User's Guide

the **redux** edition

Program & original documentation written by:
Martin "Dexter" Wodok

Layout & music:
Manfred "Pink" Linzner

Conversion to sane documentation:
Dale "lavaburn" Whinham (typesetting & rewriting)
Christopher "Deltafire" O'Neill (ripping originals)
Brendan "syphus" Ratliff (huge amounts of help)

Revision **1.0pre2** (13-Dec-06)

Contents

Preamble

An important introductory note from lavaburn

This is a full remake of the official documentation found within AHX. The original idea was to make something that was printable for easy reading, as it is very difficult to read the integrated help. But I also decided the documentation was – with all due respect – in many places inadequate and crude, especially to those who are new to the world of tracking.

I've expanded and clarified where I felt necessary. I've rewritten many sections and reordered them – there are far too many changes to list. Expect this document to go through a few revisions as things are noticed and feedback is received. :)

> # ☀ WARNING! ☀
>
> The documentation begins on the following pages with the original authors' thanks, helloes, and contact details. These notes are **only kept here for legacy reasons and completion**. You are advised **NOT** to bug the authors for updates etc. This program's final release is now over 8 years old, and it's highly doubtful that a new release will surface, and equally as doubtful that they want to be asked silly questions about it.
>
> **DO NOT contact Dexter and/or Pink regarding this document.**
> **This is an independent work, which is not endorsed by them.**
>
> I **CAN NOT** and **WILL NOT** be held responsible for your misuse of the authors' personal contact details, nor can I be held responsible for your stupidity. You have been warned.

Please see the end of this document for details on how to get in touch with me to forward your suggestions, corrections, comments and so on.

Thankyou for reading, sorry for the disclaimer (although it's necessary) and I hope you enjoy the updated manual! :)

– lavaburn

Thanks

...must go to the following for contributing with suggestions and bug-reports/tunes/code:

K-P Koljonen, Tommy, Florian Vorberger, Twice/Lego, Evert-Jan Slypen, Ramosa, Ben Preece, Kaare Johansen (thanks for writing AHX Ripper!), Frostycat, freQvibez/Offence, Oxide&Stargazer/Sonik, Geir Tjelta, ¡Tony Stuvo Aky! / Odrusba+Llfb, Rawstyle/Lego and many more.

Helloes

...go to:

Twice/Lego, Gryzor, Offa/TBL, Equalizer/TBL, Geggin/Censor Design, Peter Kunath, Ixxy/TRSI, Abyss/Future Crew, Magic Toon and all the guys I haven't written back :-)

Intro

Hi there music lovers!

We finally made it; AHX Version 2.xx has come to stay! At last we can be proud of saying that it is the best!

There has been a lot of changes this time, especially to finally include filtering, preferences and volume meters in AHX. And it works! (as far as we know, at least).

Don't hesitate to contact us if you can't get it working or there's something you can't figure out.

Some technical info:

- Sourcecode: 326k / 14500 lines
- Main Program: 186k (unpacked)
- GFX data: 60k (unpacked)
- Worktime: 9958654985476 hours
- ASM time: about 27 minutes

Dexter had quite a job this time, and since we don't have more AHX coders, and he's extraordinarily lazy, it was hell on earth. He coded the filters about 12-13 times and much time was lost on thinking about why it doesn't work but afterwards, it all ended well.

You're looking at the result now. :-) What do you think?!

AHX finally deserves to be called *"the best SID tracker around"*, which many of you have been calling it in your psychotic letters to us...

One thing that surprised us though, was the fact that all of you said it was "great", "superb" and so on.

Very few (no-one!) complained or came up with new, original ideas and/or useful features! Most of the things in AHX2.0 are self-invented, but thanks must go to the Hamre brothers who wrote the original PT docs and of course to MAHONEY&KAKTUS.

Thanks guys, you really are a great bunch of people!

<div align="center">

C64 lives - SID rulez

PC is dead - Soundblaster sucks

</div>

Contact

All AHX composers: Please send all your tunes to me (Dexter and/or Pink).

Something you'd like in the next version of AHX? Have you found any bugs? Or do you just want to play tic-tac-toe by mail?

Anyway, write to:

Martin Wodok (Dexter)
Rattenberger Str. 28
81373 Muenchen
Germany

(No games, please!)

Planned release date for AHX v3: Haven't got the foggiest idea...

