

# HoustonTracker 2.30 Manual

March 3, 2018

# Contents

<b>1</b>	<b>About</b>	<b>2</b>
1.1	What is HoustonTracker 2? . . . . .	2
1.2	Features . . . . .	2
1.3	Supported Calculator Models . . . . .	3
1.4	License . . . . .	4
<b>2</b>	<b>Conventions, Terms, and Definitions</b>	<b>5</b>
<b>3</b>	<b>Setup</b>	<b>7</b>
3.1	Requirements . . . . .	7
3.2	Installation . . . . .	8
3.3	Running HT2 . . . . .	11
<b>4</b>	<b>Using HT2</b>	<b>13</b>
4.1	Quickstart Tutorial . . . . .	13
4.2	Keys . . . . .	17
4.3	Composing Music . . . . .	21
4.3.1	Modes, Indicators, and Global Variables . . . . .	21
4.3.2	Editing the Song Sequence . . . . .	24
4.3.3	Editing Note Patterns . . . . .	26
4.3.4	Editing Drum/Effects Patterns . . . . .	27
4.3.4.1	Drums . . . . .	28
4.3.4.2	Effects Reference . . . . .	31
4.4	Loading, Saving, and Backing Up . . . . .	35
4.5	Error Codes . . . . .	37
<b>5</b>	<b>The ht2util Savestate Tool</b>	<b>38</b>

<b>6</b>	<b>Thanks, Credits, and Greetings</b>	<b>40</b>
	<b>Appendices</b>	<b>41</b>
<b>A</b>	<b>Building from Source</b>	<b>42</b>
<b>B</b>	<b>Data Format Specification</b>	<b>44</b>
<b>C</b>	<b>Synthesis Techniques</b>	<b>46</b>
<b>D</b>	<b>Troubleshooting</b>	<b>49</b>
D.1	I'm having problems installing/configuring TiLP on Windows.	49
D.2	I can't figure out how to install TiLP on OS X. . . . .	49
D.3	How do I configure my GrayLink cable under Windows? . . .	50
D.4	TI-Connect won't let me transfer files to my TI82. . . . .	51
D.5	I cleared the RAM on my TI-8x Plus, but there is still not enough space to install HT2. . . . .	52
D.6	I tried everything, but it just won't work. Help! . . . . .	52

# 1. About

## 1.1 What is HoustonTracker 2?

HoustonTracker 2 is a software sequencer that enables you to create music on Texas Instruments graphing calculators. It uses the machines' communication port to output multi-channel 1-bit music. Its interface is inspired by popular trackers such as LSDJ, Famitracker, and Milkytracker.

HT2 supports several models of the Z80-based line of TI calculators. It is mainly targetted at older, obsolete models like the TI-82, but also works on newer machines up to and including the TI-84 Plus SE. For a complete list of supported models, see section 1.3.

## 1.2 Features

- 3 tone channels
- 1 non-interrupting drum channel
- up to 128 note patterns
- up to 64 drum/fx patterns
- sequence length up to 255 pattern rows
- 16-bit frequency precision

- 10-bit speed precision, can be configured per step
- various effects, including:
  - L/C/R stereo hard-panning for tone and drum channels
  - advanced duty cycle modulation
  - noise and glitch effects
  - pitch slides
- 2 user definable samples
- up to 8 savestates
- edit during playback

### 1.3 Supported Calculator Models

The following table gives an overview over which TI calculator models are supported by HoustonTracker 2. If your calculator isn't listed in the table, chances are HT2 doesn't support it.

model	ROM version	shell	HT2 build	status
TI-73, 73 Explorer	all	mallard	–	planned
TI-76.fr	all	Ion	ht2.83p	untested
TI-81	–	–	–	not supported
TI-82	<16.0	–	–	planned
TI-82	16.0-19.0	CrASH 1.6	ht2.82p	supported
TI-82 Parcus	19.006	CrASH.19006	ht2p.82p	supported
TI-82 Advanced	–	–	–	not supported
TI-82 Stats, Stats.fr	all	Ion	ht2.83p	supported
TI-82 Plus	all	DoorsCS	ht2(s).8xp	supported
TI-83 (all versions)	all	Ion	ht2.83p	supported
TI-83 Plus (all versions)	all	DoorsCS	ht2(s).8xp	supported
TI-84 Plus/Plus SE	all	DoorsCS	ht2(s).8xp	supported
TI-84 Plus CSE/CE	–	–	–	not supported
TI-85, 86	–	–	–	not supported
TI-89, 92, 92+, V200	–	–	–	not supported

Models marked as "not supported" have differences in architecture that

would require a major rewrite of HT2. There are no plans to support these models in the near future, especially not the 84+ Color models.

## 1.4 License

HoustonTracker 2 is free, open source software. It is released under a "Revised" BSD-License. This means you're basically free to use, modify, and redistribute this software both in binary as well as source form, as long as you don't pretend that I endorse what you're doing, or try to hold me responsible for any damage done.

The full license terms are as follows:

Copyright (c) 2015-2016, utz/irrlight project  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of IRRLICHT PROJECT nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 2. Conventions, Terms, and Definitions

The following section outlines some basic terms and conventions used in HoustonTracker 2, and in this documentation.

**Keypress conventions** In this document, keypresses are denoted in bold font, surrounded by square brackets. All keypresses in HT2 are sequential, meaning you never need to press more than one key at the same time.

For example, [**ALPHA**], [**GRAPH**], [**ENTER**] means you should first press the ALPHA key, then the GRAPH key, and finally the ENTER key.

**Hexadecimal notation** Hexadecimal numbers in this document are prefixed with 0x. So  $0x20 = \$20 = 20h = 32$  decimal. All numbers in the HoustonTracker 2 user interface itself are hexadecimal, therefore no prefix is used there.

## Terminology

<b>bit</b>	This humble binary digit is at the core of HT2's inner workings. It can take a value of 0 (off) or 1 (on).
<b>byte</b>	Not related to chewing by any means, the byte is an 8-bit number. It can take any value from 0x00 (decimal 0) to 0xFF (255).
<b>duty cycle</b>	The relative amount of time each of the two half-periods of a square wave will consume.
<b>hi-byte</b>	The left two hexadecimal digits, aka most significant byte of a word.
<b>lo-byte</b>	The right two hexadecimal digits, aka least significant byte of a word.
<b>nibble</b>	A single hexadecimal digit. It consists of 4 bits, and can therefore take a value from 0x0 to 0xF (15).
<b>pattern</b>	A list of events that makes up a part of a tune. These can be notes, drum triggers, or effect commands. All patterns in HT2 are 16 steps long.
<b>pitch</b>	The (perceived) frequency of a tone.
<b>shell</b>	A program that facilitates the execution of machine language programs on your TI calculator, among other things. You'll need to install one in order to run HT2.
<b>song sequence</b>	A matrix containing the order of patterns. Think of it as a storyboard, or the song's masterplan.
<b>word</b>	A 16-bit value. Equivalent to two bytes, or four hexadecimal digits.



# 3. Setup

## 3.1 Requirements

### Hardware

- a TI graphing calculator (see Supported Calculator Models)
- a suitable PC $\leftrightarrow$ Calc link cable, e.g. TI SilverLink (recommended), TI GraphLink, homemade parallel or serial cable for data transfers
- a 2.5mm (micro-)jack adapter/cable for sound



You can use a cheap calc-to-calc link cable to make your own 2.5mm adapter.



The plastic base of many adapters is too thick to fit into TI's extra narrow socket. If that's the case, you need to carefully scrape off some plastic from the base of the jack until it fits.

