

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		

Profiling has been done in order to evaluate the resource consumption in terms of MIPS, RAM and FLASH figures may change depending on specific use case optimizations. The following figure shows the profiling results for source localization using a STM32F446 MCU based on an ARM M4 core with floating point unit running at 168 MHz (210 DMIPS available) and the IAR embedded workbench tool chain, version 7.70. Optimization has been set on High, speed.

Algorithm	Microphones	Resolution	CPU (MIPS)	FLASH (Bytes)	RAM (Bytes)
XCORR	2	30 degrees (distance = 0.08 m)	2.2	13664 + 24210 (for ARM code and tables).	2913
XCORR	4	30 degrees (distance = 0.08 m)	4.5		5641
XCORR	2	12 degrees (distance = 0.16 m)	4.0		2913
XCORR	4	12 degrees (distance = 0.16 m)	7.9		5641
GCC-PHAT	2	1 degree	49.7		17452
GCC-PHAT	4	1 degree	98.8		27093
GCC-PHAT	2	6 degrees	11.8		17452
GCC-PHAT	4	6 degrees	22.6		27093
GCC-PHAT	2	30 degrees	5.8		17452
GCC-PHAT	4	30 degrees	11.0		27093
BMPH	2	6 degrees	7.6		10920
BMPH	4	6 degrees	11.9		15016
BMPH	2	30 degrees	6.6		10920
BMPH	4	30 degrees	8.7		15016

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		

Modules

Here is a list of all modules:

[detail level [1](#) [2](#) [3](#) [4](#)]

▼ MIDDLEWARES	
▼ ACOUSTIC_SL	
▼ AcousticSL Exported Constants	
Acoustic_SL_algorithm_type	Source Localization algorithm type
Acoustic_SL_errors	Source Localization errors
AcousticSL Exported Types	
AcousticSL Exported Functions	

Generated by **doxygen** 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling	Modules	
<h2>MIDDLEWARES</h2>			

Modules

ACOUSTIC_SL

Detailed Description

Generated by  1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures		
Files	Profiling	Modules			
ACOUSTIC_SL					
MIDDLEWARES					

Modules

[AcousticSL Exported Constants](#)

[AcousticSL Exported Types](#)

[AcousticSL Exported Functions](#)

Detailed Description

Generated by  1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		

[Modules](#) | [Macros](#)

AcousticSL Exported Constants

[MIDDLEWARES](#) » [ACOUSTIC_SL](#)

Modules

Acoustic_SL_algorithm_type

Source Localization algorithm type.

Acoustic_SL_errors

Source Localization errors.

Macros

```
#define ACOUSTIC_SL_NO_AUDIO_DETECTED -100
```

Detailed Description

Generated by  1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures	
Files	Profiling			Macros
Acoustic_SL_algorithm_type				
MIDDLEWARES » ACOUSTIC_SL »				
AcousticSL Exported Constants				

Source Localization algorithm type. [More...](#)

Macros

```
#define ACOUSTIC_SL_ALGORITHM_XCORR ((uint32_t)0x0000000
```

```
#define ACOUSTIC_SL_ALGORITHM_GCCP ((uint32_t)0x00000000
```

```
#define ACOUSTIC_SL_ALGORITHM_BMPH ((uint32_t)0x00000000
```

Detailed Description

Source Localization algorithm type.

Generated by [doxygen](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		Macros

Acoustic_SL_errors

[MIDDLEWARES](#) » [ACOUSTIC_SL](#) »

[AcousticSL Exported Constants](#)

Source Localization errors. More...

Macros

```
#define ACOUSTIC_SL_ALGORITHM_ERROR ((uint32_t)0x000000  
  
#define ACOUSTIC_SL_PTR_CHANNELS_ERROR ((uint32_t)0x00000001  
  
#define ACOUSTIC_SL_CHANNEL_NUMBER_ERROR ((uint32_t)0x00000002  
  
#define ACOUSTIC_SL_SAMPLING_FREQ_ERROR ((uint32_t)0x00000003  
  
#define ACOUSTIC_SL_RESOLUTION_ERROR ((uint32_t)0x00000004  
  
#define ACOUSTIC_SL_THRESHOLD_ERROR ((uint32_t)0x0000000C  
  
#define ACOUSTIC_SL_DISTANCE_ERROR ((uint32_t)0x00000004C  
  
#define ACOUSTIC_SL_NUM_OF_SAMPLES_ERROR ((uint32_t)0x00000005  
  
#define ACOUSTIC_SL_PROCESSING_ERROR ((uint32_t)0x0000000C  
  
#define ACOUSTIC_LOCK_ERROR ((uint32_t)0x10000000)
```

Detailed Description

Source Localization errors.

