Back to main page...

MIDIbox SEQ V4

User Manual --- Hardware Options

Hardware Variants

We mainly differ between a MIDIbox SEQ V4 which is based on Wilba's frontpanel, and the "traditional" variant which is based on the MIDIbox SEQ V3 hardware with a replaced core module. Both are running with the same firmware, so that you won't miss any software feature!

Wilba's frontpanel greatly reduces the complexity of the hardware, and is therefore the prefered choice. The traditional variant is still interesting if you want to create a fully customized version of the sequencer, with a different layout and dedicated buttons/LEDs.

Premade PCBs for both variants can be purchased at SmashTV's MIDIbox Shop. He also provides kits and some other parts (like rotary encoders), but please note that the lead time for a frontpanel kit is still very long (several weeks/months!) so that it makes sense to buy the frontpanel PCB only, and to source the remaining parts from an electronic shop like Reichelt or Mouser! Alternative electronic shops can be found in the Wiki

Here a list of components which are required to build a MIDIbox SEQ V4 based on Wilba's frontpanel:

Part	Description	Price
MBHP_CORE_STM32F4 MBHP_MIDI_IO	the brain of MIDIbox SEQ V4 + MIDI IO breakout board. Please note that older core modules such as MBHP_CORE_STM32 and MBHP_CORE_LPC17 are supported as well, but the future "MIDIbox SEQ V4 Plus" firmware will require the higher amount of embedded RAM and Flash memory of the MBHP_CORE_STM32F4 module.	PCBs in SmashTV's Shop: ca. 11 EUR
MBHP_CORE_STM32F4 parts	Currently no kit is available for this newer module (but it could become available in future, please check at SmashTV's side). Here the part list: mbhp_core_stm32f4_orderlist.txt Since the STM32F4DISCOVERY board and the SD Card adapter are not available at Reichelt, but (for example) at Mouser, it actually make sense to provide an alternative list for Mouser. If somebody created such a list, please let me know and I will add a link here.	ca. 25 EUR
MBHP_MIDI_IO parts	Currently no kit is available for this newer module (but it could become available in future, please check at SmashTV's side). Here the part list: mbhp_midi_io_orderlist.txt	ca. 6 EUR
SD Card	A SD Card (no adapter required since it's part of the MBHP_CORE_STM32F4 module); size should be around 1GB (but 2GB and 4GB cards will work as well). Note that a SD Card is a *must* when you are using Wilba's frontpanel, because the dedicated button/LED assignments will be loaded from a special MBSEQ_HW.V4 file which has to be stored in the root directory of the card. Without this file the frontpanel won't work!	ca. 3 EUR
Frontpanel PCB	PCB only without parts	available in SmashTV's Shop: ca. 24 EUR
Frontpanel Parts	As listed at this Wiki page	ca. 100 EUR
Aluminium Panel	Can be ordered from Schaeffer Apparatebau, the .fpd file is available at Wilba's page	ca. 120 EUR
LCDs	two 2x40 Displaytech "LCD 402A LED" *Hot tip: such displays are mostly available at EBay for less than 10 EUR!*	ca. 50 EUR
Case	currently no standard case - take a wooden box! ;-) Or search/ask in the MIDIbox forum for options.	ca. 10..100 EUR

USB MIDI

Thanks to the integrated USB peripheral of the MIOS32 cores, USB MIDI is natively supported by the firmware. This simplifies the upload of new firmwares, and speeds up MIDI transfers to your DAW and/or virtual synthesizers up to 100 times and more - see also this forum article.

The sequencer will also run without USB of course. Up to four MIDI IN and MIDI OUT ports are provided by the MBHP_CORE_STM32F4 module in conjunction with two MBHP_MIDI_IO modules for connecting MIDI devices such as synthesizers, drum machines, Fx gear, keyboards, etc.

MBHP_IIC_MIDI modules

Up to 4 MBHP_IIC_MIDI modules can be added if desired (note: with a simple hack in the IIC firmware even 8 modules could be accessed from the sequencer - infos available on request).

Main advantage of multiple MIDI ports is reduced latency (as MIDI events can be sent in parallel), and relaxed MIDI channel handling (for 4 synths you don't need to take care about the MIDI channel at all)

The MIDI IN port is **not** supported, accordingly the "OUT only" version is sufficient.

Following diagram shows the module interconnections.

