T2 SOFTWARE

T2 Software, Inc.

LC3 API Documentation v1.5

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Module Documentation

4.1 LC3

Data Structures

- struct LC3EncodeInput_t

  LC3 encoder input PCM data structure.
- struct LC3EncodeOutput_t

  LC3 encoder output data structure.
- struct LC3DecodeInput_t

  LC3 decoder input bit stream structure.
- struct LC3DecodeOutput_t

  LC3 decoder output PCM structure.

Typedefs

- typedef void ∗ **LC3EncoderHandle_t**

  LC3 encoder handle type.
- typedef void ∗ **LC3DecoderHandle_t**

  LC3 decoder handle type.

Enumerations

- enum LC3FrameSizeConfig_t { LC3FrameSize10MsConfig , LC3FrameSize7_5MsConfig , LC3FrameSizeBothConfig }

  Frame Size for configuration.
- enum LC3FrameSize_t { LC3FrameSize10Ms , LC3FrameSize7_5Ms }

  Session Frame Size.
- enum LC3BFI_t { **GoodFrame** , **BadFrame** }

  Bad Frame Indicator (BFI).

Functions

- int32_t OSALCALL LC3Initialize (uint8_t encoderSampleRates, uint8_t decoderSampleRates, LC3FrameSizeConfig_t frameSizeConfig, uint8_t uniqueSessions, uint8_t ∗buffer, uint32_t ∗bufferSize)

  Initializes the LC3 Codec.
- int32_t OSALCALL LC3Deinitialize (void)

  Deinitializes the LC3 Codec.
- LC3EncoderHandle_t OSALCALL LC3EncodeSessionOpen (uint16_t sampleRate, uint8_t bitsPerSample, LC3FrameSize_t frameSize, uint8_t ∗buffer, uint16_t ∗bufferSize, int32_t ∗result)

  Opens and initializes an LC3 Encoder session.
- int32_t OSALCALL LC3EncodeSessionData (LC3EncoderHandle_t encodeHandle, LC3EncodeInput_t ∗encodeInput, LC3EncodeOutput_t ∗encodeOutput)

  Encodes a frame of data using LC3.
- void OSALCALL LC3EncodeSessionClose (LC3EncoderHandle_t encodeHandle)

  Closes an LC3 Encoder session.
- LC3DecoderHandle_t OSALCALL LC3DecodeSessionOpen (uint16_t sampleRate, uint8_t bitsPerSample, LC3FrameSize_t frameSize, uint8_t ∗buffer, uint16_t ∗bufferSize, int32_t ∗result)

  Opens and initializes an LC3 Decoder session.
- int32_t OSALCALL LC3DecodeSessionData (LC3DecoderHandle_t decodeHandle, LC3DecodeInput_t ∗decodeInput, LC3DecodeOutput_t ∗decodeOutput)

  Decodes a frame of data using LC3.
- void OSALCALL LC3DecodeSessionClose (LC3DecoderHandle_t decodeHandle)

  Closes an LC3 Decoder session.
- uint16_t OSALCALL LC3BitstreamBuffersize (uint16_t sampleRate, uint32_t maxBitRate, LC3FrameSize_t frameSize, int32_t ∗result)

  Calculates the buffer size required for bit stream data.
- uint16_t OSALCALL LC3PCMBuffersize (uint16_t sampleRate, uint8_t bitDepth, LC3FrameSize_t frameSize, int32_t ∗result)

  Calculates the buffer size required for PCM data.

LC3 Result Codes

- #define **LC3_ERROR_OFFSET** (-5000)

  Offset for LC3 error values.
- #define **LC3_RESULT_NO_ERROR** (0)

  Success.
- #define **LC3_RESULT_INVALID_PARAMETER** ((LC3_ERROR_OFFSET)-1)

  Error: invalid parameter detected.
- #define **LC3_RESULT_INSUFFICIENT_RESOURCES** ((LC3_ERROR_OFFSET)-2)

  Error: not enough memory available.
- #define **LC3_RESULT_NOT_INITIALIZED** ((LC3_ERROR_OFFSET)-3)

  Error: not initialized before use.
- #define **LC3_RESULT_UNKNOWN_ERROR** ((LC3_ERROR_OFFSET)-4)

  Error: unknown error detected.
- #define **LC3_RESULT_INVALID_BITRATE** ((LC3_ERROR_OFFSET)-5)

