

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

RENESAS

ユーザーズ・マニュアル

μSAP705100-B08

TTS ミドルウェア Ver.2.3

保守 / 廃止

対象デバイス

V830 ファミリ™

資料番号 U11757JJ4V0UM00 (第4版)

発行年月 April 1999 NS CP(K)

[メ モ]

目次要約

第1章	概説	... 15
第2章	ライブラリ仕様	... 21
第3章	ユーザ辞書の作成ツール	... 59
第4章	スピーチ・デザイナー対応機能	... 63
第5章	ビープ音生成機能	... 65
第6章	インストラクション	... 67
付録A	発音記号インタフェース	... 77
付録B	補足資料	... 83
付録C	総合索引	... 115

V830 ファミリ, V830, V831, V832 は, 日本電気株式会社の商標です。

Green Hills Software は, 米国 Green Hills Software, Inc.の商標です。

Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

SUN4 は米国 Sun Microsystems, Inc.の商標です。

UNIX は X/Open カンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標です。

TRON は The Realtime Operating system Nucleus の略称です。

ITRON は Industrial TRON の略称です。

- **本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。**
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

本版で改訂された主な箇所 (1/2)

箇 所	内 容
p.16	図 1 - 1 TTS のシステム構成を修正
p.17	1. 2. 1 システム構成の記述を一部修正
p.18	1. 3. 1 機 能の記述を一部修正
p.19	表 1 - 1 TTS に必要なメモリのサイズを一部修正
p.20	1. 3. 2 (3) (b) RX830 (μITRON Ver.3.0) を使用する場合は記述を一部修正
p.20	1. 3. 2 (4) サポート・ツールの記述を一部修正
p.20	1. 3. 3 性 能の記述を一部修正
p.22	2. 1. 2 テキスト解析部の記述に一部追加
p.23	2. 1. 2 (4) スペース, TAB, 改行の完全無視を追加
p.24	図 2 - 3 音声生成処理を一部修正
p.25	表 2 - 2 ライブラリ関数一覧を一部修正
p.26	図 2 - 3 アプリケーション処理フロー (テキスト解析を使用する場合) を一部修正
p.27	図 2 - 5 アプリケーション処理フロー (テキスト解析を使用しない場合) を一部修正
p.28	2. 3. 1 (1) リアルタイム・モードに注意を追加
p.28	図 2 - 6 出力バッファの使われ方を一部修正
p.29	2. 3. 2 ユーザ・OWN・コーディング関数の記述を一部修正
p.29	図 2 - 7 tts_DataReady, tts_DataReady2 が呼ばれるタイミングを一部修正
p.31	表 2 - 6 TTSDIC 構造体を一部修正
p.31	表 2 - 8 TTSNEM 構造体を一部修正
p.31	2. 4. 1 (f) TTSPHON の記述を一部修正
p.31	表 2 - 9 TTSPHON 構造体を一部修正
p.32	表 2 - 11 TTSVOIMOD 構造体を一部修正
p.33	2. 4. 1 (i) MWCVERSION の記述を一部修正
p.34	表 2 - 13 エラー・コード一覧に注を追加
p.35-57	2. 4. 3 外部インタフェースの記述を一部修正
p.36	図 2 - 8 モードの指定を一部修正
p.37	表 2 - 14 区切り文字コードを追加
p.43	表 2 - 15 音素データを一部修正
p.44	図 2 - 9 TTS 音声生成処理内部の動作を一部修正
p.57	図 2 - 11 tts_SyntheWave()内での OS インタフェース関数の呼び出しを追加
p.59	3. 3 入力フォーマットの記述を一部追加
p.63	第 4 章 スピーチ・デザイナー対応機能を追加
p.65	第 5 章 ビープ音生成機能を追加
p.68	図 6 - 1 TTS の提供形態を一部修正
p.70-72	6. 1. 1 ディレクトリとファイルを一部修正

本版で改訂された主な箇所 (2/2)

箇所	内容
p.73,74	6.2.2(2) フロッピー・ディスクの記述を一部修正
p.74	6.3 サンプル・プログラムの作成の記述を一部修正
p.76	6.6 ユーザ・OWN・コーディング関数の記述を一部修正
p.77	A.1 ローマ字表記発音記号とかな文字の対応の記述を一部修正
p.77	表 A - 1 ローマ字表記発音記号 かな文字の対応を一部修正
p.79	A.2 カナ表記発音記号とローマ字表記発音記号の対応の記述を一部修正
p.79	表 A - 2 カナ表記発音記号 ローマ字表記発音記号の対応を一部修正
p.81	表 A - 3 補助発音記号を一部修正
p.83	付録 B 補足資料を追加
旧版 p.69	付録 B OS インタフェースを削除