For musics, write to:

Manfred Linzner (Pink)
Rupert-Mayer-Str. 6
81379 Muenchen
Germany

(No handle on envelope, please!)

Introduction to tracking with AHX

This quick overview presumes you already know some of the basics of music tracking with other, more widespread trackers such as **ProTracker** or **SoundTracker**. If you aren't sure, the relevant guides for those programs may be of some help to you.

You will find further information about what the various gadgets throughout the program do in the following reference sections of this manual.

Making music

Any piece of music written with AHX is built up from **positions** (as opposed to **patterns** in **ProTracker** and friends).

Each position is built up from four **tracks**, one for each of the Amiga's audio channels. Unlike ProTracker clones, where the tracks are locked together in patterns, you can re-use **individual tracks** in different combinations, which saves you from having to duplicate note data in extra patterns.

A track is up to 64 lines long. If you need shorter tracks, you can set the **TRL** (track length) field or use the **position break** effect command, **$D** (covered in the **effect commands** section).

A track is built up like this:

Step   Note   Instrument   Effect cmd.
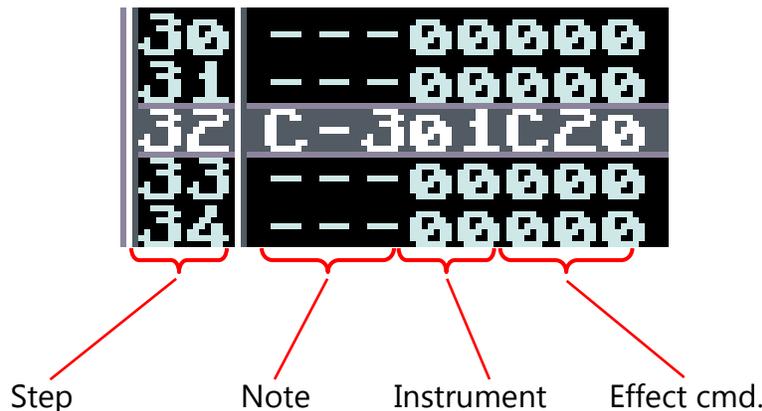
C-3 is the note being played at track step **32**. **01** is the instrument number, and the last three digits are for the effect command; in this case, **set volume** to **$20**. **C** is the command, **20** is its value).

The position table

AHX holds a table with information about the sequence in which the tracks will be played, as well as which track you want to assign to which of the four audio channel of the Amiga. One of the handy (and cool!) features of this table is that you can transpose entire tracks up and down without having to rewrite them as new tracks.

This table contains columns for you to enter the four tracks for the current position, and also columns for you to set how many semitones you'd like to transpose each track, if at all.

The **POS** field (top left of the screen) shows the current position in the table, as does the **dark grey line** highlighting the current table row.

The **LEN** field defines the total number of rows in the position table, and in essence, the length of your song. A song is built up from up to **256** different tracks, 4 of them being played at up to **999** different positions.

The position table is used like so:

