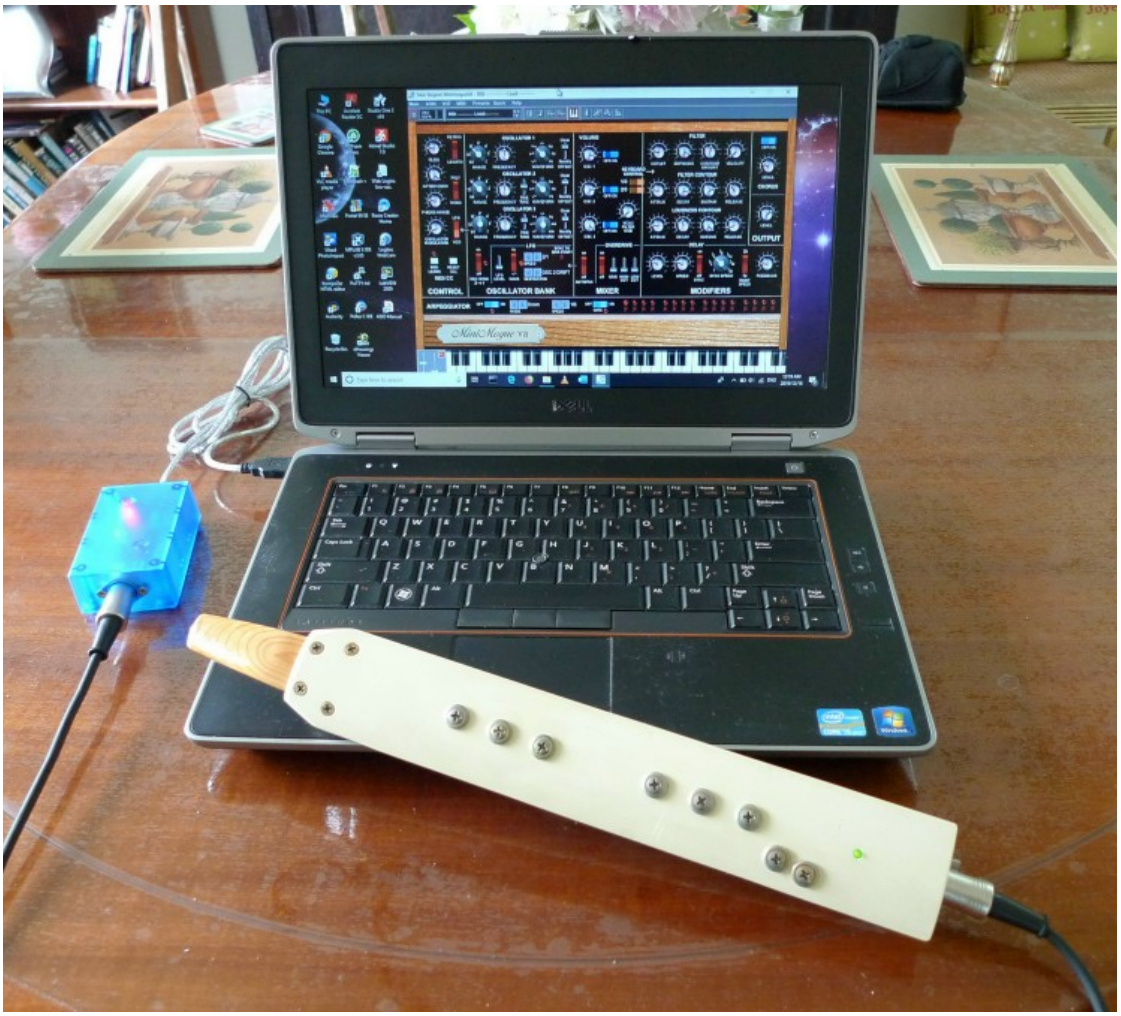
Build the "REMI 2"

Recorder-like Electronic Musical Instrument

(EWI MIDI Controller)

A DIY Project by M.J. Bauer

This web page describes the design and construction of the "second generation" (mk2) REMI handset.
For a general introduction to the project, see page: "Introducing the REMI".



REMI 'mk2' Handset with MIDI-USB adapter box

Overview

REMI 2 is an "electronic wind instrument" (EWI) with a choice of fingering schemes based on the recorder, including accurate emulation of modern English and German recorders. The "standard" REMI fingering is designed to be easier to learn, for beginners with no previous experience of wind instruments.

REMI 2 is simply a MIDI controller with standard MIDI OUT connection (5-pin DIN socket). It does not have a built-in sound synthesizer, so it must be plugged into an external MIDI sound module or keyboard synth with a classic MIDI IN socket. (If you prefer an "all-in-one" EWI design, take a look at the REMI 3.)

A low-cost MIDI-USB adapter (under $10) allows the REMI to be used with a "virtual synthesizer" (software application) running on a computer, as shown in the photo above. There are many freeware "virtual synth" app's available. (There are much better ones available at a price!)

