Section 1: Instrument Presets & Configuration

Reference

For a general overview of the REMI synth module design and construction, please refer to the author's website: www.mjbauer.biz

Note: This User Guide is applicable to the synth firmware version current at the time the document was last revised. Some information may not match other firmware versions.

Introduction

The REMI synth module responds to standard MIDI controller messages. Supported message types are listed in the Appendix: REMI Synth MIDI Implementation Chart. The synth should operate with any standard MIDI keyboard or EWI controller connected to the MIDI IN port. Preferably, your MIDI controller should be configured to use "Omni-On Mono" mode.

There is a diversion from the General MIDI standard relating to the way in which a synth "patch" (instrument sound) is selected. The REMI synth does not recognize the MIDI "Program Change" message type, because it does not implement the full set of GM instrument sounds. Instead, an instrument sound is selected from a set of eight "Presets".

An instrument preset may be selected by a MIDI System Exclusive message, as implemented in the REMI mk2 handset (MIDI EWI controller). Your MIDI controller may be configurable to support the REMI "Preset Select" system exclusive message. Alternatively, a synth preset may be selected with either of the user interfaces.

The synth firmware supports two user interfaces which may be operated concurrently:

- Command-line user interface (CLI) accessed via the console USB/Serial port.
- Front-panel graphical user interface (GUI) comprising LCD screen and 6 buttons.

The REMI synth can be operated using the console CLI alone, i.e. without the front-panel GUI. The CLI is similar to the "Command Prompt" in Windows, or the "Console Terminal" in Linux. User interaction is all text-based. The CLI has commands to do everything, e.g. set configuration options, set instrument preset parameters, perform synth patch operations, wave-table creation, hardware testing, etc. To use the console CLI requires a computer with a terminal emulator app (e.g. 'PuTTY') installed.

The front-panel GUI (comprising LCD module and 6 push-buttons) is optional. Hardware variants of the REMI synth may omit the front panel altogether. The front-panel GUI can perform only a subset of operations provided in the CLI. Nevertheless, the GUI is convenient to perform often-used tasks without the need to connect a PC terminal.

The front-panel GUI serves two main purposes:

- Allow instrument Preset parameters to be viewed and changed.
- Allow common configuration parameters to be viewed and changed.

The GUI also shows the results of the "Power-on/start-up Self-Test" (PST) routine, if a hardware problem is detected.

Instrument Presets

A Preset is a bunch of parameters (numbers) which determine how the synth is controlled and how it will sound. Some preset parameters may apply to external synthesizer operation (via MIDI OUT), but most apply to the built-in sound synthesizer.

There are 8 instrument presets provided. The "active" preset can be selected by a button on the REMI EWI handset (in combination with the touch-pads), or by a GUI operation, or by a CLI command. The active preset number (1 ~ 8) is shown on the GUI "Home" screen if your REMI module has an LCD panel connected.

The table below summarises Preset parameters which are settable by the GUI. There may be other Preset parameters which are settable only by the console CLI.

Preset Parameter	Description
REMI Synth Patch	REMI Synth Patch ID number (0..999). This parameter selects the instrument (voice) that the synthesizer will use. The GUI displays the associated Patch Name.
MIDI Program/voice	MIDI "Program" (voice) number (1..127). Sent in a "Program Change" message to an external MIDI sound module connected to MIDI OUT, whenever the Preset is selected (activated), but only if the Program/voice value is non-zero.
Pitch Transpose (Offset)	The Pitch Transpose (offset) parameter is used to shift the pitch of all notes played up or down by a given number of semitones (range: -24 to +24). Pitch Transpose is applied to both the internal REMI synth and to MIDI Note-On messages transmitted to MIDI OUT.
Vibrato Control	Vibrato Control Mode for the REMI synthesizer may be set to "Automatic" (i.e. delay then ramp-up after note-on), or by MIDI Modulation (CC-01) message, or by MIDI Effect Switch (on/off) message, etc.

When firmware is loaded for the first time into a new REMI synth module, default values are assigned to all configuration parameters and Preset parameters. Users can change the default settings to suit their individual preferences. User settings are preserved in non-volatile memory (EEPROM) while the REMI is powered down.

Each preset has an instrument **patch** assigned to it. In addition, a "user-programmable" patch may be assigned. All patches except the "user patch" are stored in flash program memory with the program code. The "user patch" is stored in EEPROM and its ID number is zero (0).

You can change the patch assigned to a preset. When you switch off the power (or do an MCU reset), the REMI will start up with the last preset selected. The active patch will be the patch assigned to the last preset selected, which in general is not the "user patch".

Detailed information to help you create user patches and wave-tables may be found in a later section of this guide.

Section 2: Front-Panel User Interface (GUI)

REMI Synth Prototype Front Panel -- Home Screen

The front-panel GUI can perform only a subset of operations provided in the CLI. The GUI is convenient to be able to perform often-used tasks without the need to connect a PC terminal. In this capacity, the front-panel GUI serves two main purposes:

- Allow instrument Preset parameters to be viewed and changed;
- Allow general configuration parameters to be viewed and changed.

