MIDI Switcher allows you to control electrical devices such as solenoids, motors, relays, LEDs and filament light bulbs using MIDI messages. While it is ideal for creating electromechanical musical instruments, it has numerous potential uses. However, there are a couple of important limitations

* The switching is “low side” power switching. This means that each of the output connects can connect a load to a common ground and complete a circuit. This means that won’t be suitable for directly switching loads that cannot be directly connected to ground (for example a channel select footswitch output on some guitar amplifiers)
* Only DC currents can be switched, not AC power or signals such as audio. **You should NEVER connect mains electricity to the switcher**

The above limitations can be worked around using external relays, such that the MIDI switcher turns the relay on and the relay makes the actual connection to the signal or current being switched. This will be explained later.



The MIDI Switcher has eight switching outputs using switching transistors (MJD-122) that are each rated for up to 8A or 100V (Power rating 24W). When switching large currents or voltages there are some special precautions to observe which will be explained later.

Each output can be configured to be triggered based on the following conditions

* MIDI notes (with optional velocity range filter)
* MIDI continuous controller message (with selectable value range)
* MIDI program change (can set all outputs with a single message)

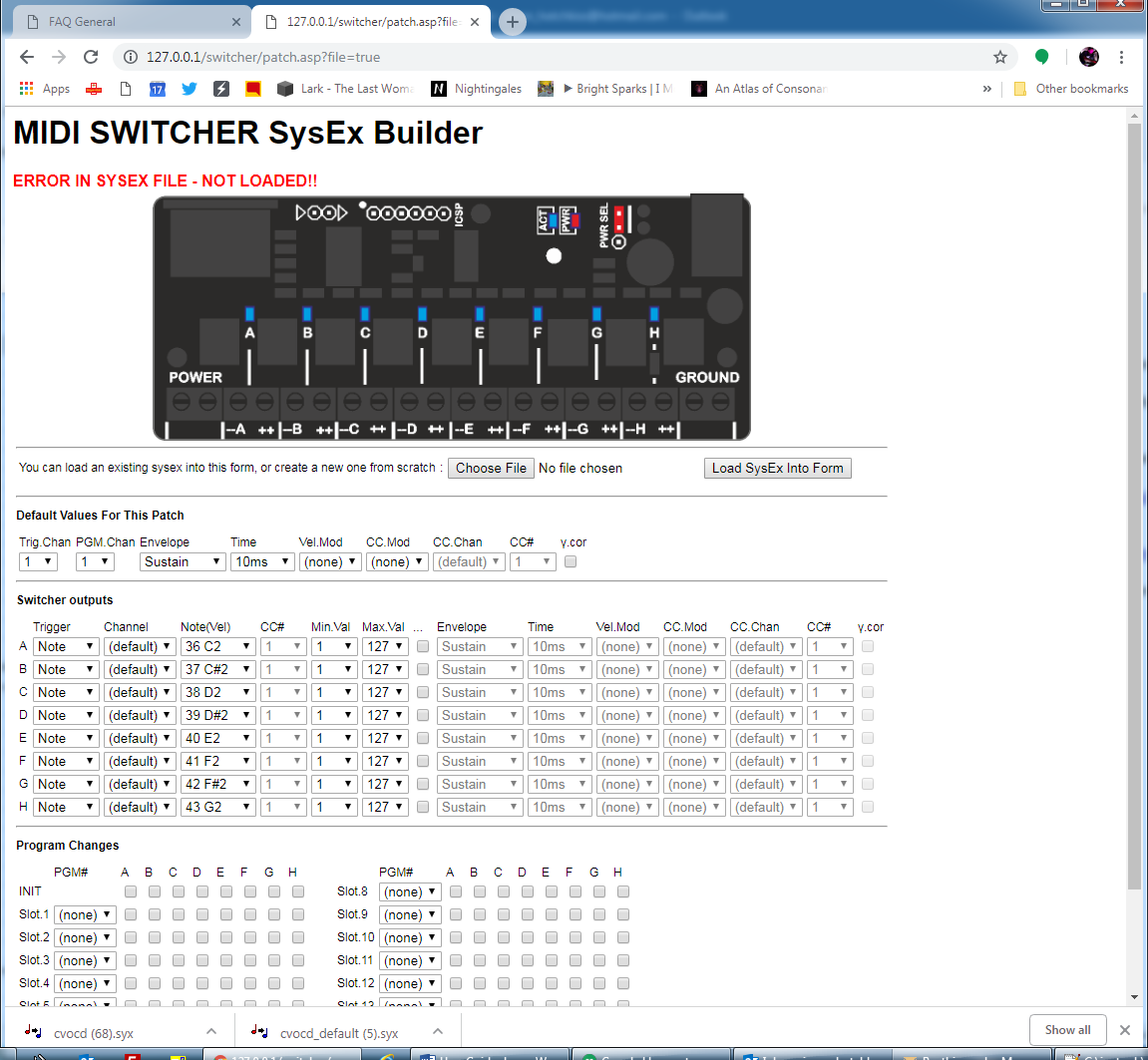
The switch output can be “shaped” following triggering in several ways

