TOSHIBA

# Speech Codec Software IP

## [TMSI-SC]

## Ver 1.00

## Development Manual

## Provisional

**February 2003**

Note: This Development Manual is provisional. Please note that the product names, functions, and function names contained in this document are subject to change.

**Toshiba LSI System Support Co., Ltd.**

The information contained herein is subject to change without notice.

The information contained herein is presented only as a guide for applications of our products. No respensibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or othrewise under any patent or patent rights of TOSHIBA or others.

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the “Handling Guide for Semiconductor Devices”, or “TOSHIBA Semiconductor Reliability Handbook”, etc.

The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).
These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage includes atomic energy control instruments, airplane or spaceship instrument, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer’s own risk.

The products described in this document may include products subject to the foreign exchange and foreign trade laws.

001227DTA1

## Preface

Thank you very much for making use of Toshiba Software IP Speech Codec products.
This product (provisional version as of January 2002) is the software that supports the development of programs for Toshiba CISC microcontrollers. This manual describes the functions of source programs and how to use these functions.

## About the Product

This product is the software IP for Toshiba 870 and 900 Family CISC microcontrollers.
When you develop applications for equipment having an 870 or 900 Family microcontroller as a processing MPU, this embedded software enables you to easily incorporate the codec (signal coding) function for speech recording/playback.
Please note that this product is provided on the assumption that applications are developed on a PC.

## Technical Support

This manual is written for users who have a basic knowledge of microcontrollers and C language. The descriptions contained in this manual are therefore concise assuming that they will be used by application programmers. If you need further explanation on any points in this manual, please contact your nearest Toshiba office.

**This manual is organized into the following chapters:**

Contents

APPENDIX

# 1 Introduction

## 1.1 Purpose

In recent years the equipment incorporating microcontrollers has seen demands for increasingly diverse functions. In keeping with such a trend, a variety of products handling speech data as a human interface are now available on the market.
At the same time, microcontroller engineers are faced with the challenges of developing devices with fewer parts and of increasing the efficiency and speed of their work.
At Toshiba efforts have been made to develop speech technology software to be embedded in CISC microcontrollers. One of the results of such efforts is this product, a software component for embedded use in speech recording/playback equipment using 870/900 Family microcontrollers.
In this product, the ROM and RAM sizes are kept to a minimum so as to be adaptable to embedded systems, and the CPU load is reduced considering the effect on application control. This product enables the user to easily configure a speech output interface in equipment using 870/900 Family microcontrollers.

## 1.2 Outline of How to Use This Product

The names of functions used in this source program are made available in this manual. When designing software for your application, prepare the timer interrupt for creating a speech sampling cycle. The functions are to be called in this timer interrupt.
Because the functions of this source program are not reentrant, nested calls must be disabled by a user application.

