

# The picoTracker User Manual

---

# Table of contents

The picoTracker User Manual .....	1
Welcome .....	3
Getting Started .....	5
Device Management .....	8
Project Management .....	12
Song Screen .....	14
Chain Screen .....	18
Phrase Screen .....	20
Instruments .....	22
Grooves .....	29
Tables .....	32
Renders and Stems .....	38
The Mixer .....	41
Tips and Tricks .....	43
Controls & Moves .....	47
FX Commands Reference .....	69
Available Scales .....	75
MIDI Implementation .....	77
MIDI Implementation .....	77
picoTracker Remote User Interface Protocol .....	79
Index .....	82

# Welcome

---

Welcome to the picoTracker Advance, a small, portable music making device based on the tracker workflow.

But what is a "tracker"?

Trackers are music production applications that started life on 16bit home computers such as the legendary Amiga 500.

Trackers later evolved to also run on devices such as the Gameboy as popularised by LSDJ and this then inspired the LittleGPTracker (LGPT) which was designed to run on portable Linux-based gaming handheld devices. LGPT then served as the starting point for the firmware for the picoTracker, though it has now substantially diverged from those roots, being customised for the specific hardware of the picoTracker Advance and gaining many new features and improvements.

## Overview

picoTracker Advance is the evolution of the original picoTracker – a portable music instrument built around the classical tracker interface. It currently supports eight stereo channels, with per-step assignable sample and MIDI instruments. The new hardware is designed to enable new creative possibilities as the firmware continues to evolve.

Your picoTracker Advance comes included with a:

- 16GB SD Card with sample projects, samples and themes
- MIDI TRS adapter (type A)
- quick reference card

(note: Does NOT include power supply or USB-C cable. Uses any standard USB-C charger)

## Hardware specs

- HiDPI 720x720 4" display
- Powerful ARM processor
- 48MB of sample memory
- MicroSD for project and sample storage
- In/Out TRS and USB MIDI
- Headphone/Line output and speaker
- Line input and microphone
- Battery lasts up to 6 hours

- Weight: 233g (8.2oz)
- Dimensions: 143x78x15mm (5.6x3x0.6in)

## Acknowledgements

As with any big open source undertaking, this project would not have been possible without standing on the shoulders of many great open source projects and libraries. We would especially like to thank [Marc Resibois aka Nostromo](#), the creator of LittleGPTacker and making it available under a liberal open source license.

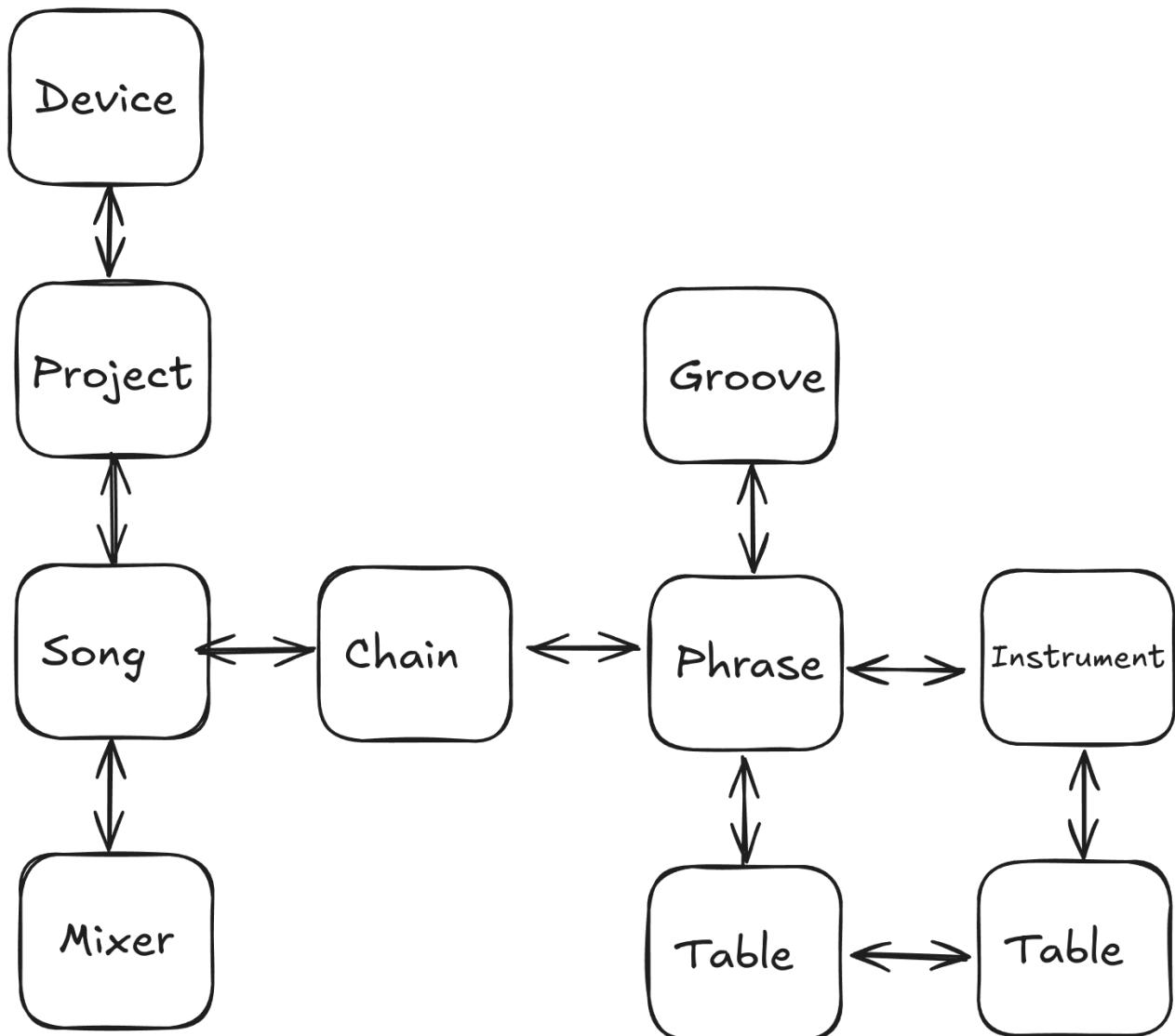
# Getting Started

---

## Navigation

The picoTracker user interface is made up of a series of screens, which are layed out in a "map" which is represented at the bottom right of each screen with a single letter representing each screen, along with highlighting with a different color which screen is currently being displayed on screen. For example below you can see the screen map with the Song screen (S) shown as being currently displayed.

## Screen Map



The picoTracker boots into the Song screen of your most recently opened project. Press and hold ENTER while restarting to instead launch a new, untitled project instead.

The picoTrackers keypad layout resembles a typical old console game controller.

You can navigate (aka move) between screens using ENTER+DOWN/UP/LEFT/RIGHT .

To get to the chain screen, you need to have your cursor on a chain in the song. To get to the phrase screen, you need to have your cursor on a pattern in the chain screen.

The names of the keys are shown below.



**Up, Down, Left, Right**  
Used for navigation, changing values



**Enter** enter new values, modify values



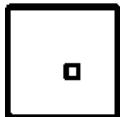
**Edit** edit functions, pg up/down,



**Play** play, stop, rec, live funcs



**Alt** Modifier key primarily used with non-directional keys



**Nav** Modifier key primarily used with directional keys for screen navigation

## Playback Modes and Controls

When you are on the Song screen, there are two modes for playback, Song Mode and Live Mode. The controls in each mode differ slightly.

You can switch between the modes by hitting ENTER+LEFT/RIGHT in the Song screen.

For a list of all the controls, see the [Controls & Moves reference](#) .

## File Browser

When selecting a sample wav file or project file you enter the file browser view.

When in the Project File Browser view, use the arrow keys UP/Down to navigate through the list of project available and press EDIT to open the currently selected (highlighted) a project.

When in the wave sample File Browser, use the arrow keys UP/DOWN to navigate through the list of available sample files and subdirectories, subdirectories are indicated with a / prefix. Press EDIT to enter a subdirectory, you can go back to the parent directory by navigating to the /.. entry and pressing ENTER . Press PLAY to audition the currently selected sample wave file. To import the currently selected wave file press ALT+PLAY . At any time, you can return to the instrument screen from the sample file browser by pressing NAV+LEFT .

## **Shutdown**

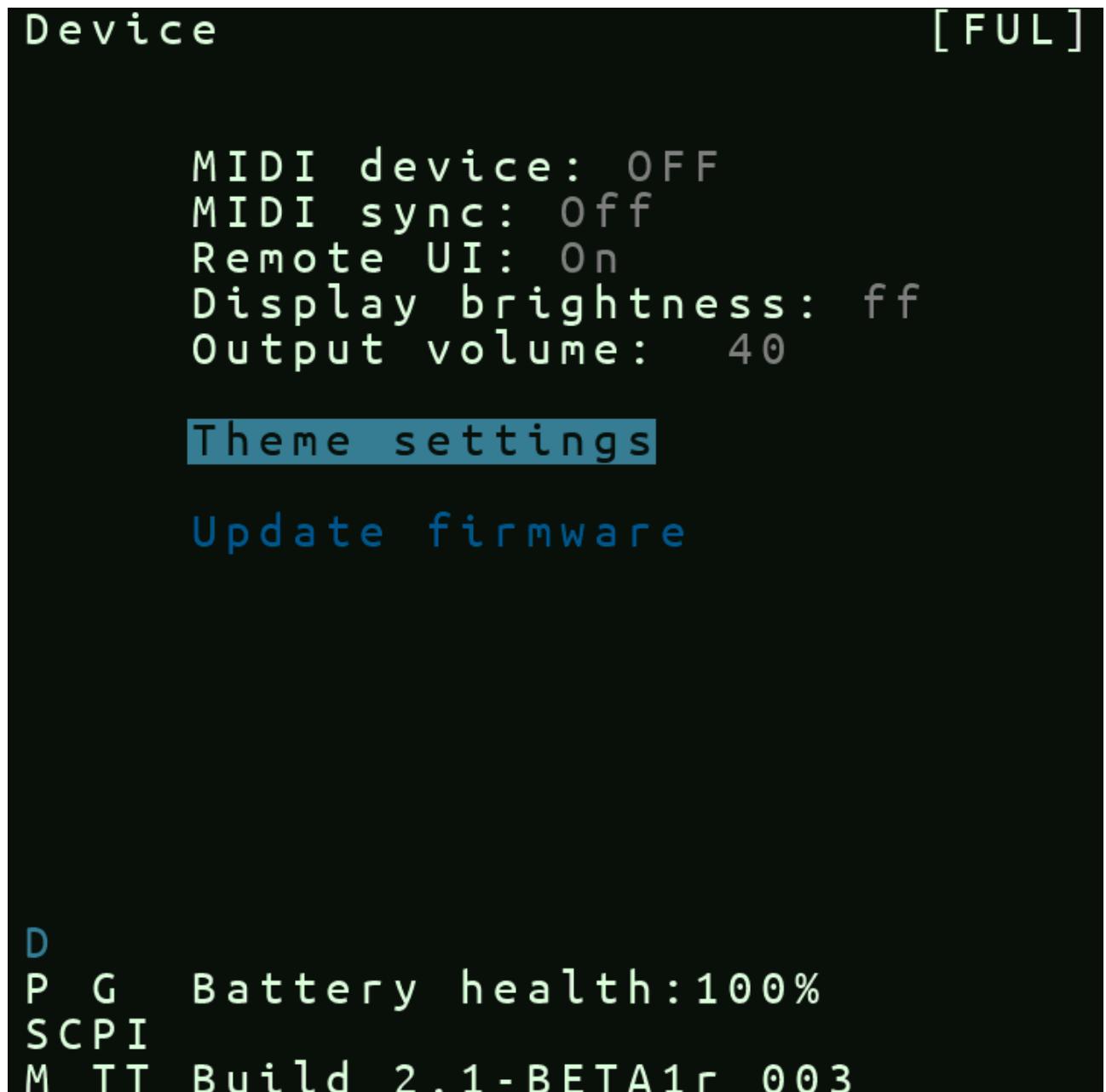
You can just turn off the picoTraceker at any time you want other than when saving a project (which is an extremely brief moment). **Save your work first if you care about it.**

# Device Management

---

The device screen allows you to manage settings for the picoTracker device hardware itself.

**Note:** the picoTracker saves its device settings to the SDCard so if you replace the sdcard in your picoTracker, the device configuration will reset to factory defaults.



MIDI

- **MIDI device:** Lists the picoTracker MIDI interfaces available. Available options are:
  - None
  - TRS
  - USB
  - TRS+USB
- **MIDI sync:** Enable/disable MIDI clock sync messages being sent by the picoTracker.

### **Line Out Mode (1st Edition RP2040 model only)**

On the first edition picoTracker, the audio output can be set to one of 3 amplification levels to make the audio level output more useful for Line Level output or Headphone output. The 3 possible settings are:

- HP Low
- HP High
- Line Level

**NOTE: A reboot of the picoTracker is required to apply the newly changed audio output level!**

*\*PLEASE exercise caution when using earphones or headphones!*

### **Remote UI**

This setting enables or disables sending commands to a computer attached via USB to the picoTracker running a "remote UI" application that can mirror the display from the picoTracker.

The [official picoTracker remote UI web application is available](#), it works only with Chromium based browsers.

### **Display Brightness**

The display brightness setting allows you to adjust the backlight level of the picoTracker's LCD display. The brightness can be set from 05 (minimum brightness) to ff (maximum brightness). The default value is 80 (medium brightness).

Adjusting the brightness can help with battery life (lower brightness uses less power) and visibility in different lighting conditions.

### **Theme Settings**

The Theme Settings screen allows you to customize the appearance of the picoTracker interface, including fonts and colors. To access the Theme Settings screen, select the "Theme settings" option on the Device screen.