本文欄外の 印は、本版で改訂された主な箇所を示しています。

巻末にアンケート・コーナを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

(メ モ)

はじめに

- 対象者** このマニュアルは、V830 ファミリの応用システムを設計、開発するユーザを対象としています。
- 目的** V830 ファミリの応用システムを設計、開発をする際にサポートするミドルウェアを、ユーザに理解していただくことを目的としています。
- 構成** このマニュアルは、大きく分けて次の内容で構成しています。

- ・概 説
- ・ライブラリ仕様
- ・ユーザ辞書の作成ツール
- ・スピーチ・デザイナー対応機能
- ・ビープ音生成機能
- ・インストラクション
- ・発音記号インタフェース

読み方 このマニュアルの読者には、電気、論理回路、マイクロコンピュータおよびC言語に関する一般知識を必要とします。

V830 ファミリのハードウェア機能を知りたいとき

各製品のユーザズ・マニュアルのハードウェア編を参照してください。

V830 ファミリの命令機能を知りたいとき

V830 ファミリ ユーザズ・マニュアル アーキテクチャ編を参照してください。

- 凡 例** 注：本文中に付けた注の説明
注意：気をつけて読んでいただきたい内容
備考：本文の補足説明
数の表記：2進数...xxxx または xxxxB
10進数...xxxx
16進数...xxxxH または 0x xxxx
2のべき数を示す接頭語（アドレス空間、メモリ容量）：
K（キロ）： $2^{10} = 1024$
M（メガ）： $2^{20} = 1024^2$

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

V830 ファミリに関する資料

製品名		データ・シート	ユーザズ・マニュアル	
愛称	品名		ハードウェア編	アーキテクチャ編
V830™	μ PD705100	U11483J	U10064J	U12496J
V831™	μ PD705101	U12979J	U12273J	
V832™	μ PD705102	U13675J	U13577J	

開発ツールに関する資料 (ユーザズ・マニュアル)

資料名		資料番号	
CA830 (C コンパイラ)	操作編 (UNIX™ ベース)	U11013J	
	操作編 (Windows™ ベース)	U11068J	
	アセンブリ言語編	U11014J	
	C 言語編	U11010J	
	プロジェクト・マネージャ編	U11991J	
RX830 (リアルタイム OS)	ITRON1	基礎編	U11730J
		インストール編	U11731J
		テクニカル編	U11713J
	μ ITRON Ver.3.0	基礎編	U13152J
		インストール編	U13151J
		テクニカル編	U13150J
AZ830 (システム・パフォーマンス・アナライザ)		U13621J	

Green Hills Software™, Inc. (GHS 社) 製ツールに関する資料

GHS 社製ツールは、日本国内では下記で取り扱っております。各種製品とそれに関する資料については、下記へお問い合わせください。

株式会社アドバンスド データ コントロールズ (ADaC) TEL (03) 3576-5351

なお、上記関連資料は予告なしに内容を変更することがあります。設計などには必ず最新の資料をご使用ください。

目 次

第1章 概 説 ...	15
1.1 ミドルウェア ...	15
1.2 TTS (Text To Speech) ...	15
1.2.1 システム構成 ...	16
1.3 システム概要 ...	18
1.3.1 機 能 ...	18
1.3.2 動作環境 ...	19
1.3.3 性 能 ...	20
第2章 ライブラリ仕様 ...	21
2.1 概 要 ...	21
2.1.1 ユーザ辞書管理部 ...	21
2.1.2 テキスト解析部 ...	22
2.1.3 音声生成部 ...	24
2.2 ライブラリ概要 ...	25
2.3 アプリケーション処理フロー ...	25
2.3.1 音声生成部の動作 ...	28
2.3.2 ユーザ・OWN・コーディング関数 ...	29
2.4 関数仕様 ...	30
2.4.1 構造体 ...	30
2.4.2 エラー・コード ...	34
2.4.3 外部インタフェース ...	35
第3章 ユーザ辞書の作成ツール ...	59
3.1 概 要 ...	59
3.2 コマンド ...	59
3.3 入力フォーマット ...	59
3.4 出力データ ...	60
3.5 エラー ...	61
第4章 スピーチ・デザイナー対応機能 ...	63
4.1 概 要 ...	63

