Bauer REMI mk2 Handset – EWI MIDI Controller
User Manual

References

For a general overview of the REMI project, please refer to the author's website:

http://www.mjbauer.biz/REMI_Intro.htm

Design notes and construction details are posted on the web-page here:

http://www.mjbauer.biz/Build_the_REMI_mk2.htm

The latest firmware release and source code are available for download on GitHub here:

https://github.com/M-J-Bauer/REMI-2-EWI-MIDI-Controller

This edition of the manual is applicable to REMI firmware version 1.3.

Introduction

This manual provides information to help you test and set up your REMI 2 (or compatible) handset when construction is (more or less) complete. It is assumed that the breath pressure sensor module is functioning properly, i.e. the output voltage span of the module is at least 1V, but preferably more, ideally around 3V.

REMI hardware testing and sensor calibration requires the USB "service port" CLI to be functioning. So, let's begin with familiarization of the CLI.

1. Command-Line User Interface (CLI)

The CLI functions like the "Command Prompt" in Windows and the "Terminal" in Linux. Commands are provided to view and set configuration parameters, to view and set preset instrument assignments and for hardware testing, sensor calibration, etc.

To use the CLI, the REMI must be connected to a USB port on your computer. You'll need a USB cable with a "mini-B" or "micro-B" type plug on the REMI end, whichever matches the USB receptacle on your USB/serial bridge (FT232 module).

Terminal Setup

The first step is to download and install a terminal emulator application on your computer. The recommended app for Windows is "PuTTY". Copy the distribution files to a folder on your computer's local drive. To start PuTTY, you just double-click the executable file (putty.exe). It is convenient to create a desktop shortcut.

Connect a USB cable from your computer to the REMI service port (USB-serial connector).

Configure your PC terminal app to use a serial COM port connection. You will need to enter the COM port number that your computer has assigned automatically to the REMI. To find the COM port number for a Windows PC, open the Device Manager utility, accessible from Windows Control Panel. Click 'Ports (COM & LPT)' and identify the USB/Serial bridge used for the REMI. The port number 'COM xx' should be shown.

Configure the baud-rate of your terminal to 34800 baud, with no parity bit.

PuTTY has the 'Backspace' key set to ASCII 'DEL' (127) by default, which is quite useless. Change it to 'Ctrl+H', which is supported by the REMI CLI.

Save the Serial terminal setup, so you don't have to re-configure it again next time.

You should see a start-up message in the terminal window – something like this:

```
* MCU reset *
Bauer {REMI} Handset mk2 – Service Port CLI
>
```

If not, hit the [Enter] key on your computer keyboard. You should see the CLI prompt (**>**).

Command Line Basics

The CLI provides commands for setting up, testing and operating the REMI 2 handset. To see a list of all available commands, enter the command "**lc**" (= list commands) at the prompt as shown below…

```
> lc
cmds        config      default     diag        dump        help
lc          preset      reset       ver         watch
>
```

Another command named "**cmds**" does the same thing (for compatibility with other versions of the CLI used in various other projects).

To get the firmware version installed, type the command "**ver**" as follows:

```
> ver
Firmware version 1.2.05, Apr  5 2022
>
```

A command line consists of a command name followed by a number of "arguments" (*aka* "parameters") separated by one or more spaces. Some commands have no arguments. An argument (*abbr*: "arg") is just a string of characters. There may be some forbidden char's.

A special type of argument is called a "switch". A switch comprises the hyphen symbol (**-**) followed by one or more characters (no spaces). Switches typically follow immediately after the command name. A switch is often used to select a sub-command or a command option.

Command line switches and arguments are generally not case-sensitive.

Most commands have a built-in "help" option which displays a usage summary. To see the usage summary, type the command name followed by the switch **-help**. For convenience, a question mark (**?**) may be typed instead of '-help'. An example follows:

```
> preset ?
Usage #1:  preset     | List Preset MIDI Prgm numbers

Usage #2:  preset <pre#> [=] <value>    | Set Preset value
   where <pre#> is the Preset number (1..8)

Example:   preset 3 = 67   | Set MIDI Prgm 67 for Preset 3
>
```

In the above example, it can be seen that there are two usages of the command "**preset**". The first usage has no arguments following the command name. This usage will list the MIDI program numbers assigned to the 8 REMI presets. The second usage lets you assign a MIDI program number to a specified preset, as shown in the example.

Most commands, when entered with the switch '-help' or '?', will show command usage information. Otherwise, a command which expects at least one argument, when entered without any, should also show command usage.