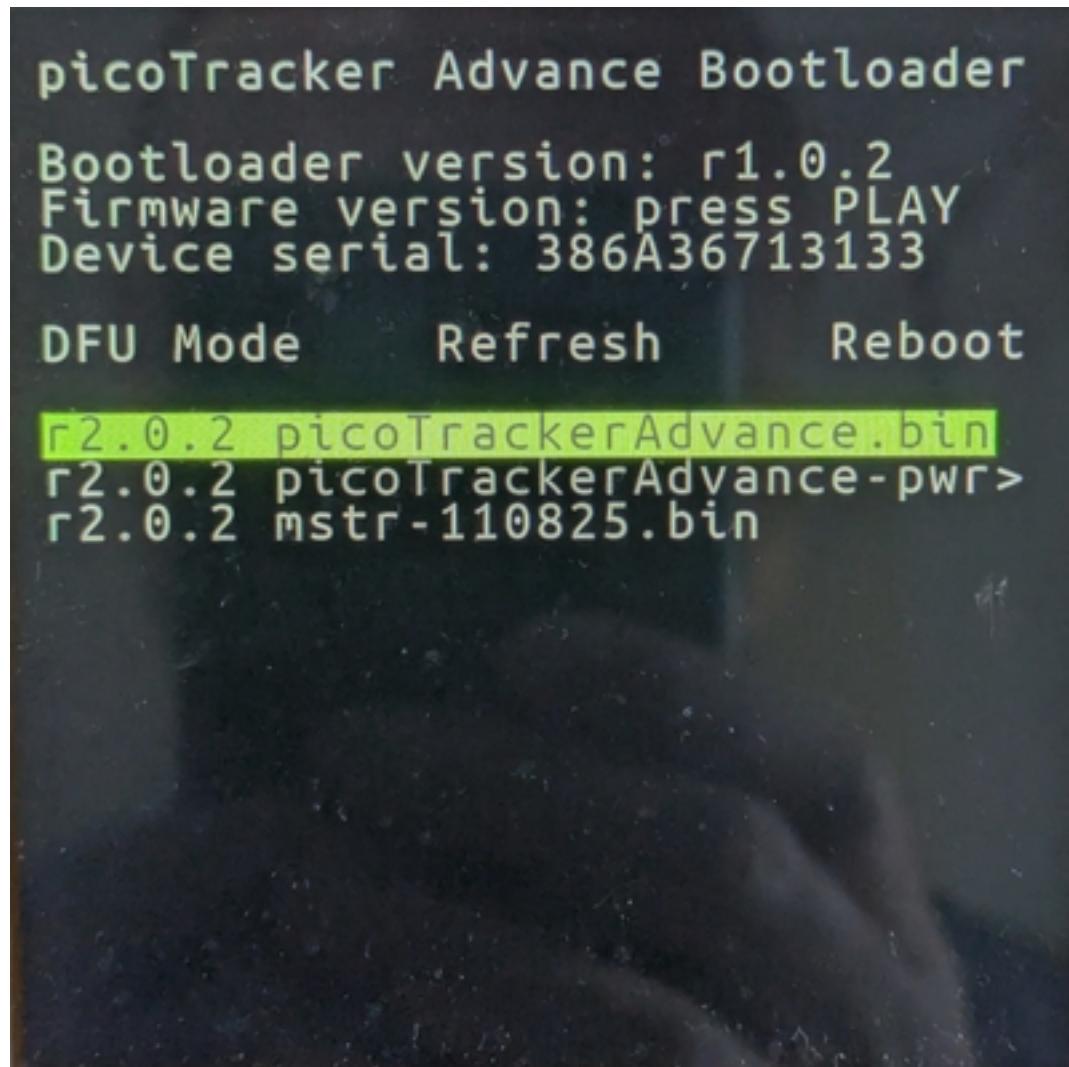
See the [Theme Settings](#) chapter for more details.

### **Updating Firmware**

First copy the new firmware file you want to install to the top level directory of your sdcard using your computer and then insert it back into your Advance.

Then go to the Device Screen and select the Update firmware menu item. This will reboot the Advance into its "bootloader" mode.

Once the Advance has booted into the bootloader mode, you will see the bootloader user interface on the screen:



You can now use the arrow buttons to select the .bin firmware file to install from the sdcard and press ENTER to install the firmware.

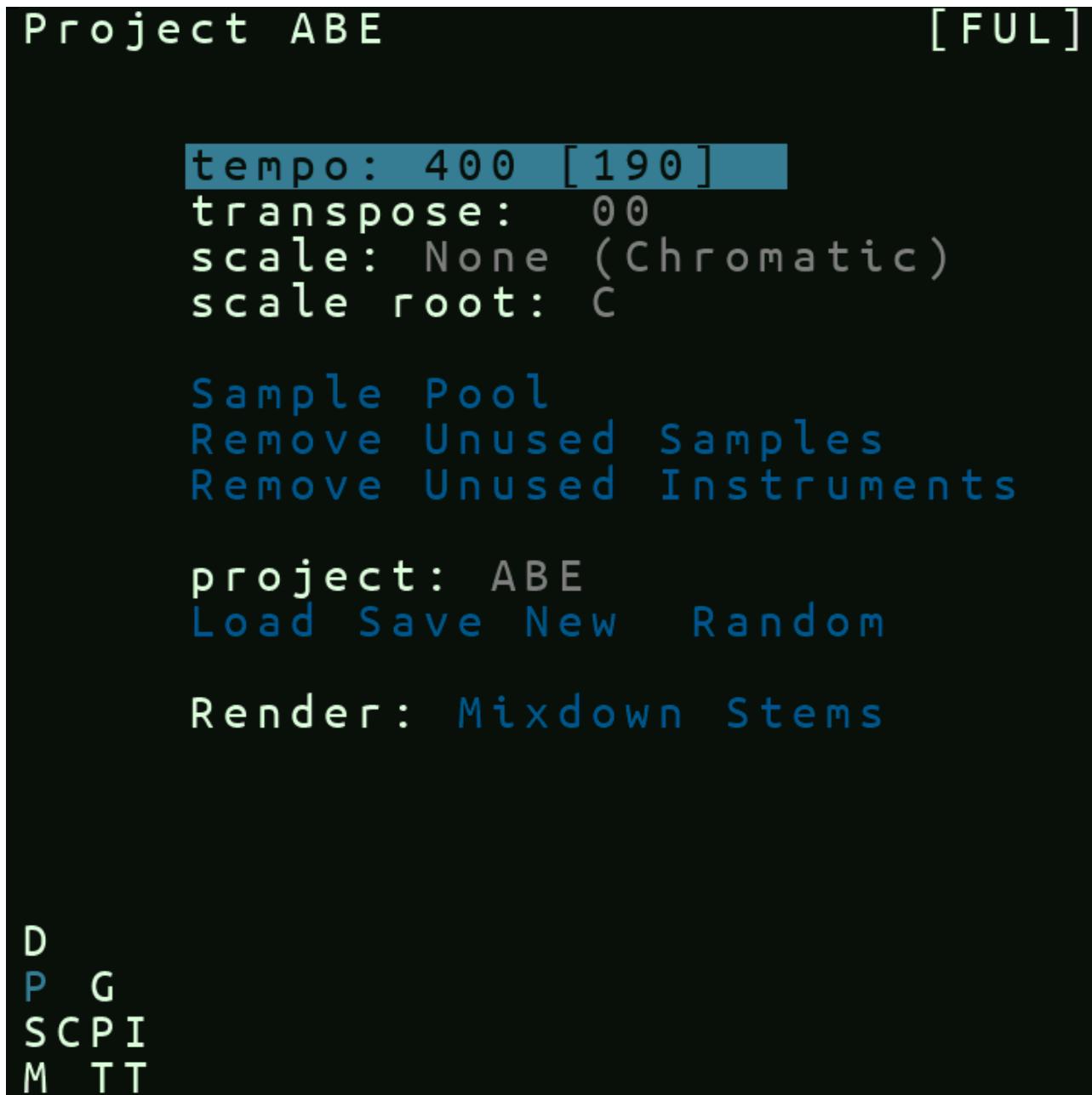
Once installation of the new firmware is completed, you can use the arrow keys to move the selection to the Reboot on screen option to reboot into normal mode and be running the newly installed firmware.

**NOTE:** If for some reason you cannot boot into the normal mode on your Advance, you can also boot the Advance into its "bootloader" mode by holding down the boot button, accessible through a small hole below the sdcard slot as you keep pressing the power button for approximately 8 seconds to reboot into the bootloader mode.



# Project Management

---



On the project screen you change various settings of the current project, save the current project, rename it (including giving it a random new name) create a new blank project or go to the project browser screen to **load** an another existing project.

Your current project settings are saved automatically every minute except when the sequencer is running, ie. when the current project is playing. This means that should you restart the picoTracker or accidentally power off or a crash occurs, your current project state within the last minute will be restored when you restart the picoTracker.

You can **explicitly** save the current project by pressing [SAVE] on the project screen. By doing this you can then later on revert to the state that you just saved by reloading the current project using the [Load] button on screen button on the project screen.

## Current Project settings

- **Tempo:** : Can be set between 60bpm [0x3c] and 400bpm [0x190]. Resolution aligned to LSDJ.
- **Transpose:** Live transposition of every triggered instruments.
- **Scale:** Set the scale that will applied to all notes entered in the project. When entering a note in the Phrase screen, you will only be able to enter notes that belong to the selected scale. See [the reference](#) for a list of all available scales.

## Current Project optimisation

- **Compact Instruments:** All unused instruments will have their sample set to (null) and the sample file will be removed from the projects sample subdirectory.

## Project Management

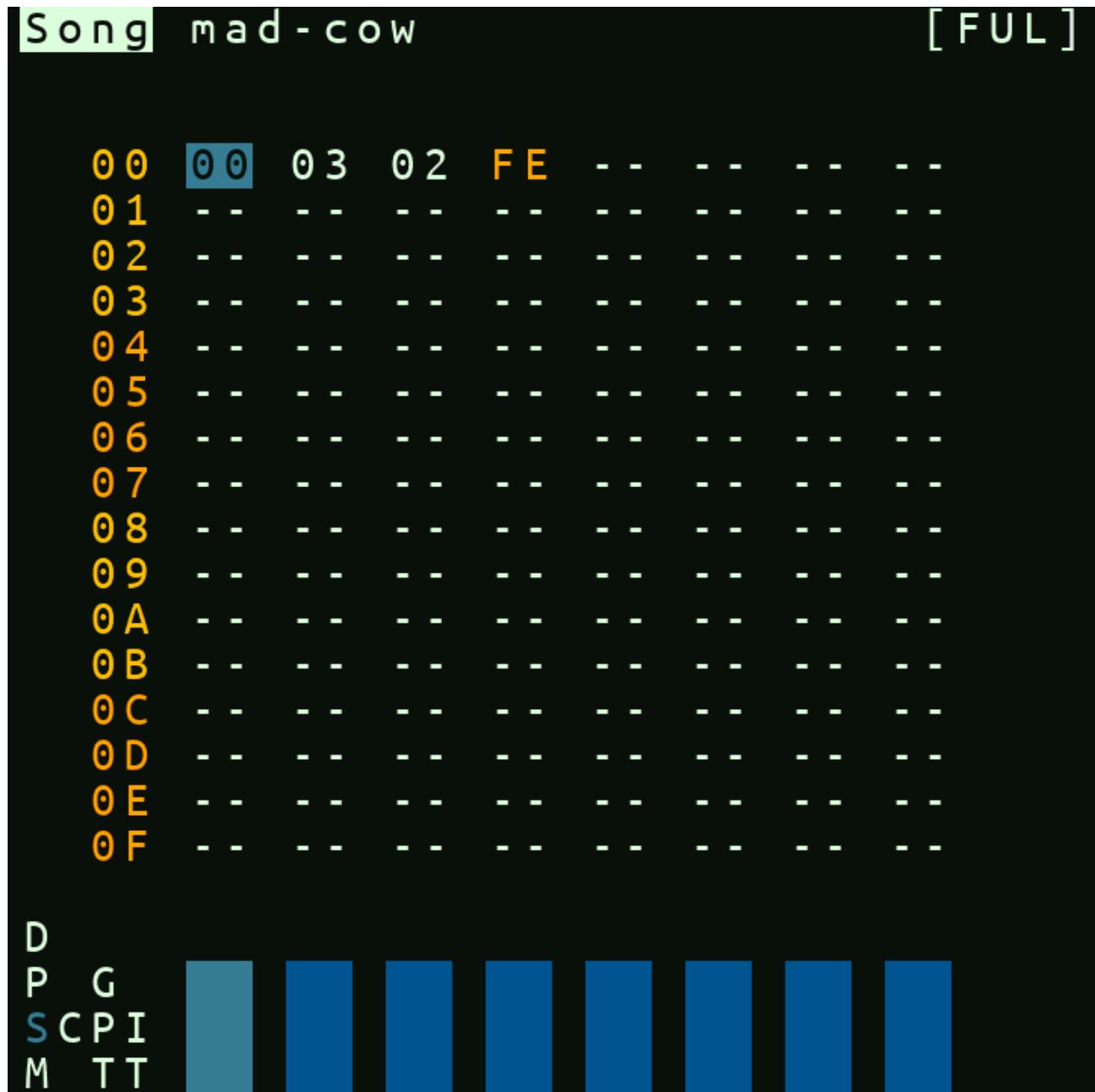
- **project:** Displays the current name of the project and allows you to edit it
- **Load** Go to the project file browser to load a different project or reload the last explicitly saved version of the current project
- **Save** Save the current project **NOTE:** *saving currently cannot be done during playback.*
- **New** *REPLACE* the current project with a new, *Blank* project.
- **Random** *RENAME* the current project with a new, *Randomly generated* name.
- **Import Sample** Access the sample import file browser with a single press of the ENTER key. When you're done in the Import Screen, you can press **NAV +LEFT** to return to the Project Screen.

The project name is **limited to 16 characters**.

You edit to project name by moving onto the name field and then holding the ENTER key while using the UP and DOWN keys to change the selected character and LEFT and RIGHT keys to move the cursor to the left or right of the current character. When on the last character, you can add character to the end of the project name by using the RIGHT key.  
To delete a character, select the character and press EDIT.

# Song Screen

---



Songs are made up of chains. Each hex number on the song screen grid is a chain. You can navigate to any of those chains by putting your cursor on a chain number and hitting NAV +RIGHT .

## Screen elements

- The top of the song screen displays if you are in SONG or LIVE mode (toggle with EDIT + LEFT / RIGHT ) and the name of the project that is currently open.