## 1.3 Provision of the RAS Source Program

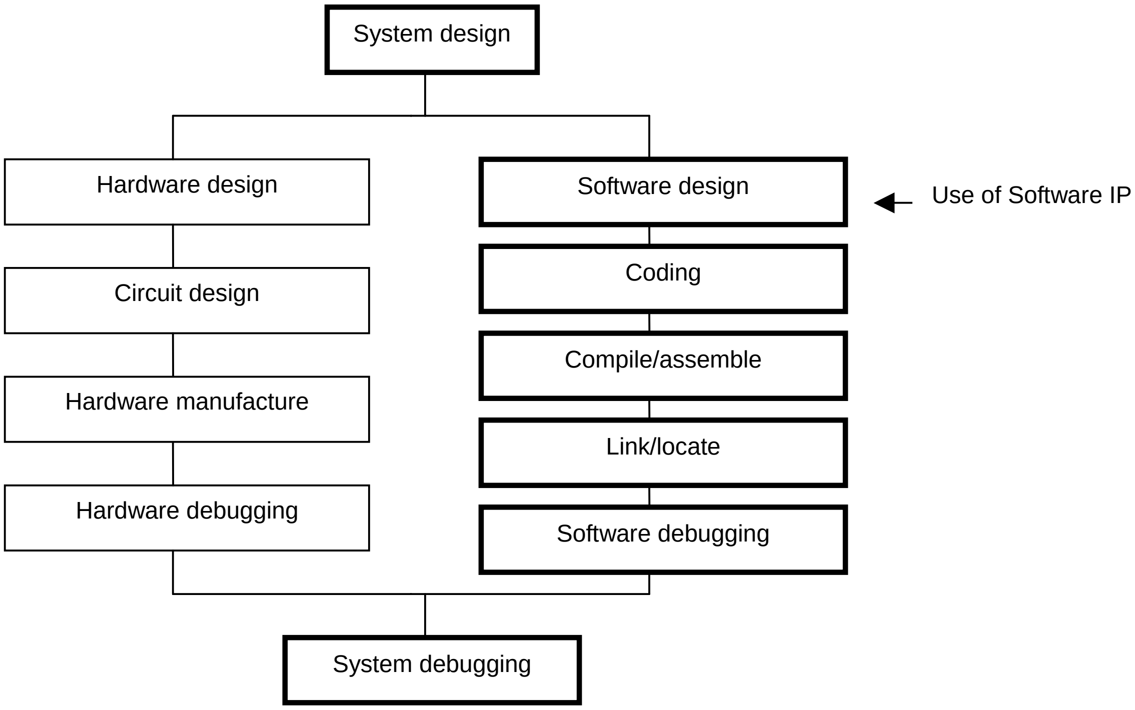
The copyright on the RAS (Reference Application Software) is owned by Toshiba Corporation (hereafter called “Toshiba”). Toshiba grants your company (hereafter called the “Company”) without charge the untransferable “license” for using this software for the purpose of creating applications for your designated microcomputer. Therefore, Toshiba shall use this software solely for the microcomputer to be supplied to the Company, and the Company shall not disclose or grant a license on this software or any modification thereof to any third parties.
The above statement is aimed at protecting the software (intellectual property) by the both parties. We would appreciate your understanding on this matter.

# 2 Developing Speech Codec Software IP Applications

## 2.1 Developing Microcomputer Systems Using the Software IP

In developing systems using microcomputers, the process of developing the software used to control the microcomputer is very important. The outline of the overall system development process is shown below.

**Microcomputer Application System Development Process**



As microprocessors become more sophisticated and increasingly diverse applications are demanded, programmers of today's microcomputer software applications face the need for streamlining the development process so as to produce high-performance software within a shorter product cycle. To achieve this goal Toshiba offers the software components (including middleware and driver software) called "software IP", which realize particular functions by software.

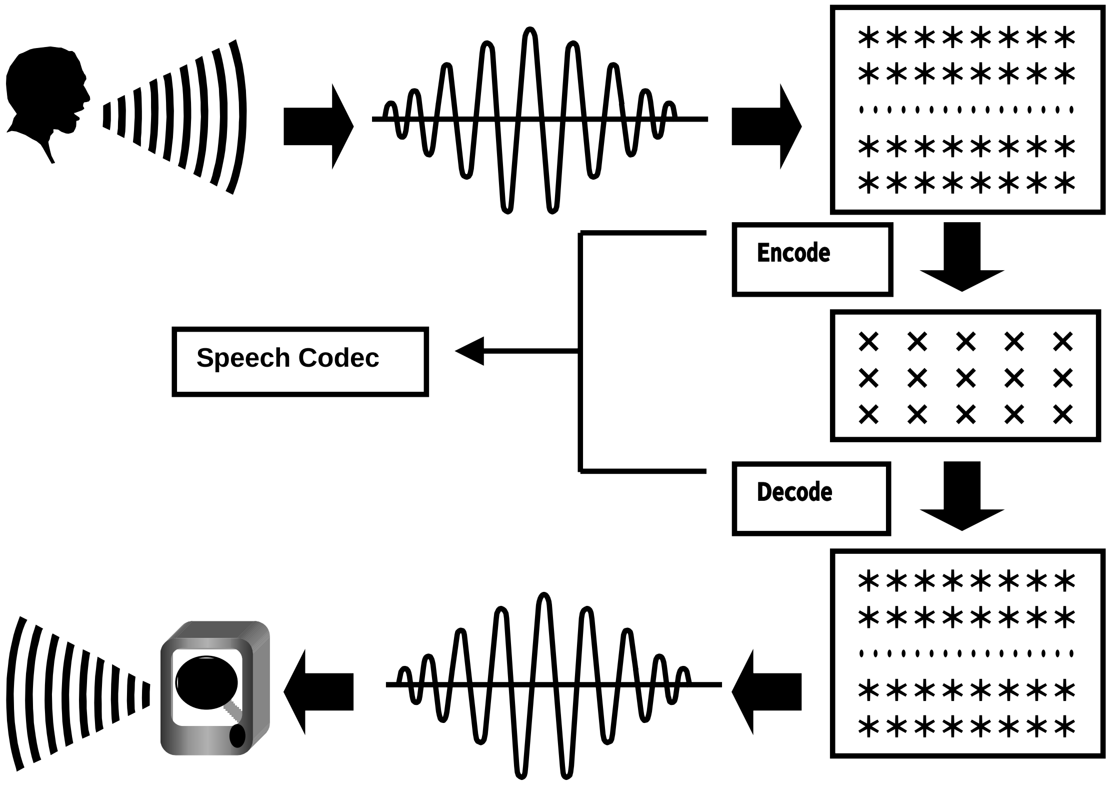
Use of software IP enables application software programmers to easily realize desired functions by simply selecting the appropriate software IP and incorporating it into the application to be developed.

