The MIDI Switcher board uses MIDI note information to activate its eight outputs. The outputs can be used to drive small solenoids, small motors, light bulbs, relays etc. For example solenoids could be used to create robotic percussion instruments that could be driven from a MIDI sequencer or DAW program.

Each channel operates as a low side switch, pairing up two channels of a ULN2803 transistor array to achieve a maximum current handling capacity of 1 Amp. The ULN2803 contains integrated flyback diodes to allow inductive loads (solenoids, motors, relays) to be driven.

The board is intended to be powered from a single DC supply of 7-24 volts. An on-board voltage regulator supplies 5V to the logic circuit. If preferred you can run the loads from a separate DC supply that shares a ground with the supply to the board.

While each channel can switch a 1 Amp load, this would strictly be for a short period of time (such as when actuating a solenoid). The PCB is not designed to large handle large continuous currents, although you could use external relays to achieve this 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.

A DC power supply is required. This needs to be at least 7 volts to allow the onboard voltage regulator to work properly. Do not exceed 24 volts and alway make sure that the supply you use is rated to provide enough current for the total load you will be driving. **Absolutely NEVER connect the board directly to the mains supply!**

Please 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 suprise, so keep fingers away from the connections when solenoids are operating!

Each of the eight switching "ports" has a number of parameters that can be user 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.

Ports can be configured so that they are ON for as long as their corresponding MIDI trigger note is held. They can also be configured so that every time the port is triggered, the output is switched ON 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.

Rather than modulating the outputs period or duty based on trigger note velocity, we can alternatively 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 switching port and separately for pulse duration and duty cycle on each port.

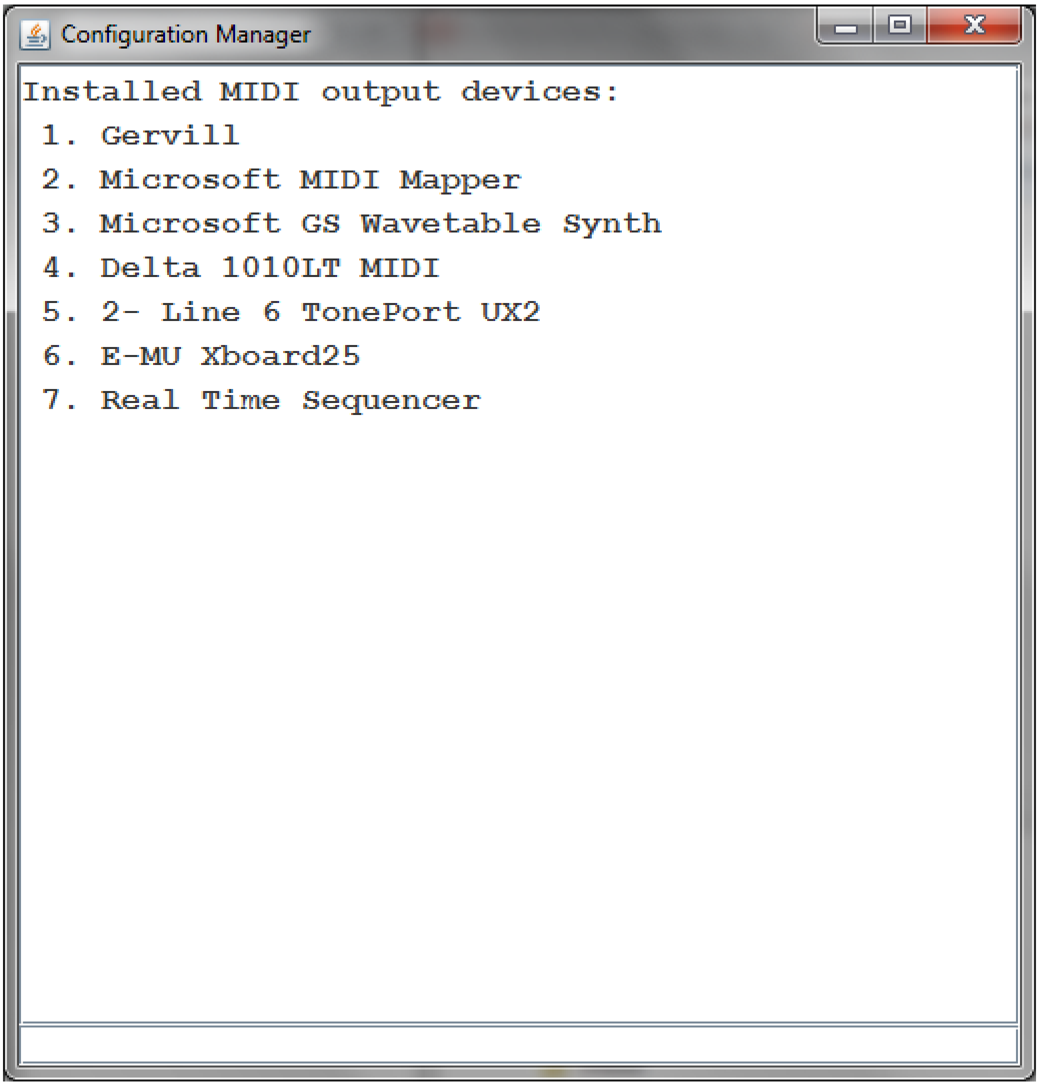
The MIDI channel/note combinations that trigger each port are user configurable over MIDI

