To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

R8C Family, H8/300H Tiny Series, and M16C/Tiny Series

M3S-SEN-Tiny: Sound Data Compression Software For Tiny Microcontrollers

Introduction

This document explains the usage of the sound data compression software library along with a sample program.

Target device

Tiny microcomputer (R8C Family, H8/300H Tiny Series, and M16C/Tiny Series)

Contents

1. Library functions

This section explains the specifications of the library functions.

1.1 Initialization (INIT_ENC)

Prototype

```
void INIT_ENC(short *inp, unsigned char *cdp, unsigned char bitn );
```

Explanation

This function initializes the sound data that is to be compressed.

This function has to be executed only once for completing the initialization before compressing consecutive ADPCM data.

The short pointer *inp points to the memory location which stores the input PCM data to be compressed. The unsigned char pointer *cdp points to the memory location where the compressed output (ADPCM) data will be stored. The argument bitn specifies the encoding specification. For this, only the value 4 is currently supported. Before executing this function, declare the input (*inp) and output (*cdp) pointer variables that are to be passed as arguments. Value need not be set to these memory locations during this initialization.

Table 1: Arguments for the initialization function

Argument	Type	Explanation
inp	short *	Top address of PCM data area.
cdp	unsigned char *	Top address of ADPCM data (compressed data) area.
bitn	unsigned char	Conversion size of ADPCM data (Currently only value 4 is supported).

【 attention 】This function changes neither the input, nor the output nor the size.

Return value

none

1.2 Compression (enc16to4)

Prototype

```
int enc16to4(short smpln);
```

Explanation

This function encodes (compresses) 16-bit PCM data to 4-bit ADPCM data. Encoded data is stored in the memory area initialized in the function INIT_ENC. The argument “smpln” specifies the number of bits into which ADPCM data is to be encoded. Value of “smpln” must be in multiples of 4. The size of the output memory location depends upon the “smpln” variable.

Return value

int - If the conversion size of the ADPCM data is an odd number, -1 is returned, else 0 is returned

1.3 Refresh Input (refresh_encipbuf)

Prototype

```
void refresh_encipbuf(short *inp);
```

Explanation

This function refreshes internal PCM input data buffer. The argument “inp” specifies the memory area where input data is stored. When this function and function enc16to4 are called in succession, enc16to4 encodes from the top of the memory area pointed out by “inp”.

Return value

none

1.4 Refresh Output (refresh_encopbuf)

Prototype

```
void refresh_encopbuf(unsigned char *cdp)
```

Explanation

This function refreshes the output ADPCM data buffer. The argument “cdp” specifies the memory area where ADPCM data is stored. When this function and function enc16to4 are called in succession, enc16to4 stores the encoded data from the top of the memory area pointed out by “cdp”

Return value

none

2. Sample program

This section explains the sample program for ADPCM data compression. The sample program is in the form of a High-Performance Embedded Workshop workspace. Change the initialization of the microcomputer and its peripherals according to the system in use.

2.1 Outline

In the main function, the function record_message is called. This function initializes the encoder and also starts the timer. Thereafter, a sample of the input signal is taken every 125µsec (for a sampling frequency of 8 KHz). The input samples are converted by the ADC and passed to the encoder for ADPCM conversion. This ADPCM data is stored in the RAM. The conversion continues till maximum data length available for storage of ADPCM encoded data is over.

The size of the ADPCM data is defined by the following macro.