A configuration setting called “Pitch Offset” is provided so that the range of musical notes produced by the REMI can be shifted, i.e. transposed, up or down by a number of semitones. Five special cases shown in the table below allow the REMI to emulate a variety of recorder pitches:

	Great Bass C3	Bass F3	Tenor C4	Alto / Treble F4	Soprano / Descant C5
Pitch Offset	–12	–7	0	+5	+12

Touch Pad(s)	OCT-	OCT- & LH3	OCT- & OCT+	OCT+ & LH1	OCT+

A USB/serial port allows a host PC to be connected to the REMI for setting up and testing. REMI firmware includes a command-line user interface (CLI) which is accessed using a "terminal" app (e.g. PuTTY) running on the host PC. CLI commands are provided for setting configuration options and for running diagnostics, etc.

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

For complete descriptions of REMI features, e.g. the CLI, fingering schemes, configuration options, hardware testing, sensor calibration, instrument presets, etc, please refer to the REMI 2 User Manual.

Note: The REMI is not a commercially available product. The hardware design and software source code are in the public domain. The design may be replicated by suitably skilled electronics hobbyists. Detailed step-by-step instructions and parts lists are not provided. The information published here may be used as a basis on which to design your own customized EWI.

REMI 2 Features

- Choice of recorder-like fingerings, incl. German
- Classic MIDI OUT connection (DIN-5 skt) with 5V power input
- Flexible configuration options for MIDI messaging, etc
- Pitch offset (transpose +/– 24 semitones)
- Select 1 of 8 synth presets directly from the handset
- Modulation control (force sensor operated by RH thumb)
- Pitch-bend control (using MMA8351 motion sensor - planned option)
- USB/serial port for setup and testing using PC terminal app
- Compact (350 x 45 x 35 mm) and light-weight (~200g)
- Very low parts cost (under AU$100, ~US$70)

Touch-pad Layout

There are two pad layout options to choose from. The firmware has a configuration parameter allowing selection of either option. The physical layout of touch pads for each of the two options is shown below.

Touch-pad layout for the "standard" (recorder-like) fingering schemes

Touch-pad layout for the "alternate" (generic EWI) fingering scheme

The first option (top picture) is more reminiscent of a recorder and should appeal to existing recorder players. The second option is a tad more abstracted from the recorder. An extra pad (LH4) operated by the left-hand little finger is added, but overall, it's a simpler scheme which is compatible with some of the commercial EWIs out there, for example, the amazing Sylphyo.

Details of fingering schemes (with fingering charts) can be found in the REMI 2 User Manual.

Handset Design

The REMI 2 handset design is based on a low-cost 8-bit microcontroller (PIC18FxxK22) with capacitive touch-sense inputs for the touch-pads. The device has a classic MIDI output connection (5-pin DIN), so it can be plugged into the 'MIDI IN' socket on the REMI mk2 Synth Module. The MIDI IN connector on the REMI synth module supplies 5V DC power to the handset. The REMI handset can connect to other MIDI sound modules, or to

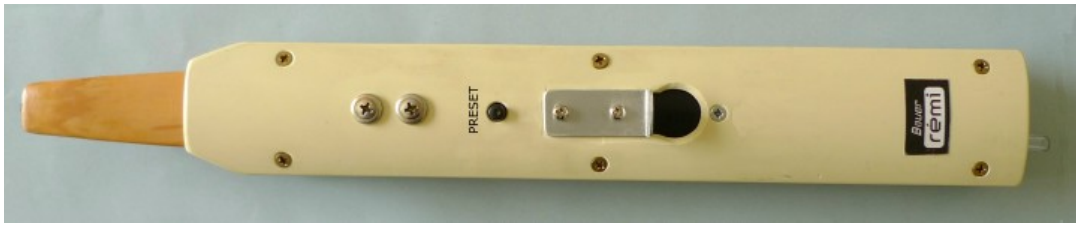
a computer, via a simple custom-made MIDI-USB adapter module.

The microcontroller board has a USB-serial bridge (FTDI FT232RL breakout board) allowing the handset to be connected to a PC running a terminal emulator app (e.g. "Putty"). The USB/serial "service port" provides a command-line user interface (CLI) to facilitate setting up and testing handset operation.