For most parameters, the GUI avoids the need to enter numeric values by allowing the user to select from a list of options. For example, the 'REMI Synth Patch' is selected from a list of names of predefined patches. Where numeric data entry is needed, the GUI will present a "data entry screen" which allows a decimal number to be entered, as in the example pictured below for the MIDI Program/voice number assigned to a Preset...

Example Data Entry Screen

When the REMI is powered up, the GUI displays the 'Home' screen. The Preset last selected will be restored. The Preset number is displayed in a large font size at the far right of the screen.

Along the bottom row of the Home screen, two main menu options are shown:

[*] SETUP   [#] PRESET

From the HOME menu screen:

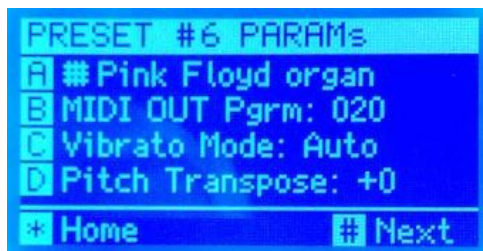Press **[A]+** to increment or **[B]-** to decrement the Preset number, which is then activated.

Press **[#]PRESET** to display Preset parameter settings for the active Preset.

Press **[*]SETUP** to display synth configuration/setup parameters.

Press **[C]Control** to display synth control parameters.

Press **[D]** to display a menu of other operational modes implemented.

Preset Param's Screen

Preset Patch (select) Screen

From the PRESET menu screen:

Press **[A]**, **[B]**, **[C]**, or **[D]**, to modify a Preset parameter listed.
This will take you to a data entry screen to enter a new parameter value.

Press **[#]Next** to advance to the next Preset parameters, without activating the Preset.

Press **[*]Exit** when done.

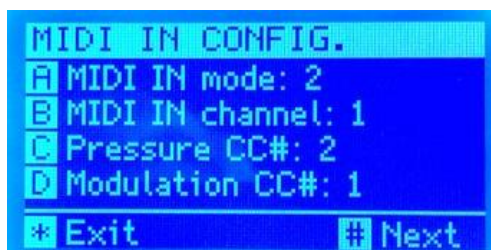**From the SETUP menu screen:**

Press **[A]**, **[B]**, **[C]**, or **[D]**, to view or modify a configuration parameter listed.
This will take you to a data entry screen to enter a new parameter value.

Press **[#]Next** to advance to the next page of configuration parameters.
The last two SETUP pages show system information – there is nothing to change.

Press **[*]Exit** when done.

Setup/Config Screen – page 1

Controls Screen – page 1

Section 3: Command-Line User Interface (CLI)

Introduction

The REMI synth can be operated completely using the console CLI alone. This is similar to the "Command Prompt" in Windows, or the "Console Terminal" in Linux. User interaction is all text-based. The CLI has commands to do everything, e.g. set configuration options, set instrument preset parameters, perform synth patch operations, wave-table creation, sound test, hardware testing, etc.

Another section of this User Guide is dedicated to the front-panel GUI option.

Terminal Setup

The first step is to download and install a terminal 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 console port (USB/serial connector). The type of cable required depends on your REMI hardware configuration. If it is based on an Olimex PIC32MX development board, you'll need a USB/Serial adaptor with a 9-pin "RS232" plug. Otherwise, consult the hardware information relevant to your REMI synth module.

Configure your PC terminal emulator software 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 seen.

Configure the data rate of your terminal emulator to 57600 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 normal.

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

Power up your REMI synth module and you should see a start-up message in the terminal window – something like this:

```
* MCU reset/startup
* Running self-test routine
! Handset not detected - Check connection.
Bauer {REMI} Synth mk2 Command Line User Interface
Firmware version 2.3.50, Dec 20 2020
>
```

Command Line Basics

The CLI provides many commands for setting up, testing and operating the REMI. To see a list of all available commands, enter the command "**lc**" (= list commands) at the prompt. Another command named "**cmds**" does the same thing (for reasons lost in the mists of time).

```
> lc
*          boot       cmds       default    diag       dump
help       lc         set        su         ver        watch
trace      eeprom     util       info       config     mimon
preset     patch      sound      wav
>
```

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 ?
Select or configure one of the 8 Instrument Presets...
Usage (1):  preset  <n>           | Select Preset (1..8)
Usage (2):  preset  <opt> [arg]   | Configure Preset...
  where <opt> =
 -i  : View Preset configurations (alias -l : List ..)
 -p  : Set REMI synth Patch ID number (0..999)
 -n  : Set MIDI OUT program/voice Number (0..127)
 -v  : Set Vibrato mode (O:off, E:fx.sw, M:mod'n, A:auto, D:detune)
 -t  : Set pitch Transpose, semitones (-24..+24)
>
```

In the above example, it can be seen that the command name "preset" may be followed by a switch to select one of several options. Some command options also require a numeric argument (arg). For example, to set the MIDI OUT program (voice) number to 79, the command would be: **preset -n 79**.

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 most common CLI commands are: **lc/cmds, help, info, config, preset** and **patch.**

If you want to experiment with your own patches and wave-tables, you'll also need the commands **sound** and **wav**. Other commands are intended primarily for development and testing purposes.

REMI Setup

The CLI provides commands to set up the synth for your individual preferences. Setup parameters are organised into 3 main classes:

- Instrument Preset parameters (for 8 Presets)
- Synth Configuration parameters (including MIDI IN and MIDI OUT setup)
- Synth Control parameters (e.g. Reverberation and other sound effects)

Setup parameters, other than Presets, are mostly "set and forget", i.e. once you adjust parameter values to suit your REMI hardware configuration, synth preferences, playing style and external MIDI equipment (if any), it is unlikely that these parameters will need to be changed again often, if ever.

Commands to view and modify setup parameters are "**preset**" and "**config**" which perform similar operations to those provided by the front-panel GUI.

Below is the output from the "config -help" command option:

