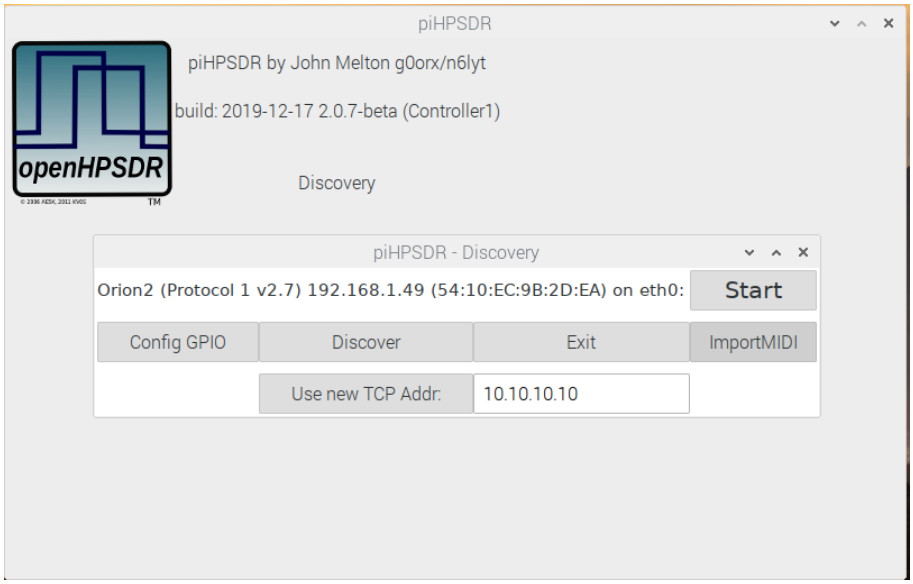
MIDI support in piHPSDR
Manual for Users

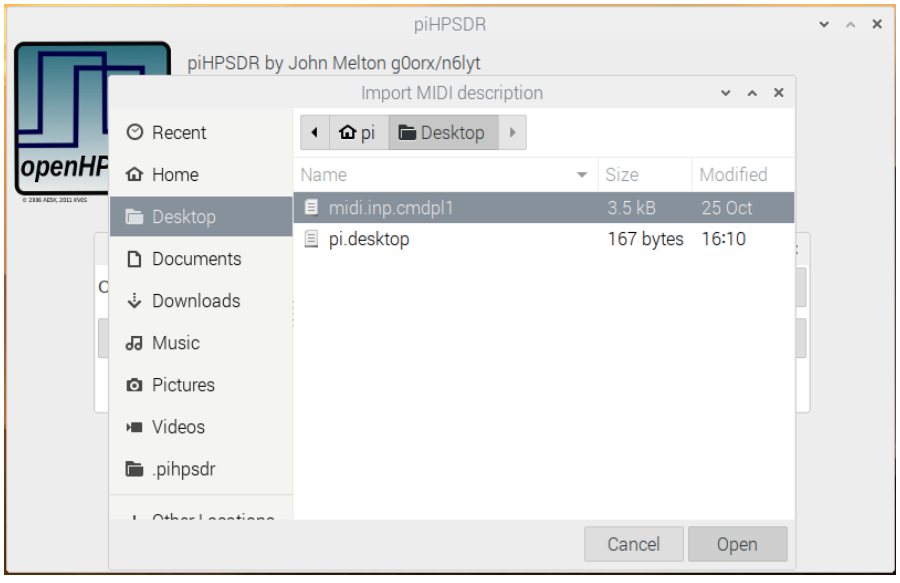
To use MIDI devices in piHPSDR, you must provide a text file defining which type of MIDI console to use and which MIDI keys/knobs are assigned to which radio functions. Three examples of such MIDI description file for some low-cost MIDI consoles (Behringer CMD PL1, Behringer Studio 2a, Hercules DJcompact) are provided in Appendix A–C.

