# Acoustic Event Detection MI User Manual

V1.0

## REVISION HISTORY

| Revision No. | Description | Date |
|---|---|---|
| 1.0 | • Created | 08/07/2017 |

## TABLE OF CONTENTS

## LIST OF FIGURES

# 1. INTRODUCTION

## 1.1. Purpose

Loud sound detection (LSD) is a function used for detecting dBFS from audio streams.

## 2. SPECIFICATION

1. For best performance, background environment should be quiet
2. If you are using audio files as the sound source, you should make sure
   甲、 There is no aliasing in audio files, see Figure 1
   乙、 There is no signal clipping in audio files, see Figure 2
   丙、 Effective sample rate is larger than 8 kHz, see Figure 3
   丁、 Speaker volume and mic gain is high enough

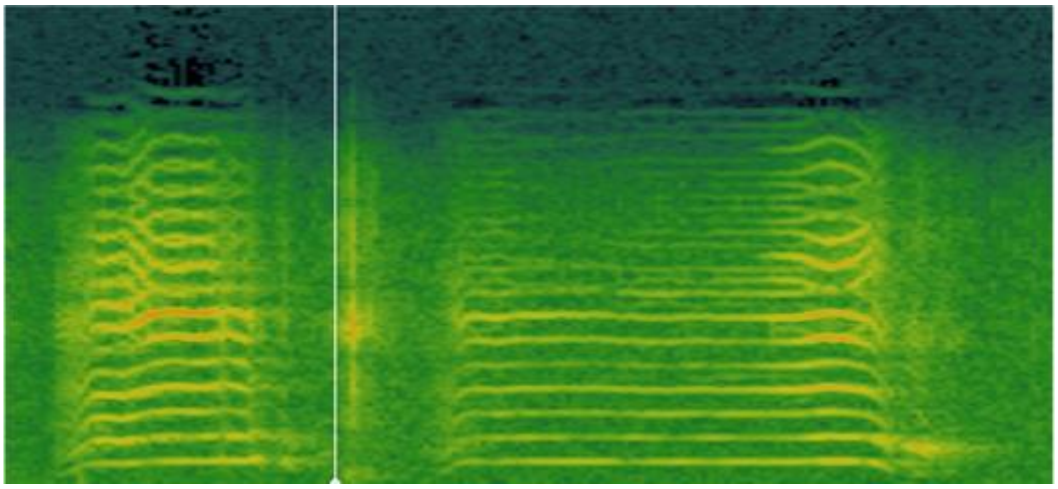


Figure 1: Audio example of aliasing.

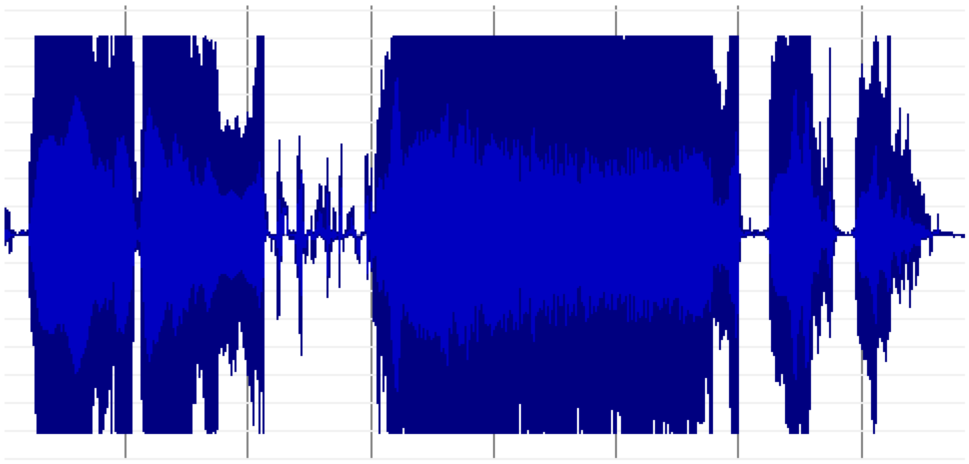


Figure 2: Audio example of clipping.