In usage summaries, square brackets enclose optional parameters, e.g. [arg]. Braces enclose alternative options, e.g. {-x|-y}, where '|' means 'OR'. So the command can include either switch '-x' or '-y', but not both.

The CLI maintains a history of commands entered. To recall a previously entered command, type Ctrl+R (hold down the Ctrl key and press R) repeatedly until the desired command reappears. This feature is somewhat quirky, but more often than not, it works!

The 5 most frequently used CLI commands in the REMI mk2 handset are:

      lc/cmds   config   preset   watch   diag

The "**config**" command is explained in the section "**Configuration Options**".

The "**preset**" command is explained in the section "**Instrument Presets**".

The "**watch**" and "**diag**" commands are explained in the section "**Hardware Testing and Sensor Calibration**".

2. Hardware Testing and Sensor Calibration

USB Service Port Test

Connect your PC to the REMI USB port and start the terminal app. The REMI board should now be powered up. If the CLI is working properly, i.e. you can enter commands and see the responses on the terminal screen, then you can skip the remainder of this test.

If the CLI terminal is not responding, first check that the terminal COM port number is correct. Next, check that the serial port TX and RX signals are wired correctly. The PIC UART TX pin (U1TX/RC6) should be wired to the RX input of the USB-serial bridge (RXI).

Hit [Enter] a few times on the terminal (PC) keyboard. The 'TX' and 'RX' LEDs on the USB/serial bridge should flash.

MIDI OUT Transmitter Test

This is an optional test. If your micro-controller board has been assembled and wired up correctly, the MIDI transmitter/driver circuit should function properly. If you suspect that there may be a defect, then come back and perform this test. You will need an oscilloscope.

Solder a red LED (or a 1k resistor) between pins 4 and 5 on a 5-pin DIN plug. Insert the plug into the MIDI OUT socket wired to the REMI micro-controller board. Hook a 'scope probe onto MIDI OUT– (pin 5 on DIN plug). Hook the 'scope ground clip to GND (pin 3).

Connect your PC to the REMI USB port and start the terminal app. The REMI board should now be powered up. Enter the CLI command "**diag -t**". This will cause the MCU to transmit a data byte (0F hex) continuously. If the MIDI transmitter/driver circuit is functioning correctly, you should see a square wave on the 'scope with frequency = 3.125 kHz.

Pressure Sensor Test and Calibration

Enter the command: "**watch**". The terminal screen will display various data on a single line, continuously updated in real-time, as shown below. The update rate is about 5 times per second.

```
> watch
Hit [Esc] to quit.

TouchPads: 0x0000 | Pad0_ADC:  234 | Pressure:  187 | Modul'n:  455
>
```

To capture the data values at any instant, press the [Enter] key on your PC.

To exit the "watch" function, press the [Esc] key.

The Pressure value is the "raw" ADC reading. The "quiescent" reading (without blowing into the sensor) should be under 200, but not zero. While blowing into the mouthpiece at a comfortable maximum pressure for normal playing, hit the [Enter] key to capture the reading. That reading should be at least 200 ADC units higher than the "quiescent" reading.

The difference between the maximum and minimum (quiescent) readings is called the "span". Set the configuration parameter nick-named "**prspan**" to the pressure span value obtained. (See next section for details.) Your breath pressure sensor is then calibrated!

The higher the span value, the harder you will need to blow to get maximum audio volume (or whatever synth effect(s) may be controlled by breath pressure). Try experimenting with the "prspan" parameter value until you are happy with the range of loudness you can achieve.

Depending on the stability of the pressure sensor, it may be necessary to re-calibrate it from time to time. Note that the quiescent (minimum) pressure reading is automatically recalibrated every time the REMI starts up. Only the span might need to be manually recalibrated and this value is not critical… just so long as it's in the right ball-park.

Modulation Sensor Test and Calibration

The "Modulation Pad" is a sensor wired to produce a voltage proportional to the force applied. The firmware reads the signal and converts it to a data value which is transmitted in MIDI modulation messages (Control Change number 01). You can set the time interval between MIDI modulation messages using the CLI. (The default is 30ms.) Most sound synthesizers can be set up to respond to CC01 (modulation) messages in various ways. For example, it could be set up to control filter resonant frequency, vibrato depth, noise level, oscillator detune, etc.

Enter the command: "**watch**". Observe the "**Modul'n**" value, which is the ADC input reading for the modulation pad (or whatever other type of sensor you may be using). The minimum value, i.e. without pressing the Mod Pad, should be close to zero, typically below 10.