This product is the software IP for the speech codec function; by using this software IP the speech codec function can be easily added to your system application.

## 2.2 What is the Speech Codec?

When speech data in analog waveform is quantized and digitalized by an A/D converter, frequent sampling improves the sound quality but the amount of data becomes enormous. On the other hand, less frequent sampling creates a smaller amount of data but the sound quality is adversely affected. Also, the speech signal contains many redundant sounds and the sounds that are audible to humans fall within a certain range.

**Conceptual Illustration of Speech Codec**



The speech codec technology offers a solution to the above problems. In the speech codec, speech data is sampled at intervals that secure good sound quality and this data is then encoded into compressed format. This makes it easier to store and transfer speech data. The encoded data can be decompressed (decoded) by a D/A converter so that the original sound (analog waveform) can be reproduced.

## 2.3 Examples of Speech Codec Technology Applications

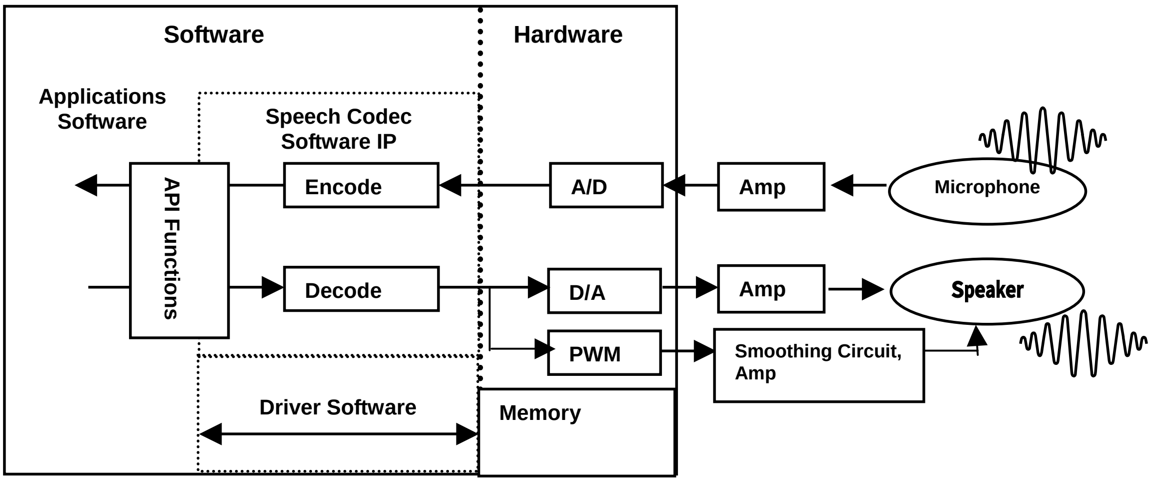
The speech codec technology can be used for equipment requiring speech recording/playback including the following:

1) Home appliances
2) Toys
3) Alarm devices
4) Health appliances
5) Other products incorporating 870/900 Family microcontrollers

## 2.4 Applications using the Speech Codec Software IP

The following shows a conceptual block diagram of how to configure a system including a user application and I/O devices.

