

~~~~~

**MIDI-SWITCHER**

~~~~~

INTRODUCTION

MIDI-Switcher is an electronic device which uses incoming MIDI note information to activate its eight output ports. These ports are electronic switches which can be used to activate small solenoids, motors, light bulbs, relays etc. For example solenoids could be used as robotic percussion instruments that could be driven from a MIDI sequencer or DAW program.

CONNECTING DEVICES

Each port operates as a **low side switch** with a maximum current handling capacity of 1 Amp. Each port provides a connector block with two screw terminals:

- One screw terminal (the one closest to the rear of the board) is permanently connected to the positive connection of the power supply.
- The other screw terminal is electronically switched to ground (the "negative" power supply connection) when the corresponding MIDI note is played into the board.

A legend on the top of the board, and labels on the underside of the board, also provide this information.

Many types of load that you might connect to the MIDI switcher (e.g. solenoids, relays, filament bulbs) are not fussy about polarity and can be connected to MIDI-Switcher's ports either way around.

However, always be sure to check if your device has a specific polarity and connect it appropriately.

CONNECTING POWER

Since we are working with relatively high current loads, your choice of power supply is important.

Power the board from a **DC power supply of between 7 and 24 volts**. Connect the power supply to the screw terminal connection at the back of the board making sure the polarity of the power supply is correct. ***If you connect the power supply with incorrect polarity you may damage the transistor arrays!***

When selecting a power supply to use with the MIDI-Switcher, you will need to ensure that it has a current rating (Amps) that is sufficient for the devices you wish to switch.

This depends on what those devices are, and whether they are constant loads (like motors or bulbs) or transient ones (like solenoids used with percussion)

Likewise, you will need a supply of an appropriate voltage to your load. 12 volts is typically a good choice for small solenoids and vehicle bulbs.

For eight small 12 volt solenoids being used for percussion, a supply of 2 Amps rating will probably work. Higher current ratings are better, giving more headroom.

ABSOLUTELY NEVER ATTEMPT TO USE A MAINS ELECTRICITY SUPPLY DIRECTLY!!!

MIDI CONNECTION

MIDI-Switcher has a single MIDI-IN connection utilising MIDI standard 5-Pin DIN socket.

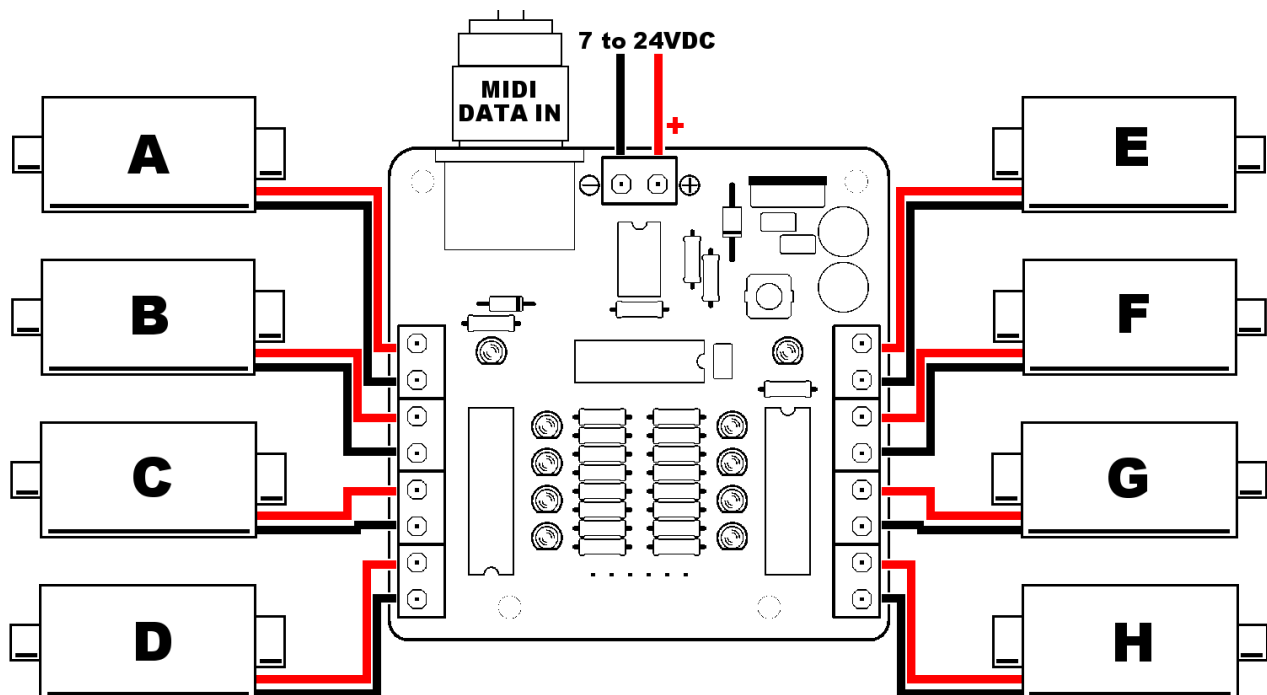
Most MIDI equipment has such a connection, but with a computer you may require a MIDI interface with the appropriate connector. A cheap and effective solution is a USB to MIDI cable which connects to a computer USB port and provides standard MIDI input and output cables terminated in 5-Pin DIN plugs.