The modules will be detected by the MIDIbox SEQ application automatically during startup. "Hot plug&play" while the application is running is not supported.

Power Supply

The power consumption of a common MIDIbox SEQ V4 configuration is about 400..1000 mA depending on the used LCDs and LEDs. Especially blue backlights and LEDs can increase the consumption dramatically!

Today it's common practice to supply the MIDIbox SEQ from a "selfpowered" USB Hub like this one from Reichelt. Advantage: it has a power-switch!

Alternatively the core module can be directly supplied from an external PSU (e.g. a wall adapter) connected to J1. Jumper J17 (USB Power Option) has to be disconnected in this case!

SD Card

A SD Card is required as data storage for patterns, songs, grooves, options, the hardware configuration, MIDI files, SysEx dumps and various other files.

The MIDIbox SEQ V4 firmware provides access to the MIOS Filebrowser, which is part of MIOS Studio. With this tool, it's possible to download, upload and edit files. The communication is handled via the USB MIDI protocol, so that active MIDI connections stay active.

Thanks to this useful feature, it's possible to build the SD Card into the case (no need for an external socket).

If you've the choice, buy a SD Card with speed factor 100x or more - it directly affects the access time to files. A speed factor of less than x60 is not sufficient for the realtime requirements (e.g. while changing patterns), the sequencer could sporadically block for a couple of mS which can result into temporal hang-ups.

SD Card size: typically only ca. 5 MB are allocated by MBSEQ_* files. The remaining disk space can be used for storing backups, SysEx dumps and MIDI files, but probably never more than 50..200 MB will be used. Accordingly, a 256MB or 512MB card is already sufficient. Using a 1, 2 or 4 GB card is great but not really required. Sizes > 4 GB are *not* supported by the file system (FAT32), and therefore not recommended.

Pots or Encoders?

MIDIbox SEQ V4 provides an user interface with 16 rotary encoders (endless knobs) + 1 data wheel (the 17th rotary encoder).

Analog pots are not supported. The reason: pot control might sound like "vintage", "old style" and "analog sequencer", and possibly could imply that pots are the best solution for perfect sequences, but this is just a popular fallacy and nothing else! In fact pots are only useful for sequencers which are controlling the analog gear directly, but for a MIDI sequencer you defenitely want to have features like "total recall" and "jumpless switching" between tracks, layers and patterns.

In addition, the menu pages of MIDIbox SEQ V4 rely on rotary encoders more than before. MIDIbox SEQ has a flat menu hierarchy, and this requires that all values can be changed directly - with the 16 rotary encoders.

Inc/Dec Buttons or Datawheel?

For the menu navigation and data entry either Inc/Dec (Up/Down) buttons and/or a datawheel can be used. The datawheel (rotary encoder with big knob) should be the preferred solution since it simplifies the selection of parameters with a wide value range (like BPM).

If you are planning a new frontpanel, it even could make sense to consider both - datawheel **and** Inc/Dec buttons, because buttons are sometimes better for finetuning parameters. The Fwd/Rew buttons can be used for the same purpose by default, but only as long as the Song mode is not active.

Detented or Non-Detented Encoders?

In difference to suggestions for MIOS based MIDI controllers, for MIDIbox SEQ I strongly recomment **detented rotary encoders**! Of course, MIOS provides three types, and all are working. But for a sequencer, where you want to select the notes exactly and not "smoothly", detented encoders are really the preferred ones.

Note that there are three different types of detented rotary encoders (see also this Wiki page). The MBSEQ_HW.V4 file (stored on SD Card) has been preconfigured for "DETENTED3", which is the most commonly used (encoders provided by Voti/SmashTV). If you notice, that your encoders only work in one direction, please change the type to DETENTED2, DETENTED2, DETENTED3 (default), DETENTED4 (works better for some encoders) or DETENTED5 at the bottom of the setup file.

