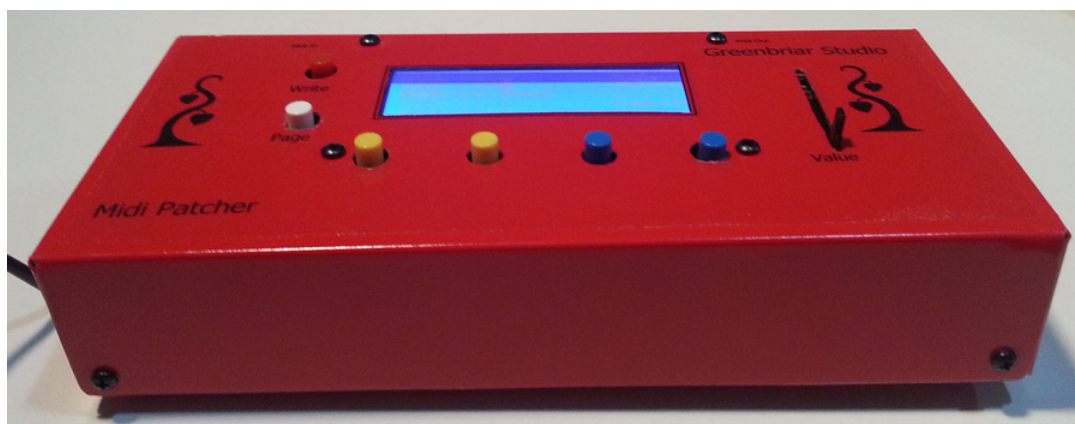


# Greenbriar Studio Midi Patcher and Assigner MK2



Old model Midi Patcher (Retired)



Old Model Midi Assigner  
New Model has Two Midi Ins and longer case



New Model Midi Assigner MK2  
Assembled and Tested version  
New Case currently being Developed

# Midi Patcher and Assigner

The basic functions of the Midi Patcher hardware are to make patch selections, run the arpeggiator, edit all settings of the patches, control the communication to the pc based remote editor and to support updates to the onboard firmware, all through the midi ports, buttons, slider and LCD display. Plus the new MK2 version includes a second Midi In port with a built in midi merger.

## Basic Functionality

The Midi Patcher project was started to provide two main functions - to simplify using multiple vintage synths with various sequencers (my favorite still being the Roland MC-50 MKII - the last sequencer to have the TR style drum pattern system built in) and to allow me to play multiple synths at the same time while I'm writing and playing.

Each of the 1792 patches (14 banks of 128) has 16 tracks or channels that can be played from any midi in to any midi out. Each track also defines a key range and optional high and low bank and program change for that track. The bank and program change commands are issued when that patch is loaded. The point is that I can set up program changes for up to sixteen synths to use for recording a song and not have to keep putting in all those program changes into the sequencer over and over again. I just add the one program change for the Midi Patcher patch and it issues all the others.

Patch change note - since I lose track of which notes were going where when you change patches, the Patcher sends an all notes off to all channels when you change patches. So if you ever get hung notes just change patch to shut them off. The only synth, so far, that does not respond to the all notes off midi message is the Encore updated Jupiter 8.

Although the default new patch has the 16 midi channels preassigned to the 16 tracks, the midi channels are not fixed, you can set each track to any channel. So you can make a split by putting the same input channel on multiple tracks going out to different channels, each with a different key range. Or you can make layers by running the same input channel to multiple output channels with no key range restrictions. Any combination of 16 routings that you like. This is like a Korg Combi or Roland Live Set, but done completely with external synths.

I make some patches for recording, where I'm mostly using the patch to hold all my program changes for a particular song I'm working on. And I make some for live playing, where in addition to making my program changes, I am using one keyboard to play multiple synths, which is how I like to have them setup when I'm working on a new song.

You will pick one channel to be the units command channel. Bank and program changes on this channel are used by the Midi Patcher to select its patches, which then executes the program change commands in the patch. Bank and program changes on any other channel are passed on through, so you can still use external program changes on any other channel. This command channel is also used to communicate, via sysex, with the pc or mac based editor. See editor manual for controlling the Midi Patcher from the pc editor. Everything, except for the arpeggiator, can be controlled from the editor.

The patches function by, first sending out any defined bank and program messages when the patch is se-

lected and then while the patch is active, routing incoming midi messages to all outputs that are setup to receive a particular input channel.