Generated by [doxygen](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling	Data Structures	

AcousticSL Exported Types

[MIDDLEWARES](#) » [ACOUSTIC_SL](#)

Data Structures

struct **AcousticSL_Handler_t**

Library handler. It keeps track of the static parameters and it handles the internal state of the algorithm. [More...](#)

struct **AcousticSL_Config_t**

Library dynamic configuration handler. It contains dynamic parameters. [More...](#)

Detailed Description

Generated by  1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
Data Structures	Data Structure Index	Data Fields	Data Fields
AcousticSL_Handler_t Struct Reference			
MIDDLEWARES » ACOUSTIC_SL » AcousticSL Exported Types			

Library handler. It keeps track of the static parameters and it handles the internal state of the algorithm. [More...](#)

```
#include <acoustic_sl.h>
```

Data Fields

uint32_t **algorithm**

uint32_t **sampling_frequency**

uint32_t **channel_number**

uint8_t **ptr_M1_channels**

uint8_t **ptr_M2_channels**

uint8_t **ptr_M3_channels**

uint8_t **ptr_M4_channels**

uint16_t **M12_distance**

uint16_t **M34_distance**

uint32_t **internal_memory_size**

uint32_t * **pInternalMemory**

int16_t **samples_to_process**

Detailed Description

Library handler. It keeps track of the static parameters and it handles the internal state of the algorithm.

Field Documentation

uint32_t algorithm

Specifies the algorithm to be used between XCORR and GCC-PHAT. This parameter can be a value of [**Acoustic_SL_algorithm_type**](#). Default value is ACOUSTIC_BF_TYPE_CARDIOID_BASIC

uint32_t channel_number

Specifies the number of channels, can be 2 for 180° estimation, 4 for 360° estimation. Default value is

1.

uint32_t internal_memory_size

Keeps track of the amount of memory required for the current setup. It's filled by the [**AcousticSL_getMemorySize\(\)**](#) function and must be used to allocate the right amount of RAM

uint16_t M12_distance

Distance between Mic1 and Mic2 in decimals of a millimeter. Default value is 150.

uint16_t M34_distance

Distance between Mic3 and Mic4 in decimals of a millimeter. Default

value is 150.

uint32_t* plnternalMemory

Pointer to the memory allocated by the user

uint8_t ptr_M1_channels

Number of channels in the stream of Microphone 1. Default value is 1.

uint8_t ptr_M2_channels

Number of channels in the stream of Microphone 2. Default value is 1.

uint8_t ptr_M3_channels

Number of channels in the stream of Microphone 3. Default value is 1.

uint8_t ptr_M4_channels

Number of channels in the stream of Microphone 4. Default value is 1.

int16_t samples_to_process

Specifies the number of samples to be processed at a time

uint32_t sampling_frequency

Specifies the sampling frequency - for future use

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

- D:/Documents.Repositories/DMIC/Private/Fw/OSX_Libraries/Source
-

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
Data Structures	Data Structure Index	Data Fields	Data Fields
AcousticSL_Config_t Struct Reference MIDDLEWARES » ACOUSTIC_SL » AcousticSL Exported Types			

Library dynamic configuration handler. It contains dynamic parameters.
[More...](#)

```
#include <acoustic_sl.h>
```

Data Fields

`uint16_t threshold`

`uint32_t resolution`

Detailed Description

Library dynamic configuration handler. It contains dynamic parameters.

Field Documentation

uint32_t resolution

Angle resolution for the algorithms. Ignored if XCORR is used.
Default value is 4.

uint16_t threshold

Specifies a value related to a voice-activity score. With values below the threshold, the algorithm does not act. The threshold value ranges from 0 to 1000 and the default value is 24.