```
> config ?
View/set synth config. parameter(s) ...
Usage (1):  config {no arg}      |  list Config param's
Usage (2):  config <param> <arg> |  set param value to <arg>
  ... where <param> = parameter mnemonic (3LA - see listing)
Example:    config mic 4         |  Set MIDI IN channel to 4>
>
```

To view the current settings, type "**config**" (without any arg) and you should see something like this (depending on the synth firmware version):

```
> config
      mib | MIDI IN Baudrate: 31250 bps
      mim | MIDI IN Mode: 2 (Omni-On-Mono)
      mic | MIDI IN Channel: 1
      mip | MIDI IN Pressure CC #: 2
      min | MIDI IN Modulation CC #: 1
      mom | MIDI OUT Mode: 2 (Omni-On-Mono)
      moc | MIDI OUT Channel: 1
      mop | MIDI OUT Pressure CC #: 2
      pti | MIDI OUT Pressure TX intrvl: 10
      rva | Reverb Attenuator gain: 80 %
      rvm | Reverb Mix (wet/dry) ratio: 10 %
      aco | Ampld Control Override: 0 (Off)
>
```

These are the default **config** parameter values. You might not need to change any of these, except maybe the Reverb Mix and Ampld Control Override (see note below).

To change a parameter value, type its nickname (3-letter abbreviation) followed by the new value, for example, to set the MIDI OUT channel number to 12, type the command:

```
config moc 12
```

Config options allow you to set up the REMI synth to work properly with a MIDI controller connected to the MIDI IN port.

If you are using a REMI mk2 (MIDI) handset, use the default values for all config options.

Note: If you are using a MIDI keyboard controller, it is probably best to set the config parameter '**aco**' to 1, with the command "**config aco 1**". This setting will force the synth to ignore the patch 'AC' (Amplitude Control) parameter and always to use the Amplitude Envelope shaper to control the audio output level.

If your keyboard is touch-sensitive, i.e. supports MIDI Velocity Control, the REMI synth will respond accordingly. Otherwise, you should set up your keyboard to send the maximum fixed velocity value (127), if possible. The REMI synth does not support MIDI Aftertouch messages.

External MIDI synths and sound modules

Config options allow you to set up the REMI to work properly with external MIDI equipment connected to the MIDI OUT port (where fitted), for example a sound module, keyboard synthesizer, or a computer with a software synth (VST) application installed. 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) and/or "Expression" Control Change commands to control audio level.

Some sound modules and VST synths are designed specifically for EWIs to respond to Breath Pressure Control Change commands (CC number = 2). If using this type of synth, set the MIDI OUT Pressure CC number = 2.

Some MIDI synths respond only to "Channel Volume" messages (CC number = 7). If using such a device, set the MIDI OUT Pressure CC number = 7.

Instrument Preset Setup

The CLI provides a command "**preset**" for the purpose of setting up Preset options. This command supports all of the operations provided in the front-panel GUI, and more. Below is the output from the "preset –help" command, showing command usage:

```
> preset ?
Select or configure one of the 8 Instrument Presets...
Usage (1):  preset  <n>          | Select Preset (1..8)
Usage (2):  preset  <opt>  [arg]  | Configure Preset...
  where <opt> =
 -i  : View Preset configurations (alias -l : List ..)
 -p  : Set REMI synth Patch ID number (0..999)
 -n  : Set MIDI OUT program/voice Number (0..127)
 -v  : Set Vibrato mode (O:off, E:fx.sw, M:mod'n, A:auto, D:detune)
 -t  : Set pitch Transpose, semitones (-24..+24)
>
```

There are two variants of the "preset" command. The first simply allows a Preset to be selected, i.e. to change the active Preset. In the example below, Preset 5 is activated:

```
> preset 5
Active Preset: 5 : Patch #48 Dulciana
>
```

The command confirms the selected preset and the patch ID assigned to it.

The second usage variant requires a switch to choose a command option, <opt>. For example, option '**-i**' will show information about all presets:

```
> preset -i
PRESET | Patch              | MIDI Prgm | Vibrato   | Transpose
`````````````````````````````````````````````````````````````````````````````
   1 | 10 Recorder, treble | 75 | Disabled | +12
   2 | 11 Recorder Celeste | 72 | Disabled | +0
   3 | 24 Oboe, soft reed | 69 | Auto ramp | +0
   4 | 26 Electric Clarinet | 74 | Auto ramp | +0
   5 | 48 Dulciana | 17 | Auto ramp | +0
   6 | 45 Pink Floyd organ | 20 | Auto ramp | +0
   7 | 46 Theatre Organ | 23 | Auto ramp | +0
   8 | 41 Jazz-rock organ | 67 | Auto ramp | +0
_____
Active Preset: 5 : Patch #48 Dulciana
>
```

The output confirms that the active Preset is number 5.