**Conceptual Block Diagram of How the Speech Codec Software IP Is Used**



The process flow shown in the above diagram is briefly described here.

**(1) Speech data input**
Speech data is captured from a microphone as an analog value and is amplified by an amplifier.

**(2) AD conversion**
The input analog speech signal is quantized and converted into a digital signal (PCM signal).

**(3) Combination of a user application and the software IP API functions**
The API functions are used to invoke or abort the software IP. For details, see Chapter 5 “Specifications of the API Functions”.

**(4) DA conversion**
The digital signal is decoded and converted into an analog speech signal. Whether to use the internal D/A converter or PWM output can be selected.

# 3 Specifications of the Speech Codec Software

## 3.1 Outline of the Specifications

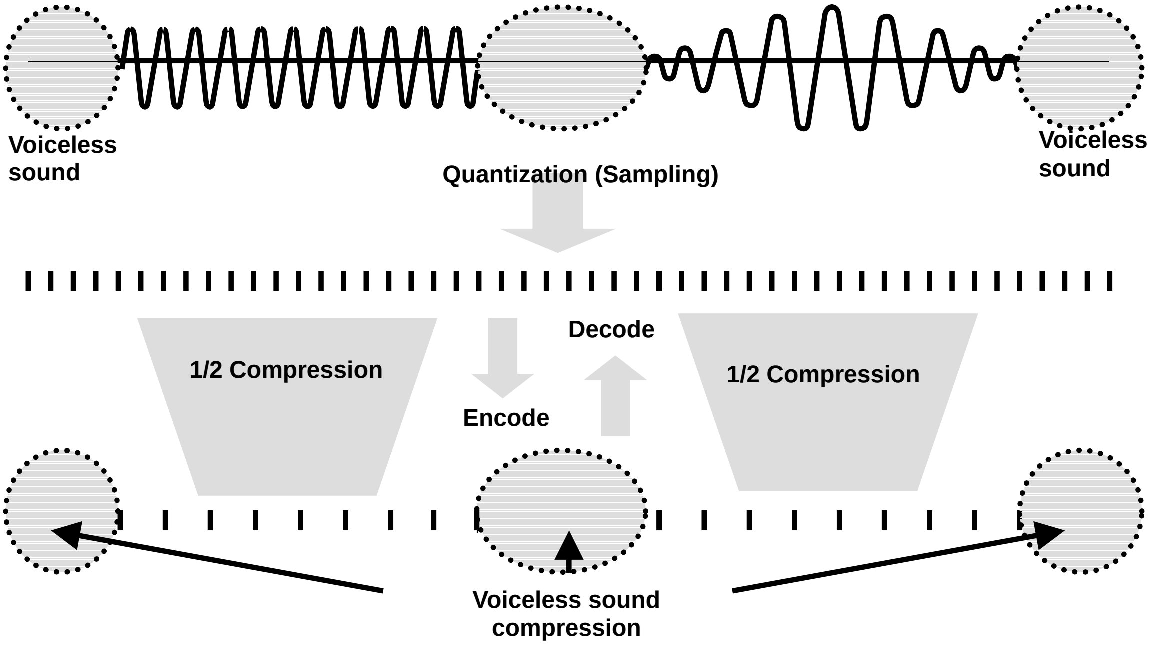
(1) **Decoder Software**

The decoder software decodes the speech data that is compressed with development tools on the PC.

(2) **Encoder Software**

The encoder software encodes the analog value that is input from the A/D converter (hardware incorporated in the microcontroller).