* Control over switch pulse duration based on a combination of how long the trigger condition lasts (e.g. how long note is held) and/or a selectable amount of time.
* Pulse width modulation (PWM) control over switch power
* Note velocity can be used to modulate the switching power or duration (e.g. for velocity sensitive percussion)
* A separate MIDI controller (CC) can be used to modulate the switching power or duration
* A “release” can be defined so that power starts at maximum value and falls off during the switch duration (good for LED fades)
* Gamma correction can be applied to switching power (intended for use with LEDs and light bulbs). This helps to expand the range of power levels that will map to brightness levels that look different to human eyes.

MIDI Notes

When an output is set up to switch based on a MIDI note, it will be switched on when the MIDI note starts playing and, based on pulse-shaping settings, it is switched off when the note stops playing, or after a period of time (see later)

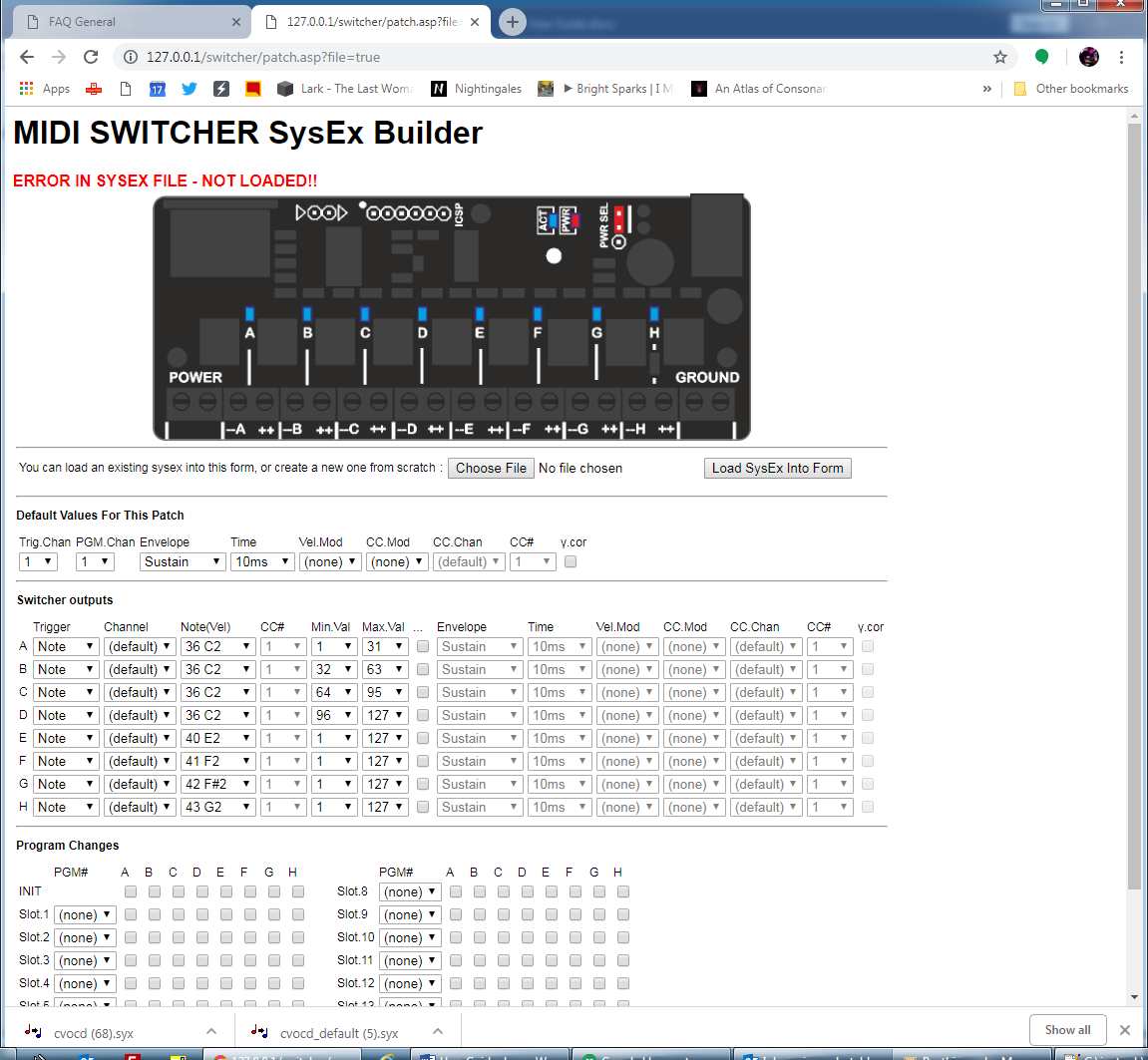
Typically, a note trigger is set up as follows