- At the bottom of the screen are the channel playback visualiser boxes. Each box correlates with the above column in the song grid. When a step is triggered in one of the columns, the note value of that trigger is displayed in the play-time visualizer.
- On the top right of the song screen we have play-time statistics. First there is the clipping indicator which displays "----" when your volume levels are ok and "clip" when you've cleared the headroom and are chopping off the top of your samples. The bottom row of the play-time statistics is the amount of time which has passed since you pressed Play.

## Controls

- You can toggle between "SONG" or "LIVE" mode with **EDIT + LEFT / RIGHT**
- You can navigate through the grid of chains using **UP / DOWN / LEFT / RIGHT**
- You can jump the cursor to the next/previous chain in a column by pressing **↑↑NAV + DOWN / UP↑**
- See the [Controls & Moves reference](#) for a list of all the controls you can use for editing the chains grid on the song screen.

## Solo and Mute

You can solo and mute specific channels on the song screen using the following key combos:

- **NAV +EDIT** : Toggles mute/unmute of cursor channel
  - if **NAV** is released before **EDIT** , channel stays muted
  - if **EDIT** is released before **NAV** , channel goes back to original state
- **NAV +ENTER** : Solo cursor channel
  - if **NAV** is released before **ENTER** , channel stays solo'ed
  - if **ENTER** is released before **NAV** , all channel go back to original state
- **ALT +NAV** : restore full playback on all channels
- **NAV +ENTER ,NAV +EDIT** can be used in conjunction with selections.
  - if a selection is present the toggle mute/solo action is done on all channels present in the selection

## Playback Modes

As mentioned above, the Song screen can be either in *song* or *live* mode. The controls in each mode differ slightly. You can switch between the modes using **ENTER +LEFT /RIGHT** while on the Song screen.

## Tempo Nudge

You can temporarily speed up or slow down the song playback using the Tempo Nudge feature. This is useful for syncing with external gear or making fine timing adjustments during live/DJ performances.

- **SHIFT + LEFT** : Nudge tempo down (slower)
- **SHIFT + RIGHT** : Nudge tempo up (faster)

The tempo will return to its original value when you release the keys.

## Song Mode

In the song mode **Play** starts and stops song playback from the currently highlighted row of the chain grid. If one of the rows channel is marked **--**, that channel will be ignored entirely for the rest of the playback session (until playback is stopped).

## Live Mode

In Live mode **Play** queues from the currently highlighted channel step.

- The Queued item will be played as soon as the playing chain on the selected channel reaches its last step.
- If there is no playing chain step on the selected channel, the next song chain that reaches its last step will trigger playback.
- Queued chains are shown with a blinking **→**.
- Pressing **Play** a second time will queue the chain using "immediate" mode.
- The queued item will be played as soon as the playing phrase on its channel reaches the last step.
- Immediate mode Queued items are shown with a fast blinking **→**.

**ALT +PLAY** will queue all channel steps on the current row.

- The queued items will be played as soon as the playing chain on their channel reaches its last step.
- Queued items are shown with a blinking **→**.
- Pressing **Play** a second time will queue the items using immediate mode.
- The queued items will be played as soon as the playing Phrase on the their channel reaches its last step.
- Immediate mode queued items are shown with a fast blinking **→**.

**NAV +Play** : Queues the selected channel step to be stopped.

- The queued channel will be stopped as soon as its playing chain reaches the last step.
- Queued Items are shown with a blinking **\_**.
- Pressing **Play** a second time will cause the selected channel step to be queued to stop using immediate mode.
- The queued channel will be stopped as soon as its playing phrase reaches the last step.
- Immediate mode queued items are shown with a fast blinking **\_**.



# Chain Screen

---

Chain 00			[ FUL ]
00	00	00	
01	01	00	
02	03	00	
03	--	00	
04	--	00	
05	--	00	
06	--	00	
07	--	00	
08	--	00	
09	--	00	
0A	--	00	
0B	--	00	
0C	--	00	
0D	--	00	
0E	--	00	
0F	--	00	

D  
P G  
S C P I  
M T T

- The three columns of the phrase screen, from left to right: (red) row counter, phrase list, and transpose.
- Chains are made up of phrases. Each hex number in the second column represents a phrase.
- You can navigate to any of those phrases by putting your cursor on one and hitting **ALT + RIGHT**.
- The rightmost column is transpose. If you use very high numbers like FF the phrase on that row will transpose down. Low numbers like 04 will cause the phrase on that row to transpose up.

- you can jump to previous / next chain on the row with with ↑↑EDIT+ LEFT/ RIGHT↑
- You can make a new phrase by hitting EDIT EDIT on a blank space in the Chain screen.
- You can clone a phrase by highlighting it with the cursor and pressing NAV +EDIT ENTER .
- You can copy a phrase/transposition/selection by highlighting and pressing EDIT .
- You can cut or delete a phrase/transposition/selection by highlighting and pressing EDIT + ENTER .
- Make a big selection by pressing NAV +EDIT , then Arrows around to highlight.
- Press NAV +ENTER to paste.
- In Song mode, Play starts/stops playback from the highlighted step, soloing the current chain
- In Live mode Play queues the currently highlighted phrase step

## Phrase Screen

---

Phrase 3a				[ FUL ]		
00	- - - I - -	- - -	0000	- - -	0000	0000
01	- - - I - -	- - -	0000	- - -	0000	0000
02	- - - I - -	- - -	0000	- - -	0000	0000
03	- - - I - -	- - -	0000	- - -	0000	0000
04	- - - I - -	- - -	0000	- - -	0000	0000
05	- - - I - -	- - -	0000	- - -	0000	0000
06	- - - I - -	- - -	0000	- - -	0000	0000
07	- - - I - -	- - -	0000	- - -	0000	0000
08	- - - I - -	- - -	0000	- - -	0000	0000
09	C# 4I00	- - -	0000	- - -	0000	0000
0A	- - - I - -	KIL	0000	- - -	0000	0000
0B	C# 4I00	- - -	0000	- - -	0000	0000
0C	F 5I01	- - -	0000	- - -	0000	0000
0D	- - - I - -	- - -	0000	- - -	0000	0000
0E	F 5I01	KIL	0004	- - -	0000	0000
0F	F 5I01	- - -	0000	- - -	0000	0000

D  
P G  
S C P I  
M T T



- The top of the phrase screen displays the name of the instrument under your cursor.

- The seven columns of the phrase screen, from left to right are:
  - row counter
  - note trigger
  - instrument number
  - FX1 command
  - parameters for FX 1
  - FX2 command
  - parameters for FX2
- You can use all the [standard picoTrackerkey editing combos](#) for editing on the phrase screen.
- If you copy/cut anything in the phrase screen, pasting will always put the data back in the same column (regardless if you've moved the cursor to another column). This means effects in column one are always pasted back there, and you can't accidentally paste a note into the effect column, etc.
- You can clone an instrument in the phrase screen by pressing **NAV +ENTER ,EDIT** on instrument number in phrase screen.
- If no instrument is set when triggering a new note, tables are not stopped, running commands are not stopped and phase of oscillator instrument is not reset (allowing for clickless transitions)
- In Song mode **Play** starts and stops playback from Step 00, soloing the current phrase
- In Live mode **Play** queues the Edited Chain Step from 00
- In Song mode **Play +NAV** starts/stops playback of the **whole song** from where the current phrase appears in the song

# Instruments

---

## Instrument Types

picoTracker supports several different types of instruments, each with its own unique capabilities and parameters. You can switch between instrument types using the **Type** field at the top of the instrument screen.

### Switching Instrument Types

1. Navigate to the instrument screen by pressing **NAV +RIGHT** from the phrase screen
2. By default the **NONE** instrument type is selected
3. Press **EDIT +LEFT** or **EDIT +RIGHT** to cycle through the available instrument types:
  - **SAMPLE** : For playing back WAV samples with various effects
  - **MIDI** : For controlling external MIDI devices
  - **SID** : Emulation of the Commodore 64 SID chip
  - **OPAL** : FM synthesis emulating classic PC soundcards
4. If you've made changes to the current instrument, you'll be asked to confirm before switching types
5. Note that you cannot change instrument types while playback is active

### Importing and Exporting Instruments

Once you've created an instrument, you can save it for use in other projects:

1. Make sure your instrument has a name set in the "name:" field
2. Select "Export" on the instrument screen
3. To import a previously saved instrument, select "Import"

### Sampler

- **sample**: Selects the .wav file to associate with this instrument. You can use the same sample in more than one instrument. Tap **ENTER**, **ENTER** to go to the Sample Import Screen which lets you load new .wav files into your project, with the last imported sample selected as the sample assigned to this instrument [1]
- **volume**: Set the volume of the instrument
- **pan**: Pans the instrument left or right (0x7F is center)
- **root note**: The root note of the sample
- **detune**: Detune the sample by the number of semitones
- **drive**: This is a volume modification before crush, and the instruments volume is after the crush
- **crush**: Decreases the bit resolution

- **downsample:** Decreases the bit rate (eg. low frequency aliasing whines). Each increase in this value will downsample the original sample by a factor of 2
- **cutoff:** Set the Filter cutoff frequency
- **reso:** Set the Filter resonance frequency
- **type:** The filter supports continuous change from low pass to high pass. Set type to 00 for low pass.  
FF for high pass and  
7F for band pass. All intermediate values morph in between them
- **dist:** Set the filter distortion. Available values are none and scream
- **interpolation:** Interpolation mode ('linear'/'none'). Selects which interpolation mode is used when in between samples.  
Linear interpolates linearly while none takes the nearest neighbor. Use none when playing samples at low range to add some typical overtones. Note using linear interpolation currently adds significantly to the CPU load during playback.
- **loop mode:** selects the looping mode.
  - none will play sample from start to finish
  - loop will start at the start and loop from loopstart to end.
  - looper sync will automatically tune a loop so that it plays exactly 16 bars. Use the root note to play twice faster/slower
  - oscillator is a special mode where the loop selection (from loopstart to end) is taken as oscillator data and automatically tuned. Note that 'root note' can be used to tune the oscillator back in a useful range
- **start:** start point of the sample regardless of if loop is enabled (note value is in hex)
- **loop Start:** start point of the sample when loop is enabled (note value is in hex)
- **loop End:** end point of the sample (note value is in hex). You can play samples backwards by setting the end value lower than the start
- **automation:** If On, the table play arrows will advance one row every time the instrument is triggered, and execute only the commands on the new rows. If this is Off, table behavior is normal (play arrows will move at the speed of 1row per tick)
- **table:** Select a table the instrument will always run. To clone a table here:  
NAV +[  
EDIT ,  
ENTER ]. Make a new table by selecting a higher number not yet in use.

## Sample Import Screen

You can enter the sample import file browser by hitting ENTER ENTER (press the ENTER key twice in quick succession) when you are in sample field on the *Sample* Instrument Screen. This double-press behavior provides a quick way to access the sample import file browser from the sample field in the *Sample* Instrument Screen.

When in the Project Screen, you can access the sample import file browser with a single press of the ENTER key on the "Import Sample" field.

Samples that you may want to import into a project can be located in any folder but it's recommended to keep them in the folder named /samples at the top-level of the sdcard as that is the default location for the sample import file browser to display when you enter it.

When you're done in the Import Screen, you can press NAV +LEFT to return to the screen you came from (either the Instrument Screen or the Project Screen).

Note: sub-directories will be sorted before files, but otherwise the files will be listed in an unspecified order (ie. not necessarily alphabetical order).

When entering the import file browser, the current folder is the library root folder /samples . All samples (.wav files) in that folder are listed.

Use the UP and DOWN arrow keys to navigate through the list of available sample files and subdirectories. Subdirectories are indicated with a / prefix. Press EDIT to enter a subdirectory, you can go back to the parent directory by navigating to the /.. entry and pressing ENTER . Hold down PLAY to audition the currently selected sample wave file. To import the currently selected wave file press ALT +PLAY .

**Single Cycle Waveforms** : Single cycle waveforms are specially marked with a l prefix in the file listing. These are WAV files with specific sizes (300 or 1344 bytes) that can be used as oscillators. When imported, they'll automatically be set to oscillator mode in the instrument settings.

A great collection of single cycle waveforms can be found in the [Adventure Kid Sample Library](#) .

The status bar at the bottom of the screen shows additional information about the selected file and other information such as:

- Current preview volume (vol:XX%)
- File size in bytes
- Available project sample storage size in bytes

## Basic Controls

- Press EDIT to enter a subdirectory
- Navigate to the /.. entry and press ENTER to go back to the parent directory
- Hold down PLAY to audition the currently selected sample wave file
- Press ALT +PLAY to import the currently selected wave file
- Use EDIT +UP /DOWN to adjust the preview volume

## Importing Multiple Samples

You can import multiple samples in a single session without leaving the Import View. This powerful workflow allows you to quickly build up your project's sample library:

1. Navigate to a sample file you want to import
2. Press ALT +PLAY to import it
3. Navigate to another sample file
4. Press ALT +PLAY again to import it
5. Repeat as needed for all samples you want to import

Each sample will be added to your project's sample pool. The last imported sample will automatically be assigned to the current instrument. Previously imported samples remain available in your project and can be assigned to other instruments later.

When you're finished importing samples, return to the instrument screen by pressing **NAV +LEFT**.

## Auditioning Volume Control

The Import View includes a convenient way to adjust the volume when previewing samples:

- **EDIT + UP** : Increase preview volume by 5%
- **EDIT + DOWN** : Decrease preview volume by 5%

The current preview volume is always displayed in the status bar at the bottom of the screen as "vol:XX%" alongside the file size information.

The preview volume uses a non-linear (quadratic) scale that provides more precise control at lower volumes, making it easier to fine-tune quiet previews. This setting is saved with your project and will be restored when you reload it.

*Note:* While there is no fixed limit for the number of sub-directory levels, there is a maximum of **256** files per directory. Also please note that while FAT formatted sdcards can support upto 256 characters per filename, picoTracker only supports upto **128** character file names and only with **ASCII** characters.

## Supported sample file formats

Only uncompressed Wave (\*.wav) files are supported using **8 or 16 bit**, mono or stereo and they **MUST** only be 44.1KHz.

