NTRQ: NES Tracker

Manual Changes

Date/NTRQ Version	Changes
1/7/2010 NTRQ V1.6	- Modified "**Song Editing**" controls table to remove "To Clear A Value" function and replace with "To Clone A Pattern. - Added "**Pattern Cloning**" section. - Updated "**Clear Data Menu**" with **Pattern Purge** feature. - Added small section listing the new NTRQ command-line tools **- Major updates to "DPCM" section detailing Looped Playback and changes to "Pitch Table" and "Duty Table" Pattern commands as used by DPCM channel.**
17/2/2011 NTRQ V1.8	- Added back clearing a value in a Song by tapping B but only for the Master Track

Overview - What Is NTRQ?

NTRQ is a native NES “tracker” enabling you to write music for the NES, on a NES!

As you’d expect, compared to a computer-based music tracker, there are some limitations but I think you’ll find it comfortable and quick to use once you learn the controls.

Requirements

To use NTRQ on NES hardware you’ll need:

- ✓ a NES with at least one controller
- ✓ a “PowerPak” cartridge (or similar) that supports MMC1 and SRAM
- ✓ a TV
- ✓ a NES controller
- ✓ a computer with which to dump the NTRQ ROM-image to the PowerPak (or similar)

You can, of course, just use the ROM-image with your favourite NES emulator. That kind-of defeats the object though :)

NTRQ Files

Storage, Loading & Saving

NTRQ utilises the whole of an 8kb SRAM (battery file, save file) as it's storage medium.

Supplied with the NTRQ ROM is a blank save file, "ntrq-blank.sav" but you don't actually need it. The first time you run NTRQ it will check for the presence of a signature in the save file and if not found, the save file will be "formatted" as an empty NTRQ file. The blank file is a fall-back in case for some reason your .sav file becomes corrupted and NTRQ cannot start. Remember that the save file and the ROM-image name must be the same i.e. "ntrq.sav", though PowerPak does let you specify a save file to load when launching a ROM.

WARNING : When using a PowerPak cartridge, make sure you know how to load and save ".sav" files before spending hours working in NTRQ. PowerPak does not "automatically" save your .sav file, you need to hold RESET on the NES for a few seconds to make PowerPak display the saving menu. Please refer to the PowerPak manual for exact instructions.

Working With Multiple NTRQ Files

Because you can only work on one NTRQ file at a time, it's a good idea to get into the practice of backing up your .sav file regularly. Just copy it to a new location and name it something relevant, like "NTRQ-AwesomeSong.sav". When you need to work on the backed-up version you'll need to copy back to the proper place and rename it "ntrq.sav". As mentioned above, this is only the case with emulators - PowerPak lets you use different .sav names and allows you to specify the .sav file to use on booting the ROM.

Can I Edit My Song With An Emulator And PowerPak?

Yes - the NTRQ files can be swapped between emulator and PowerPak at any time. Please be aware that some NES emulators (Nestopia on OSX is one) compress .sav files. See **Advanced Topic - More About NTRQ Files** at the end of this manual.

Note Regarding PowerPak and .SAV Files

It appears that you must at least have put a blank 8KB save file in your SAV folder on your CF card before PowerPak can save your data. These blank .sav files are supplied with PowerPak,

Data Limitations

Because NTRQ uses a fixed storage space, there are some imposed limitations on the type and number of “objects” that you can use in a single NTRQ file.

Object/Data	Maximum Number
Songs	8
Steps Per Song	128 steps are shared between all 8 songs and each song has to contain at least one step which makes the absolute maximum song length of $78 steps (if all other 7 songs are one step). However, there are song commands to repeat steps/sections of your song and also to loop them indefinitely.
Tracks Per Song	Always 5, corresponding to hardware voices
Patterns	Number and length of patterns is global for a single NTRQ file. You cannot specify the number of available Patterns directly but you can set the global Pattern Length which will increase or decrease the number of available Patterns (the smaller the number of Pattern steps the higher the number of available Patterns and vice versa).
Pattern Length	This is a global setting and can be changed at any time within NTRQ. The minimum Pattern Length (in “steps) is $10 (16) steps and the maximum Pattern Length (in “steps”) is $40 (64)
Instruments	32 Instruments shared by all songs
Pitch Table	64 entries for general Pitch Table use (2 bytes per entry) 32 entries for Pitch Effects such as Vibrato, Detune etc. (2 bytes per entry)
Duty Table	32 entries (2 bytes per entry)
Speed Table	20 entries (2 bytes per entry)

Audio Specifications

Tracks

NTRQ uses 5 tracks that correspond to the NES's hardware voices. As you'd expect:

- Track A = Square 1
- Track B = Square 2
- Track C = Triangle
- Track D = Noise
- Track E = DPCM

Audio Features

Feature	Tracks	Comment
Notes	A, B, C, D* & E*	*Notes on these tracks are represented by numbers instead of note names.
Tie Notes	A, B, C & D*	*Notes on these tracks are represented by numbers instead of note names.
Key Off	A, B, C, D & E	Key Off is used to stop a note playing.
Velocity (individual note volume)	A, B & D	Only these voices have amplitude control
Instruments	A, B, C & D	Instruments used on first 4 tracks though some features not available to all.
ADSR	A, B & D	Only these voices have amplitude control
Gate Time	A, B, C & D	Gate time is used to control Release time of ADSR. On Track C (triangle) it's used to cut off the sound.
Pitch Table (Arpeggios etc)	A, B, C & D	
Duty Table	A & B	Only these voice have duty setting.

Feature	Tracks	Comment
Vibrato	A, B & C	
Pitch Slide	A, B & C	Or “Portamento”. Smooth slide from one note to another.
Hardware Pitch Sweep	A & B	Hardware effect only on the two sqaure-wave voices
Software Pitch Sweep Effect	A, B, C & D	Simulated version of the NES’s hardware pitch sweep but for all voices
Detune	A, B & C	
Auto Echo	A & B	“Automatic” single-voice echo effect for the two square-wave voices
Delay	A, B, C, D & E	Small delay to retard a note on/note off command on a step

This is just intended as an overview. For specific details see relevant sections of the manual.

DPCM

NTRQ supports DPCM samples (Track E). It comes with a few drums built in (see DPCM section for list). ~~There will eventually be the ability to rebuild NTRQ with your own samples in a later release.~~ There is a stand-alone tool that enables you to “patch” the NTRQ ROM with your own DPCM samples. See section “**Command-Line Tools**”.

You can use both one-shot and looped samples. Refer to the DPCM section for details.

Tempo/Speed

NTRQ tempo is controlled by the Speed Table, the starting index which is set on a per-song basis and can also be controlled by a Speed Table Command. Refer to **Song Master Track** and **Pattern Editing : Commands** section.

Song Control

NTRQ has a “Master Track” for each song. The Master Track has it’s own set of commands that affect song playback. These are;

- jump to Song Step
- repeat Song Step (variable number of repeats)
- repeat Song Section (variable number of repeats)
- set Master Volume
- perform Master Volume fade (in/out)
- set index into Speed Table

You can also “mute” the Master Track as sometimes it’s handy to listen to your Song arrangement without repeat loops etc.

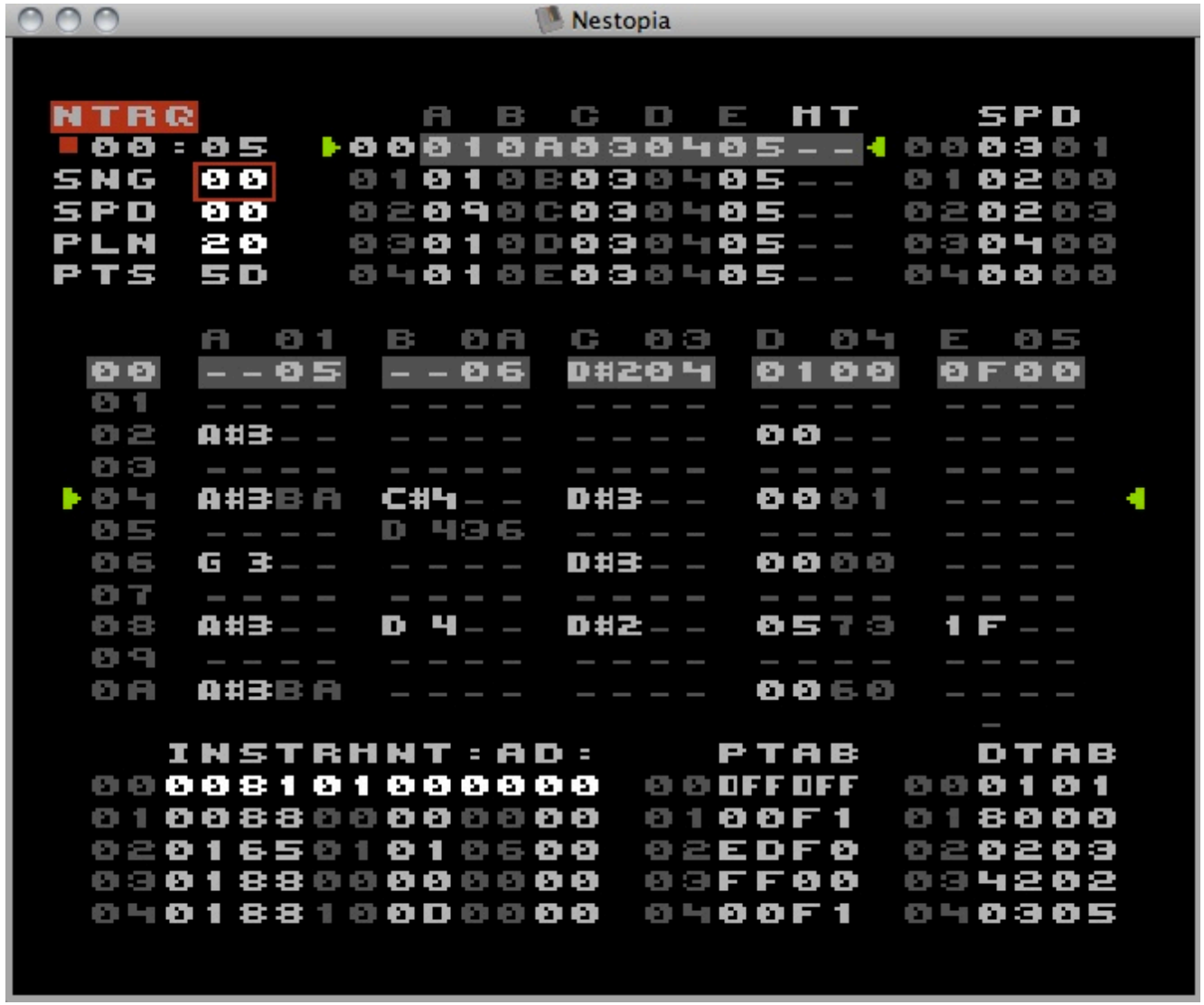
Non-destructive Transpose

Each step of an NTRQ song has a transpose value *per-Pattern* that gets added to the pitches of the notes on that step. You can, therefore, use a single pattern with different transpose values on different song steps.