## Arpeggiator

In addition to the patch functionality, there is also an arpeggiator. My first synth, a Juno 6, had a wonderful arpeggiator. When I bought another Juno much later, I bought a Juno 106 as it has patch storage, and most importantly, midi. But it lost the arpeggiator. Some of my vintage synths have one, some don't. So I wanted to be able to run an arpeggiator with any of my synths to then record to my sequencer. And some older synths, while they have an arpeggiator, won't send the arpeggiator notes out Midi Out so they can be recorded. This fixes that issue, as well.

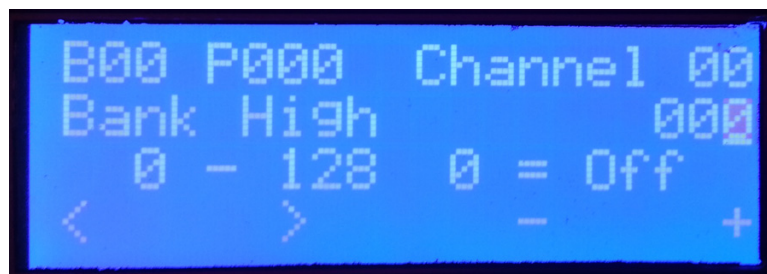
The arpeggiator has up, down, up/down, down/up and as played note directions, 1, 2, 3 or 4 octave range, rate of 1-250 BPM or external Midi Clock rate control and three modes - Play (plays notes that are currently held down), Hold (holds notes after release until new note is played - as long as one key is held down, you can keep adding notes. Once all are released, the next note will start a new note set) and Follow. Follow allows you to transpose a Hold mode running sequence. Use Hold to get an arpeggio running, then switch to Follow mode and any key you press will transpose the running arpeggio. The transpose amount is from your first note, in an as played direction, or lowest note in the others. Up to 32 notes may be in an arpeggio pattern. The large number of notes is really for using a long Hold pattern, without multiple octaves, that can also be used with the Follow transpose function. But it still works with Octave greater than one also.

Other settings for the arpeggiator are - midi input channel, low and high key range (the arpeggiator can play notes outside of this range, but notes outside of this range will not control it, they will go on to wherever you've defined for other parts of that midi channel to go). So part of the keyboard can control the arpeggio while you use other parts to play other synths. Also, you can set to use a fixed velocity or as played, define a note to use as a rest note, the note value: 1/16 - whole notes and the gate (percent of note for the note to be on). All of these values can be saved as the power on default.

## Local Controls

While the Midi Patcher has an pc based editor which is quicker for creating new patches and lets you store offline backup copies of your patches, everything that you can do with the editor can also be done directly on the box, plus the arpeggiator, which has no editor equivalent. While I use the box controls mostly for patch selection and the arpeggiator, I also use them to make quick patch changes. The physical controls on the box let you create and edit all settings of the patches, arpeggiator and global settings and save them.

Since the box itself has a small screen, instead of a big pc screen, editing is organized similarly to the



way all my 80's synths are organized - multiple pages of settings that you can step through. Screen pages are organized as four vertical pages, each with multiple horizontal settings.

The LCD shows four lines of text. On most pages, the Global settings and Arpeggiator being the exceptions, you will see the current bank and patch number top left and the bottom line shows what the four buttons below the screen will do. < and > move between the horizontal settings pages and + and - increase and decrease the setting value by one. To speed up changes, there is also a slider to the right of the LCD. This can quickly make big changes to any value. With the exception of the main arpeggiator screen, the slider and + and - buttons always affect the same thing.

The other two buttons, to the left of the LCD, are the red Write button at the top and the Page button beneath it. The Page buttons rotates through the Patch settings, Channel settings, Arpeggiator Run, Arpeggiator settings and Global settings pages. The Write button, will write the patch back to the flash memory when you are on any Patch or Channel page, write the the Arpeggiator settings to memory if you are on one of the Arpeggiator pages and has no effect if you are on a global page as those settings save automatically if you change them.

### Making Patch Selections

The Midi Patcher boots up on the Patch page, Patch setting. The word Patch top right shows you are in the Patch pages. You will see the cursor on the Patch number. The cursor shows you what the slider and + and - keys will change. On startup, the Bank and Patch values from the Global settings will be loaded and that patch will be loaded and the midi command channel is also set from the saved Global settings.

