An example of midi routing would be to connect a Midi keyboard to MidiXLate ‘in port 1’ and route midi traffic from ‘in port 1’ to a PC synth connected to MidiXLate ‘out port2’ and a Yamaha MU15 tone generator on MidiXLate ‘out port 3’. Notice midi traffic coming in one port can be routed to one or many output ports. Routing is more efficient than a hub (which broadcasts to all connected dives), more robust (management is easier thus reducing unwanted side effects), reduces latency by reducing network traffic) , and increases network Midi network performance. For more information: [Software Midi Routing](#B94)

**5.4: Midi Translation**

[Midi Translation](#B95) is the process of receiving Midi data from one port and translating it to another midi format before sending the Midi data out an output port. Currently, MidiXLate support translation of Midi Program Change, Control Change, and Note On/Off messages. For example, you might receive a Midi Program Change command on one port and ‘translate it’ to a Midi Control Change command and send it out another port.

For more information: [Software Midi Translation](#B95) and [Technical Notes - Midi Translation](#B11)

|  |  |
| --- | --- |
| **Supported Midi Translations** | |
| 1: Program Change to Program Change | 4: Program Change to Control Change |
| 2: Control Change to Control Change | 5: Program change to Note On/Off |
| 3: Note On/Off to Note On/Off | 6: Control Change to Program Change |
|  | 7: Control Change to Note On / Off |
|  | 8: Note On / Off to Program Change |
|  | 9: Note On / Off to Control Change |

Translation is only provided for the Midi message formats listed above. The remaining [midi message types](#B164) are routed as assigned without any change. For example, a ‘pitch wheel’ Midi message will be routed (as you assign) without any changes to the incoming midi data.

**5.5: Midi Channel Locking**

[Midi Channel Locking](#B96) provides a way to lock all incoming Midi data, received from an input port, to a specific channel on an outgoing Midi port. Channel locking ignores the channel on the received midi command. For example: On your keyboard KB (Midi In 1) you have buttons (controllers), knobs (controllers), sliders (controllers), and keys (note on/off). Assume you are working with PC DAW Reason 10 (Midi Out 2) and you want to switch between Reason 10 rack devices by pushing controller buttons on your KB. In MidiXLate, Assign a channel lock to Midi In port 1 / Midi Out port 2, controller X (where X is your KB button/controller number). Now, with a simple push of a KB controller / button on your keyboard, you seamlessly ‘lock’ keyboard notes (note on/off commands) to a specific Reason 10 rack device.

For more information: [Software Midi Channel Locking](#B96) and [Technical Notes – Channel Locking](#B11).

**5: Introduction to the MidiXLate Router (cont)**

**5.6: Logging Port**

The Logging button on the face of the router and the LOG / CFG USB port on the rear of the router are used to troubleshoot (view) Midi traffic passing the router. In all honesty, this button should have been placed on the rear of the router (maybe in a future release). MidiXLate logging is used to troubleshoot configuration issues and programming bugs. If you encounter issues with the router, send a log dump to [Sierra Digital Audio](http://www.SierraDigitalAudio.com) for free troubleshooting assistance.

When logging is engaged, the router will function as normal, but write volumes of activity information to your PC terminal emulator. Router latency (due to all of the terminal emulation screen IO) will make the router operate very slowly. Do not turn on logging during a live performance!

To start a troubleshooting session:

**1:** Download and install [PuTTY](https://www.putty.org/) (free) or any other [terminal emulation software](https://www.puttygen.com/windows-terminal-emulators) on your PC.

**2:** On the rear of the router is a LOG / CFG USB port.

Connect your USB A to Micro USB cable to your PC and MidiXlate LOG / CFG port on the router.

Keep the USB cable shorter than six feet. Ideally use a three foot cable.

**3:** Make sure the Config button on the face of the router is off (not illuminated).

The Logging button and the Configuration button cannot be use simultaneously!

**4:** Press the Logger button on the face of the router to activate logging.

The logging button will illuminate.

**5:** Press a note on the keyboard, select a patch change, or press a keyboard controller.

All of the internal MidiXLate activity will be dumped to your PC terminal emulation screen.

**6:** If you are having problems with a particular configuration, clear your terminal emulator screen

and recreate your problem with logging on. Send your log dump to [Sierra Digital Audio](http://www.SierraDigitalAudio.com) along with

a description of your problem. Please include a copy of your MidiXLate router configuration file.

**7**: Turn off logging to end your troubleshooting session.

**5: Introduction to the MidiXLate Router (cont)**

**5.7: Configuration Port**

The Configure button on the face of the router and the LOG / CFG USB port on the rear of the router are used to download the router configuration from your PC to the router.

The MidiXlate router configuration is created on your Windows PC, saved on your Windows PC, and downloaded to the router’s MicroSD card. When the router is started, the router’s configuration is read from the routers MicroSD card and the router configuration is loaded to the router’s CPU.

There is no upload feature to upload the router’s configuration to your PC.

The master configuration for your router resides on your PC, in a file you specify.

Multiple router configurations are possible by keeping separate router configurations in separate files.

To configure your router:

**1:** Download and install the MidiXLate routing software as described in [Section 9.1](#B91).

**2**: Launch the MidiXLate installer and create a configuration as described in the [software section](#B9) below.

**3**: Save your router configuration to a PC location of your choosing.

Initially, your desktop is a good place to save your router configuration.

**4:** Physically connect your PC to your router using the a USB cable

On the rear of the router is a LOG / CFG USB port.

Connect your USB A to Micro USB cable to your PC and MidiXlate LOG / CFG port on the router.

Keep the USB cable shorter than six feet. Ideally use a three foot cable.

**5:** Make sure the Logging button on the face of the router is off (not illuminated).

The Configure button and the Logging button cannot be use simultaneously!

**6:** Press the Configure button on the face of the router to enable the router to receive the config.

The configure button will illuminate.

**7:** Follow the [software download steps](#B98) to download the configuration from your PC to the router.

**8**: Once the configuration download is complete, press the Configure button to turn off configuration

mode / return to router run mode.

**Important Note: All USB and midi cable connections to the router must be made when the router power is OFF. Connecting cables with the power on risks damage to your music device, the MidiXLate router, and will cause midi connections to become unstable (stop functioning).**

**6: Requirements and Setup**

**6.1 Requirements**

* A Windows PC to configure the router
* A [PC USB to Midi cable](https://www.amazon.com/FORE-Interface-Converter-Adapter-Laptop/dp/B0719V8MX1/ref=sr_1_4?crid=114Q7BC21NVS3&keywords=usb%2Bto%2Bmidi&qid=1673642815&sprefix=usb%2Bto%2Bm%2Caps%2C1127&sr=8-4&th=1) for PC DAW or PC Synth connectivity (optional).

**6.2 Setup**

Here are the basic steps required to setup the MidiXLate router.

**1:** Unpack the midi router and verify all components are present.

MidiXLate router, MidiXLate power supply, SD card, Midi USB A to Micro USB cable,

Sierra Digital Audio contact and registration form.

**2:** Identify all of your midi devices to be connected to the router.

Separate your devices into midi input devices and midi output devices.

For example, a midi keyboard is generally a midi output device (connected to an ‘in port’ on the router)

A midi tone generator, PC Synth, or PC DAW is typically an input devices (connected to an ‘out port’ on the router)

**3:** Define mnemonic names for all of your devices (such as KB for a keyboard, PCS for PC synth)

**4:** Connect your midi input devices to the midi router output ports and your midi output devices to midi router input ports. Keep track of which devices are connected to which ports.

**5:** Download and install the MidiXLate configuration software.

Go to [Sierra](http://www.xxx.com)DigitalAudio.com and download the MidiXLate configuration software

**6:** Launch the MidiXLate installer.

**7:** Configure Routings

**8:** Configure Translations

**9:** Configure Channel Locks

**10:** Save configuration to your PC hard disk.

**11:** Download your configuration to the MidiXLate router

**12:** Test your MidiXLate configuration

**7: Faceplate Overview.**



**7.1: Power button**

The power button will power on and power off the Midi Router. When the router is started, the router configuration is loaded from the MicroSD card. If the MicroSD card is not found, the router stops and displays an error message on the display. Assuming the router configuration is found and successfully loaded, the router then performs a self-test of the faceplate ‘Port In’ and ‘Port Out’ LEDs to verify all LEDs are functioning properly. After the LED self-test, a configuration summary is displayed in the display window. Lastly, the Logger button and Configure button are inspected to see if either button is pressed. The router is then started in either Run mode, [Logging mode](#B56), or [Configure mode](#B57).

**7.2: Logger button**

The Logger button on the face of the router and the LOG / CFG USB port on the rear of the router are used to troubleshoot (view) Midi traffic passing the router. See [Logging mode](#B56) discussed above.

**7.3: Configure button**

The Configure button on the face of the router and the LOG / CFG USB port on the rear of the router are used to download the router configuration from your PC to the router. See [Configure mode](#B57) discussed above.

**7.4: Midi Port In (led 1-6)**

When Midi traffic is detected on an input port, the corresponding port led will blink for two milli-seconds.

These led’s are just a visual notification of Midi port activity.

**7.5: Midi Port Out (led 1-6)**

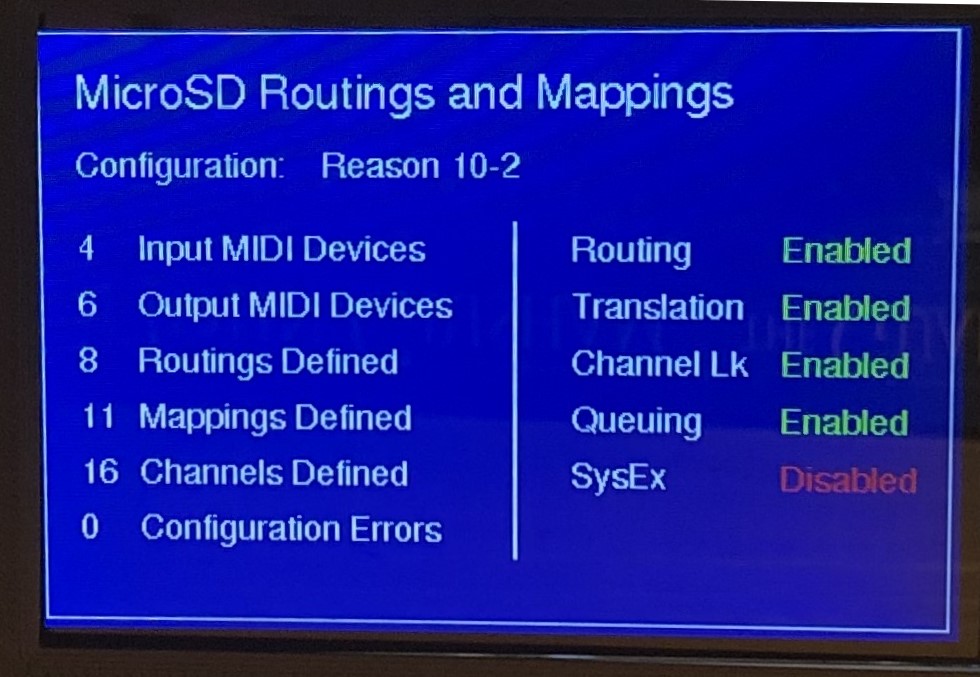
When Midi traffic is sent to an output port, the corresponding port led will blink for two milli-seconds.