To change a parameter setting, type the command name, then the option switch <opt>, then the new parameter value. For example, to change the Vibrato Mode so that vibrato depth is controlled by the MIDI Modulation message (CC-01), type the command as follows:

```
> preset -v M
>
```

To verify that the setting has changed, use the preset 'info' option again.

When the REMI module operates as a MIDI controller, outputting MIDI messages to the MIDI OUT port, the external synth Program/voice number is selected by the Preset parameter "MIDI Program/voice number". This number, if non-zero, is sent to the external sound module in a MIDI "Program Change" message, whenever a REMI preset is selected. Depending on how the external synth is set up, it may or may not respond to Program Change messages.

When the REMI module operates using its built-in synthesizer, the instrument sound is determined by the Preset parameter "REMI synth Patch ID". This number selects one of many predefined instrument patches.

To view a list of all available "predefined" patches by name, with corresponding ID numbers, enter the command "**patch -L**", as follows:

```
> patch -l
     ID# Patch Name   WT1 WT2
``````````````````````````````````````````

  0    10  Recorder, treble       13   13
  1    11  Recorder Celeste        4   13
  2    12  Clean Bright Flute      6    6
  3    13  Mellow Pipe #1         12   12
  4    14  Analogue Flute          7    3
  5    15  Pan Pipe Dream          2    8
  6    21  Reed Woodwind 1        23   23
  7    23  Alto Saxophone          9    9
  8    24  Oboe, drawbar organ    32   32
  9    25  Psychedelic Oboe       10   32
 10    26  Electric Clarinet      11   11
 11    40  Jazz Organ Celeste      5    5
 12    41  Hammond Organ 1        24   24
 13    42  Hammond Organ 2        25   25
 14    43  Rock organ #1          14   14
 15    44  Rock organ #2          25   25
 16    45  Pink Floyd organ       20   20
 17    46  Theatre Organ          27   27
 18    47  Bauer Organ            22   22
 19    48  Dulciana               30   30
 20    49  Bright Comping         35   35
 21    60  Violin #1               3    3
 22    90  Test Patch              1    3
 23    91  Wave Morph Demo #1      4    3
 _____

>
```

To set the Patch ID for the currently selected Preset, enter the command "**preset –p *nn***", where '***nn***' is the patch ID number indicated in the list above. For example, to select the patch named "Psychedelic Oboe", the command would be:

```
> preset -p 25
```

Be sure to enter the Patch ID number, not the index value in the first column.

Remember that when you change any Preset parameter, the new setting is applied only to the currently selected (active) Preset. All other Presets remain unchanged. Whenever you activate a different Preset, the parameters associated with the new Preset will take effect.

You can also select a different patch without changing the active Preset. The change is temporary and will revert to the patch assigned with a Preset when a Preset is selected again. Use the "**patch -p ##**" command to load a temporary patch, as in this example:

```
> patch -p 30
```

This command will output a list of patch parameters which define the instrument sound. A later section of this user guide is dedicated to creating a "**user patch**", which allows you to define your own instrument sound.

# Section 4:  Patch Programming & Wave-tables

## User Patch

Each **preset** has an instrument **patch** assigned to it. In addition, a "user-programmable" patch may be assigned. All patches except the "user patch" are stored in flash program memory with the program code. The "user patch" is stored in EEPROM and its ID number is zero (0).

You can change the patch assigned to a preset, but when you switch off the power (or do an MCU reset), the REMI will start up with the last preset selected. The active patch will be the patch assigned to the last preset selected, which in general is not the user patch.

The "active patch" may be modified using the CLI command "**patch $$ *nn***", where $$ is a two-letter mnemonic (abbreviation) representing a patch parameter and ***nn*** is a numeric value to be assigned to the parameter. The modified patch is only temporary. It will be lost if the preset is changed, or if the same preset is re-activated, unless the modified patch is saved to EEPROM. To save the active patch, use the command "**patch -s *name***", where *name* is optional. The user patch name may be up to 20 characters… without any spaces.

Anytime you want to re-load the user patch, use the command: "**patch -u**". Of course, if you change the preset, then the patch will change to whatever is assigned to the active preset.

The user patch is intended for experimenting with instrument sounds. When you create a new user patch and you want to assign it to a preset "persistently", first select the preset number you want, then use the command "**preset -p 0**". The user patch ID number is 0.

## Patch Programming

Instead of using knobs and switches like a "real" synthesizer, the REMI synth is patched by means of a set of numeric parameters… (see table below). A variant of the CLI "**patch**" command is provided for this purpose.

| Oscillators | Mixer & Contour Env. | Noise Gen. & Filter | Envelope & Ampld Ctrl |
|---|---|---|---|
| OSC1 Wave-Table # | Mixer Control Mode | Noise Mode | Amplitude Env Attack Time |
| OSC2 Wave-Table # | Mixer OSC2 Level (%) | Noise Level Control | Amplitude Env Peak Time |
| OSC2 Detune (cents) | Contour Env Start Level (%) | Filter Freq. Control | Amplitude Env Decay Time |
| Pitch Bend Range (cents) | Contour Env Delay Time | Filter Corner Frequency | Amplitude Env Sustain Level |
| LFO Frequency | Contour Env Ramp Time | Filter Resonance | Amplitude Env Release Time |
| Vibrato Depth (cents) | Contour Env Hold Level (%) | Filter Pitch Track (offset) | Output Ampld Control Mode |
| Vibrato Delay/Ramp Time | | | |

The REMI synth can be programmed (patched) by the user to create a new sound, without needing to modify and re-compile the firmware. But, if you want to add more than one new instrument sound, then you will need to re-build and update the firmware.

To see a listing of the active patch parameter values, type the command: "`patch -d`". The listing also shows available choices and ranges of parameter settings. Note that the listing is in a format designed to be copied and pasted into a patch data definition in a C source file.