Remember each MIDI cable works in one direction only... The MIDI-Switcher requires a MIDI input only, so you connect it to the MIDI OUT of your controlling device.

There is no communication back from the MIDI-Switcher to the controller. Keep this in mind when you are reconfiguring the device. Keep a backup of your customised configurations (as described later). You can't download them again.

EXAMPLE CONNECTIONS

The diagram below shows how 8 solenoids are connected. Most solenoids don't have a polarity and are likely to have two wires of the same colour. In that case you can connect them to the port in either direction. However, always ensure the power supply polarity is correct.. that DOES matter!



GETTING STARTED

Once you are powered up, try sending some MIDI... the default settings programmed in the Microcontroller are as follows

Port	Note	Port	Note
PORT A	C3 (60)	PORT E	E3 (64)
PORT B	C#3 (61)	PORT F	F3 (65)
PORT C	D3 (62)	PORT G	F#3 (66)
PORT D	D#3 (63)	PORT H	G3 (67)

The default receive channel is MIDI Channel 1 for all ports. Output is set at 100% duty (i.e. constantly ON when triggered) and 20 milliseconds duration. Modulation of duty and duration is disabled by default.

CUSTOMISING THE CONFIGURATON

Each of the eight switching ports have multiple parameters that can be configured over MIDI. A simple Java application is available to allow a user-edited configuration file to be loaded on to the board. The new configuration can be stored in EEPROM so it is available as the default configuration each time the board is powered on.

The MIDI channel/note combinations that trigger each port are configurable.

Ports can be configured so that they are ON for as long as their corresponding MIDI trigger note is held, or for a fixed period of time, even if the MIDI note is released before that time period completes.

There are two methods by which "velocity sensitivity" can be achieved

- The triggered output ON time can be configured to be "modulated" by the velocity of the trigger note. For solenoid percussion the striking force can easily be controlled this way (the times are usually of the order of tens of milliseconds)
- The "duty cycle" of the ON pulse can be modulated. When using this method, the output actually switches on and off very rapidly during the ON period, and the percentage ratio of the "on" versus "off" time is what we call the duty cycle. This directly controls the amount of power delivered to the load over the ON time and works well with light bulbs and motors. The note velocity can be used to modulate the duty cycle of the output ON pulse.

Instead of note velocity, we can use a MIDI "continuous controller" as the modulator (for example the Modulation Wheel) or we can use the Pitch Bend controller. Modulation sources can be set independently per port and separately for pulse duration and duty cycle on each port.

The Java application is described in detail below. Alternatively the MIDI messages that reconfigure the board can be sent from your own application or controller.

Please note that the board needs to be put into a special programming mode (by holding the MODE button for 1-2 seconds) to allow reconfiguration. This is a safety feature to prevent inadvertent change of the configuration. When not in this mode all reconfiguration messages are ignored.

A modified configuration must be saved (by pressing the MODE button again, or via a MIDI message) or it will be lost when power is removed.

LED BLINK CODES

The power LED (PWR) should illuminate when power is applied to the board.

The activity (ACT) LED indicates the status of the board as follows...