These led’s are just a visual notification of Midi port activity.

**7.6: LCD Color Display**

When the router is started, a startup splash page with Sierra Digital Audio contact information and version information is displayed. The splash page is display for two seconds, after which the microSD card configuration information is displayed. The microSD card configuration information is displayed until Midi activity is detected on an input port. When Midi activity is detected, the last Midi command processed by the router and routing information is displayed. The display data is buffered and updates slowly. This is normal.

**7.6: LCD Color Display** (cont)



**8: Rearplate Overview**

****

**8.1: Cooling vents**

The four cooling vents on the back of the router provide ventilation to the MidiXLate router.

The router’s power draw is typically less than 10 watts, so ventilation is a minimal concern.

Never the less, the router should not be placed in hot or damp environments.

**8.2: Input-Output ports 1-6**

Input ports and output ports are labelled from left to right.

Connect your music devices to these Midi ports using DIN 5 Midi cables.

Typically, one device will use one set of ‘in port’ and ‘out port’.

For example, a keyboard can be connected to both the ‘in port’ and ‘out port’ to process two way KB traffic.

If your device requires only an ‘in port’ or an ‘out port’, it is possible to mix devices connected to a port.

For example, you can connect a keyboard to Midi In 1 and a tone generator to Midi Out 1.

The MidiXLate software provides individual labeling of input ports and output ports.

**8.3: LOG / CFG USB port**

The LOG / CFG USB port is a USB Micro B connector and connects the MidiXLate router to a host Windows PC.

This port is used by the ‘Logger’ function to write internal routing data to a host PC terminal emulator.

This log data is highly abstract and may be difficult to comprehend.

This log data will be invaluable for Sierra Digital Audio to assist customers when troubleshooting issues.

More information can be found in section [5.6: Logging Port](#B56)

This port is also used to ‘Configure’ the router.

Configuration information is defined on the host Windows PC and then downloaded to the MidiXLate router.

More information can be found in section [5.7: Configuration Port](#B57)

**Important Note: All USB and midi cable connections to the router must be made when the router power is OFF. Connecting cables with the power on risks damage to your music device, the MidiXLate router, and will cause midi connections to become unstable (stop functioning).**

**8: Rearplate Overview (cont)**

**8.4: NOT USED USB & PGM**

The two Micro B USB ports are used to program the Atmel SAM3X8E ARM Cortex-M3 CPU54 processor

These two ports are currently not used by the customer.

**8.5: PWR socket**

The 9V DC / 1000 milli-amp power adapter (2.1 mm center positive, 3.5 mm sleeve) included with router

**8.6: Micro SD slot**

The microSD card contains the router configuration which is maintained and downloaded from a host Windows PC. More information can be found in section [5.7: Configuration Port](#B57)

**9.1: MidiXLate configuration software – Download, Install, Launch**

To download and install the MidiXLate routing software:

**1:** Download the MidiXlate configuration software from the [Sierra Digital Audio](http://www.SierraDigitalAudio.com) website, downloads tab.

**2:** Microsoft applications (which MidiXLate is) require Microsoft’s .Net Framework to run.

If your Windows PC has a .Net Framework version older than .Net version 4.8,

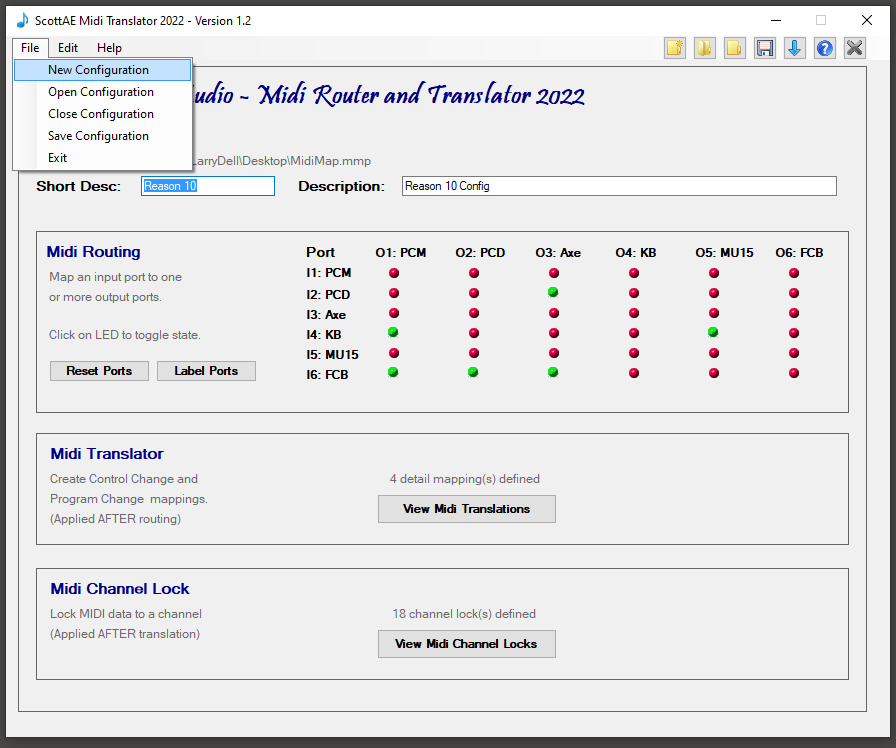
then download and install .[NetFramework Verison 4.8](https://dotnet.microsoft.com/en-us/download/dotnet-framework/net48) Runtime (or newer) from Microsoft.

**3**: Once the .Net Framework is installed, Double-click on the MidiXlate to start the install.

Accept all defaults. During install, choose to create a MidiXLate link on your desktop.

**4**: Launch the MidiXLate installer and create a configuration as described in the following sections

**Sample configuration screen**



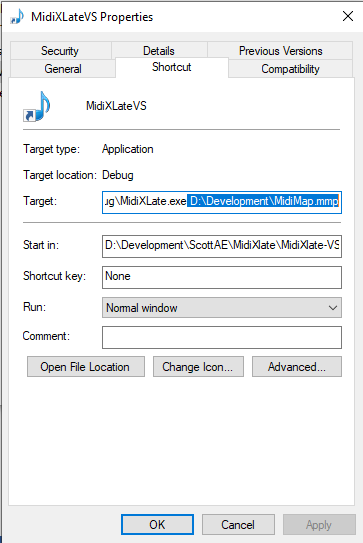
**9.1: MidiXLate configuration software – Download, Install, Launch (cont)**

This step is optional.

If you are working with only one configuration file and want to always load that configuration when MidiXLate is started, you can modify the MidiXLate desktop icon to launch a particular configuration.

On your desktop, right click on the MidiXLate icon and select ‘prroperties’.

A screen similar to this will appear.

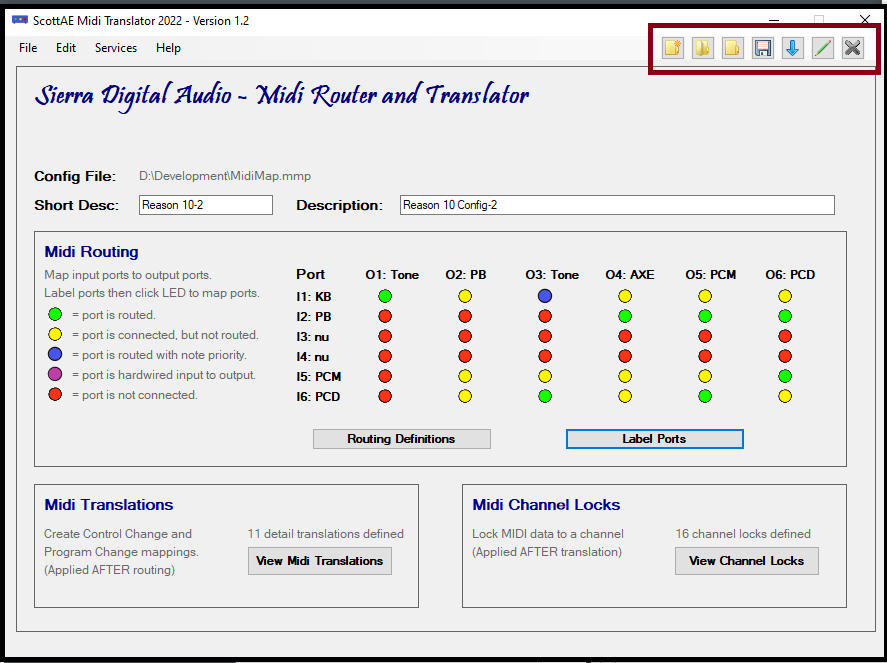


See the blue highlighted configuration file above.

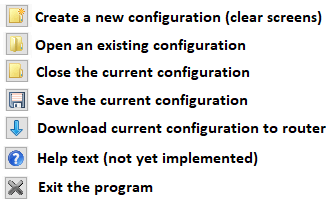
After MidiXLate.exe, add the location of the configuration file you want MidiXLate to automatically load when MidiXLate is launched. Now, MidiXLate will automatically load the configuration file when MidXLate is launched.

**9.2: MidiXLate configuration software - Navigation menus and buttons**

MidiXLate uses Windows standard menu options to create, open, close, and save a configuration to your PC. Edit, Copy Paste, and Delete is also implemented. Look and feel should be very familiar to Windows users.

****

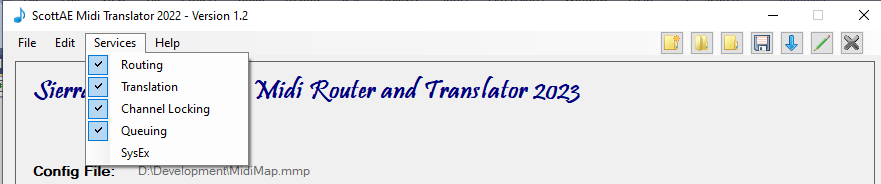
**Navigation speed buttons (with hover over hints) are also provided:**



**9.2: MidiXLate configuration software – Services**

The menu options shown below controls which services are enabled or disabled on the router.

You may choose to turn off one or all of these services, but you are still allowed to maintain the configuration tables for these services. The configuration tables may be fully populated even though the service is temporarily or permanently disabled.

****

**1: Routing**:

This feature enables or disables routing on the router. Currently, routing is enabled and can not be disabled.

This feature needs further definition and may be implemented in the future. When routing is disabled, any data received on input port X will be routed only output port X (where x = 1-6). This is envisioned to be a performance enhancement (if needed) for special circumstances.

**2: Translation**:

This feature enables or disables routing on the router.

If translation is disabled, no midi data will be translated on the router.

**3: Channel Locking**:

This feature enables or disables routing on the router.

If translation is disabled, no midi data will be translated on the router.

**4: Queuing**:

This feature enables or disables queuing on the router.

If queuing is disabled, no midi data will be queuing on the router.

Currently, queuing is only used for change control translation to note on / off processing

The midi note on / off event is queued to accommodate the delay time between note on and note off.

Queuing is not heavily used in MidiXLate’s current release but is expected to be used more in the future.