The instrument editor

The **instrument editor** is reached by pressing the **INS** button on the main screen, or by pressing **Esc** on your keyboard. Here's the overview:
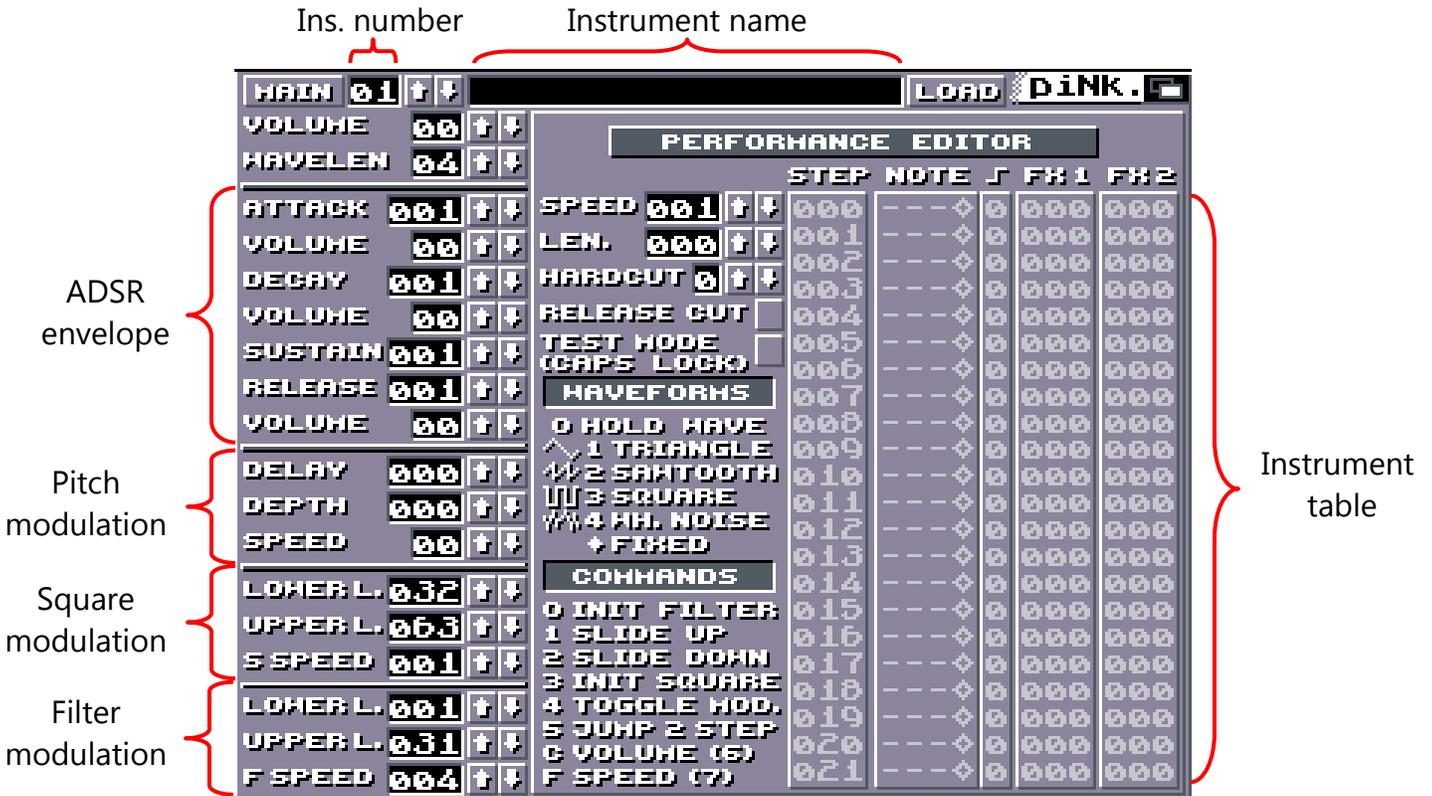
Most of the gadgets here are reasonably self-explanatory, although detailed descriptions of each one can be found in the following section of this manual.

Basically, notes are entered in the **instrument table** in order to produce sounds. Using the flexibility of the multiple rows, you can create arpeggio effects, sweeping filters and square waves, drum sounds, and much more. When creating an instrument from scratch, remember to turn up the volumes of the envelope and set waveforms for your notes in the table, or you won't hear anything.

The column of diamonds shows whether a note is **fixed** in pitch or not. Normally, the notes in the instrument table are transposed to whatever pitch when you actually play the instrument in a track or by using the keyboard. Fixed mode makes the note **stay** at the pitch you specify in the instrument table and prevents it from being transposed. Use this to create combined kick+bass instruments, for example.

Check out the included sample instruments, and try modifying the parameters to get an idea of how it all works.

Screen reference

This section is a quick reference to AHX's screens, simply explaining what each element does.

Main screen

All of the fields with numeric values have **up and down arrow gadgets** next to them, which increase and decrease the value respectively. Holding down the right mouse button whilst adjusting the values with the arrows will speed up the counting a bit.

PLAY

Will play the song from the current position in the song.

STOP

Will stop playback of songs and positions, recording, and will turn edit mode off.

INS

Switches to the **instrument editor**. The number next to this button shows the currently selected instrument. Clicking on the up/down arrow gadgets will cycle through the instruments. You can use up to **63**, or hex **$3F** instruments in a song.

POS

The current position being displayed/played.

LEN

The length of the song in positions.

RES

The position where AHX loops back to after reaching the end of the song.

TRL

The length of the tracks in steps.

SS

Number of subsongs (excluding the main song).

SSN

Number of subsong to edit/play.

SSP

Position of the subsong numbered SSN.

Speaker symbols

The small gadgets labeled with **speaker symbols** above each track represent the audio channels of your Amiga. When the gadget is depressed, that channel is on. When raised, the channel is muted and the notes in that track will not be played.