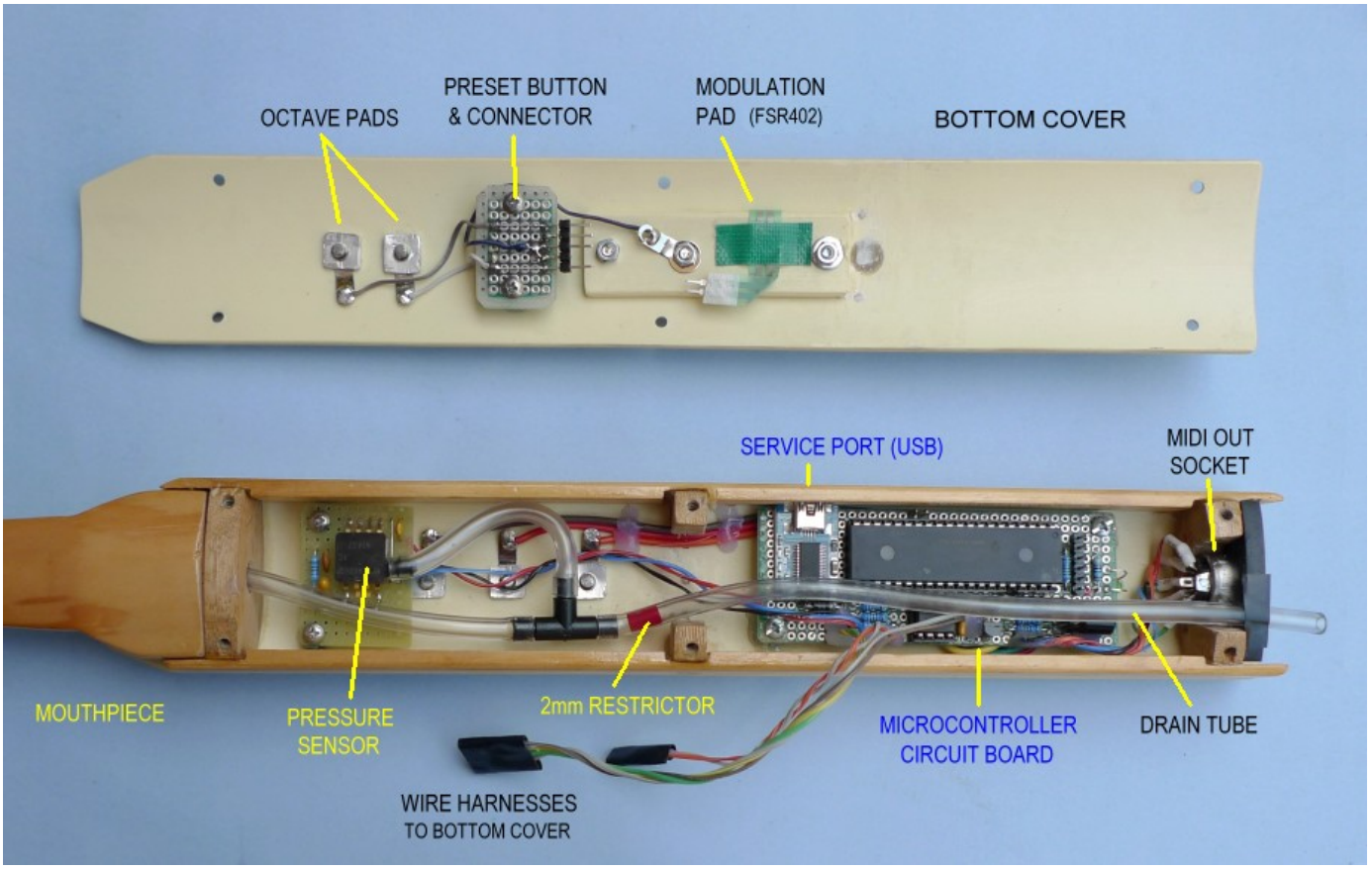
Construction

The prototype handset was made from two pieces of PVC plastic cut from a scrap of drain pipe, 110mm outside diameter, 3mm wall thickness. A suitable scrap may be found perhaps on a building site. The curved surface makes the plastic covers very rigid and strong. The upper piece has 3 wooden "ribs" (8~12mm thick) glued to it, allowing the bottom cover to be attached with self-tapping screws. Side strips of thin timber are glued to the ribs. The plastic surfaces are not painted, but rubbed with fine abrasive cloth (P400) to give a smooth matte finish. The result is a very "retro" (1950's) look and feel.

Top surface has 8 touch-pads and a status LED



Bottom cover has 2 octave pads, PRESET button and Modulation Pad
(The metal bracket near the Modulation Pad is a thumb rest.)



REMI handset mk2 -- Internal parts layout

The touch-pads are self-tapping screws with a broad flat head, cadmium-plated (I think). These are inserted through holes in the handset playing surface, fastened on the inside with aluminium retainers (~9mm square). A drill template and cross-section drawing are available for download -- see links at bottom of page.

Addendum (2022): Chicago screws, as used for leatherwork, book-binding, etc, make excellent touch-pads. These are available in a variety of metal finishes incuding brass, nickel, gun-metal, stainless-steel, etc. Size

required is 4mm (shank length) x 10mm diameter head. Available online at low cost. (Try AliExpress!)

The mouth-piece is carved out of soft-wood with a 6mm hole drilled through the middle. It should be coated with a water-based sealer* to prevent moisture absorption in the wood. The nylon tubing is inserted into the 6mm hole, all the way through to the tip, and held in place with a water-based sealer. The mouth-piece is attached to the upper plastic cover with 4 self-tappers, so that it is easily removeable.