### Software

In addition to the HT2 executable, you will also need to obtain:

- TI linking software for exchanging data between your PC and your calc (TiLP is strongly recommended, though it can be somewhat tricky to install. Check the Troubleshooting guide if you run into problems.)

- a so-called "shell" for your calculator. The following table lists the recommended shells for each model. Other shells may work, but haven't been tested.

model	shell
TI-82	CrASH 1.6
TI-82 Parcus	CrASH_19.006
TI-83, TI-82Stats	Ion 1.6U
TI-83 Plus, TI-84 Plus	Doors CS 7



HT2 is not compatible with MirageOS.



If you don't own a TI calc, you can run HT2 on an emulator. The beta version of `tilem2` is recommended. A Win32 installer can be found here, \*nix users can obtain the development version via *svn checkout* <https://tilem.svn.sourceforge.net/svnroot/tilem/trunk/tilem>. You need to have TiLP installed in any case.

## 3.2 Installation



**After installing HoustonTracker 2, your calculator will not be able to perform most of its regular tasks until HT2 is removed. Do not install HT2 if you need to use your calculator for regular work, such as in school or university.**

The following instructions assume that you have TiLP installed on your computer. If you are using another linking program, then you probably know what to do...

## Installing on TI82/82 Parcus

- If you aren't sure which TI82 version you have, press [MODE], [ALPHA], [LN] to check the ROM version. If it says "19.006", you have a 82 Parcus, else you have a regular 82. Afterwards, press any key except Enter.
- Reset your calculator by pressing [2nd], [+], [3].
- Switch off your calculator, and connect it to your computer with a link cable.
- Switch the calculator back on.
- On your computer, open TiLP. Make sure your calculator and link cable settings are correct - if not, press Ctrl+D to reconfigure.
- Put your calculator into transmission mode by pressing [2nd], [X,T,Θ], [▶], [ENTER].
- In Tilp, go to "File" -> "Restore", and select CRASH.82B (regular TI82) resp. CRASH19006.82b (82 Parcus). Doing so will prompt for confirmation on both TiLP and your calc.
- Assuming all went good, you can now put your calc in transmission mode again, and send ("File" -> "Send Files") ht2.82p (regular) resp. ht2p.82p (Parcus).



**If you intend to use a SilverLink cable in conjunction with a TI-82, do not install the official TI connectivity software on your computer. If you currently have TI Connect installed, or had it installed in the past, make sure you fully remove all of its components, especially the USB driver that came with it.**

## Installing on TI83/TI82Stats

- Reset your calculator by pressing [2nd], [+], [5].
- Switch off your calculator, and connect it to your computer with a link cable.
- Switch the calculator back on.
- On your computer, open TiLP. Make sure your calculator and link cable settings are correct - if not, press Ctrl+D to reconfigure.
- In Tilp, go to "File"->"Send Files", and select ION.83G from the Ion package.
- After Ion has been received by the calculator, press [PRGM], select "ION", and press [ENTER].
- Press [2nd], [+], [2], [7]. Highlight ION and press [ENTER] to delete it. Likewise, delete IONZ.
- Assuming all went good, you can now send ("File"->"Send Files") ht2.83p.

## Installing on TI83 Plus/TI84 Plus

- Reset your calculator by pressing [2nd], [+], [7], [▶], [▶], [ENTER].
- Switch off your calculator, and connect it to your computer with a link cable. This will switch your calc back on.
- On your computer, open TiLP. Make sure your calculator and link cable settings are correct - if not, press Ctrl+D to reconfigure.

- In Tilp, go to "File"->"Send Files", and select DoorsCS7.8xk from the DoorsCS package.
- Assuming all went good, you can now send ("File"->"Send Files") ht2.8xp.



If you don't have enough free memory because you have a large number of archived apps, you can install ht2s.8xp instead.



For faster access, you can use TiLP in GUI-less mode. To do so, open a command prompt (cmd.exe on Win or any shell on \*nix), and enter the following command:

```
tilp [calc-model] [cable-model] [filename]
```

So, to send HT2 to your TI83 Plus via a SilverLink cable, you would type

```
tilp ti83+ SilverLink ht2.8xp
```

### 3.3 Running HT2

Once you've successfully installed HT2, running it is very simple.

- Press [**PRGM**] (TI82/83), resp. [**APPS**] (TI83 Plus/84 Plus).
- Highlight "CRASH" (TI82), "A" (TI83), resp. "DoorsCS" (Plus models), and press [**ENTER**].
- Make sure HoustonTracker 2 is highlighted, and press [**ENTER**] again. Unless an error occurred, HT2 is now running.



Always plug in headphones after you have started HT2, and remove them before you quit the editor. Having headphones plugged in while HT2 is not running will massively slow down the calculator. (In case you accidentally plugged in headphones at the wrong time, don't panic - the calculator hasn't crashed, it's just very slow.)



On TI83/8x Plus, booting and shutting down HT2 may take up to 10 seconds.

## 4. Using HT2

### 4.1 Quickstart Tutorial

The following section will explain how to get started with HoustonTracker 2 in 10 easy steps. If you can't be bothered to read the whole manual, and want to get started with HT2 in 5 minutes, then this section is for you.

**Step 1** Start Houston Tracker, and plug in your audio/headphone cable. Check section 3.3 if you can't figure out how to do that.



Always plug in headphones *after* you've started HT2, and unplug them before you quit.

**Step 2** Activate AutoInc mode by pressing [MODE]. Now, let's enter some patterns into the song sequence. To do so, press the following keys: [0], [0], [0], [1], [0], [2], [0], [0]. Your main screen should now look like this:

```

00 00 01 02 00
01 -- -- -- --
02 -- -- -- --
03 -- -- -- --
04 -- -- -- --
05 -- -- -- --
06 -- -- -- --
07 -- -- -- --
08 -- -- -- --
09 -- -- -- --

```

**HT2**  
■ 123D  
LP 00  
SP 10  
BS 00  
BE 00  
D 0000  
A1 S-



Always fill all four positions of a sequence row to avoid glitches and unwanted effects.

**Step 3** Move the cursor up to the first position in the sequence again. Now press [2nd]. This will bring up the pattern screen.

```

PT 00
OC 0
0 --- 8 ---
1 --- 9 ---
2 --- A ---
3 --- B ---
4 --- C ---
5 --- D ---
6 --- E ---
7 --- F ---

```

**HT2**  
■ 123D  
LP 00  
SP 10  
BS 00  
BE 00  
D 0000  
A1 S-

**Step 4** Let's activate RowPlay mode, so we can hear what we're doing. Press [ALPHA], [MODE].

**Step 5** Ok, let's put some notes into this pattern thing. Press [PRGM], [PRGM].



**Step 6** Wow, that's a pretty low note. Let's change the octave by pressing [GRAPH], [2]. Now we can enter some more notes, for example by pressing [PRGM], [PRMG], [SIN], [SIN], [TAN], [TAN].

PT	00		HT2
0C	2		■ 123D
0	C-0	8	LP 00
1	C-0	9	SP 10
2	C-2	A	BS 00
3	C-2	B	BE 00
4	E-2	C	D 0000
5	E-2	D	A1P S-
6	G-2	E	
7	G-2	F	

**Step 7** Press [ENTER] to hear what we've got so far.

**Step 8** Hmm, sounds pretty boring, right? So let's spice things up with some drums. Press [2nd] to go back to the main screen. Now move the cursor all the way to the right, and press [2nd] again. This will bring up the Drum/FX pattern screen.