  Error: invalid bitrate value.
- #define **LC3_RESULT_INPUT_BUFFER_TOO_SMALL** ((LC3_ERROR_OFFSET)-6)

  Error: passed buffer size too small for input data.
- #define **LC3_RESULT_OUTPUT_BUFFER_TOO_SMALL** ((LC3_ERROR_OFFSET)-7)

Error: passed buffer size too small for output data.
- #define **LC3_RESULT_INVALID_BITSPERSAMPLE** ((LC3_ERROR_OFFSET)-8)

  Error: invalid bits per sample value.
- #define **LC3_RESULT_INVALID_SAMPLERATE** ((LC3_ERROR_OFFSET)-9)

  Error: invalid sample rate value.
- #define **LC3_RESULT_BITERRORCONDITION** ((LC3_ERROR_OFFSET)-10)

  Error: bit error condition detected in bit stream data.
- #define **LC3_RESULT_FEATURE_NOT_SUPPORTED** ((LC3_ERROR_OFFSET)-11)

  Error: feature not supported in this build.
- #define **LC3_RESULT_STILL_IN_USE** ((LC3_ERROR_OFFSET)-12)

  Error: Sessions not closed, buffer still in use.
- #define **LC3_RESULT_ALREADY_INITIALIZED** ((LC3_ERROR_OFFSET)-13)

  Error: Codec already initialized, call LC3Deinitialize first.

LC3 Sample Rate Bits

- #define **LC3_SAMPLE_RATE_NONE** (0x00)

  No sample rate selected.
- #define **LC3_SAMPLE_RATE_8_KHZ** (0x01)

  8 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_16_KHZ** (0x02)

  16 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_24_KHZ** (0x04)

  24 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_32_KHZ** (0x08)

  32 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_441_KHZ** (0x10)

  44.1 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_48_KHZ** (0x20)

  48 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_ALL** (0x1F)

  All sample rates selected.

4.1.1 Detailed Description

4.1.2 Enumeration Type Documentation

4.1.2.1 LC3FrameSizeConfig_t

```
enum LC3FrameSizeConfig_t
```

```
#include <LC3API.h>
```

Frame Size for configuration.

Defines the frame size used for global memory configuration purposes. Select the specific frame size if only one will be used, otherwise select both and the session open call will define which is used for a session.

Enumerator

LC3FrameSize10MsConfig	10 msec frame size.
LC3FrameSize7_5MsConfig	7.5 msec frame size.
LC3FrameSizeBothConfig	7.5 and 10 msec frame size.

4.1.2.2 LC3FrameSize_t

```
enum LC3FrameSize_t
```

```
#include <LC3API.h>
```

Session Frame Size.

Defines the valid frame sizes for an LC3 codec session.

Enumerator

LC3FrameSize10Ms	10 msec frame size.
LC3FrameSize7_5Ms	7.5 msec frame size.

4.1.2.3 LC3BFI_t

```
enum LC3BFI_t
```

```
#include <LC3API.h>
```

Bad Frame Indicator (BFI).

Identifies a bit stream data frame's status as known by the device. A frame known to contain bit errors or be otherwise corrupted should be passed with the BadFrame value when passed to the LC3 decoder. The decoder will detect bit errors internally and apply PLC automatically, but if the caller knows that the frame is in error, such as from information indicated by a Bluetooth controller, then this value is used to skip decoding and immediately execute packet loss concealment.

4.1.3 Function Documentation

4.1.3.1 LC3Initialize()

```
int32_t OSALCALL LC3Initialize (
            uint8_t encoderSampleRates,
            uint8_t decoderSampleRates,
            LC3FrameSizeConfig_t frameSizeConfig,
            uint8_t uniqueSessions,
            uint8_t * buffer,
            uint32_t * bufferSize )
```

```
#include <LC3API.h>
```

Initializes the LC3 Codec.

This function initializes the LC3 codec for the sample rates(s) specified in the encoder/decoder sample rate parameter bit masks. The sample rate bits are defined in LC3SampleRateBits. The user must provide all required encoder/decoder sample rates to this function. If the exact sample rates are unknown, the LC3_SAMPLE_RATE↩
_ALL value should be entered.

frameSizeConfig defines if the codec will support 7.5 msec frames, 10 msec frames or both. If the frame size isn't known in advance, setting this to both allows each session to use either 7.5 or 10 msec but will use more memory.