*Warning: Do not use a paint or laquer with oil-based solvent on the mouthpiece, for 3 reasons... (1) it tastes terrible, (2) it could cause a health hazard, and (3) the vapours given off by the solvent could damage the Fluoro-silicone membrane in the pressure sensor. Recommended coating for the mouthpiece is water-based polyurethane or some other safe product used to varnish wooden toys.

Pressure Sensor "plumbing"

Tubing and other bits and pieces needed to make the airways inside the handset may be sourced from the garden irrigation section of your local hardware store. The prototype handset (pictured) used clear nylon tubing (3mm ID, approx 5.5mm OD) for the internal airways linking the mouth-piece, pressure sensor IC and the "drain tube". The T-joiner needed the barbs cut off to fit the 3mm tubing.

The sensor air inlet barb is slightly less than 3mm in diameter, which is a wee bit too small to make a good seal with the 3mm ID nylon tube. An easy solution is to fit a short bit of 2.5mm (nominal diameter) heat-shrink tubing over the barb, shrink it with a hot air blower or whatever (taking care not to melt the sensor!), then fit the 3mm nylon tube over it.

A drain tube is recommended because moisture condensation occurs inside the airways and it is probably sensible to provide an exit for the moisture. Also, the drain tube allows air-flow, which is preferable to a sealed system for playability. However, the exit air flow needs to be restricted somewhat to produce a sufficient range of pressure inside the sensor. This can be achieved by fitting into the drain-tube a small plug with a 2mm hole through it, perhaps cut from a bit of plastic insulation sleeving if you can find some of suitable size (3mm OD, 2mm ID). The best place to fit the restrictor is close to the T-joiner, to minimise moisture condensation.

It is also recommended to mount the pressure sensor above the T-joiner, i.e. close to the mouthpiece, to prevent moisture from accumulating in the sensor, as shown in the picture above.

Modulation Pad

The simplest option is to use a force-sensitive resistor (FSR-402). This will give a uni-polar output, which is not ideal for a Pitch Bend controller, but quite satisfactory as an "Effect Controller" (e.g. vibrato depth, timbre modulation, etc). The FSR is mounted behind a round cutout in the bottom cover of the handset. A plastic "backing support", made from the same material as the covers, is screwed to the inside of the bottom cover. A piece of black PVC insulation tape is stuck over the top of the FSR402 to keep it in place. The right-hand thumb pushes the FSR against the backing support.

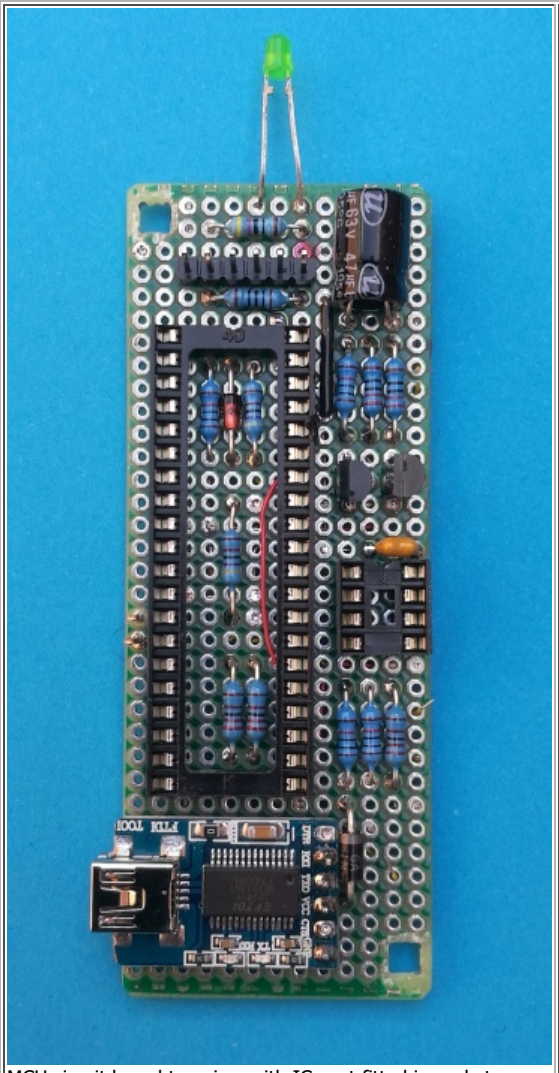
Note: The FSR402 is fragile. Take care not to damage the flex strip. Don't bend it too tightly. Insert thin washers between the handset body (bottom cover) and the FSR support piece to prevent the FSR edges from being compressed hard.