PT	00		HT2
			P 123D
0	0000	8 0000	LP 00
1	0000	9 0000	SP 10
2	0000	A 0000	BS 00
3	0000	B 0000	BE 00
4	0000	C 0000	D 0000
5	0000	D 0000	A1P S-
6	0000	E 0000	
7	0000	F 0000	

**Step 9** Press [1], [▼], [2], [2], [1], [▼], [2], [2], [1], [▼], [2], [2], [▼], [2], [▼], [2].

PT		00		MT2	
0	1000	8	1000	P	123D
1	0000	9	0000	LP	00
2	2000	A	2000	SP	10
3	2000	B	2000	BS	00
4	1000	C	0000	BE	00
5	0000	D	2000	D	0000
6	2000	E	0000	A1P	S-
7	2000	F	2000		

**Step 10** Congratulations, you've just composed your first "song" in HoustonTracker 2!

## 4.2 Keys



key	function
[ALPHA][ON]	<b>QUIT</b> HT2. The current tune will be preserved in memory. You can only exit when music isn't playing. On TI83/8x Plus, exiting will take a few moments.
[Y=]	<b>LOAD</b> a song. Prompts prompt for a save slot number (0..7). Needs to be CONFirmed.
[ALPHA][Y=]	<b>SAVE</b> a song. Prompts for a save slot number (0..7). Needs to be CONFirmed.
[WINDOW]	<b>ZAPs</b> (clears) the current <b>TUNE</b> . Needs to be CONFirmed.
[ALPHA][WINDOW]	<b>DELe</b> te save <b>SLOT</b> . Prompts for a save slot number (0..7). Needs to be CONFirmed.
[.]	<b>CONFIRM</b> action. (Cancel action with key [0].)
[2nd]	<b>TOGGLE</b> between <b>SE</b> quence and <b>PaT</b> - <b>terN</b> view. When used on sequence view, it will display the pattern selected by the cursor.
[ALPHA]	Toggle <b>ALPHA</b> mode on and off.
[MODE]	Toggle <b>AUT</b> omatic cursor <b>INC</b> rement on/off (off by default).
[ALPHA][MODE]	<b>TOGGLE ROW</b> Play on and off (off by default). When enabled, plays notes while editing. Beware that RowPlay may ignore some effect settings.
[▲]	Move cursor <b>UP</b> .
[▼]	Move cursor <b>DOWN</b> .
[◀]	Move cursor <b>LEFT</b> .
[▶]	Move cursor <b>RIGHT</b> .
[ALPHA][▲]	Jump to the <b>START</b> of the sequence.
[ALPHA][▼]	Jump to the <b>END</b> of the sequence.
[ALPHA][◀]	Move one <b>PAGE</b> (10 lines) <b>UP</b> in sequence or select the <b>PREVIOUS PaT</b> terN.
[ALPHA][▶]	Move one <b>PAGE</b> (10 lines) <b>DOWN</b> in sequence or select the <b>NEXT PaT</b> terN.
[ENTER]	Start <b>PLAY</b> ing from the start of the song, or <b>STOP</b> playback if player is running.

[−]	Start <b>PLAY</b> ing from the <b>CURRent</b> position in sequence, or hold down to <b>PAUSE</b> play-back while player is running.
[ALPHA][−]	Start playing and <b>LOOP</b> the <b>CURRent</b> position in sequence.
[CLEAR]	Toggle <b>HOLD</b> ing the current row (toggle synth mode)
[,]	<b>MUTE</b> /unmute <b>CHannel1</b> .
[ALPHA][,]	<b>UNMUTE ALL</b> channels.
[(]	<b>MUTE</b> /unmute <b>CHannel2</b> .
[ALPHA][[(	<b>MUTE</b> /unmute <b>DRUMS</b> .
)]	<b>MUTE</b> /unmute <b>CHannel3</b> .
[ALPHA][)]	<b>MUTE ALL</b> channels.
[0]	Input <b>0</b> at cursor, <b>DELe</b> te a <b>NOTE</b> , or <b>CANCEL</b> action, depending on context.
[ALPHA][0]	<b>REMO</b> ve the currently selected <b>PaTterN</b> from the sequence.
[1]..[9]	Input <b>1..9</b> at cursor.
[MATH]	Input hex digit or note <b>A</b> at cursor, depending on context.
[ALPHA][MATH]	In note pattern view, input note <b>A#</b> at cursor.
[MATRIX]/[APPS]	Input hex digit or note <b>B</b> at cursor.
[PRGM]	Input hex digit or note <b>C</b> at cursor.
[ALPHA][PRGM]	In note pattern view, input note <b>C#</b> at cursor.
[ $x^{-1}$ ]	Input hex digit or note <b>D</b> at cursor.
[ALPHA][ $x^{-1}$ ]	In note pattern view, input note <b>D#</b> at cursor.
[SIN]	Input hex digit or note <b>E</b> at cursor.
[COS]	Input hex digit or note <b>F</b> at cursor.
[ALPHA][COS]	In note pattern view, input note <b>F#</b> at cursor.
[TAN]	Input hex digit or note <b>G</b> at cursor.
[ALPHA][TAN]	In note pattern view, input note <b>G#</b> at cursor.

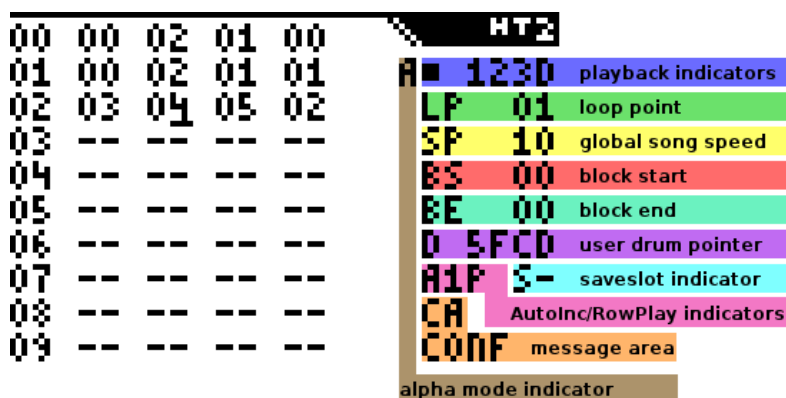
[TRACE]	<b>SET</b> the global song <b>SPeeD</b> . Refer to the section on Global Variables for details on this setting.
[ALPHA][TRACE]	<b>SET</b> the <b>USer Drum</b> pointer. Any value between 0x0000 and 0xFFFF is permitted. However, it is recommended to stick with values in the range of 0x0000 - 0x7FFF, as higher values may produce unpredictable results.
[GRAPH]	<b>SET</b> the current <b>OCTave</b> . Only works on note pattern screens. Valid values are 0..6.
[X,T,Θ]	<b>TRanSP</b> ose current pattern <b>UP</b> by one half-tone. Out-of-range notes will be replaced with rests.
[ALPHA][X,T,Θ]	<b>TRanSP</b> ose current pattern <b>DOWN</b> by one half-tone. Out-of-range notes will be replaced with rests.
[x <sup>2</sup> ]	<b>QUICK SET</b> the <b>Loop Point</b> to the current row in sequence (sequence screen only). Changes will not take effect until the next player restart.
[ALPHA][x <sup>2</sup> ]	<b>SET</b> a new <b>Loop Point</b> value. Only works on sequence screen. Changes will not take effect until the next player restart.
[LN]	<b>QUICK SET</b> the copy <b>Block Start</b> to the current row in sequence (sequence screen only).
[ALPHA][LN]	<b>SET</b> a new copy <b>Block Start</b> value (sequence screen only).
[STO▶]	<b>QUICK SET</b> the copy <b>Block End</b> to the current row in sequence (sequence screen only).
[ALPHA][STO▶]	<b>SET</b> a new copy <b>Block End</b> value (sequence screen only).
[ALPHA][÷]	<b>PaSTe</b> the selected <b>BLock</b> over the current and following rows. The target row must be outside of the selected block.