The uniqueSessions value is the number of unique (sample rates ∗ frame sizes) sessions expected to be used simultaneously; the sample rates of 48k and 44.1k are considered a single sample rate for this calculation. If nothing is known of the expected sessions' sample rates and/or frame sizes, this can be set to the number of simultaneous encode + decode sessions. Setting this to the total number of (unique sample rates ∗ frame sizes) will initialize data tables globally, otherwise they will be initialized as needed when sessions are opened.

When LC3_SAMPLE_RATE_ALL and/or LC3FrameSizeBothConfig are used and the uniqueSessions value is less than the number of (sample rates ∗ frame sizes), the codec allocates the worse case shared memory required based on the uniqueSessions value. The LC3EncodeSessionOpen and/or LC3DecodeSessionOpen call initializes the tables required for each session's sample rate and frame size. In general, the smallest memory footprint will be achieved by defining the sample rate bit fields and a single frame size.

This function can allocate its own memory or take a user-provided buffer. It can also return the amount of memory needed without initializing. These memory options are performed in the following ways:

bufferSize = NULL: Allocates memory needed. The buffer parameter may be NULL.

∗bufferSize < memory required by codec: Sets bufferSize's referenced value to the total memory required and returns LC3_RESULT_INSUFFICIENT_RESOURCES. The caller can then call this function again with a buffer of at least bufferSize's referenced value. A size of 0 will always trigger this condition. The buffer parameter may be NULL.

∗bufferSize >= memory required by codec: Uses the user-provided buffer. bufferSize's referenced value is set to the amount of memory used by the codec. The buffer parameter may not be NULL.

Parameters

`in`	*encoderSampleRates*	Bitfield containing all encoder session sample rates required.
`in`	*decoderSampleRates*	Bitfield containing all decoder session sample rates required.
`in`	*frameSizeConfig*	Frame size of 7.5, 10 msec or both.
`in`	*uniqueSessions*	The number of unique simultaneous sample rate ∗ frame size sessions.
`in`	*buffer*	Pointer to a memory buffer.
`in,out`	*bufferSize*	Size of the memory buffer passed in, number of bytes used returned.

Returns

Zero on success or one of the defined LC3 result codes on error.

4.1.3.2 LC3Deinitialize()

```
int32_t OSALCALL LC3Deinitialize (
            void  )
```

```
#include <LC3API.h>
```

Deinitializes the LC3 Codec.

This function deinitializes the LC3 Codec. All encoder and decoder sessions should be closed before calling this function. Memory passed in the LC3Initialize's buffer parameter can be freed only after this function returns LC3_RESULT_NO_ERROR. If LC3Initialize allocated memory internally, that memory will be freed upon a successful call to this function.

If the result code LC3_RESULT_STILL_IN_USE is returned, there are still open sessions. No related resources are freed until all sessions associated with the handle are closed.

Returns

Zero on success or one of the defined LC3 result codes on error.

4.1.3.3 LC3EncodeSessionOpen()

```
LC3EncoderHandle_t OSALCALL LC3EncodeSessionOpen (
            uint16_t sampleRate,
            uint8_t bitsPerSample,
            LC3FrameSize_t frameSize,
            uint8_t * buffer,
            uint16_t * bufferSize,
            int32_t * result )
```

```
#include <LC3API.h>
```

Opens and initializes an LC3 Encoder session.

This function initializes the LC3 encoder in preparation for encoding one stream of audio. It allocates the memory needed for the session's processing, including any used for tracking progress from frame to frame, and initializes the allocated memory.

A non-NULL handle to the encoder session is returned when successful. The handle value is passed to future calls of LC3EncodeSessionData() and LC3EncodeSessionClose() for this session.

This function can allocate its own memory or take a user-provided buffer. It can also return the amount of memory needed without initializing. These memory options are performed in the following ways:

bufferSize = NULL: Allocates memory needed. The buffer parameter may be NULL.

∗**bufferSize < memory required by encoder:** Sets bufferSize's referenced value to the total memory required and returns LC3_RESULT_INSUFFICIENT_RESOURCES. The caller can then call this function again with a buffer of at least bufferSize's referenced value. A size of 0 will always trigger this condition. The buffer parameter may be NULL.

∗**bufferSize >= memory required by encoder:** Uses the user-provided buffer. bufferSize's referenced value is set to the amount of memory used by the encoder. The buffer parameter may not be NULL.