*You can also use **Alt+Z**, **Alt+X**, **Alt+C**, and **Alt+V** to mute and unmute the four channels.*

FIXME: Speed control (is it really? If not, what the hell is it for?)

This small single-digit field above the step column of the track editor increases and decreases the speed of the song.

Instrument editor

Just like the main screen, the **instrument editor** contains the same up and down arrow gadgets which, again, can be sped up using the right mouse button.

MAIN

Goes back to the **main screen**.

Instrument number (next to MAIN button)

Shows current instrument. Use the adjacent arrow gadgets to cycle through instruments.

Instrument name

A field that lets you set the instrument's name – this lets you describe the instrument so that you know what it is when you're cycling through all the instruments. Some composers use this field to write a long message, by typing each line as instrument names.

VOLUME

Appears four times throughout the screen. The topmost one, next to the **MAIN** button, controls the initial master volume of the instrument. The other three control the volumes of the relevant stages of the **ADSR envelope**.

ATTACK, DECAY, SUSTAIN and RELEASE

These fields determine the lengths of each of the four stages of the **ADSR envelope** of the instrument.

WAVELENGTH

Controls the wavelength of the instrument. The longer the wavelength, the lower the pitch of the instrument, and vice versa.

DELAY, DEPTH and SPEED

Controls pitch modulation of the instrument. **DELAY** determines how long (in steps) before modulation begins. **DEPTH** determines the intensity of the modulation (how much the pitch is changed), and **SPEED** controls how fast the pitch modulates between high and low.

LOWER L. and UPPER L.

Sets the lower and upper limits of **square** and **filter** modulation.

S SPEED and F SPEED

Sets the speed of the **square** and **filter** modulation; i.e. how fast they sweep between the lower and upper limits.

SPEED (next to instrument table)

Determines the speed at which the instrument table is parsed.

LEN

Limits the parsing of the instrument table to the set number of steps, i.e. sets its length.

FIXME: HARDCUT

Don't fucking know.

FIXME: RELEASE CUT

Don't fucking know!!!

TEST MODE

Allows you to try out your instrument by using the **piano keyboard** mappings (see the keyboard reference section), and consequently prevents the keyboard from modifying your instrument table.

Effect commands (tracks)

This section details the effects available for use in the track editor. Effect commands on AHX should more-or-less be compatible with other ProTracker-like trackers.

0	-	None / Position jump (high digit)	**8**	-	External timing
1	-	Portamento up	**9**	-	Set square relation
2	-	Portamento down	**A**	-	Volume slide
3	-	Tone portamento	**B**	-	Position jump
4	-	Set / Override. filter	**C**	-	Set volume
5	-	Tone portamento + Volume slide	**D**	-	Position break
6	*-*	*Not used*	**E**	-	Misc. commands
7	*-*	*Not used*	**F**	-	Set speed

Cmd 0. Position jump (high digit) [HiPosJmp:$0-$9]

Usage: *$0 + position jump (high digit)*

Jumping to e.g. position **123** (decimal) must be done using **Cmd $0 + $01** and **Cmd $B + $23**. Value will be reset after jumping to the desired position with $B.

Example:
`---00001`   -   **0** is the command, **1** adds 100 to the following command's position.
`---00B23`   -   **B** is the command, **23** *now* stands for 123.

Cmd 1. Portamento up [Speed:$00-$FF]

Usage: *$1 + portamento speed*

Slides the sample pitch up. You can **NOT** slide higher than **B-5**! (period 113)

Example:
`C-300103`   -   **1** is the command, **3** is the portamento speed.

NOTE: The portamento effect depends upon the speed of the song.
If the song speed changes, the speed of the portamento effect will also change.
Bearing this in mind, you may need to adjust the slide rate to make it sound correct.

Cmd 2. Portamento down [Speed:$00-FF]

Usage: *$2 + portamento speed*

Just like command 1, except that it slides the pitch down instead (adds to the period).

You can **NOT** slide lower than **C-1**! (period 856)

Example:

C-300203   -   **2** is the command, **3** is the portamento speed.

Cmd 3. Tone portamento [Speed:$00-$FF]

Usage: *Destination note + $3 + slide speed*

WON'T WORK IF THE TRACK IS TRANSPOSED!

This command will automatically slide from the old note to the new. You don't have to worry about which direction to slide, you need only set the slide speed. To keep on sliding, just select the command **$3 + 00**.