[×]	INSert the selected <b>BLock</b> at the current row.
[ALPHA][×]	DELEte the selected <b>BLock</b> .
[-]	INSert a copy of the current <b>ROW</b> .
[ALPHA][-]	DELEte the current <b>ROW</b> .
[+]	PUT the <b>NEXT</b> unused <b>FREE</b> pattern in sequence at the current position.
[ALPHA][+]	CLoNe the current <b>PaTterN</b> and insert the clone at the current cursor position.

---

## 4.3 Composing Music

### 4.3.1 Modes, Indicators, and Global Variables



**Modes and Indicators** There are a number of different indicators, which will tell you about various modes and editing settings that are currently applied.

The Alpha mode indicator comes in the shape of a capital letter A, which is displayed below the diagonal left edge of the HT2 logo. When the A is visible, then Alpha mode is active. The Alpha mode is HT2's version of the SHIFT key on a PC - except that you press it *before* pressing another key, not together with the second key. Many keys in HT2 have a primary

and secondary function. In Alpha mode, the secondary functions will be triggered. You can toggle Alpha mode on and off by pressing - you guessed it - **[ALPHA]**. Alpha mode will be turned off automatically once an Alpha mode action has been performed.

The playback indicators are located directly below the HT2 logo. The first position tells you if the music player is currently stopped or running. While stopped, it will display a ■, when running, it will display the capital letter P. Next to this position, you will find four positions which normally read "123D" (for channel 1, 2, 3, and drums respectively). These positions indicate which channels are currently muted. Muted channels will display a "-" instead of their channel number/letter.

The AutoInc/RowPlay indicators tell you whether the auto-incrementing and RowPlay functions are currently activated. They are located in the lower third of the right screen side, just below the the global variables. The AutoInc indicator displays "A0" when auto-incrementing is deactivated (default), and A1 when activated. Auto-incrementing means that after you enter a value or note in HT2, the cursor will automatically advance to the next position. You can toggle AutoInc with key **[MODE]**. The RowPlay indicator tells you whether RowPlay mode is currently active. If so, it will display the capital letter P. When entering a note, HT2 will automatically play the current pattern position (including all channels that are not muted). You can toggle RowPlay by pressing **[ALPHA]**, **[MODE]**.



RowPlay mode can't reliably detect all effect settings that apply to the current position, so depending on the circumstances, it may not reflect how the position will actually sound when the whole song is played back.

Located to the right of the AutoInc/Rowplay indicators is the saveslot indicator. It will display the last used save slot once the load/save function has been used.

Below the AutoInc/RowPlay indicators, there is the general purpose message area. Here, various important messages will be displayed, such as occurring errors, or requests for the user to confirm an action. Confirming is done with



key [.] , cancelling is done with key [0].

**Global Variables** The loop point defines what row in the sequence the player will loop back to after it has reached the end of the song. Pressing [ $x^2$ ] will set the loop point to the row currently highlighted by the cursor. To set it to another row, press [ALPHA], [ $x^2$ ], then enter a 2-digit hex number.

The global song speed can be changed by pressing [TRACE], followed by a 2-digit hex number. The number refers to the number of ticks one row will last. So generally speaking, a higher number means lower speed.

Any number in the range of 0x01 - 0x3F will set the number of ticks directly. Adding 0x40 to the value will reduce the length of the first tick by  $\frac{3}{4}$ , adding 0x80 will reduce it by  $\frac{1}{2}$ , and adding 0xc0 will reduce it by  $\frac{1}{4}$ . Thus, setting the speed to 0x08 will result in a row length of 8 ticks, 0xc8 will result in a row length of  $7\frac{3}{4}$  ticks, 0x88 will result in a row length of  $7\frac{1}{2}$  ticks, and 0x48 will result in a row length of  $7\frac{1}{4}$  ticks. Note that reducing the length of the first tick will have a noticeable impact on various effects including 2xx/3xx (pitch slides), 8xx (execute note table), and Cxx (note cut).

0x00 (aka the "drone mode") is an exception. It is the slowest possible setting, equivalent to a row length of 256 ticks (minus a fraction of a tick when adding using 0x40/0x80/0xc0).

To set the user drum pointer, press [ALPHA], [TRACE], followed by a 4-digit hex number. See the section on Drums for details about the user drum pointer.



Changes to the global variables are not effective until you restart the player.

### 4.3.2 Editing the Song Sequence

	channel 1	channel 2	channel 3	drum&fx channel	
00	00	02	01	00	MT2
01	00	02	01	01	A 1230
02	03	04	05	02	LP 01
03	---	---	---	---	SP 10
04	---	---	---	---	BS 00
05	---	---	---	---	BE 00
06	---	---	---	---	D 5FCD
07	---	---	---	---	A1P S-
08	---	---	---	---	CA
09	---	---	---	---	CONF

sequence row no.

The sequence screen is where you build the general structure of your song. It consists of four editable columns, each containing a list of patterns. Each column represents a different channel. To toggle between the main (sequence) screen and the pattern screens, select a pattern in the sequence with the cursor, then press [2nd].

The first 3 columns represent the 3 tone channels, ie. they contain note patterns. The last column represents the drum channel and effects settings, ie. it contains fx patterns. The tone channels differ somewhat in sound and features. Channel 1 features a note cut Effects Reference. It can also play noise and glitchy sounds in addition to the regular tone. Channel 2 is somewhat louder than the other channels, and features some advanced duty modulation effects, including a SID-style duty cycle sweep. Channel 3 features pitch slides, note (arpeggio) tables, and another set of glitch sounds. All channels support a variable duty cycle.

**Moving around on the sequence screen** Keys [▲], [▼], [◀], and [▶] move the cursor around on the sequence screen. Pressing [ALPHA], [◀] resp. [ALPHA], [▶] will move the cursor up/down by 10 rows. Pressing [ALPHA], [▲] will move the cursor to the start of the sequence. Like-

wise, pressing **[ALPHA]**, **[▲]** mode will move the cursor to the end of the sequence.

**Editing the sequence** Use keys **[0]..[9]** and **A..F** (**[MATH]**, **[MATRX]** resp. **[APPS]**, **[PRGM]**, **[ $x^{-1}$ ]**, **[SIN]**, and **[COS]**) to enter pattern numbers. You can have a maximum of 128 note patterns, and 64 fx patterns, so the highest pattern numbers are 0x7F and 0x3F respectively. Note patterns are shared across all 3 tone channels, so you can use any note pattern on any channel. If you try to enter an invalid pattern number, HT2 will automatically replace it with an arbitrary valid one.



Note patterns and fx patterns use a different numbering, e.g. note pattern 01 is different from fx pattern 01.

To automatically put the first unused pattern at the current sequence position, press **[+]**. To clone the pattern at the current position (copy its contents to a new, unused pattern), press **[ALPHA]**, **[+]**. To delete the pattern number at the current position, press **[ALPHA]**, **[0]**.

You can press **[-]** to copy the current row. This will insert the copy at the current position. Press **[-]** to delete the current row.



Always fill all four sequence positions in a given row. Pressing play on a sequence row that has empty positions will have undesired side effects.