Modulation Pad (FSR402) resting on its backing support piece

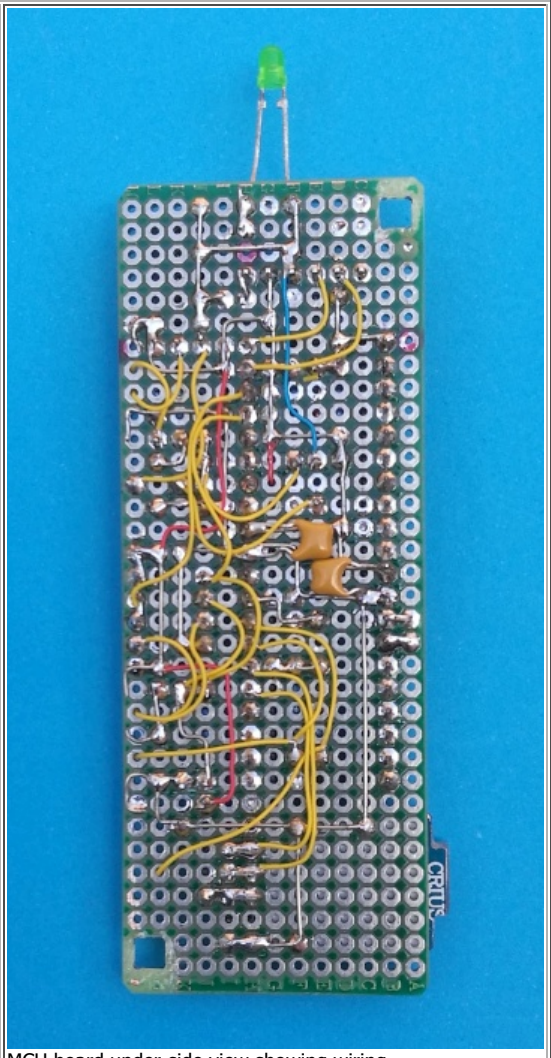
Electronics

The handset microcontroller circuit board is constructed on a piece of prototyping board measuring 90 x 35

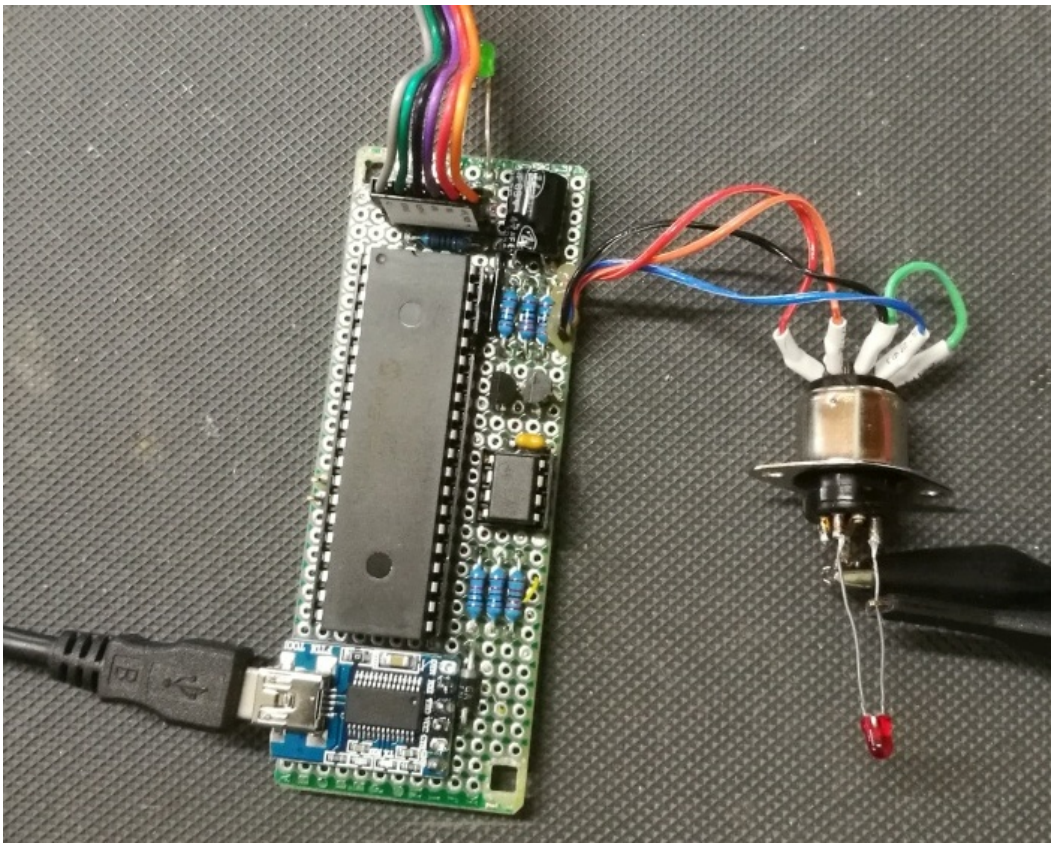
mm. A 40-pin DIL socket for the PIC18F45K22 MCU chip takes up most of the board area. A 6-pin SIL header for connection of the MCU programming tool (PICkit 3) is located at the top end of the board. The USB-Serial bridge (FT232RL breakout board) is located at the bottom end of the board. The MIDI transmitter and analogue sensor signal conditioning components (LM358, etc) are placed beside the MCU socket.