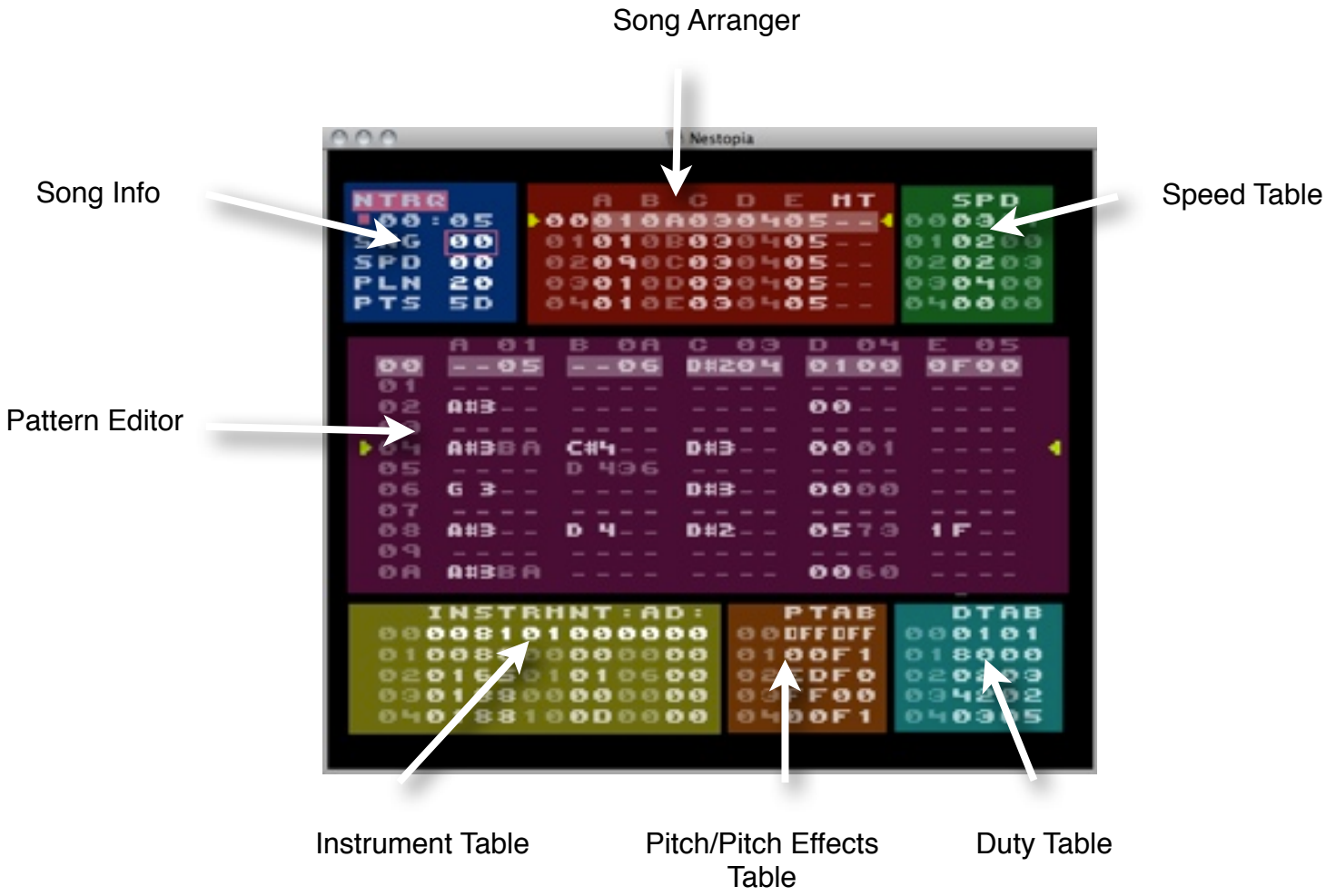
NTRQ’s Approach To Songs And Patterns

Many trackers take the approach that a Pattern contains notes and effects for all (five, in NTRQ’s case) Tracks/Voices. Therefore a Song would just be a linear chain of Pattern numbers. NTRQ’s Patterns contain only notes and effects for one Track/Voice so for each Song Step you can specify a different Pattern for each Track/Voice. NTRQ’s Songs are, therefore, a two-dimensional grid: Pattern Numbers go across, Song Steps go downwards. As the Song is played, NTRQ plays the 5 Patterns that are specified on the current Song Step before moving to the next one, and so on.

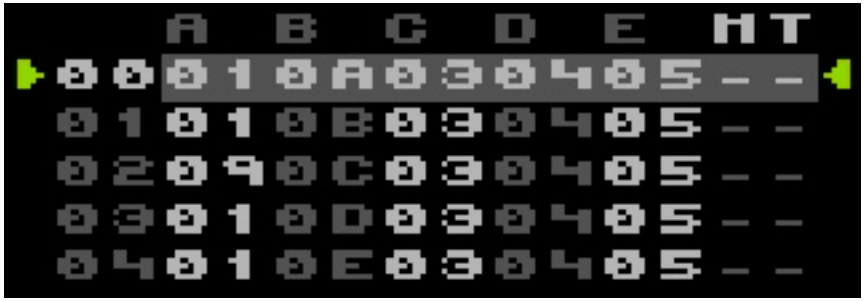
NTRQ Screen


Nestopia
NTRQ
00:05
SNG 00
SPD 00
PLN 20
PTS 5D
A B C D E MT SPD
00 01 0A 03 04 05 -- 000301
01 01 0B 03 04 05 -- 010200
02 09 0C 03 04 05 -- 020203
03 01 0D 03 04 05 -- 030400
04 01 0E 03 04 05 -- 040000
A 01 B 0A C 03 D 04 E 05
00 --05 --06 D#204 0100 0F00
01 ----- ----- ----- ----- -----
02 A#3-- ----- ----- 00-- -----
03 ----- ----- ----- ----- -----
04 A#3BA C#4-- D#3-- 0001 -----
05 ----- D 436 ----- ----- -----
06 G 3-- ----- D#3-- 0000 -----
07 ----- ----- ----- ----- -----
08 A#3-- D 4-- D#2-- 0573 1F--
09 ----- ----- ----- ----- -----
0A A#3BA ----- ----- 0060 -----
INSTRHNT:AD: PTAB DTAB
00 00081010000000 00DFFDFF 000101
01 00880000000000 0100F1 018000
02 01650101060000 02EDF0 020203
03 01880000000000 03FF00 034202
04 01881000D0000 0400F1 040305

NTRQ “Windows”



Overview - Song Arranger Window



The top line shows you the Track Header, A = Track A etc. If a track is inactive the header for that track will be dark grey. If the track is active it will be light grey. If the track is soloed it will be red. To learn about Soloing and Muting refer to the section **Common Controls.** The 6th column, labelled “MT” is the Master Track for the Song. You can “mute” the Master Track too but not “solo” it. When the Master Track is “muted” you will see “OFF” displayed instead of “MT”. To learn more about the Master Track, refer to the section **Song Master Track.**

The little green triangles show the currently playing Song Step. The arrows move down through the song as it plays. Currently the on-screen text does not follow the playback position.

The horizontally highlighted line is the Song Step you are currently editing.

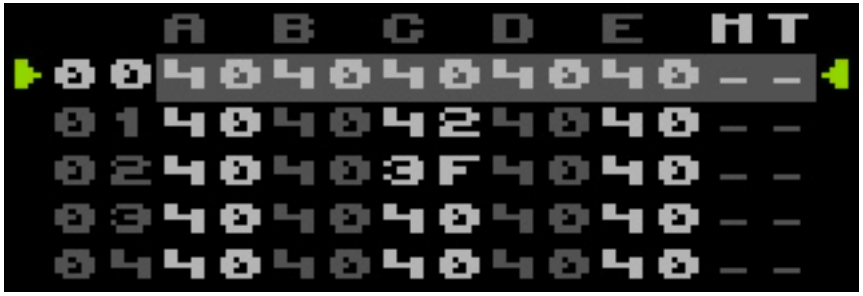
Moving downwards, the Song itself is organised as a matrix of Pattern Numbers. Step 00 (as indicated by the white “00” in the left-hand column) is the first step in a Song. As there are only 5 vertical rows displayed, steps 05 onwards are not displayed here but can be shown by scrolling the cursor downwards.

Reading across one line in the Song, you’ll see that the grey colour alternates between dark and light. This is just for readability and has no significance. In the “MT” column in the picture above you can see two dashes, “--”. This just means there’s no Master Track Command for that Song Step.

Transpose Values For Patterns

The transpose value for each Pattern is actually hidden “behind” the Pattern Numbers. To find out how this works see the section **Song Editing** but I’ll explain the display here briefly.

Song Arranger showing Transpose Values



For the Transpose Value, a value of $40 is 0 transpose. $4C = transpose +12 semitones and $34 = transpose -12 semitones etc. There is a maximum transpose of 5 octaves (plus or minus) which is 60 semitones. There is no range checking on the resulting pitch of the notes in the Pattern so it’s possible to transpose notes outside of the practical range of the NES.
Something else to note: you can see that the number “00” (i.e. Song Step 00) is highlighted in white. This shows you the Song Step that you are currently editing. This is important in the Pattern Editor as you’ll see later.

Find out how to actually edit Songs in the section **Song Editing**.

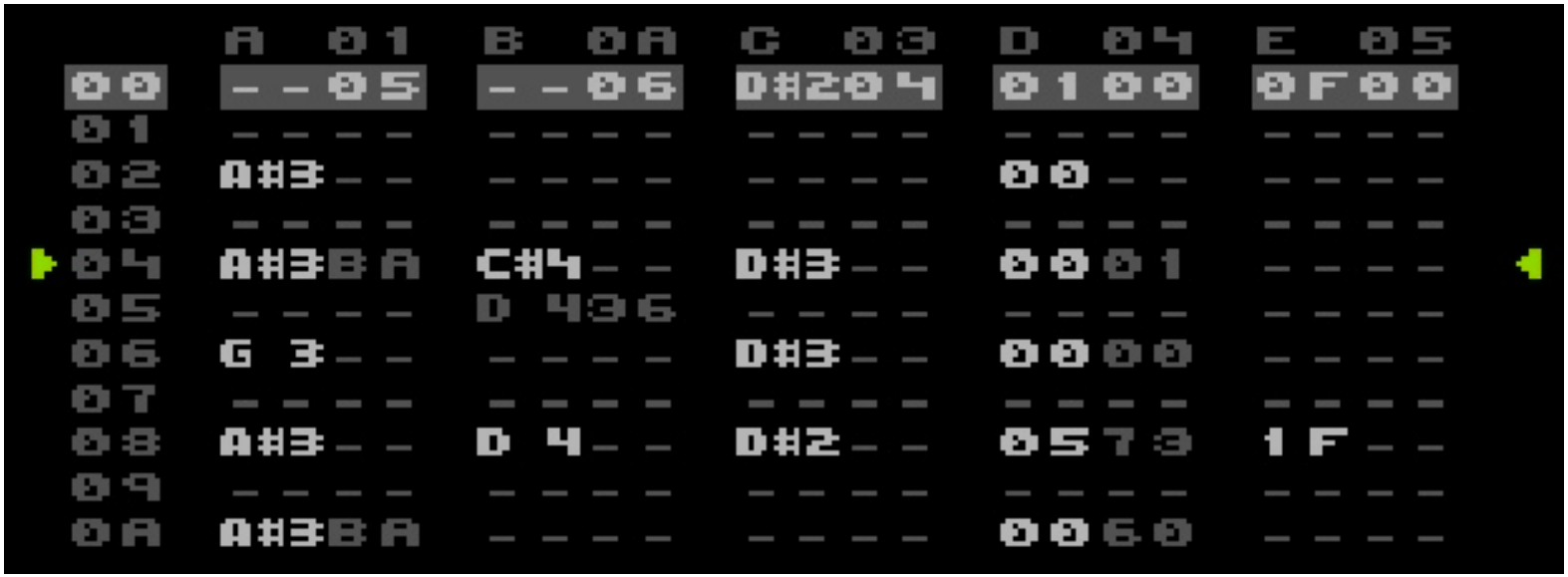
Display of “Empty” Song Steps

When you select a Song that has less than 5 steps, you’ll see that the bottom of the Song Arranger Window will show

-- .. ..

for each “empty” line. This is to show that an empty Song Step doesn’t actually exist, unlike an empty cell in the Pattern Window. Once your Song has more than 4 steps you won’t see this type of line displayed.

Overview - Pattern Editor Window



This is the Pattern Editor Window.

Across the top are the Pattern Headers. Like the Track Headers, if a track is “muted” the Pattern Header (e.g. “A 06”) will be shown in dark grey, “active” in light grey and if “soloed” it will be coloured red. Also in the Pattern Header you can see the current Patterns for each Track in the current Song Step. If you could see the Song Editor in the picture above it would look something like:

Which as you can see by the highlighted numbers, shows the same Pattern Numbers as the Pattern Header.

The left-hand column shows the Pattern Step Number. Similar to the Song Editor, only 11 Pattern Steps are shown at any one time. You can see the rest of the Pattern Steps by scrolling down (or up) the Pattern. Like the Song Editor, the little green triangles show the current play position. And, like the Song Editor, this window doesn't currently automatically scroll with the play position.

Each of the 5 Pattern columns are similar. Looking under the first column (showing Pattern 01 on Track A - "A 1") you can see two further columns. The left-hand column is the note column, the right-hand one is the command column. Generally, data in the note column is coloured light grey to distinguish it from the command column. The exceptions to this are "Tie Notes" and "Key Off" commands which are always coloured dark grey. Values in the command column are always dark grey. Anywhere you see "--" signifies an empty "cell" in the Pattern i.e. you can put "stuff" there.

The other thing you will see in the Pattern Editor Window are "Pattern Markers". The Pattern Markers are used to highlight parts of a Pattern on which to perform various functions (Transpose, Copy, Merge etc.).

Here you can see that part (or maybe all) of Pattern 01 is marked. The arrows point at the marked data > like this <. It's possible to mark just the note column or just the command column (or both, as show here). Read about that in the **Pattern Editing** section.

Overview - Sound/Instrument Editor Window

The last windows are where all (most) of the sound editing is done. It's split into 3 sections: Instrument Table, Pitch Table and Duty Table. All three tables show the index in the left-hand column. The data for each step makes up the remaining numbers in each row.

Instrument Table

Each Instrument has 6 parameters - you can see there are 6 bytes (12 digits) for each row. The currently selected (for editing) row is highlighted in white. You'll realise why when you read about **Instrument Editing** later. That's also where you'll also learn what the numbers are for.

Pitch Table

There are two quirks about the Pitch Table. Firstly, row 00 can not be edited as it has a special use within the NTRQ Player. Here you can see it says "OFF" (you set a Pitch Table Index of 00 in your instrument to turn the Pitch Table "off" for that sound).

Secondly, the table is split into two (actually, more like 2/3rd and 1/3rd) as it's also used for **Pitch Effects** such as **Vibrato** and **Pitch Sweep** (not "Slide", which is a musical effect as opposed to a "sound" effect). If you were to scroll down to row number 3F, you'll see that the colour of the row number and the row data changes at row 40. This is because that's where the "traditional" Pitch Table ends and the Pitch Effect entries for Vibrato etc. starts. It's really important that you understand this about the Pitch Table. You can read more about this in the **Pitch Table** section.

Duty Table

Fairly straight-forward, two bytes per-row. To find out more see the **Duty Table** section.

