MIDI support in piHPSDR
Manual for Users

To use MIDI devices in piHPSDR, you must provide a text file `midi.inp` that resides in the current working dir of piHPSDR (that is, where the "*.props" file and the WDSP wisdom files also reside).

A sample midi.inp file for the popular Behringer "CMD PL-1" MIDI console is provided as an example. The basic rules for the file are

- the file is line-oriented. Each line contains a directive
- in each line, a "#" sign denotes a comment, that is: a line beginning with a "#" is skipped completely, and in all other lines, the "#" and all following characters are skipped.
- There must be at least one line starting with "`DEVICE=`". Everything following the "=" is treated as the device name (without trailing blanks). Only MIDI devices are accepted whose name starts with the given device name. for example, a line reading "`DEVICE=CMD PL`" will accept the "CMD PL-1" MIDI controller.
- Lines not specifying a device must contain one (and only one) of the strings "`KEY=`", "`CTRL=`", or "`PITCH`".
- The lines then can contain additional key words which are as follows

  ```
  ONOFF
  WHEEL
  CHAN=<chan>
  ACTION=<action>
  THR=<thresholds>
  DELAY=<delay>
  ```

  where <chan> is a MIDI channel number (1–16), <action> is a key word which tells which action should be triggered (see below), <delay> is a delay (in milli seconds) which may be applied to WHEELs (see below), and <thresholds> is a list of **twelve** integer values which are used to translate the low-level values reported by a wheel into one of the six cases up/down combined with normal/fast/very fast. All these key words are optional, the default values are

`<chan>=0`	this means AnyChannel, see below
`<action>=NONE`	in this case the whole line has no meaning
`<thr>=128 -1 128 -1 128 -1 128 -1 128 -1 128 -1`	
`<delay>=0`	no delay enforced

MIDI channels

If the CHAN keyword is not there, or a value for <chan> is given that is smaller than 1 or larger than 16, then a MIDI message from any channel triggers the event specified. If the line includes, say, "`CHAN=5`", then only MIDI messages from channel 5 triggers the event specified in the line. The CHAN key word is there for future extensions, if more than one MIDI controller can be used.

KEY events: MIDI push buttons

MIDI KEYs generate NoteOn/NoteOff messages. They are usually generated by push-buttons on the MIDI controller. Normally the action will only be executed when pressing the key, but f the additional key word ONOFF is given, the action will be executed both when pressing and releasing the key.
Example. If the `TUNE` (toggle TUNE state) is associated with a push-button, then in the normal case, the radio will start tuning when pressing the button, and stop tuning when pressing it again at a later time (in between, you may go to your antenna coupler and adjust it). However when the ONOFF key word is given, the radio will start tuning when pressing (and holding) the button, and stop tuning when releasing the button.

CTRL events: MIDI controllers (knobs and wheels)

For music instruments, MIDI controllers are expression pedals etc., they send values in the range 0–127. We distinguish between "knobs" and "wheels". Knobs can be turned left or right until one meets an end point, and send values from 0 to 127 indicating where we are between the left and right extreme. Knobs can typically be used for adjusting AF and microphone level, the TX drive level, etc. So the line

```
CTRL=20 ACTION=RFPOWER
```

will use the MIDI controller with id=20 (0x14) to adjust the TX drive level, which is minimal when the knob is turned fully counter-clockwise and maximal if the knob is turned fully clockwise.

A wheel is a special type of MIDI controller. It has no fully (counter-) clockwise position, but instead can be turned either left or right without ever stopping. While being rotated, it constantly sends MIDI messages encoding the sense (left/right) and possibly the speed of rotation. Wheels can also be used, say, to set the AF volume. The software automatically takes care that if the minimum (maximum) value is reached, further rotation has no effect. Wheels need the additional keyword `THR` to translate controller values into one of the six cases

very fast left
fast left
left
right
fast right
very fast right

If t1, t2, …, t11, t12 are the twelve integer values following "`THR=`", then the algorithm for converting MIDI controller values into one of the six cases is as follows:

t1 ≤ value ≤ t2 ==> very fast left
t3 < value ≤ t4 ==> fast left
t5 < value ≤ t6 ==> left
t7 ≤ value ≤ t8 ==> right
t9 ≤ value ≤ t10 ==> fast right
t11 ≤ value ≤ t12 ==> very fast right

From this it follows that if, for example, the "very fast left" case should never be triggered, choose thresholds such that t2>t1. Such "dummy" threshold values have to be specified if e.g. the wheel is not speed sensitive such that only the "left" and "right" cases can be triggered. This also means that using the default thresholds for a "wheel" an action will never be triggered.