In this case a C2 note (C note, octave 2) on MIDI channel 1 will trigger output A. Often all the triggers will be on the same channel, so it is most convenient to use the default trigger channel, but you can select specific MIDI channels per output if needed.

As well as pitch information, MIDI note messages have velocity information (i.e. how hard a key was hit). Typically, you might only be interested in whether a note is “on” or “off” so the Min Val and Max Val fields can be left as 1 and 127 (the full velocity range of a note on message)

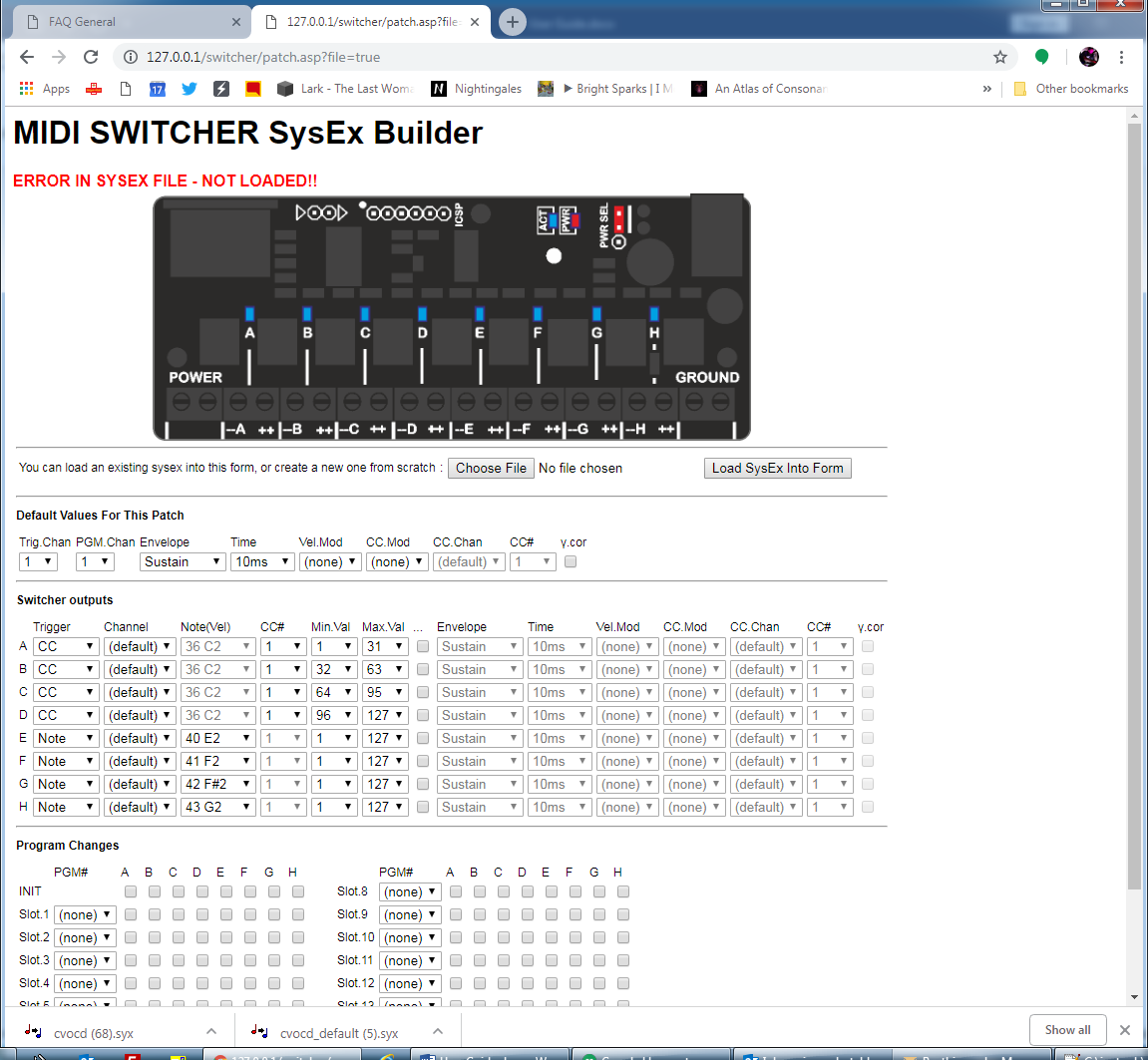
However perhaps you want to trigger different outputs based on the velocity of the same note. This is possible by setting up filters on the note velocity as shown below



Note that this is different to controlling the output power based on the note velocity. This will be described later.

MIDI Continuous Controller (CC)

When an output is set up to switch based on a MIDI CC, it will be switched on when a CC message is received that matches the controller number and where the controller value is between specified limits. The switcher output is triggered on when the CC value enters the range and, based on pulse-shaping settings, it is switched off when the CC value exits the range, or after a period of time (see later)



In this example CC #1 (Modulation wheel) on MIDI channel 1 will trigger outputs A, B, C and D depending on the position of the mod wheel. Here the triggers are on the same channel, so it is most convenient to use the default trigger channel, but we could select specific MIDI channels per output if needed.