Overview - Song Info/Setup Window

First of all the first line (under “NTRQ”) shows a play counter using the format SS:PP where SS=the current Song Step and PP=the current Pattern Strep. A little red square means the song is stopped, a green triangle means it’s playing.

The second line, SNG, is the current Song Number. You can change to a new song by editing this number,

The third line, SPD, is the initial Speed Table index for the current song. You can set a new starting speed by editing this number. Note that changes to the initial Speed Table index will only take effect when you restart the song.

The fourth line, PLN, is the global Pattern Length. “Global” is within the scope of a single NTRQ file. You can change the Pattern Length by editing this number. See section **Advanced Topic - Changing The Pattern Length** as this has some serious consequences.

The fifth line, PTS, is the number of Patterns available. You cannot edit this number directly but it will change dynamically as you change the PLN value.

From this Window you can also enter the “hidden” sub-menus: **Clear NTRQ Data, AD-R Table Editor and Options Menu.** More about these features in their relevant sections.

Overview - Speed Table Window

The Speed Table is used to control the tempo/speed of your Songs. The left-hand column is the index number, as with all tables. Each entry in the Speed Table has two values. The first value is the speed value, the second is which Speed Table index to jump to at the next tick.

See the section **Speed Table** for further details.

Overview - “Clear Data” Menu

The “Clear Data” menu is one of two “hidden” menus in NTRQ. This is where you can initialise some/all of the data within the current NTRQ save file. See **Advanced Topic - Clear Data Menu Controls** for explanation.

Overview - “AD-R Table Editor” Menu

The “AD-R Table Editor” menu is the second “hidden” menu in NTRQ. This is where you can edit the values used by the ADSR in your instruments. For a detailed explanation see **Advanced Topic - Editing The AD-R Table.**

NTRQ General Editing Principles

Warning! You're Always Editing!

WARNING : There is no loading or saving so be aware that at all times you are working on the actual data that is saved to disc/cartridge. I cannot stress this enough and it's definitely good practice to regularly back-up your save files.

OK, probably a bit dramatic but it's something you should always be aware of because there is no undo. Actually, there's is *some* undo which we'll get to later but in general, **if you change something it's changed forever.**

Moving Around The Editors

Most of the time, holding SELECT and then pressing a direction on the D-PAD will take you where you think it will. There are a few exceptions though. Firstly, moving from one of the Sound Editors (Instrument, Pitch, Duty) to another editor will remember which of the Sound Editors you were last in. So when you move back to the Sound Editors you will return to the remembered one.

Secondly, there are a few *context sensitive* movements that can be made e.g. if your cursor is in the Pattern Editor on a Pitch Table Command, pressing SELECT + DOWN will jump to the relevant entry in the Pitch Table. See **Context Navigation Shortcuts**.

Adding and Deleting Values

Almost 100% of the time, **A** will add/set a value and **B** will delete/clear one at the current editing point. The exception to this is the dual-use for button **B** - see **Pattern Editing: Notes** and **Pattern Editing: Commands** for an explanation.

Modifier Keys

There are several key combinations to access particular editing features. Most of the time you'll know when you've held the key long enough because something graphical will happen - either the cursor will change or an on-screen text hint will appear. The thing to remember is to take the time to hold the modifier keys for long enough. You'll get the hang of it quickly enough. The problem is, not 100% of these combinations are accounted for in on-screen hints so take care. It's on my list-of-things-that-need-attention.

NTRQ General Editing Principles (cont)

Tapping Vs Holding

The **A** and **B** buttons have dual use depending on whether you quickly press-and-release (tap) them or you hold them for a short while.

Immediate Changes Vs Set-n-Release

Certain values are changed (in the physical data) as soon as you modify them. However, most values can be changed as much as you like (while holding the key to modify a value) and the new value will only be committed to the physical data when you let go of the key. This of course only works when modifying existing numbers. Value entered using the Tap method are entered immediately.

Auto-Repeating Keys

Most areas that you'd want auto-repeating keys have them. It's something else on my list-of-things-that-need-attention.

Auto-Repeating Keys And Tables

The Pattern Editor Window, Instrument Table, Pitch Table and Duty Table all wrap around when your cursor reaches the top or bottom.

If you're holding UP or DOWN and it's auto-repeating, when the cursor hits the top or bottom of these tables, the cursor will pause for a short while before wrapping to the top/bottom row (depending on direction). When the cursor stops, if you let go of UP/DOWN the cursor will remain on the top/bottom row.

There is one (very deliberate) quirk with this when moving around the Pitch Table. As well as pausing at the extremities of the table it also pauses at row $40 (as this is the start of Vibrato/Pitch Effects section of the Pitch Table). You can read more about this in the **Pitch Table** section.

Hint System

In order to help remember various command values, as you enter/modify them, a small text hint will appear just below the Pattern Editor.

The hints work for : **Pattern Commands, Song Master Track Command, Pitch Effect Commands**

Common Controls

Certain functions work all the time, regardless of which editing window your are currently in.

To Mute / Un-mute A Track	**Hold SELECT then press A**
To Solo / Un-solo A Track	**Hold SELECT then press B**

These work in every editing window except the Song Info/Setup one. When in the Instrument, Pitch or Duty Table. The affected Track will be the last Track that the cursor was in before you jumped to these tables.

To Play/Stop Song From Start	**Press Start**
To Set Playback Start Marker And Play From There	**Move to required Song Step, Hold B+A then press START**
To Play/Stop Song From Start Marker	**Hold B then press START**
To Play/Stop Current Song Step As Loop	**Move to required Song Step, Hold SELECT then press START**

For the play modes that work based on the current Song Step, the “current” Song Step is always highlighted in the Song Arranger Window so even if the cursor is not currently in the Song Arranger you can still use these modes.

Note: the initial Speed Table Index (set by the SPD parameter in the Song Info Window) is only used when playing the Song normally from the start (pressing just **START**). Any other time the Speed Table Index will remain at whatever value it was when the Song was stopped.

Song Editing

To Move Around The Song	**U/D/L/R**
To Modify An Existing Value: *Pattern Numbers and Master Track Commands*	**Hold A then:** **U/D to increase/decrease value by $01** **L/R to increase/decrease value by $10**
To Modify An Existing Value: *Transpose Value For A Pattern*	**Hold B, song display changes to show Transpose values then:** **U/D to increase/decrease the transpose value by $01** **L/R to decrease/increase the transpose value by $0C (octave)**
To Clear A Value In the Master Track	**Tap B** This only works for values in the Master Track. Tapping B in any other Track will clone the Pattern under the cursor.
To Clone A Pattern	**Tap B** If successful, the Pattern number under cursor will change If there are no available Patterns, nothing will happen. See section **Pattern Cloning**
To Set Value to Same As Last Modified Value	**Tap A** This doesn't work for editing Transpose values.
To Add A Row To The Song	**Hold A + B then press DOWN** A new "blank" line will be inserted above the cursor unless the cursor is on the last line of a Song in which case the current line will be duplicated below itself.
To Remove A Row From The Song	**Hold A + B then press UP** The current line will be removed and subsequent lines moved up.

To Mark A Section Of A Song To Copy	**Hold A + B then press SELECT** You need to do this twice: first at the start of the block you want to mark and then at the end of the block you want to mark. When successful, the lines you marked will appear coloured RED.
To Paste A Block Of Song Data Over Existing Lines	**Hold A + B then press RIGHT** The marked section of song will be pasted over the song starting at the line the cursor is on. If there is not enough steps in the Song over which to paste the marked lines you will get an error message.
To Insert A Marked Block Of Song Data Into A Song	**Hold A + B then Press LEFT** The marked section of Song is inserted into Song at the current cursor position. If there are not enough total free Song Steps (128 shared between all Songs) to insert the marked lines you will get an error message.
To Clear The Marked Area	**Hold A + B then press SELECT** This doesn't clear any song data, just "unmarks" a previously marked block of a song. See below for note.*1

Other Controls

There are some common key functions that work in any Editor. See **Common Controls** for details.

*1 Technically, when you do this, NTRQ is then expecting you to repeat the button combo to mark the end of the block but it is a quick and cheap way to get rid of the red Song rows.

Pattern Cloning

Pattern Cloning was added in version XXX. It was necessary to make a small but significant modification to NTRQ data because of this feature. Any of your existing NTRQ .sav files will be automatically converted when you run the latest version (and re-save your .sav file, however you do that on your platform of choice). This also affects the NTRQ2NSF tool. A new version is included with the latest release. To convert any existing NTRQ .sav files (from old NTRQ versions) to NSF you must first "open" and "save" them in the latest version of NTRQ so that they get "upgraded". Otherwise the NSF will come out as (playable) garbage.

Pattern Cloning

Any LSDJ users will already be familiar with this feature as it work in much the same was in NTRQ.

Pattern Cloning works in the Song Arranger only. It's used to clone the Pattern under the cursor. If successful, the Pattern number under the cursor will change to a new Pattern number, the contents of which will be a copy of the old Pattern. This only works if you have unused (i.e. not referenced by any of the songs in the current .sav file) empty Patterns free. If you have no unused empty Patterns, pressing the clone button (**TAP B,** see **Song Editing**) will have no effect.

Pattern Purging

If the clone function has stopped working it's because you have no unused empty Patterns left in your .sav file. It might be the case that you do actually have unused Patterns but they contain data. If you're sure you won't need these unused Patterns, it will be possible to "reclaim" them by using the Pattern Purge function in the **Clear Data Menu**.

Please use with care! NTRQ will scan the songs in your current .sav file. Any Patterns that are not used will be wiped. There is no undo.

Song Master Track

As described earlier, the 6th column in the Song Arranger Window is the “Master Track” (MT) for the Song. The Master Track has it’s own set of commands with which to control various playback functions.

Command	Value	Note
Jump To Song Step	$00 to $7F	At the end of the current step, playback will continue at the specified step number.
Repeat Single Song Step	$80 to $9F	Repeat current song step a specified number of times ($80=1, $9F=32 times)
Repeat Song Section	$A0 to $BF	Repeat section of song a specified number of times ($A0=1, $BF=32). The first occurrence of this command sets the start of the repeating section and also sets the count. Place the command again on the last step of the section you wish to repeat. Any value between $A0 and $BF is valid as the count is set by the first command.
Set Master Volume	$C0 to $CF	Sets the master volume for the Song
Set Master Volume Fade	$D0 to $DF	Causes the master volume of the song to be faded out (or faded in if the master volume is $00). The speed is $0 to $F, fast to slow.
Set Speed Table Index	$E0 to $F3	Sets the Speed Table index for the Song.

Note: when copying sections of a Song, all Master Track commands are copied except Jump Commands. This is deliberate to maintain the structure of your song. When inserting/deleting/pasting steps/sections of a Song, any Jump Commands are “transposed” so that (*hopefully!)* the Jump Command still jumps to the same Song Step as it did before, *not the same Song Step Number.*

“Muting” The Master Track

When the Master Track is “muted” all Master Track Commands are ignored during playback.

Pattern Editing

To Move Around The Patterns/Tracks	**U/D/L/R**
To Modify An Existing Value	**Hold A then:** **U/D to increase/decrease value by 1** **L/R to increase/decrease value by $10 (context dependent)** e.g. If value is a note, L/R decreases/increases octave If editing a “note” in a Pattern for Track D or E, pressing L/R will add/subtract $10 from the value as opposed to Tracks A,B & C where it will change the octave (+/- $0C semitones)
To Clear A Value	**Tap B on cell with data.** The value is deleted but temporarily saved, see next command. Cursor will move down one row.
To Paste Last Deleted Value	**Tap B on empty cell.** Cursor will move down one row.
To Set Value to Same As Last Modified Value	**Tap A** The actual value is context dependent e.g. if in note column, last note will be set.
To Insert A Row In The Pattern	**Hold A + B then press DOWN** A new blank line will be inserted at the current step and subsequent lines moved down one step. The length of the Pattern will not change. The last step of the Pattern will be pushed off the end and lost unless your cursor is on the first step. If the cursor is on the first step the Pattern will be rotated one step downwards.

To Delete A Row From The Pattern	**Hold A + B then press UP** The current step will be deleted and subsequent lines moved up one step. The length of the Pattern will not change. The last step of the Pattern will be cleared unless your cursor is on the first step. If the cursor is on the first step the Pattern will be rotated one step upwards.
To Enter A "Key Off" Command	**Hold B then press UP** This only works in the Note column.
To Convert A Note To/From a "Tie Note"	**Hold B then press DOWN** If cursor is on existing note, note will toggle between normal and tie. If cursor in on empty cell, a new tie note will be entered, the pitch will be the same as the last note entered.
To Change the Pattern Assigned To Current Track On Current Step Of Song	**Hold SELECT then press L/R** Changes will only affect the currently selected step in the Song Window (indicated by white highlighting of the Song Step in Song Window).
To Mark A Section Of A Pattern	**Put Cursor at start of area to mark** **Hold A + B then press SELECT** then **Put Cursor at end of area to mark** **Hold A + B then press SELECT** You can mark just the Note column, just the Command column or straddle both columns. If you try to set the mark across different tracks, the start mark will be set again on the second selected track.
To Clear The Pattern Mark	**Hold A + B then press SELECT, twice on same cell.**

To Paste The Marked Pattern Area	**Put Cursor on destination pattern at destination step.** **Hold A + B then press RIGHT** The data in the marked area/pattern will be pasted into the destination pattern. The contents of the destination cells will be replaced. If source data exceeds destination pattern it will be clipped. If you have only marked one column in the source Pattern, only that kind of data will be copied to the destination. If you have marked across both columns of the source Pattern, both columns will be copied to the destination. Source and Destination can be same Pattern.
To Merge The Marked Pattern Area With Another Pattern (or same Pattern)	**Put Cursor on destination pattern at destination step.** **Hold A + B then press LEFT** Works in the same way as Paste (above) but: a) any empty cells in the source Pattern will be ignored b) only empty cells in the destination Pattern will have data written to them Source and Destination can be same Pattern.
To Modify A Marked Area Of A Pattern	**With the area marked (see above), Hold B then press L/R** Any data within the marked area will be incremented/decremented. There is no error/range checking so use with care/thought.