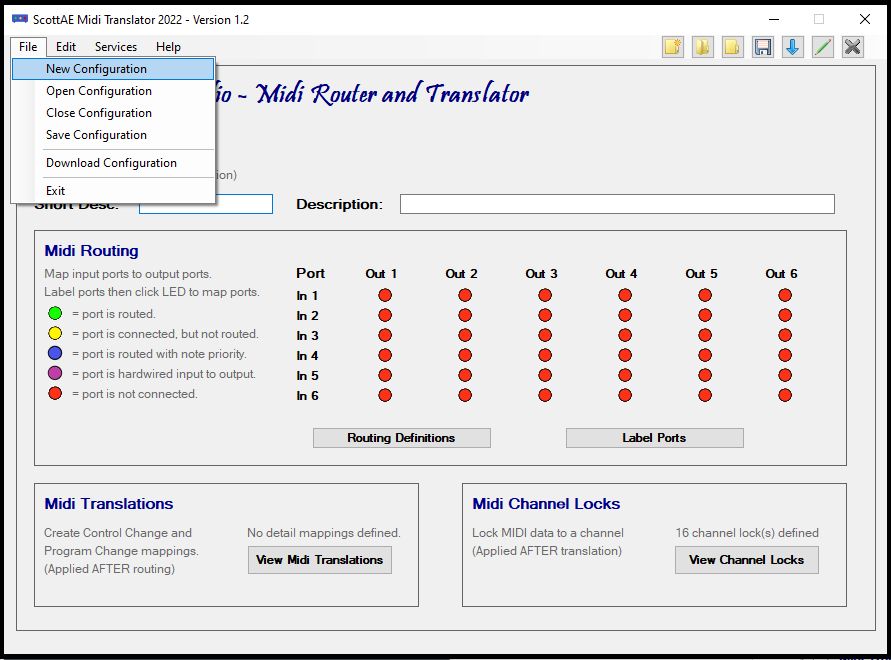
**5: SysEx:**

This switch tells the router how to handle SysEx (System Exclusive) commands. SysEx commands are typically midi vendor specific commands used to control features of a vendor’s device that can not be controlled via the general midi protocol. When the SysEx switch is enabled, SysEx commands are processed. When SysEx switch is disabled, SysEx commands are ignored. By default, SysEx processing is disabled.

**9.3: MidiXLate configuration software - Creating a New Configuration**

To create a new MidiXLate router configuration, select the ‘New Configuration’ option as shown below.

All screen data will be cleared and the default routing will be set as shown blow

****

Create one or more router configuration(s) as described in the screens below.

Save the router configuration(s) to your Windows PC and download to the MidiXLate router as needed.

**9.4: MidiXLate configuration software - Midi Routing**

[Midi Routing](#B54) specifies how music devices are connected and communicate with each other.

You do this by specifying which midi IN ports are connected to which midi OUT ports .

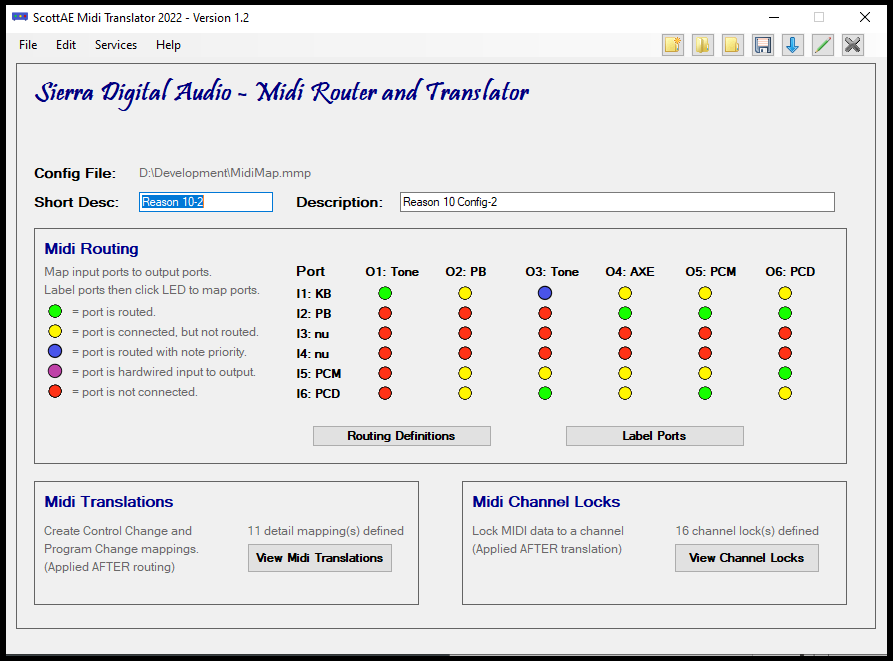
There are six input ports and six output ports.

A new Midi configuration is shown in the prior screen. A sample of my Midi Routing is shown below.

Notice the ‘Midi Routing’ section below.

The first step to define your routing is to define the music devices connected to the rear panel midi ports.

Press the ‘Label Ports’ button and define your midi port connection (as described on the next page)

****

Notice the vertical (I1-I6) input ports and horizontal (O1-O6) output ports (defined on the next page).

To connect an input midi device to an output midi device, click on the red led to turn it green (connected).

To disconnect an input midi device from an output midi device, click on the green led to turn it red (disconnected).

**9.4: MidiXLate configuration software - Midi Routing**

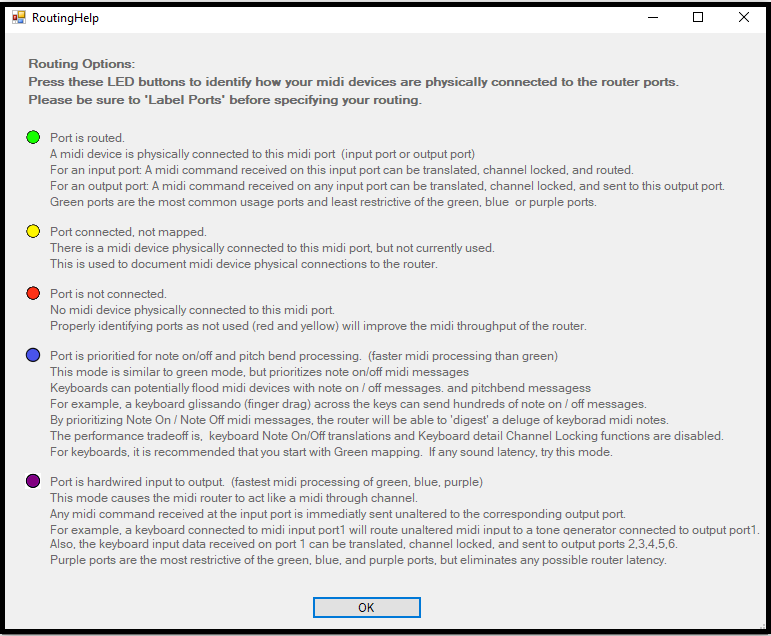
Notice one input device can be ‘routed’ to multiple output devices.

For example, above, my (I6) FCB 1010 pedalboard routes midi data to my (O1) Music PC, (O2) Development PC, and (O3) Axe Guitar FX processor.

Excessive midi routing (green leds) should be avoided.

The router can handle excessive routing, but operates faster when few routes are configured.

Midi Routing is the only required configuration. Translation and Channel Locks are optional



**9.4: MidiXLate configuration software - Midi Routing (cont)**

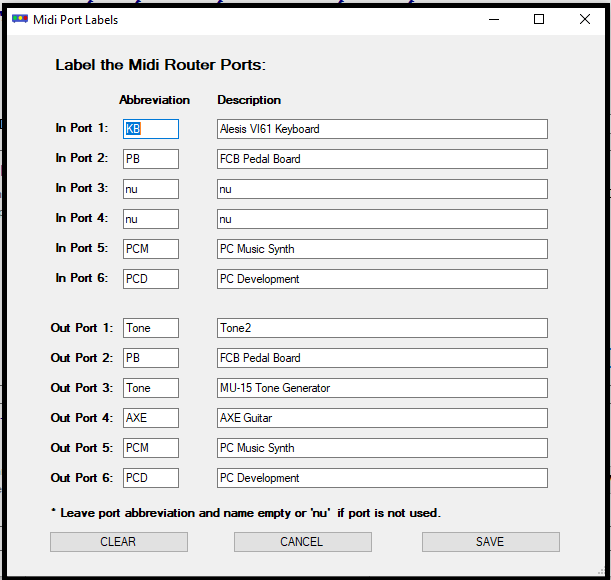
Define Midi devices attached to ports 1-6.

When the ‘Label Ports’ button on the prior screen, the following screen is displayed.

Provide both an abbreviated description (4 char max) and a full description for each midi device.

Abbreviated descriptions are used by the router when midi transactions are displayed

If you do not use a particular port, leave the description empty



Press the SAVE button to save your work.

**9.5: MidiXLate configuration software - Midi Translation**

[Section 5.4 Midi Translation](#B55) provides a summary of Midi Translation.

On the MidiXLate opening page, press button ‘View Midi Translations’ to view the following screen.

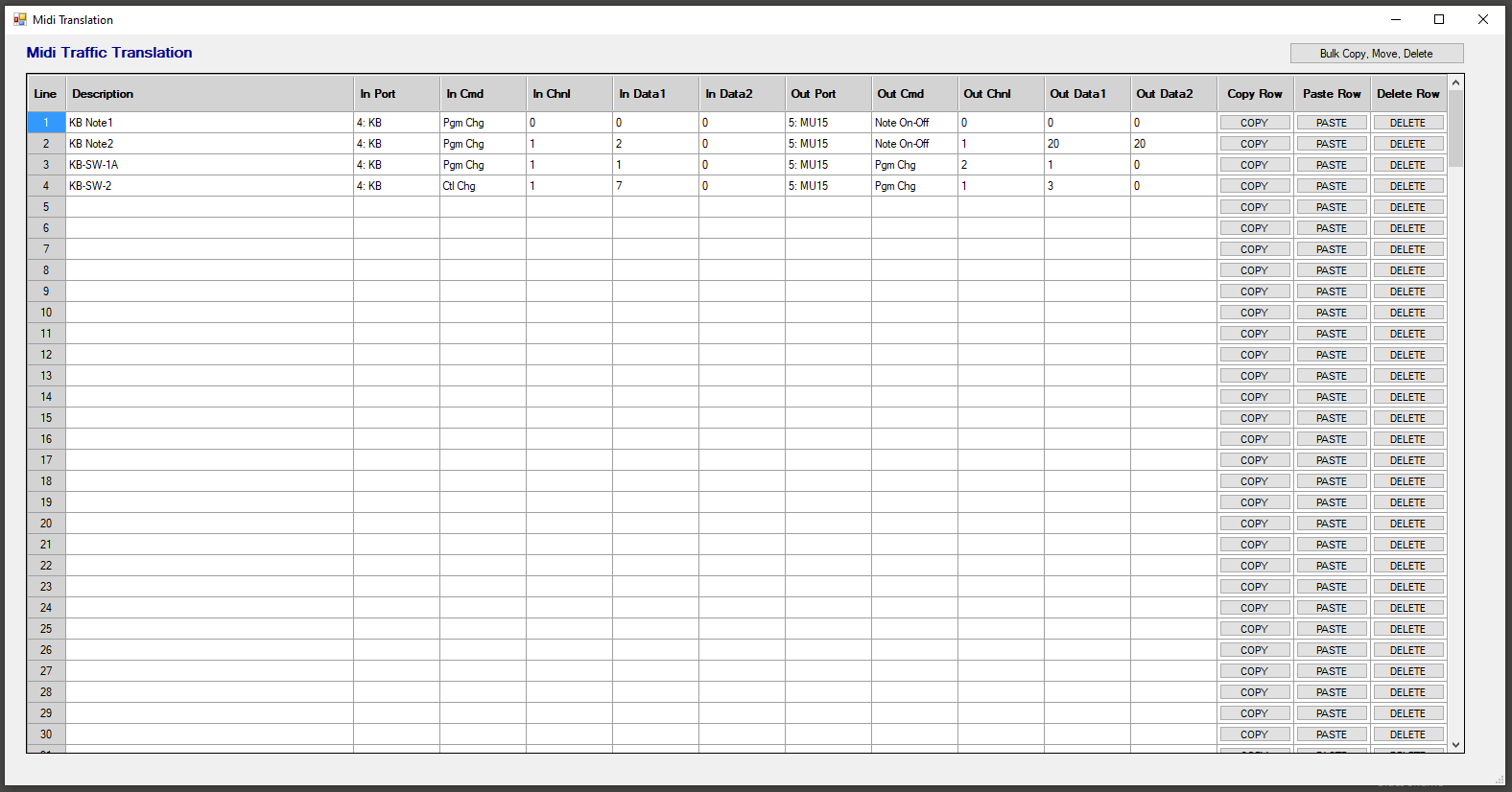
This screen is a summary of the midi translations defined in MidiXLate.