MCU circuit board top view with ICs not fitted in sockets. The space inside the 40-pin MCU socket accommodates 5 resistors and the 8.2V zener diode.



MCU board under-side view showing wiring. Leads to touch-pads, MIDI socket, sensors and other off-board parts are soldered directly onto the board.



Assembled microcontroller board under test.

The picture above shows the MIDI OUT socket wired to the MCU board. The socket has a plug inserted with a LED wired between the OUT+ and OUT- pins. An oscilloscope was used to test the performance of the MIDI transmitter circuit, in particular for compliant rise and fall times in the output signal. The USB "service port" provides diagnostics for hardware testing.

The USB/Serial bridge (FT232 breakout board) supplies 5V power to the REMI MCU (PIC18) when using the "service port" CLI, so there is no need to plug in a MIDI cable at the same time. There are many different types of USB/Serial modules available. Be sure to buy a device that outputs 5V on the Vcc pin and has 5V logic levels on the TX and RX signal lines. (Some devices are configurable for either 5V or 3.3V operation.)

Note: These pictures are provided as a rough guide to makers intending to build a REMI handset. Don't rely on the photo of the underside of the board as a guide to wiring. It's best to follow closely the schematic diagram (see download links below) and check your wiring carefully before applying power. It is also recommended to test MCU board operation before wiring it into the handset (as shown in the above photo). After soldering leads to off-board parts, e.g. MIDI socket, pressure sensor, modulation pad, PRESET button and touch-pads, apply hot-melt glue to the board around the entry points of leads to prevent them from breaking off.

Breath Pressure Sensor

The prefered sensor type is the MPXV5010-GP, but this part is now very expensive (> AU$40) and hard to obtain. There are other options for the pressure sensor, easier to find and at lower cost. Whatever your choice of breath sensor, the essential requirement is to produce an output signal of sufficient magnitude to feed into an ADC input on the micro-controller. The ADC input voltage span, i.e. difference between minimum and maximum pressure, should be about 3V ideally, but at least 1 volt. The voltage at minimum pressure (i.e. without blowing) should be below 1V.

There is a sensor device type MPS20N0040D available on AliExpress for about A$3. This device has a full-scale pressure range of 40kPa, which is much higher than required for an EWI, but it should work OK with appropriate signal conditioning circuitry. A high-gain differential amplifier is required. A suitable circuit is described on another website here: Lang Blogspot. (More details to come.)

For use in the REMI, the sensitivity (amplifier gain) must be increased by a factor of 5 or thereabouts. To do this, change the value of R8 (feedback resistor of 2nd op-amp) to 12k. If the output voltage is higher than 1V with zero pressure, you'll need to add a small offset voltage to the non-inverting input of the first op-amp (junction of R4 & R6). [Send me an email if you need help.]

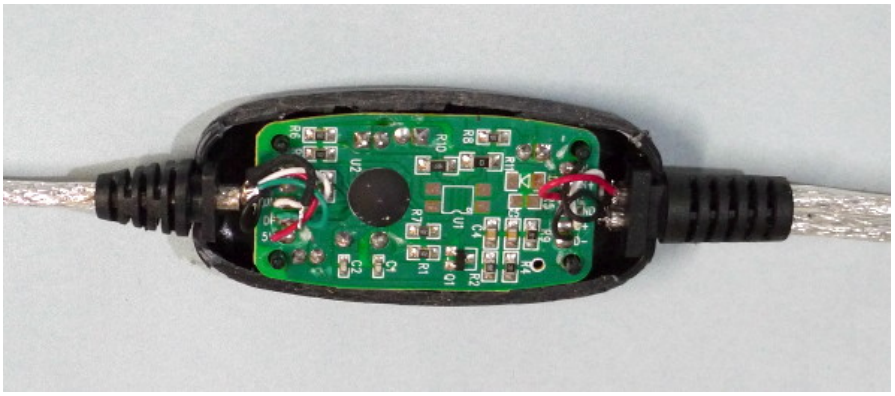
PS: I'm working on a very low cost air-flow sensor design using an optical proximity sensor. These will be fiddly to construct, so I will make one for any REMI maker who wants to try this option.
(More details to come... watch this space!)

Instructions for hardware testing and sensor calibration are given in the REMI 2 Handset User Manual.