"Importing" a MIDI description file

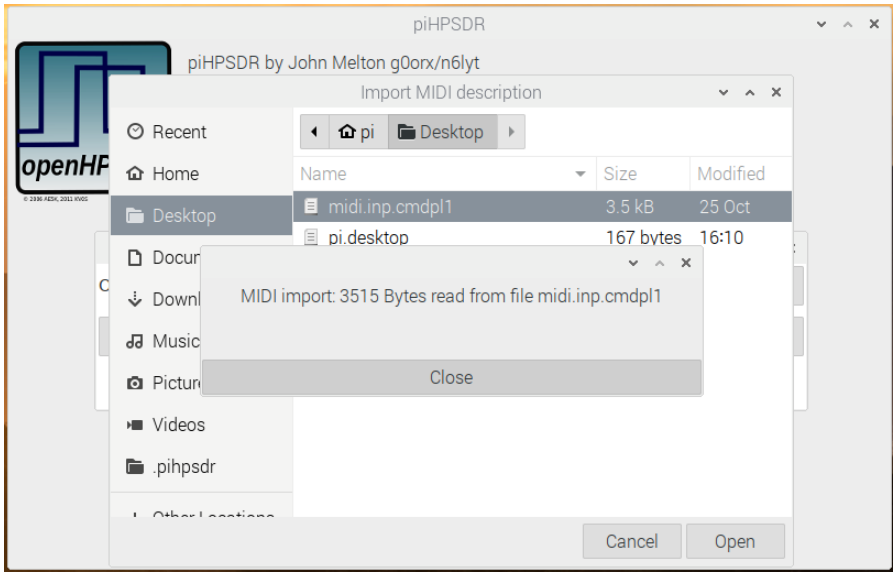
Before starting the radio (via the `START` button on the discovery screen), you must "import" the MIDI description file by hitting the `ImportMIDI` button on the same screen:



This opens an "open file" dialog from which you can select your MIDI description file and then click the "Open" button:



You get a short notification about the length of the file that has been "imported", which must be acknowledged by clicking the "Close" button. Then, you are back at the discovery screen and can start the radio.



The MIDI settings are stored with the radio settings, so upon subsequent invocations of piHPSDR, you do not need to “import” the MIDI settings again. However, once you have changed your MIDI description file, or if you want to use another one, this can simply be done by “importing” the settings again from the discovery screen. This way you can easily choose appropriate MIDI settings, e.g. for rag-chewing or contests, when starting the radio.

The MIDI description file

The MIDI description file is a plain text file you can create with your favourite text editor (nano, vi, etc.) The basic rules for the MIDI description 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, both a line reading "`DEVICE=CMD`" and "`DEVICE=CMD PL-1`" 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                    (only effective in "KEY=" lines)
WHEEL                    (only effective in "CTRL=" lines)
CHAN=<chan>
ACTION=<action>
THR=<thresholds>         (only effective in "CTRL=" lines which contain "WHEEL")
DELAY=<delay>            (only effective in "CTRL=" lines which contain "WHEEL")
```

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 possible delay (in milli seconds) which may be applied to WHEELs 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 left/right combined with normal/fast/very fast. All these key words are optional, the default values are

