

thx-play

COLLABORATORS

	<i>TITLE :</i> thx-play		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		May 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	thx-play	1
1.1	thx-play.doc	1
1.2	thx-play.m/--overview--	1
1.3	thx-play.m/thxFree	2
1.4	thx-play.m/thxGetNumSongs	2
1.5	thx-play.m/thxGetSyncByte	3
1.6	thx-play.m/thxGetVolume	3
1.7	thx-play.m/thxInit	4
1.8	thx-play.m/thxNoteFX	4
1.9	thx-play.m/thxPause	5
1.10	thx-play.m/thxPlay	5
1.11	thx-play.m/thxPlayNote	5
1.12	thx-play.m/thxSetSong	6
1.13	thx-play.m/thxSetVolume	7
1.14	thx-play.m/thxSignalEnd	7
1.15	thx-play.m/thxSongEnded	8
1.16	thx-play.m/thxStop	8
1.17	thx-play.m/thxStopNote	8
1.18	thx-play.m/thxWind	9

Chapter 1

thx-play

1.1 thx-play.doc

```
--overview--
thxFree()
thxGetNumSongs()
thxGetSyncByte()
thxGetVolume()
thxInit()
thxNoteFX()
thxPause()
thxPlay()
thxPlayNote()
thxSetSong()
thxSetVolume()
thxSignalEnd()
thxSongEnded()
thxStop()
thxStopNote()
thxWind()
```

1.2 thx-play.m/--overview--

PURPOSE

To provide an interface to the THX2 player.

OVERVIEW

THX2 is a 'chip' music tracker by Martin Wodok (Dexter/Abyss). It comes with a rather cumbersome binary replayer, so you may play THX2 songs in your own programs. This module provides an easy and powerful interface to the THX2 player, providing a wide range of functions.

- Basic controls: Init, Free, Play, Stop, Pause, Wind
 - Volume controls: GetVolume, SetVolume
 - MultiSong controls: GetNumSongs, SetSong
 - Sound FX controls: PlayNote, StopNote, NoteFX
 - Misc controls: SignalEnd, SongEnded, SyncByte
-

You need to read a THX song in from disk or INCBIN it, this is not one of the functions in thx-play.m. You should load it into PUBLIC memory, it does not have to be CHIP memory. The module 'tools/file' is handy for loading files from disk.

The music play is, as you would expect, interrupt-driven, and asynchronous. This interface automatically provides fallback support for a VSYNC interrupt driven replayer if it cannot grab a CIA timer.

The interface is 68000 compatible, an optimised version for the 68020 or better is also included as thx-play_020.m

EXAMPLE

More thorough examples are included with the distribution.

```
MODULE 'tools/thx-play','tools/file'
PROC main()
  DEF mod
  IF mod:=loadfile(arg, 0, MEMF_PUBLIC)
    IF thxInit(mod)=0
      thxPlay()
      REPEAT; WaitTOF(); UNTIL CtrlC() OR thxSongEnded()
      thxStop()

      thxFree()
    ENDIF
    freefile(mod)
  ENDIF
ENDPROC
```

1.3 thx-play.m/thxFree

NAME

thxFree -- free resources held by player.

SYNOPSIS

```
void thxFree()
```

FUNCTION

Stops any THX module playing and frees resources used by the player. You can call this whether thxInit() succeeded or not.

SEE ALSO

thxInit()

1.4 thx-play.m/thxGetNumSongs

NAME

thxGetNumSongs -- get number of subsongs.

SYNOPSIS

```
songs = thxGetNumSongs()
```

FUNCTION

Returns the number of subsongs in the module, if any. You can use the `thxSetSong()` function to play one of the subsongs, if that's possible.

RESULT

songs - 0 if there are no subsongs (only the main song), otherwise returns the number of subsongs.