## MIDI

### Midi Instrument Screen

A MIDI instrument has the following settings:

- **Channel** - This can be set **01-16** (**in decimal** not hex!) which is midi channel 1-16 respectively
- **Volume** - The volume any NOTE ON will be sent to your device: FF=127, 00=00
- **Program** - MIDI program change value to send (0x00-0x7F). Program changes for each MIDI instrument are sent only once at sequencer start. Setting this to **--** will disable sending program change messages entirely.
- **Length** - Sets note gate length in number of ticks
- **Automation** - When on, the table play arrows will advance one row every time the instrument is triggered, and execute only the commands on the new rows. If this is **Off**, table behavior is normal (play arrows will move at the speed of 1 row per tick)
- **Table** - As above, select a table the instrument will always run. Clone a table here: **NAV +EDIT ,ENTER**. Make a new table by selecting a higher number not yet in use.

## Synths

### SID

- **SID Chip:** Currently only "SID #1" is supported
- **OSC:** The oscillator to use: 0-2
- **VPW:** Pulse Width Modulation (PWM) of the oscillator, 000-FFF [1]
- **WF:** Waveform to use: TSQN (Triangle, Sawtooth, Square PWM, Noise (white-ish))
- **Sync:** Sync the oscillator (false/true)
- **Ring:** Enable or disable Ring modulation (false/true)
- **A/D/S/R:** Attack/Decay/Sustain/Release, each is single digit 0-15 value

#### Global SID chip settings:

- **Filter:** Set if the filter is on or off (false/true)
- **Flt cut:** The filter cutoff frequency 0-F
- **Flt res:** The filter resonance 0-F
- **Flt mode:** The filter mode: LP (lowpass), BP (bandpass), HP (highpass)
- **Volume:** Output volume: 0-F

The picoTracker currently only supports up to 3 SID instruments. Each of these represents a single monophonic oscillator with a single emulated "SID chip". Thus some of the settings above are shared between the 3 SID instruments because they are a global setting for the entire SID chip and *not* per oscillator.

#### SID Notes:

[1] In the SID, the pulse waveform can have its width dynamically adjusted. This parameter controls the duty cycle of the square wave, which determines the harmonic content and timbre of the sound.

A pulse width of 0 creates a very thin pulse (almost a spike), while a value of FFF (4095) creates a full square wave. Values in between create asymmetrical square waves with varying harmonic characteristics.

### OPAL

The OPAL instrument is an emulation of the FM synth "retro soundcards" of the PC DOS era. Each "OPAL" instrument is made up of 2 FM operators that can be configured for either standard 2 OP FM or in parallel as 2 additive oscillators. Each operator can have one of 8 different waveforms as shown below.

The OPAL instrument has settings for both the instrument as a whole and per each of the 2 operators.

The picoTracker currently only supports up to 3 OPAL instruments. Because each of the instruments is a single monophonic voice, using the same instrument simultaneously in the picoTracker 8 track sequencer will work as if you were controlling an external monophonic synth via MIDI output and cause the following notes to cut off the previously playing note on that OPAL instrument.

## Instrument wide settings

- **Algorithm:** Select either 2 Operator FM (1\*2) or Additive (1+2)
- **Deep tremolo/vibrato:** Enable/disable Deep tremolo or vibrato effect
- **Feedback:** Turn on/off feedback for Operator 1

## Per Operator settings

- **Level:** Operator output level **attenuation**, hence 0 is *highest* output level
- **Multiplier:** Frequency multiplier, 0=1/2, 1=1, 2=2, 3=3 etc
- **A/D/S/R:** Attack/Decay/Sustain/Release, each is single digit 0-15 value
- **Shape:** Waveform share, 8 to choose from, see waveform drawings below
- **TR/VB/SU/KSR:** Tremolo, Vibrato, Sustain and Envelope Scaling (KSR)
- **Keyscale:** Attenuates output level towards higher pitch: 1=3.0 dB/oct, 2=1.5 dB/oct, 3=6.0 dB/oct

## Exporting an Instrument

1. Make sure your instrument has a name set in the "name:" field
  - Each instrument must have a unique name before it can be exported
  - The default instrument type name (e.g., "Sampler", "MIDI", etc.) is not considered a valid name
2. Navigate to the instrument you want to export
3. Select "Export" from the instrument menu
4. Press **OK** to continue after the export is complete message is shown

Exported instruments are stored in /instruments/ on your SD card.

**NOTE:** In the future the sample for the instrument will be stored in the same directory as the instrument file but for now is not exported.

## Importing an Instrument

1. Navigate to the instrument you want to import
2. Select "Import" from the instrument menu
3. A file browser will appear showing all available .pti files in the /instruments directory
4. Select a .pti file to import, use the key combo **ALT +PLAY** to import it
5. The imported instrument will replace the currently selected instrument
6. Press **OK** to continue after the import is complete message is shown

## Tips for Instrument Management

- You can organise your instrument files into subfolders inside the `/instruments` directory but exported files will always be saved in the root `/instruments` directory
- Use descriptive names in the instrument's name field to easily identify them when importing later
- The instrument name is used for the export filename, so ensure it's set before exporting
- Back up your `/instruments` directory when backing up your picoTracker data on your sdcard

## **Limitations of instrument performance**

The picoTrackers CPU limits the number of simultaneous instruments that can be played at once. The specific limit depends on the instrument type and the settings of each instrument. In general the limit is:

- 4-5 Sample instruments or
- 3 OPAL instruments or
- 3 SID instruments
- 8 MIDI instruments

Because they are very light weight when it comes to CPU usage, 8 MIDI instruments can be sequenced at once, the limit then coming from the limit of 8 channels (aka tracks) available for sequencing on the picoTracker.

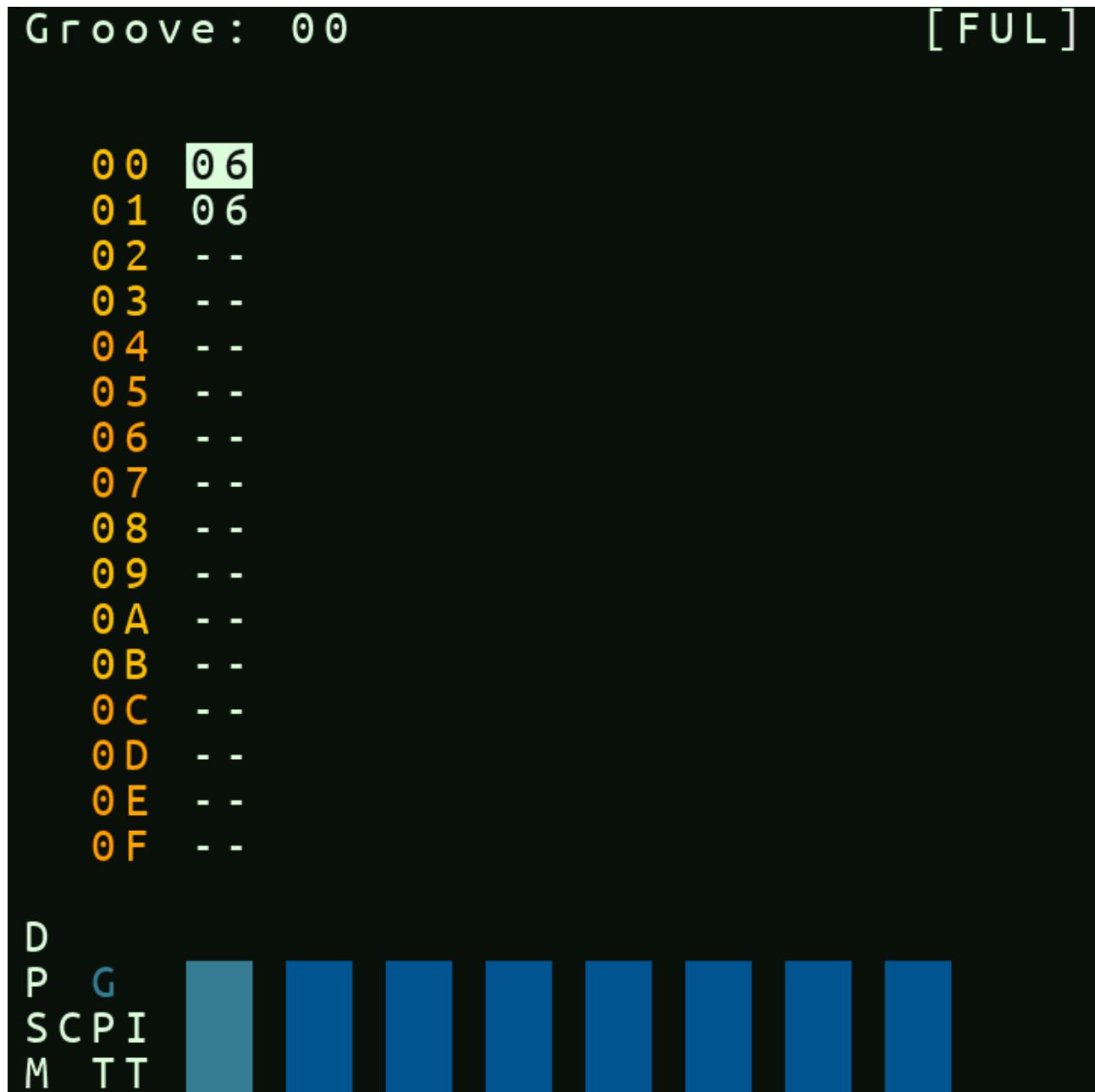
Even given the above limits, it is still possible to mix and match instruments of different types *roughly* within the above limits. For example 2 samplers, 1 OPAL, 1 SID and 4 MIDI should in theory be possible. Note this only applies to simultaneously sounding (playing) instruments and if care is taken to limit the number of simultaneously sounding instruments, a larger number of instruments can be defined within a project.

## **Exporting and Importing Instruments**

picoTracker allows you to save and reuse your instrument settings across different projects through the instrument export and import functionality. This feature is particularly useful for building a library of your favorite synth instruments or using instruments created by other picoTracker users.

# Grooves

---



## Introduction to Grooves

The Groove screen allows you to modify the timing of steps in your patterns, creating swing, shuffle, and other rhythmic variations. Grooves are an essential tool for adding a more human feel to your compositions and breaking away from rigid timing.

## Understanding Ticks

To understand how grooves work, it's important to first understand the concept of ticks:

- A tick is the smallest unit of time measurement in picoTracker
- By default, each row (step) in a phrase receives 6 ticks
- The groove screen allows you to redistribute these ticks across steps

## How Grooves Work

The groove screen displays a sequence of numbers, each representing the number of ticks assigned to consecutive steps in your phrases:

- The default groove is "6/6" - meaning each step gets 6 ticks (equal timing)
- If you change this to "9/3", odd-numbered steps will last 9 ticks while even-numbered steps will last 3 ticks
- This creates a swing or shuffle feel, as every other step is played longer

## Groove Examples

Groove Pattern	Effect
6/6	Default timing, equal distribution
9/3	Classic swing feel with longer first beat
4/8/4	Three-step pattern with emphasis on middle beat
1/F	Extreme swing with very short first beat (1) and very long second beat (15)

## Experimenting with Grooves

To understand how grooves affect playback:

1. Set a groove of "1/F" and watch the play cursor in the phrase screen
2. Notice how the cursor stays on even-numbered steps much longer than odd-numbered steps
3. Try a groove of "1/1/F" and observe how the cursor now emphasizes every third step
4. Experiment with more subtle values for different rhythmic feels

## Groove Screen Controls

The following controls are available in the groove screen:

- **EDIT** : Add a new step to the groove pattern if one doesn't exist
- **EDIT +Left/Right** : Modify the current step value
- **ENTER +Arrows** : Navigate between different groove patterns
- **EDIT +ENTER** : Clear the current step

## Tips for Using Grooves

- Subtle groove settings (like 7/5) can add a slight human feel without being obvious
- More extreme settings can create distinctive rhythmic patterns
- Grooves can be triggered from tables using the GRV command

- Different groove patterns can be used in different sections of your song

The GRV command (which is only active in the phrase screen) selects the current groove.

# Tables

---

Table 05				[ FUL ]			
00	PSL	08F2	- - -	0000	- - -	0000	0000
01	- - -	0000	VOL	0100	- - -	0000	0000
02	- - -	0000	- - -	0000	- - -	0000	0000
03	PSL	08F2	VOL	00FF	IRT	0000	0000
04	- - -	0000	VOL	0100	- - -	0000	0000
05	- - -	0000	- - -	0000	- - -	0000	0000
06	- - -	0000	- - -	0000	- - -	0000	0000
07	- - -	0000	- - -	0000	- - -	0000	0000
08	- - -	0000	- - -	0000	- - -	0000	0000
09	- - -	0000	- - -	0000	- - -	0000	0000
0A	- - -	0000	- - -	0000	- - -	0000	0000
0B	- - -	0000	- - -	0000	- - -	0000	0000
0C	- - -	0000	- - -	0000	- - -	0000	0000
0D	- - -	0000	- - -	0000	- - -	0000	0000
0E	- - -	0000	- - -	0000	- - -	0000	0000
0F	- - -	0000	- - -	0000	- - -	0000	0000

D  
P G  
SCPI  
M T T

## Introduction to Tables

Tables are one of the most powerful features in picoTracker, allowing you to create complex effect sequences, parameter automation, and even custom arpeggiators. Tables work as a sequence of commands that can be triggered from phrases or directly from instruments.

## Table Basics

- picoTracker provides 16 tables (0-15) that can be used throughout your project
- Each table can contain up to 16 rows of commands
- Tables can be looped, played once, or used to jump between different sections
- You can navigate between tables by pressing **EDIT +LEFT /RIGHT**
- You can use all the [standard picoTracker key editing combos](#) for navigating and editing the table screen.

## Table Interface

The table screen consists of seven columns:

Column	Description
Row #	The row number (00-15)
FX1	First effect command
Param1	Parameter for the first effect
FX2	Second effect command
Param2	Parameter for the second effect
FX3	Third effect command
Param3	Parameter for the third effect

## Using Tables from Phrases

Tables can be triggered directly from the phrase screen by using the **TBL** command in the effect column:

C-4 01 -- TBL 03

This will trigger table 3 when this step in the phrase is played. The table will be applied to the instrument in the current track.