Example:

A-200000   -   First play a note.
C-300305   -   **C-3** is the note to slide to, **3** the command, and **5** the speed.

Cmd 4. Set / Override filter [$01-$3F,$41-$7F]

Usage: *$4 + filter / filter+$40*

Values $01-$3F will override filter commands in the instrument table and set the filter when Cmd $0 is reached in the instrument table. Add $40 to the filter value, and the filter is set immediately.

$01-$1F	Low-pass filter (**$01** = heavily filtered, **$1F** = slightly filtered)
$20	No filter (standard value)
$21-$3F	High-pass filter (**$21** = slightly filtered, **$3F** = heavily filtered)

Example:

C-300405   –   **4** is the command, **5** is filter **$05** (when Cmd 0 is reached in ins. table).
C-300443   –   **4** is the command, **43** is filter **$03** and is set immediately.

Cmd 5. Tone portamento + Volume slide [Speed:$0-$F/$0-$F]

Usage: $5 + upspeed + downspeed

This command will continue the current tone portamento and slide the volume at the same time. Stolen from NT2.0.

Example:

C-300503   -   **3** is the speed to slide the volume down.
C-300540   -   **4** is the speed to slide it up.

Cmd 8. External timing [User:$01-$FF]

Usage: $8 + external timing data

This command will set the external timing byte in replayer's public-mem section. Refer to replayer's documentation for further information/usage.

Example:

---00827   -   **27** is the value set for external timing.

Cmd 9. Set square relation [SR:$01-$3F]

Usage: $9 + square relation

This command will set the square wave width (set in the instrument table using **Cmd $3xx**) immediately and will override $3xx Cmds appearing in the instrument table.

Example:

C-300920   -   **20** is the square relation
---00918   -   **18** is the square relation
---00910   -   **10** is the square relation

Cmd A. Volume slide [Speed:$0-$F/$0-$F]

Usage: *$A + upspeed + downspeed*

Example:
`C-300A05`   -   **5** is the speed to slide the volume down.
`C-300A40`   -   **4** is the speed to slide it up.

>   ***NOTE:*** *The slide will be called as many times as the speed of the song.*
> *The slower the song, the more the volume will be changed on each note.*

Cmd B. Position jump [Pos:$00-$99]

Usage: *$B + position to continue at*

Example:
`C-300B00`   -   **B** is the command, **0** is the position to restart the song at.

This command will also perform a position break (see **Cmd D** below). You have to enter the position in **decimal** (**NOT** in hex as in ProTracker).

>   ***NOTE:*** *Use Cmd **$0** to jump to positions >100.*

Cmd C. Set volume [Volume:various]

Usage: *$C+$00-$40: Individual note volume*
               $C+$50-$90: Master volume (affects all channels)
               $C+$A0-$E0: Current channel master volume

This familiar command will set the current volume to your preference. The highest volume is **$40**. All volumes are represented in **hex** (programmers do it in hex, you know!).

Example:
`C-300C10`   -   **C** is the command, **10** is the volume (16 decimal).

Cmd D. Position break [Step-pos:00-63, decimal]

Usage: *$D + step-position*

This command just jumps to the next song position, and continues play from the step you specify.

Example:

`C-300D00`   -   Jump to the next song position and continue play from step **00**, *or...*
`C-300D32`   -   Jump to the next song position and continue play from step **32** instead.

E-commands

The E command has been altered to contain multiple sub-commands.

E0	-	*Not used*		E8	-	*Not used*
E1	-	Fine pitch bend up		E9	-	*Not used*
E2	-	Fine pitch bend down		EA	-	Fine volume slide up
E3	-	*Not used*		EB	-	Fine volume slide down
E4	-	Vibrato control		EC	-	Note cut
E5	-	*Not used*		ED	-	Note delay
E6	-	*Not used*		EE	-	*Not used*
E7	-	*Not used*		EF	-	*Not used*

Cmd E1. Fine pitch bend up [Range:$0-$F]

Usage: *$E1 + value*

This command works just like the normal portamento up, except that it only slides up once. It does not continue sliding during the length of the note. Great for creating chorus effects!

Example:

`C-300E11`   -   Slide up **1** at the beginning of the note.

Cmd E2. Fine pitch bend down [Range:$0-$F]

Usage: *$E2 + value*

This command works just like the normal portamento down, except that it only slides down once. It does not continue sliding during the length of the note.