SEE ALSO

```
thxSetSong()
```

1.5 thx-play.m/thxGetSyncByte

NAME

```
thxGetSyncByte -- get sync byte value.
```

SYNOPSIS

```
value = thxGetSyncByte()
```

FUNCTION

Gets the current setting of the 'external timing' byte, which can be set to any byte value at any moment in time during play of the song BY the song itself, using the 8 command in the tracker. This function is here to allow you to mark specific events in the music with the 8 command and a value, then wait until calling `thxSyncByte()` returns that value. The returned value doesn't change until another 8 command in the song changes it.

NOTE

Be very careful not to busy-wait on a new value if there is the possibility the song is paused or not playing.

RESULT

value - current value of the sync byte.

1.6 thx-play.m/thxGetVolume

NAME

```
thxGetVolume -- get master volume.
```

SYNOPSIS

```
volume = thxGetVolume()
```

FUNCTION

Returns the current master volume value. Does not stop play.

RESULT

volume - current volume setting from 0 (silent) to 64 (loudest)

SEE ALSO

thxSetVolume()

1.7 thx-play.m/thxInit

NAME

thxInit -- initialise player and module.

SYNOPSIS

error = thxInit(moduleptr)

FUNCTION

Initialises the player (if needed) and initializes the module. You may also call thxInit(NIL) to initialise the player but not the module. Does not start to play the module until you call thxPlay(). You must call this each time you want to play a different module. The allocations made for the player are made only the first time you call thxInit(), no matter how many modules you want. If allocations fail, they will be automatically freed.

INPUTS

module - pointer to a THX module or NIL

RESULT

error - 0 means all went OK, any other value means something FAILED.

SEE ALSO

thxFree(), thxPlay()

1.8 thx-play.m/thxNoteFX

NAME

thxNoteFX -- perform FX command on user-specified note.

SYNOPSIS

void thxNoteFX(command, parameter)

FUNCTION

Performs an effect command on the user-specified note. You can call this at any time, even before you play the note, if you want the note to start off with an initial effect. See THX Sound System's documentation for the full list of commands and their parameters.

INPUTS

command - the effect command, eg \$C is the Set Volume command.
parameter - the parameter to the command, eg \$40 is full volume.

NOTE

No validation of the command or its parameter is done. Beware feeding wrong

or out of range values. Range for command is \$0 to \$F, parameter is \$00 to \$FF.

SEE ALSO
thxPlayNote()

1.9 thx-play.m/thxPause

NAME
thxPause -- pause play of a song.

SYNOPSIS
void thxPause()

FUNCTION
Pauses the playing module. Call thxPlay() to continue play again.

SEE ALSO
thxPlay()

1.10 thx-play.m/thxPlay

NAME
thxPlay -- start playing the song.

SYNOPSIS
void thxPlay()

FUNCTION
Starts playing the module. If the module has just been initialised or stopped, or the subsong has just been changed, then play will start at the beginning of the song/subsong. Otherwise, it will continue from where it was paused.

SEE ALSO
thxStop(), thxPause()

1.11 thx-play.m/thxPlayNote

NAME
thxPlayNote -- start playing a user-specified note.

SYNOPSIS
void thxPlayNote(instrument, note, channel)

FUNCTION
Plays one of the instruments in the THX module at a particular note on a particular channel. It is up to you to ensure that the channel you play the note on is empty and so will not interfere with the

note being played. This function is to allow you to play your own notes during THX play, for example as part of a game as sound effects. The note played is subject to the same conditions as the song itself, such as the global volume control. In addition, you can apply 'FX' commands to the note. In effect, what is happening when you call `thxPlayNote()` is that the 'track data' for the chosen channel being played is overwritten (not the module itself, just the data output). It is overwritten on the first line by your specified instrument with note and FX, then on consecutive lines by the 'blank' note and instrument. This 'overwriting' stops only when you call `thxStopNote()`, or stop the module naturally.

INPUTS

`instrument` - an instrument from the song, from 1-63. You should know which instrument you want to play!
`note` - The halfnote (pitch) at which the instrument is to be played, from 1 (C-1) to 60 (B-5).
`channel` - The channel on which the note is played, from 0 to 3.