```
> patch -d
Patch ID#:  45 -> Pink Floyd organ

        20,     // W1: OSC1 Wave-table ID (0..250)
        20,     // W2: OSC2 Wave-table ID (0..250)
        -5,     // OD: OSC2 Detune, cents (+/-1200)
        100,    // PB: Pitch Bend range, cents (0..1200)
        70,     // LF: LFO Freq x10 Hz (1..250)
        30,     // VD: Vibrato Depth, cents (0..200)
        500,    // VR: Vibrato Ramp time (5..5000+ ms)

        0,      // MC: Mixer Control (0:Fixed, 1:Contour, 2:LFO, 3:Exprn, 4:Modn)
        50,     // ML: Mixer OSC2 Level in Fixed mode (0..100 %)
        0,      // CS: Contour Env Start level (0..100 %)
        5,      // CD: Contour Env Delay time (5..5000+ ms)
        200,    // CR: Contour Env Ramp time (5..5000+ ms)
        80,     // CH: Contour Env Hold level (0..100 %)

        0,      // NM: Noise Mode (0:No wave, 1:Add wave, 2:Mix wave; +4:Pitched)
        0,      // NC: Noise level Ctrl (0:Off, 1:Fixed, 2:Env, 3:Exprn, 4:Modn)
        0,      // FC: Filter Ctrl (0:Fixed, 1:Contour, 2:Env+, 3:Env-, 4:LFO, 5:Modn)
        0,      // FR: Filter Resonance x10000  (0..9999, 0:Bypass)
        0,      // FF: Filter Freq. (10..10000 Hz, 0:NoteTracking)
        0,      // FT: Filter Tracking offset (0..60 semitones)