To change a patch, use the slider and/or + and - buttons to make a selection. You will see the 16 character patch name on line 2. To select a patch from another bank, use the < button to move from patch to bank select, select a bank, use > to go back to patch select and then select a patch. NOTE - changing the bank will not load a patch. A new patch will not be loaded until you change the patch number.

The Patch section only has three pages - Bank select, Patch select and Patch rename. From Patch select use > to go to Patch rename. Entering names on a LCD is always a pain, but it can be done. Use + and - and the slider to change the letter under the cursor, and < and > to move the cursor back and forth. You exit name entry mode by using > to run over 16 characters, < to run back off the beginning, or Page to leave the Patch section. Remember to use Write before changing to another patch.

### Editing Patches

Editing Patches is done from the Channel pages. From the Patch pages, hit the Page button to switch to the Channel pages. (Note - when you use the Page button to switch between page groups, which setting page you were on in that page group is remembered. It does not always take you back to the first item.)

Each editable setting appears on its own page. In order, they are - Midi In : possible values 1 - 16, Midi Out : 1 - 16, Key Low : 0 - 127, Key High : 0 - 127, Program : 0 - 128, where 0 means off or none, Bank High : 0 - 128 - OBS - OBM - OB1 (see Oberheim note below), where 0 means off or none, and Bank Low : 0 - 128, where 0 means off or none. (See assigner section for details on the voice assigner settings.)

Multiple patch channels can refer to the same Midi channel as either Midi In or Midi out. To make a simple Dual or Layer type patch, have more than one channel use the same Midi In and different Midi Outs. To make a split, set multiple channels with the same Midi In, different Midi Outs and a different key range for each

channel. You can have a single Midi In go to as many as 16 Midi Outs in one patch.

You can make changes to as many channels as you like without having to hit the Save button. You can go to any page, including the Globals and Arpeggiator pages, and make any changes EXCEPT change the patch number. Changing the patch number will load a new patch and your changes will be lost. So hit the Save button before changing patches.

## Oberheim Additions

With the 1.02 versions of the firmware, one of the additions is a set of special Bank commands for certain Oberheim synths. These are the Xpander, Matrix 12, Matrix 6, Matrix 6R and Matrix 1000. The Matrix 1000 has 10 banks of 100 sounds and the others have 100 single patches and 100 multi (Xpander and Matrix 12) or split (Matrix 6 and Matrix 6R) patches. Instead of the later convention of using bank high and low controller messages, the Oberheim synths use special Oberheim specific sysex commands to switch between these bank sets. So I have added to the bank high values three special values, that when selected will transmit the special sysex codes needed instead of the bank high controller command. They are OBS - for switch to Oberheim Single mode, OBM - for switch to Oberheim Multi or Split mode and OB1 - for Oberheim 1000 bank change. With all of these you also set the program change to the 1-100 value of the patch you want. (I send 1 less than selected value.) You can set bank low and I will send it, but the Oberheims will ignore it.

## Note on Bank and Program Change values

I allow values for these to run from 0 to 128 (plus the three special values for Oberheim bank high) so that I can use 0 to mean off or none. Also the Roland standard has been to list them as 1 to 128. I always send 1 less than the value you select (just like Midi Channels are selected as 1 to 16, but are actually transmitted as 0 to 15). You will find some synths, such as Rolands, will give you the numbers you need to use starting as 1, while others will start at 0, like the Oberheims. If the values you have are zero based, just add 1 before entering it. It's easy to tell which you have. Just enter the number you want and try it. It will either be correct on the synth or come up off by one. Change as needed. Some older synths, of course, use the 1-1 to 8-8 numbering where 1-1 is 1 and 8-8 is 64. You can make a chart (or grab one online) that shows the patch numbering for these. Like the Juno 106 that uses group A 1-1 as 1 all the way up to group B 8-8 as 128. No bank commands used, just program change.

## Running the Arpeggiator

To make it easier to run, the Arpeggiator first page is set up differently from all the other pages. Instead of a single value that can be changed, it has five, so that all major functions needed to play the Arpeggiator can be accessed from a single page while you play. Each of the four buttons below the LCD toggles through the following settings, from left to right - On and Off, Mode, Octaves and Direction. The slider controls the BPM rate. To go to the next Arpeggiator page, you use the Page button again. From there, the < and > buttons work as they do elsewhere and step you through the remaining settings, the first being Rate where you have both the slider and the + and - buttons to make detailed changes.