Parameters

`in`	*sampleRate*	Input PCM sample rate in Hz.
`in`	*bitsPerSample*	Input PCM bits per sample.
`in`	*frameSize*	Frame size of 7.5 or 10 msec.
`in`	*buffer*	Pointer to a memory buffer.
`in,out`	*bufferSize*	Size of the memory buffer passed in, number of bytes used returned.
`out`	*result*	Pointer to an LC3 Return Code integer.

Returns

Non-NULL LC3EncoderHandle_t on success or NULL on failure.

4.1.3.4 LC3EncodeSessionData()

```
int32_t OSALCALL LC3EncodeSessionData (
            LC3EncoderHandle_t encodeHandle,
            LC3EncodeInput_t * encodeInput,
            LC3EncodeOutput_t * encodeOutput )
```

```
#include <LC3API.h>
```

Encodes a frame of data using LC3.

This function processes the audio frame data passed in the encodeInput structure. It returns the encoded data in the encodeOutput structure.

Parameters

`in`	*encodeHandle*	Handle to an encoder instance.
`in`	*encodeInput*	Pointer to a structure pointing to the frame's audio samples to encode.
`in`	*encodeOutput*	Pointer to a structure to receive the frame's encoded data.

Returns

Zero on success or one of the defined LC3 result codes on error.

4.1.3.5 LC3EncodeSessionClose()

```
void OSALCALL LC3EncodeSessionClose (
            LC3EncoderHandle_t encodeHandle )
```

```
#include <LC3API.h>
```

Closes an LC3 Encoder session.

This function closes the encodeHandle's session and releases allocated session memory. Memory passed in the LC3EncodeSessionOpen's buffer parameter can be freed after this returns. The encodeHandle is no longer valid and should not be reused.

Parameters

in	*encodeHandle*	Handle to the encoder instance to close.

Returns

None.

4.1.3.6 LC3DecodeSessionOpen()

```
LC3DecoderHandle_t OSALCALL LC3DecodeSessionOpen (
             uint16_t sampleRate,
             uint8_t bitsPerSample,
             LC3FrameSize_t frameSize,
             uint8_t * buffer,
             uint16_t * bufferSize,
             int32_t * result )
```

```
#include <LC3API.h>
```

Opens and initializes an LC3 Decoder session.

This function initializes the LC3 decoder in preparation for decoding one stream of audio. It allocates the memory needed for the session's processing, including any used for tracking progress from frame to frame, and initializes the allocated memory.

A non-NULL handle to the decoder session is returned when successful. The handle value is passed to future calls of LC3DecodeSessionData() and LC3DecodeSessionClose() for this session.

This function can allocate its own memory or take a user-provided buffer. It can also return the amount of memory needed without initializing. These memory options are performed in the following ways:

bufferSize = NULL: Allocates memory needed. The buffer parameter may be NULL.

∗**bufferSize < memory required by decoder:** Sets bufferSize's referenced value to the total memory required and returns LC3_RESULT_INSUFFICIENT_RESOURCES. The caller can then call this function again with a buffer of at least bufferSize's referenced value. A size of 0 will always trigger this condition. The buffer parameter may be NULL.

∗**bufferSize >= memory required by decoder:** Uses the user-provided buffer. bufferSize's referenced value is set to the amount of memory used by the decoder. The buffer parameter may not be NULL.

Parameters

in	*sampleRate*	Output PCM sample rate in Hz.
in	*bitsPerSample*	Output PCM bits per sample.
in	*frameSize*	Frame size of 7.5 or 10 msec.
in	*buffer*	Pointer to a memory buffer.
in,out	*bufferSize*	Size of the memory buffer passed in, number of bytes used returned.
out	*result*	Pointer to an LC3 Return Code integer value.

Returns

Non-NULL LC3DecoderHandle_t on success or NULL on failure.

4.1.3.7 LC3DecodeSessionData()

```
int32_t OSALCALL LC3DecodeSessionData (
             LC3DecoderHandle_t decodeHandle,
             LC3DecodeInput_t * decodeInput,
             LC3DecodeOutput_t * decodeOutput )
```

```
#include <LC3API.h>
```

Decodes a frame of data using LC3.

Processes the frame of encoded bit stream data passed by the decodeInput structure. Returns the decoded audio PCM data via the decodeOutput structure.