```
<chan>=0
<action>=NONE
<thr>=128 -1 128 -1 128 -1 128 -1 128 -1 128 -1
<delay>=0
```

The default channel value (zero) specifies that MIDI events from any channel (1–16) are accepted, the default action keyword ("NONE") specifies that no radio function should be performed, the default delay value (zero) speficies that no delays are enforced, and the default values for the "wheel speed" thresholds are such that no action is ever triggered. Below, a detailed explanation is given.

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 contains, say, "CHAN=5", then only MIDI messages from channel 5 trigger the event specified in the line. The CHAN key word is normally not used, but may become useful if more than one MIDI controller is attached.

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 if the additional key word ONOFF is given, an action will be executed both when pressing and releasing the key.
Example. If the TUNE keyword (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 sending only increments. 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.

Wheels can be used e.g. to change VFO frequencies, 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 of the corresponding key word). 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 in the other direction. 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 (`FILTERUP/FILTERDOWN`), modes (`MODEUP/MODEDOWN`), and changing the VFO step size (`VFOSTEPUP/VFOSTEPDOWN`). 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 controls	Description
A2B	key	copy frequency from VFO A to VFO B
AFGAIN	knob, wheel	set AF volume level
AGCATTACK	key	cycle through AGC attack (fast/med/slow)
AGCVAL	knob, wheel	change AGC value
ATT	key	cycle through ALEX attenuator settings
ATT	wheel, knob	set value of step attenuator
B2A	key	copy frequency from VFO B to VFO A
BANDDOWN	key, wheel, knob	cycle through bands downwards
BANDUP	key, wheel, knob	cycle through bands upwards
COMPRESS	wheel, knob	set TX compression value
CTUN	key	toggle CTUN mode
CURRVFO	wheel	change frequency of current VFO
CWL	key	press/release left CW paddle[c)]
CWR	key	press/release right CW paddle[c)]
CWSPEED	wheel, knob	change speed of CW keyer
DIVGAIN	wheel, knob	change DIVERSITY gain value
DIVPHASE	wheel, knob	change DIVERSITY phase value

DIVTOGGLE	key	toggle DIVERSITY on/off
DUP	key	toggle DUP (duplex) on/off
FILTERDOWN	key, wheel, knob	cycle through filters downwards
FILTERUP	key, wheel, knob	cycle through filters upwards
LOCK	key	toggle LOCK on/off
MICGAIN	knob, wheel	change microphone gain
MODEDOWN	key, wheel, knob	cycle through modes downwards
MODEUP	key, wheel, knob	cycle through modes upwards
MOX	key	toggle MOX state
NOISEBLANKER	key	cycle through NB settings
NOISEREDUCTION	key	cycle through NR settings
NONE	key, wheel, knob	no action
PANHIGH	wheel, knob	change "high" value of current panadapter
PANLOW	wheel, knob	change "low" value of current panadapter
PREAMP	key	toggle preamp on/off [a)]
PURESIGNAL	key	toggle PURESIGNAL enable/disable
RFGAIN	knob, wheel	change RF gain (for current receiver)
RFPOWER	knob, wheel	change TX drive level
RITCLEAR	key	clear (zero) RIT/XIT value
RITSTEP	key, wheel	change RIT/XIT increment
RITTOGGLE	key	toggle RIT on/off
RITVAL	wheel, knob	change RIT value [b)]
SAT	key	cycle through SAT modes (none/SAT/RSAT)
SPLIT	key	toggle SPLIT enable/disable
SWAPRX	key	toggle between RX0 and RX1
SWAPVFO	key	swap VFOs A and B
TUNE	key	toggle TUNE state
VFOA	wheel	change frequency of VFO A
VFOB	wheel	change frequency of VFO B
VFOSTEPDOWN	key, wheel	decrease VFO step size
VFOSTEPUP	key, wheel	increase VFO step size
VOX	key	toggle VOX enable/disable
VOXLEVEL	wheel, knob	Change VOX threshold
XITCLEAR	key	clear XIT value
XITVAL	wheel, knob	change XIT value [b)]

[a)] for CHARLY25 filter board: cycle through preamp settings 0/18/36 dB

[b)] if resulting value == 0, disable RIT/XIT, otherwise enable

[c)] use ONOFF keyword in this line to have both "key down" and "key up" actions

Some functions come in pairs, such as **BANDUP** and **BANDDOWN**. They can be mapped to separate keys, but when using a wheel or a knob, it is only necessary to assign one of these two functions. The same applies to **FILTERUP**/**FILTERDOWN**, **MODEUP**/**MODEDOWN**, **VFOSTEPUP**/**VFOSTEPDOWN**.

Note on the CW functions (CWL, CWR)

These functions in principle allow you to generate Morse code by pressing the buttons. In order to generate a message both upon “key down” and “key up”, the ONOFF keyword

must be used with both `CWL` and `CWR`. Either you assign the buttons and use one to generate dahs and the other one to generate dits, or you use the "straight" mode of the keyer. However, this is not very practical and thus seems to make little sense.
However, it is fairly easy to open the MIDI console and solder three wires to the PCB as to "shorten" two buttons with either arm of the CW paddle (or “shorten” one button, if you want to attach a straight key). Then you can hook up your paddle (or straight key) to the MIDI console and quite smoothly do CW. This possibility is of interest if piHPSDR runs on a computer which does not have GPIO input lines (for example, a Macintosh or a PC/Laptop running LINUX).

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=CURRVFO
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=AGCVAL
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: toggle RIT on/off
# 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=RITTOGGLE
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=SWAPRX              # SCRATCH button: Swap active/inactive RX
KEY=32 ACTION=A2B                 # SYNC    button: Frequency VFO A -> VFO B
KEY=33 ACTION=B2A                 # 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 Behringer CMD Studio2a controller