        10,     // AA: Ampld Env Attack time (5..5000+ ms)
        0,      // AP: Ampld Env Peak time (0..5000+ ms)
        10,     // AD: Ampld Env Decay time (5..5000+ ms)
        10,     // AR: Ampld Env Release time (5..5000+ ms)
        100,    // AS: Ampld Env Sustain level (0..100 %)
        2       // AC: Ampld Control (0:Max, 1:Fixed, 2:Exprn, 3:Env, 4:Env*Vel)
>
```

REMI users who are prepared to re-compile the firmware can add their own patches and wave-tables, limited only by the amount of MCU flash program memory. CLI commands "patch" and "wav" include options to dump patch parameters and wave-table data (resp.) as C source code definitions.

**Oscillator Wave-table Selection: W1, W2 parameters**

Two patch parameters specify which wave-tables out of a large selection will be used by the synth oscillators. The assigned wave-tables determine the waveforms and hence the harmonic content of the oscillator outputs. The "user patch" can specify any pre-compiled wave-table (stored in MCU program memory) or a user-created wave-table (see next sub-section).

Pre-compiled wave-table definitions including table ID numbers can be found in the synth source file "remi_synth2_data.c". Refer to the wave-table descriptor array.

To change the wave-table assigned to oscillator #1 (OSC1) in the active patch, use the command "`patch w1 nn`", where '*nn*' is the wave-table number.

### Oscillator (OSC2) De-tune: OD parameter

The pitch of the secondary oscillator can be "detuned", i.e. increased or decreased relative to the primary oscillator. The "detune" factor is a patch parameter having units of "cents", so that the detune resolution is 1/100th of a semitone. If the detune factor is a fraction of a semitone, typically in the range 3 to 30 cents, and both oscillators are driven from wave-tables with similar harmonic content, the resulting effect is known as "Voix Celeste" (heavenly voice). This effect greatly enriches the soundscape possibilities of the synthesizer.

Maximum allowed value for the OD param is ±1200 cents (±1 octave).

### LFO Frequency: LF parameter

In addition to the two wave-table oscillators, a low-frequency oscillator (LFO) is provided. The LFO can be used to modulate the audio oscillator frequency to implement vibrato, or the LFO can be used to modulate the oscillator mix ratio. In the latter case, the mixer envelope output level determines the modulation depth.

The LF parameter value is 10 times the actual LFO frequency. For example to set the LFO freq. to 4.5 Hz, use the command "`patch LF 45`".

### Vibrato Depth and Ramp-up Time: VD, VR parameters

Vibrato Depth (VD) is the maximum deviation in pitch from the normal note pitch, specified in cents (1/100$^{th}$ of a semitone), up to 200 cents (2 semitones).

If vibrato mode is set to "Automatic", vibrato will start after a delay from Note-On and after a Legato note change. The delay time is variable and is set by the parameter VR (milliseconds). After the delay time ends, the vibrato depth ramps up from zero to the value specified by the parameter VD. The time taken to ramp up is also set by the parameter VR.

Note: Vibrato Control Mode is not a patch setting – it is a Preset parameter. Vibrato can be controlled automatically by the Vibrato Delay-Ramp function, or by MIDI Modulation (CC-01) messages, or it may be disabled… for any Instrument Preset.

### Wave Mixer Control: MC, ML parameters

The REMI synth implements a pair of high-accuracy audio wave-table oscillators which use independent wave-tables. The two oscillator outputs are fed into a mixer which scales and adds the two signals in a variable ratio. The oscillator mix ratio can be fixed, or it may be varied in time as the note progresses, depending on the Mixer Control Mode (MC).

If the mix ratio is fixed (MC = 0), the synth will use the "Mixer Osc2 Level" parameter (ML) to determine the mix ratio. For example, to set the OSC2 level to 20% (hence OSC1 = 80%), use the command "`patch ML 20`".

The wave mixer has its own dedicated envelope shaper, called the "Contour Envelope" to control the oscillator mix ratio. This capability is used to implement "waveform morphing", a technique used to vary the "timbre" (harmonic content) of the sound as the note progresses. To activate this mode, use the command "`patch MC 1`".

Set the Contour Envelope parameters to achieve the desired morph effect. This technique works more dramatically with two wave-forms having very different harmonic content.

Alternatively, the oscillator mix ratio may be modulated by the LFO, or by MIDI Control Change messages, i.e. Breath Pressure (Expression) or Modulation (CC-01). When the mixer is controlled by the LFO, the vibrato depth parameter (VD) sets the "modulation index" in the range 0 ~ 100%. For example, with VD = 50%, the mix ratio (OSC2 %) will swing from 25% to 75%. With VD = 100, the mix ratio will swing from 0 to 100%.

### Contour Envelope Generator: CS, CH, CD & CR param's

The Contour Envelope output is simply a linear ramp with a "delayed start". The output level maximum range is 0 ~ 100 (%). Patch parameters CS and CH specify the output level at the start and end of the ramp, respectively. When a note is initiated, the output level is CS. This level is maintained for a delay time, specified by parameter CD (range 0 ~ 5000 milliseconds). When the initial delay expires, the output level ramps up or down to the "hold level" (CH) over the "ramp time" specified by CR. The "hold level" is maintained until the note ends.

### Noise Generator and Filter: NM, NC param's

The synth incorporates a Noise Generator function, the output of which is fed into a "noise mixer" where it is mixed with the output of the wave mixer, if noise is enabled. There are various options available for the type of noise generated and the way it can be modulated.

