BLE 智能语音编解码指导手册

Latest Version: V1.0

Date: 2018-07-20

Declaration

Disclaimer

Information given in this document is provided just as a reference or example for the purpose of using Actions' products, and cannot be treated as a part of any quotation or contract for sale.

Actions products may contain design defects or errors known as anomalies or errata which may cause the products' functions to deviate from published specifications. Designers must not rely on the instructions of Actions' products marked "reserved" or "undefined". Actions reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

ACTIONS DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL EXPRESS OR IMPLIED WARRANTIES OF MERCHANTABILITY, ACCURACY, SECURITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY AND THE LIKE TO THE INFORMATON OF THIS DOCUMENT AND ACTIONS PRODUCTS.

IN NO EVENT SHALL ACTIONS BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION FOR LOST OF DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND ARISING FROM USING THE INFORMATON OF THIS DOCUMENT AND ACTIONS PRODUCTS. REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF ACTIONS OR OTHERS; STRICT LIABILITY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER ACTIONS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR NOT.

Actions' products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of Actions and further testing and/or modification will be fully at the risk of the customer.

Ways of obtaining information

Copies of this document and/or other Actions product literature, as well as the Terms and Conditions of Sale Agreement, may be obtained by visiting Actions' website at: http://www.actions-semi.com or from an authorized Actions representative.

Trademarks

The word "Actions" and the logo are the trademarks of Actions Se 麦克风 onductor Co., Ltd, and Actions (Zhuhai) Technology Co., Limited is authorized to use them. Word "炬芯" is the trademark of Actions (Zhuhai) Technology Co., Limited. Names and brands of other companies and their products that may from time to time descriptively appear in this document are the trademarks of their respective holders, no affiliation, authorization, or endorsement by such persons are claimed or implied except as may be expressly stated therein.

Rights Reserved

The provision of this document shall not be deemed to grant buyers any right in and to patent, copyright, trademark, trade secret, know how, and any other intellectual property of Actions or others.

Miscellaneous

Information contained or described herein relates only to the Actions products and as of the release date of this publication, abrogates and supersedes all previously published data and specifications relating to such products provided by Actions or by any other person purporting to distribute such information.

Actions reserves the rights to make changes to information described herein at any time without notice. Please contact your Actions sales representatives to obtain the latest information before placing your product order.

Additional Support

Additional products and company information can be obtained by visiting the Actions website at:
http://www.actions-semi.com

Revision History

Version	Revision Content	Date
V1.0	初始版本	2018-07-20

Contents

一、语音规格：

1.1 采样率：8/16KHz

1.2 Sample：16bit

1.3 声道：单声道 mono

1.4 码率：16kbps, 32kbps, 64kbps

1.5 组合：16KHz-16kbps/32kbps/64kbps, 8KHz-32kbps

二、语音编码：

2.1 编码规格：

Encoder	Ratio	Codec	SampleRate	BitRate	LIB Name	CPU
ASC_I	16:1	10	16KHz	16kbps	asc_enc1.lib	32MHz
ASC_II	8:1	20/21	16KHz	32kbps	asc_enc2.lib	32MHz
ASC_III	4:1	30/31	8/16KHz	32/64kbps	asc_enc3.lib	32MHz

2.2 编码接口

2.2.1 short ASC_I_Encode_Init(short codec);

```
/*
**      input, short codec: 10,    @16:1
**      return Sample length,    @320 (x 2Bytes)
*/
```

2.2.2 short ASC_I_Encoder(short *indata, short *outdata, short len);

```
/*
**      input, short *indata, 320*2 bytes,    @16KHz PCM data
**      output, short *outdata, 20*2 bytes,
**      input, short len: 320
**      return Bitstream length,    @20 (x 2Bytes)
*/
```

2.2.3 short ASC_II_Encoder_Init(short codec);

```
/*
**      input, short codec: 20 ,    @8:1
**      return Sample length,    @80 (x 2Bytes)
*/
```

2.2.4 short ASC_II_Encoder(short *indata, short *outdata, short len);

```
/*
**      input, short *indata, 80*2 bytes,    @16KHz PCM data
**      output, short *outdata, 10*2 bytes,
**      input, short len: 80
**      return Bitstream length,    @10 (x 2Bytes)
*/
```


2.2.5 short ASC_II_B_Encoder_Init(short codec);

```
/*
**      input, short codec: 21,    @8:1
**      return Sample length,    @80 (x 2Bytes)
*/
```

2.2.6 short ASC_II_B _Encoder(short *indata, short *outdata, short len);

```
/*
**      input, short *indata, 80*2 bytes,    @16KHz PCM data
**      output, short *outdata, 10*2 bytes,
**      input, short len: 80
**      return Bitstream length,    @10 (x 2Bytes)
*/
```