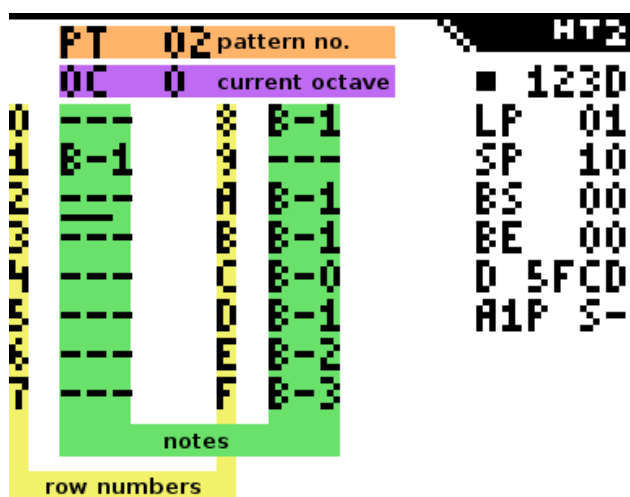
**Block Operations** HT2 uses two block markers, block start (BS) and block end (BE) to select a section of sequence data for copying, pasting, and cutting. Block operations work only on the whole sequence, not on individual channels.

To set the block start to the current row in the sequence, press **[LN]**. Likewise, to set the block end to the current row, press **[STO▶]**. Pressing these keys in **[ALPHA]** mode will allow you to set the BS/BE positions manually.

To copy the marked block and insert it at the current cursor position, press [X]. Pressing [ALPHA], [÷] will paste the block over the following sequence rows. Note that you cannot insert/paste a block into itself, so the target must always be outside the selected block.

Press [ALPHA], [X] to delete the selected block and move the following sequence data up.

### 4.3.3 Editing Note Patterns



All note patterns have a fixed length of 16 steps, which are organized in two columns à 8 steps.

Notes can be entered with keys [MATH] (A), [MATRX] resp. [APPS] (B), [PRGM] (C), [ $x^{-1}$ ] (D), [SIN] (E), [COS] (F), and [TAN] (G). Sharp notes (black keys) can be reached with [ALPHA] + note key.

In order to hear notes while you're entering them, you can toggle RowPlay by pressing [ALPHA], [MODE]. A "P" will be displayed next to the AutoInc indicator to let you know that RowPlay is active. Note that RowPlay is somewhat limited. Among other things, it may ignore current effect settings. Also, it will fail if the pattern you're editing is not on the current step in the song sequence.

To change the current octave, press [GRAPH], then enter a number between 0 and 6. You can also edit octaves manually in the pattern.

- ☞ You can cycle through note patterns with [ALPHA], [◀] and [ALPHA], [▶].
- ☞ In order to use a different pattern length, use the B00 effect command.

#### 4.3.4 Editing Drum/Effects Patterns

The screenshot shows a music sequencer interface. At the top left, it says "PT 01 pattern no." and at the top right "HT2". The main area is a grid of hex digits. The left column contains row numbers 0-7. The right column contains hex digits. The grid is divided into four sections: drums (green), effect commands (blue), and effect parameters (pink). To the right of the grid, there is a list of effect commands and parameters: LP 01, SP 10, BS 00, BE 00, D 5FCD, and A1P S-.

Like with note patterns, fx patterns are organized in two columns à 8 steps. Each step consists of four hex digits. The *first digit* sets the drum sound to be used on that step. A value of 0 means no drum. Refer to the following section to learn more about drums in HT2.

The *second digit* sets the effect command. A value of 0 means no effect. Finally, the *last two digits* set the effect parameters. Refer to the Effects Reference to learn all about effects.

#### 4.3.4.1 Drums

There are 15 different drums to chose from (0x1..0xF). Some of the drums use the TI-OS as sample data, so their sound may vary across different calculator models.

Drums can be played in different modes, which affects the way they sound. Use command Cxx to change the drum mode, with xx = 0x00..0x4f. There are 80 different drum modes, though not all of them are particularly useful.

Setting a non-zero value for the *lower nibble* of the parameter causes the drum data to be manipulated in various ways. Refer to the appendix on Synthesis Techniques for details about this functionality.

The *upper nibble* of the parameter defines the behaviour of the drum data pointer. The effects are as follows:

mode	cmd	effect
0	C00	Increment pointer, ie. use the "standard" drum set.
1	C1x	Decrement pointer, ie. use the "alternative" drum set. Generally speaking, these drums are less useful than the standard set. Also, some drums will produce unpredictable results, namely drum 0, 1, C, and D.
2	C2x	Increment and loop pointer. Even less useful than mode 1, and suffers from the same problems. Also, this mode causes a slight global pitch shift.
3	C3x	Decrement and loop pointer. Like mode 2, with different sounds.
4	C4x	Don't move pointer. Instead of drums, in this mode triggering a drum will play a fixed, most likely out-of-tune frequency. This mode causes a slight global pitch shift.

Drum 0xE and 0xF are special cases.

0xF is the user drum pointer, which can be configured individually. To change it, press [ALPHA], [TRACE]. The user drum pointer can be set to any value between 0x0000 and 0xFFFF. However, if you want to play it safe

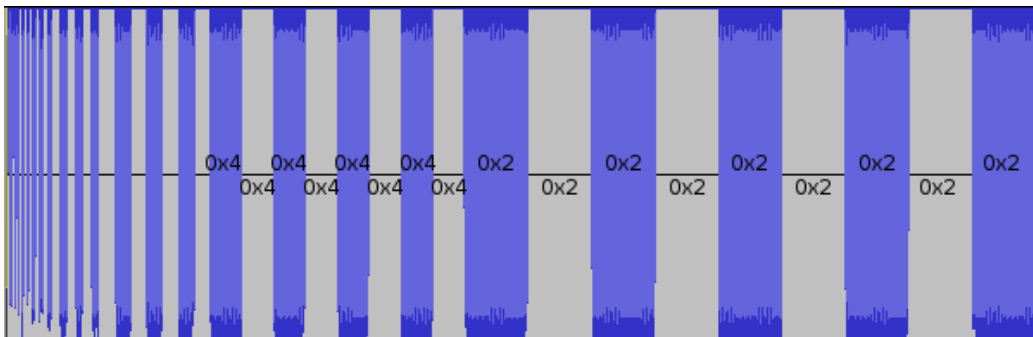
you should keep the pointer in range 0x0000 - 0x7FFF. This is because the user drum is actually a pointer to ROM/RAM. Pointing the user drum to RAM (0x8000 and up) will give unpredictable results, as the RAM contents change frequently.

0xE is the user defined sample. This feature is a bit quirky in use. In order to create a custom sample, go to FX pattern 0x20. Now, instead of entering the usual drum triggers, fx commands, and fx parameters, you enter your sample data. You can use more than one pattern, so after you've filled up FX pattern 0x20, go to 0x21.

The data is in 1-bit PWM format, meaning each byte denotes a phase length. Think of it as the amount of time taken until the beeper output will toggle again. The values are inverse, ie. a lower value denotes a longer phase. Yes, it's a bit confusing... As an example, this is what drum sample 1 (the kick drum) looks like:

```
0x80, 0x80, 0x70, 0x70, 0x60, 0x60, 0x50, 0x50, 0x40, 0x40,
0x40, 0x30, 0x30, 0x30, 0x30, 0x20, 0x20, 0x20, 0x20, 0x20,
0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x8, 0x8, 0x8, 0x8, 0x8,
0x8, 0x8, 0x4, 0x4, 0x4, 0x4, 0x4, 0x4, 0x4, 0x4, 0x2, 0x2,
0x2, 0x2, 0x2, 0x2, 0x2, 0x2, 0x2, 0x0
```

This will give you a waveform looking like this:





The last byte of the sample must be 0.