Currently, 256 translations (maximum) can be defined.

Translations (lines) can be copied and deleted on this screen.

Double click on a given row to view / edit the channel lock detail (described on the next page).

Alternatively, press the ‘[Bulk Copy, Move, Delete](#B95)’ button to perform bulk maintenance.



Apologies for the small screen print, but it was necessary to show the entire screen.

**Translation occurs when a received midi command matches an In Port, In Command, In Channel, In Data1.**

Notice on line 2, a translation is defined from my keyboard (in port 4) to a MU15 tone generator (out port 5).

In this example, when I receive a ‘Program Change’ (PC) midi message from my KB, port 4, channel 1, patch 2, the midi message is ‘translated’ to a Note On-Off command, sent to my MU15 tone generator on port 5, channel 1, note 20, with velocity 20.

You have the option to define multiple translations for the same input midi message by defining two lines with the same In Port, In Cmd, In Chnl, In Data 1, with different Out Port, Out Cmd, Out Chnl, Out Data 1.

Using the example above, if I receive a ‘Program Change’ (PC) midi message from my KB, port 4, channel 1, patch 2, the midi message can be ‘translated’ to a Note On-Off command, sent to my MU15 tone generator on port 5 and a Reason 10 rack device on port 1.

Currently, out Note On / Note Off commands have a fixed delay of 50 milliseconds between Note On and Off

**9.5: MidiXLate configuration software - Midi Translation (cont)**

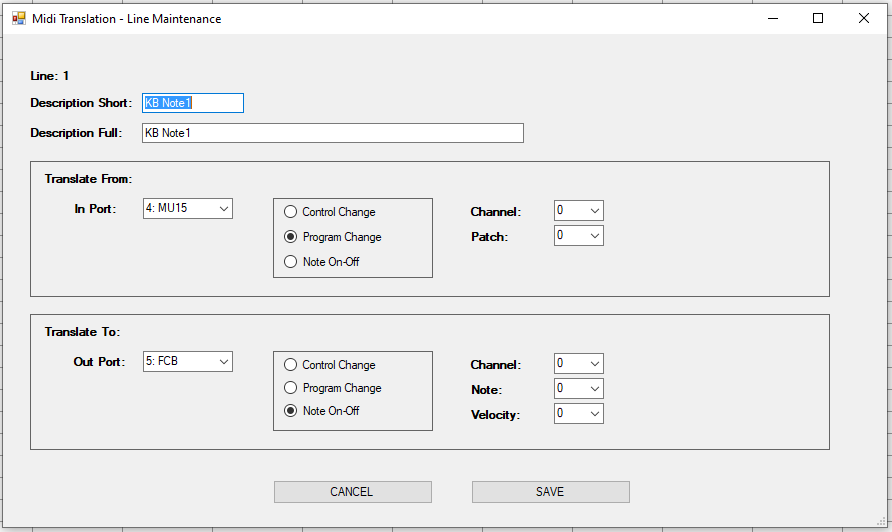
Midi translation was summarized on the prior page.

When a line is ‘double clicked’ on the prior screen, the following maintenance screen is displayed.

Here, you define a Midi Translation.

Description Short is the description displayed in the router’s LCD when this translation is encountered.

Description Full is a full description of the translation only used for documentation.



**Translation occurs when a received midi command matches an In Port, In Command, In Channel, In Data1.**

The ‘Translate From’ and ‘Translate To’ panels above describe the data received (in port), how it is translated, and how it is routed (out port).

Notice how the Channel / Patch and Channel / Note / Velocity parameters change as the radio buttons (Control Change, Program Change, Note On-Off) are changed.

This screen relies on the reader’s knowledge of Midi commands and message formats.

For a basic overview Midi Message Types see [Appendix B](#AppendixB)

For a detailed description of Midi Message Formats see [Appendix C4](#C4)

Armed with a complete understanding of Midi Messages Types and Midi Message Formats, the screen above will make sense.

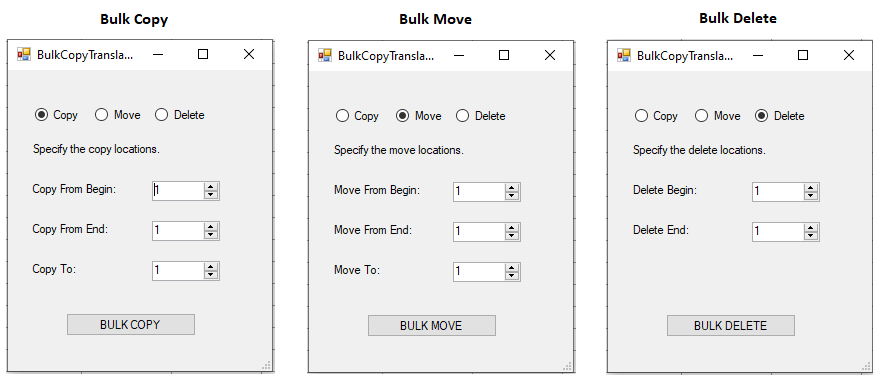
**9.5: MidiXLate configuration software - Midi Translation (cont)**

The bulk copy, move, delete function provides an easy method to copy, move, or delete many lines in one simple operation. This functionality will be useful when cloning translation configurations.

Press the Bulk Copy, Move, Delete button on the Midi Translation screen to display the screens below.

The Bulk Copy, Bulk Move, and Bulk Delete screens are shown below.

First, specify if you want to perform a bulk copy, move, delete.



Next, specify the Copy Begin / Copy End locations and then specify where you want to Copy To.

Press the BULK COPY button to begin the copy operation.

Close the screen to view the updates in the Midi Translation table

**9.5: MidiXLate configuration software - Midi Translation Notes**

\*\* Consider the scenario where midi data is received on an input port A and routed to output ports X, Y, and Z:

If no translation exists for the midi data received on port A then the midi data is routed to output ports X, Y, and Z. If a translation is defined for port X, the midi data received on port A is translated and sent to port X. The original midi data received on port A is NOT sent to output ports Y and Z. Two additional mappings are required if you want to send the original data to output ports Y and Z. This may seem counter-intuitive, but this facilitates sending midi data, such as from a keyboard to multiple devices, such as a DAW, synth, and tone generator, where 3 translations are required.

\*\* Every Midi command received at the router can have a maximum of 25 translations.

For example, if the router receives a ‘Transport Play’ command from a keyboard, it can be translated and sent to a maximum or 25 devices (DAW, Synth, Drum Machine, Tone Generator, ….). 25 is a limitation that should never be reached.

**9.6: MidiXLate configuration software - Midi Channel Locking**

[Section 5.5 Channel Locking](#B55) provides a summary of Midi Channel Locking.

On the MidiXLate opening page, press button ‘View Midi Channel Locks’ to view the following screen.

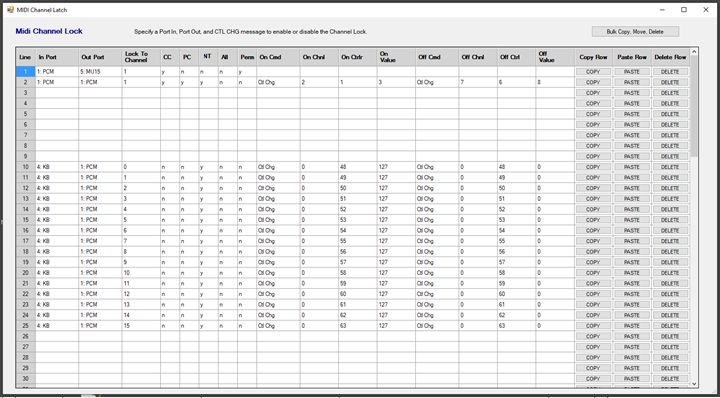
This screen is a summary of the channel locks defined in MidiXLate.

Currently, 96 channel locks (maximum) can be defined.

Channel locks (lines) can be copied and deleted on this screen.

Double click on a given row to view / edit the channel lock detail (described on the next page).

Alternatively, press the ‘[Bulk Copy, Move, Delete](#B95)’ button to perform bulk maintenance.



Apologies for the small screen print, but it was necessary to show the entire screen.

Notice on line 10, a channel lock is defined from my keyboard (in port 4) to Reason PC (out port 1).

The channel lock is set when port 4 receives a Ctl Chg command, on channel 0, controller 48, and value 127.

The channel lock is removed when port 4 receives a Ctl Chg cmd, on channel 0, controller 48, and value 0.

When the channel lock is set and a keyboard note (NT column=Y), port 4 outbound midi will be on channel 0.

Most Midi keyboards and control surfaces provide many assignable knobs, sliders, and buttons (controllers).

My inexpensive Alesis keyboard has 16 knobs and 48 buttons.

In lines 10-25 above, I assigned 16 buttons on my keyboard to toggle between 16 Reason 10 rack devices.

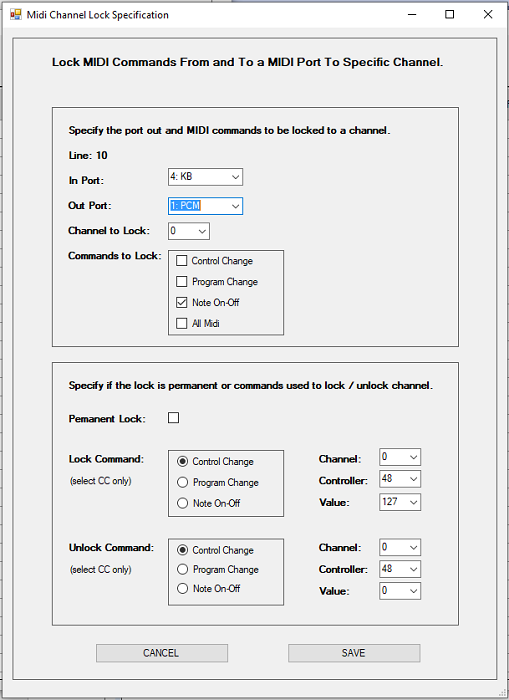
Thus, any keyboard note I press is connected to / played by the selected Reason 10 device.