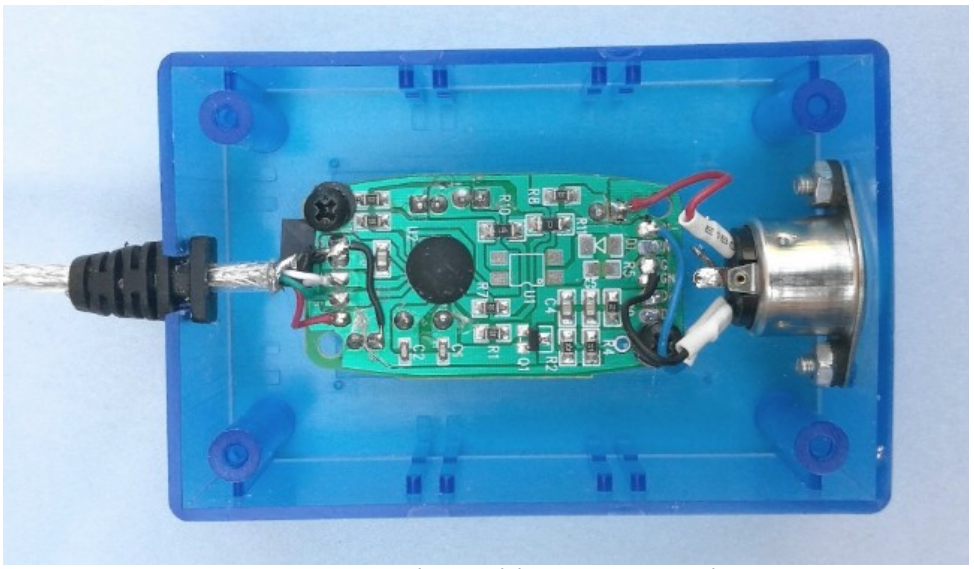
MIDI-USB Adapter box

The REMI MIDI-USB adapter is simply a commercial device re-fitted and re-wired into a new housing. The adapter pictured below is readily available at very low cost (under A$10) from online suppliers.

The original adapter casing can be easily broken apart by cutting around the join with a hacksaw or utility knife, taking care not to damage the circuit board inside. The two MIDI cables with DIN-5 plugs on the end are unsoldered from the circuit board and discarded. The USB cable and connections are kept intact.



The circuit board is removed from its casing and mounted in a blue translucent box together with a 5-pin panel-mount DIN socket. A couple of plastic standoffs are glued into the box to hold the circuit board in place. A translucent box was chosen to allow the LED indicators to be visible through the top of the box without needing to drill holes for them. The assembled MIDI-USB adapter is pictured below. These photos were taken before optional MIDI THRU and MIDI OUT connectors were fitted.



REMI MIDI-USB adapter with bottom cover removed

REMI MIDI-USB adapter - DIN socket view

MIDI-USB adapter - USB cable entry

The table below gives wiring details for the DIN-5 socket. In the MIDI 1.0 specification, pins 1 and 3 are not allocated for any purpose. The REMI MIDI-USB adapter uses pins 1 and 3 to supply 5V DC power to the handset. Pin 4 is usually required for MIDI IN+ which connects to the anode of an opto-coupler LED. These cheap Chinese MIDI-USB adapters omit the opto-coupler, although there is provision for one on the PCB (U1). Hence, pin4 (OUT+ on the DIN-5 skt) is not connected to IN+ on the adapter board.

The table also shows connections for an optional 'MIDI THRU' socket. This may be another DIN-5 type, or a 2.5mm "micro" TRS type, as fitted on some low-end mini synthesizers. Provision of a MIDI THRU connector allows the REMI to operate with any synth having a classic MIDI IN socket, for example a keyboard synth. The USB cable supplies power to the REMI from any regular 5V USB power pack.

MIDI IN DIN5 pin	MIDI THRU 2.5mm TRS skt	MIDI-USB Adapter PCB
1	-	+5V
2	SLEEVE (SCN)	-
3		GND
4	RING (OUT+)	-
5	TIP (OUT-)	IN-

NB: Since a standard MIDI connection requires only pins 4 and 5 to be wired, some ready-made MIDI cables may have only 2 cores inside (plus screen = pin 2). Be sure to buy (or make) a MIDI cable with all 4 cores (plus screen) to use with the MIDI-USB adapter box and REMI handset.

Firmware

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 configuration parameters are provided so that it is quick and easy to switch control between two of your favorite synthesizers or sound modules. The required synth parameter set is selected by the 'Mode' switch. [Note: Provision of a 'Mode' switch is optional. If your REMI omits the switch, it is recommended to tie the switch input pin to GND. The Mode switch function is supported by firmware version 1.3 and later.]

Configuration parameters are stored in non-volatile memory, so their values will be preserved when the REMI is powered off. Be aware that after a firmware update, however, all parameters will revert to default values, so it will be necessary to re-enter any different values required for your particular REMI and synth setup.

