

Krawall Reference Manual

Generated by Doxygen 1.2.16

Mon Jun 10 11:51:08 2002

Contents

1 Krawall File Index	1
2 Krawall File Documentation	1

1 Krawall File Index

1.1 Krawall File List

Here is a list of all files with brief descriptions:

krawall.h	1
---------------------------	---

2 Krawall File Documentation

2.1 krawall.h File Reference

Typedefs

- typedef unsigned int [chandle](#)

Functions

- void [kragInit](#) (int stereo)
Init function.
- void [krapPlay](#) (const Module *m, int mode, int song)
Start music.
- void [krapStop](#) ()
Stop music.
- void [krapCallback](#) (void(*func)(int, int))
Install callback.
- void [krapPause](#) (int sfx)
Pause music.
- void [krapUnpause](#) ()
Unpause music.
- int [krapIsPaused](#) ()
Get Pause status.
- void [krapSetMusicVol](#) (uint vol, int fade)
Set music volume.

- int [kramWorker](#) () LONG_CALL
Worker procedure.
- int [kramGetActiveChannels](#) ()
Get number of currently active channels.
- void [kramQualityMode](#) (int)
Set quality mode.
- [chandle](#) [kramPlay](#) (const Sample *s, int sfx, [chandle](#) c) LONG_CALL
Play a sample.
- [chandle](#) [kramPlayExt](#) (const Sample *s, int sfx, [chandle](#) c, uint freq, uint vol, int pan) LONG_CALL
Play a sample Ext.
- int [kramStop](#) ([chandle](#) c) LONG_CALL
Stop a channel.
- int [kramSetFreq](#) ([chandle](#) c, uint freq) LONG_CALL
Set frequency.
- int [kramSetVol](#) ([chandle](#) c, uint vol) LONG_CALL
Set volume.
- int [kramSetPan](#) ([chandle](#) c, int pan) LONG_CALL
Set panning.
- int [kramSetPos](#) ([chandle](#) c, uint pos) LONG_CALL
Set Position.
- void [kramSetSFXVol](#) (uint vol)
Set SFX volume.
- void [kradInterrupt](#) ()
Directsound Interrupt.

2.1.1 Typedef Documentation

2.1.1.1 typedef unsigned int chandle

2.1.2 Function Documentation

2.1.2.1 void kradInterrupt ()

This function resets the DMA and must be tied to the Timer1-IRQ.

2.1.2.2 void kragInit (int *stereo*)

Call this function once at startup.

Parameters:

stereo Whether Krawall should operate stereo or not

2.1.2.3 int kramGetActiveChannels ()

Returns number of currently active channels.

Returns:

Number of currently active channels

2.1.2.4 [chandle](#) kramPlay (const Sample * *s*, int *sfx*, [chandle](#) *c*)

Plays a sample with it's C2 (neutral) frequency.

Parameters:

s Pointer to sample

sfx Whether sample to play is an SFX

c Old handle, will be recycled if given

See also:

[kramPlayExt\(\)](#)

Returns:

Channel handle

2.1.2.5 [chandle](#) kramPlayExt (const Sample * *s*, int *sfx*, [chandle](#) *c*, uint *freq*, uint *vol*, int *pan*)

Just like kramPlay, but all of the attribs can be specified.

Parameters:

s Pointer to sample

sfx Whether sample to play is an SFX

c Old handle, will be recycled if given

freq Frequency in hertz to play sample at

vol Volume to play sample with (0..64)

pan Panning to play sample with (-64..64)

See also:

[kramPlay\(\)](#)

Returns:

Channel handle

2.1.2.6 void kramQualityMode (int)

This sets the quality mode of the mixing routines. KRAM_QM_NORMAL is the default, KRAM_QM_MARKED only plays the marked samples (see docs) in HQ and KRAM_QM_HQ plays everything in HQ.

2.1.2.7 int kramSetFreq (chandle *c*, uint *freq*)

Sets frequency of an active channel. Note that if the channel has already stopped this call will not do anything and return false.

Parameters:

c Channel handle
freq Frequency in hertz

Returns:

true if successful

2.1.2.8 int kramSetPan (chandle *c*, int *pan*)

Sets the panning-position of an active channel. Note that if the channel has already stopped this call will not do anything and return false.

Parameters:

c Channel handle
pan Panning (-64..0..64), KRAM_SP_LEFT, KRAM_SP_RIGHT, KRAM_SP_CENTER

Returns:

true if successful

2.1.2.9 int kramSetPos (chandle *c*, uint *pos*)

Sets the sample-position of an active channel. Note that if the channel has already stopped this call will not do anything and return false.

Parameters:

c Channel handle
pos Sample offset to set

Returns:

true if successful

2.1.2.10 void kramSetSFXVol (uint *vol*)

Sets the volume of all active and future sfx.