**9.6: MidiXLate configuration software - Midi Channel Locking (cont)**

This midi channel lock was summarized on the prior page.

When a line is ‘double clicked’ on the prior screen, the following maintenance screen is displayed.

Here, you define a Midi Channel Lock.



In this example, a channel lock is defined from my keyboard (in port 4) to my Reason PC (out port 1).

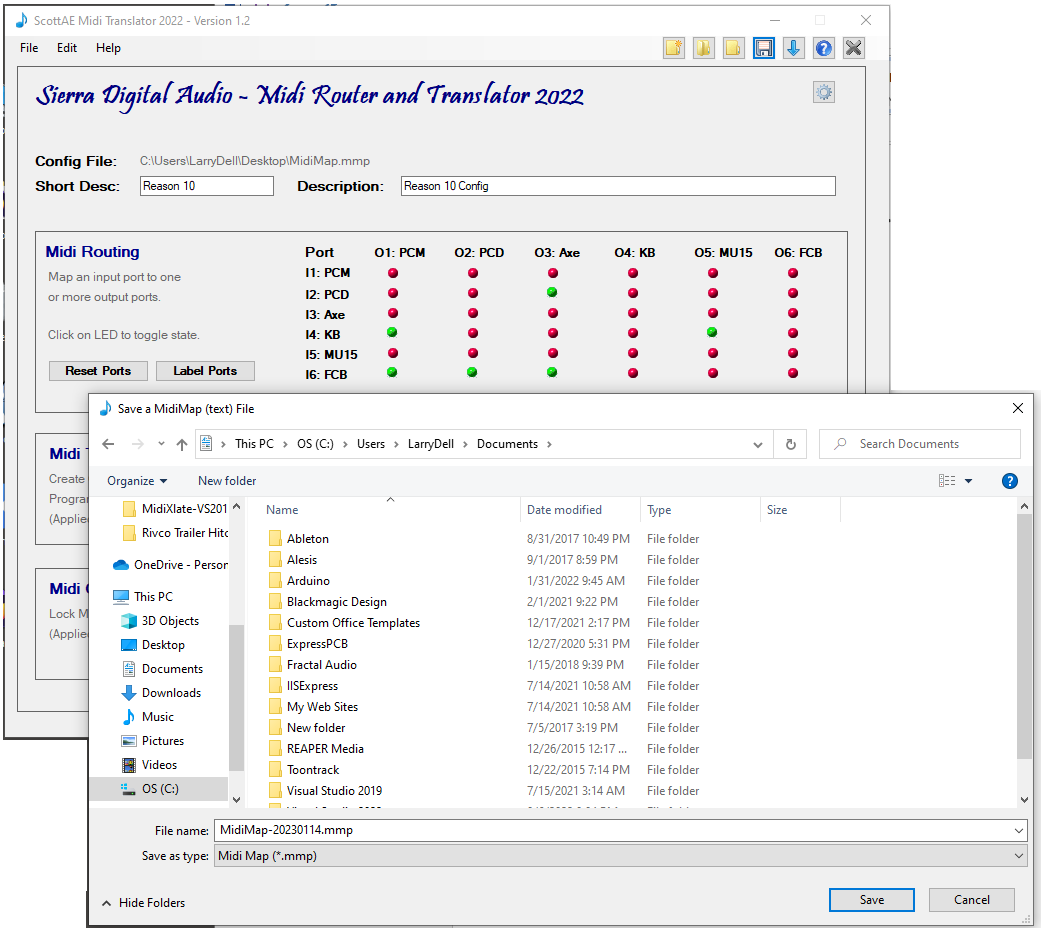
The channel lock is set when port 4 receives a Control Chg command, on channel 0, controller 48, value 127.

The channel lock is removed when port 4 receives a Control Chg cmd, on channel 0, controller 48, value 0.

When the channel lock is set and a keyboard key is pressed (NT column=Y), port 4 outbound midi data will be sent / locked to midi channel 0.

**9.7: MidiXLate configuration software - Saving a configuration to your PC**

Opening a MidiXLate router configuration and saving a MidiXLate router configuration use the standard Windows open dialog box and save dialog box. The look and feel will be familiar to Windows users.



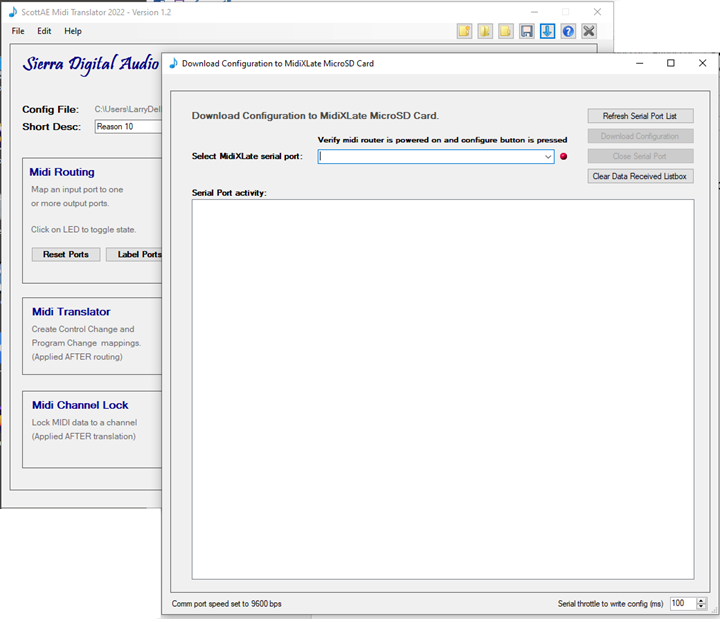
In the save dialog box shown above, notice the default file name extension (.mmp). This identifies a file as a MidiXLate ‘midi map’ file format. Also notice the file name has the current year, month, day appended to the default file name for ‘versioning’ of configurations. Change the file name as you required.

**9.8: MidiXLate configuration software - Download configuration to MidiXLate router**

The following screen is used to download the configuration you created on the PC to the MidiXLate router.

This process will download the router configuration currently opened (identified on the home screen).

Before starting the download, please review the download prerequisites in section [5.7: Configuration Port](#B57).



Once all the prerequisites have been handled, follow the instructions on the next page to download your configuration.

**9.8: MidiXLate configuration software - Download configuration to MidiXLate router (cont)**

**1: Establish the PC connection to the MidiXLate router**

* Select the MidiXLate serial port.
* Press the down arrow on the combobox to see a list of active serial ports. Select a serial port.
* When you select a serial port the PC will send a {HSK} handshake command to the router to verify the router is connected to the serial port. The router will respond to the {HSK} request with an {ACK} response letting the PC know the router is connected and ready to receive the download.
* When the PC to MidiXLate router ‘handshake’ is successful, the red ‘connection’ led will turn green and the ‘Download Configuration’ button and ‘Close Serial Port’ button will be enabled.
* You are now ready to download the configuration.

**2: Download the configuration.**

* Press the ‘Download Configuration’ button.
* The currently opened PC configuration file will slowly start downloading to your router. Download statistics will display on your screen as the download progresses. Additionally, the contents of the currently opened configuration will be written to ‘Serial Port activity’ listbox. When the download is complete, the statistics counter will stop and an {EDL} message will appear at the end of the listbox.

**3: Return to ‘Run Mode’**

* Close the PC download page and optionally close the MidiXLate router configuration program.
* Depress the ‘Configure’ button on the routers faceplate. The router configuration will be reloaded from the router’s MicroSD card.
* The router is back in ‘run mode’ and ready to resume processing of midi data.

**10: MidiXLate Specifications**

MidiXLate Router V3 specifications as of March 2023.

|  |  |
| --- | --- |
| **MidiXLate Router Specifications** | |
| MIDI Interface | Six SPI-MIDI input ports and six SPI-MIDI output ports (type 5-pin DIN) |
| Display | LCD 200 x 500 pixel color display |
| Processor | Atmel SAM3X8E ARM Cortex-M3 CPU54 processor running at 84 MHz clock |
| MicroSD | 256 Meg Micro SD card |
| Power Supply | 9V DC / 1000 milli-amp power adapter, 2.1 mm center positive, 3.5 mm sleeve |
| Power consumption | 10 watts |
| Dimensions (H x W x D) | **4.5 in x 14 in x 6 in** |
| Weight | **2.25 lbs / 1kilogram** |

Sierra Digital Audio is constantly striving to continually improve the MidiXLate router.

As a result, modifications may be made to the router without prior notice.

Specifications and appearance may differ from those listed.

For the latest router information, please see our website at **www.scottae.github.com**

**11: Technical Notes**

**Midi Translation:**

MidiXLate currently (only) supports translation of Control Change (CC), Program Change (PC), and Note On/Off Midi commands. Based on demand, the remaining Midi message types may be added in the future.

The translation you create will only affect the in port and out port you specify. If your routing definition sends a midi message to multiple ports, the translation will only occur for the in port and out port you defined in the translation. The remaining ports will receive the midi message without the translation. If desired, you can setup multiple translations when routing midi messages to multiple ports.

**Midi Channel Locking:**

The channel lock you create will only lock the channel for the in port and out port you specify. If your routing definition sends a midi message to multiple ports, the channel lock will only occur for the in port and out port you defined in the channel lock. The remaining ports will receive the midi message without the channel lock. If desired, you can setup multiple channel locks when routing midi messages to multiple ports.

**Cables:**

All cable lengths should be kept as short as possible.

USB cables can be up to 6 feet, but recommend using cables no longer than 3 feet.

Midi cables can range up to 50 feet, but recommend keeping your midi cables under 36 feet in length

**MicroSD card:**

You can remove the MicroSD card from the router, insert it into your PC’s MicroSD reader, and manually manipulate and maintain the router’s configuration. To remove the SD card, very gently push the SD card further into the router. A spring mechanism will pop it outwards. Put the SD card in your [PC’s micro SD adapter](https://www.amazon.com/uni-Adapter-Supports-Compatible-MacBook/dp/B081VHSB2V/ref=sr_1_1_sspa?crid=3MUKNCPN7K3JI&keywords=micro+sd+adapter&qid=1673664193&sprefix=%2Caps%2C12070&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzS080N1BTMTZYT1RaJmVuY3J5cHRlZElkPUEwMDUwMzc2MUdWQkI0S1VTNk1IUSZlbmNyeXB0ZWRBZElkPUEwNTU1MzAzM0VJRkMzRU1MU0JXMyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=) and edit the MidiMap.mmp file with Notepad or any other text file editor. Be sure to backup the MidiMap.mmp file before you manually change it!

**Latency:**

Midi network latency is the amount of time it takes for a midi message to go from one device to another.

Lowering midi latency is an important part of building a good midi network.

Lower midi latency by minimizing excess routing (ie: routing midi packets to devices where not needed)

For example, pressing a note on a keyboard and sending it to an unused pedal board is not good routing.