Pattern Markers - Strange Behaviour?

Because of the way the Pattern Marking works you sometimes might see strange things.

The marking actually tags the data within the Pattern as opposed to marking what is displayed on screen. This means that you can have a marked Pattern that is not visible. Because the only way to unmark a pattern is to do it manually (**To Clear The Pattern Mark** above), you should get into the habit of doing it just in case you accidentally hit one of the button combos associated with marked Patterns.

The other *issue* is when you have the same Pattern assigned to more than one Track on a single Song Step. In the Pattern Editor Window, if the same Pattern is assigned to two or more tracks, what you are seeing is multiple views of the same data. Thus, because as the marking is done on the actual Pattern data, you'd end up seeing multiple marked areas and that would be confusing (and slower for

the code to draw the Pattern Window!). So instead, the Pattern Mark is displayed only on the left-most Pattern. For example. If you had marked an area of Pattern 02 and the current Song Step had Patterns 02 02 02 03 04, the mark would appear only in the first Pattern (Track A). Similarly, if the Song Step had 04 03 02 01 02, the mark would appear in the third Pattern (Track C).

And possibly the strangest behaviour (but logical!) is if you have the same Pattern on multiple Tracks in a single Song Step when you actual perform the Pattern marking. Taking the example above, if the current Song Step was 02 02 02 03 04 and your cursor was in Track C, if you then mark an area of the Pattern in Track C, the marked area will “jump” to Track A. This is because I’m displaying the mark on the left-most Track, as described above.

Easiest thing to do is try it and see. Set all the tracks in a Song Step to the same Pattern Number and try marking an area in Track E. It’s not perfect but it works!

Hearing Notes As You Enter Them

A recently added feature means that you can hear notes as you enter them into a Pattern. This works on all voices. To determine the sound character of the note, starting at the step your cursor is on, NTRQ searches backwards through the current Pattern for a Select Instrument command. The sound is output for as long as you hold down button A (but obeys the “Gate” setting in the instrument) and will be “keyed off” when you release button A. If it should ever happen that you get notes “stuck on” then playing another note on that Track or Starting/Stopping the Song should clear it.

Please note that the actual note you’ll hear is affected by the Transpose value of the Pattern you are currently editing.

Turning Audible Note Entry On/Off

If you prefer not to hear notes as you enter them, you can turn the feature on/off in the **Options Menu.**

Pattern Editing: Notes

As we've already covered, in the Pattern Editor Window, under each Track Header is two columns of values (or "--" if you're starting from scratch!). The left-hand column is where the notes go, the right-hand column is where the commands go.

Notes: Track A, B & C

When adding/editing values in the note columns of Track A, B & C, actual note names + octave numbers are displayed. Normal Notes are what you'll generally enter and these are coloured light grey. The valid range for notes is A1 to B8 and this is range checked as you edit so you can't go outside of these values.

Note: notes on Track C are deliberately displayed an octave below A & B because that's how they sound.

Tie Notes: Track A, B, C

Tie Notes are like Normal Notes except they don't initialise stuff like ADSR, vibrato, pitch table index, duty table index etc. You enter (or convert an existing normal note to) a Tie Note by holding down **B** then pressing **DOWN** as described in the controls section. You can modify the note + octave of a Tie Note just like a normal note. You can convert a Tie Note to a Normal Note by repeating the same button combo.

Key Off: Track A, B, C, D & E

You can also enter a Key Off command in the note column. Depending on what Track the command is on it will either force the ADSR into the Release Phase (Track A,B & D) or just kill the note (Track C & E).

Notes: Track D

Notes on Track D are just numbers. Normal range is $00 to $0F (32-kbit mode) but if you enter $10 to $1F you'll play a note in the second random generator mode (93-bit mode). You can also use Tie Notes on Track D.

Notes: Track E (DPCM)

See **DPCM** section for explanation.

Pattern Editing: Commands

The physical editing/entering of Pattern Command is just the same as with notes. However, Pattern Commands are just hex numbers so you have to know what the numbers represent. There is no error/range checking so pay attention to the value ranges described for each Pattern Command.

You can obviously set a Pattern Command on the same step as a note. In all cases apart from Note Slide, the command is processed first (you'll see why when you read "Note Slide") but in reality both are processed in the same refresh frame.

Set Instrument : $00 to $1F (Tracks A, B, C & D)

> This selects the Instrument $00 to $1F. All subsequent notes will play with the selected Instrument. This works the same for all Tracks except Track E. See **DPCM** section for details.

Note Slide : $20 to $3F (Tracks A, B & C)

> This performs a smooth slide to the note specified on the same Pattern Step. The speed of the slide depends on the value. **$20** actually stops the slide, **$21** is the slowest speed while **$3F** is the fastest.
>
> Using a Note Slide without specifying a Note on the same step has no effect.
>
> *Tip : you can use a tie note for a properly smooth slide.*

Set Pitch Effect : $40 to $5F (Tracks A, B, C & D)

> This selects a Pitch Effect for the Track. The Command value corresponds exactly to an index in the Pitch Effect section of the Pitch Table. You can use the Command at any time in a Pattern, it does not need to appear on the same step as a note (though you can do this if you like). If a note is already playing the effect will be applied to that note. The effect will also apply to all subsequent notes until either another Pitch Effect Command or a Set Instrument Command whereby the Aux parameter from the Instrument will be used to set the Vibrato. **$40** turns off the effect too.

Pitch Table Index : $60 to $9F (Tracks A, B, C & D)

Similar to the Vibrato Command, this selects a Pitch Table index to apply to the current Track. However, unlike the Vibrato Command, the actual index is the Command Value minus $60. So $6F would be index $0F of the Pitch Table and so on. Like Vibrato, you can set this command on the same step as a note or not. If a not is already playing the effect will be applied to the note at that point. The effect continues until a new Pitch Table Command or a Set Instrument Command. **$60** turns off the effect off.

Pattern Note Velocity : $A0 to $AF (Tracks A, B & D)

For Tracks A, B & D you can scale the envelope amplitude by using this command. $AF = maximum volume (don't scale) while $A0 = silence. $A8 is about half-volume etc.

When you use this effect, it's sets the Velocity for all subsequent notes until a new Velocity command is used.

Note Damper : $B0 to $BF (Tracks A, B & D)

This command is similar to Pattern Note Velocity as it scales the envelope amplitude in the same way.

However, unlike the Pattern Velocity Command, the effect is temporary and only lasts for the note on which it was set. This command also works on currently-playing notes - if used on a step that doesn't contain a note it will modify the volume of any note that is currently playing.

Duty Table : $C0 to $DF

This command will override the Duty Table Index for the current instrument, $00 to $1F. The effect will last until you select a new Instrument or another Duty Table command. You can also set this command on a currently playing note and it will change the Duty Table index for the note at that point.

Speed Table : $E0 to $F3

This command will set a new Speed Table index for the Song, $00 to $13.

Note Delay : $F4 to $FD

Experimental feature that when placed on the same step as a note will delay the keying-on of that note for a specified number of frames. As I only have one byte and a couple of bits, $F4 = 0 frames delay, $F5 = 1 frame delay ..... $FD = 9 frames delay. This also works on the “KEY OFF” command in the same way.

Terminate Pattern : $FE

This is a special command that will interrupt a Pattern at the step you place it on. The playback then continues with the next Song Step. All Patterns on the same Song Step will be interrupted at the same point to maintain sync between Tracks.

This is a compromise feature so that you can have Pattern Steps that are shorter than the Global Pattern Length e.g if you want an intro to a song that is only half-a-Pattern long.

Persistence Of Pattern Effects

Pattern Command	Value	Persist?	Reset By?
Instrument	$00-$1F	Yes	Next *Instrument Command*
Pitch Slide	$20-$3F	No	Target note is reached or new *Instrument Command*
Pitch Effects: Vibrato	$40-$5F	No	Next note (unless it's a tie note)
Pitch Effects: Software Sweep	$40-$5F	Yes	*Instrument Command* or *Clear Pitch Effect Command* ($40)
Pitch Effects: Hardware Sweep	$40-$5F	Yes	*Instrument Command* or *Clear Pitch Effect Command* ($40)
Pitch Effects: Detune	$40-$5F	Yes	*Instrument Command* or *Clear Pitch Effect Command* ($40)
Pitch Effects: Auto Echo	$40-$5F	Yes	*Restarting Song or setting new Auto Echo value*
Pitch Table Index	$60-$9F	Yes	*Instrument Command* or new *Pitch Table Index Command*
Pattern Volume	$A0-$AF	Yes	Only reset by a new *Pattern Volume Command*
Note Volume	$B0-$BF	No	Lasts until next note
Duty Table Index	$C0-$DF	Yes	*Instrument Command* or new *Duty Table Index Command*
Speed Table Index	$E0-$F3	Yes	New *Speed Table Command* (or selecting new Song)
Note Delay	$F4-$FD	No	Lasts until next note
Terminate Pattern	$FE	No	Not applicable

Instrument & Sound Editing

The Instrument, Pitch and Duty Tables are all edited in pretty much the same way.

To Move Around	**U/D/L/R**
To Modify An Existing Value	**Hold A then:** **U/D to increase/decrease value by $01** **L/R to increase/decrease value by $10**
To Clear A Value	**Tap B** The “clear” value will depend on parameter being edited
To Set Value to Same As Last Modified Value	**Tap A**

Pitch Table & Duty Table Specific Controls

There are some controls that are only available in **Pitch Table** and **Duty Table.**

To Insert A Row In Table (Pitch & Duty Only)	**Hold A + B then press DOWN** A new blank line will be inserted at the current step and subsequent lines moved down one step. Values at the end of the table will be lost. See **Advanced Information For Pitch & Duty Table** for more information.
To Delete A Row From The Table (Pitch & Duty Only)	**Hold A + B then press UP** The current step will be deleted and subsequent lines moved up one step. A blank line will be added to the end of the table. See **Advanced Information For Pitch & Duty Table** for more information.
To Set The Pitch Table Index For Currently Highlighted Instrument From Within Pitch Table	**Hold A then tap B** Pitch Table Index for Instrument is set to row of table that cursor is on.

To Set The Duty Table Index For Currently Highlighted Instrument From Within Duty Table	**Hold A then tap B** Duty Table Index for Instrument is set to row of table that cursor is on.

Advanced Information For Pitch & Duty Table

Because you can insert/delete rows in the Pitch Table and Duty Table, you might wonder what happens to the parameters that define the starting index for these in the Instrument definitions. They get “intelligently” updated, that’s what! For example, and you probably need to read **Pitch Table** and **Duty Table** sections for this to make total sense, if you’d set up an arpeggio at position 05 in the table with 3 steps that jumps back to step 05, and then you insert a row into the Pitch Table, any instruments that have their Pitch Table parameter set to 05 will be modified to be 06 etc. Also, the jump command in your arpeggio will be updated to jump to the right point. Same principle if you delete a row.

This also works with the “Pitch Table” Pattern Command. If you have any of these commands in your Patterns they will be updated accordingly.

Oh, this also works with the Duty Table too. Good, eh?

Difference In Operation Between Pitch Table and Pitch Effects Table

I mentioned earlier about the Pitch Table actually being split into two different halves. Just a quick note to say that you can only insert/delete lines from the actual Pitch Table bit of the Pitch Table (I know, it’s confusing). It’s not possible or necessary to insert/delete lines in the Pitch Effects (Vibrato etc.) section.

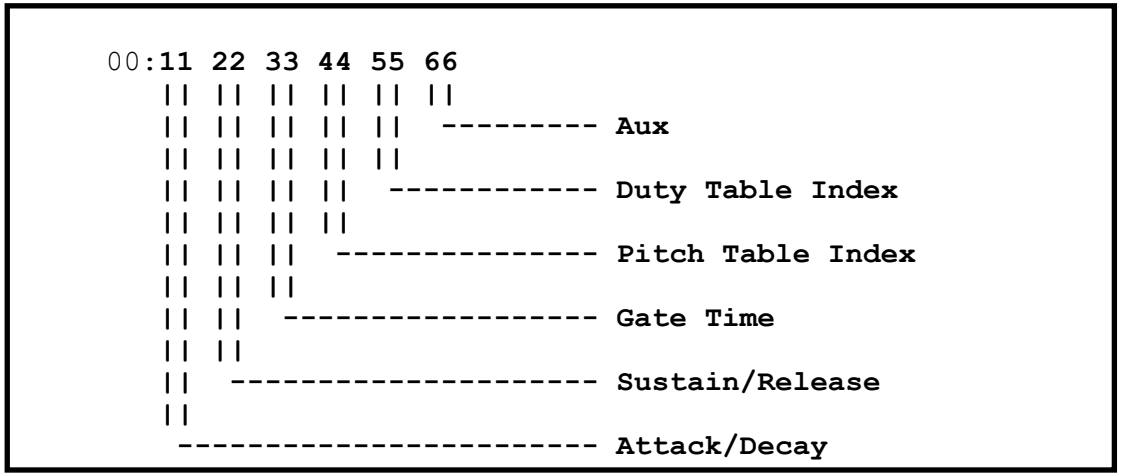
Setting Instrument Pitch and Duty Table Index From Within Pitch & Duty Tables

Just a quick note about these two features (as detailed in the above table).