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

- D:/Documents.Repositories/DMIC/Private/Fw/OSX_Libraries/Source

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling	Functions	

AcousticSL Exported Functions

[MIDDLEWARES](#) » [ACOUSTIC_SL](#)

Functions

uint32_t **AcousticSL_getMemorySize** (**AcousticSL_Handler_t** *pHandler)

Fills the "internal_memory_size" of the pHandler parameter passed as argument with a value representing the right amount of memory needed by the library, depending on the specific static parameters adopted. [More...](#)

uint32_t **AcousticSL_Init** (**AcousticSL_Handler_t** *pHandler)

Library initialization. [More...](#)

uint32_t **AcousticSL_Data_Input** (void *pM1, void *pM2, void *pM3, void *pM4, **AcousticSL_Handler_t** *pHandler)

Library data input. [More...](#)

uint32_t **AcousticSL_Process** (int32_t *Estimated_Angle,

AcousticSL_Handler_t *pHandler)

Library run function, performs audio analysis when all required data has been collected. [More...](#)

uint32_t **AcousticSL_setConfig** (**AcousticSL_Handler_t**

*pHandler, **AcousticSL_Config_t** *pConfig)

Library setup function, it sets the values for threshold and resolution. It can be called at runtime to change dynamic parameters. [More...](#)

uint32_t **AcousticSL_getConfig** (**AcousticSL_Handler_t**

*pHandler, **AcousticSL_Config_t** *pConfig)

Fills the pConfig structure with the actual dynamic parameters as they are currently used inside the library.

[More...](#)

uint32_t **AcousticSL_GetLibVersion** (char *version)

To be used to retrieve version information. [More...](#)

Detailed Description

Function Documentation

```
uint32_t  
AcousticSL_Data_Input ( void *  
                      void *  
                      void *  
                      void *  
                      AcousticSL_Handler_t * pHandler  
                    )
```

Library data input.

Parameters

- pM1** pointer to an array that contains PCM samples (16 bit signed int) representing 1 ms of data acquired by the first channel.
- pM2** pointer to an array that contains PCM samples (16 bit signed int) representing 1 ms of data acquired by the second channel.
- pM3** pointer to an array that contains PCM samples (16 bit signed int) representing 1 ms of data acquired by the third channel.
- pM4** pointer to an array that contains PCM samples (16 bit signed int) representing 1 ms of data acquired by the fourth channel.
- pHandler** pointer to the handler of the current Source Localization instance running.

Return values

- 1 if data collection is finished and libSoundSourceLoc_Process

must be called, 0 otherwise.

Note

Input function reads samples skipping the required number of values depending on the Ptr_Mx_Channels configuration.
pM3 and pM4 are ignored in the case the library is setup for using 2 channels.

```
uint32_t  
AcousticSL_getConfig ( AcousticSL_Handler_t * pHandler,  
                      AcousticSL_Config_t * pConfig  
)
```

Fills the pConfig structure with the actual dynamic parameters as they are currently used inside the library.

Parameters

pHandler pointer to the handler of the current Source Localization instance running.

pConfig pointer to the dynamic parameters handler that will be filled with the current library configuration

Return values

0 if everything is fine.

```
uint32_t AcousticSL_GetLibVersion ( char * version )
```

To be used to retrieve version information.

Parameters

version char array to be filled with the current library version

Return values

0 if everything is fine.

```
uint32_t  
AcousticSL_getMemorySize ( AcousticSL_Handler_t * pHandler )
```

Fills the "internal_memory_size" of the pHandler parameter passed as argument with a value representing the right amount of memory needed by the library, depending on the specific static parameters adopted.

Parameters

pHandler `AcousticSL_Handler_t` filled with desired parameters.

Return values

0 if everything is fine.

```
uint32 t AcousticSL Init (AcousticSL Handler t * pHandler)
```

Library initialization.

Parameters

pHandler `AcousticSL_Handler_t` filled with desired parameters.

Return values

0 if everything is fine. different from 0 if erroneous parameters have been passed to the Init function and the default value has been used. The specific error can be recognized by checking the relative bit in the returned word.

```
uint32_t  
AcousticSL_Process ( int32_t *           Estimated_Angle,  
                     AcousticSL_Handler_t * pHandler  
)  
{
```

Library run function, performs audio analysis when all required data has been collected.

Parameters

Estimated_Angle	pointer to the int32_t variable that will contain the computed value.
pHandler	pointer to the handler of the current Source Localization instance running.