Press the Mod Pad and verify that the ADC reading increases with the force applied. The maximum value should be above 700, typically about 750. If not, check your Mod Pad sensor wiring and the op-amp buffer circuit. Actually, the op-amp buffer is probably redundant. The sensor signal could be fed directly into the ADC input. The main purpose for the buffer is to protect the MCU from excessive voltage, e.g. ESD, or whatever. It also presents a very low impedance signal to the ADC input.

The LM358 op-amp maximum output voltage is about (Vcc – 1.5) volts, i.e. about +3.5V. This explains why the sensor maximum reading is about 750, not 1023 as you might expect because the PIC18F_K22 has a 10-bit ADC. If you bypass the op-amp or substitute an op-amp with "rail-to-rail" output voltage range, the maximum reading obtainable should be 1023.

Set the configuration parameter nick-named "**modmax**" to the maximum ADC reading obtained. (See next section for details.) Your modulation pad sensor is then calibrated!

If your REMI uses a force-sensitive resistor (e.g. FSR-402) to implement the Modulation Pad, there will be a non-linear relationship between the force applied and the voltage across the load resistor, i.e. voltage at the ADC input. This non-linear characteristic is not wanted. Without compensation, the Mod Pad would feel too sensitive at the low end of the force scale.

Don't worry… the REMI has a configuration parameter named "**modband**" which defines a range of sensor readings at the low end of the scale called a "dead band". In this range, the data value sent in MIDI modulation messages will be zero. If the force applied to the sensor is above the dead-band range, then the MIDI modulation data value will increase proportionally. The "**modband**" parameter is the sensor ADC reading at the top end of the dead-band range. The parameter value should be set so that the Mod Pad feels right when controlling a synth effect, for example vibrato depth, filter cut-off frequency, or whatever. A good starting point is to set the parameter value to half of the maximum reading, e.g. about 350.

Touch-pad Test

Enter the command: "**watch**". Observe the "**TouchPads**" value, which is a hexadecimal number of size 16 bits. Each bit represents the state of a touch-pad: 0 = pad not touched; 1 = pad touched. So, if no pad is touched, the value shown should be zero (0x0000). If one or more pads are touched, the value should be non-zero.

If you convert the hex number to binary, you can determine which pad(s) are touched and hence verify that the touch-pad inputs on the PIC MCU are wired correctly. The pad bit allocations are shown below:

```
                  |   octave    | ---------- s e m i t o n e ---------- | LH4 |
Touch-pad ->      | OCT+ | OCT- | LH1 | LH2 | LH3 | RH1 | RH2 | RH3 | RH4 | RH5 |
Bit position ->   |  9   |  8   |  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
```

Example: If the pad LH4 (or RH5, depending on touch-pad layout) is touched, the hex value displayed should be 0x0001.

The 'watch' command also shows the ADC reading for the touch-pad LH4/RH5. The reading should be higher than 200 when the pad is not touched. The reading should drop below 150 when the pad is touched. The threshold value which determines whether a pad is touched or not is normally set to 180. If for some reason any of your touch pad readings fall outside the normal range, the threshold value can be changed using a configuration parameter named "**thres**". (See next section for details.)

Mode Switch

Provision of a 'Mode' switch is optional. If your REMI omits the switch, it is recommended to tie the MCU switch input pin to GND. The purpose of the Mode Switch is to select one of two sets of configuration parameters, so that it is easy to re-configure the REMI to control one or other of your two favourite synths without needing the CLI.

The Mode switch position (logic state) can be checked using the CLI command: "**diag -m**".

3. Configuration Options

Configuration options allow you to set up the REMI to work properly with a MIDI sound module, keyboard synthesizer, or a computer with a software synth (VST) application. These may differ in the MIDI message type used to control audio output level (volume). Keyboard synthesizers typically use "Note-on Velocity", "Aftertouch" (key pressure), or Control Change commands "Channel Volume" or "Expression" to control audio level.

The REMI firmware maintains a bunch of user-settable "configuration parameters" allowing it to function properly with a variety of MIDI-controlled sound modules and synthesizers, whether hardware or software. Some parameters define the physical properties of the handset, for example, the touch-pad layout and fingering scheme. Other parameters are provided for calibration of the breath sensor and modulation pad.

Duplicate sets of synth configuration parameters are provided so that it is quick and easy to switch between two of your favourite synthesizers or sound modules. The required synth parameter set is selected by the 'Mode' switch (if fitted). The '**config'** command addresses only the currently selected parameter set. The default parameter values in Mode 0 (i.e. switch input in low state) are intended for general MIDI synths, whereas Mode 1 defaults are intended for the REMI mk2 Sound Module (*aka* "Bauer EWI Synth").