You might remember I mentioned earlier that the last selected Instrument in the Instrument Table is always highlighted in white. Now you know why - it’s so that when you use these two editing tricks you know which Instrument you are affecting.

Instrument Table

Each Instrument comprises 6 bytes of information. I've spaced them out here for clarity and the numbers are irrelevant. "00:" indicates the position in the table.



An explanation of the parameters follows but I just thought I'd mention the universality of Instruments.

As I'd hope you already know, not all of the NES's hardware voices are made equal. You can see from the Instrument Parameters that some of them don't apply to certain voices. You can, however, use Instruments fairly universally across all voices (or Tracks) and NTRQ will generally ignore the Instrument parameters that are irrelevant to the particular voice.

One other thing to note: there are no Instrument definitions for the DPCM voice (Track E). More on this later.

See **Pattern Editing: Commands** for details on how to select an Instrument in a Pattern.

Instrument Table (cont)

I recently added something to help you navigate around the Instruments. At the side of the Instrument Table name i.e. “INSTRMNT” in NTRQ, you’ll see a semi-colon “:” followed by a two letter code. This changes as you move between the columns of the Instrument.

AD = Attack/Decay, SR = Sustain/Release, GT = Gate Time, PT = Pitch Table Index, DT = Duty Table and AU = Aux

Attack/Decay [AD] & Sustain/Release [SR]

Attack = time taken for envelope amplitude to reach maximum amplitude
Decay = time taken for envelope to go from maximum to the Sustain amplitude
Sustain = amplitude that note is held at
Release = time taken for amplitude to go from Sustain level to zero

Maximum amplitude is always 15 ($F). Envelopes can be scaled using volume commands in Patterns but more on this later.

I’ll deal with both **AD** & **SR** together as both parameters rely on a table of values that is not immediately editable from the Instrument Window.

The two values are split into nibbles (two single hex digits). For the **AD** parameter, the Attack value is the left-hand digit and the Decay value is the right-hand digit. This gives you a range for Attack and Decay of 0 to 15 ($0 to $F) each. This is also true of the **SR** (Sustain/Release) parameters.

Of the four values, Sustain is the easiest to explain as it’s the amplitude ($0 to $F) at which the note is held.

However, Attack/Decay/Release are all indexes into the **AD-R Table** that is normally hidden from view. You can read how to reveal and edit this table in the section **Advanced Topic - Editing The AD-R Table.**

If you want more details on the ADSR envelopes refer to section **ADSR Envelopes.**

Instrument Table (cont)

Gate Time [GT]

Gate Time is the time, in frames, that a note is held for at the ADSR sustain amplitude. If you specify “00” then the note is held indefinitely until either:

- a new note is played
- a Key Off Command is encountered

See **Pattern Editing** for information about notes and Key Off Command.

Pitch Table Index [PT]

This is the index into the Pitch Table for notes played using this Instrument. Each time a new note is played the Pitch Table index is reset to this value (unless you play a Tie Note). If you don’t want a Pitch Table effect on this instrument, set this value to 00. See **Pattern Editing** for information on Notes and Tie Notes.

You can also override the Pitch Table Index for an instrument from within the Pattern with the **Pitch Table Index Command**.

Duty Table Index [DT]

Similar to the [PT] parameter, this defines the starting index into the Duty Table for notes played with this Instrument, unless you play a tie note. Index “00” is used in the Duty Table as you always need to specify a setting, thus “00” logically points to entry 00 in the Duty Table. See **Pattern Editing** for information on Notes and Tie Notes.

Aux Parameter [AU]

This is going to be used for future features/effects. Currently it can be used to select a Pitch Effect from the Pitch Table: *Vibrato*, *Hardware Pitch Sweep*, *Software Pitch Sweep* and *Detune.*

Currently only values $40 to $5F are used. Other values should not be used and are for future expansion.

It's also possible to select a Pitch Effect from within a Pattern using the **Pitch Effect Command.**

If you don't want to explicitly set a Pitch Effect for an Instrument, leave the value set to "00". However, **be aware that many pitch effects are actually persistent after they have been set via a Pattern Command**. So if you want an Instrument that **definitely** has no Pitch Effects, set the Aux value to $40. (Incidentally, this is the same value as the Pattern Command to cancel Pitch Effects).

See **Pitch Effects Table** and also **Persistence Of Pattern Effects** for a thorough explanation. It's a tricky area of NTRQ so make sure to understand it well.

Pitch Table

As I've explained earlier, the Pitch Table has dual use. The first $40 entries ($00-$3F) are used for traditional Pitch Table functions such as Arpeggios. Each entry in the table is made up of two bytes.

```
00:11 22
   || ||
   || ||
   ||  --------- If byte 1 = $F0, byte 2 is delay
   ||            If byte 1 = $FF, byte 2 is loop point
   ||
   ||            Otherwise, byte 2 = $VD, where
   ||            $Vx = amplitude scale, $0-$F
   ||            $xD = delay before next step, $0-$F
   ||
   ||
    ------------ $00-$40 = positive semitone offset
                           $00 to $40
                 $50-$AE = absolute note
                           $00 to $5E
                 $B0-$EF = negative semitone offset
                           -$40 to -$01
                 $F0     = delay, byte 2 delay in frames
                           until next step
                 $FF     = loop-back, byte 2 is index to
                           loop to
```

Pitch Table Examples

Major Triad Arpeggio, no volume scale, fastest speed (0)

```
01: 00 F0       ;0 semitone offset, volume 15, speed 0
02: 04 F0       ;4 semitone offset
03: 07 F0       ;7 semitone offset
04: FF 01       ;loop back to step 01
```

Minor Triad Arpeggio, inverted, no volume scale, speed 1

```
01: 00 F1       ;0 semitone offset, volume 15, speed 1
02: EB F1       ; -5 semitone offset
03: E7 F1       ; -9 semitone offset
04: FF 01       ;loop back to step 01
```

Bass with Kick Sound

```
01: 74 F0       ;absolute note $24
02: 70 F0       ;absolute note $20
03: 00 F0       ;no offset
04: FF 03       ;loop back to step 3
```

Note About Volume Scaling

As you might expect, a volume scale of “F” = no scaling. A value of “8” is about half-volume. For Track C, where you have no amplitude, values “1” to “F” is “on” and “0” is “off”.

Pitch Effects Table

Because of the limitations of trying to cram every Pattern Command possible into one byte (I only have one command column in the Patterns), I had to try to come up with a solution to “extend” this in order to give you access to certain pitch effects. My solution was to split the Pitch Table into two halves.

The first half, as described above, is for traditional Pitch Table uses such as Arpeggio, where each effect is made up of several steps that modify the pitch over time (and can be looped etc.).

The second half of the table is dedicated to “Pitch Effects”. I’d describe Pitch Effects as ways to modify the pitch of a note that are not directly controlled by a table structure in the way that arpeggios are. Kind of set-and-forget. Currently there are five of these effects;

- Vibrato (voices A, B & C)
- Hardware Pitch Sweep (voices A&B only)
- Software Pitch Sweep (voices A, B, C & D)
- Detune (voices A, B & C)
- Auto Echo (voices A & B)

Each entry in the Pitch Effects Table comprises two bytes. The first byte indicates the type of effect and the second byte is a parameter associated with each effect.

As opposed to “normal” Pitch Table selection where you specify the starting index and then the table is stepped through (and looped), for Pitch Table Effects you’re just telling NTRQ where to find parameters to set up the individual effect. So the Pitch Effect Table is not stepped through at all. See the following table for the values used to specify each effect. First an explanation of each effect.

Pitch Effects Descriptions

Vibrato (Voices A, B & C)

Vibrato is the cyclical modulation of the pitch of a note. NTRQ uses a sine-wave modulation type. You can specify a delay before the vibrato starts (number of frames after the start of a new note) and also the depth and speed of the modulation. NTRQ also employs “blargg”’s trick to stop software vibrato from resetting the phase (voice A & B) and causing cyclical “clicking”. (http://nesdev.parodius.com/bbs/viewtopic.php?t=231)

Software Pitch Sweep (Voices A, B, C & D)

Because Hardware Pitch Sweep is limited to voices A & B, I added a software version to mimic the behaviour (loosely) for voices C & D. (you can use the effect on A & B too if you wish). Every frame (50/60hz), the effect adds a semi-tone value to the current pitch. You can set a multiplier value for the semi-tone value and the value can be positive (for upwards sweeps) or negative (for downwards sweeps). You can also set the step speed, the step speed being the number of frames to wait before re-adding the semi-tone offset again. You can also set a delay before the effect is applied in the same way you can with Vibrato.

There are two modes: **Overflow** and **Limit**.

“Overflow” allows the modified note number to exceed the maximum allowed on the NES. If sweeping upwards, once the highest note is reached the note number will wrap around to the lowest note and continue, and vice versa.

“Limit” mimics the behaviour of the Hardware Sweep in that once the upper/lower note limit is reached, output of the voice is silenced. This is useful for creating, say, synth drums using a fast downwards sweep.

Hardware Pitch Sweep (Voices A & B only)

Please see the section **Advanced Topic - Hardware Sweep Register**.

Detune (Voices A, B & C)

Detune adds a tiny constant offset to the pitch of the voice it is applied to. Handy for “thickening” the sound of two channels playing simultaneously or for doing those neat phasing effects.

Auto Echo (Voices A & B only)

When I developed “Nijuu” (see http://dutycyclegenerator.com for info/examples) I came up with a technique to create a parameter-driven “auto echo” effect. What this mean was that instead of having to manually insert notes into your Patterns with low volume to create an echo, you could just turn the Auto Echo on, give it a couple of parameters and let the music player code do all the echoing for you. In addition to making echo effects easier, it also gave you scope to create echo effects that would melt your brain if you tried to figure out how to do them by hand, in essence making it a creative tool in itself. To cut a long story short, I figured out a way to put this effect into NTRQ.

There are 3 parameters to set in the Pitch Effects Table.

Echo Speed

“Echo Speed” is counted in Pattern Steps (or “ticks”) and so the effect is tied to the Speed Table (the quicker your Speed setting, the quicker the echo etc.). It defines the speed of the echo feedback, the lower the number, the quicker the feedback cycle.

Echo Level

“Echo Level” defines the initial level of the echo signal. The parameter is actually the value subtracted from the note amplitude before writing the amplitude to the echo buffer. Low values will mean that the echo signal is initially very prominent in the mix, high values will produce a quieter echo. Obviously this value will depend on the ADSR setting of the voice you apply the echo to. If you have a Sustain value of, say, 8, setting an echo level of 8 will result in no echo being heard. If you set the echo level to 4 it will start at amplitude 4 (8-4 = 4).

Echo Delay

“Echo Delay” is the speed at which the echo signal decays to 0. The parameter is actually the value subtracted from the note amplitude at a rate defined by Echo Speed. Lower numbers will cause the echo signal to decay slower, higher numbers will make it decay faster. Again, like Echo Level, results will depend on the ADSR setting for the voice. Play around with Echo Level and Echo Decay until you get the effect you want.

Please read the section **Advanced Topic - Auto Echo** for a fuller explanation of the effect.

Pitch Effects Command Values

Pitch Effect	FIrst Byte	Explanation/Second Byte
Vibrato	$00 to $3F	First byte is the delay in frames before the effect is heard, $00 to $3F frames. Second byte defines the speed and depth of the Vibrato effect. $Sx = speed, 0 to F, slow to fast $xD = depth, 0 to F, small to large
Software Pitch Sweep (Overflow)	$40 to $5F	First byte is the step speed in frames, $00 to $1F Second byte is the signed number added to the note number every step, $01-$7F = positive, $FF to $80 = negative. The pitch is allowed to overflow the pitch table so will eventually wrap around when the table limits are exceeded.
Software Pitch Sweep (Limit)	$60 to $7F	First byte is the step speed in frames, $00 to $1F Second byte is the signed number added to the note number every step, $01-$7F = positive, $FF to $80 = negative. Unlike the “Overflow” mode, the output of the note will be muted if the pitch limits are exceeded. This mimics the NES’s hardware sweep but unlike the NES’s hardware sweep it can be applied to Voices C & D as well as A & B.
Hardware Pitch Sweep	$80 to $9F	First byte is the delay in frames before the sweep effect is applied. Second byte is the value written directly to the NES’s sweep unit register. For an explanation of what values to use see **Hardware Sweep Register**.
Detune	$A0	First byte is $A0 Second byte is signed offset to add to pitch setting.