You can even define a second drum sample, which you can then trigger in drum mode C1x/C3x. This sample also starts at FX pattern 0x20, but needs to be written backwards, so the second byte of the sample is on the last position (FX parameter of the last row) of FX pattern 0x1F, third byte is just before that (drum/FX command of the last row), and so forth.



#### 4.3.4.2 Effects Reference

cmd	effect	description
1xx	SET PAN	<p>Set panning for all channels. To determine how the panning will be set, you need to look at the individual bits of the effect parameter, starting at the rightmost one (aka bit 0).</p> <p>bit 0 set: pan ch1 right (add 0x1 to xx) bit 1 set: pan ch1 left (add 0x2) bit 0,1 reset: pan ch1 center (add nothing) bit 2 set: pan ch2 right (add 0x4) bit 3 set: pan ch2 left (add 0x8) bit 2,3 reset: pan ch2 center (add nothing) bit 4 set: pan ch3 right (add 0x10) bit 5 set: pan ch3 left (add 0x20) bit 4,5 reset: pan ch3 center (add nothing) bit 6 set: pan drums right (add 0x40) bit 7 set: pan drums left (add 0x80) bit 6,7 reset: pan drums center (add nothing)</p>

#### Examples:

*Pan all channels to center*

$0x0 + 0x0 + 0x0 + 0x0 = 0x0 \rightarrow$  use 100

*pan all channels right*

$0x1 + 0x4 + 0x10 + 0x40 = 0x55 \rightarrow$  use 155

*pan ch1 left, ch2 center, ch3 right, drums center*

$0x2 + 0x0 + 0x10 + 0x0 = 0x12 \rightarrow$  use 112

If you're having trouble figuring out the right values, you can use this online tool by garvalf to calculate the parameters.

---

2xx	PITCH SLIDE UP CH3	Perform an upward pitch slide on channel 3. xx defines the speed of the slide, lower values mean slower slides. xx can be any value, but beware that the pitch counter will eventually wrap. 200 disables the effect.
3xx	PITCH SLIDE DOWN CH3	Perform a downward pitch slide on channel 3. Using this will disable effect 9xx. xx defines the speed of the slide, lower values mean slower slides. xx can be any value, but beware that the pitch counter will eventually wrap. 300 disables the effect.
4xx	DUTY CY- CLE/NOISE CH1	Set the duty cycle for channel 1, and toggle noise mode. xx <= 0x80 - set duty cycle and disable noise mode xx > 0x80 - set duty cycle and enable noise mode
5xx	DUTY CY- CLE/SWEEP CH2	Set the duty cycle for channel 2, or enable duty cycle sweep. xx <= 0x80 - set duty cycle and disable duty cycle sweep. A value of 0x80 produces the default 50:50 wave. Very low values will cause glitches. xx > 0x80 - enable SID-style duty cycle sweep. Sweep speed = (xx & 0x7F), 581 will produce the classic sweep effect known from HT versions <= 2.20. Some parameters are shared with effect 7xx, hence these two effects impact each other.

6xx	DUTY CY- CLE/GRIND CH3	Set the duty cycle for channel 3, and toggle grind mode. xx <= 0x80 - set duty cycle and disable grind mode. xx > 0x80 - set duty cycle to (xx*2)&0xff and enable grind mode.
7xx	AUTO CHORD/ DRAWBAR CH2	Add a chord or drawbar organ effect to channel 2. xx < 0x80 - enable unsynced auto chord. The chord created varies depending on the note used, and is not necessarily harmonic. xx >= 0x80 - enable synced auto chord (drawbar organ effect). This will produce an octave chord, depending to some extent on the current duty setting. A higher value for xx will generally produce stronger harmonics. 700 switches off the effect. Some parameters are shared with effect 5xx, hence these two effects impact each other. See the description of effect 5xx for details.
8xx	EXEC NOTE TABLE CH3	Execute a given pattern as a note table for channel 3. This effect operates on a per-tick basis. Execution starts after the first tick. xx is the pattern to be executed as note table. If the current tempo is greater than 0x10, table execution will continue at the following pattern. To disable the effect, set xx to a value greater than 0x7F. Using this effect will disable the Cxx (note cut ch1) effect.
9xx	GLITCH CH3	Add a nasty glitch effect to channel 3. xx can be any value, 900 turns off the effect.

Axx	CH3 PHASE	Set the phase offset for channel 3. This has little effect on itself, but will cause interference when used together with another channel that plays the same note as ch3. In this case, it can be used as a primitive form of volume control. xx can be any value, values around 0x80 will work best. A00 turns off the effect.
Bxy	BREAK PTN/LOOP SECTION	xy = 0 - break pattern immediately and jump to the next position in the sequence. B00 is ignored on the first line of a pattern. xy > 0 - jump back y rows in the pattern, repeating the section x times. The Bxy effect should not be nested (ie. don't put a Bxy loop within another Bxy loop), because doing so will lead to an infinite loop. Beware, HT2 does not check this.
Cxx	NOTE CUT CH1	Cut the note on channel 1 after xx ticks. Using this effect can make the sound output slightly more noisy. Use C00 to disable the effect. Using this effect will disable the 8xx (exec note table ch3) effect.
Dxx	DRUM MODE	Set the drum mode, where xx is 0x00..0x4F. Refer to the Drums section for details.

Exx	EXTENDED FX	<p>Various extended effects. Using this effect causes a slightly longer than normal delay when triggered, leading to additional transition noise.</p> <p>Exx with xx &lt; 0x40 - execute up to 5 effects stored at the start of effects pattern xx at once. Effects Bxx and Exx will be ignored.</p> <p>E80 - reset all effects and restore global variables to their defaults.</p> <p>E81 - as above, but does not reset speed.</p> <p>E82 - as above, but does not reset speed and duty cycles.</p> <p>E83 - as above, but does not reset speed, duty cycles, and panning settings.</p>
Fxx	SET SPEED	<p>Set the current speed. This command temporarily overrides the global speed setting, but does not permanently change it. For details on how the speed setting works, refer to the section on global variables.</p> <p>TIP: When entering this effect while the player is running, make sure to enter the parameter first, and then the F command.</p>

## 4.4 Loading, Saving, and Backing Up

**Saving** To save your current work in progress, press [ALPHA], [Y=]. The letters "SA" will appear in the message area. Now, enter the number of the save slot you want to save (0..7), and confirm by pressing [.] (If you got here by accident, press [0] to cancel.)

If the save was successful, error code E0 will be displayed in the message area. If the save was unsuccessful due to insufficient memory, error code E2 will

be displayed instead. Note that you may run out of memory rather quickly. You will always be able to maintain at least two save slots, however.



Any data after the first unfilled position (- -) in the sequence is *not* saved.

**Loading** To load a previously saved tune, simply press [Y=]. The letters "LD" will appear in the message area. Now, enter the number of the save slot you want to load (0..7), and confirm by pressing [.]. (If you got here by accident, press [0] to cancel.) The tune in question will be loaded, or error code E5 will be printed if the selected save slot was empty.

**Deleting a save slot** In case you run out of memory, or want to get rid of one of your save slots for other reasons, you can specify a save slot to be deleted. In order to do so, press [ALPHA], [WINDOW]. The letters "DS" will appear in the message area. Now, enter the number of the save slot you want to delete (0..7), and confirm by pressing [.]. (If you got here by accident, press [0] to cancel.) And that's all you need to know about deleting save slots!

**Backups** Creating a backup of your HT2 songs is trivial: Simply send the HT2 PRGM file to your computer or another calc of the same model. The program file contains all the save slots as well as the currently loaded song.