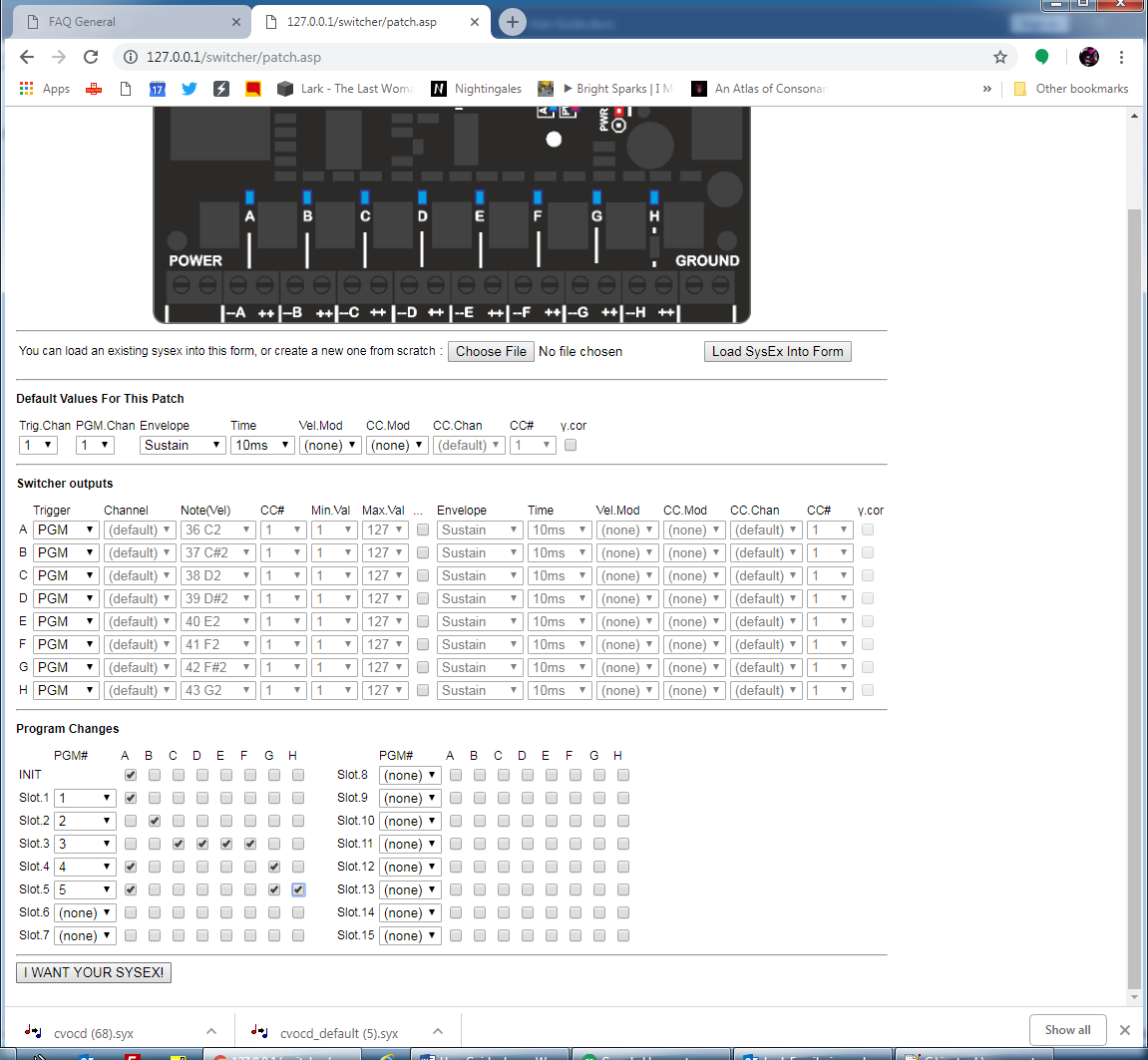
Program Changes

The MIDI switcher is able to listen for MIDI Program Change messages (usually used to select a patch on a synth). A program change message has a MIDI channel and a program number 0-127.

To use program changes you need to setup the PGM Chan (in the Default Values section) to the appropriate MIDI channel on which program changes will be received.

Next configure the outputs that need to be switched by program changes by selecting PGM as their trigger. If you like you can have some outputs that are controlled by program changes and others that are controlled by notes etc, however an output that is controlled by program changes will only be controlled that way.

Now you need to select which outputs are controlled by which program changes. There are a total of 15 user defined trigger combination slots available, and for each you need to select the specific program number that will be matched by the slot, and what the states of the outputs will be. Note that only outputs that are in PGM trigger mode will be affected, even if you check the boxes for them.



When a program change message is matched, it will “trigger” the appropriate channels in the same way that a MIDI note on message might trigger channels, so the usual pulse shaping options can be used. To have the outputs remain on continuously until another program change switches them off, they should be in the “Sustain” envelope mode.

The INIT settings are triggered at power on or when the outputs are reset.

Always On

The “Always” trigger mode means that a given output is triggered at power on and when the reset button is pressed. By setting the envelope type to Sustain, this means the output is on all the time. This can be useful if you simply want to control the power of an output using a CC and without having to send a note message to trigger it.

The LEDs

* The PWR LED is on all the time the board has power. If you are powering the board and the LED is off, check the position of the PWR SEL jumper (see below)
* The ACT LED flickers when there is MIDI activity. It also blinks slowly in midi lockout mode (see below)
* The eight LEDs A-G show the status of the switcher outputs. The LED is on when the output is switched, and the LED brightness shows the PWM duty.

