

## **Sonoclast Plastic Pitch Pro User Manual v0.5**

(Obviously this *manual* is a work in progress. The Plastic Pitch Pro is a complex and unique device with a lot of features and thus demands a lot of explanation. Please do let us know if you have questions.)

### **Basic Description**

The Plastic Pitch Pro (PPPro) is a high resolution MIDI microtuner with a musically intuitive real time tuning interface. It can tune any MIDI synthesizer using MIDI Pitch Bend messages, MIDI Tuning Standard (MTS) messages, and MIDI Polyphonic Expression (MPE) messages.

The basic idea is that the PPPro processes incoming MIDI and generates custom tuned outgoing MIDI. The incoming MIDI comes from one or more MIDI controllers, and the outgoing MIDI is sent to one or more MIDI synthesizers.

### **Connecting Your PPPro**

The PPPro can be connected to multiple controllers and multiple synthesizers. To start, below are the minimal connections to make it work.

For Pitch Bend and MPE tuning message types:  
MIDI controller output > MIDI or USB cable > PPPro input  
PPPro output > MIDI or USB cable > MIDI synthesizer input

For MTS tuning message types:  
PPPro output > MIDI or USB cable > MIDI synthesizer input

Note that a USB port is both an input and output, whereas a MIDI DIN port is either an input or an output.

A powered USB hub can be used with the PPPro's USB A (host) port to add three more (four total) USB A controller inputs.

In sum, there are up to six MIDI controller input ports:

- 1 USB B (most likely from a computer)
- 4 USB A (with a powered hub or 1 without)
- 1 DIN

And there are up to four MIDI output ports to synthesizers:

- 1 USB B (most likely to a computer)

- 1 USB A
- 2 DIN

## **Hardware Description**

14 high quality TT Electronics P260 potentiometers. They are not the cheapest but are very long lasting—rated at 1 million cycles (i.e., a lot more than most synths on the market)—and have a sturdy and consistently smooth feel which is especially important for tuning. The knobs have a decently large diameter and are thus suitable for fine tuning control. 16-bit ADCs, also not the cheapest and not used in most MIDI controllers, are used to allow for tuning at a one-cent resolution even over a large pitch range.

2 Sanwa Denshi arcade buttons. Sturdy and fun to press.

1 Bourns rotary encoder paired with a nice grippy knob.

160 character display that shows real time tuning at 1-cent resolution among other things.

MIDI I/O via 1 USB A (host) jack, 1 USB B (device) jack, 1 MIDI input DIN, 2 MIDI output DINs.

1 micro SD card for Scala (.scl) files, presets (also .scl files), and storing the system configuration from session to session.

1 power jack.

1 power switch.

## **User Interface Overview**

The twelve black and white knobs correspond to each note in an octave, and each tune flat or sharp the respective note across all octaves. Each knob has a wide zone at 12 o'clock to make tuning to neutral easier.

The two grey knobs set the width and range. The width knob sets the width of an octave of keys from 600 cents to 2400 cents. There are wide zones at 600, 1200, 1901.955, and 2400 cents.

The range knob sets the range of the twelve black and white knobs from 0% to 100%. 0% means the twelve knobs are all neutralized and your scale is equal temperament. 100% means the twelve knobs have their full range as defined in the global settings (via the menu system) ranging from +/-100 to +/-700 in 100 cent increments. So the range knob lets you, for example, scale back the tuning toward equal temperament and easily return to where you started.

The fat rotary encoder knob and two buttons are used to navigate what is displayed. Generally, the encoder is used to increment or decrement values, the back button is used to go back, and the enter button is used to go forward and access the configuration menu system.

On the main 12-tone tuning display (with all the real-time tuning information), the back button lets you toggle between relative and absolute cents. Relative cents show the tuning of each note relative to its neutral position, whereas absolute cents show the tuning of every note relative to the root tuning note. The root tuning note can be changed with the rotary encoder.

## **Supported Features**

### **3 Tuning Modes:**

Make custom 12-tone tunings using the real time tuning knobs.

Play presets / Scala (.scl) tunings. Custom tunings can be saved as presets in the Scala (.scl) format. Scala tunings downloaded from the Internet can also be played.

Router mode allows you to bypass the PPPro tuning functionality and play your synth without reconfiguring your setup.

12-tone tuning mode allows you to tune each note across all octaves independently. All tuning is at a one cent resolution. The “octave” width can be tuned from 600 cents to 2400 cents. The range of the tuning knobs can be dialed back to zero which allows for getting back to “neutral” among other things. The root tuning note is set with the rotary encoder. The BACK button toggles the display between showing the relative cents tuning and the absolute cents tuning (relative to the root note). The ENTER button takes you to the configuration menu. In that menu, you can choose to save the current tuning as a preset. Presets are saved in the Scala (.scl) format to the micro SD card and can be copied to your computer to use with other Scala things.