Service Port CLI

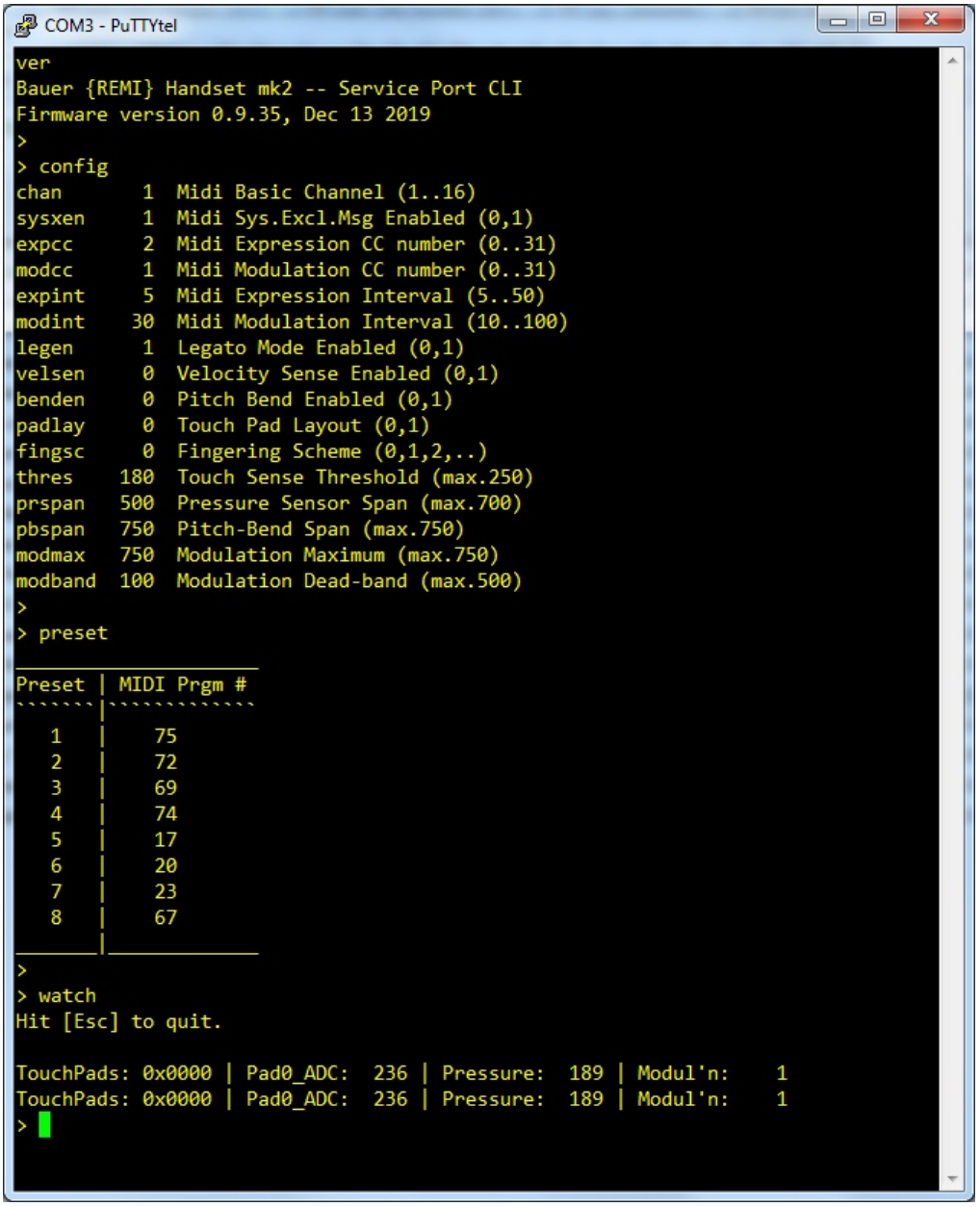
The REMI provides a "command-line user interface" (CLI), accessed via a computer connected to the USB "service port" (FT232 USB/serial bridge) on the REMI micro-controller board. The host computer needs a terminal emulator program (e.g. "PuTTY") installed.

Various CLI commands are provided to view and set configuration parameters, to view and set instrument preset assignments and for hardware testing, sensor calibration, etc.

Details of service port CLI operation are given in the REMI 2 User Manual.

The screen-shot below shows CLI dialogue for a few of the commands, i.e. "ver" (version), "config" (lists user-settable configuration parameters), "preset" (lists Preset MIDI channel voice assignments) and "watch" (outputs data from touch-pads, pressure sensor and modulation pad in real time, for setup and testing).

Note: This screen-shot was taken with an early firmware version. Some parameters have since changed.



Programming Tools

A PIC programming tool, e.g. Microchip PICkit-3, is required to install the REMI application firmware. Low-cost PICkit-X clones are available from online suppliers via AliExpress, eBay, etc.

REMI firmware is built using Microchip PIC development tools - MPLAB.X IDE with XC8 compiler - free to download from Microchip's website. If you intend to modify or extend the firmware, you will need these tools. Otherwise, you just need to install the PIC programmer application (IPE) on your computer. Please refer to Microchip documentation for instructions to use the programming tool.

Downloads & Links

REMI 2 Handset schematic

Drill Template drawing

Body cross-section drawing

Firmware Repository
(Source code, hex files, MPLAB.X project, etc)

REMI 2 User Manual

If you are interested in building a REMI and/or if you have enjoyed following the project here, kindly send me an email. Technical support is offerred to readers who wish to build a REMI or similar EWI controller.