This function will automatically apply packet loss concealment (PLC) to any frame where the decode input's bad↩ FrameIndicator is set to BadFrame or if the bit stream data contains errors. The caller should set badFrame↩ Indicator to BadFrame if the caller knows in advance of bit stream errors; this allows the decoder to skip decoding and directly apply PLC. If the caller is unaware of bit stream errors, but they are present, the decoder will detect them regardless. When PLC is applied successfully, this function still returns LC3_RESULT_NO_ERROR as when a frame is decoded successfully, but the decode output's PLCCounter parameter is incremented. Upon a successful decode, PLCCounter is reset. Therefore, a non-zero PLCCounter indicates PLC has been applied to the current frame.

Parameters

in	*decodeHandle*	Handle to a decoder instance.
in	*decodeInput*	Pointer to a structure pointing to the encoded bit stream frame to decode.
in	*decodeOutput*	Pointer to a structure to receive the frame's decoded PCM data.

Returns

Zero on success or one of the defined LC3 result codes on error.

4.1.3.8 LC3DecodeSessionClose()

```
void OSALCALL LC3DecodeSessionClose (
            LC3DecoderHandle_t decodeHandle )
```

```
#include <LC3API.h>
```

Closes an LC3 Decoder session.

This function closes the decodeHandle's session and and releases allocated session memory. Memory passed in the LC3DecodeSessionOpen's buffer parameter can be freed after this returns. The decodeHandle is no longer valid and should not be reused.

Parameters

in	*decodeHandle*	Handle to the decoder instance to close.

Returns

None.

4.1.3.9 LC3BitstreamBuffersize()

```
uint16_t OSALCALL LC3BitstreamBuffersize (
            uint16_t sampleRate,
            uint32_t maxBitRate,
            LC3FrameSize_t frameSize,
            int32_t * result )
```

```
#include <LC3API.h>
```

Calculates the buffer size required for bit stream data.

Returns the minimum buffer size in bytes needed to hold a frame of bit stream data for the specified sample rate and bit rate.

The maxBitRate parameter is the maximum bitrate value that will be passed to LC3EncodeSessionData() at any point during the session's encoding.

Parameters

in	*sampleRate*	Sample rate.
in	*maxBitRate*	Session's maximum encoding bit rate.
in	*frameSize*	Frame size of 7.5 or 10 msec.
out	*result*	Pointer to an LC3 Return Code integer.

Returns

Bit stream buffer size on success or 0 on error.

4.1.3.10 LC3PCMBuffersize()

```
uint16_t OSALCALL LC3PCMBuffersize (
            uint16_t sampleRate,
            uint8_t bitDepth,
            LC3FrameSize_t frameSize,
            int32_t * result )
```

```
#include <LC3API.h>
```

Calculates the buffer size required for PCM data.

Returns the minimum buffer size in bytes needed to hold a frame of PCM data for the specified sample rate and bit depth. A bitDepth value in the range (16,24] uses a 3 byte/sample PCM output while (24,32] uses 4 bytes/sample.

Parameters

`in`	*sampleRate*	Sample rate.
`in`	*bitDepth*	Bit depth of generated PCM data.
`in`	*frameSize*	Frame size of 7.5 or 10 msec.
`out`	*result*	Pointer to an LC3 Return Code integer.

Returns

PCM buffer size on success or 0 on error.

Chapter 5

Data Structure Documentation

5.1 LC3DecodeInput_t Struct Reference

LC3 decoder input bit stream structure.

```
#include "LC3API.h"
```

Data Fields

- const uint8_t ∗ **inputData**
  Pointer to the input LC3 data to decode.
- uint16_t **inputDataLength**
  Number of data bytes in the encoded frame.
- LC3BFI_t **badFrameIndicator**
  Frame condition as detected by device.

5.1.1 Detailed Description

LC3 decoder input bit stream structure.

Structure used to pass a frame of incoming bit stream data to the LC3 decoder. If there is a known problem with the frame, such as an error condition indicated by a Bluetooth controller, then badFrameIndicator should be set to BadFrame. inputDataLength must represent the number of bytes exactly in the frame to be decoded.

The documentation for this struct was generated from the following file:

- LC3API.h

5.2 LC3DecodeOutput_t Struct Reference

LC3 decoder output PCM structure.

```
#include "LC3API.h"
```

Data Fields

- void ∗ **PCMData**

  Output buffer pointer.
- uint16_t **PCMDataLength**

  Output buffer size in bytes.
- uint16_t **bytesWritten**

  Number of bytes written to the output buffer.
- uint16_t **PLCCounter**

  Number of successive frames to which PLC has been applied.