```
#
# Sample midi.inp file, suitable for a Behringer CMD Studio2a MIDI controller
#
DEVICE=Studio 2A
#
# Left Big Wheel: VFO knob for active receiver
#
CTRL=3 WHEEL THR=0 59 60 61 62 63 65 66 67 68 69 127 ACTION=CURRVFO
#
# Left big vertical slider: AF Gain
#
CTRL=24 ACTION=AFGAIN
#
KEY=1 ACTION=MOX                                    # Key Cue
KEY=2 ACTION=SPLIT                                  # Key >||
KEY=2 ACTION=LOCK                                   # Key Sync
KEY=6 ACTION=BANDDOWN                               # Key -
KEY=7 ACTION=BANDUP                                 # Key +
KEY=8 ACTION=VFOSTEPUP                              # Key A
KEY=9 ACTION=VFOSTEPDOWN                            # Key B
KEY=10 ACTION=MODEUP                                # Key 1
KEY=11 ACTION=MODEDOWN                              # Key 2
KEY=12 ACTION=FILTERUP                              # Key 3
KEY=13 ACTION=FILTERDOWN                            # Key 4
KEY=14 ACTION=A2B                                   # Key 1
KEY=15 ACTION=B2A                                   # Key 2
KEY=17 ACTION=SWAPVFO                               # Key 3
KEY=18 ACTION=FILTERDOWN                            # Key 4
```

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

```
# Hercules DJControl Compact HerculesDJ-midi.inp file.
# This file contains Control and key numbers specific
# to the Hercules DJContoller Compact device

# Rename this file to midi.inp and place in the Pihpsdr folder.
# Restart pihpsdr to incorporate settings included in this file.

# THE # Character indicates following text is a comment, not actionable instructions.
# Actions not commented out (No # letter at start of line) must be assigned.

DEVICE=DJControl Compact

# DJControl includes 2 large wheels which are assigned to VFO's
# Left wheel, VFOA increasing freq when rotated to right
CTRL=48 WHEEL THR=-1 -1 -1 -1 64 127 1 63 -1 -1 -1 -1 ACTION=VFOA

# Reverse VFO frequency rotation example:
# Left wheel, VFOA decreasing frequency when rotated to right.
# CTRL=48 WHEEL THR=-1 -1 -1 -1 1 63 64 127 128 128 128 128 ACTION=VFOA

# Right wheel, VFOB increasing frequency when rotated to right
CTRL=49 WHEEL THR=-1 -1 -1 -1 64 127 1 63 -1 -1 -1 -1 ACTION=VFOB

#Knobs or sliders
CTRL=54 ACTION=AFGAIN                           # Slider, bottom center, Selected RX AF gain
CTRL=57 ACTION=RFPOWER                          # Left volume  : RF Drive,
CTRL=59 ACTION=MICGAIN                          # Left Medium  : Mic Gain
CTRL=60 ACTION=AGCVAL                           # Left Bass    : Selected Receiver AGC gain
CTRL=61 ACTION=ATT                              # Right Volume : Step attenuator
CTRL=63 ACTION=BANDUP                           # Right Medium : Band Up
CTRL=64 ACTION=COMPRESS                         # Right Bass   : Tx Compression

# Buttons
# There are 2 4 button pads bottom center left and right.
# Additional 4 button pads can be virtually
# selected by holding down the MODE button and pressing 1,2,3 or 4 on the left or right pads.
# The selected pad is indicated by illuminating the Loop, FX, Sample or Cue light
# above the pad.