You can use the ht2util savestate manager to extract/insert save slots into your HT2 PRGM file.

## 4.5 Error Codes

If a major error happens, HT2 will attempt to detect it, and print an error code in the message area. An special error message will also be displayed if saving a backup of the current song was successful.

The following table lists all possible error codes:

<b>code</b>	<b>description</b>
E0	<b>SAVE OK:</b> Saving backup was successful, no error occurred.
E1	<b>INVALID BLOCK:</b> Block End is set before Block Start, or the row where the block should be inserted/pasted to is within the selected block.
E2	<b>INSUFFICIENT MEMORY:</b> There is not enough RAM to perform the desired action (usually saving).
E3	<b>INVALID PTN:</b> An invalid pattern number was set in the song sequence.
E4	<b>LP INVALID:</b> Invalid loop point. Row to loop to is empty.
E5	<b>FILE I/O ERR:</b> Loading song failed. Save slot is empty or corrupt.
E6	<b>SAVESTATE ERR:</b> Wrong savestate version, or savestate corrupt.
E7	<b>NO FREE PTN:</b> Could not find a free pattern.
E8	<b>BLOCK TOO LARGE:</b> The selected block is too large to be inserted.

Small errors (like trying to set an inexistent note or an invalid pattern number) will be simply ignored by HT2.

## 5. The ht2util Savestate Tool

**ht2util** is a PC-based savestate manager application for HoustonTracker 2. It allows you to export and import songs from/to the savestate block of your HT2 binary, among other things.

Even though ht2util comes bundled with HT2 releases, it is maintained separately. Check the ht2util git for the latest version.

**Installation** ht2util works as a stand-alone application, no installation is required. Linux users can compile the program from source by running "make" in the /ht2util-gui directory. ht2util depends on wxWidgets 2.8 or higher, so install that first if necessary. Most Linux distros provide a native wxWidgets implementation in their repositories.

**Usage** Always start by loading a HT2 program file. This can be a blank ht2.8\*p, or one you've backed up from your calculator. Once the file has loaded successfully, the contents of its savestate table will be displayed in the left panel.

To *import* a savestate, use "File→Insert savestate...", or drag one or more files from the right-hand (filebrowser) panel to the left-hand (savestate) panel. If the savestate was extracted from an older version of HT2, ht2util will offer to automatically upgrade the it. This will always work with stable HT2 releases (except for some effects/settings that can not be upgraded automatically), but might, in rare cases, cause some problems with beta versions.

To *delete* one or more savestates, mark the desired state(s) in the savestate panel, then use "File→Delete savestate" or simply press "Delete".



To *extract* one or more savestates, mark the desired state(s) in the savestate panel and then use "File→Extract savestate..." to save it with a ".ht2s" extension. You can also simply drag the desired savestate(s) from the savestate panel to the filebrowser panel; this will save the state(s) with a generic filename.

Furthermore, you can decompress and export savestates to .asm, however this is not particularly useful yet unless you're working with the HT2 source.



ht2util does not automatically update effect commands yet. If you've imported a savestate from an older version of HT2, you will need to adjust those manually. Refer to the changelog to see which commands need to be updated.

## 6. Thanks, Credits, and Greetings

Thanks to...

- TylerBarnes, Tronimal, garvalf, Flashbob, giako9000, nonfinite, and Imaginary for beta testing
- Jankenpopp and unexpectedbowtie for figuring out the install process under non-standard configurations
- extra thanks to garvalf for preparing the HT2 cheatsheet
- Shiru, introspec, and Alone Coder for their expertise and advice
- Brandon Wilson, Lionel Debroux, Benjamin Moody, critor, and everybody who has been hacking/documenting TI calcs

The TI-82 image that was used as the basis of the keymap was authored by Martin Olsson, and is released under a GFDL/Creative Commons BY-SA license.

# Appendices

# A. Building from Source

The latest version of the HT2 source code can always be found at [.](#)

HoustonTracker 2 is developed under Linux. If you're running Linux or another \*nix derivate, building HT2 is a matter of simply running the build script, provided you have a few basic tools installed. If you're developing on Windows or another non-Posix compliant system, you'll need to provide your own build script or build from hand.

## Building Requirements

- The Pasm assembler must be installed and present in your search path (unless you want to adapt the sources for another assembler).
- For building documentation, pdflatex must be installed and present in your search path.
- Optionally, you may want to install an TI calculator emulator. For \*nix systems, using the development version of tilem2 is recommended.

**Building with the Build Script (\*nix only)** It's as simple as opening a terminal, navigating to the folder containing HT2, and typing

```
./build.sh [-model] [-docs] [-test]
```

where *[-model]* is one of

-82        build for TI-82  
-8p        build for TI-82 Parcus  
-83        build for TI-83, TI-82 Stats  
-8x        build for TI-83 Plus, TI-84 Plus etc.  
-8xs      build "small" version for TI-83 Plus, TI-84 Plus etc.  
-all      build all targets.

Not specifying a model has the same effect as passing "-all". You can additionally build documentation by passing the "-docs" flag.

You can test the build by passing the "-test" flag. This requires that you have a test script set up as `_scripts/test.sh`. The `-model` argument is passed to `test.sh`. A simple test script might look like this:

```
#!/bin/sh
```

```
tilem2 -a -r "/path/to/rom/xyz.rom" -m [model] [build]
```

where *[model]* is one of `ti82`, `ti83`, or `ti83p`, and *[build]* is the path to the `ht2*.8*p` file that should be tested. Optionally you can pass `-p [macro]` to trigger suitable keypress macro recording from `tilem2`.

**Building by Hand** The following steps are required to build HT2 by hand:

1. Assemble the sources with the following command:

```
pasmo --equ MODEL=[model] --alocal main.asm main.bin
```

where *[model]* can be either 1 (TI-82 build), 2 (TI-83 build), 3 (Plus models build), 4 (TI-82 Parcus build), or 5 (Plus models small build).

2. Pack the resulting `main.bin` into a TI executable (82p, 83p, 8xp) using a packer of your choice (`oysterpac`, `bin8x`, etc...).

## B. Data Format Specification

**Work Area Data Format** The work area is where HT2 stores the currently loaded song. It begins at label "musicData", and is 5125 bytes in size. The actual address in memory depends on the version of HT2.

offset	length	description
+0	byte	global speed
+1	word	user drum pointer
+3	byte	loop point
+4	1 KB	pattern sequence list. 1 byte per pattern. Order is ch1, ch2, ch3, fx. Empty positions are filled with 0xFF.
+1028	byte	one 0xFF byte to mark the end of the sequence list.
+1029	2 KB	note patterns. Each note pattern has an uncompressed length of 16 bytes. Empty positions contain 0x00.
+3077	2 KB	FX patterns. Each fx pattern has an uncompressed length of 32 bytes. Empty positions contain 0x00.

**Compressed Savestate Format** HT2 uses a very simple compression scheme to store song data backups. They are set up as follows:

offset	length	description
+0	byte	global speed
+1	word	user drum pointer
+3	byte	loop point
+4	?	pattern sequence list. 1 byte per pattern. Order is ch1, ch2, ch3, fx.
?	byte	one 0xFF byte to mark the end of the sequence list.
?	?	note patterns. if byte value at offset $\geq$ 0xE0, then the following (value - 0xDF) patterns are empty. else, if byte at offset $\geq$ 0xD0, then the following (value - 0xCF) rows are empty. else, if byte at offset $<$ 0xD0, it's a regular note value.
?	byte	one 0xFF byte to mark the end of the note pattern area.
?	?	FX patterns. FX patterns are preceded by their pattern number. A pattern number of 0xFF signals that the savestate contains no FX patterns. If bit 7 of the pattern number is set, it's the last pattern to be loaded. Setting bit 7 is optional if the last saved FX pattern is 0x3F.