**Block Diagram of the Comparison Algorithm**



Speech data in analog waveform is sampled and converted into digital format by an A/D converter and quantized as PCM data. Then, all the PCM data is compressed into half of its original size by the Toshiba proprietary algorithm.

To store speech data in ROM, the data size is further reduced by compressing voiceless sound. (The compression ratio varies with the length of voiceless sound.)

## 3.2 Package Contents

The product package contains the following items.
【Package contents】

| Item | Quantity |
|---|---|
| 3.5-inch floppy disk | 1 |
| Instruction Manual | 1 |

## 3.3 Operating Environment

### 3.3.1 Applicable CPUs

Toshiba 870 and 900 Family CISC microcontrollers.

### 3.3.2 Memory Capacity and CPU Load Ratio

The required ROM/RAM size and the CPU load ratio during operation when this software is incorporated are shown below.

**(1) Decoder Software Specifications**

| | |
|---|---|
| Program space | 150 bytes |
| Work space | 8 bytes |
| CPU load ratio | 30% |

* Sampling frequency: 6 KHz, CPU: TMP86CS44 16 MHz

**(2) Encoder Software Specifications**

| | |
|---|---|
| Program space | 200 bytes |
| Work space | 10 bytes |
| CPU load ratio | 50% |

* Sampling frequency: 6 KHz, CPU: TMP86CS44 16 MHz

## 3.4 Requirements for Speech Data (Guaranteed Specifications)

This software IP is a software library for the speech codec processing unit to be incorporated into user applications. To use this software IP and maintain the sound quality, the following specifications must be observed when creating speech data.

**Required specifications**

- Sampling frequency : D/A output 8 KHz ~ 6 KHz
  PWM output 8 KHz
- Quantization bit count : 8 bits/sample

## 3.5 Description of the Sample Source Code

This section provides the specifications of the sample source code included with this product package and briefly describes how the sample source code works.
The sample application software was created and its operation was verified using the TMP86PS44 (16 MHz) evaluation board. Customize this sample application software according to your MCU.

### 3.5.1 Specifications of the Sample Source Code

**(1) Recording mode**

The following 3 types of speech output are played back repeatedly.
There is an interval of 3 seconds between each speech output.

(1)Push record key
↓
(2) " Recording start Please start recording your message after beep sound."
↓
(3) Beep sound
↓0.5s
(4)10sec record with compression(No key accept)

**(2) Playback mode**

(1) Push playback key
↓
(2) " Playback"
↓ 3s
(3) Beep sound
↓0.5s
(4) Playback recorded voice

### 3.5.2 The table below lists the sample source files.

**Package Sample Source Files**

| Folder name | File name | Description |
|---|---|---|
| Mcu | Io86x44.c | I/O header file for the TMP86CS44. |
| Mcu | Io86x44.c | I/O definition file for the TMP86CS44. |
| Mcu | Stc86x44.c | Startup routine. |
| App | main.c | Main routine file for sample source code operation. (Example of using API) |
| App | main.h | Header file for main.c |
| App | voice.c | Speech related function file.(Example of using API) |
| App | voice.h | Header file for Voice.c |
| App | timer.c | Timer related function file.(Example of using API) |
| App | timer.h | Header file for timer.c |
| App | key.c | Key related function file.(Example of using API) |
| App | key.h | Header file for key.c |
| App | e2prom.c | E2prom related function file.(Example of using API) |
| App | e2prom.h | Header file for E2prom.c |
| App | buzzer.c | Buzzer related function file.(Example of using API) |
| App | buzzer.h | Header file for buzzer.c |
| App | Typedef.h | Type definition file. |
| App | Voicetbl.h | ROM table file for the above 2 types of speech data. |
| Lib | Tvoiceip.c | Library of speech API functions. |
| Lib | Tvoiceip.h | Header file for speech API functions. |

* The library files in the Lib folder must not be modified. If the user modify these files, Toshiba cannot guarantee the operation of the software. In such a case, the user shall be responsible for evaluating the software.