Here a configuration example for the standard layout:

```
##################################################
# Encoder Functions
# SR = 0: encoder disabled
# SR = 1..16: DIN assignment
# Types: NON_DETENTED, DETENTED1, DETENTED2, DETENTED3
##################################################

#              SR  Pin  Type
ENC_DATAWHEEL   6   2   DETENTED3

# the speed value for the datawheel which is used when the "FAST" button is activated:
ENC_DATAWHEEL_FAST_SPEED 3

#        SR  Pin  Type
ENC_GP1   1   6   DETENTED3
ENC_GP2   1   4   DETENTED3
ENC_GP3   1   2   DETENTED3
ENC_GP4   1   0   DETENTED3
ENC_GP5   3   6   DETENTED3
ENC_GP6   3   4   DETENTED3
ENC_GP7   3   2   DETENTED3
ENC_GP8   3   0   DETENTED3
ENC_GP9   4   6   DETENTED3
ENC_GP10  4   4   DETENTED3
ENC_GP11  4   2   DETENTED3
ENC_GP12  4   0   DETENTED3
ENC_GP13  5   6   DETENTED3
ENC_GP14  5   4   DETENTED3
ENC_GP15  5   2   DETENTED3
ENC_GP16  5   0   DETENTED3
```

How many buttons?

This question is only relevant if you want to build a fully customized frontpanel: it's highly recommended to mount at least the buttons which are used in the original V3/V4 design to ensure best ergonomic handling. This means, that you want to use at least 44 buttons!

With MIDIbox SEQ V4, some new button functions have been introduced, which should be considered on new frontpanel layouts. These are:

- **Group Select Button 1/2/3/4**: Four buttons which provide a direct access to the four track groups. In combination with the four track buttons, this allows you to change between the 16 tracks quickly.
  With the MBSEQ V2/V3 frontpanel, the track group has to be selected with the F4 button.
- **Trigger Layer A/B/C:** Three buttons which provide a direct access to the trigger layers.
  With the MBSEQ V2/V3 frontpanel, the trigger layer has to be selected with the F3 button.
- **Step View:** Changes to the Step View selection display.
  With the MBSEQ V2/V3 frontpanel, the step view has to be selected with the F2 button.
- **Utility:** changes to the Utility page, e.g. to access the Copy/Paste/Clear function.
  With the MBSEQ V2/V3 frontpanel, the Utility page has to be selected with the F1 button.

A lot of additional button functions have been added later, mostly based on user requests. A full list can be found in the MBSEQ_HW.V4 template (search for BUTTON_)

For illustration purposes, here a configuration example which assigns the Stop/Pause/Play/Rew buttons to the first DIN, Pin D4..D7, and the Fwd button to the second DIN, pin D0. The Loop and Follow buttons are not assigned, but you could do this adding these buttons to your frontpanel, connecting them to unused DIN pins and define the appr. SR & pin numbers in your own MBSEQ_HW.V4 file:

```
##################################################
# Button assignments to DIN pins
# SR = 0: Button disabled
# SR = 1..23:  directly triggered from DIN pin
# SR = M1..M8: triggered from a 8x8 button matrix
##################################################

#             SR  Pin
```

```
BUTTON_STOP     1    4
BUTTON_PAUSE    1    5
BUTTON_PLAY     1    6
BUTTON_REW      1    7
BUTTON_FWD      2    0
BUTTON_LOOP     0    0
BUTTON_FOLLOW   0    0
```

Note that there is also a MIDI Remote feature available which allows you to trigger button functions with a MIDI keyboard. However, the handling isn't so perfect like with dedicated buttons (especially since the LEDs are missing on a master keyboard), but it works!

It isn't required that the buttons are assigned to the same DIN input pins like predefined in the MBSEQ_HW.V4 file (stored on SD Card). You are able to change the pinning for your (existing) hardware. But if you want to keep it compatible, then use exactly the same pinning like me.

DIN Pinning of hwcfg/standard_v4/MBSEQ_HW.V4

Note: alternatively the MIDIbox SEQ firmware allows to connect up to 64 buttons in a matrix configuration. See "Reduced DIN/DOUT Option" topic below.

How many LEDs?

This question is only relevant if you want to build a fully customized frontpanel: some button functions require a LED as some kind of feedback which is not given by the LCD. For example, the selected Track and Layer is not always displayed by the LCD, also Solo/Fast/All and especially the played (unmuted) steps.

This means that you want to have at least 30 LEDs. One additional LED can be added as beat indicator.