Start and end addresses of backup savestates are stored in the savestate lookup table, which is located directly before the compressed savestate buffer. Since version 2.20, the savestate LUT is marked in memory with the ASCII string "XSAVE".

## C. Synthesis Techniques

*The information given in this appendix may be out of date, and may not provide an accurate description of the techniques used in the current version of HT2.*

The following section describes how the multi-channel synthesis is achieved in HT2, and how the effects are produced.

**General Notes** Sound mixing in HT2 is achieved through a 1-bit synthesis technique called Pulse Interleaving.

Typically, a 1-bit DAC can only output one single square wave frequency. What Pulse Interleaving does is interlace the multiple software channels at a rapid rate. Due to their own hardware latency, common output devices such as speakers or headphones cannot fully adjust their speaker cones at this rate. Therefore the cones will be left in a floating state between full extension and full contraction. This is how the different volume levels needed to mix HT2's four software channels are achieved.

The average time taken to interlace one sample of all 4 channels is  $(1/16300)$ s, meaning channel states are swapped at a rate of about 65200 Hz. The exact timings depend on various factors such as actual tone frequencies, effect and drum settings, hardware model, and battery state.

**Tone Channels** Once per interlacing cycle, a 16-bit constant (the "frequency base value") is added to a 16-bit counter. The hi-byte of the counter is then compared against a threshold value. If the threshold constant is



greater than the hi-byte of the add counter, the output state will be high (1), otherwise it will be low (0).

The threshold value determines the duty cycle ratio. When the threshold is 0x80, the output state will be high 50% of the time, and low for the other 50%. When the threshold is, say, 0x40, the ratio will be 25%:75%.

The 9xx effect (glitch channel 3) is achieved by constantly rotating the hi-byte of the base frequency value. Similarly, the A01 effect (noise/glitch channel 1) is achieved by rotating the hi-byte of the add counter. This may or may not generate enough pseudo-randomness to produce a noise-like effect, depending on the frequency base value.

The D01 effect (duty cycle sweep) is achieved by incrementing the threshold value once every interlacing cycle. As it is an 8-bit value, it will wrap to 0 once it reaches 0xFF.

The pitch slides are achieved by adding a constant to the frequency base value once per interlacing cycle.

**Drum Channel** Drums are generated from values in either ROM or RAM. The values read will be interpreted as 8-bit counters, which are decremented once per interlacing cycle. When the counter reaches 0, the output state of the drum channel is toggled, and the next value is read from ROM/RAM. When a 0-byte is read, the output stops.

Drums 0x1, 0xC, and 0xD are hard-coded in RAM. The "user drum sample" (drum 0xE) is effectively a pointer to FX pattern 0x20. The "user drum pointer" (drum 0xF) is a simple user-definable pointer which can point to either RAM or ROM. All other drums point to ROM, which explains why they sound different depending on what calculator model is used.

The "drum mode" effect (Cxx) will do two things. The upper nibble of the effect parameter determines the direction of the ROM/RAM pointer used to retrieve the counter values. If the upper nibble is 0, the pointer will move forward. If 1, it will move backward. If 2 or 3, only the lo-byte of the pointer will be in/decremented, which means the pointer stays on the same 256-byte memory page. If 4, the pointer will not be moved at all.

The lower nibble of the effect parameter will determine how the values are

modified after they are read from memory. The manipulations are as follows:

<b>value</b>	<b>effect</b>
0x0	no manipulation
0x1	transform into binary coded decimal
0x2	multiply by 2
0x3	divide by 2
0x4	1's complement
0x5	use lo-byte of pointer as value
0x6	add lo-byte of pointer
0x7	add hi-byte of pointer
0x8	subtract hi-byte of pointer
0x9	subtract lo-byte of pointer
0xA	logical AND with hi-byte of pointer
0xB	logical AND with lo-byte of pointer
0xC	logical OR with hi-byte of pointer
0xD	logical OR with lo-byte of pointer
0xE	exclusive OR with hi-byte of pointer
0xF	exclusive OR with lo-byte of pointer

In some drum modes, the drum channel will produce sound even if a 0-byte is encountered in the data. This is due to the fact that the modifications are applied before checking if the value read from memory is 0.

For a more detailed description on 1-bit sound techniques, check out [How to Write a 1-Bit Music Routine on the 1-bit Forum](#).

## D. Troubleshooting

### D.1 I'm having problems installing/configuring TiLP on Windows.

**I'm getting weird errors when trying to install/run TiLP on Vista/7/8/10.** Check the extended Win32 readme, it has solutions for the most common problems.

**TiLP refuses to run on my Windows XP.** Windows XP is no longer supported in TiLP2 1.17. You'll need to use an older version.

**TiLP fails to fetch the GTK+ installer.** You need to install the GTK+ package manually. Afterwards, make sure you uncheck the GTK+ installation box in the TiLP installer.

**My TI-82 isn't auto-detected.** That's ok, just configure the Device Settings manually and you should be good to go.

### D.2 I can't figure out how to install TiLP on OS X.

An easy way to install TiLP is by using MacPorts. After you have set up MacPorts, you can install TiLP with the command

```
[sudo] port install tilp2
```

Thanks to unexpectedbowtie for coming up with this solution!

Alternatively, you could use the official TI-Connect linking software. However, TI-Connect does not fully support the TI82, so you won't be able to upload HT2 on that model. Alternatively, you could run Linux in a virtual machine, and use TiLP from there.

## D.3 How do I configure my GrayLink cable under Windows?

### 1. Install a driver for GrayLink

1. boot computer and do not plug your link device
2. If your graylink cable is PL2303TA based, you need this specific usb-serial driver (others will not work)
3. plug your link device then check in your device Manager to see if everything is ok, set the COM port to 1, 2, 3 or 4... more than 4 will not work.
4. unplug your link device
5. reboot computer

### 2. Install TILP2

1. do not plug the link device
2. first install GTK+ 2.12.9-win32 dependencies
3. then install TiLP (be sure to uncheck GTK+ installation box, because you already did it)

4. plug the link device
5. open TILP, then 'control+D' and set your connection parameters:
  - (a) cable → graylink
  - (b) port → the COM port you just set in #1
  - (c) calc → your calc model (TI82)
6. then 'OK'

Try to send some program from your TI (in link mode), TILP should automatically ask you to save the file somewhere, which means linking finally works ;)

Thanks to jankenpopp for coming up with the solution to this problem.

## D.4 TI-Connect won't let me transfer files to my TI82.

**Are you using a SilverLink USB cable?** Sorry, that won't work with TI-Connect and TI82

**Are you using a serial or parallel cable?** TI-Connect has problems with CrASH's file naming conventions. One work-around is to

1. install an emulator to emulate a TI82, using the same ROM version that your calculator runs
2. install CrASH and HT2 on the emulator
3. dump a TI82 backup file (.82b) from the emulator

You can now upload the .82b file you created to your real TI82, using TI-Connect.

## **D.5 I cleared the RAM on my TI-8x Plus, but there is still not enough space to install HT2.**

Try deleting some of your APPS. While APPS themselves are stored in Flash memory, some information on them is stored in RAM, taking up space needed by HT2.

## **D.6 I tried everything, but it just won't work. Help!**

Head over to the 1-bit Forum and ask there. We're a friendly bunch.