# Left Cue (Press Mode,4 (lower right button) to access pad button 1-4
KEY=1 ACTION=SPLIT                              # Cue P1 button : Toggle VFO Split operation
KEY=2 ACTION=SAT                                # Cue P2 button : Sync VFOA and VFOB
KEY=3 ACTION=MOX                                # Cue P3 button : MOX on/off
KEY=4 ACTION=TUNE                               # Cue P4 button : TUNE on/off

# Right Cue (Press Mode,4 (lower right button) to access pad button 1-4
KEY=49 ACTION=MODEUP                            # Cue P1 button : Modulation mode up
KEY=50 ACTION=MODEDOWN                          # Cue P2 button : Modulation mode down
KEY=51 ACTION=FILTERUP                          # Cue P3 button : RX Filter up
KEY=52 ACTION=FILTERDOWN                        # Cue P4 button : RX Filter down

# Left Sample (Press Mode,3 (lower left button) to access pad button 1-4
KEY=17 ACTION=NONE                              # Sample P1 button:
KEY=18 ACTION=NONE                              # Sample P2 button:
KEY=19 ACTION=NONE                              # Sample P3 button:
KEY=20 ACTION=NONE                              # Sample P4 button:

# Right Sample (Press Mode,3 (lower left button) to access pad button 1-4
KEY=65 ACTION=NONE                              # Sample P1 button:
KEY=66 ACTION=NONE                              # Sample P2 button:
KEY=67 ACTION=NONE                              # Sample P3 button:
```

```
KEY=68 ACTION=NONE                              # Sample P4 button:


# Left FX (Press Mode,2 (Upper right button) to access pad button 1-4
#KEY=9 ACTION=NONE                              # FX P1 button:
#KEY=10 ACTION=NONE                             # FX P2 button:
#KEY=11 ACTION=NONE                             # FX P3 button:
#KEY=12 ACTION=NONE                             # FX P4 button:

# Right FX (Press Mode,2 (Upper right button) to access pad button 1-4
KEY=57 ACTION=NONE                              # FX P1 button:
KEY=58 ACTION=NONE                              # FX P2 button:
KEY=59 ACTION=NONE                              # FX P3 button:
KEY=60 ACTION=NONE                              # FX P4 button:

# Left Loop (Press Mode,1 (Upper left button) to access pad button 1-4
KEY=25 ACTION=VFOSTEPUP                         # Loop P1 button:
KEY=26 ACTION=VFOSTEPDOWN                       # Loop P2 button:
KEY=27 ACTION=NONE                              # Loop P3 button:
KEY=28 ACTION=NONE                              # Loop P4 button:

# Right Loop (Press Mode,1 (Upper left button) to access pad button 1-4
KEY=73 ACTION=NONE                              # Loop P1 button:
KEY=74 ACTION=NONE                              # Loop P2 button:
KEY=75 ACTION=NONE                              # Loop P3 button:
KEY=76 ACTION=NONE                              # Loop P4 button:

# Buttons under left wheel
KEY=33 ACTION=SWAPRX                            # Left -> play btn    : SYNCVFO VFO A & VFO B
KEY=34 ACTION=A2B                               # left Cue btn        : Copy VFO B to VFO A
KEY=35 ACTION=VFOSTEPUP                         # Left Sync btn       : VFO Step up

# Buttons under right wheel
KEY=81 ACTION=SWAPVFO                           # Right -> play btn   : Swap VFO a & B
KEY=82 ACTION=B2A                               # Right Cue Btn       : Copy VFO A to VFO B
KEY=83 ACTION=VFOSTEPDOWN                       # Sync btn            : VFO STEP DOWN.

# Buttons above pads
KEY=43 ACTION=NOISEREDUCTION                    # Rec btn.            : Toggle NR, NR1, NR2
KEY=45 ACTION=NOISEBLANKER                      # Scratch/AutoMix btn : Toggle NB, NB1
KEY=47 ACTION=SWAPRX                            # Shift key.          : Swap active receivers.
```