Return values

0 if everything is ok, 1 otherwise

uint32_t

AcousticSL_setConfig ([AcousticSL_Handler_t](#) * pHandler,
[AcousticSL_Config_t](#) * pConfig
)

Library setup function, it sets the values for threshold and resolution. It can be called at runtime to change dynamic parameters.

Note

Only the threshold and resolution are evaluated by the SetConfig function.

Return values

0 if everything is fine. different from 0 if erroneous parameters have been passed to the Init function and the default value has been used. The specific error can be recognized by checking the relative bit in the returned word.

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
Data Structures	Data Structure Index	Data Fields	

Data Structures

Here are the data structures with brief descriptions:

 AcousticSL_Config_t	Library dynamic configuration handler. It contains dynamic parameters
 AcousticSL_Handler_t	Library handler. It keeps track of the static parameters and it handles the internal state of the algorithm

Generated by [doxygen](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
Data Structures	Data Structure Index	Data Fields	

Data Structure Index

A

[**AcousticSL_Handler_t**](#)

[**AcousticSL_Config_t**](#)

A

Generated by [doxygen](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
Data Structures		Data Structure Index	Data Fields
All	Variables		

Here is a list of all documented struct and union fields with links to the struct/union documentation for each field:

- algorithm : [AcousticSL_Handler_t](#)
- channel_number : [AcousticSL_Handler_t](#)
- internal_memory_size : [AcousticSL_Handler_t](#)
- M12_distance : [AcousticSL_Handler_t](#)
- M34_distance : [AcousticSL_Handler_t](#)
- pInternalMemory : [AcousticSL_Handler_t](#)
- ptr_M1_channels : [AcousticSL_Handler_t](#)
- ptr_M2_channels : [AcousticSL_Handler_t](#)
- ptr_M3_channels : [AcousticSL_Handler_t](#)
- ptr_M4_channels : [AcousticSL_Handler_t](#)
- resolution : [AcousticSL_Config_t](#)
- samples_to_process : [AcousticSL_Handler_t](#)
- sampling_frequency : [AcousticSL_Handler_t](#)
- threshold : [AcousticSL_Config_t](#)

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
Data Structures	Data Structure Index	Data Fields	
All	Variables		

- algorithm : [AcousticSL_Handler_t](#)
- channel_number : [AcousticSL_Handler_t](#)
- internal_memory_size : [AcousticSL_Handler_t](#)
- M12_distance : [AcousticSL_Handler_t](#)
- M34_distance : [AcousticSL_Handler_t](#)
- plInternalMemory : [AcousticSL_Handler_t](#)
- ptr_M1_channels : [AcousticSL_Handler_t](#)
- ptr_M2_channels : [AcousticSL_Handler_t](#)
- ptr_M3_channels : [AcousticSL_Handler_t](#)
- ptr_M4_channels : [AcousticSL_Handler_t](#)
- resolution : [AcousticSL_Config_t](#)
- samples_to_process : [AcousticSL_Handler_t](#)
- sampling_frequency : [AcousticSL_Handler_t](#)
- threshold : [AcousticSL_Config_t](#)

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		

File List

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

[detail level [1](#) [2](#) [3](#) [4](#)]

SourceLocalization	
trunk	
export	
acoustic_sl.h	This file contains Acoustic Sound Source Localization library definitions

Generated by [**doxygen**](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		
SourceLocalization			
<h2>SourceLocalization Directory Reference</h2>			

Directories

Generated by **doxygen** 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		
SourceLocalization > trunk			
<h2>trunk Directory Reference</h2>			

Directories

Generated by **doxygen** 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		
SourceLocalization > trunk > export			
export Directory Reference			

Files

file [**acoustic_sl.h \[code\]**](#)

This file contains Acoustic Sound Source Localization library definitions.