The router’s Atmel SAM3X8E ARM Cortex-M3 CPU54 processor running at 84 MHz is blazing fast, but regardless of the CPU speed, wasting CPU cycles with poorly defined routing is not ideal.

**Active Sensing:**

Active Sense is another type of MIDI message. It's used to implement a "safety feature" that will cause stuck notes and other undesirable effects to be resolved whenever the MIDI connection between devices is broken or impeded in some way. Active Sensing can be a helpful tool, but active sensing can degrade Midi performance by flooding your network with Active Sensing messages. This extraneous network traffic can cause latency in the network and router. In an ideal Midi network, a device will send Active Sensing messages every few seconds, not 100 times per second.

**11: Technical Notes (cont)**

**System Exclusive Messages**

System Exclusive messages pass thru the MidiXLate router unchanged.

System Exclusive data is read and written one byte at a time.

System Exclusive messages are routed, but cannot be translated and cannot be channel locked.

For large sysex files, please throttle your transmit speed to not overrun the MidiXLate receive buffers.

(the receiving device also prefers to receive sysex data at a slower rate).

**Unrecognized data**

If MidiXLate receives data that does not match any midi format, the data will be logged and disregarded.

**Router Configuration File**

The xxxxxxx.mmp router configuration file is a standard format text file which and can be manually manipulated with a Windows (not Unix) text editor. Manual manipulation is not recommended as the file’s structure and delimiters are very specific.

**Important Note: All USB and midi cable connections to the router must be made when the router power is OFF. Connecting cables with the power on risks damage to your music device, the MidiXLate router, and will cause midi connections to become unstable (stop functioning).**

**Appendix A**:

MidiXLate SD Card: Configuration File Format

**Appendix B**

MidiXLate Logger Format

**Appendix C**:

On the Sierra Digital Audio website you will find the following five instructional videos:

1: [MidiXLate Router Overview](http://www.SierraDigitalAudio.com)

2: [MidiXLate Software Installation](http://www.SierraDigitalAudio.com)

3: [MidiXLate Midi Routing](http://www.SierraDigitalAudio.com)

3: [MidiXLate Midi Translation](http://www.SierraDigitalAudio.com)

4: [MidiXLate Midi Channel Locking](http://www.SierraDigitalAudio.com)

5: MidiXLate Software Download (Windows 10 and above)

6: [.NetFramework Verison 4.8](https://dotnet.microsoft.com/en-us/download/dotnet-framework/net48)

**Appendix D**: **Midi Concepts from Roland**

A Midi message type introduction from [Roland](http://cdn.roland.com/assets/media/pdf/MIDIBasics.pdf)

**The Language of MIDI**  
Like all digital communication schemes, MIDI is made of up of discrete messages, each performing a specific  
function. The following is a brief discussion of the primary MIDI messages. Please note that, while these  
messages are very common, not all MIDI devices will have the ability to transmit and/or receive each message.  
You can refer to the MIDI Implementation chart for a specific MIDI device to determine which messages it is  
capable of sending and/or receiving.

**A. Note Messages**  
Note messages are the most basic type of MIDI messages a keyboard can send. Note On/Off and Velocity  
messages are the most common type of Note message.

**1. Note On/Note Off**When a key is pressed on a MIDI keyboard, a Note On message is sent along with a pitch value telling the  
receiving MIDI device to play a note at a certain pitch. Releasing the key sends a Note Off message,  
which tells the receiving device to stop playing the note.

**2. Velocity**If the transmitting keyboard is touch sensitive, a Velocity message will be sent along with the Note On  
message. This message corresponds with how hard the key was played and tells the receiving device how  
loud the sound should be.

**B. Program Change**A Program Change message tells the receiving device to change from one sound to another. This is useful for  
automatically changing sounds during playback of a sequence or changing the sounds of several sound modules  
from one keyboard during live performance. There are 128 Program Change numbers in MIDI. To access more  
than 128 sounds through MIDI, you need to use Bank Select messages (see Controller Messages).

**C. Controller Messages**  
There are a number of MIDI Messages classified as Continuous Controller (CC) messages. These messages are usually  
used to add subtle nuances to your playing. Some of the more common controller messages include the following:

**1. Modulation**Modulation is a type of MIDI message known as a Control Change (CC) message. Like Pitch Bend, Modulation is  
usually sent by moving a lever or wheel on the keyboard. Although Modulation is usually used to control vibrato, it  
can usually be assigned to other parameters such as pitch and volume. In MIDI, Modulation is assigned to Control  
Change #1.

**2. Volume**  
Like Modulation, Volume is a Control Change message. It can usually be assigned to control various  
parameters, but typically is used to control the volume level of the receiving device. In MIDI, Volume  
messages are assigned to Control Change #7.

**3. Bank Select**  
There are only 128 program change messages in MIDI. In order to access sounds beyond 128 via MIDI,  
manufacturers group sounds into banks. Controller messages are used to select different groups (banks)  
of sounds. These messages are referred to as Bank Select messages and are normally followed by a  
Program Change message. In MIDI, Bank Select messages are assigned to Control Change #0 and #32.

**Appendix D**: **Midi Concepts from Roland (cont)**

**D. Pitch Bend**  
Pitch Bend messages tell the receiving device to change the pitch of a sound. These messages are usually sent  
by moving a lever or wheel on the left side of a MIDI keyboard.

**E. After-Touch**  
After-Touch is a MIDI message transmitted when pressure is applied to a key after it is initially pressed. The  
receiving device can usually be configured to have these messages control things such as vibrato, brilliance,  
volume, etc. There are two types of After-Touch messages: Polyphonic After-Touch and Channel After-Touch.  
Polyphonic After-Touch affects only the note that is played. Channel After-Touch affects every note on a specific  
MIDI Channel.

**F. System Exclusive**  
System Exclusive messages are exclusive to a particular manufacturer. These messages can be used to perform  
specific functions such as saving or editing patches on a computer. Generally, one MIDI device will not be able to  
read another MIDI devices System Exclusive messages unless they are the same model

**Appendix E**: **MIDI McGill**

Section 16 is all reprinted from [McGill Music Department](http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html#BMA1_1)

**E1: Midi Introduction**

This document details the structure of MIDI Files. The purpose of MIDI Files is to provide a way of interchanging time-stamped MIDI data between different programs on the same or different computers. One of the primary design goals is compact representation, which makes it very appropriate for disk-based file format, but which might make it inappropriate for storing in memory for quick access by a sequencer program.

MIDI Files contain one or more MIDI streams, with time information for each event. Song, sequence, and track structures, tempo and time signature information, are all supported. Track names and other descriptive information may be stored with the MIDI data. This format supports multiple tracks and multiple sequences so that if the user of a program which supports multiple tracks intends to move a file to another one, this format can allow that to happen.

The specification defines the 8-bit binary data stream used in the file. The data can be stored in a binary file, nibbilized, 7-bit-ized for efficient MIDI transmission, converted to Hex ASCII, or translated symbolically to a printable text file. This spec addresses what's in the 8-bit stream. It does not address how a MIDI File will be transmitted over MIDI.

**E2: Midi Messages**

A MIDI message is made up of an eight-bit status byte which is generally followed by one or two data bytes. There are a number of different types of MIDI messages. At the highest level, MIDI messages are classified as being either Channel Messages or System Messages. Channel messages are those which apply to a specific Channel, and the Channel number is included in the status byte for these messages. System messages are not Channel specific, and no Channel number is indicated in their status bytes.

Channel Messages may be further classified as being either Channel Voice Messages, or Mode Messages. Channel Voice Messages carry musical performance data, and these messages comprise most of the traffic in a typical MIDI data stream. Channel Mode messages affect the way a receiving instrument will respond to the Channel Voice messages.

MIDI System Messages are classified as being System Common Messages, System Real Time Messages, or System Exclusive Messages. System Common messages are intended for all receivers in the system. System Real Time messages are used for synchronization between clock-based MIDI components. System Exclusive messages include a Manufacturer's Identification (ID) code, and are used to transfer any number of data bytes in a format specified by the referenced manufacturer.

### E3: Midi Message Format

### A MIDI message is made up of an eight-bit status byte which is generally followed by one or two data bytes. There are a number of different types of MIDI messages. At the highest level, MIDI messages are classified as being either Channel Messages or System Messages. Channel messages are those which apply to a specific Channel, and the Channel number is included in the status byte for these messages. System messages are not Channel specific, and no Channel number is indicated in their status bytes.

Channel Messages may be further classified as being either Channel Voice Messages, or Mode Messages. Channel Voice Messages carry musical performance data, and these messages comprise most of the traffic in a typical MIDI data stream. Channel Mode messages affect the way a receiving instrument will respond to the Channel Voice messages.

MIDI System Messages are classified as being System Common Messages, System Real Time Messages, or System Exclusive Messages. System Common messages are intended for all receivers in the system. System Real Time messages are used for synchronization between clock-based MIDI components. System Exclusive messages include a Manufacturer's Identification (ID) code, and are used to transfer any number of data bytes in a format specified by the referenced manufacturer.

### E4: Table of Major MIDI Messages

|  |  |  |
| --- | --- | --- |
| **Channel Messages** | | |
| **Status D7----D0 nnnn is the MIDI channel no.** | **Data Byte(s) D7----D0** | **Description** |
| 1000nnnn | 0kkkkkkk 0vvvvvvv | Note Off event. This message is sent when a note is released (ended). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity. |
| 1001nnnn | 0kkkkkkk 0vvvvvvv | Note On event. This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity. |
| 1010nnnn | 0kkkkkkk 0vvvvvvv | Polyphonic Key Pressure (Aftertouch). This message is most often sent by pressing down on the key after it "bottoms out". (kkkkkkk) is the key (note) number. (vvvvvvv) is the pressure value. |
| 1011nnnn | 0ccccccc 0vvvvvvv | Control Change. This message is sent when a controller value changes. Controllers include devices such as pedals and levers. Certain controller numbers are reserved for specific purposes. See Channel Mode Messages. (ccccccc) is the controller number. (vvvvvvv) is the new value. |
| 1100nnnn | 0ppppppp | Program Change. This message sent when the patch number changes. (ppppppp) is the new program number. |
| 1101nnnn | 0vvvvvvv | Channel Pressure (After-touch). This message is most often sent by pressing down on the key after it "bottoms out". This message is different from polyphonic after-touch. Use this message to send the single greatest pressure value (of all the current depressed keys). (vvvvvvv) is the pressure value. |
| 1110nnnn | 0lllllll 0mmmmmmm | Pitch Wheel Change. This message is sent to indicate a change in the pitch wheel. The pitch wheel is measured by a fourteen bit value. Centre (no pitch change) is 2000H. Sensitivity is a function of the transmitter. (lllllll) are the least significant 7 bits. (mmmmmmm) are the most significant 7 bits. |

