

SID™ Software

The HearHere Toolbox

Copyright © 1989 by SID Products

HearHere is a toolbox of subroutines which makes it easy to add sound recording into applications. All data produced is compatible with the Macintosh toolbox sound manager and can be played through the standard Mac calls (StartSound, SndDoCommand).

HearHere can be used with SID as well as other commercially available serial sound interfaces. The software is written entirely in Think C - no assembly language is needed! Using C alone will give most developers a clear understanding about how to control a sound input device and allow the code to be easily adapted into existing applications.

Subroutine calls allow the recording of sound directly into a RAM buffer or double-buffered onto a hard disk. Capturing the real-time sound directly onto disk makes it possible to record large sound sequences, even on machines that have minimal RAM configurations. Playback can also be performed from a RAM buffer or double-buffered from a hard disk.

Interface Background

Typically in the recording process, there can be a great amount of “dead recording” before the sound begins and after the sound ends. This ends up wasting precious memory resources as well as expanding the playback time with unnecessary silence. This toolbox gives the developer the option of automatically eliminating these dead times and recording only the important sound data.

HearHere defines a front window and a rear window on all sound input. The front window is made up of:

- starting threshold - the sound threshold required before any recording will take place
- front window length - the amount of the front window that is kept as part of the recording after the starting threshold has been satisfied
- maximum sample wait count - the maximum number of samples to wait for before the recording is aborted

All sound levels are numbers between 0 and 255 representing the analog signal amplitude as received from the input device. A level of 128 is centered on the signal and represents pure silence. All thresholds are measured as changes from pure silence. For example, a starting threshold of 20 would require a sound level greater than 147 or less than 109 in order to satisfy the front window. A starting threshold of 0 will force the recording of sound level data without waiting for the front window to be satisfied.

The front window length is used to keep a small part of the front window during recording. Many words, such as “Macintosh” start with a fairly quiet sound (the “M”) and will not satisfy the starting threshold until later in the sound (the “acintosh”). When the front window length is set, that number of samples will be kept before the starting threshold was received. In most cases, the number of samples should be about 1/4 second in duration.

The maximum sample wait count is a way to abort a recording when no response is received within a specified amount of time (as measured in samples).

The rear window that HearHere defines is made up of:

- quiet threshold - the sound threshold that is considered to be silence
- rear length - the number of samples in the rear window

The rear window is satisfied when a rear length amount of quiet samples are received in succession. This should typically be one to two seconds, if a rear window is used. Setting the rear window length to 0 will disable all rear window processing.

The front window and the rear window method of clipping the sound input do not have to be used. Sound level recording can begin as soon as the recording subroutine is called (when the starting threshold is set to 0). Instead of using rear window stopping, the recording can be specified to stop under the following conditions:

- the mouse is moved vertically
- the specified RAM buffer is completely filled
- the specified hard disk is completely filled

Different status values are returned to the application describing the exact way that the recording was stopped.

Subroutine Descriptions

All symbolic references to definitions are included in HearHere.h. All Boolean subroutines return TRUE when the subroutine was successfully completed and return FALSE when additional status and error information is available by calling HHError().

Boolean HHInit(int port)
port - MODEM_PORT or PRINTER_PORT

Function:

Initializes all internal structures of HearHere and programs the serial controller for sound input.

Errors:

INVALID_PORT - port parameter is invalid

NOT_RESPONDING - the hardware device connected to the port is not responding as expected

Boolean HHSample(int *level)
level - returned instantaneous current sound level (0 - 255)

Function:

Reads the current sound level from the attached hardware device and returns immediately. No front window or rear window clipping is done. This routine should not be used to collect sound data for later playback. It is only for obtaining one sound level value.

Errors:

UNKNOWN_STATE - HearHere was not initialized

NOT_RESPONDING - the hardware device connected to the port is not responding as expected

Boolean HHSampleRate(int rate)
rate - sample rate to use in recording and playback
1 - 22K samples per second
2 - 11K samples per second
3 - 7.33K samples per second
4 - 5.5K samples per second

Function:

Sets the sample rate for all future calls. Note: HHInit() did not have to be called before calling this routine. The sample rate is also used within

the playback routines which don't require the serial port to be initialized. Calling HHIInit() will reset the sample rate to the 22K samples per second rate.

Errors:

INVALID_RATE - rate must be 1 - 4

Boolean HHFrontWindow(int startThresh, int frontLength, int maxWait)

startThresh - the starting threshold

frontLength - the front window length

maxWait - the maximum sample wait count (in thousands)

Function:

Sets the front window parameters for all future calls.

Errors:

UNKNOWN_STATE - HearHere was not initialized

INVALID_FRONT - parameter error

Boolean HHRearWindow(int quietThresh, int rearLength)

quietThresh - rear window quiet threshold

rearLength - rear window length (maximum of 32K samples can be waited)

Function:

Sets the rear window parameters for all future calls.

Errors:

UNKNOWN_STATE - HearHere was not initialized

INVALID_REAR - parameter error

Boolean HHRecord(unsigned char *buffer, long bufLen, int diskUse,

Boolean mouseFlag, long *recordLen)

buffer - an allocated RAM buffer to use for recording the sound data (enough for both buffers used with double buffered disk recording if specified)

bufLen - the size of the buffer

diskUse - 0 for RAM recording, otherwise it is the refNum of a previously opened hard disk file

mouseFlag - TRUE if mouse action can interrupt recording, FALSE if mouse action should not interrupt recording

recordLen

>= 0: number of samples recorded

-1: RAM recording halted because buffer was filled

<-1: OSErr returned from file manager during disk recording

Function:

Records sound input into RAM or onto the hard disk following the current settings for the front window, the rear window, and the sample rate. The mouseFlag parameter will allow you to specify whether any mouse movement will stop the recording (note that if mouseFlag is FALSE and mouse movement occurs, interrupts caused by the mouse handler may cause clicking in the recorded data). If recording is being saved to a hard disk, the data will be written to the data fork of the previously opened file. Note: current implementation of recording to disk does not support the front window frontLength or the maxWait parameters. When recording to disk, recording begins as soon as the startThresh is satisfied and ends when the rear window is complete (or the mouse was moved with mouseFlag set to TRUE). When recording to disk, some "skips" in the recording may take place depending on the speed of the disk being used (and the speed of the Macintosh). For best

results, record on the fastest Mac with the fastest disk using the largest amount of RAM possible. Playback will then sound best on any Mac, disk, and buffer size (within reason).

Errors:

UNKNOWN_STATE - HearHere was not initialized

NOT_RESPONDING - the hardware device connected to the port is not responding as expected

TIMEOUT - front window timeout (the maximum sample wait count was met without meeting the front window starting threshold)

INTERRUPT - the mouse was moved or clicked and mouseFlag is TRUE

MEMALLOC_ERROR - the second buffer for sound capture to disk could not be allocated

DISKFAIL - an error occurred writing sound data to disk

Boolean HHDiskPlay(int refNum, unsigned char *buffer, long bufLen)

refNum - the refNum of the disk file that contains the sound level data

buffer - a buffer to use for reading data from the disk

bufLen - the size of the buffer

Function:

Plays the sound level data contained in the disk file recorded from the use of HHDiskRecord recording to disk.

Errors:

DISKFAIL - an error occurred writing sound data to disk

Boolean HHEnd()

Function:

Returns HearHere to the unknown state and releases the serial port.

Errors:

UNKNOWN_STATE - HearHere was not initialized

int HHError()

Function:

Returns the error or status message from the previous HearHere call.