Noise may be low-pass filtered, or it may be "pitched" (narrow band-pass filtered with centre frequency equal to the note pitch). Pitched noise is produced by ring modulating filtered noise with the output of oscillator #2. For best results, OSC2 should generate a pure sine wave at the fundamental note pitch. The filter cut-off frequency should be set low (10Hz) and the resonance factor should be set high ( > 0.9 ).

Note: There is only one filter implemented in the synth audio sampling routine. The filter may be patched into the Noise Generator, or it can be patched into the main audio stream following the wave mixer, but it cannot perform both functions together. If noise is enabled, the filter will be dedicated to the noise generator.

To disable noise completely, set parameter NC to zero. This will make the filter available for filtering the output of the wave mixer. Otherwise, the filter is dedicated to the noise function. Set the filter parameters according to the type of noise you want.

To set the noise output to a fixed (constant) level, set NC = 1. To modulate the noise level with the Amplitude Envelope, set NC = 2. To modulate the noise level with MIDI "Pressure" or "Expression" messages (CC-02, CC-07, or CC-11), set NC = 3. To modulate the noise level with MIDI "Modulation" messages (CC-01), set NC = 4.

Note: The default MIDI CC number for "Expression" is 02: Breath Pressure. This can be changed in the Configuration Settings if required to suit your MIDI controller.

The noise mode is selected by the patch parameter NM. Set NM = 0 to bypass the wave (oscillator) mixer, so that the output is noise only. Set NM = 1 to add the noise generator output to the wave mixer output. Set NM = 2 to blend noise with the wave signal proportionally. To enable the Ring Modulator for pitched noise, add 4 to the value of NM.

**Bi-quad Resonant Filter: FC, FR, FF & FT param's**

The synth algorithm incorporates a "bi-quad" resonant low-pass DSP filter with variable corner frequency (Fc) and resonance factor. This filter can be "patched" into the post-mixer audio stream to further modify the "timbre" of the sound, or it can be utilized in the noise generator function, but not both. (See note in Noise Generator section above.)

The wave filter will be disabled if the noise parameter NM is non-zero, or FR is zero.

The filter corner/resonant frequency (Fc) may be constant, or it can be made to track the pitch of the note playing. To enable note pitch tracking, set parameter FF to zero (0). The corner frequency can be "offset" by a specified number of semitones above the fundamental pitch of the note playing. Set the "Filter Tracking Offset" parameter FT accordingly.

If the patch parameter FF is non-zero, it is used to specify a fixed corner frequency in the range 10 ~ 10,000 Hz.

The filter resonance factor can be adjusted over the range 0.0001 to 0.9999. Higher values produce a narrower band-width, typically used for a "wah-wah" effect. The actual parameter value entered is the resonance factor times 10,000, i.e. shift the decimal point 4 places right. For example, to enter a resonance factor of 0.95, set parameter FR = 9500.

The filter frequency may be modulated in real time while a note is playing. The modulation control source is determined by the parameter FC. Available options include Contour Envelope, Amplitude Envelope (normal or inverse), LFO, Modulation CC messages, etc.

**Amplitude Envelope generator: AA, AP, AD, AS & AR parameters**

A five-segment envelope shaper provides the classic "attack, peak-hold, decay, sustain, release" (AHDSR) amplitude profile. The Amplitude Envelope output may be used to control a variety of effects, depending on the patch.

The "peak-hold" and "decay" segments of the Amplitude Envelope may be eliminated by setting the peak-hold time parameter AP = 0. With this setting, when a note is triggered, the envelope output will ramp up to the sustain level during the "attack" phase, bypassing the "peak-hold" and "decay" segments.

**Audio Output Attenuator: AC parameter**

The final audio output level (loudness) is controlled by a variable-gain attenuator.

Output level can be controlled by MIDI "Expression" Control Change messages, typically Breath Pressure (CC-02) for an EWI controller, or by the Amplitude Envelope for a MIDI keyboard controller, with or without MIDI "Note-On Velocity" message data. Set the patch AC parameter accordingly.

Note: If you are using a MIDI keyboard controller, it is probably best to set the config parameter '**aco**' to 1, with the command "**config aco 1**". This setting will force the synth to ignore the patch 'AC' (Amplitude Control) parameter and always to use the Amplitude Envelope to control the audio output level. Hence, you will be able to use all of the predefined patches, many of which have the patch 'AC' parameter set to control audio level by Breath Pressure to suit EWIs.

## Wave-table creation and sound test

The firmware provides a utility for users to create their own wave-tables. A CLI command "**wav**" is provided for this purpose. A user-created wave-table can be tested by modifying the active patch, but the user wave-table is only temporary... it will be lost when the synth is powered down or reset, unless it is saved.

A user-created wave-table can be saved in EEPROM by entering the command "**wav -s**". Actually, the wave-table data is not stored in EEPROM because the device doesn't have enough capacity. However, a "wave-table descriptor" (data structure containing a list of partial amplitudes) is stored in the EEPROM. When the synth is powered up or reset, the user wave-table is regenerated in data memory (SRAM) from information in the descriptor.

To assign a user-created wave-table to a *predefined* patch, you must re-build the firmware. Patches and wave-tables are defined in the source code file "remi_synth2_data.c".

The "wav" command has several options or "sub-commands". Type "**wav**" by itself to show a list of them, with a brief usage summary, as follows...