Play presets mode lets you play Scala (.scl) tunings. The Scala tunings could be ones downloaded from the Internet and copied to your micro SD card, or they could be presets created and saved with your PPPro. In this mode you can load any Scala tuning stored on your micro SD card. While playing a tuning, you can modulate the root tuning note either relative to the current Scala tuning or relative to the absolute MIDI note. The modulation can be performed with the rotary encoder and buttons, and it can also be performed with a MIDI controller on a dedicated channel. In this mode you can also load the current preset into 12-tone tuning mode (see above). When loading a scale with less than 12 tones, the PPPro extrapolates to make a 12-tone scale. When loading a scale with more than 12 tones, the PPPro ignores tones beyond the first 12.

Router mode allows you to bypass the PPPro tuning functionality and just route MIDI messages as configured (see below). This is useful for playing your MIDI synthesizer without having to unplug cables and reconfigure settings.

## **Configuration Menu:**

The configuration menu gives access to settings organized by each MIDI output port, global settings, and MIDI utilities.

### **MIDI Output Port Settings:**

There are four output ports labeled DIN A, DIN B, USB A (host), and USB B (device). Each has the following identical set of configuration parameters. Note that some parameters are only accessible when compatible options are selected.

**MIDI Channel In** is a unique MIDI channel for messages received by the PPPro that will route out the selected output port. You can select any unique channel from 1-16 or OFF. If you are not using this output port, it's recommended that you set it to OFF to avoid unnecessary processing and/or sending of MIDI messages.

**Channel Out** is the base MIDI channel (1-16) to route messages out to a synthesizer to be tuned and/or controlled some other way. Make sure the synth on the receiving end is open to the corresponding channel(s).

**Message Type** is the type of MIDI tuning message to use. Three ways are supported: MIDI pitch bend (PB), MIDI Tuning Standard SysEx (MTS), and MIDI Polyphonic Expression (MPE). Each type has its own pros and cons. See the Caveats section below for more on this.

**Number of Channels** is the number of MIDI channels (1-16 for PB, 1-15 for MPE) to use starting with the base Channel Out (see above).

**Mono Retrig** is the type of monophonic retriggering and can be high, low, or last note.

**MPE Zone** can be lower or higher. (See the MIDI MPE spec for more details on what that means.)

**MTS Preset** can be 0-127. I have not yet seen this actually used in the wild, so it's fine to leave it set to 0 unless you know of a reason not to.

**MTS Bank** can be 0-127 or None. There are different MTS message formats with a bank and without a bank. Setting this to 0-127 uses the bank message format and sends the bank, and setting to None uses the message format without a bank. Again, I have not yet seen this used in the wild, so 0 or None are fine depending on what MTS format you want.

**MTS Size** can be Std (Standard) or Moog. The MTS specification states that 100 cents are to be spanned by 16,384 equal tuning steps. In my experimentation with the Moog Sub37, I have found that it does not follow the specification and 100 cents seem to be spanned by 8,192 steps

(or half the standard size). Selecting Std means the PPPro uses 16,384 steps in MTS tuning calculations, and Moog means it uses 8,192 steps.

### **Global Settings:**

The **maximum range** of the 12 tuning knobs can be set from 100 cents to 700 cents. This is how much sharp or flat each note of an octave can be changed with a real time tuning knob.

The **tuning mode** can be set to tune in real time or on the next played note. When tuning in real time, any sounding notes change pitch immediately. When tuning on the next played note, any sounding notes will keep the same pitch until the next time they are played.

The **MIDI channels** used for modulating the root tuning note when playing presets (see above) can be set to a unique value (1 to 16). One channel is for modulating relative to the current Scala tuning, and another is for modulating to an absolute MIDI note.

The **minimum note** and **maximum note** used for MTS tuning can be set. Setting to a smaller range makes for smaller MTS messages and lower MIDI latency.

Many types of MIDI messages are routed by incoming channel and echoed thru the PPPro by default according to the settings for each output port (see above). SysEx messages do not have a channel, so they cannot be routed in the same way. The **SysEx Thru** option allows SysEx messages to be configured to echo out only one of the four output ports or not be echoed out at all.

Finally, there is an option to **recalibrate** the 14 real time tuning potentiometers. If you select this option, you will be instructed to turn every pot fully clockwise and then press ENTER. You can also press BACK to cancel the recalibration. The recalibration happens very quickly. It is performed at the factory, so it should not need to be repeated often if at all. The sign that the pots are not calibrated is that neutral tuning is not at 12 o'clock.

### **MIDI Utilities:**

The MIDI utilities allow you to send program change messages, perform an MTS Tuning Dump, send All Notes Off MIDI messages, and send the pitch bend range RPN.

When an output port is configured to use the MIDI pitch bend tuning type, it's helpful to be able to **send program change messages** to all the MIDI channels at once to change patch. Otherwise you might have to manually browse patches 16 times for 16 channels.

An **MTS Tuning Dump** lets you write a tuning to a synth that supports it. This means you can store a tuning you create with the PPPro on the synth itself and use the tuning without the PPPro. When in 12-tone mode, the tuning dump is for the current 12-tone tuning. When in Preset mode, the tuning dump is the preset tuning.