When a table is triggered from a phrase:

1. The table starts playing from row 0
2. Each row in the table is processed at the same rate as the phrase's tick rate
3. The table effects are applied to the currently playing note

## Assigning Tables to Instruments

Tables can also be assigned directly to sampler instruments:

1. Navigate to the instrument edit screen
2. Find the table: field at the bottom of the screen
3. Set it to the table number you want to assign (0-15)
4. Use the automation field to control how the table is triggered

When a table is assigned to an instrument, it will automatically trigger whenever that instrument plays a note.

## Automation Settings

The automation field controls how the table state is handled when an instrument plays a note:

Setting	Behavior

OFF	Table state is reset each time the instrument plays
ON	Table state is saved with the instrument between notes

When automation is ON, the table's current position and state are saved with the instrument. This means that if you play a note with that instrument, then play another note later, the table will continue from where it left off rather than starting over from the beginning.

This is particularly useful for creating evolving sounds or complex effects that span across multiple notes. For example, you could create a table that slowly changes a filter cutoff or panning position across several notes, creating a continuous effect even when playing different notes with pauses between them.

When automation is OFF, the table will always start from the beginning each time the instrument plays a new note, which is useful for effects that should be consistent for each note.

## Table Commands

Here are some commonly used table commands:

### HOP - Jump to Another Row

#### HOP 0305

Jumps to row 03 of the current table and repeats this 05 times before continuing.

Hopping to the same row creates a hold effect. For example:

#### 05 HOP 0505

This will hold at row 05 for 5 ticks before continuing.

Hopping to self is particularly useful for creating complex envelopes. For example:

00 VOL 0400 : starts short volume decay to zero

01 HOP 0110 : holds enough ticks for VOL to complete

02 VOL 0560 : raise volume to 60

03 HOP 0303 : hold for a long time to allow VOL to complete and hold volume at 60

### STOP - Stop Table Execution

#### STOP

Stops the table from processing any further rows.

### VOL - Set Volume

#### VOL 40

Sets the volume to 64 (40 in hex).

### PAN - Set Panning

#### PAN 80

Sets the panning to center (80 in hex, range 00-FF).

## FRQ - Frequency/Pitch Adjustment

### FRQ 0108

Adjusts the pitch up by 8 semitones (08) in the first oscillator (01).

## Creative Uses for Tables

### Creating Arpeggiators

You can create custom arpeggiators by using the FRQ command to change the pitch in a sequence:

```
00 FRQ 0100 -----  
01 FRQ 0104 -----  
02 FRQ 0107 -----  
03 HOP 0003 -----  
04 HOP 0000 -----
```

This creates a simple major arpeggio pattern that repeats 3 times before looping back to the beginning.

### Trance Gate Effect

Create a rhythmic gating effect by rapidly changing volume:

```
00 VOL FF -----  
01 VOL 00 -----  
02 VOL FF -----  
03 VOL 00 -----  
04 VOL 80 -----  
05 VOL 00 -----  
06 VOL FF -----  
07 HOP 0000 -----
```

This creates a rhythmic on/off pattern that's great for dance music.

### Vibrato Effect

Create a vibrato effect by oscillating pitch slightly:

```
00 FRQ 0101 -----  
01 FRQ 0100 -----  
02 FRQ 00FF -----  
03 FRQ 0100 -----  
04 HOP 0000 -----
```

This creates a subtle pitch wobble that adds character to sustained notes.

### Parameter Automation

Tables excel at automating parameters over time:

```
00 VOL FF PAN 00 ----  
01 VOL F0 PAN 10 ----  
02 VOL E0 PAN 20 ----  
03 VOL D0 PAN 30 ----  
04 VOL C0 PAN 40 ----  
05 VOL B0 PAN 50 ----  
06 VOL A0 PAN 60 ----  
07 VOL 90 PAN 70 ----  
08 VOL 80 PAN 80 ----  
09 STOP -----
```

This creates a fade-in effect while panning from left to right, then stops.

## Filter Sweep

Create a filter sweep effect:

```
00 FCUT 00 -----  
01 FCUT 20 -----  
02 FCUT 40 -----  
03 FCUT 60 -----  
04 FCUT 80 -----  
05 FCUT A0 -----  
06 FCUT C0 -----  
07 FCUT E0 -----  
08 FCUT FF -----  
09 HOP 0000 -----
```

This gradually opens a filter from fully closed to fully open.

## Complex Drum Pattern

Use a table to create a complex drum pattern with retriggers:

```
00 RTG 04 -----  
01 VOL 80 -----  
02 VOL 40 -----  
03 VOL 20 -----  
04 HOP 0703 -----  
05 RTG 08 -----  
06 VOL FF -----  
07 HOP 0000 -----
```

This creates a drum pattern with initial quick retriggers, then a pause, followed by faster retriggers.

## Sample Position Effects with POF

Create interesting effects by manipulating the playback position of samples using POF (Play Offset):

```
00 VOL FF -----  
01 POF 1000 -----  
02 VOL C0 -----  
03 POF 2000 -----  
04 VOL 80 -----  
05 POF 3000 -----  
06 VOL 40 -----  
07 STP -----
```

This jumps to different positions in the sample while decreasing the volume, creating a stuttering effect that can sound like echoes or glitches depending on the sample content. The POF command takes a hexadecimal offset value that determines where in the sample to start playback from.

## Multi-Parameter Modulation

Combine multiple effects for rich sound design:

```
00 VOL FF FCT 20 PAN 40  
01 VOL E0 FCT 40 PAN 50  
02 VOL C0 FCT 60 PAN 60  
03 VOL A0 FCT 80 PAN 70  
04 VOL 80 FCT A0 PAN 80  
05 VOL A0 FCT 80 PAN 70  
06 VOL C0 FCT 60 PAN 60  
07 VOL E0 FCT 40 PAN 50  
08 HOP 0000 -----
```

This creates a complex pattern that simultaneously modulates volume, filter cutoff, and panning in a wave-like pattern.

## Tips and Tricks

- Tables are processed at the same rate as the phrase's tick rate
- Use the HOP command to create repeating patterns within a table
- Combine multiple tables using JMP for complex sequences
- For subtle effects, use small parameter changes between rows
- Tables can be shared between multiple instruments
- After Hopping count is reached, the table pointer moves directly to the line after the HOP rather than staying on the HOP line for one tick

# Renders and Stems

---



## Audio Rendering in picoTracker

picoTracker allows you to render your compositions to audio files, providing two different rendering options: **Mixdown** and **Stems**. These features enable you to export your music for sharing, further processing in a DAW, or archiving. The render options are available in the **Project View**.

## Accessing the Render Options

The render options can be found in the **Project View**. Navigate to the Project View and look for the "Render:" section, which contains two options:

- **Mixdown** - Creates a single stereo audio file of your entire composition
- **Stems** - Creates separate audio files for each channel/instrument in your composition

**NOTE:** Any existing files in the `/renders` directory for the current project will be **overwritten** by the rendering process.

### Rendering a Mixdown

A mixdown creates a single stereo audio file containing your entire composition. This is useful when you want to share your completed track or use it in other applications.

To create a mixdown:

1. Navigate to the Project View
2. Make sure your song is not currently playing
3. Select the "Mixdown" option under "Render:"
4. The rendering process will begin, and a progress dialog will display the current render time
5. The song will play through once from beginning to end while rendering
6. You can press "OK" at any time to stop the rendering process
7. When the song has completed playing through, the rendering will automatically finish

**NOTE:** The time displayed during the rendering process is the elapsed time of the audio being rendered, not the actual "clock" time of how long it takes to perform the render.

**NOTE:** You may hear noise or audio artifacts while the rendering is in progress or the audio being played at slower speed than expected, but this is normal due to the extra processing required during rendering and it will **not** affect the final rendered audio.

### Rendering Stems

Stems are separate audio files for each individual channel/instrument in your composition. This is particularly useful when you want to:

- Further process individual elements of your track in a DAW
- Create remixes or alternative versions of your composition
- Have more control over the final mix

To render stems:

1. Navigate to the Project View
2. Make sure your song is not currently playing
3. Select the "Stems" option under "Render:"
4. The rendering process will begin, and a progress dialog will display the current render time
5. The song will play through once from beginning to end while rendering
6. You can press "OK" at any time to stop the rendering process
7. When the song has completed playing through, the rendering will automatically finish

## Render Output

The rendered audio files are saved to your device's sdcard in the `/renders` top level directory. The exact location and format of the files depends on your picoTracker configuration:

- Mixdown files are named `(projectname)-mixdown.wav`
- Stem files are named `(projectname)-channel(number).wav`

## Render Quality

All audio is rendered at 44.1kHz sample rate with 16-bit depth.

## Tips for Rendering

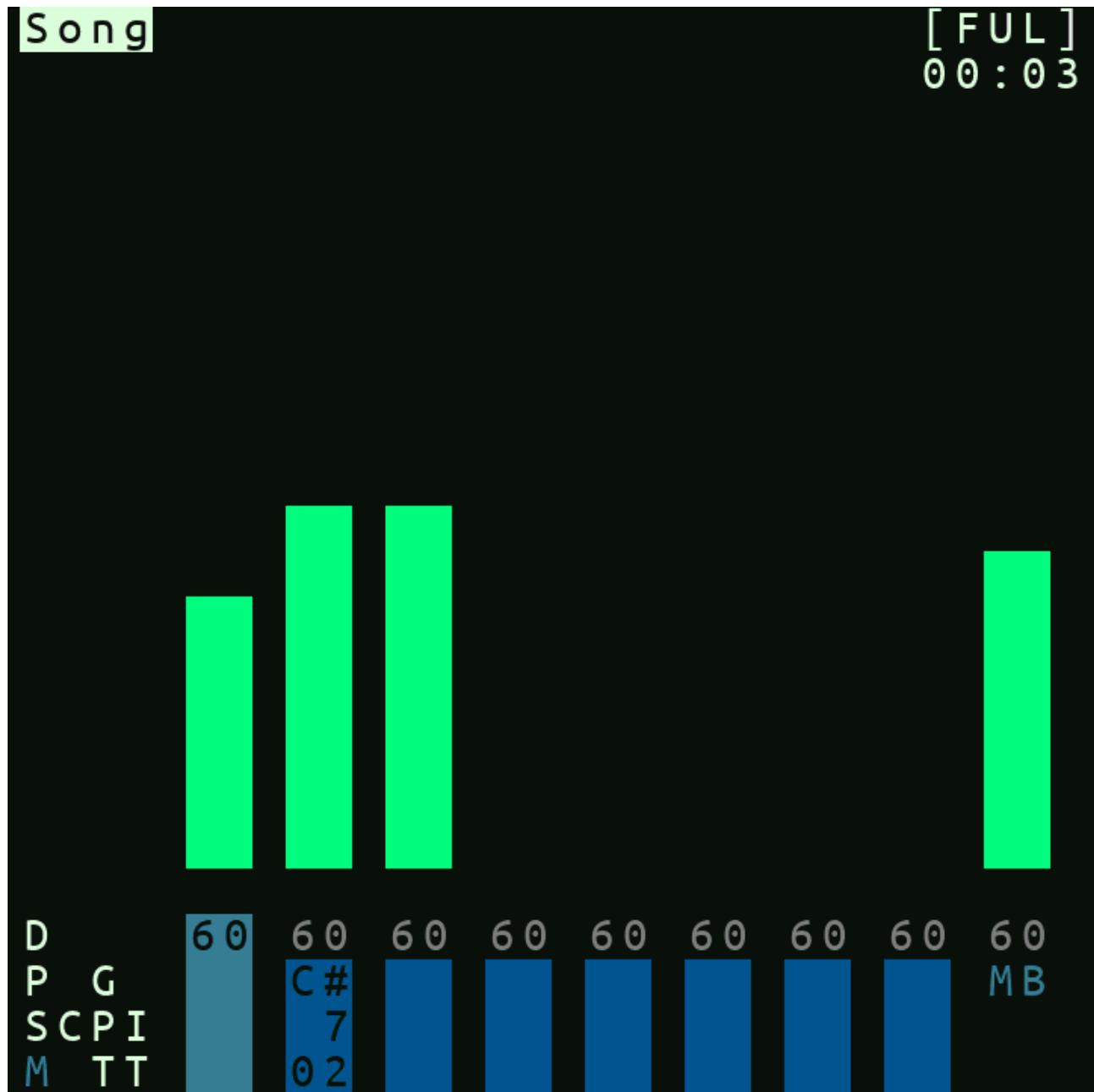
- Make sure your composition is complete and sounds as intended before rendering
- Check that all channel volumes and the master volume are set appropriately
- For the best quality output, avoid clipping by ensuring your levels aren't too high
- If you need to stop a render in progress, press the "OK" button on the render progress dialog
- Rendering will automatically stop when the song gets to the end and will *not* loop back to the beginning as it does in the song screen

## Accessing Rendered Files

After rendering, you can access your files in the `/renders` directory on your device's sdcard by taking out the sdcard and using a computer or other device to access it.

# The Mixer

---



The Mixer screen provides a visual overview and control center for the audio levels of each channel in your song, as well as the master output. It allows you to quickly adjust volumes, monitor audio levels and mute/solo individual channels.

## Mixer Channels

The Mixer screen displays a set of vertical stereo level (VU) meters, one for each of the eight available channels. Each channel strip represents a single stereo channel in your song and provides the following information:

- **Channel Level Meter:** 2 vertical bars representing the left and right channels that dynamically displays the current audio level of the channel. The higher the bar, the louder the channel's output.
- **Mute:** Each channel can be muted. The 'M' under each channel indicates when a channel is muted.

## Master Output

The rightmost section of the Mixer screen displays the **Master Output** level meter. This meter shows the combined audio level of all channels after they have been mixed together.

- **Master Level Meter:** Similar to the channel level meters, the master level meter displays the overall audio level of the final mix.
- **Clip Indicator:** The master level meter has a red clip indicator at the top. If this is lit, the master output is clipping and the audio will be distorted.