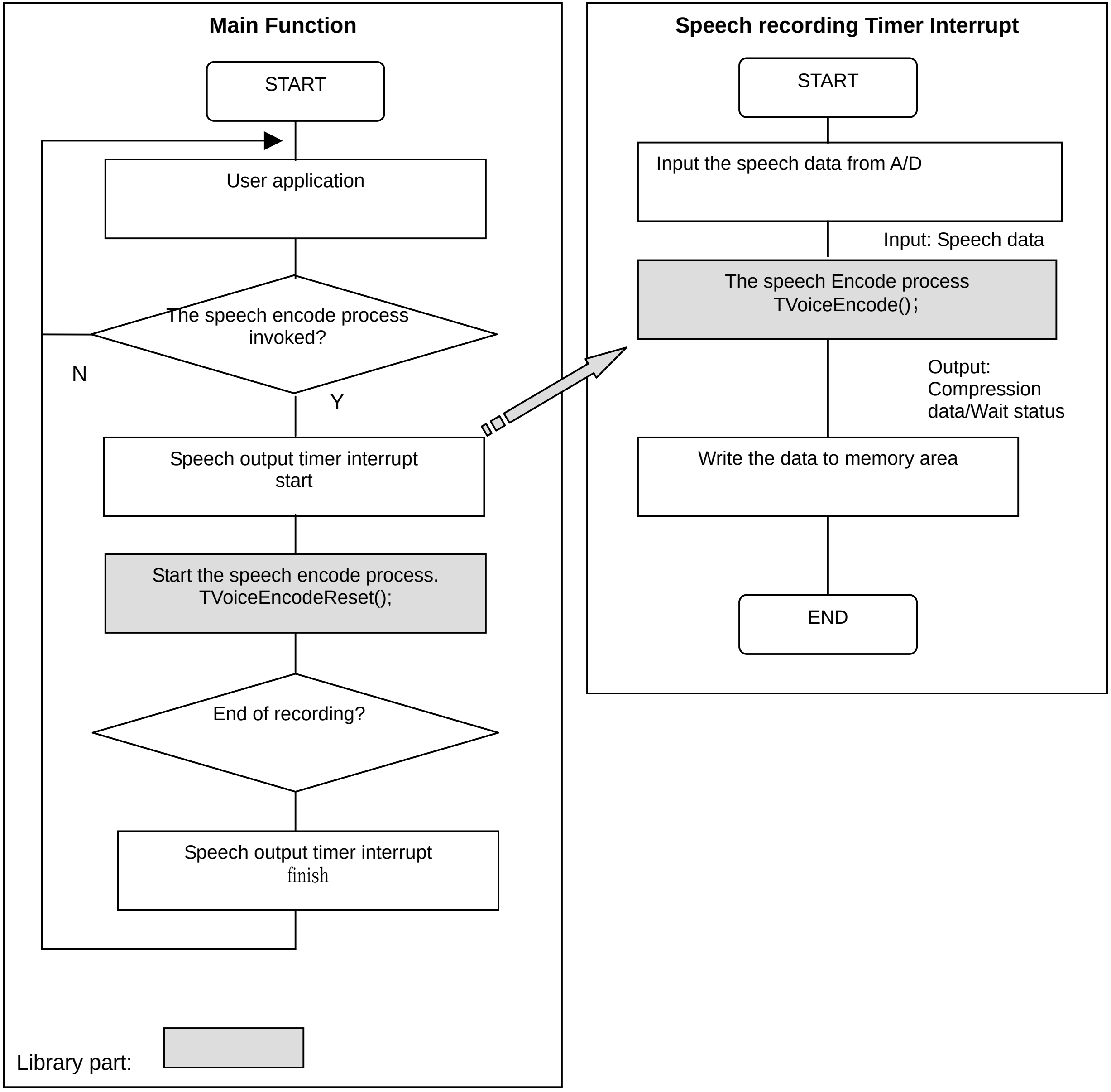
# 4 The Speech Encode Process by Software

This chapter describes how to encode speech data using this software IP. The following processing flow should be used as a basis when you create your speech encode application program using Toshiba API functions.

## 4.1 Flow of the Speech Encode Process

The flow of the encode process is shown below.