A typical use case is for samll sol

NRPN (MSB)	NRPN (LSB)	
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

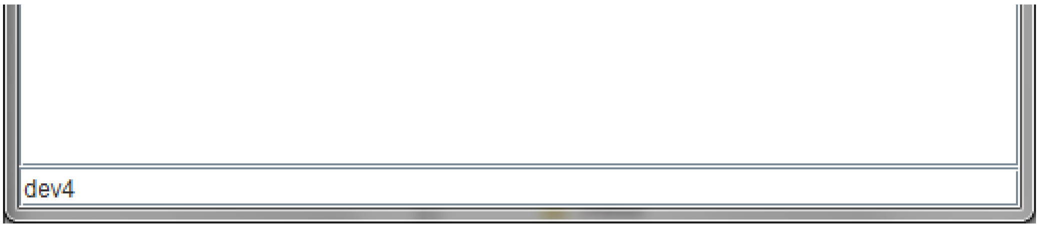
Using the Configuration Manager Application

Download and run the .JAR file



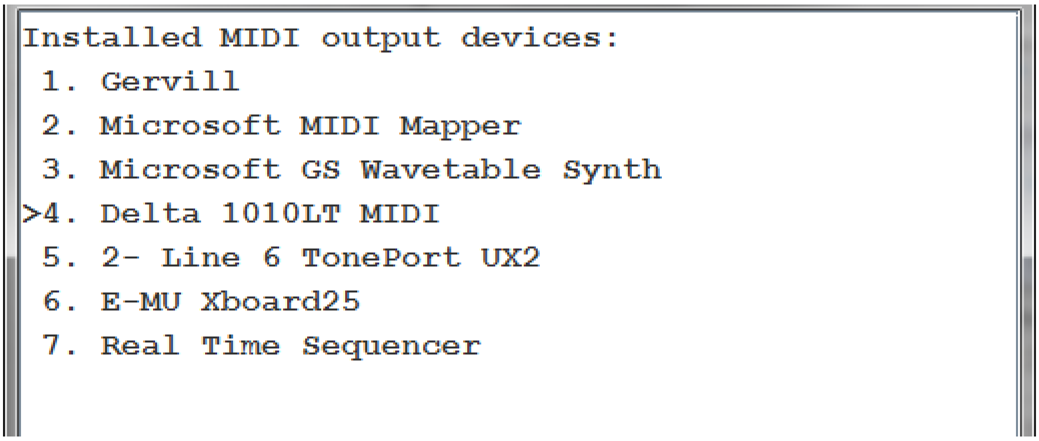
```
Installed MIDI output devices:
 1. Gervill
 2. Microsoft MIDI Mapper
 3. Microsoft GS Wavetable Synth
 4. Delta 1010LT MIDI
 5. 2- Line 6 TonePort UX2
 6. E-MU Xboard25
 7. Real Time Sequencer
```

When the program first loads, you need to select the MIDI output device (port) that you are using to communicate with the MIDI Switcher board. Using the number shown on the display, enter (in the input area at the bottom of the screen) the command **dev4** (replacing 4 with the appropriate device number



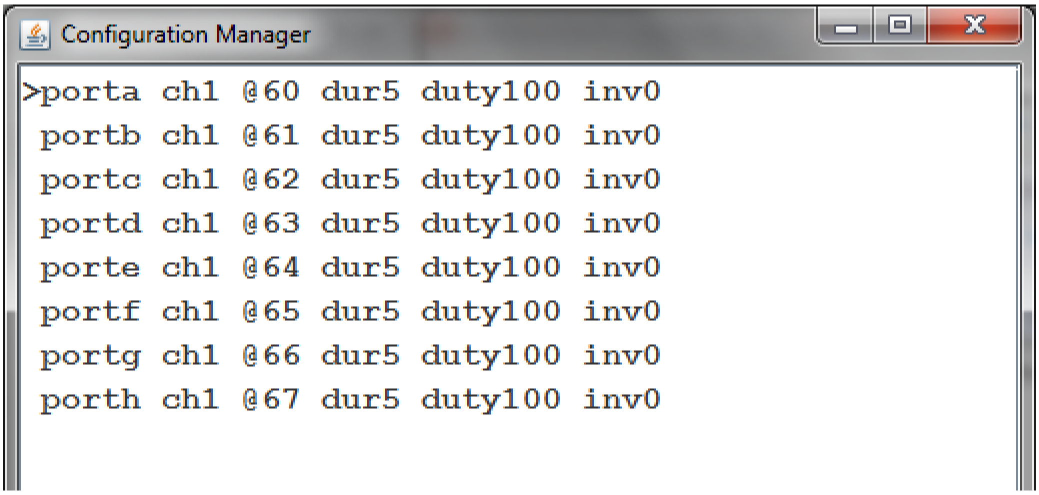
```
dev4
```

The selected device should be displayed with a ">"



```
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



```
>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
```

you can configure the port by entering commands such as

- **ch2** - Change the trigger MIDI channel for the selected port to channel 2
- **@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
- **duty50** - Set the ON pulse duty to 50% for selected port
- **inv1** - Set selected port output to be inverted (active low)

You can specify modulation sources as follows (for duration)

- **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

For duty cycle modulation use commands like **duty~p**

Any command 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

Change between ports a-h using the port command (**porta**, **portb** etc). As a shortcut you can enter **/a** instead of porta etc.

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 copy/pasted from the top window and saved as a text file to be edited/pasted back into the command line. 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 hardware 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.