## VU Meter Details

The VU meters are designed to give you a clear visual representation of the audio levels. Here's a breakdown of the meter's features:

- **Dynamic Bars:** The bars move in real-time, reflecting the current audio level in dB.
- **Color-Coded Regions:**
  - **Green:** Indicates a safe and healthy audio level.
  - **Yellow:** Indicates that the audio level is approaching the maximum.
  - **Red:** Indicates that the audio level is clipping.
- **Stereo:** Each VU meter is stereo, with the left channel on the left and the right channel on the right.

## Controls

The mixer screen is primarily a monitoring tool, but it also provides same channel control key combos as are available on the song screen:

- **NAV +EDIT :** Toggles mute/unmute of cursor channel
  - if NAV is released before EDIT , channel stays muted
  - if EDIT is released before NAV , channel goes back to original state
- **NAV +ENTER :** Solo cursor channel
  - if NAV is released before ENTER , channel stays solo'ed
  - if ENTER is released before NAV , all channel go back to original state
- **ALT +NAV :** restore full playback on all channels

# Tips and Tricks

---

These tips and tricks come [courtesy of the LGPT community](#).

The command names have been changed to match picoTracker's naming convention and some sections have been edited for clarity.

---

## Delays and Echoes

### Simulating LSDj's D command

One command that is from LSDj but doesn't exist in picoTracker is the [D]elay command. However, it is possible to emulate it. It's a little tricky but gives a good view of several commands so I'll explain it here:

For this, we'll need to use POF (play offset). POF is the command that allows to position the sample playback cursor both/either to an absolute point in the sample or relative to the current play position. For example, POF 8000 will put the playback head to the middle of the sample, POF 4000 to the first quarter. POF 0040 will jump ahead from current position of a offset equivalent to a quarter of the sample size.

The basic trick is to use pof to re-start a running sample from position zero. Normally, the syntax should be POF 0000 but in order to keep things tidy, this value has no effect. So the trick is to use POF to jump to position 01 the relative backward of 01, which has the same effect. So POF 01FF (or POF FF01) will have the effect of positioning the sample to position zero.

Using that in a table in combination with VOL, you can:

- turn the VOL to zero
- wait for x ticks
- position the sample back to zero and VOL back to normal.

This will have the same effect as a Dx command in LSDj.

So, link with the instrument a table containing:

```
00 VOL 0000 ---- 0000 ---- 0000  
01 VOL 0080 POF 01FF ---- 0000  
02 HOP 0002 HOP 0002 ---- 0000
```

Will do the trick :)

Note: if you are using very short samples, it might be that before one tick, the sample is finished and the voice is automatically killed (to save CPU). In that case, POF'ing it will have no effect. So, in general, when using re-positioning with POF, it is better to put a short loop (for example of 1 sample long at the end) in the instrument definition.

## Doing Echoes

Using the 'retriggering' system we just saw, we can easily emulate echoes without having to enter notes at each step. Since doing POF 01FF will restart the sound from it's beginning we can do the following table :

```
00 POF 01FF ----- VOL 2000  
01 ----- HOP 0001  
02 -----  
03 HOP 0000 -----
```

The first column will generate repeating echoes (change the HOP line to adapt the echo's length) while the last one will provide a decay on the instrument volume. You can also add FCT command to get the filter to damp more the hi-frequencies, emulating the loss of regular delays.

Note that you can also emulate triplet using this technique.

## Oscillator mode

### Oscillator base

Oscillator is a special looping mode that allows to generate timbre from basically anything. It basically the loop start and loop end point of the instrument settings and adapts the sample scanning speed so that it becomes pitched to the played note.

The most straightforward use is when you have short waveform samples (a single square wave for example). Rather than having to compute the sample length or try to find the note that is played when looped, just turn the instrument's loop mode to 'oscillator' and it'll be automatically tuned. Of course, nothing prevents you to grab loops in ANY kind of sample, voices, drum loops whatever. The oscillator mode is a great way to get timbres from material you already got.

### Wavetables

Wavetables are a special process to allow to alter oscillator's content by switching or scanning the waveform of the oscillator while the note is playing. To get that effect, build a sample containing several waveforms (of the same loopsize preferably) in the same sample. For example, a few waves from the musicline pack. Each wave is 0x100 sample long so setup a loop from 0x00 to 0x100. Now to switch to the next wave, all you have to do is to offset the loop of 0x100. This can be easily achieved by using:

```
LOF 0100
```

Alternatively, you can also scan slowly from one wave to the other by doing a table

```
00 LOF 0001 -----  
01 HOP 0000 -----
```

This trick works also very neatly to change the timbre of some notes only by doing single LOF commands in the phrase view, changing slightly (depending on the value used) the timbre for each note.

## Pulse Width Modulation

Pulse width modulation (PWM) is a classic effect found in synths to adapt the duty cycle of a square wave. Using the oscillator mode and LOF, you can achieve some coarse PWM. Take a square wave of 0x100 length (use the oscillator mode), and imagine a window of half the wave size (0x080) travelling over the wave. If the 0x80-size loop is at the beginning, you get a square sample with duty cycle 0. If centered (0x40), you get a square wave and if all the way at the end, you get a duty cycle of 100%.

So taking a square waveform of 0x100 length, applying a loop of 0x80 in oscillator mode on it and adapting the loop position using LOF, we can achieve PWM quite easily :)

## Drones

Using the principle of wave scanning, you can take a long sample (a whole musical phrase for example), setup a short loop in oscillator mode on it and scan slowly across it. This will result into a timbre that is always tuned but evolves very slowly. Try using the same instrument on different channel playing a chord for very ghostly effect.

See for example the [PeteDrone](#) sample

## Breakz !

### Loop choppin' (old method)

Loop chopping is really easy in picoTracker through the POF command. POF sees the whole sample as 256 chunks of the same length. It allows to put the current playback head to the beginning of any of those chunks. To get it, we'll take a really simple drum loop sample going like this:

BD1 -- SD1 -- BD2 HH1 SD2 --

BD2 is in the middle of the loop to jump there, you issue LOF 8000 SD1 is at the first quarter of the loop, use LOF 40000 HH1 is at LOF A000 SD2 is at LOF C000

You can see by combining a lot of LOF commands one after the other, you can re-order the hits of the drum loop in pretty much any order and with quite a bit of resolution. Depending on the BPM and the style of the original wave, you'll end up with breakbeat/jungle/breakcore or just plain ol' mess.

Here's a nice example, courtesy of jonbro, chopping drums he had recorded previously:

[Jonbro - the thing is the thing](#)

### Slice (new chopping method)

Using this, you can assign up to 256 individual slices from C-2 (C minus two) up to the amount of slices you set. How handy!

Of course, you are not limited to drum loops. Just chop anything away !

### Grooves, BPM, Math and You

If you've ever messed around with grooves in either picoTracker or LSDJ, you've probably noticed that changing the total number of tics in a groove/pattern will change the perceived BPM of your track. Purists will say this is a new time signature, but in many cases, it will still sound like 4/4 time, but at a different tempo. So, if you're trying to sync by ear to a basic drum machine incapable of changing tics per beat or even loading a rendered WAV into a DAW, the following formula is for you:

$$(96 * \text{projectBPM}) / \text{total pattern tics} = \text{relative BPM}$$

The constant of proportionality in this case, 96, comes from picoTracker's default behaviour where a 16 step pattern with each step being 6 tics will give you the BPM set in the project screen, where one beat equals 4 steps. So, 16 steps \* 6 tics will give you 96 tics per pattern.

*Example*

You make a song with a groove of 4/4. This groove will play 8 times in one pattern since it's 2 steps long. So, total pattern tics = (4 tics + 4 tics) \* 8 = 64 tics.

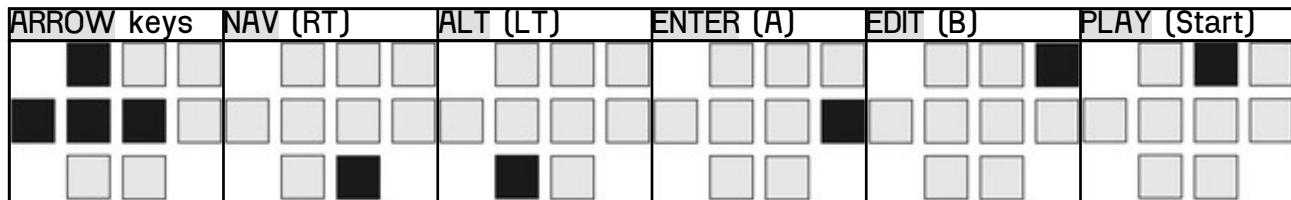
The project screen says the BPM is 100.

$$(96 * 100) / 64 = 150 \text{ is the perceived BPM}$$

# Controls & Moves

---

The keyboard layout resembles a typical old console game controller (controller button names in brackets):



The **NAV** ([]) and **ALT** keys are modifier keys for the **ENTER**, **EDIT** and **ARROW** keys. They are designed to modify the opposite keys to their location, so **ALT** will modify the **ARROW** keys and **NAV** will modify **ENTER** and **EDIT** (though there are exceptions).

## Remark about navigation between screens

Screens are getting more detailed from left to right.

Navigating to the next right screen is only possible if the current cursor position is at field with a filled value.

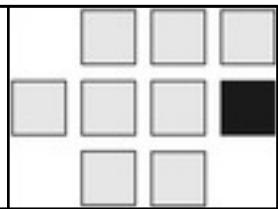
For example, in the song mode, to switch to the chain view, you need to have a filled value for a chain at the cursor position.

## Song View

### No Modifier

Function	Key Combination	Image
Move cursor on screen	ARROW keys	
Start/stop playback	PLAY	

enter last used value or start with 0 ENTER



## NAV Modifier

Function	Key Combination	Image
Switch to Project view	NAV + UP	
Switch to Mixer view	NAV + DOWN	
Switch to Chain view (if current position has a chain)	NAV + RIGHT	
Stop playback of selected chain, once it finished	NAV + PLAY	
solo selected track	NAV + ENTER	
mute selected track	NAV + EDIT	
reset all muted & soloed tracks	NAV + ALT	

## Solo & Mute

When NAV is released first, the (solo / mute) mode will be kept active (toggled). Pressing the same key combination but releasing NAV first will reset the (solo / mute) mode of the currently selected track.

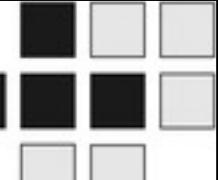
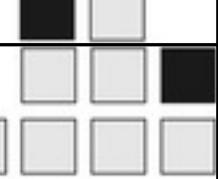
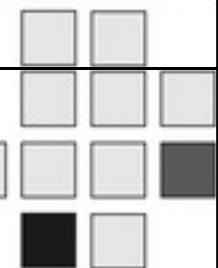
To reset all muted and soloed tracks, press NAV + ALT .

## ALT Modifier

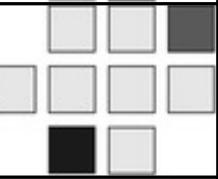
Function	Key Combination	Image
Jump up/down to next populated row after a blank row	ALT + (UP / DOWN)	
Nudge tempo down/up	ALT + (LEFT / RIGHT)	
query current row for playback	ALT + PLAY	
reset all muted & soloed tracks	ALT + NAV	
cut the current cursor position if filled, paste otherwise	ALT + ENTER	
Clone: Overwrite current highlighted Item with a copy of itself using the next unused item available.	ALT + EDIT + ENTER	
start selection	ALT + EDIT	
start selection with row selected	ALT + EDIT + EDIT	
start selection with current screen selected	ALT + EDIT + EDIT + EDIT	

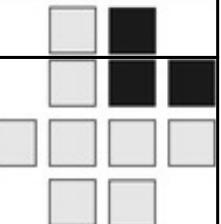
## Use Selection

Once a selection is started you can do a few more things:

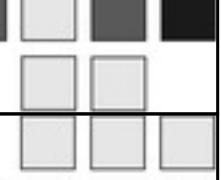
Description	Key Combination	Image
change selection	ARROWS	
increase selection to full row / screen	ALT + EDIT	
copy selection to clipboard	EDIT	
cut selection to clipboard	ALT + ENTER	

## EDIT Modifier

Function	Key Combination	Image
Cuts the current Highlighted Item.	EDIT + ENTER	
start selection	EDIT + ALT	
Page up/down in Song Screen.	EDIT + (UP / DOWN)	
Switch between Song and Live Mode	EDIT + (LEFT / RIGHT)	

mute selected track	EDIT + NAV	
[advance only] Sample recording, only accessible when sequencer is stopped	EDIT + PLAY	

## ENTER Key Combinations

Function	Key Combination	Image
insert if cursor position is empty	ENTER	
Insert next unused chain	ENTER, ENTER	
cut the current cursor position if filled, paste otherwise	ENTER + ALT	
Change chain at cursor position by 0x10	ENTER + (UP / DOWN)	
Change chain at cursor position by 0x01	ENTER + (RIGHT / LEFT)	
solo selected track	ENTER + NAV	

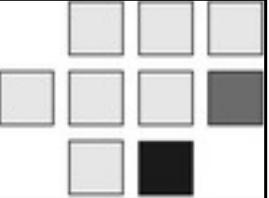
## Chain View

### No Modifier

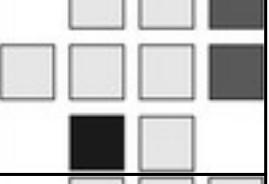
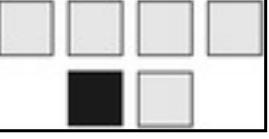
Function	Key Combination	Image
Move cursor on screen	ARROW keys	
Start/stop playback of current chain	PLAY	
enter last used phrase or start with 0	ENTER	

## NAV Modifier

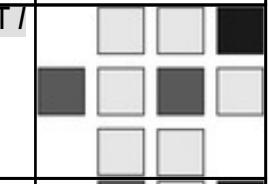
Function	Key Combination	Image
switch to song view	NAV + LEFT	
switch to phrase view	NAV + RIGHT	
play chain in song	NAV + PLAY	
unmute all	NAV + ALT	
toggle mute	NAV + EDIT	