- When there is activity on the MIDI input, the LED flickers. This is regardless of whether the MIDI message is processed by the MIDI-Switcher or ignored.
- MIDI-Switcher is placed into programming mode by pressing and holding the MODE button. In programming mode, the LED **pulses briefly once every 2 seconds (approx)**. Programming mode remains in effect until the board is powered down.
- In Programming mode, the device configuration can be changed by special MIDI messages (usually from the configuration applet). When a change has been made to the device configuration, but not yet saved, the LED **pulses approx twice a second**. This continues until the changed configuration is saved to EEPROM, which can be performed via MIDI, or by pressing the MODE button again.
- On power-up the device reads settings from EEPROM. A checksum test is used to ensure integrity of the configuration. If the check fails at power-up, the LED **blinks 10 times** and the default settings are saved to EEPROM and made active. It is normal to see this the very first time you power on the device, when the EEPROM is uninitialised but if this is happening unexpectedly on power-on this could indicate a problem.

OTHER INFO

- Do not exceed the current rating of your DC power supply. **Ensure that the supply is not becoming excessively hot.**
- While each channel of MIDI-Switcher can switch a 1 Amp load, this would usually be for a short period of time (such as when actuating a solenoid). **The board is not designed to large handle large continuous currents through multiple ports.** You could use external relays to increase the continuous current capacity if you like - the channels of the MIDI Switcher board could then be used to actuate the relay coils rather than switch the load directly.
- Be aware that some of the components can become quite warm or even hot during normal use; The voltage regulator (mounted vertically at the rear right side of the board) and the two transistor array ICs (close to each row of terminal blocks) will warm up. If these parts are getting excessively hot you may be overloading the board - allow it to cool and reduce the loading. **Be very careful when handling hot components - in exceptional situations they can become hot enough to burn your fingers.**
- When running inductive loads such as solenoids it is sometimes possible to feel the "flyback" voltage spike as a small electric shock when you touch the terminal block. While this is harmless, it can come as a nasty surprise, so keep fingers away from the connections when solenoids are operating!
- If desired, you could connect the "high side" (positive) terminals of your loads to second power supply separate from the supply used for the board (but sharing a common ground). This configuration could be used to switch loads requiring higher voltages than the 24 Volt maximum for the board supply. The transistor arrays (ULN2803A) can handle 50 Volts in this configuration.
- **Absolutely NEVER connect the board directly to the mains supply!**

USING THE JAVA APPLICATION

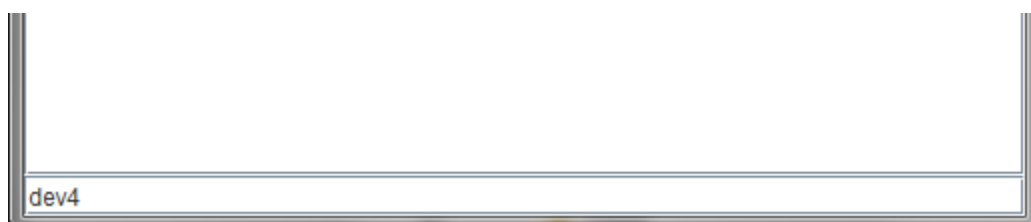
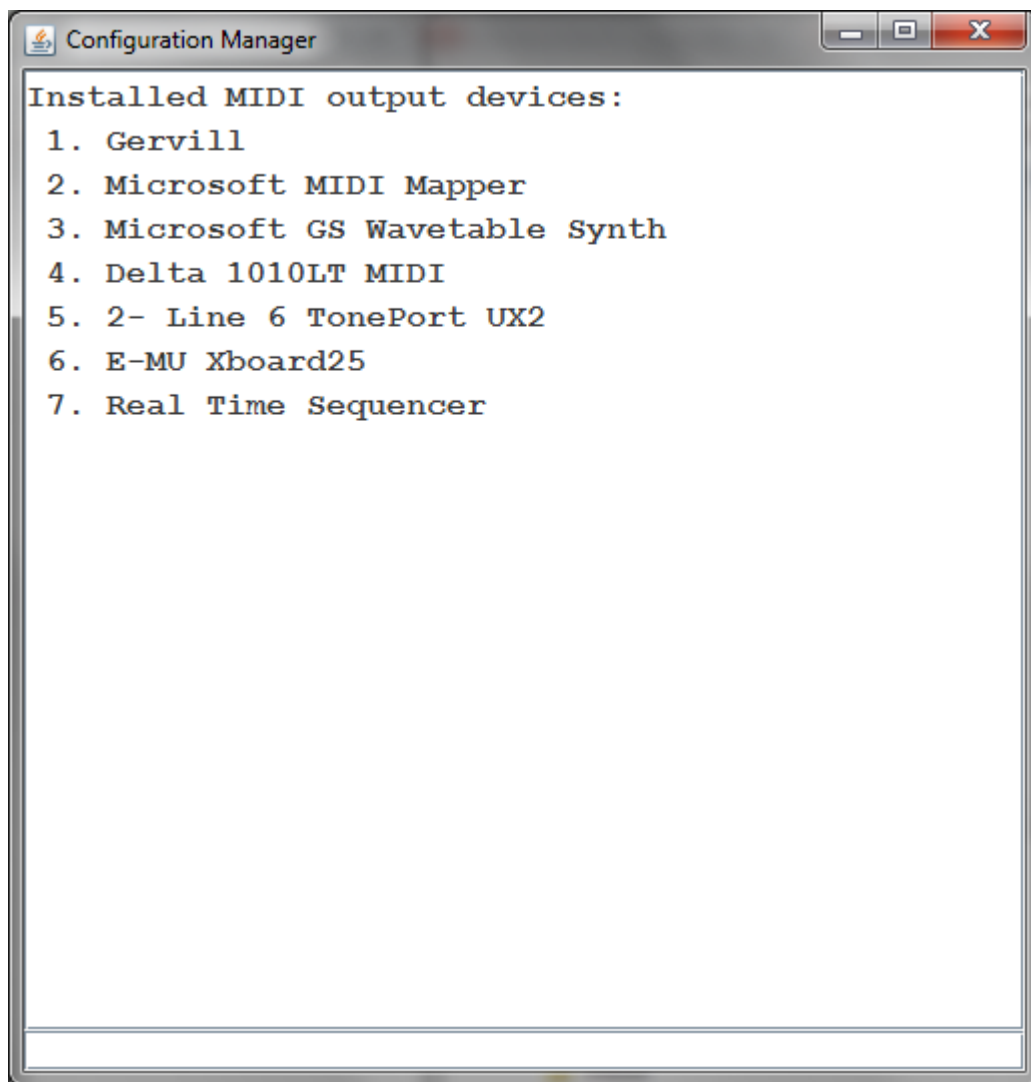
The configuration application is distributed as a runnable JAR file, which requires Java to be installed on your computer.