4.2 動作概要 ... 63

4.3 仕様 ... 63

第5章 ビープ音生成機能 ... 65

5.1 概要 ... 65

5.2 仕様 ... 65

第6章 インストレーション ... 67

6.1 提供形態 ... 67

6.1.1 ディレクトリとファイル ... 70

6.2 ホスト・マシンへのファイル展開 ... 72

6.2.1 UNIX 版 (CGMT) ... 72

6.2.2 Windows 版 ... 73

6.3 サンプル・プログラムの作成 ... 74

6.4 ロケーションの変更 ... 75

6.5 シンボル名規約 ... 76

6.6 ユーザ・OWN・コーディング関数 ... 76

付録 A 発音記号インタフェース ... 77

A.1 ローマ字表記発音記号とかな文字の対応 ... 77

A.2 カナ表記発音記号とローマ字表記発音記号の対応 ... 79

A.3 補助発音記号 ... 81

付録 B 補足資料 ... 83

B.1 テキスト解析処理部 ... 83

B.2 音声生成処理部 ... 87

B.3 ユーザ辞書作成時の注意事項 ... 88

B.4 記号コードの読みについて ... 89

B.4.1 TTS SJIS コード表 ... 102

付録 C 総合索引 ... 115

C.1 50音で始まる語句の索引 ... 115

C.2 アルファベットで始まる語句の索引 ... 117

図の目次

図番号	タイトル, ページ
1 - 1	TTS のシステム構成 ... 16
2 - 1	ユーザ辞書作成処理 ... 21
2 - 2	テキスト解析処理 ... 22
2 - 3	音声生成処理 ... 24
2 - 4	アプリケーション処理フロー (テキスト解析を使用する場合) ... 26
2 - 5	アプリケーション処理フロー (テキスト解析を使用しない場合) ... 27
2 - 6	出力バッファの使われ方 ... 28
2 - 7	tts_DataReady, tts_DataReady2 が呼ばれるタイミング ... 29
2 - 8	モードの指定 ... 36
2 - 9	TTS 音声生成処理内部の動作 ... 44
2 - 10	バージョン番号のフォーマット ... 52
2 - 11	tts_SyntheWave()内での OS インタフェース関数の呼び出し ... 57
4 - 1	動作概要 ... 63
6 - 1	TTS の提供形態 ... 68

表の目次

表番号	タイトル, ページ
1 - 1	TTSに必要なメモリのサイズ ... 19
2 - 1	ユーザ辞書作成処理の種類 ... 21
2 - 2	ライブラリ関数一覧 ... 25
2 - 3	ライブラリ参照定数 ... 30
2 - 4	TTSTEXT 構造体 ... 30
2 - 5	TTSPHNBUF 構造体 ... 30
2 - 6	TTSDIC 構造体 ... 31
2 - 7	TTSOSINFO 構造体 ... 31
2 - 8	TTSMEM 構造体 ... 31
2 - 9	TTSPHON 構造体 ... 31
2 - 10	TTSVOICE 構造体 ... 32
2 - 11	TTSVOIMOD 構造体 ... 32
2 - 12	MWCVERSION 構造体 ... 33
2 - 13	エラー・コード一覧 ... 34
2 - 14	区切り文字コード ... 37
2 - 15	音素データ ... 43
2 - 16	品詞コード ... 51
3 - 1	品詞コード ... 60
3 - 2	エラー・メッセージ ... 61
4 - 1	拡張発音情報に対するの任意語入れ替え例 ... 64
4 - 2	制御記号 ... 64
5 - 1	ビープ音の種類 ... 65
6 - 1	セクション名 ... 75
A - 1	ローマ字表記発音記号 かな文字の対応 ... 77
A - 2	カナ表記発音記号 ローマ字表記発音記号の対応 ... 79
A - 3	補助発音記号 ... 81
B - 1	小文字の単独発音 ... 85
B - 2	半角記号の読み (「記号読む」の場合のみ) ... 89
B - 3	全角記号の読み (「記号読む」の場合のみ) ... 90
B - 4	半角記号の読み ... 92
B - 5	全角記号の読み ... 92