```
> wav
Utility to create, test and save REMI synth wave-tables.
Any cmd option except '-x' will activate the user wave-table.
Enter 'wav -x' cmd when done to restore previous patch.

Usage:  wav  <option>  [arg1] [arg2] ...
<option>------------------------------------
  -n : New wave-table  [arg1 = table size]
  -f : Set Osc.Freq.Div <arg1> for sound test
  -l : List wave-table and partial param's
  -a : Add partial to wave-table buffer --
          <arg1> = partial order (1..16)
          <arg2> = partial ampld (1..99 %FS)
  -c : Clear partial, <arg1> = order (1..16)
  -h : Create table from Hammond organ drawbars
        (Enter 'wav -h' for usage details)
  -z : Scale table peak magnitude [arg = %FS]
  -k : Apply anti-alias filter (K = 5..500)
  -d : Dump wave-table as C array def'n
          [arg1 = Waveform_name (no spaces)]
  -s : Save user wave-table def'n in EEPROM
  -x : Exit -- restore last patch selected
------------------------------------------------
>
```

When you use any of the "wav" command options, except "-x", the active patch will be replaced with a "test patch" having the user wave-table assigned to OSC1. However, the pre-existing patch will be remembered. When you are done with wave-table creating, type the command "**wav -x**" to restore the original patch. Or simply select a Preset.

You can select any other patch you like to test your wave-tables, but you will need to assign the user wave-table to one or both oscillators using the command "**patch w# 0**", where # is 1 for OSC1, or 2 for OSC2, and the wave-table number zero (0) identifies the user wave-table.

To check the wave-tables currently selected, enter the command "`patch -w`".

To create a new wave-table, there are two options: "`wav -n`" or "`wav -h`".

**Additive Synthesis**

Theoretically, any periodic waveform can be constructed by adding a series of pure sine-waves, comprising a "fundamental" and a number of "harmonics" (also known as "partials"). This theory was discovered by a mathematician by the name of Fourier.

The technique for creating waveforms using addition of "partials" is called "Additive Synthesis". In contrast, another technique called "Subtractive Synthesis" works by passing a complex waveform (rich in harmonics) through a filter to modify the spectral content. The latter technique was used in classic voltage-controlled analogue synthesizers.

The REMI wave-table creator uses Additive Synthesis. Up to 16 partials, including the fundamental, may be included in the composite wave. But there is no rule that says the fundamental must be present. You can obtain many "interesting" sounds by mixing partials which are not rational multiples of each other. For example, try adding the 4th-order and 7th-order partial, without a fundamental.

In case of waveforms without a fundamental, as in the preceding example, the resulting sound may not be in tune with "concert pitch". To fix this, each wave-table descriptor has an associated parameter called the "Oscillator Divide Ratio". The output frequency of the synth oscillator is divided by this factor so that the perceived pitch can be adjusted.

For a detailed description of the REMI synth wave-table oscillator algorithm, please refer to the document "Wave-Table-Oscillator-Notes" which is included in the synth firmware development kit.

There is practically an infinite number of combinations of amplitudes of 16 partials, hence it can be a tedious process to find just the right sound you're trying to make, or to find an appealing new sound that you haven't heard before. The inventor of the drawbar electric organ, Laurens Hammond, was probably attempting to alleviate this conundrum by simplifying the process of additive synthesis. A compromise was necessary in any case to design an instrument which could be manufactured and sold at a reasonable cost.

Hammond's idea was to limit the number of harmonics to 9, including the fundamental and two "sub-harmonics". The amplitude of each harmonic is selected by a slider control called a "drawbar" with 9 possible settings (marked 0 to 8). The drawbar scale is logarithmic -- each mark (except zero) is about 3 or 4 decibels below the next higher mark. Details of the fascinating Hammond organ technology are easy to find on the internet.

The REMI synth wave-table utility has a command option "`wav -h ...`" to populate the user wave-table with harmonics based on the Hammond drawbar principle. To obtain the first sub-harmonic (one octave below the fundamental), the oscillator frequency must be divided by 2. All of the partials contained in the wave-table are exactly double the actual frequencies of the harmonics generated by the synth oscillators.

Note that a simple wave-table oscillator cannot emulate a real Hammond organ accurately. This is because the tone-wheels which generate the harmonics in the organ are not precisely in synchronism. However, a synth with dual wave-table oscillators, one of which may be

detuned slightly relative to the other, can produce many sounds similar to a Hammond organ, particularly with some vibrato and reverb applied.

Drawbar organs use a numeric short-hand notation to represent drawbar settings, also known as the "registration". This is written as a set of 9 digits, each in the range 0 ~ 8.

For example, 00-8050-200 sets the fundamental to 8 (maximum = 0dB), the 3$^{rd}$ harmonic to 5 (approx. −12dB) and the 5$^{th}$ harmonic to 2 (approx. −24dB). All other harmonics have zero amplitude. This is a rough approximation of a descant recorder spectrum.

For help to use Hammond drawbar registrations to create a wave-table, simply type the command "**wav -h**" without any amplitude values, as follows...

```
> wav -h
Create a wave-table using Hammond organ drawbar settings.
Command usage:
wav -h  [arg1] [arg2] [arg3] ... [arg9]
   where <arg1>..<arg9> are drawbar settings (range 0..8)
The table peak level is scaled to the highest drawbar setting.
Set Osc.Freq.Divider = 2.0 in the wave-table descriptor.


  +-----------------------------------------------------------+
  | Drawbar # |  1  |  2  |  3 |  4 |  5 |  6 |  7 |  8 |  9 |
  | Harmonic  | 0.5 | 1.5 |  1 |  2 |  3 |  4 |  5 |  6 |  8 |
  | Partial # |  1  |  3  |  2 |  4 |  6 |  8 | 10 | 12 | 16 |
  +-----------------------------------------------------------+


>
```

Music scores written for organ usually show drawbar registration settings, sometimes in addition to conventional organ "stops", which should produce the intended sound. Such music scores provide a wealth of "tried and proven" registrations which you can use to create wave-tables for the REMI synth. The internet is also a good source of popular drawbar registrations covering a broad range of instrument sounds.

Perhaps the best way to learn how to use the wave-table creator utility is to experiment with the commands and listen to the resulting sound. You can use a MIDI controller to "play" the synth, or you can use the CLI "**sound**" command to test your created sounds.

By default (in the **sound** test patch), the audio output level is controlled by the Amplitude Envelope shaper. This can be changed using the command "**patch AC n**" to suit your MIDI controller. Set AC = 2 for a breath controller (EWI); AC = 3 for a keyboard without velocity sensing, or AC = 4 for a MIDI keyboard with velocity sensing.

Note:  This revision of the User Guide correlates to synth firmware version 2.3.55.