2.2.7 short ASC_III_Encoder_Init(short codec);

```
/*
**      input, short codec: 30,    @4:1
**      return Sample length,    @40 (x 2Bytes)
*/
```

2.2.8 short ASC_III_Encoder(short *indata, short *outdata, short len);

```
/*
**      input, short *indata, 40*2 bytes,    @8KHz PCM data
**      output, short *outdata, 10*2 bytes,
**      input, short len: 40
**      return Bitstream length,    @10 (x 2Bytes)
*/
```

2.2.9 short ASC_III_Encoder_Init(short codec);

```
/*
**      input, short codec: 31,    @4:1
**      return Sample length,    @40 (x 2Bytes)
*/
```

2.2.10 short ASC_III_Encoder(short *indata, short *outdata, short len);

```
/*
**      input, short *indata, 40*2 bytes,    @16KHz PCM data
**      output, short *outdata, 10*2 bytes,
**      input, short len: 40
**      return Bitstream length,    @10 (x 2Bytes)
*/
```


2.3 编码调用说明

```
#include <stdio.h>
#include <stdlib.h>
#include "asc_x_encoder.h"



#define LEN      320
#define LEN2     80


short       input[LEN] = {0};
short       output[LEN2] = {0};
FILE          *fp_in,*fp_out;
short       Samplelen,Bitstreamlen, codec;

main( int argc, char *argv[ ])
{
      if(argc!=3)
      {
            printf("Please input :[input filename][output filename] \n");
            exit(1);
      }

      if( (fp_in=fopen(argv[1],"rb"))==NULL )
      {
            printf("Can't read file !\n");
            exit(1);
      }

      if( (fp_out=fopen(argv[2],"wb"))==NULL )
      {
            printf("Can't write file !\n");
            exit(1);
      }

      codec = 10; //20,21,30,31
      Samplelen = ASC_X_Encoder_Init(codec);
      if(Samplelen == 0)
      {
            printf("ASC_X_Encoder_Init error !\n");
            return 0;
      }
```



```
    while( fread(input, sizeof(short), Samplelen, fp_in) == (unsigned)Samplelen)
    {

        Bitstreamlen = ASC_X_Encoder(input, output, Samplelen);
        fwrite(output, sizeof(short), Bitstreamlen, fp_out);

    }
    fcloseall();
    return 0;
}
```


三、语音解码：

3.1 解码规格：

Decoder	Ratio	Codec	SampleRate	BitRate	LIB Name
ASC	16:1	1	16KHz	16kbps	libasc_dec.so
ASC	8:1	2/5	16KHz	32kbps	libasc_dec.so
ASC	4:1	3/4	8/16KHz	32/64kbps	libasc_dec.so

3.2 解码接口

3.2.1 short ASC_Decoder_Init(short codec)

```
/*
**      input, short codec, 1/2/3/4/5
**      return Bitstream length,    @ n(x 2Bytes)
*/
```

3.2.2 short ASC_Decoder (short *indata, short *outdata, short len, short codec)

```
/*
**      input, short *indata,
**      output, short *outdata, @16KHz PCM data
**      input, short len,
**      input, short codec, 1/2/3/4/5
**      return Sample length, @n (x 2Bytes)
*/
```

3.3 解码调用说明

```
#include <stdio.h>
#include <stdlib.h>
#include "asc_decoder.h"



#define LEN     320
#define LEN2    80

short  inbuf[LEN2] = {0};
short  outbuf[LEN] = {0};

FILE        *fp_in,*fp_out;
short       n;
short       Samplelen,Bitstreamlen;
short       codec;
```



```
main( int argc, char *argv[ ])
{

 if(argc!=4)
 {
      printf("Please input :[input filename][output filename] [codec]\n");
 }

 if( (fp_in=fopen(argv[1],"rb"))==NULL )
 {
      printf("Can't read file !\n");
 }

 if( (fp_out=fopen(argv[2],"wb"))==NULL )
 {
      printf("Can't write file !\n");
 }

 codec = atoi(argv[3]);

 Bitstreamlen = ASC_Decoder_Init(codec);
 if(Bitstreamlen == 0)
 {
      printf("ASC_Decoder_Init error !\n");
      return 0;
 }

 while( fread(inbuf, sizeof(short), Bitstreamlen, fp_in) == (unsigned)Bitstreamlen)
 {

      Samplelen = ASC_Decoder(inbuf, outbuf, Bitstreamlen, codec);

      fwrite(outbuf,sizeof(short),Samplelen,fp_out);

 }

 fcloseall();
 return 0;
}
```