5.2.1 Detailed Description

LC3 decoder output PCM structure.

This structure is used to hold the decoded PCM audio samples of an audio stream. The buffer is externally allocated and its pointer and size are passed to the LC3 decoder. bytesWritten contains the number of bytes written to the buffer. The number of PCM bytes needed per frame can be calculated with LC3PCMBuffersize(). If PLC has been applied to the frame, PLCCounter will increment by one. A frame decoded without any bit errors resets PLCCounter to zero.

The documentation for this struct was generated from the following file:

- LC3API.h

5.3 LC3EncodeInput_t Struct Reference

LC3 encoder input PCM data structure.

```
#include "LC3API.h"
```

Data Fields

- const void ∗ **PCMData**

  Pointer to the PCM data to encode.
- uint16_t **PCMDataLength**

  Length of PCM data in bytes.
- uint32_t **encodeBitrate**

  Bit rate, in bits per second, at which to encode frame.
- uint16_t **bytesRead**

  Number of PCM data bytes read.

5.3.1 Detailed Description

LC3 encoder input PCM data structure.

Structure used to pass an audio frame to the encoder. Incoming PCM samples must be linear with a bit depth rounded to the nearest byte. A bit depth of 16 uses 2 bytes per sample, bit depths of 17-24 use 3 bytes per sample, and bit depths of 25-32 use 4 bytes per sample. The number of PCM bytes needed per frame can be calculated using LC3PCMBuffersize().

The documentation for this struct was generated from the following file:

- LC3API.h

5.4 LC3EncodeOutput_t Struct Reference

LC3 encoder output data structure.

```
#include "LC3API.h"
```

Data Fields

- uint8_t ∗ **outputData**

  Output buffer pointer.
- uint16_t **outputDataLength**

  Output buffer size in bytes.
- uint16_t **bytesWritten**

  Number of bit stream bytes written.

5.4.1 Detailed Description

LC3 encoder output data structure.

Structure that holds the LC3 encoder output. The output buffer is externally allocated and its pointer and available byte length are passed into the LC3 Encoder. The number of output bit stream data bytes can be calculated with LC3BitstreamBuffersize().

The documentation for this struct was generated from the following file:

- LC3API.h

Chapter 6

File Documentation

6.1 LC3API.h File Reference

This file contains the API functions for the LC3 Codec implementation.

```
#include "osal.h"
```

Data Structures

- struct LC3EncodeInput_t
  LC3 encoder input PCM data structure.
- struct LC3EncodeOutput_t
  LC3 encoder output data structure.
- struct LC3DecodeInput_t
  LC3 decoder input bit stream structure.
- struct LC3DecodeOutput_t
  LC3 decoder output PCM structure.

Macros

LC3 Result Codes

- #define **LC3_ERROR_OFFSET** (-5000)
  Offset for LC3 error values.
- #define **LC3_RESULT_NO_ERROR** (0)
  Success.
- #define **LC3_RESULT_INVALID_PARAMETER** ((LC3_ERROR_OFFSET)-1)
  Error: invalid parameter detected.
- #define **LC3_RESULT_INSUFFICIENT_RESOURCES** ((LC3_ERROR_OFFSET)-2)
  Error: not enough memory available.
- #define **LC3_RESULT_NOT_INITIALIZED** ((LC3_ERROR_OFFSET)-3)
  Error: not initialized before use.
- #define **LC3_RESULT_UNKNOWN_ERROR** ((LC3_ERROR_OFFSET)-4)

Error: unknown error detected.
- #define **LC3_RESULT_INVALID_BITRATE** ((LC3_ERROR_OFFSET)-5)
  Error: invalid bitrate value.
- #define **LC3_RESULT_INPUT_BUFFER_TOO_SMALL** ((LC3_ERROR_OFFSET)-6)
  Error: passed buffer size too small for input data.
- #define **LC3_RESULT_OUTPUT_BUFFER_TOO_SMALL** ((LC3_ERROR_OFFSET)-7)
  Error: passed buffer size too small for output data.
- #define **LC3_RESULT_INVALID_BITSPERSAMPLE** ((LC3_ERROR_OFFSET)-8)
  Error: invalid bits per sample value.
- #define **LC3_RESULT_INVALID_SAMPLERATE** ((LC3_ERROR_OFFSET)-9)
  Error: invalid sample rate value.
- #define **LC3_RESULT_BITERRORCONDITION** ((LC3_ERROR_OFFSET)-10)
  Error: bit error condition detected in bit stream data.
- #define **LC3_RESULT_FEATURE_NOT_SUPPORTED** ((LC3_ERROR_OFFSET)-11)
  Error: feature not supported in this build.
- #define **LC3_RESULT_STILL_IN_USE** ((LC3_ERROR_OFFSET)-12)
  Error: Sessions not closed, buffer still in use.
- #define **LC3_RESULT_ALREADY_INITIALIZED** ((LC3_ERROR_OFFSET)-13)
  Error: Codec already initialized, call LC3Deinitialize first.