Pitch Effect	FIrst Byte	Explanation/Second Byte
Auto Echo	$B0 to $CF	First byte is the echo speed in Pattern Steps (ticks), $00 to $1F Second byte define the amount of amplitude attenuation of the echo signal: $Lx = initial level $xD = decay See **Auto Echo** for explanation

Pitch Effects Example - Vibrato

To set Vibrato on an Instrument:

- set the Aux (AU) value of the instrument to $41
- at position $41 in the Pitch Table, enter these numbers: $20 $62

This will set Vibrato with depth 2 and speed of 6, with a pre-delay of $20 frames. Note that you don't need to specify the Vibrato setting in the instrument itself. As you've already set the Vibrato parameters in the table, just use a Pattern Command (in this case, $41) at any point in a Pattern and vibrato.

Pitch Effects Example - Software Pitch Sweep

- at position $42 (for example) in the Pitch Table, enter these numbers: $41 $01

This will define an upwards pitch sweep that adds 1 to the note number every second frame ($41 = step delay of 1 frame). Same as Vibrato, you can either set this in the Instrument by putting $42 (in this case) in the Aux (AU) parameter, or you can use a Pattern Command (in this case, $42) to cause the effect to be applied to notes in a Pattern.

As the first byte of the effect is $41 (and not $61), if the pitch exceeds the limits of the internal note table, the pitch will wrap around.

Pitch Effects Example - Hardware Pitch Sweep

- at position $43 (for example) in the Pitch Table, enter these numbers: $88 $82

This will define a fast downwards pitch sweep that will cut off the note when the pitch is too low (or high) for the NES to play. In this example there will also be a delay of 8 frames before the effect is applied. See **Hardware Pitch Sweep** for explanation of the bits and their meanings.

Pitch Effects Are “Stackable”

A side benefit of the implementation of these effects in NTRQ is that they are generally “stackable”. It’s possible to sequentially enter combinations of Pattern Commands (or using the Aux Parameter in an Instrument in combination with Pattern Commands) to set several Pitch Effects and they will all be applied to the notes of that Pattern/Track. It’s not particularly easy to predict what the overall effect on the sound will be but it lends itself to experimentation.

With the exception of Vibrato, Pitch Effects (when selected via a Pattern Command) are persistent. Vibrato only lasts for the note it is selected for unless you specify a Vibrato setting in the Aux Parameter of an Instrument. See table **Persistence Of Pattern Effects.**

Pitch Effects Comment

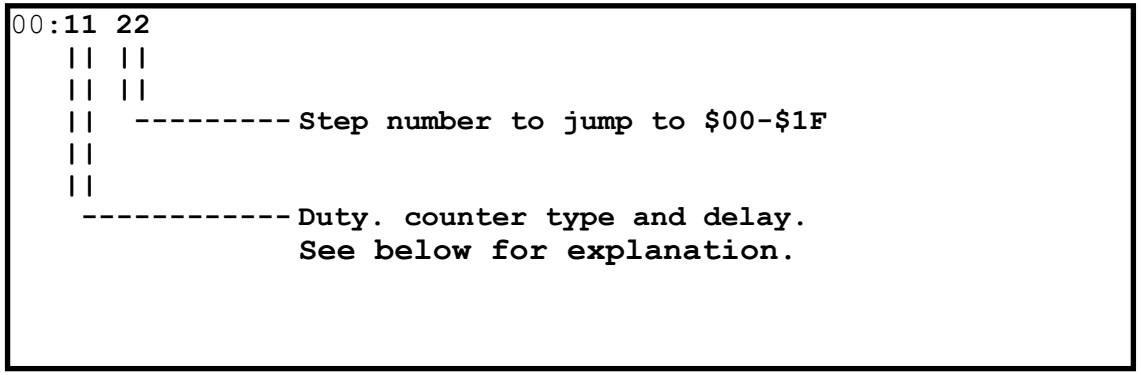
As you may realise by the complexity, Pitch Effects are one of the most powerful and flexible synthesis methods in NTRQ. Spend some time understanding how they work. With a little experimentation you should quickly be able to get the hang of them.

Probably the most important thing to figure out is the distinction between Pitch Table and Pitch Effects.

Also, because of the complexity, this is likely to be one of the areas of NTRQ that is going to have quirks/bugs.

Duty Table

The Duty Table, like the Pitch Table, has two bytes per-step.



Duty Table Values

The second value for each Duty Table entry is self-explanatory: it's the index to jump to next after the current index has been processed. The first byte takes a little more explaining as it combines three elements into one value: the actual duty setting, the counter type and the delay.

On the NES it's possible to set the tone of the two square-wave channels (or Track A & B as they are known to NTRQ). It's this tone that is known as the "Duty Cycle". It's often known as Pulse Width on other systems. On the NES you can only select from 4 possible values.

00 : 12.5 %, very thin sound
40 : 25 %, thicker, warmer sound
80 : 50%, pure square wave
C0 : 75%. actually tonally the same as "40" but the phase is inverted

These 4 values are what go in the first value of a Duty Table step. However, there's slightly more to it with NTRQ! Read on.

You also use the first value to set the delay (number of ticks to wait on the current Duty Value before moving to the next step) by combining it with the Duty Value. The range for the delay is 00 to 1F. Therefore if you want to set a Duty Value of 80 with a delay of 7, you'd put in the value "87", a Duty Value of 40 with a delay of 18 would be 58 etc. (number in hexadecimal).

The last thing to know is that the first value is also where you set the Duty Counter Mode. There are two modes: "Tick Counter" and "Note Counter" (see below for explanation of the two modes). "Tick Counter" is the normal operation which doesn't require any further tweaking of the first Duty Table value. However, if you want to use "Note Counter" mode, you need to add 20 (hex) to the value. Therefore, to use "Note Counter" mode with a Duty Value of 80 and a delay of 7, you'd set the value A7 (80+20+07 = A7 in hex) etc.

Counter Mode: Normal Vs Note-Triggered

In normal mode, the counter speed is in frames/ticks. In "Note-Triggered" mode, the counter actually counts the number of notes triggered. In normal mode the index is reset to the Instrument setting for each new note. In Note-Triggered mode, the index is only reset at the next **Instrument Command** in the Pattern.

Duty Table Example

Toggle between duty $80 and duty $40, every 2 frames

```
01: 82 02          ;set duty $80, wait 2 ticks then jump to step 02
02: 42 01          ;set duty $40, wait 2 ticks then jump back to step 01
```

Toggle between duty 00 and 80 every 7 notes

```
01: 27 02          ;set duty 00, mode "Note", wait 7 notes then jump to step 02
02: A7 01          ;set duty 80, mode "Note", wait 7 notes then jump back to step 01
```

ADSR Envelopes

NTRQ uses software ADSR Envelopes to shape the amplitude of instruments on Track A, B & D. “ADSR” stands for “Attack-Decay-Sustain-Release” and refer to the four phases of the envelope over time.

Attack (A) = time taken for amplitude to rise from 0 to max (see Note-1)
Decay (D) = time taken for amplitude to then drop to the Sustain Level
Sustain (S) = the amplitude at which the note is held (see Note-2)
Release (R) = the time taken for the amplitude to diminish from the Sustain Level to 0

Note-1: Generally, “max” = $0F (15) as this is the highest amplitude that can be set in the hardware. So, generally the Attack Phase starts at amplitude $0 and rises (speed depending on the Attack setting) to $F. However, if the Amplitude setting is “0”, the ADSR Envelope will start immediately at the maximum level. An additional feature is if you have a Decay setting of “0” this will set the “maximum” amplitude to whatever value you have set in “Sustain”.

Note-2: The “Gate” parameter (see **Instrument Table)** determines whether or not a note is held in the Sustain Phase. If the Gate parameter is 0 then the note will be held until a new note or a Key Off Command is encountered. If the Gate parameter is any other number, ‘X’, the ADSR Envelope will enter the Release Phase after ‘X’ frames. Regardless of the Gate parameter, the ADSR Envelope will be forced into the Release Phase when a Key Off is encountered in a Pattern.

ADSR And Tie Notes

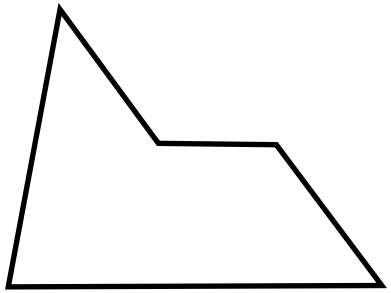
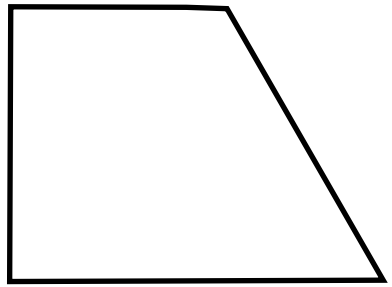
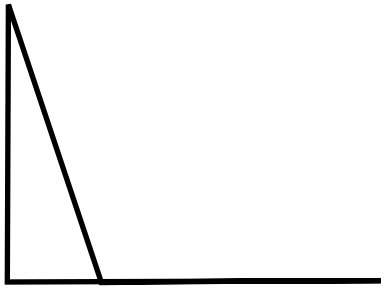
With normal notes, the ADSR Envelop is reset to the start (Attack Phase) every time a new note is played. However, when a Tie Note is played, the ADSR Envelope is not restarted.

Track C and Track E

As these voices don’t have amplitude, ADSR is not used. However, for Track C (triangle) the Gate Time parameter is used. It’s the number of frames to count from the start of a new note before silencing the voice output. Similarly, Key Off on Track C just cuts the output .

See next page for some ADSR examples.

ADSR Envelope Examples

Example 1 48 88 Attack : 4 Decay : 8 Sustain : 8 Release : 8		Example 2 01 F8 Attack : 0 Decay : 1 Sustain : F Release : 8	
Example 3 40 48 Attack : 4 Decay : 0 Sustain : 4 Release : 8		Example 4 00 04 Attack : 0 Decay : 0 Sustain : 0 Release : 4	

These are just approximations of different shapes and they don't really illustrate the effect of the Gate parameter. In fact all of these examples should be imagined to have a Gate Time of 01 (or thereabouts). In Example 1 & 3, the Gate Time is the flat bit in the middle.

The other thing to remember is that using the two Pattern Volume Commands (see **Pattern Commands**), the envelope shape will be scaled so you have even more scope to sculpt the amplitude exactly how you want it.

Speed Table

The Speed Table is what controls the playback speed of your songs. I've already covered setting an initial Speed Table index for each song (Song Info) and the Pattern Command and the Song Command for changing the Speed Table index so here's just some more information on how the Speed Table works.

As always, the number in the left-hand column is the table position/index. The other two numbers on each Speed Table row are the "speed" for that index followed by which Speed Table index to jump to next.

Speed Limitations And How NTRQ Tries To Get Around Them

Warning: Technical Stuff Ahead!

You might be able to figure out from the picture that entry 00 in this Pitch table sets a speed of 3, then jumps to step 1 to set a speed of 2, then jumps back to step 0. The effect of this will be a speed somewhere about in between speed 2 and speed 3. If you wanted to just have speed 3, you'd change the top-right number to 00 which would just loop around step 0 of the table.

That explanation of how the Speed Table in the picture is set up might seem a bit weird. Why would you want to fluctuate between two speeds? It's an inherent problem of playing back music on a fixed rate refresh system (i.e. on an NTSC, the resolution of the refresh is 60 times-per-second and is fixed) that your playback speed is always a multiple of the refresh rate. A speed of 3 therefore, gives you a Pattern Step time of 3 frames which = 20 steps (or ticks) per second. Now, if you change the speed instead to 2, you then get a speed of

30 steps per second. A speed of 4 = 15 steps per second. All divisions of 60 because it's based on a refresh rate of 60hz (NTSC). The problem here is that the difference between those speeds is pretty big in musical terms.

So how do you get finer control over your song? Well, as you've probably already worked out, you use the Speed Table to constantly manipulate the step speed by varying it between two or more speeds. Generally it works fine and your ear is fooled. When it doesn't really work is when you have very "busy" Patterns. I'll explain.

Because of the way the Speed Table works, the steps in the Speed Table are in sync with the steps in your Patterns. In the above example, the speed is switched between 02 and 03 every two steps: speed 02 on the first step, 03 on the second and so on. This means that you're getting an average speed over two Pattern steps. This is how the trick works. However, if you don't leave empty steps in your Pattern you might have;

```
C 4
D 4
E 4
```

instead of;

```
C 4
----
D 4
----
E 4
----
```

you'll hear that every other note is at a different speed. This is quite a useful effect in itself but it's important to be aware of how this works so that if you want to use the speed trick you shouldn't fill every step in your Patterns but use, say, every other step.

Another NTRQ Speed Table Trick - "Overflow Counting"

As well as constantly fluctuating the speed between steps there's another commonly used trick to get finer resolution for your playback speed. If we refer to the normal method as "frame counting" i.e. your speed setting is a count of frames before moving to the next step/tick, the second method is what is sometimes known as "overflow counting". This method uses 8-bit arithmetic to count when the results of an addition has exceeded the limit of an 8-bit number. NTRQ then moves to the next step every time this happens. Let me illustrate.