第1章 概 説

この章では、ミドルウェアの概要と TTS (Text To Speech) の概要について説明します。

1.1 ミドルウェア

ミドルウェアとは、プロセッサの性能を最大限に引き出すようにチューニングされたソフトウェア群のことです。

現在では高性能 RISC プロセッサが比較的安く市場に投入され、従来、専用ハードウェアに頼っていた処理を「高性能 RISC プロセッサ」+「ソフトウェア」というアプローチで実現できるようになりました。この「ソフトウェア」をミドルウェアと呼んでいます。

NEC では、ヒューマン・マシン・インタフェースおよび信号処理技術をミドルウェアの形で用意しています。さまざまなユーザのニーズに対応して、優れたシステム・ソリューションを提供しています。

備考 RISC : Reduced Instruction Set Computer

1.2 TTS (Text To Speech)

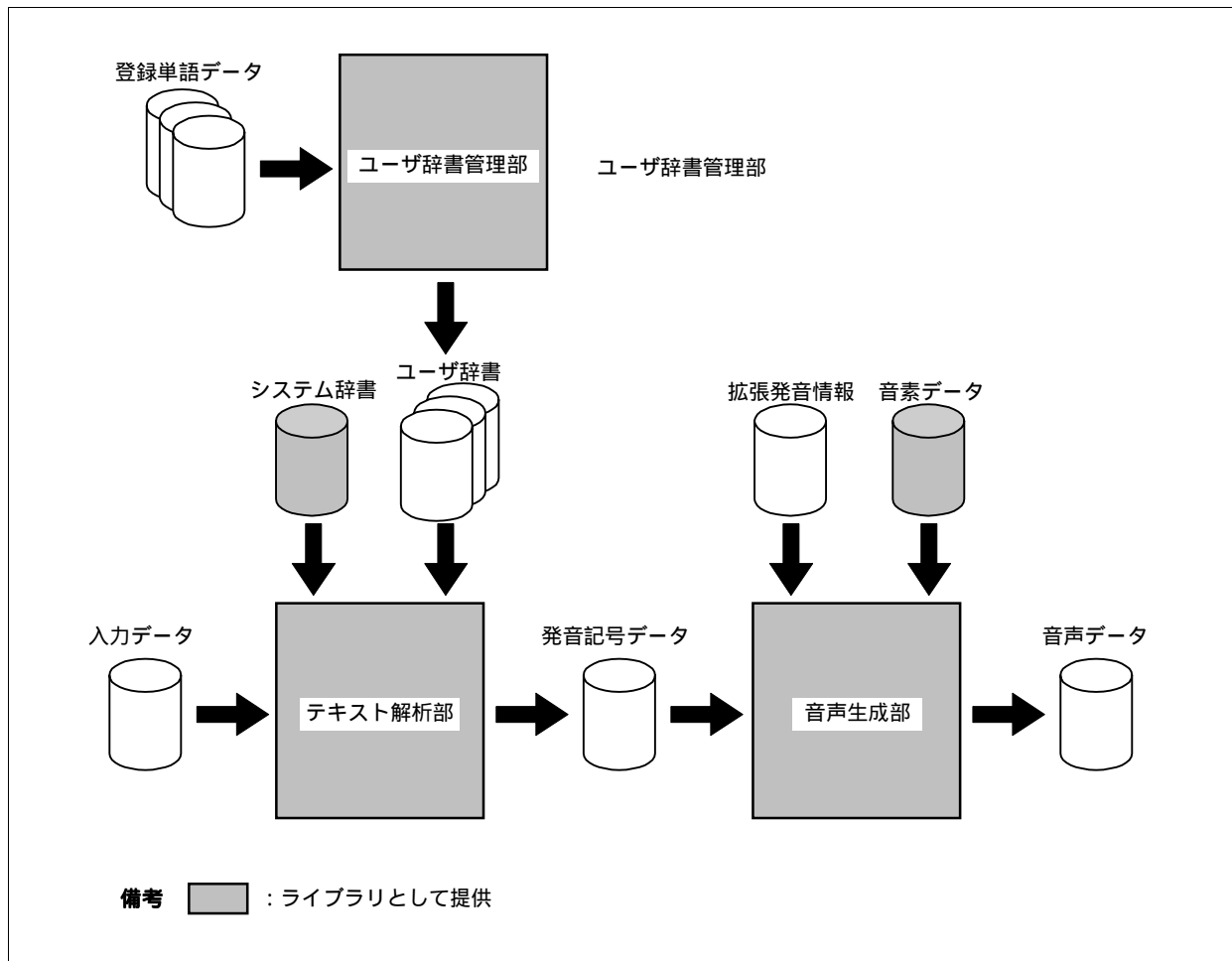
TTS は、日本語かな漢字テキストから音声を合成する技術 (規則合成方式) です。規則合成方式は、テキストを音素や音節を組み合わせることにより、音声を合成して任意のテキストを発声させる方式です。