The Button

The button on the MIDI switcher has the following functions

* A short press will reset all the outputs to their power-on state
* A long hold will cause the switcher to go into midi-lockout mode. The activity LED will regularly switch on and off and all MIDI input will be ignored. This mode is useful if you have multiple switchers connected to a common MIDI input and only want to access one of them at a time (e.g. for sending a configuration sysex)
* Holding the button at power up causes the switcher to enter bootloader mode, which allows new firmware to be installed over MIDI. Do not do this if you simply want to load a configuration patch!

PWR SEL jumper

The MIDI switcher has two power circuits

* The microprocessor and associated electronics run from the 2.1mm socket on the back of the board. This socket can accept a 7-30V DC (positive centre) supply.
* The devices (or “loads”) being switched are powered from the POWER and GROUND terminal blocks on the front of the board (each of these terminals has two identical screw connectors, either or both of which can be used).

When the PWR SEL jumper is in the top position (closest to back of board, the default position), these two power circuits are connected together. This enables both the circuits to run from the same supply, via the socket or the terminal blocks.

When PWR SEL is in the top position please note that

* The power socket should only be used if the current needed by the loads is 2 amps or less. For higher currents the terminal blocks should be used.
* Do not exceed a 30V DC supply

When the PWR SEL jumper is in the bottom position the two power circuits are disconnected from each other. This means that a separate power supply needs to be plugged into the socket to power the microprocessor.

This scenario allows voltages higher than 30V to be switched. The transistors are rated up to 100V DC.

Gamma Correction

PWM can be used effectively to control the brightness of LEDs and light bulbs, however our eyes tend to become less sensitive to changes in brightness as a light source becomes brighter. As a light is steadily faded up, the brightness seems to increase a lot at the beginning of the range, but all levels at the top part of the range seem to look the same, which kind of crams all the useful brightness values at the bottom of the range.

The MIDI switcher has a way to scale brightness to make best use of the range, by using a technique called “gamma correction”. Since the issue is specific to our vision, this is only really useful when fading light sources such as LEDs and filament bulbs and is not intended to be used when controlling power to other types of device.

Default Values

Each output has several settings for pulse shaping and modulation. However, in many cases these settings for all of the outputs may be the same with only the trigger condition being different (e.g. a different MIDI note for each output). To help apply the same pulse shaping to every output, they all share a set of “default” values until you want to apply specific settings to an output.

Likewise, there is a default MIDI channel defined, where the switcher listens for incoming MIDI note and controller messages for triggers.

There is a separate MIDI channel defined for program change messages, All program change messages acted upon by the switcher must be received on the same MIDI channel, which can be the same channel as that used by trigger messages if desired

Pulse shaping

When one of the MIDI switcher output ports is triggered (for example by a MIDI note “on” message), the negative output of that port becomes connected to the negative part of the power supply via an electronic switch, called a transistor.

This allows a current to flow through a connected device (for example a light bulb) from its positive connection to the power supply, via the switcher to the negative power supply and power it up (turn on the light bulb)

When the trigger ends (for example by a MIDI note “off” message). The transistor turns off and the current can no longer flow. Therefore a “pulse” of power flows through our connected device…



This pulse is “on” for a specific period of time (its “duration”) and the length of the pulse will have an effect on the response from the device. For example, a longer pulse to a light bulb will look brighter and a longer pulse to a solenoid will make it “hit” harder.

The MIDI switcher lets us control the total length of the pulse in special ways, such as basing it on the velocity of the MIDI note, so that we can achieve “velocity sensitivity” or a control over the amount of power sent to the connected device.

A second way to control power is via something called “pulse width modulation” (PWM). In a PWM scenario we actually turn the power on and off very quickly, so quickly in fact that that we cannot directly see flickering, but instead the bulb behaves like it is getting less power during the trigger period and looks dimmer.