This means the Page button cycles through - Patch settings pages, Channel settings pages, Arp run page, Arp settings pages, Global settings pages and then back to the Patch settings pages.

The first switch on the Arp run page is On or Off. This is also saved with the other arp values and is reloaded at system startup.

Arp Octave is only set from the Arp run page and can be set from 1 to 4.

Arp Direction is also only set from the Arp run page and can be Up, Down, Up then Down, Down then Up or Play (in order played). Note that Up/Down and Down/Up do not repeat the top and bottom notes as they play, so if you hold three keys, an Up 2 octave arp plays 6 notes, but an Up/Down 2 octave arp plays 10 notes, not 12.

Arp Mode is also only set from the Arp run page and had three values - Play, Hold and Follow. Play means that only notes currently held down are arpeggiated. When you release the key they stop playing. Hold allows you to play one or more notes, release the keys and they will continue to play. As long as one note is held down, you can continue to add notes (up to 32). Once you release all notes, the next note you play will erase the previous note list and start a new one, so once you release them all, you can't add any more. Follow is a special type of hold. If you start an arpeggio with hold and then select follow as the mode, the arpeggio will keep playing, but the next note you play will, instead of starting a new note list, will set a transpose amount that will be in effect when the pattern starts again from the beginning. So you can hit a note any time the sequence is playing and it will not be in effect until the end of the current arp cycle. This lets you move the arpeggio to different keys with a single key press. The transpose amount is the difference between the first note, if direction is as played, or the lowest note, if the direction is any of the up or down directions.

The other arp settings are individual pages where you have access to the slider and + and - buttons to make changes. Rate is the first one and is the only one that is on the run page and also on an individual page. A rate of 0 enables external midi clock timing, while 1 - 250 will give a bpm rate running off of the internal timer. Note that the arp will not play if rate is set to 0 and there is no midi clock coming in.

The Arp rest note allows you to specify a note that can be used in a arp sequence that will not be played, allowing you to have rests in your sequence. I set mine to the lowest Db on my keyboard.

An arp velocity of 0 means to use the velocity that you generated when you played each note, a greater than 0 value will be used as the fixed velocity for all notes.

Arp Midi In is which midi channel will drive the arpeggiator. The arpeggiated notes will then be sent out to any channel in the patch that has the same Midi In defined. So the arpeggiator will only respond to notes on a single incoming channel, but can output to multiple channels at the same time. It will work with your dual and split channels you have in the patch.

Arp Key Low and Key High set the range of which notes will be added to the arp sequence. They do not limit the range selected by the octave setting. So you could have, for example, the key range set for only below middle C (60). Only notes played below middle C will be arpeggiated, but if the octave range is more than 1, they can play well above middle C. This key range, a split function really, allows you to use part of the keyboard to run the arpeggio and the rest of the keyboard to run one or more other synths. They can be the same synth the arpeggio is playing, or different ones. Only one arpeggio running at a time is currently supported.

Arp Beat is the note value for each note to be played. It can be from 1/16 note to a whole note. Midi clock is defined as 24 clock ticks per quarter note, so the internal timer is set up the same.

Arp Gate is the percent of time the note is on. So for a quarter note, 50 percent on would mean the note is on for 12 clock ticks before it is turned off. Note that with 1/16 notes, that are played every 6 clock ticks, gate is not a very fine adjustment, since it has to be 1 to 5. To work correctly with midi clock, the internal timer use the same tick timing. The Gate value is 1 to 100 percent, not clock ticks.

Arp sync - if you are running in Midi Clock timing mode (Arp Rate = 0), the clock counter will reset to

zero on the receipt of Midi Start so that it will be synced to the Midi clock (ie - it will be 24 clock messages until the start of the next note). But if you don't have a start command, generally only sent on sequencer playback, not record, the first note you play will set the internal clock counter to zero, so the arpeggiator will play on the beat with your recorded tracks. This first note auto reset is blocked if a Start command was received and used to sync the clock count.

## Global Startup Parameters

The global parameters do not use the save button, instead they are always saved any time you make a change. They are - startup bank, startup patch, Midi control channel, which is used to receive bank high and program change messages to switch patches, and to receive and send the sysex messages between the hardware unit and the computer based software editor. There are also three filter settings, allowing you to block certain midi messages to lower the amount of data on the midi bus. They are Allow Sensing, to allow the auto sensing real time command to pass through, AllowAftertouch, to allow channel and polyphonic aftertouch to pass through, and Allow Clock, which allows midi clock, start, stop and continue real time messages to pass through. With this turned off, my drum machines, etc that would start playing sequences, don't, which I usually want as I'm playing them as sound engines from data recorded in my sequencer. I keep drum patterns recorded in the sequencer for each song, so I don't have to load any sequences in the drum machines for each song.