**Flow Diagram of the Speech Decode Process**


Main Function
START
User application
The speech encode process invoked?
N
Y
Speech output timer interrupt start
Start the speech encode process. TVoiceEncodeReset();
End of recording?
Speech output timer interrupt finish
Library part:
Speech recording Timer Interrupt
START
Input the speech data from A/D
Input: Speech data
The speech Encode process TVoiceEncode()；
Output: Compression data/Wait status
Write the data to memory area
END

## 4.2 Description of sample source code for encode

Outline of sample source operation for encode is described in below
We developed and evaluated this sample software in evaluation board for TMP86PS44 (16MHz). The customer needs customize for using MCU.

### 4.2.1 Initialize speech compression data area

Initialize data area is required for using speech record. It is achieved by calling TvoiceEncodeReset() function.

### 4.2.2 Speech data input operation

Input speech data from A/D converter and compress it by TvoiceEncode() function. The speech data is not compressing normally if input cycle is not regulated. You need to regulate speech input cycle as below.

**(1) Nested interrupts**
You need enable the speech input interrupt even in other interrupt.

**(2) Priority of interrupts**
Set the highest priority for the timer interrupt for speech input.

**(3) Processing inside the timer interrupt for speech input**
Don't operate any other functions before A/D input. You need place A/D input at the top of interrupt routine basically.

### 4.2.3 Voice compression routine

Encode the speech data by calling TVoiceEncode() function. This function input the A/D input data and output compressed speech data.

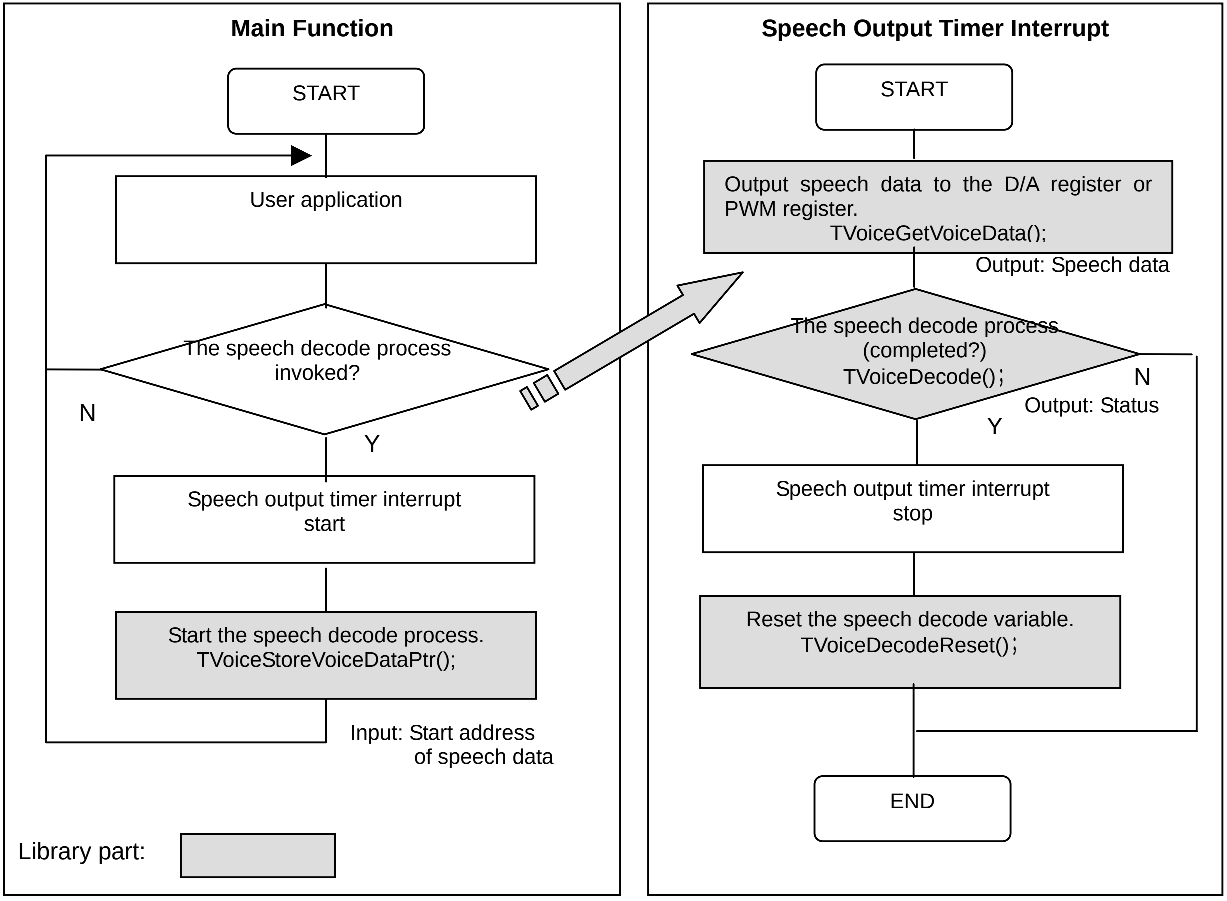
# 5 The Speech Decode Process by Software