Generated by [doxygen](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		

SourceLocalization > trunk > export

Data Structures | Macros | Functions

acoustic_sl.h File Reference

This file contains Acoustic Sound Source Localization library definitions. [More...](#)

```
#include "stdint.h"
```

[Go to the source code of this file.](#)

Data Structures

struct **AcousticSL_Handler_t**

Library handler. It keeps track of the static parameters and it handles the internal state of the algorithm. [More...](#)

struct **AcousticSL_Config_t**

Library dynamic configuration handler. It contains dynamic parameters. [More...](#)

Macros

```
#define ACOUSTIC_SL_ALGORITHM_XCORR ((uint32_t)0x00000000

#define ACOUSTIC_SL_ALGORITHM_GCCP ((uint32_t)0x00000000

#define ACOUSTIC_SL_ALGORITHM_BMPH ((uint32_t)0x00000000

#define ACOUSTIC_SL_ALGORITHM_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_PTR_CHANNELS_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_CHANNEL_NUMBER_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_SAMPLING_FREQ_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_RESOLUTION_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_THRESHOLD_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_DISTANCE_ERROR ((uint32_t)0x00000004C

#define ACOUSTIC_SL_NUM_OF_SAMPLES_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_SL_PROCESSING_ERROR ((uint32_t)0x00000000

#define ACOUSTIC_LOCK_ERROR ((uint32_t)0x10000000)

#define ACOUSTIC_SL_NO_AUDIO_DETECTED -100
```

Functions

- uint32_t **AcousticSL_getMemorySize** (**AcousticSL_Handler_t** *pHandler)
Fills the "internal_memory_size" of the pHandler parameter passed as argument with a value representing the right amount of memory needed by the library, depending on the specific static parameters adopted. [More...](#)
- uint32_t **AcousticSL_Init** (**AcousticSL_Handler_t** *pHandler)
Library initialization. [More...](#)
- uint32_t **AcousticSL_Data_Input** (void *pM1, void *pM2, void *pM3, void *pM4, **AcousticSL_Handler_t** *pHandler)
Library data input. [More...](#)
- uint32_t **AcousticSL_Process** (int32_t *Estimated_Angle, **AcousticSL_Handler_t** *pHandler)
Library run function, performs audio analysis when all required data has been collected. [More...](#)
- uint32_t **AcousticSL_setConfig** (**AcousticSL_Handler_t** *pHandler, **AcousticSL_Config_t** *pConfig)
Library setup function, it sets the values for threshold and resolution. It can be called at runtime to change dynamic parameters. [More...](#)
- uint32_t **AcousticSL_getConfig** (**AcousticSL_Handler_t** *pHandler, **AcousticSL_Config_t** *pConfig)
Fills the pConfig structure with the actual dynamic parameters as they are currently used inside the library. [More...](#)
- uint32_t **AcousticSL_GetLibVersion** (char *version)
To be used to retrieve version information. [More...](#)

Detailed Description

This file contains Acoustic Sound Source Localization library definitions.

Author

Central Labs

Version

V2.0.0

Date

08-March-2017

Attention

© COPYRIGHT 2015 STMicroelectronics

Licensed under Software License Agreement SLA0077, (the "License"). You may not use this package except in compliance with the License. You may obtain a copy of the License at:

http://www.st.com/content/st_com/en/search.html?q=SLA0077-t=keywords-page=1

Some of the library code is based on the CMSIS DSP software library by ARM, a suite of common signal processing functions for use on Cortex-M processor based devices. Licencing terms are available in the attached release_note.html file, in the libSoundSourceLoc100 application note, in the next lines of this document and it's available on the web at: <http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>

ARM licence note:

Copyright (C) 2009-2012 ARM Limited. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of ARM nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL

COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Generated by  1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		
All	Functions		

Here is a list of all documented functions, variables, defines, enums, and typedefs with links to the documentation:

- AcousticSL_Data_Input() : [acoustic_sl.h](#)
- AcousticSL_getConfig() : [acoustic_sl.h](#)
- AcousticSL_GetLibVersion() : [acoustic_sl.h](#)
- AcousticSL_getMemorySize() : [acoustic_sl.h](#)
- AcousticSL_Init() : [acoustic_sl.h](#)
- AcousticSL_Process() : [acoustic_sl.h](#)
- AcousticSL_setConfig() : [acoustic_sl.h](#)

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		
All	Functions		

- AcousticSL_Data_Input() : [acoustic_sl.h](#)
- AcousticSL_getConfig() : [acoustic_sl.h](#)
- AcousticSL_GetLibVersion() : [acoustic_sl.h](#)
- AcousticSL_getMemorySize() : [acoustic_sl.h](#)
- AcousticSL_Init() : [acoustic_sl.h](#)
- AcousticSL_Process() : [acoustic_sl.h](#)
- AcousticSL_setConfig() : [acoustic_sl.h](#)

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		

Related Pages

Here is a list of all related documentation pages:

