Siren7 Library Reference Guide

V1.00.001

Publication Release Date: Aug. 2011

Support Chips:

ISD9160

Support Platforms:

NuvotonPlatform_Keil

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Table of Contents

1. Introduction

1.1.Siren7 Introduction

Siren7 provide a software solution for compression/decompression voice data in system. It contains the 16 bits fixed-point for encoder and decoder of ITU-T Recommendation G.722.1 including its Annex C. Siren7 implements the 7 KHz mode of G.722.1.

1.2.Siren7 Feature

The Siren7 includes following features:

- Encoder/Decoder voice data as G.722.1 format.
- Bit Rate from 4K to 32K bit per sample.
- Bandwidth 7KHz
- 2 of Synchronize channels
- Programming Features

1.3.Siren7 Limitation

Now the device only can support 2 channels of Siren7 at the same time. Because the resources, like ram size, computing power, time complexity, are not enough to run smoothly when adding one channel.
For speeding up the read/writer functions, it can use word (4 bytes) alignment for buffer.

2. APIs Specification

2.1.Data Type Definition

```
typedef struct tsSiren7_CODEC_CTL
{
    signed long    bit_rate;
    signed short   bandwidth;
    signed short   number_of_bits_per_frame;
    signed short   number_of_regions;
    signed short   frame_size;
}sSiren7_CODEC_CTL;

typedef struct tsSiren7_ENC_CTX
{
        signed short    history[MAX_FRAMESIZE];
        signed short    mlt_coefs[MAX_FRAMESIZE]; // this is not history data
}sSiren7_ENC_CTX;

typedef struct
{
        signed short seed0;
        signed short seed1;
        signed short seed2;
        signed short seed3;
}Rand_Obj;

typedef struct tsSiren7_DEC_CTX
{
        signed short    old_mag_shift;
#ifndef NO_FRAME_ERROR_CHECK
        signed short    old_decoder_mlt_coefs[MAX_DCT_LENGTH];
#endif
        signed short    old_samples[MAX_DCT_LENGTH>>1];
        Rand_Obj randobj;
        signed short    decoder_mlt_coefs[MAX_DCT_LENGTH]; // this is not history data
}sSiren7_DEC_CTX;
```

2.2.API Functions

LibS7Init

Prototype

```
void    LibS7Init(
    sSiren7_CODEC_CTL *sCodecCtl,
    Word32 w32BitRate,
    Word16 w16Bandwidth
);
```

Description

Initialize encode/decoder’s control values.

Parameter

sCodecCtl

Information of encode/decoder’s control value.

w32BitRate

Bit Rate, range from 4K~32K.

w16Bandwidth

7K.

Include

LibSiren7.h

Return Value

None

LibS7EnBufReset

Prototype

```
void LibS7EnBufReset(
    signed short ssFrameSize,
    sSiren7_ENC_CTX *sS7EncCtx
);
```

Description

Reset encoder’s history buffer values to zero.

Parameter

ssFrameSize

Encoder/Decoder's control value frame_size.

sS7EncCtx

Structure of storing encoder's history value.

Include

LibSiren7.h

Return Value

None

LibS7Encode

Prototype

```
void    LibS7Encode(
    const sSiren7_CODEC_CTL *sCctl,
    sSiren7_ENC_CTX *sS7EncCtx,
    Word16 *w16InData,
    Word16 *w16OutData
);
```

Description

Encode the raw data as Siren7 format.

Parameter

sCctl

Information of encoder’s control value.

sS7EncCtx

Structure of storing encoder's history value.

w16InData

Input encoding data.

w16OutData

Output encoded data.

Include

LibSiren7.h

Return Value

None

LibS7DeBufReset

Prototype

```
void    LibS7DeBufReset(
    signed short ssFrameSize,
    sSiren7_DEC_CTX *sS7DecCtx
);
```

Description

Reset decoder’s history buffer values to zero.

Parameter

ssFrameSize

Encoder/Decoder's control value frame_size.

sS7DecCtx

Structure of storing decoder's history value.

Include

LibSiren7.h

Return Value

None

LibS7Decode

Prototype

```
void    LibS7Decode(
    const sSiren7_CODEC_CTL *sCctl,
    sSiren7_DEC_CTX *sS7DecCtx,
    Word16 *w16InData,
    Word16 *w16OutData
);
```

Description

Decode the compression data as normal format.

Parameter

sCctl

Information of decoder’s control value.

sS7DecCtx

Structure of storing decoder's history value.

w16InData

Input decoding data.

w16OutData

Output decoded data.

Include

LibSiren7.h

Return Value

None

3. Revision History

Version	Date	Description
V1.00.001	Aug.30, 2011	● Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.