TTS ミドルウェアは、CPU デバイス (V830 ファミリ) 上で動作するミドルウェアです。メモリ上に配置された日本語かな漢字テキスト (SJIS コード) を入力して、合成した音声データ (16 ビット・データ) をメモリ上に出力します。

1.2.1 システム構成

TTS のシステム構成を図 1 - 1 に示します。

図1 - 1 TTSのシステム構成



(1) テキスト解析部

テキスト解析部は、システム辞書およびユーザ辞書を使用して、入力した日本語かな漢字テキストの解析を行い、発音記号データ列に変換します。

(2) 音声生成部

音声生成部は、音素データを使用して、発音記号データ列から音声（8 kHz/11 kHz の 16 ビット PCM データ）を合成します。

(3) システム辞書

テキストを解析するための辞書です。テキスト解析部が動作する際に必要になります。システム辞書は NEC が提供する辞書で、内容の変更 / 追加はできません。

(4) ユーザ辞書

システム辞書に登録されていない語彙を登録するための補助辞書です。ユーザ辞書はユーザにより定義され、複数の辞書を使用することができます。また、ユーザ辞書はオンボード/オフボードでの作成ができます。

(5) 音素データ

音声生成部が使用する音素データを格納しています。音素データは NEC が提供するデータで、内容の変更/追加はできません。NEC が提供する音素データには、次の5種類があります。

- ・ 男声 (11 kHz サンプリング)
- ・ 女声 1 (11 kHz サンプリング)
- ・ 女声 2 (11 kHz サンプリング)
- ・ 男声 (8 kHz サンプリング)
- ・ 女声 (8 kHz サンプリング)

(6) ユーザ辞書管理部

入力されたユーザ辞書 (登録単語データ) をオンボード上で作成します。オフボードで作成することもできます。

(7) 拡張発音情報

スピーチ・デザイン機能を使用する際の入力データ列です。スピーチ・デザイン機能を使用しない場合には、必要ありません。

(8) 入力データ

テキスト解析部の入力データ列 (日本語かな漢字テキスト) です。入力データは、SJIS コードを使用します。

(9) 発音記号データ

音声生成部の入力データ列です。データ・フォーマットは NEC 独自仕様の発音記号列 (テキスト解析部の出力) です。

(10) 音声データ

音声生成部が出力する音声データ (16 ビット PCM データ) です。

(11) 登録単語データ

オンボードでユーザ辞書を作成する際の入力データ列です。入力データ・フォーマットについては **3.3 入力フォーマット**を参照してください。

1.3 システム概要

1.3.1 機 能

TTS は、日本語かな漢字テキスト (SJIS コード) を入力して、合成した音声データ (16 ビット) をメモリ上に出力します。次に特徴を示します。

- 男性の声, 女性 (2 種類) の声を発声可能
- 2 種類 (8 kHz / 11 kHz) の音声データを生成可能
- ピッチ, 速度, アクセントの強さを調整可能
- 日本語解析用辞書として, システム辞書 (約 8 万語を格納) を提供
- 特殊文字読み上げ用に, ユーザ辞書を用意
- スピーチ・デザイナー機能をサポート (11 kHz サンプリング使用時のみ)
- リアルタイム OS (RX830) 対応