### E4: Table of Major MIDI Messages (cont)

|  |  |  |
| --- | --- | --- |
| **Channel Mode Messages (See also Control Change, above)** | | |
| 1011nnnn | 0ccccccc 0vvvvvvv | Channel Mode Messages. This the same code as the Control Change (above), but implements Mode control by using reserved controller numbers. The numbers are: Local Control. When Local Control is Off, all devices on a given channel will respond only to data received over MIDI. Played data, etc. will be ignored. Local Control On restores the functions of the normal controllers. c = 122, v = 0: Local Control Off c = 122, v = 127: Local Control On  All Notes Off. When an All Notes Off is received all oscillators will turn off. c = 123, v = 0: All Notes Off c = 124, v = 0: Omni Mode Off c = 125, v = 0: Omni Mode On c = 126, v = M: Mono Mode On (Poly Off) where M is the number of channels (Omni Off) or 0 (Omni On) c = 127, v = 0: Poly Mode On (Mono Off) (Note: These four messages also cause All Notes Off) |

|  |  |  |  |
| --- | --- | --- | --- |
| **System Common Messages** | | | |
| 11110000 | 0iiiiiii 0ddddddd .. .. 0ddddddd 11110111 | | System Exclusive. This message makes up for all that MIDI doesn't support. (iiiiiii) is usually a seven-bit Manufacturer's I.D. code. If the synthesiser recognises the I.D. code as its own, it will listen to the rest of the message (ddddddd). Otherwise, the message will be ignored. System Exclusive is used to send bulk dumps such as patch parameters and other non-spec data. (Note: Real-Time messages ONLY may be interleaved with a System Exclusive.) This message also is used for extensions called Universal Exclusive Messages. |
| 11110001 |  | | Undefined. |
| 11110010 | 0lllllll 0mmmmmmm | | Song Position Pointer. This is an internal 14 bit register that holds the number of MIDI beats (1 beat= six MIDI clocks) since the start of the song. l is the LSB, m the MSB. |
| 11110011 | 0sssssss | | Song Select. The Song Select specifies which sequence or song is to be played. |
| 11110100 |  | | Undefined. |
| 11110101 |  | | Undefined. |
| 11110110 |  | Tune Request. Upon receiving a Tune Request, all analog synthesisers should tune their oscillators. | |
| 11110111 |  | End of Exclusive. Used to terminate a System Exclusive dump (see above). | |

### E4: Table of Major MIDI Messages (cont)

|  |  |  |
| --- | --- | --- |
| **System Real-Time Messages** | | |
| 11111000 | Timing Clock. Sent 24 times per quarter note when synchronisation is required. |
| 11111001 | Undefined. |
| 11111010 | Start. Start the current sequence playing. (This message will be followed with Timing Clocks). |
| 11111011 | Continue. Continue at the point the sequence was Stopped. |
| 11111100 | Stop. Stop the current sequence. |
| 11111101 | Undefined. |
| 11111110 | Active Sensing. Use of this message is optional. When initially sent, the receiver will expect to receive another Active Sensing message each 300ms (max), or it will be assume that the connection has been terminated. At termination, the receiver will turn off all voices and return to normal (non-active sensing) operation. |
| 11111111 | Reset. Reset all receivers in the system to power-up status. This should be used sparingly, preferably under manual control. In particular, it should not be sent on power-up. In a MIDI file this is used as an escape to introduce <meta events>. |

### E5: Table of MIDI Controller Messages (Data Bytes)

The following table lists the MIDI Controller messages in numerical (binary) order.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **2nd Byte Value** | | | **Function** | **3rd Byte** | |
| **Binary** | **Hex** | **Dec** |  | **Value** | **Use** |
| 00000000 | 00 | 0 | Bank Select | 0-127 | MSB |
| 00000001 | 01 | 1 | \* Modulation wheel | 0-127 | MSB |
| 00000010 | 02 | 2 | Breath control | 0-127 | MSB |
| 00000011 | 03 | 3 | Undefined | 0-127 | MSB |
| 00000100 | 04 | 4 | Foot controller | 0-127 | MSB |
| 00000101 | 05 | 5 | Portamento time | 0-127 | MSB |
| 00000110 | 06 | 6 | Data Entry | 0-127 | MSB |
| 00000111 | 07 | 7 | \* Channel Volume (formerly Main Volume) | 0-127 | MSB |
| 00001000 | 08 | 8 | Balance | 0-127 | MSB |
| 00001001 | 09 | 9 | Undefined | 0-127 | MSB |
| 00001010 | 0A | 10 | \* Pan | 0-127 | MSB |
| 00001011 | 0B | 11 | \* Expression Controller | 0-127 | MSB |
| 00001100 | 0C | 12 | Effect control 1 | 0-127 | MSB |
| 00001101 | 0D | 13 | Effect control 2 | 0-127 | MSB |
| 00001110 | 0E | 14 | Undefined | 0-127 | MSB |
| 00001111 | 0F | 15 | Undefined | 0-127 | MSB |
| 00010000 | 10 | 16 | General Purpose Controller #1 | 0-127 | MSB |
| 00010001 | 11 | 17 | General Purpose Controller #2 | 0-127 | MSB |
| 00010010 | 12 | 18 | General Purpose Controller #3 | 0-127 | MSB |
| 00010011 | 13 | 19 | General Purpose Controller #4 | 0-127 | MSB |
| 00010100 | 14 | 20 | Undefined | 0-127 | MSB |
| 00010101 | 15 | 21 | Undefined | 0-127 | MSB |
| 00010110 | 16 | 22 | Undefined | 0-127 | MSB |
| 00010111 | 17 | 23 | Undefined | 0-127 | MSB |
| 00011000 | 18 | 24 | Undefined | 0-127 | MSB |
| 00011001 | 19 | 25 | Undefined | 0-127 | MSB |
| 00011010 | 1A | 26 | Undefined | 0-127 | MSB |
| 00011011 | 1B | 27 | Undefined | 0-127 | MSB |
| 00011100 | 1C | 28 | Undefined | 0-127 | MSB |
| 00011101 | 1D | 29 | Undefined | 0-127 | MSB |
| 00011110 | 1E | 30 | Undefined | 0-127 | MSB |

### E5: Table of MIDI Controller Messages (Data Bytes) (cont)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 00011111 | 1F | 31 | Undefined | 0-127 | MSB |
| 00100000 | 20 | 32 | Bank Select | 0-127 | LSB |
| 00100001 | 21 | 33 | Modulation wheel | 0-127 | LSB |
| 00100010 | 22 | 34 | Breath control | 0-127 | LSB |
| 00100011 | 23 | 35 | Undefined | 0-127 | LSB |
| 00100100 | 24 | 36 | Foot controller | 0-127 | LSB |
| 00100101 | 25 | 37 | Portamento time | 0-127 | LSB |
| 00100110 | 26 | 38 | Data entry | 0-127 | LSB |
| 00100111 | 27 | 39 | Channel Volume (formerly Main Volume) | 0-127 | LSB |
| 00101000 | 28 | 40 | Balance | 0-127 | LSB |
| 00101001 | 29 | 41 | Undefined | 0-127 | LSB |
| 00101010 | 2A | 42 | Pan | 0-127 | LSB |
| 00101011 | 2B | 43 | Expression Controller | 0-127 | LSB |
| 00101100 | 2C | 44 | Effect control 1 | 0-127 | LSB |
| 00101101 | 2D | 45 | Effect control 2 | 0-127 | LSB |
| 00101110 | 2E | 46 | Undefined | 0-127 | LSB |
| 00101111 | 2F | 47 | Undefined | 0-127 | LSB |
| 00110000 | 30 | 48 | General Purpose Controller #1 | 0-127 | LSB |
| 00110001 | 31 | 49 | General Purpose Controller #2 | 0-127 | LSB |
| 00110010 | 32 | 50 | General Purpose Controller #3 | 0-127 | LSB |
| 00110011 | 33 | 51 | General Purpose Controller #4 | 0-127 | LSB |
| 00110100 | 34 | 52 | Undefined | 0-127 | LSB |
| 00110101 | 35 | 53 | Undefined | 0-127 | LSB |
| 00110110 | 36 | 54 | Undefined | 0-127 | LSB |
| 00110111 | 37 | 55 | Undefined | 0-127 | LSB |
| 00111000 | 38 | 56 | Undefined | 0-127 | LSB |
| 00111001 | 39 | 57 | Undefined | 0-127 | LSB |
| 00111010 | 3A | 58 | Undefined | 0-127 | LSB |
| 00111011 | 3B | 59 | Undefined | 0-127 | LSB |
| 00111100 | 3C | 60 | Undefined | 0-127 | LSB |
| 00111101 | 3D | 61 | Undefined | 0-127 | LSB |
| 00111110 | 3E | 62 | Undefined | 0-127 | LSB |
| 00111111 | 3F | 63 | Undefined | 0-127 | LSB |

### E5: Table of MIDI Controller Messages (Data Bytes) (cont)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01000000 | 40 | 64 | \* Damper pedal on/off (Sustain) | <63=off | >64=on |
| 01000001 | 41 | 65 | Portamento on/off | <63=off | >64=on |
| 01000010 | 42 | 66 | Sustenuto on/off | <63=off | >64=on |
| 01000011 | 43 | 67 | Soft pedal on/off | <63=off | >64=on |
| 01000100 | 44 | 68 | Legato Footswitch | <63=off | >64=on |
| 01000101 | 45 | 69 | Hold 2 | <63=off | >64=on |
| 01000110 | 46 | 70 | Sound Controller 1 (Sound Variation) | 0-127 | LSB |
| 01000111 | 47 | 71 | Sound Controller 2 (Timbre) | 0-127 | LSB |
| 01001000 | 48 | 72 | Sound Controller 3 (Release Time) | 0-127 | LSB |
| 01001001 | 49 | 73 | Sound Controller 4 (Attack Time) | 0-127 | LSB |
| 01001010 | 4A | 74 | Sound Controller 5 (Brightness) | 0-127 | LSB |
| 01001011 | 4B | 75 | Sound Controller 6 | 0-127 | LSB |
| 01001100 | 4C | 76 | Sound Controller 7 | 0-127 | LSB |
| 01001101 | 4D | 77 | Sound Controller 8 | 0-127 | LSB |
| 01001110 | 4E | 78 | Sound Controller 9 | 0-127 | LSB |
| 01001111 | 4F | 79 | Sound Controller 10 | 0-127 | LSB |
| 01010000 | 50 | 80 | General Purpose Controller #5 | 0-127 | LSB |
| 01010001 | 51 | 81 | General Purpose Controller #6 | 0-127 | LSB |
| 01010010 | 52 | 82 | General Purpose Controller #7 | 0-127 | LSB |
| 01010011 | 53 | 83 | General Purpose Controller #8 | 0-127 | LSB |
| 01010100 | 54 | 84 | Portamento Control | 0-127 | Source Note |
| 01010101 | 55 | 85 | Undefined | 0-127 | LSB |
| 01010110 | 56 | 86 | Undefined | 0-127 | LSB |
| 01010111 | 57 | 87 | Undefined | 0-127 | LSB |
| 01011000 | 58 | 88 | Undefined | 0-127 | LSB |
| 01011001 | 59 | 89 | Undefined | 0-127 | LSB |
| 01011010 | 5A | 90 | Undefined | 0-127 | LSB |
| 01011011 | 5B | 91 | Effects 1 Depth | 0-127 | LSB |
| 01011100 | 5C | 92 | Effects 2 Depth | 0-127 | LSB |
| 01011101 | 5D | 93 | Effects 3 Depth | 0-127 | LSB |
| 01011110 | 5E | 94 | Effects 4 Depth | 0-127 | LSB |
| 01011111 | 5F | 95 | Effects 5 Depth | 0-127 | LSB |
|  | | | | | |