Size of ADPCM data Macro name: MAX_DATA_LENGTH

2.2 Flow

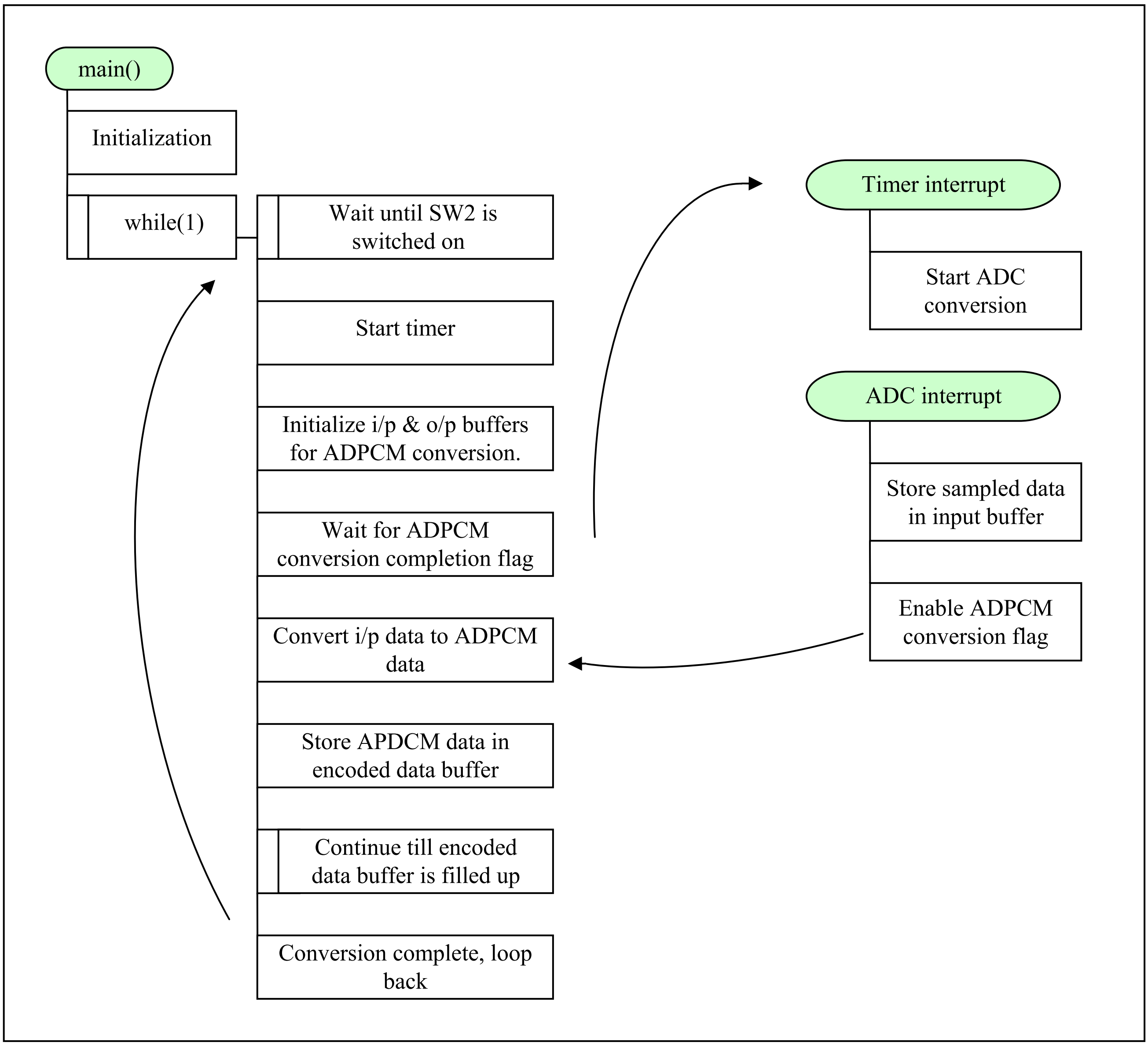


Figure 1: Flow of sample program

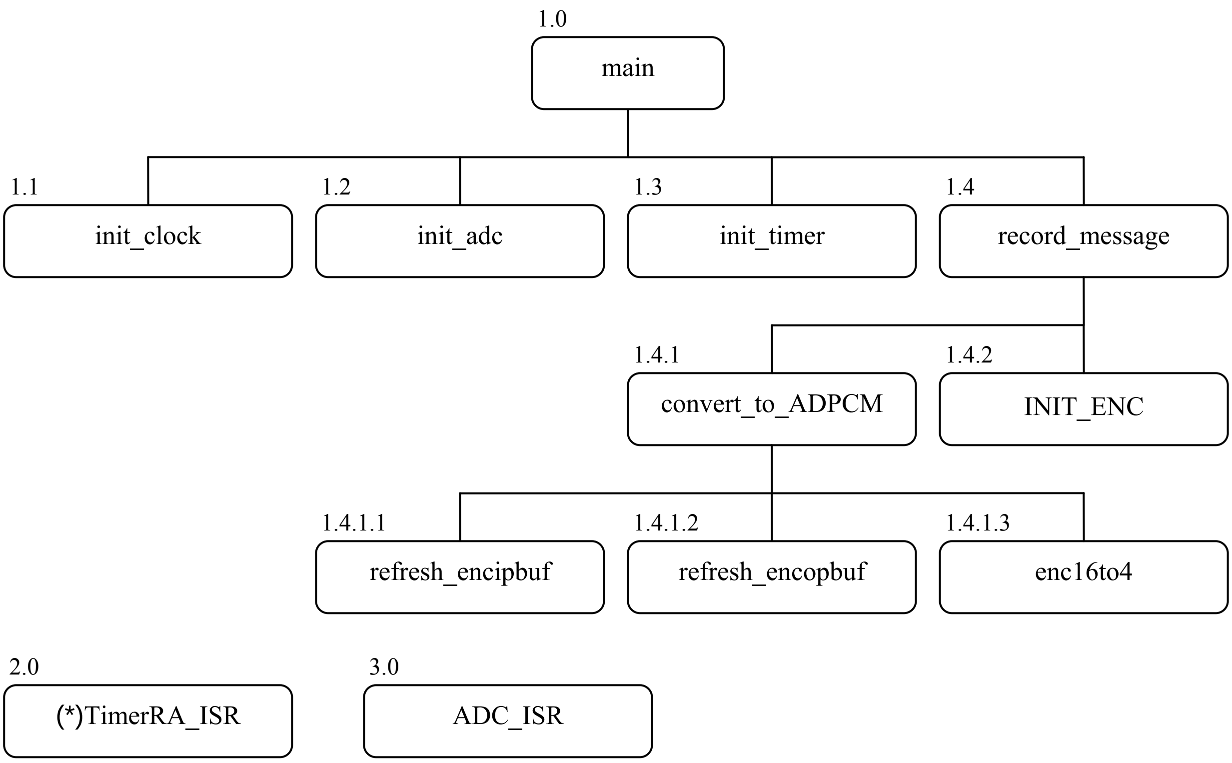
2.3 Function list

Table 2: List of functions in the sample program

No.	Function name	Outline
1.0	main	The input voice signal is sampled to produce PCM data. This PCM data is then further compressed using ADPCM. The ADPCM data is stored in RAM.
1.1	init_clock	The clock of the microcomputer and other clock related registers are initialized.
1.2	init_adc	ADC is initialized for conversion of analog input sound signal into digital PCM data.
1.3	init_timer	The timer for interrupt processing is set. The interrupt processing starts according to the sampling rate of the sound data.
1.4	record_message	Starts the timer and the ADC and checks for conversion completion
1.4.1	convert_to_ADPCM	Converts the PCM data into ADPCM data by calling the library functions.
1.4.2	INIT_ENC	The sound data for compression is initialized. - This is a library function
1.4.1.1	refresh_encipbuf	Refreshes the input PCM data buffer - This is a library function
1.4.1.2	refresh_encopbuf	Refreshes the output ADPCM data buffer - This is a library function
1.4.1.3	enc16to4	Encodes the 16-bit PCM data into 4-bit ADPCM data. - This is a library function
2.0	(*)TimerRA_ISR	Interrupt Service Routine for Timer. Takes a sample of the input signal every cycle of the sampling frequency.
3.0	ADC_ISR	Interrupt Service Routine for Timer. Passes the PCM data for conversion to ADPCM after every four samples are taken.

(*) Function name for the sample software for R8C.
"TimerB1_ISR" in the sample program of H8/Tiny. "TimerA0_ISR" in the sample program of M16C/Tiny.

2.4 Function chart



(*)"TimerB1_ISR" in the sample program of H8/Tiny.

"TimerA0_ISR" in the sample program of M16C/Tiny.

Figure 2: Function chart

2.5 Folder composition in workspace

```
SEN_sample_***Tiny                  Workspace directory (*** is microcomputer name)
|
|
|-- Type name of microcomputer      Project directory
        |
        |-- Debug                   Configuration directory
        |
        |-- Debug_***_E8_SYSTEM     Configuration directory (*** is microcomputer name)
        |
        |-- lib                     Sound compression library storage directory
        |
        |-- Release                 Configuration directory
        |
        |-- src                     Sample source storage directory.
        |                           The header file "vmadpcm_encoder.h" is included.
        |
        |-- HEW_files               High-Performance Embedded Workshop auto-generated files
```

3. Sample software usage

This section explains the details related to sample software execution.