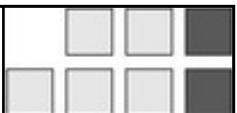
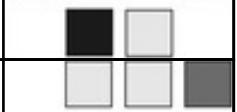
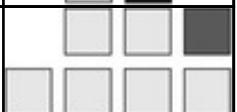
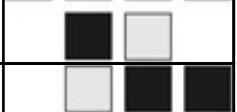
switch solo mode	NAV + ENTER	
------------------	-------------	---

## ALT Modifier

Function	Key Combination	Image
unmute all	ALT + NAV	
clone current position	ALT + EDIT + ENTER	
paste clipboard	ALT + ENTER	
start selection	ALT + EDIT	

## EDIT Modifier

Function	Key Combination	Image
warp to (previous / next) channel	EDIT + (LEFT / RIGHT)	
warp to (previous / next) chain of current channel	EDIT + (UP / DOWN)	
cut current position into clipboard	EDIT + ENTER	

clone current position	EDIT + ALT + ENTER	
toggle mute	EDIT + NAV	
start selection	ALT + EDIT	
start selection with row selected	ALT + EDIT + EDIT	
start selection with current screen selected	ALT + EDIT + EDIT + EDIT	
[advance only] Sample recording, only accessible when sequencer is stopped	EDIT + PLAY	

## Use Selection

Once a selection is started you can do a few more things:

Description	Key Combination	Image
change selection	ARROWS	
copy selection to clipboard	EDIT	

toggle mute	EDIT + NAV	
update selection values by +/- 0x10	ENTER + (UP + DOWN)	
update selection values by +/- 0x01	ENTER + (LEFT + RIGHT)	
cut the current selection	ALT + ENTER	
switch solo mode	NAV + ENTER	
switch to song view	NAV + LEFT	
switch to phrase view	NAV + RIGHT	
play chain	PLAY	
play chain in song	NAV + PLAY	

unmute all	ALT + NAV	
------------	-----------	--

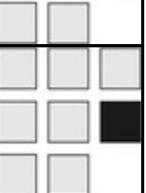
## ENTER Key Combinations

Function	Key Combination	Image
update cursor value by +/- 0x10	ENTER + (UP / DOWN)	
update cursor value by +/- 0x01	ENTER + (LEFT / RIGHT)	
paste clipboard	ENTER + ALT	
switch solo mode	ENTER + NAV	

## Phrase View

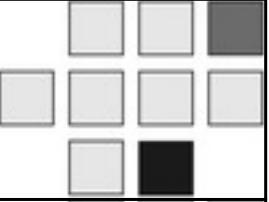
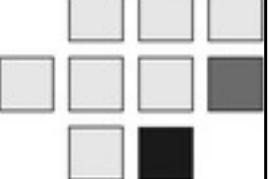
### No Modifier

Function	Key Combination	Image
Move cursor on screen	ARROW keys	
Start/stop playback of current phrase	PLAY	
enter last used value or start with 0	ENTER	

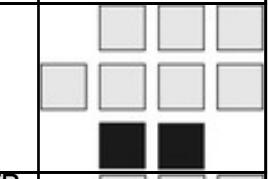
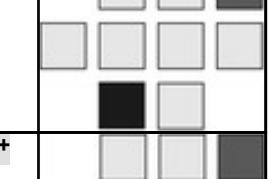
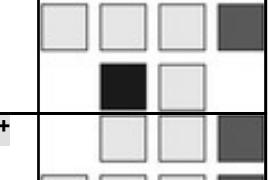
in instrument column: insert first unused instrument number	ENTER, ENTER	
in argument column of TBL command: insert first unused table number (TODO: see issue #753)	ENTER, ENTER	
in note / instrument column: audition note	ENTER held	

## NAV Modifier

Function	Key Combination	Image
switch to chain view	NAV + LEFT	
switch to instrument view	NAV + RIGHT	
switch to table view	NAV + DOWN	
switch to groove view	NAV + UP	
play phrase in song	NAV + PLAY	
unmute all	NAV + ALT	

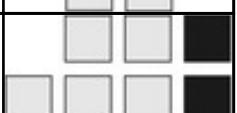
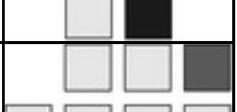
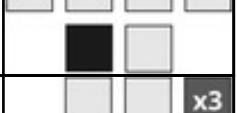
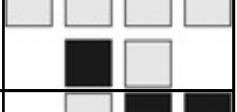
switch mute mode	NAV + EDIT	
switch solo mode	NAV + ENTER	

## ALT Modifier

Function	Key Combination	Image
unmute all	ALT + NAV	
paste clipboard	ALT + ENTER	
start selection	ALT + EDIT	
clone current instrument (cursor in note / instrument column)	ALT + EDIT + ENTER	
clone current table (cursor in note / instrument column) (TODO: see issue #753)	ALT + EDIT + ENTER	

## EDIT Modifier

Function	Key Combination	Image
warp to (previous / next) track	EDIT + (LEFT / RIGHT)	

warp to [previous / next] phrase in chain	EDIT + (UP / DOWN)	
cut current position into clipboard	EDIT + ENTER	
toggle mute	EDIT + NAV	
start selection	EDIT + ALT	
start selection with row selected	ALT + EDIT + EDIT	
start selection with current screen selected	ALT + EDIT + EDIT + EDIT	
[advance only] Sample recording, only accessible when sequencer is stopped	EDIT + PLAY	

## Use Selection

Once a selection is started you can do a few more things:

Description	Key Combination	Image
change selection	ARROWS	
increase selection to full row / screen	ALT + EDIT	

copy selection to clipboard	<b>EDIT</b>	
cut the current selection	<b>ALT + ENTER</b>	
update selection values by +/- 0x10	<b>ENTER + (UP + DOWN)</b>	
update selection values by +/- 0x01	<b>ENTER + (LEFT + RIGHT)</b>	

## ENTER Key Combinations

Function	Key Combination	Image
update cursor value by +/- 0x10	<b>ENTER + (UP / DOWN)</b>	
update cursor value by +/- 0x01	<b>ENTER + (LEFT / RIGHT)</b>	
paste clipboard	<b>ENTER + ALT</b>	
switch solo mode	<b>ENTER + NAV</b>	

## Instrument View

## No Modifier

Function	Key Combination	Image
Move cursor on screen	ARROW keys	
Start/stop playback	PLAY	

## ENTER Key

Function	Key Combination	Image
Import sample from samplelib (double-tap on sample field)	ENTER, ENTER	
Get next available table (on table field)	ENTER	

## NAV Modifier

Function	Key Combination	Image
Switch to Phrase view	NAV + LEFT	
Switch to Table view (if instrument has table)	NAV + DOWN	
Start playback (phrase mode)	NAV + PLAY	

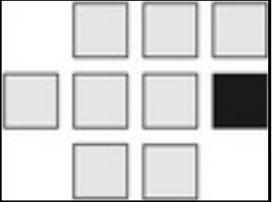
## EDIT Modifier

Function	Key Combination	Image
Previous/Next instrument (-1/+1)	EDIT + (LEFT / RIGHT)	
Previous/Next instrument (-16/+16)	EDIT + (DOWN / UP)	
Cut/purge instrument (sample) or clear table	EDIT + ENTER	
clone current table	EDIT + ALT + ENTER	
[advance only] Sample recording, only accessible when sequencer is stopped	EDIT + PLAY	

## Import View

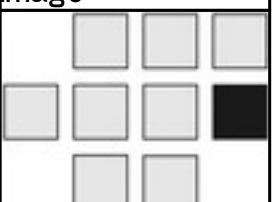
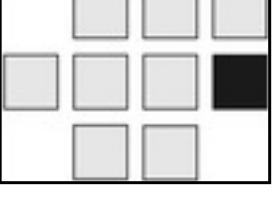
### No Modifier

Function	Key Combination	Image
Move up/down in file list	UP / DOWN	
Toggle selected button	LEFT / RIGHT	
Preview selected sample (hold)	PLAY	

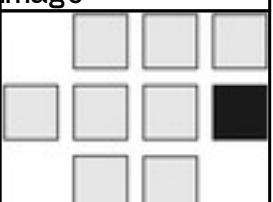
Navigate into directory	ENTER	
-------------------------	-------	---

## ENTER Key

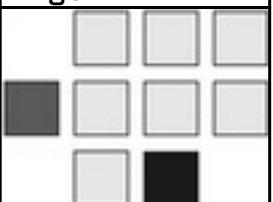
### In File Browser Mode:

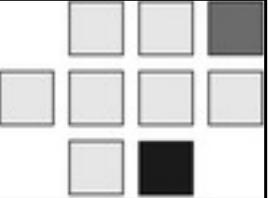
Function	Key Combination	Image
Import sample	ENTER (with Import button selected)	
Edit sample	ENTER (with Edit button selected)	

### In Project Pool Mode:

Function	Key Combination	Image
Edit sample	ENTER (with Edit button selected)	
[advance only] Remove sample from project	ENTER (with Remove button selected)	

### NAV Modifier

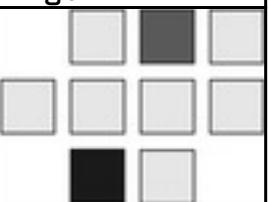
Function	Key Combination	Image
Return to source view (Phrase or Instrument)	NAV + LEFT	

Toggle between sample library and project pool	NAV + EDIT	
--	------------	---

## EDIT Modifier

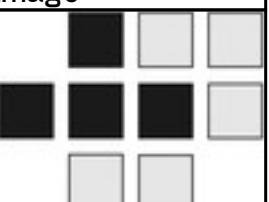
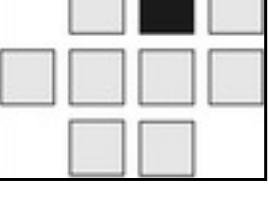
Function	Key Combination	Image
Increase / Decrease preview volume	EDIT + (UP / DOWN)	

## ALT Modifier

Function	Key Combination	Image
Import sample	ALT + PLAY	

## Sample Editor View

### No Modifier

Function	Key Combination	Image
Move cursor on screen	ARROW keys	
Preview sample (hold)	PLAY	

### ENTER Key

Function	Key Combination	Image
update cursor value	ENTER + (UP / DOWN) (with start, end or name field selected)	

update cursor position in field	ENTER + (LEFT / RIGHT)	
Save sample	ENTER (with Save button selected)	

## EDIT Key

Function	Key Combination	Image
delete character at cursor position	EDIT (with name field selected)	

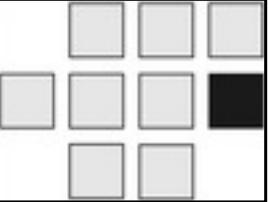
## NAV Modifier

Function	Key Combination	Image
Return to sample browser	NAV + LEFT	

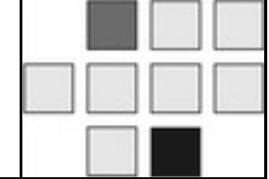
## Table View

### No Modifier

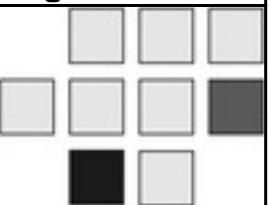
Function	Key Combination	Image
Move cursor on screen	ARROW keys	
Start/stop playback of current phrase	PLAY	

Paste last used command (in command column)	ENTER	
---	-------	---

## NAV Modifier

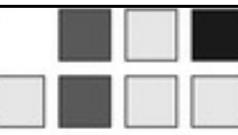
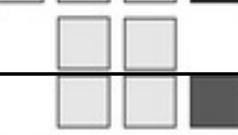
Function	Key Combination	Image
Switch to Phrase View (from Table1) or Instrument View (from Table2)	NAV + UP	
Switch to Table View (from Table2)	NAV + LEFT	
Switch to Table2 View (from Table1)	NAV + RIGHT	
Play phrase in song context	NAV + PLAY	

## ALT Modifier

Function	Key Combination	Image
Paste clipboard	ALT + ENTER	

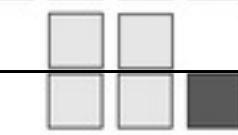
## EDIT Modifier

Function	Key Combination	Image
Warp to previous/next table (-1/+1)	EDIT + (LEFT / RIGHT)	

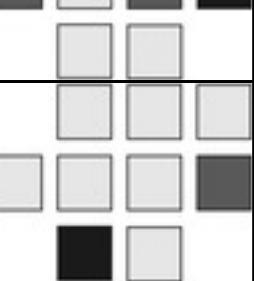
Warp to previous/next table (-16/+16)	EDIT + (DOWN / UP)	
Cut current position into clipboard	EDIT + ENTER	
Start selection mode	EDIT + ALT	
[advance only] Switch to Record View (when sequencer stopped)	EDIT + PLAY	

## Use Selection

Once a selection is started you can do a few more things:

Function	Key Combination	Image
change selection	ARROW keys	
increase selection to full row / screen	ALT + EDIT	
copy selection to clipboard	EDIT	
cut the current selection	ALT + ENTER	

## ENTER Modifier

Function	Key Combination	Image
update cursor value by +/- 0x10	ENTER + (UP / DOWN)	
update cursor value by +/- 0x01	ENTER + (LEFT / RIGHT)	
Paste clipboard	ENTER + ALT	

# FX Commands Reference

---

There can be upto two commands on every row of the phrase screen and upto three on a row in a table. Commands which effect instruments can be run on any step of the instruments playback, including the step where the instrument is triggered.

\*\* Note:\*\* most commands only effect sampler instruments unless otherwise specified in the documentation for the specific command.