With MIDIbox SEQ V4, some new LED functions have been introduced, which should be considered on new frontpanel layouts. These are:

- **Group Select LED 1/2/3/4**: to display the selected group
- **Trigger Layer A/B/C:** to display the selected trigger layer
- **16 Bicolour GP LEDs (Red/Green)**: improve the visualisation of the current step position

Also LEDs for Play/Stop/Pause are available now. These LEDs have been requested, but if you ask me: they are not required... however, a full list can be found in the MBSEQ_HW.V4 template (search for LED_)

DOUT Pinning of hwcfg/standard_v4/MBSEQ_HW.V4

For illustration purposes, here a configuration example which assigns the Track 1-4 LEDs to DOUT, Pin D0..3. Stop/Pause/Play/Rew buttons to the first DIN, Pin D4..D7, and the Fwd button to the second DIN, pin D0. The Loop and Follow buttons are not assigned, but you could do this adding these buttons to your frontpanel, connecting them to unused DIN pins and define the appr. SR & pin numbers in your own MBSEQ_HW.V4 file:

```
##################################################
# LED assignments to DOUT pins
# SR = 0: LED disabled
# SR = 1..23:  directly forwarded to DOUT pin
# SR = M1..M8: forwarded to a 8x8 LED matrix
#
# Please note: due to historical reasons, pin 0-7
# are mirrored!
# Pin  0 = DOUT Pin D7
# Pin  1 = DOUT Pin D6
# Pin  2 = DOUT Pin D5
# ...
# Pin  7 = DOUT Pin D0
##################################################

#           SR  Pin
LED_TRACK1   1    0
LED_TRACK2   1    1
LED_TRACK3   1    2
LED_TRACK4   1    3
```

Note: alternatively the MIDIbox SEQ firmware allows to connect up to 64 LEDs in a matrix configuration. See "Reduced DIN/DOUT Option" topic below.

Reduced DIN/DOUT Option

For Wilba's frontpanel a feature has been built into the firmware which reduces the number of required DIN/DOUT shift registers. This is achieved by connecting the buttons and LEDs to a 8x8 matrix.

The so called BLM8x8 mode will be activated with:

```
##################################################
# Additional 8x8 BLM as used for Wilba's Frontpanel
```

```
##################################################

# set to 1 to enable 8x8 BLM driver
BLM8X8_ENABLED  1

# to which shift register are the select lines connected?
# Allowed values: 0 to disable, 1..16 to assign shift register
BLM8X8_DOUT_CATHODES_SR  1

# set an inversion mask for the DOUT shift registers if sink drivers (transistors)
# have been added to the cathode lines
BLM8X8_DOUT_CATHODES_INV_MASK 0x00

# to which shift register are the LED anode lines connected?
# Allowed values: 0 to disable, 1..16 to assign shift register
BLM8X8_DOUT_LED_SR      2

# 0: no mapping of 8x8 LEDs
# 1: enable GP LED -> 8x8 matrix mapping for Wilba's MB-SEQ PCB
BLM8X8_DOUT_GP_MAPPING  1

# 8x8 matrix for misc. button functions
BLM8X8_DIN_SR  2
```

A preconfigured MBSEQ_HW.V4 file for this setup can be found in the hwcfg/wilba/MBSEQ_HW.V4 file which is part of the release package. The appr. configuration parameters are starting with BLM8x8_*, and BUTTON_*/LED_* entries are assigned to the M1..M8 'SR' instead of specific DIN/DOUT SR numbers.

Example:

```
#              SR  Pin
BUTTON_STOP    M3   5
BUTTON_PAUSE   M2   4
BUTTON_PLAY    M2   5
BUTTON_REW     M3   4
BUTTON_FWD     M1   4
```

A connection diagram (for those who even want to DIY the frontpanel) is available here:
DIN/DOUT Pinning for hwcfg/wilba/MBSEQ_HW.V4

You will notice a strange order in the button/LED assignments. It's layout driven, but for the end user it doesn't really matter (because they get a preconfigured MBSEQ_HW.V4 file :-).

However, if you are planning to use this approach for your selfmade frontpanel, consider that all assignments can be freely changed, except for the GP Duo-Colour LEDs, which are hardcoded in the firmware!

Simple 4x16 Button/LED matrix

By default the 16 "general purpose" LEDs (which belong to the 16 "general purpose" buttons) display the triggers of the current selected track. If the sequencer is running, they also display the current sequencer position by inverting the appr. step LED.

MIDIbox SEQ supports an option to handle 64 LEDs for all 4*16 steps of a pattern (which consists of 4 tracks). This option requires just 4 additional DOUT pins which are driving the cathodes of the LED rows.

It's even possible to use Duo-Colour LEDs, so that the step pointer gets an individual colour (no inversion of the active step required)

Another hardware option allows to use 64 instead of 16 GP buttons, or, to use the 64 buttons in addition to the 16 GP buttons.