3.1 Sample software execution procedure

3.1.1 Input analog signal connection

The sample software is configured to accept the signal on the channel AD0** of the Analog to Digital Converter (ADC) as the input analog signal. The user input signal needs to be connected to the channel AD0 of the ADC. This channel can be accessed using the JA1 Application Header on the RSK*. The input channel AD0 of the microcontroller is connected to pin no. 9 of JA1. The mapping of the pins is as follows:

JA1 Pin	JA1 Header Pin Name	RSK* Signal Name	Microcontroller	Microcontroller Pin
9	ADC0	AD0	R8C	38
9	AD0	AN0	H8/Tiny	62
9	ADC0	AD0	M16C/Tiny	45

(*)RSK refers to

Renesas Solutions Kit RSKR8C25 – ROK521256C000BR (for R8C)

Renesas Solutions Kit RSKH836077– ROK436079S000BE (for H8/Tiny)

Renesas Solutions Kit RSKM16C26A – ROK33026AC001BR (for M16C/Tiny)

(**) AN0 in case of H8/Tiny

3.1.2 Switch and LED operation during execution

- Build the sample software workspace and download the x30 file to the RSK.
- Connect the input signal as mentioned in the previous section
- After the "Reset Go" button is clicked, program starts running. The program is running waiting for input from the user.
- When the switch SW2 on the RSK is pressed, the ADPCM conversion of the input signal begins. The conversion continues till the allocated maximum data length for the ADPCM data storage is filled up.
- After SW2 is pressed, LED3(*) on the RSK starts glowing indicating that ADPCM data conversion is in progress. After the conversion is completed, the LED3(*) switches off indicating conversion is complete.
- After conversion is complete, the program again loops back and continues running waiting for the next switch press by the user.
- The ADPCM encoded data is stored in the RAM. The ADPCM data retrieval procedure is given in the next section.

  (*)LED2 in case of RSKM16C26A.

4. Sound data retrieval procedure

This section explains sound data retrieval procedure for the sample program. The sound data encoded by the sample program can be retrieved using the following procedure.

4.1 Uploading of sound data

4.1.1 Preparation of binary (.bin) file

The encoded data array g_EncodedData can be seen in the memory window in High-Performance Embedded Workshop using the label _g_EncodedData. Upload this entire encoded data array and save it as a binary file. Please refer to the manual of High-Performance Embedded Workshop for the uploading procedure.

Table 3: Upload specifications

Format	Binary
Access Size	1
Start Address	_g_EncodedData
End Address	Calculate the end address as per the length of the encoded data array.

4.1.2 Decompression of sound data

Make the Wave file (.wav) from the ADPCM data (binary form) using the bundled "ADPCM_TOOL (ADPCM.exe)". Please refer the manual of the tool “ADPCM_TOOL” for its usage.

Table 4: Decode specifications

File Format	Binary
Conversion Type	Decode (ADPCM => Wave)
Sample Rate	8 KHz

4.2 Change in sample program

4.2.1 MAX_DATA_LENGTH

The value of the macro MAX_DATA_LENGTH should be set in encoder_sample.h file of the sample program. It should be set according to the size of the RAM available for storage of ADPCM encoded data. More is the size available for ADPCM data storage, more will be the length of the sound sample that can be encoded.

```
[E.g. when available size for storage of ADPCM data is 1499 bytes, setting will be]
#define MAX_DATA_LENGTH        (1499u)    /* Max. size of ADPCM encoded data */
```

4.3 Verification of sound data

The wave file generated from the uploaded ADPCM binary file can be verified to be similar in nature to the input file.

5. Library characteristic

This section explains the memory occupation and the speed performance (reference values) of the library functions.

5.1 Occupied memory size

Table 4 Occupied memory

Micro-computer	Mode/ Option	ROM	RAM(*)	Stack			
				INIT_ENC	enc16to4	refresh_ encipbuf	refresh_ encopbuf
R8C	R8C	555	0	5	20	5	5
	R8CE	584	0	5	20	5	5
H8/Tiny	Normal	574	0	6	30	2	2
	Advanced	644	0	30	36	4	4
M16C/Tiny	-	648	0	9	26	7	7

Unit: Byte

(*)The structure variable area (10 bytes or more) for the library function is necessary.

5.2 Compression processing speed (reference value)

Table 5 Compression processing speed (reference value)

Microcomputer	Mode/ Option	Compression processing time(μsec) (size=4)
R8C	R8C	76
	R8CE	78
H8/Tiny	Normal	83
	Advanced	114
M16C/Tiny	-	59

* The above values are reference values at an operating frequency of 20 MHz for the given microcomputers.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

Rev.	Date	Description	
		Page	Summary
2.00	Mar.02.09	—	First edition (M3S-SEN-Tiny Ver.1.00 Release01E)

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.