To view a list of the configuration parameters, enter the command "**config**". This will display the nick-name for each parameter along with its current setting and a brief description...

```
> config
chan       1  Midi Basic Channel (1..16)
sysxen     1  Midi Sys.Excl.Msg Enabled (0,1)
prchen     1  Midi Prog. Change Enabled (0,1)
expncc     2  Midi Expression CC number (2,7,11)
expnint    5  Midi Expression Interval (5..50 ms)
expn14b    0  Send 14-bit Expression data (0,1)
modint    30  Midi Modulation Interval (10..100 ms)
legato     1  Legato Mode Enabled (0,1)
velsen     0  Velocity Sense Enabled (0,1)
pbenden    0  Pitch Bend Enabled (0,1)
padlay     0  Touch Pad Layout (0,1)
fingsc     0  Fingering Scheme (0,1,2,..)
thres    150  Touch Sense Threshold (max.250)
prspan   750  Pressure Sensor Span (max.750)
modmax  1000  Modulation Maximum (max.1000)
modband  250  Modulation Dead-band (max.500)
>
```

To change the value of a parameter, enter the command:

```
config <param> [=] <value>
```

... where *<param>* is a parameter nick-name (as in the listing above) and *<value>* is the new value to be set. The square brackets indicate that the equals sign (**=**) is optional.

Example: To set a value of 590 for the pressure sensor span, enter the command:

```
config prspan = 590
```
or…   ```config prspan 590```

Configuration Parameters

Following is a description of the config. parameters implemented in firmware version 1.3.

The first group determines how your synth will respond to MIDI control messages sent by the REMI. There are two sets of parameters in this group, one for each position of the 'Mode' switch, allowing you to swap easily between two favourite synth's.

chan   MIDI basic channel number (1..16). The REMI will send MIDI messages addressed to this channel. The channel number must match the MIDI channel set on your synth. The default setting is 1.

sysxen   MIDI "System Exclusive" messages enabled. This parameter is intended for use with the REMI mk2 Sound Module. If you are using a REMI Synth, set this parameter to 1; otherwise set it to 0.

prchen   MIDI Program Change enabled. Set this parameter to 1 if you want to select synth instrument voices using the PRESET button on the REMI. Otherwise, set it to 0. [Note: MIDI Program Change messages are ignored by the REMI mk2 Synth. A "Sys. Ex." Message is used instead.]

expncc   MIDI Expression Control Change number (1..31). If your synth supports Breath Pressure CC messages, set this param to 2 (default); otherwise, if your synth supports Channel Volume CC messages, set this param to 7; otherwise, set it to 11 for Expression. (See note 1 below.)

expn14b   MIDI Expression data size. MIDI Control Change messages can support 7-bit or 14-bit resolution for the expression data value. If your synth supports 14-bit CC data, set this param to 1; otherwise set it to 0.

legato   Legato Mode enabled. If your synth supports legato pitch changes, set this param to 1 (enable); otherwise set it to 0. (See note 2 below.)

velsen   Velocity Sense enabled. If your synth responds to "Key Velocity" data in MIDI Note-On messages, you might want to enable this function, depending on which synth effect(s) you want to control with it. (See note 3 below.)

pbenden   Pitch Bend enable. If your REMI has a Pitch Bend sensor fitted* and your synth supports MIDI Pitch Bend messages, set this param to 1; otherwise set it to 0. (* Planned future upgrade of REMI handset.)

The second group of parameters (below) defines hardware characteristics of your REMI, including sensor and touch-pad calibration data. There is only one set of parameters in this group because none affect MIDI messaging or synth operation.

expnint   MIDI Expression Interval (5..50 ms). This param sets the time interval between transmission of MIDI Expression CC messages. This should be set as low as your MIDI network data flow will permit. (Default: 5ms)

modint   MIDI Modulation CC message interval (10..100 ms). This param sets the time interval between transmission of MIDI Modulation CC messages. (Default: 30ms)

padlay   Touch-Pad Layout. The REMI firmware supports two (or more) options for the physical layout of pads on the instrument. Option 0 (default) has pads at finger positions: LH1, LH2, LH3 and RH1, RH2, RH3, RH4, RH5. Option 1 has pads at LH1, LH2, LH3, LH4 and RH1, RH2, RH3, RH4. (There may be more options supported by later firmware versions.)