Without getting into a lot of technical detail and simplifying things for the purposes of this illustration, the largest *unsigned* (i.e. positive) 8-bit number is 127 (this equates to $7F in hexadecimal). If you perform an addition and the result exceeds this, a CPU flag is set. The "overflow count" method takes advantage of this. For example, if you set a speed of 30, this number is added to the overflow counter every frame, so the overflow counter would go;

00
30
60
90
120
150

As you can see, on the 6th frame the result is 150 which is larger than 127. At this point, NTRQ will move to the next step/tick. However, the actual result has 127 subtracted from it i.e. 23. So once the step/tick has been updated, the overflow counter doesn't start at 00 again but this time at 23. Eventually happens is that the overflow counter exceeds 127 one frame quicker (on the 5th frame instead of the 6th). It's this trick that gives you finer resolution of the speed.

The downside to this method though is that you can't really predict when the refresh speed is going to do that little skip (well, you could but it's beyond most people's mental arithmetic!) so you do notice the speed fluctuations in your song.

Anyway, I figured out a way of giving you both methods in the one Speed Table in NTRQ.

If you use speed value 00 to 10, the "frame counter" method is used.

If you use speed 11 onwards, the "overflow counter" method is used.

This is the value in the table and not the index number so you can, if you really wanted to, mix both methods in a Song either by switching between them with Speed Table Index Commands or even have the Speed Table loop between two steps that use a different method on each!

DPCM

DPCM Support

DPCM support is currently fairly limited unless you are comfortable and able to build the NTRQ ROM yourself (which may/may not be supported on initial release).

DPCM Notes and Commands

OK, entering notes for the DPCM channel is a bit odd. If you never have more than 8 samples in the ROM then it's fairly straight-forward.

Enter a value in the Note Column that takes the format "SP" where;

S = the sample number, $0 to $7
P = the playback rate, $0 to $F

DPCM And The Instrument Command

The DPCM track doesn't have instruments as such but the Instrument Command has a special use. You might have noticed that when entering a DPCM note, the maximum sample number is 7. This means that you can only use 8 samples. However, the Instrument Command acts like a "bank switch" command for DPCM. If you have, say, 12 samples, to access samples $8 to $B you need to set the DPCM bank to bank 1 by using the Instrument Command, $01.

Samples $00 to $07 = bank 0 (Instrument Command $00, sample number $0 to $7)
Samples $08 to $0F = bank 1 (Instrument Command $01, sample number $0 to $7)

and so on.

DPCM Looped Playback

You can play any of your samples normally (i.e. not looped) or looped. To do this you specify $00-$07 as the "Instrument" for non-looped or $10-$17 as the "Instrument" for looped. In both cases the "bank" range is still 00 to 07 so you can play any sample in either mode.

DPCM Pattern Commands

There are only a few Pattern Commands that work on the DPCM track.

Pattern Command	Value	Notes
Instrument	$00-$07 $10-$17	Set DPCM “bank” 0 to 7, normal playback Set DPCM “bank” 0 to 7, looped playback
Pitch Table Index	$60-$9F	** SPECIAL USE, SEE BELOW**
Duty Table Index	$C0-$DF	** SPECIAL USE, SEE BELOW**
Speed Table Index	$E0-$F3	Works the same as any other Track
Note Delay	$F4-$FD	Works the same as any other Track
Terminate Pattern	$FE	Works the same as any other Track

DPCM Special Use of Pitch Table Index Command $60-$9F

The Pitch Table Index Command ($60-$9F) has a special use for DPCM. You can use it to truncate the length of a sample. $60 = play whole sample, the maximum value depends on the samples you’re using of course. There is no error/range checking of this command so you’ll need to figure out for yourself what are “safe” values to use.

DPCM Special Use of Duty Table Index Command $C0-$DF

The Duty Table Index Command ($C0-$DF) has a special use for DPCM. You can use it to offset the start of a sample. $C0 = play from beginning, the maximum value depends on the samples you’re using of course. There is no error/range checking of this command so you’ll need to figure out for yourself what are “safe” values to use.

Advanced Topic - Editing the AD-R Table

First off, how to display this hidden table.

To Reveal The AD-R Table Editor: you need to move the cursor to the Song Info Window, then **hold SELECT+B** then press **UP.**

To recap the ADSR settings from Instruments section: the first two values in an instrument are Attack/Decay, Sustain/Release.

Apart from Sustain, the values are not actually the values used in the software envelope but are indexes ($0 to $0F) into 3 small tables. Those 3 tables are what you edit in the AD-R Table Editor (notice it's not "AD**S**R" Table Editor because you don't edit the Sustain value in this way). These 3 tables contain the actual frame counter values that are used in the software envelope and you have one set of values per NTRQ save file.

Looking at the screen-grab, you will see the labels "ATK", "DEC" and "REL" which correspond to what you'd expect. Reading across from "ATK" you'll see a single digit, "0". This is the current Attack value you are editing (0 to F). As you move the cursor around the table this number will change. It's just a visual reference so you know which value you are changing. Looking further across, there are eight hex

numbers (indexes $0 to $7) and immediately underneath are a further eight hex numbers which are values for indexes $8 to $F. This is the same for “DEC” and “REL”.

Something you may have noticed is that some of the numbers are different colours. This is an important aspect of editing the AD-R tables.

Firstly, the first value (index $0) in each table is not editable. This is because the value in these positions (00) is used for when you want zero Attack (or Decay/Release). This number is displayed in dark grey to show that it’s not editable.

Secondly, the next 3 hex numbers (indexes $1 to $3) are displayed in Red. This is because the first 3 numbers have a special use that is different from the remainder in each table.

I’ll deal with the normal numbers first (indexes $4 to $F). For each phase of the software envelope, this value is loaded into an internal frame counter which defines the time taken by each phase. However, because the amplitude of the envelope is only incremented (or decremented) by 1 each time the counter elapses, the true time taken (in frames) is the amount that the amplitude needs to be changed multiplied by the frame counter. For example, the Attack Phase always starts at 0 and always ends at F amplitude. So if you set an Attack Speed of 1 (in the AD-R Table), it will take 16 frames for the envelope to reach the end of the Attack Phase. An Attack value of 2 will take 32 frames and so on.

Now, here’s where the first 3 numbers (indexes $1 to $3) differ. Instead of loading these values into a frame counter, these values define the amplitude step size and are actually added (or subtracted, depending on the Phase) to the envelope amplitude each frame until the target amplitude is reached. So, setting a value of 4 in one of these indexes would actually only take the Attack Phase 4 frames before the envelope moved to the Decay Phase. This is because the amplitude only ever goes from $0 to $F (16 steps in total). So 16 steps divided by 4 (step size) = 4. Thus, 4 frames to go from $0 to $F amplitude. This gives you a way to have very fast Phase transitions in your ADSR envelopes.

So, back to Instruments: the values you put in your AD/SR parameters corresponds to indexes in the AD-R tables. Simple!

To Edit The AD-R Table: just move around with U/D/L/R, hold **A** and then use U/D/L/R to change values. At any time **press B to quit**. You can change the values while the song is playing.

Advanced Topic - “Clear Data” Menu Controls

The “Clear Data” Menu is where you can initialise all or some of the data in your current NTRQ save file.

To reveal the Clear Data Menu: move the cursor to the Song Info Window, hold **SELECT** and **B** then press **DOWN.** You can close the menu at any time by pressing **B.**

To initialise data, move the cursor onto each check box and press **A** to toggle it on/off (off=empty grey square). Then move down to the “NO/OK” line. If you press **A** on “NO” the menu will just cancel. If you press **A** on “OK” the data you have “checked” to be cleared will be cleared and the menu will close.

Purge Patterns is used to clear Patterns that are currently not being used in your song(s). It’s handy for reclaiming Pattens if, say, the Clone function has stopped working (you’ve run out of empty unused Patterns). Please use with care!

Advanced Topic - “Options Menu”

To reveal the **Options Menu**: when in the Song Info window, hold **SELECT + B** then press **LEFT**

To close the **Options Menu**: press **B**

From this menu you can set various NTRQ options.

Note Entry Audio - turns on/off audio when entering notes. **Please note :** the actual note played will be affected by the Transpose value of the Song Step that is currently selected (i.e. highlighted in the Song Arranger). This is to ensure that the pitch you hear as you enter/ modify it will be the same when the Pattern is played.

B-Paste - normally, pressing B on an empty Pattern cell will paste the last deleted value. Use this to enable/disable the paste feature.

Display BPM - a new feature enables an actual BPM value to be calculated in real-time based on your Speed Table settings. This option turns the display of the BPM on/off.

Alt Note Entry - normally, note entry uses U/D to set the semi-tone value and L/R to set the octave. Select this option to swap these controls around. This also affects all other editors. When setting numerical values, the controls to +/-1 and +/-10 are swapped around.

Invert Octave Entry - normally when setting the octave of a note, L will decrease the octave number and R will increase it. Select this option to reverse this.

Live Play : No Sync
See **Advanced Topic - Live Play!** for details.

These option settings are saved with you NTRQ .sav file.

Advanced Topic - Context Navigation Shortcuts

There are a couple of shortcut navigation tricks you can use in the Pattern Editor and the Instrument Table and Song Arranger.

Pattern Editor Advanced Navigation

If your cursor is in a Command Column, holding **SELECT** then pressing **DOWN:**

- if the cursor is on an empty cell, NTRQ searches backwards in the Pattern until a “Set Instrument Command” is found. The cursor then jumps to the Instrument Table on that Instrument. This also works if the cursor is on a Set Instrument Command, of course.
- If the cursor is on a Vibrato Command (or Pitch Sweep), the cursor will jump to the Vibrato section of the Pitch Table on the entry specified in the command.
- If the cursor is on a Pitch Table Index Command, the cursor will jump to the Pitch Table on the entry specified in the command.
- If the cursor is on a Pitch Effect Command, the cursor with jump to the Pitch Effects section of the Pitch Table on the entry specified in the command.
- If the cursor is on a Duty Index Command, the cursor will jump to the Duty Table at the entry specified in the command.

or holding **SELECT** and pressing **UP:**

- If the cursor is on a Speed Table Index Command, the cursor will jump to the Speed Table at the entry specified in the command.

Instrument Table Advanced Navigation

If your cursor is on the Pitch Table [PT] column, holding **SELECT** and pressing **RIGHT** will make the cursor jump to the Pitch Table at the specified index.

If your cursor is on the Duty Table [DT] column, holding **SELECT** and pressing **RIGHT** will make the cursor jump to the Duty Table at the specified index.

If your cursor is on the Auxiliary [AU] column and you’ve specified a Pitch Effect value, holding **SELECT** and pressing **RIGHT** will make the cursor jump to the Pitch Efffects section of the Pitch Table at the specified index.

Song Arranger Advanced Navigation

If your cursor is on a Speed Table Index Command, holding **SELECT** and pressing **RIGHT** will make the cursor jump to the Speed Table at the index specified in the command.

Advanced Topic - Changing The Pattern Length

Because each pattern is the same length in an NTRQ file, I tried to make things as flexible as possible by allowing you some freedom to change this length setting. I had to put some restrictions on the maximum and minimum size but there should be enough scope there to suit your way of working, especially also with the inclusion of the **Terminate Pattern Command.**

The Pattern Length you decide on will depend a lot on how you work and what you're working on. The default length is 32 steps ($00-$1F). This gives you a maximum of 94 unique Patterns. If you think about the fact that the maximum size any one Song can be is 120 ($78) steps, that means (without the use of commands to repeat Song Steps), the maximum size of a Song with the default Pattern size of 32 steps is 3840 steps. Which may seem a lot but at an average speed setting, say 03, this gives you a maximum song time of about 4 minutes (though don't quote me, that was just worked out in my head - it's close enough).

If you were to increase the Pattern Length to, say, 64 steps, you'd then double the maximum song time to about 8 minutes but sacrifice Patterns: you'd only have about 46 unique Patterns in this case. Of course, as I said earlier, with the use of the **Repeat Song Step** and **Repeat Song Section** commands this is a bit arbitrary but it illustrates how the Pattern Length is important.

Here's where the flexibility requires some planning though. Even though it is possible to change the Pattern Length *at any time*, even if you've already created a song (or part of a song), the results of altering this value with existing data is far from predictable.

First off, it will destroy your Patterns. It won't actually get rid of any of the data contained within the Patterns but internally the start and end of each Pattern will be changed. Similarly your Song, while remaining structurally intact, will be altered i.e. the Pattern numbers etc. will still be the same but what those Patterns play will not.

Secondly (and related to that last point), if you *increase* the Pattern length, you're going to lose Patterns. As a safety measure, if you change the Pattern Length, NTRQ scans your Songs for any Pattern Numbers that no longer exist and puts a zero in there instead.

Advanced Topic - Hardware Sweep Register

The two square-wave voices have an 8-bit register that can be used to sweep their frequency. When triggered, the pitch will be continuously swept at the rate and direction specified by the value set in the register (see below). Sound output of the voice will be muted if the frequency exceeds (low and high) the limits of the NES's 11-bit frequency registers. The sweep continues to be active until switched off.