LC3 Sample Rate Bits

- #define **LC3_SAMPLE_RATE_NONE** (0x00)
  No sample rate selected.
- #define **LC3_SAMPLE_RATE_8_KHZ** (0x01)
  8 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_16_KHZ** (0x02)
  16 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_24_KHZ** (0x04)
  24 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_32_KHZ** (0x08)
  32 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_441_KHZ** (0x10)
  44.1 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_48_KHZ** (0x20)
  48 kHz sample rate bit.
- #define **LC3_SAMPLE_RATE_ALL** (0x1F)
  All sample rates selected.

Typedefs

- typedef void ∗ **LC3EncoderHandle_t**
  LC3 encoder handle type.
- typedef void ∗ **LC3DecoderHandle_t**
  LC3 decoder handle type.

Enumerations

- enum LC3FrameSizeConfig_t { LC3FrameSize10MsConfig , LC3FrameSize7_5MsConfig , LC3FrameSizeBothConfig }
  Frame Size for configuration.
- enum LC3FrameSize_t { LC3FrameSize10Ms , LC3FrameSize7_5Ms }
  Session Frame Size.
- enum LC3BFI_t { **GoodFrame** , **BadFrame** }
  Bad Frame Indicator (BFI).

Functions

- int32_t OSALCALL LC3Initialize (uint8_t encoderSampleRates, uint8_t decoderSampleRates, LC3FrameSizeConfig_t frameSizeConfig, uint8_t uniqueSessions, uint8_t ∗buffer, uint32_t ∗bufferSize)

  Initializes the LC3 Codec.
- int32_t OSALCALL LC3Deinitialize (void)

  Deinitializes the LC3 Codec.
- LC3EncoderHandle_t OSALCALL LC3EncodeSessionOpen (uint16_t sampleRate, uint8_t bitsPerSample, LC3FrameSize_t frameSize, uint8_t ∗buffer, uint16_t ∗bufferSize, int32_t ∗result)

  Opens and initializes an LC3 Encoder session.
- int32_t OSALCALL LC3EncodeSessionData (LC3EncoderHandle_t encodeHandle, LC3EncodeInput_t ∗encodeInput, LC3EncodeOutput_t ∗encodeOutput)

  Encodes a frame of data using LC3.
- void OSALCALL LC3EncodeSessionClose (LC3EncoderHandle_t encodeHandle)

  Closes an LC3 Encoder session.
- LC3DecoderHandle_t OSALCALL LC3DecodeSessionOpen (uint16_t sampleRate, uint8_t bitsPerSample, LC3FrameSize_t frameSize, uint8_t ∗buffer, uint16_t ∗bufferSize, int32_t ∗result)

  Opens and initializes an LC3 Decoder session.
- int32_t OSALCALL LC3DecodeSessionData (LC3DecoderHandle_t decodeHandle, LC3DecodeInput_t ∗decodeInput, LC3DecodeOutput_t ∗decodeOutput)

  Decodes a frame of data using LC3.
- void OSALCALL LC3DecodeSessionClose (LC3DecoderHandle_t decodeHandle)

  Closes an LC3 Decoder session.
- uint16_t OSALCALL LC3BitstreamBuffersize (uint16_t sampleRate, uint32_t maxBitRate, LC3FrameSize_t frameSize, int32_t ∗result)

  Calculates the buffer size required for bit stream data.
- uint16_t OSALCALL LC3PCMBuffersize (uint16_t sampleRate, uint8_t bitDepth, LC3FrameSize_t frameSize, int32_t ∗result)

  Calculates the buffer size required for PCM data.

6.1.1 Detailed Description

This file contains the API functions for the LC3 Codec implementation.