Example:
`C-300E26`   -   Slide down **6** at the beginning of the note.

Cmd E4. Vibrato control [Value:$0-$F]

Usage: *$E4 + vibrato depth.*

This command will set/override the vibrato depth value set in the current instrument's performance list!

Example:
`C-301000`   -   Play a C-3 note with instrument **1**.
`C-300E40`   -   Stop vibrato.
`C-300E48`   -   Medium vibrato.

Cmd EA. Fine volume slide up [Range:$0-$F]

Usage: *$EA + value*

This command works just like the normal volume slide up, except that it only slides up once. It does not continue sliding during the length of the note.

Example:
`C-301EA3`   -   Slide volume up **3** at the beginning of the note.

Cmd EB. Fine volume slide down [Range:$0-$F]

Usage: *$EB + value*

This command works just like the normal volume slide down, except that it only slides down once. It does not continue sliding during the length of the note.

Example:
`C-301EB6`   -   Slide volume down **6** at the beginning of the note.

Cmd EC. Note cut [Value:$0-$F]

Usage: *$EC + Tick to Cut note at.*

This command will cut the note at the selected tick, creating extremely short notes.

Example:
`C-300F06` -   Set speed to **6**.
`C-300EC3` -   Cut at tick **3** out of 6.

Note that the note is not really cut, the volume is just turned down.

Cmd ED. Note delay [Value:$1-$F]

Usage: *$ED + ticks to delay note.*

This command will delay the note to the selected tick.

Example:
`C-300F06` -   Set speed to **6**.
`C-300ED3` -   Play note at tick **3** out of 6.

Cmd F. Set speed [Speed:$00-$FF]

Usage: *$F + speed*

This command will set the speed of the song.

VBlank	Range **01-FF** - Normal timing
CIA	Range **01-FF** - Normal timing
Both	Range **00** - STOP song.

Effect commands (instrument table)

Effect commands on AHX v2.3d should be compatible with older versions of AHX.

0	-	Init filter
1	-	Portamento up
2	-	Portamento down
3	-	Set square relation
4	-	Toggle modulation (filter + square)
5	-	Jump to step
6	-	Volume (C)
7	-	Speed (F)

Cmd 0. Init filter [Filter:$01-$3F]

Usage: *$0 + filter*

Sets the filter.

$01-$1F	Low-pass filter (**$01** = heavily filtered, **$1F** = slightly filtered)
$20	No filter (standard value)
$21-$3F	High-pass filter (**$21** = slightly filtered, **$3F** = heavily filtered)

Cmd 1. Portamento up [Speed:$00-$FF]

Usage: *$1 + portamento speed*

Slides the sample pitch up. You can **NOT** slide higher than **B-5**! (period 113)

Example:
C-3◊103000   -   **1** is the command, **3** is the portamento speed.

NOTE: The portamento effect depends upon the speed of the song.
If the song speed changes, the speed of the portamento effect will also change.
Bearing this in mind, you may need to adjust the slide rate to make it sound correct.

Cmd 2. Portamento down [Speed:$00-FF]

Usage: *$2 + portamento speed*

Just like command 1, except that this one slides the pitch down instead (adds to the period).

You can **NOT** slide lower than **C-1**! (period 856)

Example:
C-3◊203000   -   **2** is the command, **3** is the portamento speed.

Cmd 3. Set square relation [SR:$01-$3F]

Usage: *$3 + square relation*

Sets the square wave width. Values depend on the wavelength of the instrument.
Don't use values lower than:

Wavelength	Range
$80	$01-$3F
$40	$02-$3F
$20	$04-$3F
$10	$08-$3F
$08	$10-$3F
$04	$20-$3F

If a new wavelength is set, only the lower square limit will be corrected.

Example:
C-3◊308000   -   **3** is the command, **8** is the square relation.

Cmd 4. Toggle modulation (filter + square) [$00,$1/$f,$1/$f]

Usage: $4+toggle filter modulation/toggle square modulation

The upper nibble will toggle filter modulation, the lower nibble will toggle square modulation:

$1	Positive start direction
$F	Negative start direction

Cmd $4 + $00 is kept compatible with older versions of AHX, and is the same as **Cmd $4 + $01** (toggle square modulation with positive start direction).

NOTE: You can also set the start direction value to anything else, in which case the specific direction will be ignored.

Cmd 5. Position jump [Pos:$00-$ff]

Usage:$5 + position to loop to

Values for this command must be written in **hex**.