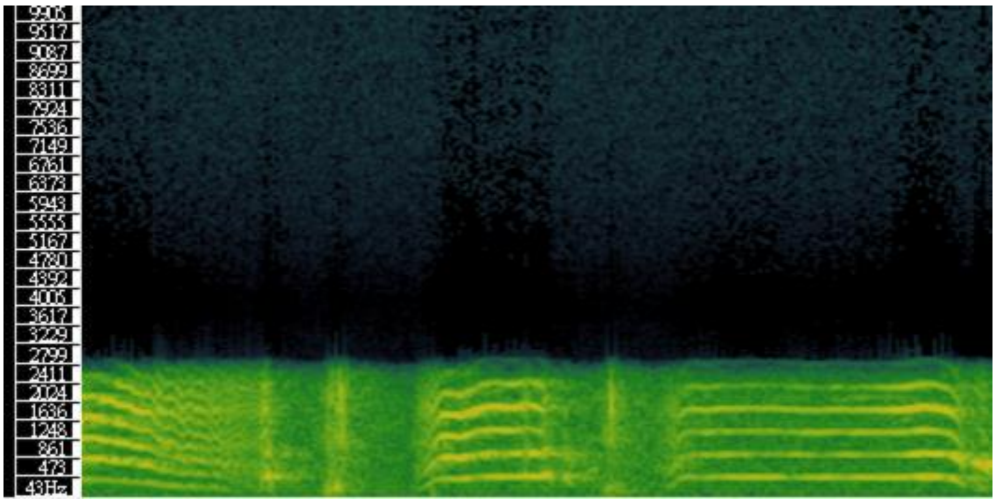


Figure 3. Audio example of effective sample rate is below 8kHz

# 3. API REFERENCE

## 3.1. API Overview

- MI_LSD_Init: Initialize LSD library
- MI_LSD_Uninit: To exit the lib function and release memory
- MI_LSD_GetLsdResult: Get dBFS result of LSD process
- MI_LSD_Run: Perform LSD
- MI_LSD_GetResult: Get result of LSD library
- MI_LSD_SetThreshold: Set LSD threshold in dBFS

## 3.2. API Lists

### MI_LSD_Init

Purpose

Initialize LSD library

Function Prototype

*LSD_HANDLE* MI_LSD_Init(*LSD_PARAMS* *lsd_params, *S32* *point_length)*;*

Arguments

| Name | Description |
|---|---|
| lsd_params | Structure of lsd_params |
| point_length | Input data length |

Return value

| Return value | Description |
|---|---|
| LSD_HANDLE | LSD handle pointer address |
| NULL | Initialization failure |

Requirement

Header files: mi_lsd.h

Library files: libLSD_Linux.a or libLSD_Linux.so

## MI_LSD_Uninit

Purpose
To exit the lib function and release memory

Function Prototype
*MI_RET* MI_LSD_Uninit(*LSD_HANDLE* lsd_handle);

Arguments

| Name | Description |
|---|---|
| lsd_handle | Pointer to the LSD_HANDLE |

Return value

| Return value | Description |
|---|---|
| MI_RET_SUCCESS | Success |
| MI_LSD_RET_INVALID_HANDLE | Invalid LSD handle |

Requirement
Header files: mi_lsd.h
Library files: libLSD_Linux.a or libLSD_Linux.so

## MI_LSD_SetLsdThreshold

Purpose
Set LSD threshold in dBFS

Function Prototype
*MI_RET* MI_LSD_SetLsdThreshold*(LSD_HANDLE* lsd_handle, *S32* threshold_db);

Arguments

| Name | Description |
|---|---|
| lsd_handle | Pointer to the LSD_HANDLE |
| threshold_db | Default threshold is -15 (dBFS) |