Important note with Midi Clock. If you block sending clock messages through, this does NOT impact using midi clock for the arpeggiator. If midi clock is received and the arp rate is set to 0, the clock messages will still drive the arpeggiator, they are just not sent on out through Midi Out.

## Remote Editor

Also provided with the Midi Patcher is a PC based editor, for Windows and Mac, that allows you to see all of the settings for an entire 16 channel patch at once, upload and download single patches and entire banks and save banks to and from your computer for backup. All settings of the midi patcher can be accessed and edited from the remote editor except for the arpeggiator. The editors are available to download from the product webpage along with the manuals.

## Software Updates

The firmware is updated by sending a special Midi sysex file to the Midi In A port (left most port). The PC editors have a built in function to send update files, or you can use any sysex program, like C6, that allows you to set a wait after each sysex command. Set the delay to 300 milliseconds. This is to allow time for the flash to write between sysex commands. The most current update is available on the product webpage, along with the manuals and PC and Mac editors.

When you are doing an update, it is important to directly connect your computer's Midi Out to the unit's Midi In, with no other Midi devices connected. The bootloader is designed to be as simple and small as possible since it is not easily changeable. It expects to only receive the sysex firmware data and, while it will ignore other midi and sysex messages, it will error if Midi real time messages pop up inside of the program sysex, changing the data length. You will also get an error if you send an update that is for a different product (Midi Patcher update to a Midi Assigner box, for example). This will generate an error and the old firmware will not be overwritten.

If an error occurs, you can just reload. You can not hurt the bootloader.

The remote editor Sysex Update function is just a general sysex send function with a 300 millisecond



delay built in. So you can use it to send patch dumps back to your synths as well.

## Voice Assigner Functions

While working on the Midi Patcher I realized that it could be a great tool for playing multiple 4 voice Roland boutique synths or multiple older 6 voice synths at once with the addition of voice assign software. Having written a voice assigner for my multiple output homemade synth a while back, I thought it would be a great addition to the project.

The voice assigner additions allow you to specify a pair of output synths per channel instead of one and specify how many voices to play on each one and how to distribute the notes. It allows you to specify from 1 to 8 voices per synth. (If you have a synth with more than 8 voices, which is likely to be 12 or more, you don't need the voice sharing function. You'll probably be using it as its own layer.) Since this doubling of the outputs would quickly eat up available midi channels, I decided to add a second midi out port to the box. And as the patches were much larger than the ones for the base Midi Patcher, I added more memory to the Assigner version. The second midi outport can be used for any output, not just the second synth of a shared pair.

Due to its larger patch size and even with the memory being doubled from the Midi Patcher, the Midi Assigner holds fewer patches. It holds 1792 patches in 14 banks of 128. The new MK2 version has the same memory capacity.

New settings were added to each channel to allow for the second midi out and the voice assign function.

### New Channel Settings

There is now a MidiOutA and a MidiOutB along with matching MidiPortA and MidiPortB. MidiOutA is the same as the Midi Patcher MidiOut with the addition of the MidiPortA that allows you to send the output to either midi out port. The choices being A or B. The MidiOutB and MidiPortB are only used for the second synth of a voice assign pair, so for a non voice assign channel you will only use MidiOutA and MidiPortA. The use of midi port B is not required. You can set all outputs to midi out A, which is how the blank default patches are set up, and use different midi channels for the two synths in a voice assign pair. Midi out B is just an available option if you wish or need to use it.

There is only a single Key Low and Key High setting. If two synths are being played as one, they have to have the same key range.

But Bank High, Bank Low and Program Change now have both A and B versions, in case you need different selections for each of the two output synths.

In addition to the A and B versions of midi and patch selection settings, there are four new ones. Assign Patch, which is just on or off. Is this channel using the assign voice function? VoicesA and VoicesB, which are how many voices to use for each synth. The numbers do not have to be the same. And the Mode which can currently be Dual - send the same notes to each synth, Overflow - which will use the voices assigned to A first and then if all voices in A are used, send additional notes to B. In this case, once all voices are used, no additional notes will be played until some are released. Or the mode can be Alternate, which assigns the first note to A, the next to B and so forth. In this mode, if all voices are used and another note comes in, the oldest note being played on either synth will be stolen and used for the new note.