Example:

`---◊501000`   -   **5** is the command, **1** is the step to jump to.

Cmd C. Set volume [Volume:see below]

Usage: $C+$00-$40: Individual note volume
* $C+$50-$90: Master volume for the instrument*
* $C+$A0-$E0: Current channel master volume*

Similar to **Cmd C** for song tracks. Again, the highest volume is **$40**, and volumes are represented in **hex**.

Example:

`C-3◊C10000`   -   **C** is the command, **10** is the volume.

Cmd F. Set speed [Speed:$00-$FF]

Usage: *$F + pList-speed*

This command will set the speed of the instrument table.

VBlank	Range **01-FF** - Normal timing
CIA	Range **01-FF** - Normal timing
Both	Range **00** - STOP instrument table – however, it's senseless to use this. Just set the instrument table length.

Keyboard reference

Piano keyboard

You can enter notes into AHX and jam with an instrument by using the Amiga's keyboard. The following diagrams show how the keys on your computer keyboard correspond to piano keyboard notes (US keyboard layout).

Upper octave

2	3		5	6	7		9	0		=	
Q	W	E	R	T	Y	U	I	O	P	[	]

Lower octave

	S	D		G	H	J		L	;	
Z	X	C	V	B	N	M	,	.	/
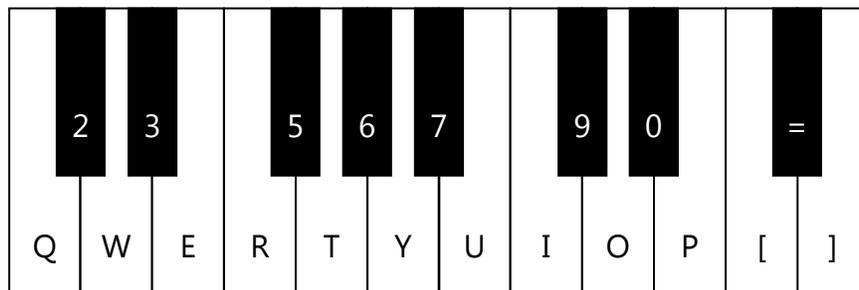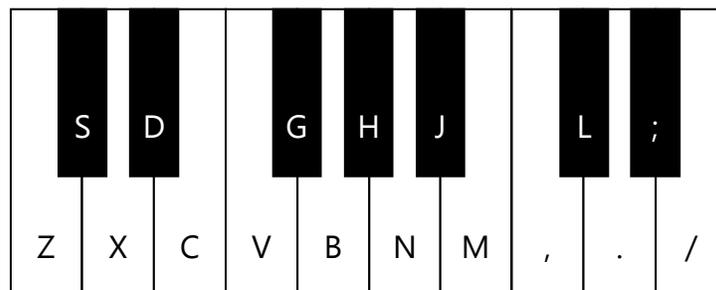
You can switch between higher and lower octaves using the F-keys as follows:

F1	Lowest octave (C-1 to G-3)
F2	Second octave (C-2 to G-4)
F3	Third octave (C-3 to G-5)
F4	Fourth octave (C-4 to B-5)
F5	Highest octave (C-5 to B-5)

Keyboard commands

The keymap on AHX is a standard US layout. Remember to always use the **left** Shift and Alt keys as modifiers unless otherwise stated, as the right ones are used for other things.

Moving through the song

F6	Go to step 0 (or beginning of track).
F7	Go to step 16 (or ¼ of the way through track).
F8	Go to step 32 (or ½ of the way through track).
F9	Go to step 48 (or ¾ of the way through track).
F10	Go to step 63 (or end of track).
Amiga+Return	Update track display according to the tracks shown on the current line of the position editor.
Alt+RgtArrow	Increase currently selected track number.
Alt+LftArrow	Decrease currently selected track number.
Ctrl+UpArrow	Move up a line in the position editor.
Ctrl+DnArrow	Move down a line in the position editor.

Cutting, copying and pasting

Ctrl+B	Mark block.
Ctrl+C	Copy block to buffer.
Ctrl+P	Paste block-buffer to track.
Ctrl+I	Insert block-buffer to track; pushes notes down.
Shift+F3	Cut track to buffer.
Shift+F4	Copy track to buffer.
Shift+F5	Paste track-buffer to track.
Ctrl+F3	Cut commands to buffer.
Ctrl+F4	Copy commands to buffer.
Ctrl+F5	Paste commands in buffer to track.
Alt+F3	Cut instrument to buffer.
Alt+F4	Copy instrument to buffer.
Alt+F5	Paste instrument from buffer.