[profiling](#)

Generated by [doxygen](#) 1.8.11

AcousticSL Software Library

Main Page	Related Pages	Modules	Data Structures
Files	Profiling		
File List	Globals		
SourceLocalization	trunk	export	

acoustic_sl.h

Go to the documentation of this file.

```
1
57 /* Define to prevent recursive inclusion ---
   -----
58 #ifndef __ACOUSTIC_SL_H
59 #define __ACOUSTIC_SL_H
60
61 #include "stdint.h"
62
63 /* Includes -----
   -----
64 /* Exported types -----
   -----
65 /* Exported constants -----
   -----
66 /* Exported macro -----
   -----
67 /* Exported define -----
   -----
68 /* Exported functions -----
   -----
69
86 #define ACOUSTIC_SL_ALGORITHM_XCORR
     ((uint32_t)0x00000001)
87 #define ACOUSTIC_SL_ALGORITHM_GCCP
```

```
    ((uint32_t)0x00000002)
88 | #define ACOUSTIC_SL_ALGORITHM_BMPH
     ((uint32_t)0x00000004)
89 |
97 | #define ACOUSTIC_SL_ALGORITHM_ERROR
     ((uint32_t)0x00000001)
98 | #define ACOUSTIC_SL_PTR_CHANNELS_ERROR
     ((uint32_t)0x00000002)
99 | #define ACOUSTIC_SL_CHANNEL_NUMBER_ERROR
     ((uint32_t)0x00000004)
100 | #define ACOUSTIC_SL_SAMPLING_FREQ_ERROR
     ((uint32_t)0x00000008)
101 | #define ACOUSTIC_SL_RESOLUTION_ERROR
     ((uint32_t)0x00000010)
102 | #define ACOUSTIC_SL_THRESHOLD_ERROR
     ((uint32_t)0x00000020)
103 | #define ACOUSTIC_SL_DISTANCE_ERROR
     ((uint32_t)0x00000040)
104 | #define ACOUSTIC_SL_NUM_OF_SAMPLES_ERROR
     ((uint32_t)0x00000080)
105 | #define ACOUSTIC_SL_PROCESSING_ERROR
     ((uint32_t)0x00000100)
106 |
107 | #ifndef ACOUSTIC_LOCK_ERROR
108 | #define ACOUSTIC_LOCK_ERROR
     ((uint32_t)0x10000000)
109 | #endif
110 |
113 | #define ACOUSTIC_SL_NO_AUDIO_DETECTED
     -100
114 |
125 | typedef struct
126 | {
127 |     uint32_t algorithm;
130 |     uint32_t sampling_frequency;
132 |     uint32_t channel_number;
134 |     uint8_t ptr_M1_channels;
```

```
135     uint8_t ptr_M2_channels;
136     uint8_t ptr_M3_channels;
137     uint8_t ptr_M4_channels;
139     uint16_t M12_distance;
140     uint16_t M34_distance;
142     uint32_t internal_memory_size;
145     uint32_t * pInternalMemory;
146     int16_t samples_to_process;
148 } AcousticSL_Handler_t;
149
153 typedef struct
154 {
155     uint16_t threshold;
156     uint32_t resolution;
157 } AcousticSL_Config_t;
158
159
173 uint32_t
AcousticSL_getMemorySize(AcousticSL_Handler_t
* pHandler);
174
182 uint32_t
AcousticSL_Init(AcousticSL_Handler_t *
pHandler);
183
199 uint32_t AcousticSL_Data_Input(void *pM1,
void *pM2, void *pM3, void *pM4,
AcousticSL_Handler_t * pHandler);
200
207 uint32_t AcousticSL_Process(int32_t *
Estimated_Angle, AcousticSL_Handler_t *
pHandler);
208
217 uint32_t
AcousticSL_SetConfig(AcousticSL_Handler_t *
pHandler, AcousticSL_Config_t * pConfig);
218
```

```
225 uint32_t
  AcousticSL_getConfig(AcousticSL_Handler_t *
    pHandler, AcousticSL_Config_t * pConfig);
226
232 uint32_t AcousticSL_GetLibVersion(char
  *version);
233
234
245 #endif /* __ACOUSTIC_SL_H */
246
247 /***** (C) COPYRIGHT 2010
  STMicroelectronics *****END OF FILE****/
```