Register Details:

Bit(s)	Function
7	0 = sweep off, 1 = sweep on
6-4	Step speed, 0-7. Lower numbers = faster step speed
3	0 = sweep down, 1 = sweep up
2-0	Sweep amount. Lower numbers = higher sweep amount

If you want a more in-depth explanation, search out the extensive NES APU reference document on the NesDev website: http://nesdev.parodius.com/apu_ref.txt

Advanced Topic - More About NTRQ Files

NOTE : Nestopia (on Mac/OSX at least) uses ZIP compression when writing save files. This format is totally incompatible with NTRQ and must be decompressed first before using (clue: if it's smaller than 8kb then it's compressed). On Mac/OSX you can decompress them easily with the built-in Archive Utility. Nestopia copes well with either uncompressed or compressed save files but, most annoyingly, even if you load up NTRQ using an uncompressed one, Netsopia will re-compress it when you quit. I've tried to ask for this to be changed but so far I've not managed to make it happen.

Forcing The NTRQ Save File To Be "Reformatted"

If you hold SELECT+UP on controller 1 when powering-on the NES (or resetting it), your ".sav" file will be erased and "reformatted". You'll also see the start-up logo again.

Actually, here's a little tip for you : if you perform this reset and then while still on the logo screen, if you reset the NES/emulator before pressing start, NTRQ will start up with the logo "music" still loaded up. This is because when playing the logo animation music, I decompress it from ROM into SRAM to play it.

Advanced Topic - Auto Echo

How NTRQ Auto Echo Works

What happens is that the code intercepts the values that are written to the pitch and amplitude registers of Voice A & B each time the NTRQ player is refreshed (50/60hz). It then stores these values in a circular buffer in RAM. Depending on the speed you have set for the echo, the amplitude values in the RAM buffer are modified (gradually down to zero) and these modified values are inserted back into the audio signal when there are “gaps” in between the notes in the Pattern.

It’s knowing how the “gaps” work that will help you understand how to get good results from Auto Echo. You need to “engineer” the gaps in order to hear the echo.

NTRQ looks for either silence (no note playing) or for playing notes whose ADSR has reached the Release phase. During these times is when the Auto Echo “signal” is fed back into the audio output. As more notes are encountered the echo output is not heard until there is more silence or the new note has reached the Release Phase.

There are 2 ways that a note enters the Release Phase. Either by way of setting a Gate (GT) value in the instrument (which forces the ADSR into release after the number of frames specified) or by way of using the Key Off (“OFF”) command in a Pattern. So either use an Instrument that has a short Gate Time, or use “OFF” commands in your Pattern to tell NTRQ where it can insert some echo signal.

It’s that easy!

Auto Echo Limitations

Because Auto Echo uses RAM buffers (two of them actually, one for voice A and one for voice B), it means I have to put a limit on the maximum echo speed - the higher the echo speed setting, the slower the echo, the bigger the buffer required. I’ve set the buffer size to 48 frames, maximum. If your song has a Speed setting of 3, the slowest echo speed you can set is 12($C) - 48 frames / 4 frames per Pattern Step (which is what you get with Song Speed of “3”) = 12. If you have a slower song, the maximum you can set the echo speed to decreases and vice versa.

However, because you’re free to change the Song Speed independently of any Echo Speed parameters you might use, there’s no way that I can check that you’re using a “legal” value for the Echo Speed. Therefore, if you think you’re hearing weird effects (though it’s not easy to tell!) it might be worth digging out a calculator and doing this quick calculation:

(Song Speed + 1) x Echo Speed x 3

If that comes out greater than 48, you need to decrease the Echo Speed (or increase the Song Speed!). In the case of having one of those hybrid speeds (multiple Speed Table entries in a loop), you'll have to use your best guess. At the time of writing this I've not done any real tests with Speed that use the "Overflow Counter" method. Probably best to avoid using that method with Auto Echo I think.

Advanced Topic - External Sync

Coming soon.

Advanced Topic - Live Play!

Bit of an experimental feature and is in very early stages of development (as of NTRQ V1.0 release) but if it becomes a popular thing I might look at improving it for future versions. Or removing it if the opposite is true!

“Live Play” enables the use of a second controller in port 2 to play along to your songs (or even if a song is not playing so it can be handy for previewing sounds while you are making adjustments to your instruments).

First you need to enable Live Play in the **Options Menu**. When you do you’ll see an extra bit of display text underneath the Song Info window.

Live Play Display (from left to right)

“T” indicates the Track (voice) that you will be playing on.
“I” indicates the current instrument Live Play is using.
“O” indicates the current octave you will be playing in.
“+” or “-” indicates the current semi-tone transpose value
“S” indicates the current Pitch Slide setting for Live Play.

Live Play Controls

This is where the fun starts! **Remember, this is all on Controller 2.**

Function	Keys	Notes
To play notes	Hold down: D/L/U/R to play A, A#, B, C B+D/L/U/R to play C#, D, D#, E A+D/L/U/R to play F, F#, G, G# B+A + D/L/U/R to play A, A#, B, C (octave up) The note will play for as long as you hold the key combo but will obey the Gate setting in the instrument also.	The notes start at A instead of C because the lowest NES note is an A. Currently there is no way to transpose by a semi-tone value.
To change Track/Voice	Hold START and press L/R	
To change Instrument	Hold SELECT and press B/A	
To change Octave	Hold SELECT and press U/D	
To change Transpose	Hold SELECT and press U/D	Limited to +/- 12 semi-tones (+/- C) Only works on Track A, B & C
To change Pitch Slide	Hold START and press: U/D to increment/decrement slide value	Will constantly slide between each note you play. Only works on Track A, B & C

Live Play - Track D (Noise)

Track D (Noise) is treated slightly differently.

The Octave setting selects between the two clock modes. Odd numbered octaves is the normal “Noise” and even numbered ones give you the “metalic sounding” notes.

Live Play - Track E (DPCM)

Track E (DPCM) is treated slightly differently.

The Octave setting sets the pitch of the sample, 1 = lowest, 7 = highest. A setting of “7” is equal to maximum/normal pitch (“F” in an NTRQ Pattern).

The Instrument setting selects the DPCM “bank” (see **DPCM** for details). Valid banks are currently only 00 or 01.

This then leaves the “play notes” controls to play each sample at a fixed pitch (fixed by the Octave setting as described) and makes it more like drum pads.

Pitch Effects

Obviously, as you don’t have the Pattern timeline, what you can achieve from Pitch Effects in Live Play is going to be slightly limited. Best thing to remember is that many Pitch Effects are persistent on the Track/Voice they are used on until you clear them (with 40 in the Aux parameter). So, for example, if you use the Aux parameter to enable echo on an Instrument, once you’ve played at least one note with the instrument, Auto Echo will remain switched on so you could then go back and change the Aux parameter to something else (Vibrato, Pitch Sweep etc.)

You’ll just have to have a play around to find combinations of things that work well.

Experimental

Because this is a new feature (for V1.0) and has only really had limited testing, there may be a combination of buttons/conditions that causes glitches. Probably the most likely thing is for notes to get stuck on. If this is the case press START a couple of times (depending if your song is playing or not) and the notes will be cancelled. There also might be issues if you have “Note Entry Audio” on and are trying to enter notes (with Controller 1) while trying to play live notes (with Controller 2). If you posses such octopus-like dexterity I’d recommend turning “Note Entry Audio” off, just to be safe. Your Pattern data won’t be affected by Live Play at all though so no need to worry about that aspect.

No Sync?

You'll have also probably spotted in the Options Menu that the actual option is "Live Play : No Sync". This is for future versions because the plans for adding an external sync feature to NTRQ utilises the second controller port. Therefore, if Live Play is enabled, the sync signal will not be available and vice versa. More of this in future releases. Hopefully.

Advanced Topic - Building the NTRQ ROM

Coming soon.

DPCM - List Of Included Samples

This list is subject to change but for the first release, here's what samples are included in the ROM;

"Bank" 00

00 TR808 Kick
01 TR808 Snare
02 TR808 Clap
03 TR808 Cowbell
04 TR808 Rim Shot
05 TR909 Kick
06 TR909 Snare 1
07 TR909 Snare 2

"Bank" 01

00 TR707 Kick
01 TR707 Snare 1
02 TR707 Snare 2
03 TR707 Rim Shot
04 TR707 Hi Tom
05 TR707 Clap
06 TR707 Snare & Clap (mix)

NTRQ Quick Guide - Song Commands and Speed Table

Song Commands

Feature	Command Value	Notes
Jump To Song Step	$00 - $7F	Jump is performed after the current step is finished.
Repeat Current Song Step	$80 - $9F	$80 = once, $9F = 32 times
Repeat Song Section	$A0 - $BF	$A0 = once, $BF = 32 times Place command on first/last step you wish to repeat. Count is set by first command.
Set Master Volume	$C0 - $CF	$C0 = 0, $CF = F
Set Master Volume Fade		Fade volume down if volume >0, otherwise fade up. $D0 = fastest, $DF=slowest.
Set Speed Table Index	$E0 - $F3	$E0 = step 00, $F3 = step $13

Speed Table

Value 1	Value 2
$00 - $10 = speed using frame-counter method *$00 = fastest, $10 = slowest*	Speed Table index to jump to next
$10 - $80 = speed using overflow-counter method *$10 = slowest, $80 = fastest*	Speed Table index to jump to next

NTRQ Quick Guide - Pattern Commands

Feature	Command Value	Notes
Select Instrument	$00 - $1F	
Slide To Note	$20 - $3F	$20 = off, $21 = slowest, $3F = fastest Pitch will slide to note on same step as this command.
Select Pitch Effect	$40 - $5F	$40 = cancel Pitch Effect. Other values = index into Pitch Effect Table
Set Pitch Table Index	$60 - $9F	$60 = off. $61 = Pitch Table index 01, $9F = index 3F
Set Pattern Volume	$A0 - $AF	Command affects all subsequent notes
Set Note Volume	$B0 - $BF	Command only affects note on same step as the command or modifies an already playing note.
Set Duty Table Index	$C0 - $DF	$C0 = Duty Table index 00, $DF = index 1F
Set Speed Table Index	$E0 - $F3	$E0 = Speed Table index 00, $F3 = index 13
Delay Note/Key Off	$F4 - $FD	Delay note/key off by several frames. $F4 = 0 delay, $FD = 9 frames delay
Terminate Pattern	$FE	End current Pattern on step that command is on. All Patterns on current Song Step are truncated. Playback continues with next Song Step

NTRQ Quick Guide - Pitch Effects Commands

Remember: these go in the Pitch Table > index $40

Feature	Value 1	Notes / Value 2
Vibrato	$00 - $3F	Value 1 is delay before vibrato starts Value 2 is *$SD* where *S*=speed, 0=off, 1=slow, F=fast and *D*=depth, 0=off, 1=shallow, F=deep
Software Pitch Sweep (overflow)	$40 - $5F	Value 1 gives step time, $00 to $1F Value 2 is semi-tone to add per step, (positive/up) $00-$7F, (negative/down) $FF-$80
Software Pitch Sweep (limit)	$60 - $7F	Value 1 gives step time, $00 to $1F Value 2 is semi-tone to add per step, (positive/up) $00-$7F, (negative/down) $FF-$80
Hardware Pitch Sweep (A&B)	$80 - $9F	Value 1 is delay before sweep, $00 to $1F Value 2 is raw value to write to hardware sweep unit. See **“Advanced Topic - Hardware Sweep Register”**
Detune	$A0	Value 1 is always $A0 Value 2 is detune amount, (positive) $00-$7F, (negative) $FF-$80
Auto Echo	$B0 - $CF	Value 1 is echo speed, $00 to $1F. $00 = echo off. Value 2 is *$LD* where *L*=initial amount to subtract from note amplitude, 0 to F and *D*=amount to subtract from amplitude per echo, 0 to F. See **“Auto Echo”** for explanation.

NTRQ Quick Guide - Pitch Table / Duty Table Commands

Pitch Table Commands - Remember these go in Pitch Table, indexes $01 to $3F

Value 1	Value 1 Meaning	Value 2
$00 - $40	Positive semi-tone offset	*$VD* where *$V*= volume scale, $0 to $F and *D*=delay in frames until next step
$50 - $AE	Absolute note value *$50*= note $00 (A 1), *$AE*= note $5E (B 8)	*$VD where $V= volume scale, $0 to $F and D=delay in frames until next step*
$B0 - $EF	Negative semi-tone offset *$B0 = -64, $EF = -1*	$VD where $V= volume scale, $0 to $F and D=delay in frames until next step
$F0	Delay Command	Value 2 is delay in frames. (Value 1 is always $F0).
$FF	Jump Command	Value 2 is index to jump to, $00-$3F

Duty Table Commands

Value 1	Value 1 Meaning	Value 2
$0x, $4x, $8x, $Cx	Set Duty $00, $40, $80 or $C0 *x* = delay until next step, $00-$1F in **FRAMES**	Duty Table index to jump to next, $00-$1F
$2x, $6x, $Ax, $Ex	Set Duty $00, $40, $80 or $C0 x = delay until next step, $00-$1F in **NOTES**	Duty Table index to jump to next, $00-$1F

Command-Line Tools

There are a number of command-line tools included with NTRQ. Where possible I will try to provide the source code and also Windows and UNIX (Mac) binary files for these tools.

NTRQ2NSF

NTRQ2NSF allows you to output a .NSF file from your NTRQ .sav file. Please see the included text file “Using the Command-line Tools.txt” for usage and explanation.

DCM2NTRQ

DCM2NTRQ allows you to “patch” your NTRQ ROM with your own DPCM samples. Obviously, if you are using a patched ROM, for each new release of NTRQ you’ll need to re-patch the new ROM. Please see the text file “Using the Command-line Tools.txt” for usage and explanation.