Return value

| Return value | Description |
|---|---|
| MI_RET_SUCCESS | Success |
| MI_LSD_RET_INVALID_HANDLE | Invalid LSD handle |

Requirement
Header files: mi_lsd.h

Library files: libLSD_Linux.a or libLSD_Linux.so

## MI_LSD_GetdBResult

Purpose
Get dBFS result of LSD

Function Prototype
*MI_RET* MI_LSD_GetdBResult(*LSD_HANDLE* lsd_handle, *S16* *audio_input, *S16* *lsd_db_result);

Arguments

| Name | Description |
|---|---|
| lsd_handle | Pointer to the LSD_HANDLE |
| audio_input | Audio input address. The input array should have point_number*channel (fields of LSDProcessStruct) elements. For example, for 8 kHz stereo, the input array should have 256*2 elements; for 32 kHz mono, the input array should have 1024*1 elements |
| lsd_db_result | Pointer to the value of dBFS |

Return value

| Return value | Description |
|---|---|
| MI_RET_SUCCESS | Success |
| MI_LSD_RET_INVALID_HANDLE | Invalid LSD handle |

Requirement
Header files: mi_lsd.h
Library files: libLSD_Linux.a or libLSD_Linux.so

Note
- MI_LSD_GetdBResult should be called before MI_LSD_Run for each frame

## MI_LSD_Run

Purpose
Perform LSD

Function Prototype
*MI_RET* MI_LSD_Run(*LSD_HANDLE* lsd_handle, *S16* *lsd_db_result);

Arguments

| Name | Description |
|---|---|
| lsd_handle | Pointer to the LSD_HANDLE |
| lsd_db_result | Pointer to the dBFS value |

Return value

| Return value | Description |
|---|---|
| MI_RET_SUCCESS | Success |
| MI_LSD_RET_INIT_ERROR | LSD Init error |

Requirement

Header files: mi_lsd.h
Library files: libLSD_Linux.a or libLSD_Linux.so

### MI_LSD_GetResult

Purpose

Get result of LSD

Function Prototype

*MI_RET* MI_LSD_GetResult(*LSD_HANDLE* lsd_handle, *S16* *lsd_result);

Arguments

| Name | Description |
|---|---|
| lsd_handle | Pointer to the LSD_HANDLE |
| lsd_result | Pointer to the result of loud sound detected |

Return value

| Return value | Description |
|---|---|
| MI_RET_SUCCESS | Success |
| MI_LSD_RET_INVALID_HANDLE | Invalid LSD handle |

Requirement

Header files: mi_lsd.h
Library files: libLSD_Linux.a or libLSD_Linux.so

# 4. DATA TYPE

## 4.1. Overview

| | |
|---|---|
| LSD_PARAMS | Define the audio sample rate and channel number of LSD |
| MI_RET | Define error code of LSD |

## 4.2. Structure Lists

### LSD_PARAMS

Description

Define the audio sample rate and channel number of LSD

Syntax

```
typedef struct {
      unsigned int sample_rate;
      unsigned int channel;
} LSD_PARAMS;
```

Member

| Member | Description |
|---|---|
| sample_rate | The sample rate of audio input |
| channel | Channel number |