**All Notes Off** is useful if you have MIDI message hiccups and a Note On message never gets turned off.

**MIDI RPN** (Registered Parameter Number) 0 is used to configure the pitch bend range in a synth. The PPPro sends RPN 0 whenever pitch bend range is changed, but if your synth is not connected with a MIDI cable or turned off you might want to send the RPN again. This lets you do that.

### **Notes and Caveats (In No Particular Order)**

Obviously the PPPro has a limited (albeit decently large) amount of processing power and limited available bandwidth for MIDI. Since the PPPro is designed for maximal configurability and tuning of multiple synths with multiple controllers, it is possible to overload it. When using it you should consider how to minimize the extraneous MIDI you send into it. For example, turn off MIDI clock from your controller (assuming you don't need it). Another example--the Yamaha MODX by default strangely sends MIDI Note messages on multiple channels at a time for a single key press.

Using MTS to tune requires relatively large MIDI SysEx messages, so you may encounter noticeable latency when tuning with MTS. In particular, the Range and Width knobs affect the tuning of all MIDI notes, so they generate particularly large MIDI SysEx messages. (They can be made smaller by setting the minimum and maximum notes detailed above.)

Tuning in real time versus on the next note with MTS SysEx messages depends on the synthesizer's handling of MTS. So it may or may not work as configured in the PPPro. For example, the Moog Sub 37 supports MTS tuning but can only tune in real time.

Whereas MTS and MPE are supported by a minority of synthesizers (as of 2021), MIDI pitch bend is supported by most if not all MIDI synthesizers. However, tuning with pitch bend has its limitations. In order to tune polyphonically (i.e., with more than one tuned note sounding at a time), notes must be sent to their own MIDI channels. This works well with multitimbral synths by assigning a duplicate voice to every channel, but monotimbral synths can generally only handle pitch bend tuning one note at a time. (They still may work with MTS or MPE.)

When playing Scala/preset files, the PPPro assumes that the synth is tuned to 12-ET. In other words, the PPPro cannot account for your synth being set to an alternate tuning.

When a Scala/preset file is loaded into 12-tone mode for editing, the PPPro does its best to map the Scala file. Besides forcing every scale into 12-tones, there are other limits such as: the scale width can only be between 600 and 2400 cents, and the maximum tuning range per note is +/-700 cents.

Cents and Hertz values that are displayed on the PPPro are merely calculated based on knob positions or the currently loaded preset, and they assume that your synth is tuned to 12-ET with MIDI note 69 tuned to A-440. The PPPro has no way of knowing what actual frequencies are produced when your synth receives a MIDI message that tunes.

When tuning with MIDI pitch bend or MPE, the polyphony is limited to one less than the number of channels configured for the output port being used. For example, if the port is configured for 10 channels, then only 9 voices are played at once. If you exceed the possible polyphony, then the oldest voice is dropped. Polyphony is limited to one less because, for minimal latency, there needs to be a channel available for the next voice to use.

Monophonic retriggering depends on the implementation in the synthesizer, and there is no single official way that it gets implemented. The PPPro attempts to do monophonic retriggering, but some synths may not play along.

The PPPro outputs MIDI RPN 0 messages to set the pitch bend range of a synthesizer. For 12-tone mode, to maximize real time tuning possibilities, the pitch bend range is set to 3 times the "maximum range" as set in the Global Settings (see above). For Preset mode, to maximize tuning resolution, the pitch bend range is set to (1 times) the "maximum range." Some synthesizers do not support RPN 0, so they would need their pitch bend range manually set whenever you change the "maximum range" setting and/or the operating mode. For example, if the "maximum range" is set to 200 cents (or 2 semitones) in 12-tone mode, then the pitch bend range of the synthesizer should be set to  $3 \times 2 = (+/-)6$  semitones.

The micro SD card contains a PPPRO directory. All Scala tunings should be placed in that directory. Any presets saved with the PPPro are saved in that directory as PPPROXX.SCL where XX is a number from 00 to 99. You should use your computer to rename the preset files, modify the description to something meaningful, and back them up. The filename and description are shown when browsing files on the PPPro. Also in the PPPRO directory is a file called PPPRO.DAT that contains all the settings configured on your PPPro. If you delete it, your PPPro will boot up with default settings and generate a new PPPRO.DAT file. If you boot your PPPro without a micro SD card, your PPPro will also start with default settings, and you obviously will not be able to play Scala presets. If your micro SD card is lost or damaged, you can format a new one with FAT32 on your computer and then boot your PPPro with it. The PPPro will create the PPPRO directory and add a PPPRO.DAT file with default settings. (You will need to manually reconfigure your PPPro, but there are not that many settings.)

The firmware is updateable using the Teensy Loader Application (<https://www.pjrc.com/teensy/loader.html>) and the small hole on the bottom of the case. More detailed instructions to be provided if/when firmware updates are needed.

**Manual Revisions:**

v0.5 - released May 2021

**Limited Warranty:**

Sonoclast warrants the Plastic Pitch Pro to the original owner to be free from defect or workmanship for 1 year from the date of purchase.