There are no changes in the Global settings or Arpeggiator settings. The only Arpeggiator change is that

the Midi Assigner version will work with the new midi out B port and can work through the dual synth voice assigner mode.

## Construction Notes

With the MK2 version, I have moved to the Atmega 1284 and added it to the pcb. I use it one for two reasons - one, it is the largest atmel processor that can be bought in DIP and not surface mount format, (I had never used surface mount components before, as I learned to build pcbs in the 80's) and it has the largest ram of all their microprocessors (twice the 2560).

I thought the Arduino editor would make programming and especially debugging faster. Yes and no, the bootloader could not be done in the Arduino system at all, I had to write it in Atmel Studio, the platform is rather buggy, generating random weird compile errors, and some libraries had strange bugs. The biggest part of my 1.02 rewrite was to throw out the Arduino libs for the serial ports and midi and write my own. It now runs faster and correctly handles the interrupting real time midi messages. I was only able to use the library the LCD manufacturer provided and the I2C eeprom lib that writes to the memory I added. Neither of which are used in the runtime path of this project. The execution of the hardware ports, midi, patch and voice system is all in my code.

The new 1.04 version is back on Atmel Studio so no extra Arduino code is in the project.

The nice new 2Mb memory I wanted to use was only available in a soic surface mount format, so I finally had to learn how to hand solder surface mount components. I only fried one before getting the hang of it! But because it has surface mount components, I'm not planning at this time to offer kits.

## Metal Box

I wanted a sturdy metal, not plastic box for this project, but as soon as I got a quote for a custom metal box, I knew I had to build my own. I have a metal and wood shop at home, so I bought a small bending brake to make a more regular looking box (I normally do raising and other more free hand metal work) and some .032 food grade aluminum and started developing box patterns.

So these units are made at home and entirely by hand. The only part I send out are the pcbs which are made by a commercial pcb fab.

So my boxes are not as pretty as commercial boxes, but they are all metal and sturdy. Boards are mounted with metal bolts and screws on flexible rubber spaces to allow for some shock absorbtion. Hopefully they'll get nicer looking as I go and get more practice.

A new metal box is being designed for the MK2 version. Plus an assembled and tested version is now being sold without a box to reduce the cost.

## Known Issue Remaining

It is possible to hang notes by hitting as many keys as possible with both hands, in spite of my large input buffer, but it only occurs on certain synths. Still working on this issue, but it is why I added the all notes off on patch change as an easy to use midi panic.

Version 1.02 of both the Midi Patcher and Midi Assigner firmware added:

- Arpeggiator

- New from scratch serial port and Midi implementation. The Midi library used for initial development and testing seemed to have issues with the combination of running status and real time messages. Real time messages can come in the middle of any other Midi message and not all code seems to handle them correctly. After experiencing random dropped and hung notes, I wrote a new midi implementation based on the Midi code I had done earlier for my synth project. I also added the ability to support multiple output ports within a single instance of my Midi class. Other libs required multiple instances to get multiple outputs, which meant I also ended up with multiple large wasteful input buffers. So the new code was designed to support just the hardware needed for these projects.

- Oberheim special Bank codes that generate the sysex required for switching Matrix 1000 banks and between Single and Multi/Split modes for the Xpander, Matrix 12, Matrix 6 and Matrix 6R. So for these synths, using the special Bank High code and then a normal program change will now let you select any patch in these synths.

- This manual was updated to include these new features.

Version 1.04 of the new combined Midi Patcher and Assigner firmware added:

- Second Midi in port support with full midi merge function.

NOTE - the firmware update bootloader only reads from Midi In A (left most midi jack). Otherwise, it makes no difference which Midi In port you use. The two inputs are merged together. Sysex is still limited to 1024 bytes per message.

- Changed to new microprocessor and switched to Atmel Studio for development.

- New kit version.

- New Assembled and tested version with case.

- New Initialize memory command to allow for kit builds. To reinit all patch memory - Hold down Page button while turning unit on. You will be asked if you want to reinit all patches. Yes or No to complete. If Yes, it will take several minutes. Needed when you build from a kit. Units bought Assembled or Complete with case are already initialized.