The "standard" use for wheels are VFO knobs, but they can also be used in all cases where knobs can be used. It may happen that when turning a wheel, the MIDI messages are generated at such a fast rate that it is difficult, e.g., to adjust the TX drive level to a desired value. Here the `DELAY` keyword helps. It specifies a delay (in milli seconds) during which further MIDI controller messages are ignored. Specifying, for example `DELAY=100` for a wheel where `ACTION=RFPOWER`, this implies that the TX drive level (which is between 0 and 100) changes at most by 10 units per second if the corresponding wheel is turned.

PITCH events: MIDI PITCH bends

Some MIDI controllers have a pitch controller. There can at most be one pitch controller per MIDI channel, and the MIDI protocol specifies that it can encode values from 0 to 16383. To give an example, the line

```
PITCH ACTION=AFGAIN
```

indicates that the AF volume level should be controlled with the pitch controller. Functionally, a pitch controller is fully equivalent to a knob, since it has a minimum and a maximum value.

List of Actions

The following table documents which MIDI actions are available (in alphabetical order). Some actions require a special case of MIDI event: it makes no sense to control the VFO frequency using a push-button. We will designate by "knob" either a MIDI controller without the "WHEEL" key word or a MIDI pitch controller, by "wheel" a MIDI controller with the "WHEEL" key word and by "key" a MIDI key.

In order not to overload the user interface, we have no action "go to 20m band". Instead, we implement a "band up" and "band down" action. With a push-button, you may cycle through the bands 160m - 80m - 60m etc, while with another push-button you may cycle the other way round. It is also possible to assign both "band-up" and "band-down" to a wheel, then turning the wheel in either direction will cycle through the bands in either direction. This is also possible for filter sizes and modes.
Sometimes the same keyword may have a different meaning if combined with a key or with a knob: for example, the ATT key word combined with a key cycles through the ALEX attenuator settings (0 dB,10 dB, 20 dB, 30 dB), while ATT combined with a knob or with a wheel changes the value of the step-attenuator (0–31 dB in 1 dB steps).

Action key word	Valid elements	Description
AFGAIN	knob, wheel	set AF volume level
AGC	knob, wheel	change AGC value
AGCATTACK	key	cycle through AGC attack (fast/med/slow)
ATT	key	cycle through ALEX attenuator settings
ATT	wheel, knob	set value of step attenuator
BANDDOWN	key, wheel	cycle through bands downwards
BANDUP	key, wheel	cycle through bands upwards
COMPRESS	wheel, knob	set TX compression value
CTUN	key	toggle CTUN mode
FILTERDOWN	key, wheel	cycle through filters downwards
FILTERUP	key, wheel	cycle through filters upwards
LOCK	key	toggle LOCK on/off
MICGAIN	knob, wheel	change microphone gain
MODEDOWN	key, wheel	cycle through modes downwards
MODEUP	key, wheel	cycle through modes upwards
MOX	key	toggle MOX state
NOISEBLANKER	key	cycle through NB settings
NOISEREDUCTION	key	cycle through NR settings
PANHIGH	wheel	change "high" value of current panadapter
PANLOW	wheel	change "low" value of current panadapter
PREAMP	key	cycle through preamp settings [a)]
PURESIGNAL	key	toggle PURESIGNAL enable/disable
RFPOWER	knob, wheel	change TX drive level
RITCLEAR	key	turn off RIT and set RIT value to zero
RITVAL	wheel	change RIT value
SPLIT	key	toggle SPLIT enable/disable
SWAPVFO	key	swap VFOs A and B
TUNE	key	toggle TUNE state
VFO	wheel	change frequency of current VFO
VFOA	wheel	change frequency of VFO A
VFOB	wheel	change frequency of VFO B
VFO_A2B	key	copy frequency from VFO A to VFO B
VFO_B2A	key	copy frequency from VFO B to VFO A
VOX	key	toggle VOX enable/disable
NONE	key, knob, wheel	no action