All possible options, and the appr. settings are described in the MBSEQ_HW.V4 file of the release package.

Meanwhile Ilmenator created a PCB for the 4x16 BLM. More informations can be found at this Wiki page.

Giant 16x16+X Button/LED matrix

The most flexible and probably also most preferred BLM solution, as it provides much more entry modes.

PCBs will be created by the community!

More informations, schematics and videos are available in a special manual chapter.

Which display size?

For MIDIbox SEQ V4, **two 2x40 LCDs are required**! The menu pages won't be displayed completely on smaller LCDs, a software solution for 2x16 or

2x20 LCDs is not supported (and very hard to realize)!

A single 4x40 LCD as replacement for the two 2x20 LCDs is possible, but keep in mind, that this requires to arrange the GP buttons and rotary encoders at the top and the bottom of the LCD, so that they are aligned with the 8*5 character segments of the display. Without this alignment, menu item values cannot be edited intuitively.

BPM LED digits

Optionally MIDIbox SEQ can be stuffed with 4 LED digits which display the current BPM value (the 4th digit prints the value after the dot for higher precession - it can be omitted). Note that this only works in master mode, in slave mode SLA will be displayed instead. LED digits with common anode or cathode are supported. They are driven time multiplexed, and have to be connected to two dedicated shift registers like documented in this schematic. The DOUT shift register pinning can be customized in MBSEQ_HW.V4

Configuration example (Segments connected to 3rd DOUT SR, selection lines to 4th DOUT SR):

```
##################################################
# Optional BPM digits
##################################################

# set to 1 or 2 to enable the 3 optional BPM digits
# 0: BPM digits disabled
# 1: BPM digits with common cathode
# 2: BPM digits with common anode
BPM_DIGITS_ENABLED 1

# define the DOUT shift register to which the segments are connected (0=disabled)
BPM_DIGITS_SEGMENTS_SR 3

# define the DOUT SR and pin to which the common pins are connected
# we are counting from right to left
# Example: 140.5 BPM: (COMMON1 = .5, COMMON2=0., COMMON3=4, COMMON4=1)
#                       SR  Pin
BPM_DIGITS_COMMON1_PIN   4   0
BPM_DIGITS_COMMON2_PIN   4   1
BPM_DIGITS_COMMON3_PIN   4   2
BPM_DIGITS_COMMON4_PIN   4   3
```

Dedicated BPM (Tempo) Encoder

An encoder for directly controlling the tempo can be added as well. The DIN shift register connections have to be specified in MBSEQ_HW.V4

Step LED digits

It's also possible to add 3 LED digits to display the current selected step (1..256). Interconnections are similar to the BPM digits, DOUT shift register pinning can be customized in MBSEQ_HW.V4.

Configuration example (Segments connected to 3rd DOUT SR, selection lines to 4th DOUT SR):

```
##################################################
# Optional Step digits
##################################################

# set to 1 or 2 to enable the 3 optional STEP digits
# 0: STEP digits disabled
# 1: STEP digits with common cathode
# 2: STEP digits with common anode
STEP_DIGITS_ENABLED 1

# define the DOUT shift register to which the segments are connected (0=disabled)
STEP_DIGITS_SEGMENTS_SR 3

# define the DOUT SR and pin to which the common pins are connected
# we are counting from right to left
# Example: Step 123: (COMMON1 = 3, COMMON2=2, COMMON3=1)
#                       SR  Pin
STEP_DIGITS_COMMON1_PIN   4   4
STEP_DIGITS_COMMON2_PIN   4   5
STEP_DIGITS_COMMON3_PIN   4   6
```

Track Position Display

For those who are searching for more blink on their box, the Track Position Display, created by Hawkeye and Ilmenator, could be for interest!
Ilmenator created a special PCB which combines BPM, Step and TPD + encoder + four buttons on a small PCB. More informations on this Wiki page.

● CV Outputs

MIDIbox SEQ V4 can control up to 8 analog outputs by connecting an MBHP_AOUT or MBHP_AOUT_NG module, or up to 4 MBHP_AOUT_LC modules to the core.