1.3.2 動作環境

(1) 対象 CPU

- ・ V830 ファミリ

(2) 必要メモリ

TTS を動作させるために必要な ROM/RAM のサイズを表 1 - 1 に示します。

作成するユーザ辞書のサイズは、登録するテキストのバイト数とほぼ同じになります。登録する内容によって大幅に異なる場合があります。

表1 - 1 TTSに必要なメモリのサイズ

テキスト解析部	ROM	約 30 K バイト	text
		約 9 K バイト	data
		約 1.2 M バイト	辞書データ (約 8 万語格納)
	RAM	23 K バイト	work ^{注1}
		256 バイト	stack ^{注2}
音声生成部	ROM	約 60 K バイト	text
		約 28 K バイト	data
		約 665 K バイト	女声音素データ (8 kHz サンプリングで使用する場合)
		約 675 K バイト	男声音素データ (8 kHz サンプリングで使用する場合)
		約 845 K バイト	女声音素データ 1 (11 kHz サンプリングで使用する場合)
		約 1400 K バイト	女声音素データ 2 (11 kHz サンプリングで使用する場合)
		約 875 K バイト	男声音素データ (11 kHz サンプリングで使用する場合)
	RAM	12 バイト	reserve ^{注3}
		128 K バイト	work ^{注1}
256 バイト		stack ^{注2}	
ユーザ辞書管理部	ROM	約 4 K バイト	text
	RAM	((登録数 × 8) + 100) バイト	reserve ^{注3}
		200 バイト	work ^{注1}
		64 バイト	stack ^{注2}

注 1 . work エリアは、それぞれのミドルウェア関数を使用する領域です (2.4.1 構造体参照)。

2 . stack エリアは、ユーザ・スタックを使用します。

3 . reserve エリアは、複数のミドルウェア関数で使用する共通領域です (2.4.1 構造体参照)。

(3) OS 資源

マルチタスク版の音声生成部では、OS 資源としてイベント・フラグを 1 つ使用しています。また、OS インタフェース関数を通して次のシステム・コールを使用しているため、OS の設定でこれらのシステム・コールが必ずリンクされるようにしてください。

(a) RX830 (ITRON1) を使用する場合

- ・ cre_flg()
- ・ del_flg()
- ・ set_flg()
- ・ wai_flg()

(b) RX830 (μITRON Ver.3.0) を使用する場合

- ・ cre_flg()
- ・ del_flg()
- ・ set_flg()
- ・ wai_flg()
- ・ clr_flg()
- ・ pol_flg()

(4) サポート・ツール

- ・ NEC 製リアルタイム OS : RX830 (Windows 版, SUN4™ 版)
- ・ NEC 製 C コンパイラ・パッケージ : CA830 (Windows 版, SUN4 版)
- ・ GHS 社製 C コンパイラ/アセンブラ : CCV800 (Windows 版, SUN4 版)

1.3.3 性 能

【入力文字列】「八重洲通り八丁堀交差点付近です。」(11 kHz 音声)

【処理時間】120 ms (バッチ・モード時)

テキスト解析部 : 10 ms (ユーザ辞書未使用)

音声生成部 : 110 ms (女性1音素使用時)

: 125 ms (女性2音素使用時)

<参考> Ver.2.1 の場合

130 ms (バッチ・モード時)

テキスト解析部 : 10 ms (ユーザ辞書未使用)

音声生成部 : 120 ms (女性1音素使用時)

【条 件】マイダス・ラボ製 V831 評価ボード (RTE-V831-PC)

CPU : V830 (動作周波数: 内部 100 MHz (外部 33 MHz))