[a)] preamps only available for CHARLY25 filter board

Appendix A. Sample midi.inp file for a Behringer CMD PL-1 controller

The file is given in a small font to avoid line breaks. Use copy+paste to get this onto you computer.

```
#
# Sample midi.inp file, suitable for a Behringer CMD PL-1 MIDI controller
#
# Note that the Attenuator is implemented twice, as a key and as a wheel
# The key is suitable for radios with a step (ALEX) attenuator, the wheel
# fits best for radios with a programmable attenuator (0-31 dB).
#
DEVICE=CMD PL-1
#
# Big Wheel: main VFO knob
# Note that there is a "KEY" event (id=31) triggered by pushing the VFO wheel.
# Ignore this, because this happens unintentionally most of the time you
# use the wheel.
#
CTRL=31 WHEEL THR=0 59 60 61 62 63 65 66 67 68 69 127 ACTION=VFO
KEY=31 ACTION=NONE
#
# Big slider (pitch-bend controller): AF volume
#
PITCH ACTION=AFGAIN
#
# 8 Knobs (top left). These are "wheels" assigned to
#
# id=0: top row, leftmost:  StepAttenuator
# id=1: top row, 2nd from left: TX compression
# id=2: top row, 2nd from right: RIT value
# id=3: top row, rightmost:  Panadapter low
# id=4: bottom row, leftmost:  AGC value
# id=5: bottom row, 2nd from left: MIC gain
# id=6: bottom row, 2nd from right: TX drive
# id=7: bottom row, rightmost:  cycle through filters
#
CTRL=0 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=ATT
CTRL=1 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=COMPRESS
CTRL=2 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=RITVAL
CTRL=3 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=PANLOW
CTRL=4 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=AGC
CTRL=5 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=MICGAIN
CTRL=6 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=RFPOWER
CTRL=7 WHEEL THR=-1 -1 -1 -1 1 63 65 127 128 128 128 128 ACTION=FILTERUP
#
# Push actions on these knobs generate KEY events are are not assigned
#
KEY=0 ACTION=NONE
KEY=1 ACTION=NONE
KEY=2 ACTION=NONE
KEY=3 ACTION=NONE
KEY=4 ACTION=NONE
KEY=5 ACTION=NONE
KEY=6 ACTION=NONE
KEY=7 ACTION=NONE
#
# 8 Keys (below the 8 Knobs)
#
# id=16 label="1": top row, leftmost:  not assigned
# id=17 label="2": top row, 2nd from left: cycle through Alex ATT settings
# id=18 label="3": top row, 2nd from right: clear RIT value and turn off RIT
# id=19 label="4": top row, rightmost:  toggle CTUN
# id=20 label="5": bottom row, leftmost: cycle through noise blanker settings
# id=21 label="6": bottom row, 2nd from left: cycle through noise reduction settings
# id=22 label="7": bottom row, 2nd from right:  toggle VOX
# id=23 label="8": bottom row, rightmost: cycle through AGC fast/medium/slow
```

```
#
KEY=16 ACTION=NONE
KEY=17 ACTION=ATT
KEY=18 ACTION=RITCLEAR
KEY=19 ACTION=CTUN
KEY=20 ACTION=NOISEBLANKER
KEY=21 ACTION=NOISEREDUCTION
KEY=22 ACTION=VOX
KEY=23 ACTION=AGCATTACK
#
# "other" keys
# Note that the "DECK" key switches the MIDI channel of the device.
#
KEY=24 ACTION=TUNE                # LOAD    button: TUNE on/off
KEY=25 ACTION=LOCK                # LOCK    button: Lock VFO(s)
KEY=26 ACTION=PURESIGNAL          # DECK    button: toggle PURESIGNAL
KEY=27 ACTION=SWAPVFO             # SCRATCH button: Swap VFOs A and B
KEY=32 ACTION=VFOA2B              # SYNC    button: Frequency VFO A -> VFO B
KEY=33 ACTION=VFOB2A              # TAP     button: Frequency VFO B -> VFO A
KEY=34 ACTION=MOX                 # CUE     button: MOX on/off
KEY=35 ACTION=SPLIT               # >||     button: toggle Split
KEY=36 ACTION=MODEDOWN            # <<      button: Mode down
KEY=37 ACTION=MODEUP              # >>      button: Mode up
KEY=38 ACTION=BANDDOWN            # -       button: Band down
KEY=39 ACTION=BANDUP              # +       button: Band up
```

Appendix B. Sample midi.inp file for a Hercules DJ compact controller

To be added.