Parameters:

vol Volume (0..128)

2.1.2.11 int kramSetVol ([chandle](#) *c*, uint *vol*)

Sets volume of an active channel. Note that if the channel has already stopped this call will not do anything and return false.

Parameters:

c Channel handle
vol Volume (0..64)

Returns:

true if successful

2.1.2.12 int kramStop ([chandle](#) *c*)

Stops playback of a channel. Note that if the channel has already stopped this call will not do anything and return false.

Parameters:

c Channel handle

Returns:

true if successful

2.1.2.13 int kramWorker ()

This is where the actual work is done, you **MUST** call this once per frame after kraInit() to get sound

See also:

[kragInit\(\)](#)

Returns:

True if actual work has been done

2.1.2.14 void krapCallback (void(* *func*)(int, int))

Installs a callback. The callback should return as quickly as possible. When the callback gets called the first numeric parameter describes the event, the second numeric parameter (if any) is the parameter to the event. The events are as following:

- KRAP_CB_FADE Destination volume has been reached
- KRAP_CB_DONE Module is done (also when KRAP_MODE_LOOP)
- KRAP_CB_MARK Mark-Effect Zxx (xx in param 2)
- KRAP_CB_SONG Song-boundary hit (+++-Marker)
- KRAP_CB_JDONE Jingle is done

See also:

[krapSetMusicVol](#)

2.1.2.15 int krapIsPaused ()

Returns whether playback is currently paused or not

Returns:

True if paused

See also:

[krapPause\(\)](#) , [krapUnpause\(\)](#)

2.1.2.16 void krapPause (int *sfx*)

Pauses all currently active channels. You still can play SFX's. The paused channels will be frozen until [krapUnpause\(\)](#) gets called.

Parameters:

sfx If true pause sfx as well; if false pause music only

See also:

[krapUnpause\(\)](#)

2.1.2.17 void krapPlay (const Module * *m*, int *mode*, int *song*)

Parameters:

m Pointer to module

mode is one or more of:

- KRAP_MODE_LOOP Loop module
- KRAP_MODE_SONG Enable song-mode
- KRAP_MODE_JINGLE Play module as jingle

Parameters:

song Song of module to play

See also:

[krapStop\(\)](#)

2.1.2.18 void krapSetMusicVol (uint *vol*, int *fade*)

You can either set the music volume immediately or fade slowly to the specified volume. The fadespeed depends on the speed of the currently active module. If module is paused then volume is always set immediately. If a callback is installed it will get triggered when fading is done.

Parameters:

vol Music volume (0..128)

fade If true fade, if false set immediately

See also:

[kramSetSFXVol](#) , [krapCallback](#)

2.1.2.19 void krapStop ()

Immediately stops playback of music.

See also:

[krapPlay\(\)](#)

2.1.2.20 void krapUnpause ()

Reactivates all channels that have been paused with [krapPause\(\)](#)

See also:

[krapPause\(\)](#)

Index

chandle
 krawall.h, [2](#)

kradInterrupt
 krawall.h, [2](#)

kragInit
 krawall.h, [2](#)

kramGetActiveChannels
 krawall.h, [3](#)

kramPlay
 krawall.h, [3](#)

kramPlayExt
 krawall.h, [3](#)

kramQualityMode
 krawall.h, [3](#)

kramSetFreq
 krawall.h, [4](#)

kramSetPan
 krawall.h, [4](#)

kramSetPos
 krawall.h, [4](#)

kramSetSFXVol
 krawall.h, [4](#)

kramSetVol
 krawall.h, [4](#)

kramStop
 krawall.h, [5](#)

kramWorker
 krawall.h, [5](#)

krapCallback
 krawall.h, [5](#)

krapIsPaused
 krawall.h, [5](#)

krapPause
 krawall.h, [6](#)

krapPlay
 krawall.h, [6](#)

krapSetMusicVol
 krawall.h, [6](#)

krapStop
 krawall.h, [6](#)

krapUnpause
 krawall.h, [7](#)

krawall.h, [1](#)
 chandle, [2](#)
 kradInterrupt, [2](#)
 kragInit, [2](#)
 kramGetActiveChannels, [3](#)
 kramPlay, [3](#)
 kramPlayExt, [3](#)
 kramQualityMode, [3](#)
 kramSetFreq, [4](#)
 kramSetPan, [4](#)
 kramSetPos, [4](#)
 kramSetSFXVol, [4](#)
 kramSetVol, [4](#)
 kramStop, [5](#)
 kramWorker, [5](#)
 krapCallback, [5](#)
 krapIsPaused, [5](#)
 krapPause, [6](#)
 krapPlay, [6](#)
 krapSetMusicVol, [6](#)
 krapStop, [6](#)
 krapUnpause, [7](#)