## ARP abcd (ARPG in Igpt)

**cycle through relative pitches a, b, c, and d (starting with original pitch, then up a semitones, b semitones and so forth). The cycle loops if there's only zero's past a given post**

Examples:

ARP 3000: loops between original pitch and +3 semitones

ARP 4050: loops between original pitch, +4 semitones, +0 semitones, + 5 semitones

- speed of arpeggiator is constant and can not be changed

## CSH aabb (CRSH in Igpt)

**aa = pre crush drive (from 1 to 0xFF, 00 is no change) & bb = crush setting (from 0 to 0xF, 0x0 is 1 bit, 0xF is 16bit )**

## DLY --bb (DLAY in Igpt)

**Delays the note to be played by bb tics**

## FCT aabb (FCUT in Igpt)

**adjust the filter cutoff to bb at speed aa**

- FCT 0080 will instantly set the filter cutoff to 50%
- FCT 1000 will close the filter entirely at speed 10

## FLT aabb (FLTR in Igpt)

**lowpass filter, set absolute frequency value for cutoff aa & resonance bb**

- FLT 00FF is un-adulterated sound

## FRS aabb (FRES in Igpt)

**adjust the filter resonance to bb at speed aa**

- FRS 08FF will raise the resonance to screeching at speed 08

## GOF

**Close Gate OFF for Synth Instruments only**

## GRV aabb

### set Groove to bb

- In Phrases, if aa → 0 then Groove will be set for *all* tracks
- In Tables, the Groove command has a maximum value of 1F (15)

## HOP aabb

**play position will jump to the next phrase in a chain, jumping directly at position bb in the phrase.**

- hop is instant: instrument triggers and commands on the same row will be run.
- no effect on instruments
- in TABLES, cursor position will jump to row bb aa times, then pass thru the hop command and continue thru the rest of the table

## IRT aabb (IRTG in Igpt)

**Instrument Retrigger, will retrigger the current instrument. It gives a table the ability to work as programmable phrases that then can be triggered simply by changing tables.**

- IRT -bb will retrigger the current instrument transposed by bb semi-tones. Note that each IRT transposition is cumulatively added. So a table with IRT 0001 will keep going a semi tone up. Great for dubby echoes :)
- The retriggered instrument is NOT reset (as if you enter a note with no instrument number).
- The table (obviously) will continue to run and all running variable (filter,etc) won't be reset.
- This system is also pretty useful to implement temporary non 4/4 signature without having to switch grooves, since you have the ability to re-trigger the instrument at tick resolution
- don't forget trying to combine it with complex hop structure !

## KIL --bb (KILL in Igpt)

**instrument will stop playing after aa ticks.**

## LEG aabb (LEGA in Igpt)

**performs an exponential pitch slide from previous note value to pitch bb at speed aa.**

- 00 is the fastest speed for aa (instant, useless)

- bb values are relative: 00-7F are up, 80-FF are down, expressed in semi-tones
- if LEG is put on a row where a note is present and the pitch offset is 0 (e.g. C4 I3 LEG 1000 )
- the slide will occur automatically from previous note to the current one at the given speed.
- If an instrument is not triggered on the same row as LEG, the command will re-trigger the previous instrument (unless the previous instrument is still playing).
- LEG does exponential pitch change (i.e. it goes at same speed through all octaves) while PITCH is linear

When used with MIDI instruments, the LEG command also acts as an exponential MIDI pitch bend controller:

- As with internal instruments, aa sets the speed (with 00 being instant).
- bb sets the target pitch bend position, scaled to the MIDI 14-bit pitch bend range (0–16383):
  - 7F is the center (no bend).
  - 00 is full downward bend.
  - FF is full upward bend.
- MIDI pitch bend is persistent across notes – if you want to return to normal pitch, you must manually reset the bend to center.
  - This can be done by sending the aabb value **LEG 007F** on the next note or at any time.
- MIDI pitch bend can be sent without triggering a note, allowing for continuous pitch control.

### **LOF aaaa (LPOF in Igpt)**

#### **LooP OFset: Shift both the loop start & loop end values aaaa digits**

- LOF 0001 adds one to both values, LOF FFFF removes one (so values → 0x800 moves the loop backward)
- reset everytime you start a new note (same as volume, pitch)
- LOF is absolute
- you can't trigger a note with the LOF, it has to be executed after a sample is playing
- every time you trigger a sample LOF is set back to the instrument parameters

### **MCC aabb (MDCC in Igpt)**

Sends a **MIDI [continuous control] message**. aa is the control number and bb is the value. It will be sent on the MIDI channel of the currently running instrument.

### **MCH abcd**

**Sends a Chord via MIDI note on messages. The notes a,b,c,d relative semitone offsets from the current note as the root note of the chord.**

- For example, if the current note is C3, MCH 0047 will send a E3 note on and a G3 note on to give a C major triad chord.

- Some more examples for a C root note:
  - 0027 Suspended 2nd (**C D G**)
  - 0036 Diminished triad (**C D# F#**)
  - 0037 Minor (**C D# G**)
  - 0047 Major (**C E G**)
  - 0048 Augmented (**C E G#**)
  - 0057 Diminished 7th (**C F G**)
  - 037A Minor 7th (**C D# G A**)
  - 047B Major 7th (**C E G B**)
- Note as the maximum of 4 notes can be sent at once, this limits the maximum chord size to a 5 note chord with the maximum distance of 15 semitones from the root note (the note on the current step).

### **MPC --bb (MDPG in Igpt)**

sends a program change command on the current channel. 0000 is program change 1

### **PAN aabb**

**PAN aabb:** where bb is the pan destination and aa is the speed to get there

### **PFT aabb (PFIN in Igpt)**

**PitchFineTune:** where bb is the width and aa is the speed to get there

- Tunes the root note one semitone up (01-80) or down (FF-81)
- 00 in bb returns the note to the root center
- 00 is the fastest speed for aa

### **POF aabb (PLOF in Igpt)**

**PlayOffset** virtually cuts any sample in 256 chunks. jump absolutely to chunk aa or relatively move forward/back bb chunks.

### **PSL aabb (PTCH in Igpt)**

**PitchSLide** performs a linear pitch slide from previous note value to pitch bb at speed aa

- PSL is also time for the first two byte nibble
- PITCH is linear pitch change

The PSL command also acts as a linear MIDI pitch bend controller for MIDI instruments.

- As with internal instruments, aa sets the speed (with 00 being instant).
- bb sets the target pitch bend position, scaled to the MIDI 14-bit pitch bend range (0–16383):
  - 7F is the center (no bend).
  - 00 is full downward bend.
  - FF is full upward bend.
- MIDI pitch bend is persistent across notes – if you want to return to normal pitch, you must manually reset the bend to center.
  - This can be done by sending the aabb value PSL 007F on the next note or at any time.
- MIDI pitch bend can be sent without triggering a note, allowing for continuous pitch control.

### **RTG aabb (RTRG in Igpt)**

**retrigger the sound by looping the from current play position over a certain amount of ticks.**

- aa allows to move the loop forward of aa ticks each time the loop has been done (loop offset per retrigger)
- bb is the number of ticks used for the looping (speed of retrigger effect)

RTG 0001: loop one tick from current play position

RTG 0102: loop of two ticks but move the loop one tick every loop

RTG 0101: does not do anything because after looping one tick, you move forward one tick and therefore go back to the current position

### **TBL --bb (TABL in Igpt)**

**triggers table bb**

### **TPO --bb (TMPO in Igpt)**

**sets the tempo to hex value \bb.**

- TPO 0000 is safe and doesn't effect the tempo at all.
- TPO 003C (60bpm) is the lowest acceptable value and TPO 0190 (400bpm) is the highest acceptable value.  
Values outside the allowable range will be clamped to the nearest value within the range.

### **STP (STOP in Igpt)**

**Stops the table from processing any further rows.**

- This command is only valid in tables, not in phrases
- When the table processor encounters STP, it immediately stops executing the current table
- Useful for creating one-shot effects that should run once and then stop

### **VEL --bb**

Set the velocity of the note being played on the current step for a MIDI instrument. This valid for MIDI instruments *only* and this command is not supported for use in tables.

### **VOL aabb (VOLM in Igpt)**

**starting from the instrument's volume setting, approach volume bb at speed aa. 00 is the lowest volume and 00 is the fastest speed (instant).**

- to achieve sounds that grow in volume, make an instrument with volume 0 and then apply the VOL command

**NOTE:** For MIDI instruments the VOL command sets the velocity for that step. Only 1 VOL command in the first FX column is supported for MIDI. Only bb is used for velocity, aa has no effect for MIDI instruments.

# Available Scales

---

The following scales are available and can be selected from the Scale menu in the Project screen:

- None (Chromatic)
- Acoustic
- Adonal malakh
- Aeolian mode (minor)
- Algerian
- Altered
- Augmented
- Bebop dominant
- Blues
- Dorian
- Double harmonic
- Enigmatic
- Flamenco
- Gypsy
- Half diminished
- Harmonic major
- Harmonic minor
- Hirajoshi
- Hungarian gypsy
- Hungarian minor
- Insen
- Ionian mode (major)
- Istrian
- Iwato
- Locrian
- Lydian augmented

- Lydian
- Major bebop
- Major locrian
- Major pentatonic
- Melodic minor
- Melodic minor (asc)
- Minor pentatonic
- Mixolydian
- Neapolitan major
- Neapolitan minor
- Octatonic
- Persian
- Phrygian dominant
- Phrygian
- Prometheus
- Tritone
- Ukrainian
- Whole tone

# MIDI Implementation

---

## MIDI Implementation

---

### MIDI Input

picoTracker supports MIDI input for real-time playback and control. This allows you to connect a MIDI keyboard or controller to play notes and control various parameters.

#### Input Methods

picoTracker supports two MIDI input methods:

1. **USB MIDI** : Connect any USB MIDI device directly to the picoTracker's USB port. USB MIDI is supported on all picoTracker hardware versions.
2. **TRS MIDI (3.5mm)** : Connect standard MIDI devices using a MIDI to TRS adapter.

**Note:** TRS MIDI input requires a v2.1 or newer picoTracker PCB  
\*.

Both USB and TRS MIDI inputs are always enabled by default when the picoTracker starts up.

### MIDI Note Playback

When a MIDI device is connected and configured, incoming MIDI notes will trigger the corresponding instruments in picoTracker. Each MIDI channel is mapped to instrument slot in picoTracker. For now this mapping is fixed, eg. MIDI channel 1 will trigger instrument 00, MIDI channel 2 will trigger instrument 01, and so on.

### Polyphonic Playback Limitations

**Important:** Not all instrument types support polyphonic MIDI input playback:

- **Sample Instruments** : Fully support polyphonic playback. Multiple notes can be played simultaneously, and each note can be stopped independently.
- **OPAL Instruments** : Should be considered monophonic from a MIDI input perspective. However they DONT behave completely like a monophonic instrument, so care needs to be taken to ensure that only one note is played on a midi channel at a time.
- **SID Instruments** : Should be considered monophonic from a MIDI input perspective. However they DONT behave completely like a monophonic instrument, so care needs to be taken to ensure that only one note is played on a midi channel at a time.

### Supported MIDI Messages

picoTracker currently supports the following MIDI message types:

- **Note On** : Triggers instrument playback. Each MIDI channel maps to a corresponding instrument.
- **Note Off** : Stops the corresponding note that was previously triggered by a Note On message.
- **Start** : Starts playback when receiving a MIDI Start message.
- **Stop** : Stops playback when receiving a MIDI Stop message.

The following message types are recognized but **not** implemented yet:

- **Clock** : Synchronizes picoTracker's tempo to an external MIDI clock source.
- **Aftertouch** (Polyphonic Pressure)
- **Control Change** (CC)
- **Program Change**
- **Channel Aftertouch** (Channel Pressure)
- **Pitch Bend**
- **Continue**

## MIDI Configuration

Currently, MIDI input is always enabled by default when picoTracker starts up. There is no specific configuration required to use MIDI input.

Each MIDI channel directly maps to an instrument index in picoTracker. For example, MIDI channel 1 will trigger instrument 1, MIDI channel 2 will trigger instrument 2, and so on.

## MIDI Output

*[MIDI output documentation to be added]*

(v1.2 and v2.0 PCBs require a soldered hardware modification to enable TRS MIDI input)

# picoTracker Remote User Interface Protocol

---

## Overview

The Remote UI protocol is a communication mechanism that allows rendering the picoTracker's user interface over a USB serial connection. It enables a remote client to:

- Receive UI rendering commands
- (NOT YET IMPLEMENTED) Send button input events back to the device

## Command Structure

### Command Marker

Every command starts with a fixed marker: `0xFE` (REMOTE\_UI\_CMD\_MARKER). This allows clients to verify the start of a valid command.

**Note:** That the use of this value as a marker to start commands means that the characters `0xFE` and `0xFF` from the extended ASCII range are not allowed in the protocols command parameter values.

### Command Types

Note: `ASCII_SPACE_OFFSET = 0xF`

1. `TEXT_CMD (0x2)`: Draw a character

Parameters:

- Character to draw
- X position (offset by `ASCII_SPACE_OFFSET`)
- Y position (offset by `ASCII_SPACE_OFFSET`)
- Invert flag (0 for normal, `0x7F` for inverted)

1. `CLEAR_CMD (0x3)`: Clear screen

Parameters:

- Background color in RGB888 format

1. `SETCOLOR_CMD (0x4)`: Set foreground color

Parameters:

- Color in RGB888 format

## 1. SETFONT\_CMD (0x5)

Parameters:

- Font index (offset by ASCII\_SPACE\_OFFSET)

Currently the only available fonts are:

Index	Font
0	Hourglass
1	You Squared

## Example Transmission Flow

To give a concrete example, below is a simple example of how a command in the picoTracker firmware is transmitted over USB serial:

```
// Drawing a character 'A' at position (2,3), not inverted
remoteUIBuffer[0] = 0xFE;      // Command marker
remoteUIBuffer[1] = DRAW_CMD;  // Draw command
remoteUIBuffer[2] = 'A';       // Character
remoteUIBuffer[3] = 34;        // X position (2 + 32)
remoteUIBuffer[4] = 35;        // Y position (3 + 32)
remoteUIBuffer[5] = 32;        // Not inverted
sendToUSBCDC(remoteUIBuffer, 6);
```

## Input Commands

Clients are also able to send input events to the picoTracker. As for output commands, the input commands are prefixed by sending the REMOTE\_UI\_CMD\_MARKER byte.

The events currently supported are:

### 1. FULL\_REFRESH\_CMD (0x2): Request sending all the current screen and current font

Parameters:

- None

## Limitations

- Input events other than FULL\_REFRESH\_CMD are not yet implemented.

## Client Implementation Guidelines

1. Look for REMOTE\_UI\_CMD\_MARKER (0xFD)
2. Verify command type
3. Subtract 32 from positional/color values
4. Handle potential transmission errors
5. Implement appropriate rendering based on received commands



# Index

---