Editing

Space	Toggle edit mode on/off.
Ctrl+M	Toggle multi-keyboard mode on/off. When enabled, the cursor moves to the next track after you play a note, allowing you to play chords.
Ctrl+0-9	Set how many rows the cursor will jump down every time you insert a note, aka. "editskip" or "add". *Also works in the instrument table!*
Tab	Move cursor to next track.
Shift+Tab	Move cursor to previous track.
Ctrl+Tab	Move cursor to note column of current track.
Alt+Tab	Move cursor to note column of next track.
Shift+Alt+Tab	Move cursor to note column of previous track.
Del	Delete note under cursor.
Alt+Del	Delete command only.
Shift+Del	Delete both note and command.
Shift+Return	Insert blank note at cursor position and move the others down. *Notes beyond track step 63 will be lost!*
Shift+BkSpc	Delete note above cursor position and move the others up. *You can **NOT** do this if you're at track step 0!*
Ctrl+Return	Insert blank command at cursor position and move the others down. *Commands beyond track step 63 will be lost!*
Ctrl+BckSpc	Delete command above cursor position and move the others up. *You can **NOT** do this if you're at track step 0!*
Alt+\	Copy command above cursor to current cursor position.
Alt+=	Copy command above cursor to current cursor position and increase its value by one.
Alt+−	Copy command above cursor to current cursor position and decrease its value by one.
Ctrl+K	Zap entire track.
Ctrl+T	Swap tracks.

Instruments

NumPad0-9	Choose instruments (changes right digit; i.e. units).
NPEnter+NP0-9	Choose instruments (changes left digit; i.e. tens).
NP. (period)	Zap current instrument; a confirmation requester is shown.
CapsLock	Toggles test mode in instrument editor.

Playback control

RAlt	Play song.
RAmiga	Play tracks in the current song position.
RShift	Begin recording; records to tracks in the current song position.
Alt+Z	Mute/unmute channel 1.
Alt+X	Mute/unmute channel 2.
Alt+C	Mute/unmute channel 3.
Alt+V	Mute/unmute channel 4.
\ (left of Z)	Kills Amiga audio DMA and zeroes all volumes. The presence of this key and/or its position on your Amiga's keyboard may vary. My US keyboard has a blank key to the left of Z which performs this command. Most of the time, the key is a \ or has < and > symbols.

Miscellaneous commands

Esc	Toggle between the main screen and the instrument editor. Also exits from on-screen help.
Help	Open the on-screen help box. This updated manual is based on what you find in here. :)
Shift+Help	Open AmigaGuide help on your Workbench screen. However, AHX doesn't actually come with an AmigaGuide document, so this is of no use.
~ (tilde)	Switch to your Workbench screen.
Amiga+Q	Quit AHX; confirmation requester is shown.

Closing words

This is the end of the manual. Here are some of my own words of thanks, and other such wastes of paper.

Contact me

Do you have any comments, suggestions, potential corrections, additions, words of praise, or flames to send my way? Here's some of the ways in which you can get in touch with me.

- Email – daleyo@gmail.com
- WWW – http://lavaburn.com
- MSN Messenger – paradox8@hotmail.co.uk
- IRC - #milkytracker on EsperNet

Thanks go out to

- **Dexter** and **Pink** of **Abyss** – Amiga music creation has never been this much fun. :) AHX is simply amazing, and your hard work is received with endless amounts of appreciation. Thankyou!
- **Deltafire** and **syphus** – This just wasn't possible at all without you guys' help. You've been ace. Cheers, lads! :)
- **Xeron**; author of **HivelyTracker** – the nice, clear documentation for your tracker was exactly what I needed to help clarify numerous sections of this manual. Many thanks for all your hard work, and thanks for bringing the stunning HivelyTracker into the world! Long live the AHX format.

Greetz

- Regulars of the **#milkytracker** channel on EsperNet – **pailes**, **raina**, **Kmuland**, **Strobe**, **syphus**, **Deltafire**, **Blokey**, **setrodox**, **m0d**, **ogge** and anyone else who happens to drop by there from time to time.
- Friends – **Ricky aka. Wicky**, **Nick**, **Matt**, and **Timo aka. iLLER**.