fingsc	Fingering Scheme. The REMI firmware supports two (or more) options for the fingering scheme, i.e. the mapping of finger/pad combinations to respective musical notes (pitch). The fingering scheme is usually dependent on the touch-pad layout. Fingering option 0 uses pad RH5 (or LH4) to flatten most notes by a semitone (except C and F). Option 1 uses LH4 to sharpen notes C and F by a semitone, i.e. to play C# or F#.
thres	Touch Sense Threshold. This parameter sets the logic level threshold of the touch-pad sensing function. It is normally in the range 100 to 200. (Default = 150.) See also section: "Hardware Testing ..."
prspan	Pressure Sensor Span. This parameter sets the sensor ADC reading span. See section: "Hardware Testing and Sensor Calibration".
modmax	Modulation sensor maximum ADC reading (full-scale). See section: "Hardware Testing and Sensor Calibration".
modband	Modulation sensor dead-band adjustment. ADC reading at upper end of dead-band range. Typically about 25% to 50% of modmax for FSR-402. See section: "Hardware Testing and Sensor Calibration".

Notes:

(1)   Some sound modules and software synths are designed specifically for EWIs and respond to Breath Pressure Control Change commands (CC number = 2). If using this type of synth, set the MIDI Expression CC number = 2. Some MIDI synths respond only to "Channel Volume" messages (CC number = 7). If using such a device, set the MIDI Pressure CC number = 7.

(2)   The REMI sends a MIDI Note-On/Velocity command when the breath pressure exceeds a preset threshold. A corresponding Note-Off command is sent when the breath pressure drops below the "note-off pressure level". With legato mode enabled, while a note is playing, a change in fingering will cause another Note-On/Velocity command to be transmitted without first sending a Note-Off command. If your sound module or synth is monophonic, or set to Mono mode, the change in fingering should result in a different note sounding, i.e. a change in pitch, without "re-triggering" the envelope generator(s). Some polyphonic synthesizers do not support MIDI legato mode and may not function properly with legato mode enabled. To disable it, set the REMI legato parameter to 0.

(3)   Many touch-sensitive keyboard synthesizers respond to "key velocity" data received in MIDI Note-On/Velocity messages. The velocity data may be used to control audio amplitude (loudness) or various other synth effects, e.g. envelope attack rate, filter cut-off frequency, resonance, etc. For EWI controllers, it is generally preferred not to use velocity data to control audio amplitude, except if this is the only method supported by your synth. If you are not using velocity data to control a synth effect, it is best to disable it, because velocity sensing causes a delay in note triggering.

4. Instrument Presets

REMI 2 provides a facility to select 1 of 8 presets directly from the handset, without needing to use the synth control panel. The desired preset is selected by a button on the handset, in combination with the 8 upper touch-pads. MIDI Program Numbers corresponding to the 8 REMI presets can be set up using a CLI command.

The PRESET button can be used to change the MIDI program number (instrument voice) on your synth, assuming the synth responds to MIDI Program Change messages.

To select a Preset: While holding the touch-pad corresponding to the preset required, press the PRESET button. Preset #1 corresponds to the lowest touch-pad (near the connector end). Preset #8 is the highest pad (nearest to the mouth-piece).

To allow the REMI to send MIDI Program Change messages, the configuration parameter "**prchen**" must be set to 1. If it is not already set to 1, enter the command:

```
config prchen = 1
```

Of course, you can always use the synth control panel to change the instrument voice. If you prefer this option, it is probably best to disable MIDI Program Change messages, so that the REMI will not send this message if the PRESET button is hit accidentally. To configure this option, simply enter the command:

```
config prchen = 0
```

As stated before, there is a MIDI program number assigned to each of the 8 REMI presets. To see these numbers, enter the command "**preset**" as follows:

```
> preset
 _____
Preset | MIDI Prgm #
```````|````````````
    1 | 75
    2 | 72
    3 | 69
    4 | 74
    5 | 17
    6 | 20
    7 | 23
    8 | 67
 _____|_____
```

To change any of the assigned program numbers, enter the command:

```
preset p = nn
```

... where *p* is a preset number (1 to 8) and *nn* is the desired MIDI program (voice) number.

Example: To assign MIDI program number 97 to Preset #1, enter the command:

```
preset 1 = 97
```

Your synthesizer manual should have a list of MIDI program numbers corresponding to the instrument voices available. The "General MIDI" (GM) standard defines 128 instrument voices. Most keyboard synth's and sound modules provide all the GM voices.