In addition, 8 digital triggers (gates) and 8 freely configurable clock triggers (which can also be used as Start/Stop) are available on two dedicated DOUT shift registers which have to be configured in the MBSEQ_HW.V4 file (search for CV_GATE_SR1 and CLK_SR).
Note: these outputs are also available at J5 and J10B of the core module, but only at 3V level, and unprotected from the microcontroller IO pins. A 74HC541 based level shifter would be required for a proper output signal to analog synths, but this is also given by the DOUT based solution. Therefore the "direct connection" option is not recommended anymore!

For those who cannot get enough, 64 additional drum triggers are available in the DOUT chain. They can be controlled with individual key numbers and trigger a pulse of 1 mS. The used DOUT shift registers have to be defined in the MBSEQ_HW.V4 file (DOUT_GATE_SR1..8)

Tracks can be assigned to the outputs in the MIDI event menu. It's also possible to route incoming MIDI events to the AOUT/gate triggers in addition (this replaces a CV interface completely). The AOUT channel and trigger routing is described in the "AOUT Port" chapter of the User Manual.

The "old" CV Breakout Port

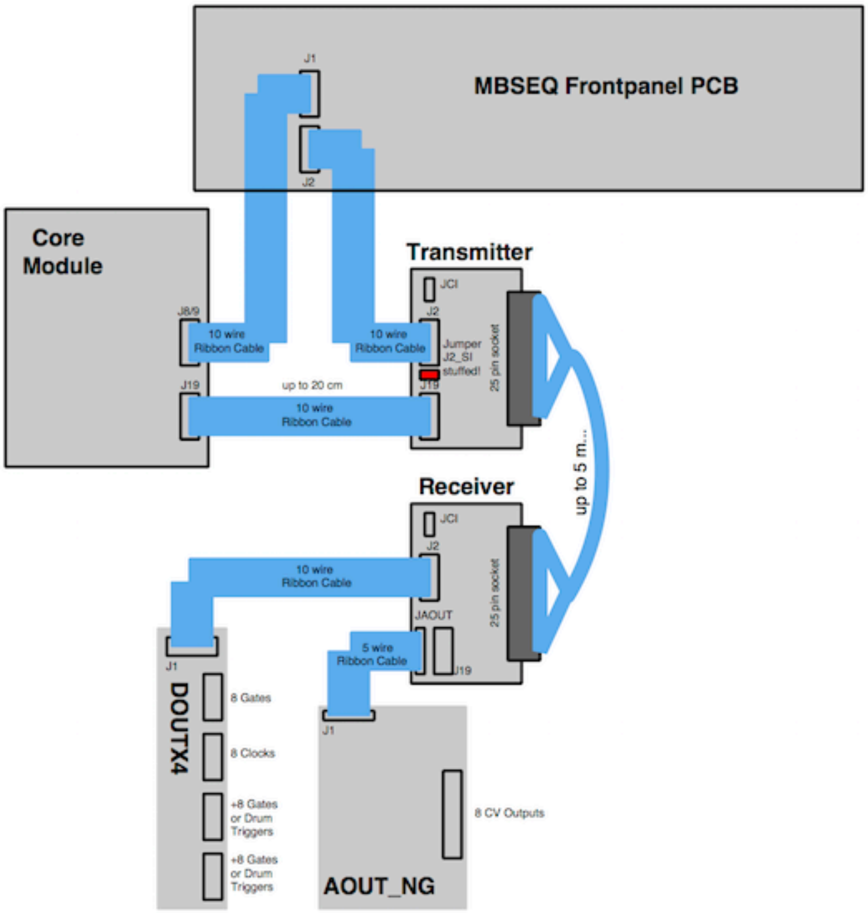
Following schematics show the pinout of the "classic" 25-pin CV Breakout port which was used many years by some users: STM32F1 connections and LPC17 connections.

It provides direct gate and DIN sync outputs at 3V level, AOUT and SRIO interface signals + ground. As mentioned above, the 3V outputs were problematic, an external buffer was required to for level shifting and microcontroller protection.

Another problem was the limited cable length to the AOUT module! Cables with a length of more than 30cm could lead to instable analog CV values (or no value changes at all!). Also SRIO transfers could get instable, which affected all shift registers of the DIN/DOUT chain (also by SRs used by the control surface).