This chapter describes how to decode speech data using this software IP. The following processing flow should be used as a basis when you create your speech decode application program using Toshiba API functions.

## 5.1 Flow of the Speech Decode Process

The flow of the decode process is shown below.

**Flow Diagram of the Speech Decode Process**



Main Function
START
User application
The speech decode process invoked?
N
Y
Speech output timer interrupt start
Start the speech decode process. TVoiceStoreVoiceDataPtr();
Input: Start address of speech data
Library part:
Speech Output Timer Interrupt
START
Output speech data to the D/A register or PWM register. TVoiceGetVoiceData();
Output: Speech data
The speech decode process (completed?) TVoiceDecode()；
N
Output: Status
Y
Speech output timer interrupt stop
Reset the speech decode variable. TVoiceDecodeReset()；
END

### 5.1.1 Storing the Pointer to Speech Data

To output speech data, it is necessary to first set the start address of the compressed ROM table file corresponding to the speech data to be reproduced. The specified speech data is reproduced by passing its start address as an argument when calling the TVoiceStoreVoiceDataPtr() function.

### 5.1.2 Updating the Selected Speech Data

The variable of the speech data output value is updated. Calling the TVoiceUpdateVoiceData() function sets the initial value of D/A output or PWM output.

### 5.1.3 Getting the Selected Speech Data

The decode speech data to be output to the D/A converter is obtained. The return value of the TVoiceGetVoiceData() function is output to the D/A converter or the PWM register. If the speech output cycle varies, speech data will not be decoded correctly. Make sure that the speech output cycle is kept constant by observing the following points.

**(1) <u>Nested interrupts</u>**
In addition to the timer interrupt for speech output, enable other interrupts for speech output.

**(2) <u>Priority of interrupts</u>**
Set the highest priority for the timer interrupt for speech output.

**(3) <u>Processing inside the timer interrupt for speech output</u>**
Do not place a comparison statement before outputting speech data to the D/A register or the PWM register. Basically the output to the D/A register or the PWM register should be performed at the head of the interrupt service routine.

### 5.1.4 Getting the Decode Process Status

Whether or not speech data is being output can be checked by calling the TVoiceGetDecodeStatus() function. This function makes it possible for a user application to decide the processing to be preformed depending on the status of the decode process.

### 5.1.5 Decoding Speech Data

The TVoiceDecode() function decodes speech data. The table pointer is updated in this function and the continue/end status is output as a return value. When the end status is returned, the decode process is finished.

### 5.1.6 Initializing the Internal Decode Variable

The TVoiceDecodeReset() function initializes the variable used for speech decoding. Be sure to call this function to initialize the variable when starting and stopping speech output.

## 5.2 Compression of Speech Data

The speech data used in the sample software has the following format.

Compressed speech data format

- Sampling frequency : 8 KHz
- Quantization bit count : 8 bits

## APPENDIX

### A. Revision History

This manual has been revised as shown below.

| Version | Issue date | Revision details |
|---|---|---|
| Ver 1.00 | 6 February 2003 | First issue |