SPIflash_Audio

Add-on Library for SPIflash

Manual

Introduction:

This library is an add-on to the SPIflash library and will not work on its own.

This library adds a simple way to play audio samples from an SPI flash chip. The audio samples must be contained within the SPIflash file system. Audio samples can be added to the flash chips using the FlashUploader tool supplied with the SPIflash library.

EXAMPLE DATASETS USED:

These files can be found in the **/SPIflash_Audio/DataSet** folder.

Full name	Short name	Minimum Flash Chip Size (Mbits)
Audio_Demo.*	AUDIO.SFD	8 Mbits

The specific dataset required by an example sketch it will be noted in the opening comments of that sketch.

IMPORTANT:

This library requires SPIflash v1.30 or higher to work.

Audio output is hardcoded to pin D3 on Arduino Uno and D9 on Arduino Mega.

This library uses both Timer1 and Timer2. This may cause conflicts with other libraries. The timer interrupts are only used while an audio sample is being played.

You can always find the latest version of the library at **http://www.RinkyDinkElectronics.com/**

For version information, please refer to **version.txt**.

FUNCTIONS:

SPIflash_Audio(SPIflash[, activeLED]);

The main class constructor.

Parameters: SPIflash: a reference to an already created SPIflash object
activeLED: **<optional>** pin number for activity LED

Usage: `SPIflash_Audio myAudio(&myFlash, 8); // Create an instance of the SPIflash_Audio class`

Notes: Remember the '&' in front of the SPIflash object name

begin();

Initialize the instance for use.

Parameters: None

Usage: `myAudio.begin(); // Initialize the myAudio object`

play(fileID);

Start playing an audio sample.

Parameters: fileID: ID of the file you want to start playing

Returns: (uint32_t) ERR_NO_ERROR or a general error (see defined literals for the SPIflash library)

Usage: `result = myAudio.play(8); // Attempt to start playing the file with ID #8`

Notes: If a sample is already being played it will be stopped before starting playback of the requested file.
Access to the flash memory chip will be restricted while a sample is playing.

stop();

Stop playing.

Parameters: None

Usage: `myAudio.stop(); // Stop all playback`

isPlaying();

Check if an audio sample is currently being played.

Parameters: None

Returns: (boolean) TRUE if playback is active, otherwise FALSE

Usage: `result = myAudio.isPlaying(); // Check if a sample is being played`

getAudioSamplerate(fileID);

Get the sample rate for an audio sample.

Parameters: fileID: ID of the file containing the audio sample you want the sample rate for

Returns: (uint16_t) sample rate in Hz *or* a general error (see defined literals for the SPIflash library)

Usage: bitrate = myAudio.getAudioSamplerate(14); // Get the sample rate for file ID #14

Notes: This function will return **ERR_FILETYPE_INCORRECT** if you try to get the size of a non-audio file.

getAudioBPS(fileID);

Get the bits per sample for an audio sample.

Parameters: fileID: ID of the file containing the audio sample you want the bits per sample for

Returns: (uint16_t) bits per sample *or* a general error (see defined literals for the SPIflash library)

Usage: bps = myAudio.getAudioBPS(14); // Get the bits per sample for file ID #14

Notes: This function will return **ERR_FILETYPE_INCORRECT** if you try to get the size of a non-audio file.

getAudioChannels(fileID);

Get the number of channels for an audio sample.

Parameters: fileID: ID of the file containing the audio sample you want the number of channels for

Returns: (uint16_t) number of channels *or* a general error (see defined literals for the SPIflash library)

Usage: channels = myAudio.getAudioChannels(14); // Get the number of channels for file ID #14

Notes: This function will return **ERR_FILETYPE_INCORRECT** if you try to get the size of a non-audio file.

getAudioSampleCount(fileID);

Get the number of sample points for an audio sample.

Parameters: fileID: ID of the file containing the audio sample you want the sample count for

Returns: (uint32_t) number of sample points *or* a general error (see defined literals for the SPIflash library)

Usage: count = myAudio.getAudioSampleCount(14); // Get the number of sample points for file ID #14

Notes: This function will return **ERR_FILETYPE_INCORRECT** if you try to get the size of a non-audio file.

getAudioPlaytime(fileID);

Get the playtime for an audio sample.

Parameters: fileID: ID of the file containing the audio sample you want the playtime for

Returns: (float) playtime in seconds *or* a general error (see defined literals for the SPIflash library)

Usage: playtime = myAudio.getAudioPlaytime(14); // Get the playtime for file ID #14

Notes: This function will return **ERR_FILETYPE_INCORRECT** if you try to get the size of a non-audio file.