### E5: Table of MIDI Controller Messages (Data Bytes) (cont)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01100000 | 60 | 96 | Data entry +1 | N/A |  |
| 01100001 | 61 | 97 | Data entry -1 | N/A |  |
| 01100010 | 62 | 98 | Non-Registered Parameter Number LSB | 0-127 | LSB |
| 01100011 | 63 | 99 | Non-Registered Parameter Number MSB | 0-127 | MSB |
| 01100100 | 64 | 100 | \* Registered Parameter Number LSB | 0-127 | LSB |
| 01100101 | 65 | 101 | \* Registered Parameter Number MSB | 0-127 | MSB |
| 01100110 | 66 | 102 | Undefined | ? |  |
| 01100111 | 67 | 103 | Undefined | ? |  |
| 01101000 | 68 | 104 | Undefined | ? |  |
| 01101001 | 69 | 105 | Undefined | ? |  |
| 01101010 | 6A | 106 | Undefined | ? |  |
| 01101011 | 6B | 107 | Undefined | ? |  |
| 01101100 | 6C | 108 | Undefined | ? |  |
| 01101101 | 6D | 109 | Undefined | ? |  |
| 01101110 | 6E | 110 | Undefined | ? |  |
| 01101111 | 6F | 111 | Undefined | ? |  |
| 01110000 | 70 | 112 | Undefined | ? |  |
| 01110001 | 71 | 113 | Undefined | ? |  |
| 01110010 | 72 | 114 | Undefined | ? |  |
| 01110011 | 73 | 115 | Undefined | ? |  |
| 01110100 | 74 | 116 | Undefined | ? |  |
| 01110101 | 75 | 117 | Undefined | ? |  |
| 01110110 | 76 | 118 | Undefined | ? |  |
| 01110111 | 77 | 119 | Undefined | ? |  |
|  | | | | | |
| 01111000 | 78 | 120 | All Sound Off | 0 |  |
| 01111001 | 79 | 121 | \* Reset All Controllers | 0 |  |
| 01111010 | 7A | 122 | Local control on/off | 0=off | 127=on |
| 01111011 | 7B | 123 | \* All notes off | 0 |  |
| 01111100 | 7C | 124 | Omni mode off (+ all notes off) | 0 |  |
| 01111101 | 7D | 125 | Omni mode on (+ all notes off) | 0 |  |
| 01111110 | 7E | 126 | Poly mode on/off (+ all notes off) | \*\* |  |
| 01111111 | 7F | 127 | Poly mode on (incl mono=off +all notes off) | 0 |  |

### E6: Table of MIDI Note Numbers

This table lists all MIDI Note Numbers by octave.

The absolute octave number designations are based on Middle C = C4, which is an arbitrary but widely used assignment.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Octave #** | **Note Numbers** | | | | | | | | | | | |
|  | **C** | **C#** | **D** | **D#** | **E** | **F** | **F#** | **G** | **G#** | **A** | **A#** | **B** |
| -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 2 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 3 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 4 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 |
| 5 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 |
| 6 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 7 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 |
| 8 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 9 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |  |  |  |  |

### E7: General MIDI Instrument Families

The General MIDI instrument sounds are grouped by families.

In each family are 8 specific instruments.

|  |  |  |  |
| --- | --- | --- | --- |
| **PC#** | **Family** | **PC#** | **Family** |
| 1-8 | Piano | 65-72 | Reed |
| 9-16 | Chromatic Percussion | 73-80 | Pipe |
| 17-24 | Organ | 81-88 | Synth Lead |
| 25-32 | Guitar | 89-96 | Synth Pad |
| 33-40 | Bass | 97-104 | Synth Effects |
| 41-48 | Strings | 105-112 | Ethnic |
| 49-56 | Ensemble | 113-120 | Percussive |
| 57-64 | Brass | 121-128 | Sound Effects |

### E8: General MIDI Patch Map

**Note:** While GM does not define the actual characteristics of any sounds, the names in parentheses after each of the synth leads, pads, and sound effects are, in particular, intended only as guides.

|  |  |  |  |
| --- | --- | --- | --- |
| **PC#** | **Instrument** | **PC#** | **Instrument** |
| 1. | Acoustic Grand Piano | 65. | Soprano Sax |
| 2. | Bright Acoustic Piano | 66. | Alto Sax |
| 3. | Electric Grand Piano | 67. | Tenor Sax |
| 4. | Honky-tonk Piano | 68. | Baritone Sax |
| 5. | Electric Piano 1 (Rhodes Piano) | 69. | Oboe |
| 6. | Electric Piano 2 (Chorused Piano) | 70. | English Horn |
| 7. | Harpsichord | 71. | Bassoon |
| 8. | Clavinet | 72. | Clarinet |
| 9. | Celesta | 73. | Piccolo |
| 10. | Glockenspiel | 74. | Flute |
| 11. | Music Box | 75. | Recorder |
| 12. | Vibraphone | 76. | Pan Flute |
| 13. | Marimba | 77. | Blown Bottle |
| 14. | Xylophone | 78. | Shakuhachi |
| 15. | Tubular Bells | 79. | Whistle |
| 16. | Dulcimer (Santur) | 80. | Ocarina |
| 17. | Drawbar Organ (Hammond) | 81. | Lead 1 (square wave) |
| 18. | Percussive Organ | 82. | Lead 2 (sawtooth wave) |
| 19. | Rock Organ | 83. | Lead 3 (calliope) |
| 20. | Church Organ | 84. | Lead 4 (chiffer) |
| 21. | Reed Organ | 85. | Lead 5 (charang) |
| 22. | Accordion (French) | 86. | Lead 6 (voice solo) |
| 23. | Harmonica | 87. | Lead 7 (fifths) |
| 24. | Tango Accordion (Band neon) | 88. | Lead 8 (bass + lead) |
| 25. | Acoustic Guitar (nylon) | 89. | Pad 1 (new age Fantasia) |
| 26. | Acoustic Guitar (steel) | 90. | Pad 2 (warm) |
| 27. | Electric Guitar (jazz) | 91. | Pad 3 (polysynth) |
| 28. | Electric Guitar (clean) | 92. | Pad 4 (choir space voice) |
| 29. | Electric Guitar (muted) | 93. | Pad 5 (bowed glass) |
| 30. | Overdriven Guitar | 94. | Pad 6 (metallic pro) |

### E8: General MIDI Patch Map (cont)

|  |  |  |  |
| --- | --- | --- | --- |
| 31. | Distortion Guitar | 95. | Pad 7 (halo) |
| 32. | Guitar harmonics | 96. | Pad 8 (sweep) |
| 33. | Acoustic Bass | 97. | FX 1 (rain) |
| 34. | Electric Bass (fingered) | 98. | FX 2 (soundtrack) |
| 35. | Electric Bass (picked) | 99. | FX 3 (crystal) |
| 36. | Fretless Bass | 100. | FX 4 (atmosphere) |
| 37. | Slap Bass 1 | 101. | FX 5 (brightness) |
| 38. | Slap Bass 2 | 102. | FX 6 (goblins) |
| 39. | Synth Bass 1 | 103. | FX 7 (echoes, drops) |
| 40. | Synth Bass 2 | 104. | FX 8 (sci-fi, star theme) |
| 41. | Violin | 105. | Sitar |
| 42. | Viola | 106. | Banjo |
| 43. | Cello | 107. | Shamisen |
| 44. | Contrabass | 108. | Koto |
| 45. | Tremolo Strings | 109. | Kalimba |
| 46. | Pizzicato Strings | 110. | Bag pipe |
| 47. | Orchestral Harp | 111. | Fiddle |
| 48. | Timpani | 112. | Shanai |
| 49. | String Ensemble 1 (strings) | 113. | Tinkle Bell |
| 50. | String Ensemble 2 (slow strings) | 114. | Agogo |
| 51. | SynthStrings 1 | 115. | Steel Drums |
| 52. | SynthStrings 2 | 116. | Woodblock |
| 53. | Choir Aahs | 117. | Taiko Drum |
| 54. | Voice Oohs | 118. | Melodic Tom |
| 55. | Synth Voice | 119. | Synth Drum |
| 56. | Orchestra Hit | 120. | Reverse Cymbal |
| 57. | Trumpet | 121. | Guitar Fret Noise |
| 58. | Trombone | 122. | Breath Noise |
| 59. | Tuba | 123. | Seashore |
| 60. | Muted Trumpet | 124. | Bird Tweet |
| 61. | French Horn | 125. | Telephone Ring |
| 62. | Brass Section | 126. | Helicopter |
| 63. | SynthBrass 1 | 127. | Applause |
| 64. | SynthBrass 2 | 128. | Gunshot |

### E9: General MIDI Percussion Key Map

On MIDI Channel 10, each MIDI Note number ("Key#") corresponds to a different drum sound, as shown below. GM-compatible instruments must have the sounds on the keys shown here. While many current instruments also have additional sounds above or below the range show here, and may even have additional "kits" with variations of these sounds, only these sounds are supported by General MIDI.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key#** | **Note** | **Drum Sound** | **Key#** | **Note** | **Drum Sound** |
| 35 | B1 | Acoustic Bass Drum | 59 | B3 | Ride Cymbal 2 |
| 36 | C2 | Bass Drum 1 | 60 | C4 | Hi Bongo |
| 37 | C#2 | Side Stick | 61 | C#4 | Low Bongo |
| 38 | D2 | Acoustic Snare | 62 | D4 | Mute Hi Conga |
| 39 | D#2 | Hand Clap | 63 | D#4 | Open Hi Conga |
| 40 | E2 | Electric Snare | 64 | E4 | Low Conga |
| 41 | F2 | Low Floor Tom | 65 | F4 | High Timbale |
| 42 | F#2 | Closed Hi Hat | 66 | F#4 | Low Timbale |
| 43 | G2 | High Floor Tom | 67 | G4 | High Agogo |
| 44 | G#2 | Pedal Hi-Hat | 68 | G#4 | Low Agogo |
| 45 | A2 | Low Tom | 69 | A4 | Cabasa |
| 46 | A#2 | Open Hi-Hat | 70 | A#4 | Maracas |
| 47 | B2 | Low-Mid Tom | 71 | B4 | Short Whistle |
| 48 | C3 | Hi Mid Tom | 72 | C5 | Long Whistle |
| 49 | C#3 | Crash Cymbal 1 | 73 | C#5 | Short Guiro |
| 50 | D3 | High Tom | 74 | D5 | Long Guiro |
| 51 | D#3 | Ride Cymbal 1 | 75 | D#5 | Claves |
| 52 | E3 | Chinese Cymbal | 76 | E5 | Hi Wood Block |
| 53 | F3 | Ride Bell | 77 | F5 | Low Wood Block |
| 54 | F#3 | Tambourine | 78 | F#5 | Mute Cuica |
| 55 | G3 | Splash Cymbal | 79 | G5 | Open Cuica |
| 56 | G#3 | Cowbell | 80 | G#5 | Mute Triangle |
| 57 | A3 | Crash Cymbal 2 | 81 | A5 | Open Triangle |
| 58 | A#3 | Vibraslap |  |  |  |