## 4.3. Enumeration Lists

### MI_RET

Description

Define error code of LSD

Syntax

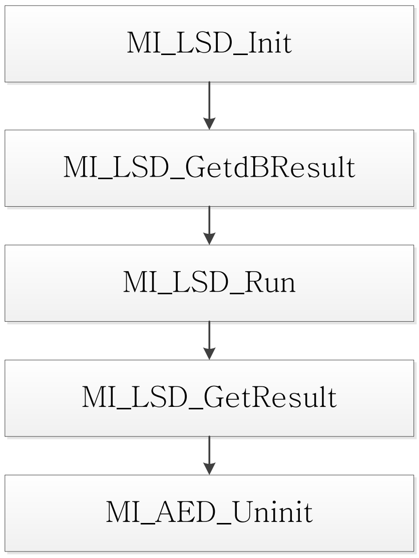
```
typedef enum {
      MI_RET_SUCCESS                    = 0x00000000,
      MI_LSD_RET_INIT_ERROR                  = 0x10000701,
      MI_LSD_RET_IC_CHECK_ERROR              = 0x10000702,
      MI_LSD_RET_INVALID_HANDLE              = 0x10000703,
      MI_LSD_RET_INVALID_SAMPLERATE          = 0x10000704
} MI_LSD_RET;
```

Member

| **Member** | **Description** |
|---|---|
| MI_RET_SUCCESS | Success |
| MI_LSD_RET_INIT_ERROR | LSD init error |
| MI_LSD_RET_IC_CHECK_ERROR | Incorrect platform check for LSD |
| MI_LSD_RET_INVALID_HANDLE | Invalid LSD handle |
| MI_LSD_RET_INVALID_SAMPLERATE | Invalid Sample rate of LSD |

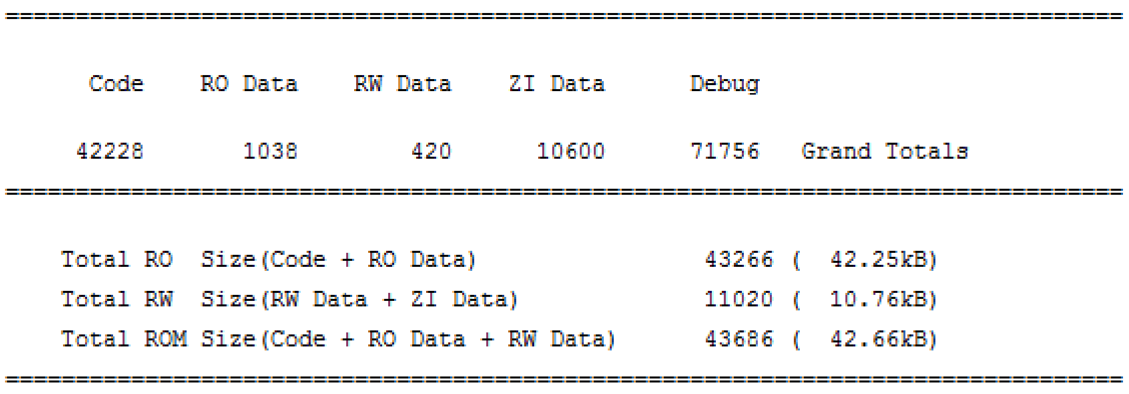
# 5. FLOW

## 5.1. Loud Sound Detection



MI_LSD_GetdBResult should be called before MI_LSD_Run for each frame

# 6. CODE/DATA SIZE INFORMATION



```
================================================================================

      Code     RO Data     RW Data     ZI Data      Debug

     42228        1038         420       10600      71756   Grand Totals
================================================================================

    Total RO  Size(Code + RO Data)                   43266 (  42.25kB)
    Total RW  Size(RW Data + ZI Data)                11020 (  10.76kB)
    Total ROM Size(Code + RO Data + RW Data)         43686 (  42.66kB)
================================================================================
```

Figure 4: Code/data size information

# 7. DRAME USAGE INFORMATION (WORKING BUFFER)

| Sample Rate of Audio Input | Buffer Size (bytes) |
|---|---|
| 8kHz | 0 |
| 16kHz | 16992 |
| 32kHz | 16992 |

# 8. CPU MIPS/CLOCK CYCLES ESTIMATION

- Loud sound detection

  run MI_LSD_GetdBResult , MI_LSD_Run every 32 msec

  I3, CPU freq = 400 MHz

  - 8 kHz/mono
    - 0.05 ms
  - 16 kHz/mono
    - 0.20 ms
  - 32 kHz/mono
    - 0.22 ms