When the program first loads, you need to select the MIDI output device that you are using to communicate with the MIDI Switcher board.

The program lists all available devices on your system. Locate the correct device and enter (in the input area at the bottom of the screen) the command **dev<x>** (replacing <x> with the appropriate number for your device).

For example to use device 4, enter **dev4**

Make sure the device is not already in use by other programs.



Having entered the dev command, the device should be displayed with a ">" symbol

```
Installed MIDI output devices:
 1. Gervill
 2. Microsoft MIDI Mapper
 3. Microsoft GS Wavetable Synth
>4. Delta 1010LT MIDI
 5. 2- Line 6 TonePort UX2
 6. E-MU Xboard25
 7. Real Time Sequencer
```

Enter **porta** to access the port configuration screen

```
Configuration Manager
>porta ch1 @60 dur5 duty100 inv0
portb ch1 @61 dur5 duty100 inv0
portc ch1 @62 dur5 duty100 inv0
portd ch1 @63 dur5 duty100 inv0
porte ch1 @64 dur5 duty100 inv0
portf ch1 @65 dur5 duty100 inv0
portg ch1 @66 dur5 duty100 inv0
porth ch1 @67 dur5 duty100 inv0
```

The ">" symbol now shows that **porta** is the selected port, to which entered commands apply. You can change the selection between ports using commands **porta**, **portb** etc. As a shortcut you can enter **/a** or **/b** instead.

Configure the current port by entering commands like the following:

- **ch2** - Change the trigger MIDI channel for the selected port to channel 2. This is the MIDI channel that this port listens on for its trigger note.
- **@30** - Change the trigger note of the selected port to MIDI note 30
- **dur30** - Set the ON pulse duration to 30 milliseconds for selected port. Enter 0 for the duration to indicate that the duration is for as long as the MIDI note is held.
- **duty50** - Set the ON pulse duty to 50% for selected port. Use 100 to have the port use a continuous ON pulse when triggered. Use 0 to disable the output.
- **inv1** - Set selected port output to be inverted (active low). This means the output is ON when not triggered and OFF when triggered.