In the above picture

Switcher outputs E/F/G/H use a special feature of the microprocessor to achieve stable PWM signals at about 1kHz. On outputs A/B/C/D the PWM is emulated in hardware and runs at a lower carrier frequency (about 200Hz) and can be less stable when the switcher is busy (e.g. processing a lot of MIDI). Therefore outputs E-H should be used if possible where PWM stability is important (such as when driving LEDs, where instability can result in flickering of the LEDs)

**Sustain**

In this mode, the length of the switching pulse is controlled only by the length of the trigger condition and the hold time is not used.

|  |  |
| --- | --- |
|  |  |

**Hold**

In this mode, length of the switching pulse is controlled only by the hold time and the length of the trigger condition is not used. The trigger condition might last a longer or shorter period than the hold time but the output is always switched on for the hold time

|  |  |
| --- | --- |
|  |  |

**Hold->Sus(tain)**

In this mode, length of the switching pulse is controlled first by the hold time and then by the remaining length of the trigger condition. The output is therefore switched for the longest of either the trigger condition or the hold time.

|  |  |
| --- | --- |
|  |  |

**Sus(tain)->Hold**

In this mode, length of the switching pulse is controlled first by the trigger condition and then by the hold time. The output is therefore switched for the total of the trigger condition time and the hold time.

|  |  |
| --- | --- |
|  |  |

**Release**

In this mode, length of the switching pulse is controlled only by the hold time and the length of the trigger condition is not used, so it is similar to Hold mode. However, during the hold time, the output duty is reduced linearly down to zero.

|  |  |
| --- | --- |
|  |  |

**Sus(tain)->Rel(ease)**

In this mode, length of the switching pulse is controlled first by the trigger condition and then by the hold time, so it is similar to the Sustain->Hold mode. However, during the hold time, the output duty is reduced linearly down to zero.

|  |  |
| --- | --- |
|  |  |

The release modes are most useful with light bulbs and LEDs where some kind of “fade” is desirable before the output switches off.

**Modulation**

The hold time and the maximum PWM duty for an output can each be modulated by either the velocity of the last MIDI note which triggered the output, or by a MIDI continuous controller (CC)

**Low Power Scenario**

Here all power is being received via the 2.1mm power socket. The PWR SEL jumper is in the top position so that power is shared with the switcher outputs.

Each load is connected between the (+) and (–) terminals for the appropriate switcher output (here a single solenoid on output A).

Note that the (+) terminal is always “live” (connected to the input power supply positive) and the (-) terminal is connected to power supply ground only when the switch output is active.

The 2.1mm socket can take a supply up to 30V and a continuous current of 2A. If the current requirements is higher than this, the terminals at the front of the board must be used (see next section)



**High Current Scenarios**

The second scenario is suitable for loads which draw higher current and where the power supply does not exceed 30V. In this scenario the PWR SEL jumper remains in the top position so that the microprocessor on the board can be powered from the front power terminals.

Note that the POWER terminal block has two identical (+) connections and the GROUND terminal block has two identical (-) connections.



Alternatively the “high side” of each load can be powered directly without going via the board, which may simplify wiring in some situations, since only one wire needs to run from each load to the board. This might also help in high current situations since only the current to ground is flowing through the board.



In situations where the supply voltage to the loads exceeds 30V, the loads must be powered separately from the board. **The PWR SEL jumper is moved to the lower position** and a supply between 7-30V is attached to the 2.1mm power socket



The loads can then be supplied by a voltage as high as 100V DC and each switching transistor is rated to 8A, however the maximum power (volts x amps) should not exceed 20W per channel.

If preferred (+) side of the power supply can be connected to the POWER terminal at the left side of the board (**the PWR SEL jumper must be in the lower position**)

Working with Multiple Switchers

The MIDI Switcher has eight outputs available, but in some situations you might need more outputs. In these cases you can use multiple MIDI switcher modules, each configured to listen out for its own triggers, then you can send the same MIDI input to each of the switchers

If you have only two modules you can have them share a single MIDI input and power supply as shown below.



The important factor is that both modules have a common ground connection (shown in green) and that there is a connection (shown in yellow) between the **right hand pin** of the MIDI bus connector of the “sending” board to the **left hand pin** on the connector of the “receiving” board.