Due to these problems, the breakout port has been revised:

The "new" CV Breakout Port

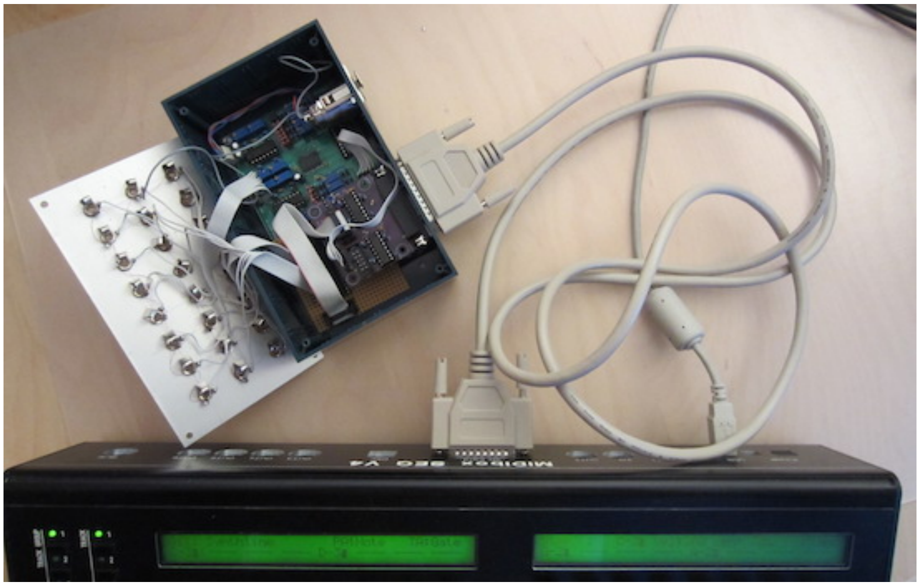


The revised version provides two SPI connections via bipolar (RS422-like) MC3487/MC3486 based line drivers. This extends the cable length to (at least) 2m!

- Transmitter Side (inside MBSEQ case): mbhp_line_driver_transmitter.pdf
- Receiver Side (inside Breakout Box): mbhp_line_driver_receiver.pdf

The line driver can also extends J19 (for AOUT module) + J8/9 (for additional DOUT and DIN modules), so that not only DOUT based gates, drum triggers and clock outputs can be made remotely available, but also for external modules such as the TPD. Two additional pins are reserved for a DIN Sync clock and start/stop input (not implemented yet!).

See the MBHP_LINE_DRIVER page for additional information.



Here a MBSEQ_HW.V4 configuration example which works with the MBSEQ Frontpanel PCB:

```
##################################################
# CV and Gate/Trigger/Sync Setup
##################################################

# the 8 CV gates can be assigned to a shift register (0=off, 1-32: number of shift register):
# - 1st CV Gate available at DOUT SR output D7
# - 2nd CV Gate available at DOUT SR output D6
# - 3rd CV Gate available at DOUT SR output D5
# - ...
# - 8th CV Gate available at DOUT SR output D0
CV_GATE_SR1   3

# and DIN Clock Outputs can be assigned to a shift register as well (0=off, 1-32: number of shift register):
# D7..D0 will output individual clock or start/stop signals which can be configured in the CV configuration page
CLK_SR        4

# additional gate triggers are available on common digital output pins of the
# DOUT shift register chain - they are assigned to AOUT channel #16 (Note C-1, C#1, D-1, ...)
# define the shift registers which should be used here (each provides 8 gates)
# Note that SRs assigned to this function cannot be used as LED outputs (exclusive function)
# Allowed values: 1-32, 0 disables the function, all other values invalid and not allowed
DOUT_GATE_SR1   5
DOUT_GATE_SR2   6
DOUT_GATE_SR3   0
DOUT_GATE_SR4   0
DOUT_GATE_SR5   0
DOUT_GATE_SR6   0
DOUT_GATE_SR7   0
DOUT_GATE_SR8   0
```

The gates are available at the 3rd DOUT shift register, the clocks at the 4th DOUT shift register, and additional gates/drum triggers at the 5th and 7th DOUT shift registers. This works, since Wilba's Frontpanel PCB only uses two DOUT shift registers; a MBHP_DOUTX4 module connected to J2 of the PCB will represent DOUT number 3, 4, 5 and 6.
If more gates are required, just add another MBHP_DOUTX4 module and assign the number 7..10 to DOUT_GATE_SR3..6 accordingly.

Ethernet Interface

An Ethernet port can be optionally added to send and receive OSC messages. It requires a MBHP_ETH module. Details about the possibilities are explained at this page.

Last update: 2017-11-09