Specify duration modulation sources as shown below. "Duration modulation" is how we can use the note velocity, or another MIDI controller, to change the output pulse length (good for changing the "hit" velocity of a solenoid)

- **dur~v** - modulate ON pulse duration using note velocity
- **dur~1** - modulate ON pulse duration using controller #1 (Mod wheel)
- **dur~p** - modulate ON pulse duration using pitch bend
- **dur~** - remove modulation

Similarly we can modulate the duty cycle modulation using

- **duty~v** - modulate ON pulse duty using note velocity
- **duty~1** - modulate ON pulse duty using controller #1 (Mod wheel)
- **duty~p** - modulate ON pulse duty using pitch bend
- **duty~** - remove modulation

Any of the above commands can be applied to **all ports simultaneously** by prefixing a command with the asterisk symbol. For example ***ch3** sets all eight ports to trigger from MIDI channel 3

Multiple commands can be entered at the same time as long as they are separated by white space (including newlines). This allows the entire configuration to be copied to the clipboard and saved as a text file to be edited/pasted back into the command line at a later time. In this way you can keep entire configuration "patches" as blocks of text.

If a given command is not understood, it will remain on the command line and a beep will sound. If multiple commands are entered, they will be processed until the first unrecognised command. The successful commands will be removed from the command line.

Commands are sent to MIDI as they are entered. If you need to resend (for example the MIDI Switcher board was not connected) you can resend the configuration for a specific channel by entering **!** as a command. To resend all channels enter ***!**

The MIDI-Switcher uses the new parameters as they are received, but does not save them to EEPROM (so they will be lost when powered down). If you wish to save the new set of parameters to EEPROM so they are used as the new power-on defaults you can send the command **^** (caret). Alternatively the MODE button on the board can be pressed to save the new settings.

Note that the initial information shown in the window reflects the default settings for the ports and not necessarily the current settings that are active on the board. There is no communication back from the board, so the application does not know the current settings. This is why it is useful to copy a finalised configuration to the clipboard and paste it somewhere safe!

MIDI IMPLEMENTATION

MIDI-Switcher is reconfigured over MIDI using Non-Registered Parameter Number (NRPN) Messages, which it receives on MIDI channel 1. This is one of the ways that MIDI allows device-specific data can be exchanged with a MIDI device.

To avoid inadvertant changes to configuration, MIDI-Switcher can only receive these NRPN messages when it is placed in "configuration" mode by holding the MODE button down.

Each parameter is uploaded by means of a set of MIDI messages that first provide the target parameter number, then the data to load into the parameter. Each message is 3 bytes long (shown in hexadecimal notation below)

Non-Registered Parameter Number: Most Significant Byte (MSB)

0xB0	0x63	Param# MSB (see table)
------	------	---------------------------

Non-Registered Parameter Number: Least Significant Byte (LSB)

0xB0	0x62	Param# LSB (see table)
------	------	---------------------------

Data Entry: Most Significant Byte (MSB)

0xB0	0x06	Data / 128
------	------	------------

Data Entry: Least Significant Byte (MSB)

0xB0	0x26	Data % 128
------	------	------------

Data Entry for a given parameter must occur in the order MSB, then LSB. When the MSB message is received, the LSB is cleared.

<u>NRPN (MSB)</u>	<u>NRPN (LSB)</u>	<u>DATA VALUE (MSB * 128 + LSB)</u>
Port number 1-8	1	MIDI Trigger Channel 0-15
	2	MIDI Trigger Note 0-127
	3	1-16384 = Fixed Trigger Duration in milliseconds. 0 = Trigger for as long as trigger note is held
	4	Duration modulator Specifies a modulation source that scales trigger duration 0-127 = MIDI Continuous controller on trigger channel (use 1 for modulation wheel) 128 = No modulation; always apply full duration 129 = Use the trigger note velocity to modulate the trigger duration 130 = use pitchbend to modulate the trigger duration (no bend = 50% of maximum duration)
	5	0-100 = Trigger duty cycle percentage.

		The carrier is approx 80Hz.
	6	Duty cycle modulator Specifies a modulation source that scales the duty cycle up to the maximum set 0-127 = MIDI Continuous controller on trigger channel (use 1 for modulation wheel) 128 = No modulation; always apply the maximum duty cycle value 129 = Use the trigger note velocity to modulate the duty cycle 130 = use pitchbend to modulate the duty cycle (no bend = half of maximum value)
	7	Invert output 0 = Normal 1 = Switch output off when triggered
100	1	Write configuration to EEPROM. This can also be performed by pressing the mode button on the board after uploading modifications to the configuration