EXAMPLE

`thxPlayNote(12, 8, 2)` is equivalent to this in THX Sound System's tracker view:

```
---00000 | ---00000 | G-112000 | ---00000
---00000 | ---00000 | ---00000 | ---00000
---00000 | ---00000 | ---00000 | ---00000
[...]
```

SEE ALSO

`thxStopNote()`, `thxNoteFX()`

1.12 thx-play.m/thxSetSong

NAME

`thxSetSong -- set song to be played.`

SYNOPSIS

`thxSetSong(song)`

FUNCTION

Sets which song to play, if a module contains more than one song. Most modules only contain one song, but some modules contain sub-songs as well as the main one. You can use this function to specify which one should be played. If you call this function and there is already a song playing, it will be stopped first.

INPUTS

`song` - 0 to set the main song to be played, any other number will change to that subsong, if it exists. Otherwise, no change will be made (other than the stoppage).

NOTE

It is up to you to start playing the module again.

SEE ALSO

```
thxGetNumSongs()
```

1.13 thx-play.m/thxSetVolume

NAME

```
thxSetVolume -- set master volume.
```

SYNOPSIS

```
void thxSetVolume(volume)
```

FUNCTION

Sets the master volume. Does not stop play.

INPUTS

volume - from 0 (silent) to 64 (loudest)

NOTE

This function can take up to two frames to take an audible effect. If the song is paused, will not take effect until unpaused.

SEE ALSO

```
thxGetVolume()
```

1.14 thx-play.m/thxSignalEnd

NAME

```
thxSignalEnd -- Signal() when song ends.
```

SYNOPSIS

```
void thxSignalEnd(signals)
```

FUNCTION

Asks THX to send the signals you specify back to you when the song ends. If songend occurs and the signal is sent, it will not be sent again unless you call thxSignalEnd() again to reload the trigger. The signal will also be cancelled if you call thxStop() directly, or indirectly through thxSetSong() or thxFree(). You may call this function at any time, even when the player is not initialised, or while a song is playing. The task you call it from is stored, so don't call it from asynchronous threads that may disappear before songend is reached.

NOTE

The detection of songend is crap (sorry Dexter :^)

INPUTS

signals - a 32bit set of signals, to be sent back to calling task when songend occurs.

EXAMPLE

```
thxSignalEnd(SIGBREAKF_CTRL_C) will send you a CTRL-C when the song ends.
```

SEE ALSO
thxSongEnded(), exec.library/Signal()

1.15 thx-play.m/thxSongEnded

NAME
thxSongEnded -- detect if song has ended.

SYNOPSIS
ended = thxSongEnded()

FUNCTION
Returns nonzero value if the player has detected the end of a song and is now looping.

NOTE
The detection of songend is crap (sorry Dexter :^)

RESULT
value - nonzero if song is now looping, zero otherwise.

SEE ALSO
thxSignalEnd()

1.16 thx-play.m/thxStop

NAME
thxStop -- stop playing a song/module.

SYNOPSIS
void thxStop()

FUNCTION
Stops the module. Can be restarted from the beginning again with thxPlay(). If you call this, you can free the memory used by your thx module without calling thxFree() which has the unpleasant effect of requiring to recalculate all the filters if you call thxInit() again.

SEE ALSO
thxPlay(), thxFree()

1.17 thx-play.m/thxStopNote

NAME
thxStopNote -- stop playing user-specified note.

SYNOPSIS
void thxStopNote()

FUNCTION

Stops anything you started with `thxPlayNote()`. Please be aware that notes which don't fade away on their own will first need to be silenced with `thxNoteFX($C, $00)`, or such

SEE ALSO

`thxPlayNote()`

1.18 thx-play.m/thxWind

NAME

`thxWind` -- wind the song forward or back.

SYNOPSIS

`void thxWind(direction)`

FUNCTION

Advances forward or backwards through the song by a specified number of positions. Please use the value 1 to skip forward and -1 to skip back, for future compatibility.

INPUTS

`direction` - if positive, winds on to the next position.
if negative, winds back to the previous position,
if 0, ignored.

NOTE

Be wary of stepping beyond the end of a song. Also note this function only takes effect only once a frame.