メモリ : EDO DRAM

第2章 ライブラリ仕様

2.1 概要

TTS の処理は、大きく分けて次の3つに分類されます（図1-1 TTS のシステム構成参照）。

- ・ ユーザ辞書管理部
- ・ テキスト解析部
- ・ 音声生成部

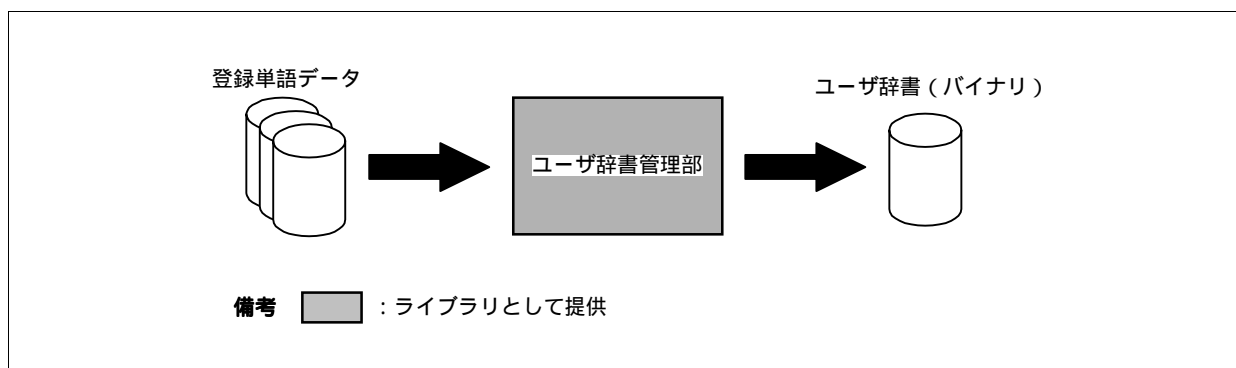
2.1.1 ユーザ辞書管理部

ユーザ辞書管理部は、辞書登録単語を TTS のテキスト解析部で利用できるフォーマットに変換します。ユーザ辞書作成処理は、オンライン（V830 ファミリ上で作成）とオフライン（パソコンまたはワークステーション上で udicmake コマンドを使用して作成）の2通りがあります。

表2-1 ユーザ辞書作成処理の種類

	概要	V830 上での辞書内容の変更
オンライン	ミドルウェアのユーザ辞書管理部を使用	可能
オフライン	パソコン/ワークステーション上で オーサリング・ツールを使用	不可能

図2-1 ユーザ辞書作成処理



2.1.2 テキスト解析部

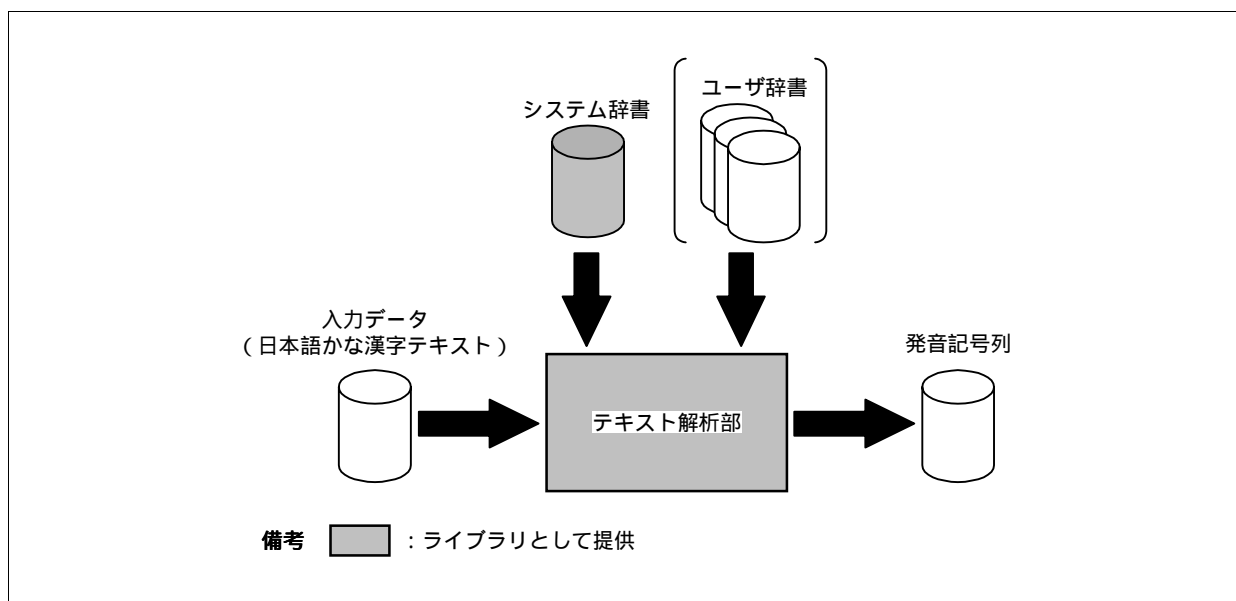
テキスト解析部は、日本語テキストをシステム辞書、ユーザ辞書を使用して、発音記号列に変換します。発音記号列の表記については付録 A 発音記号インタフェースを参照してください。

テキスト解析処理には、必ずシステム辞書が必要です。

TTS では次のシステム辞書を提供しています。

- ・システム辞書：tts_SysDic

図2-2 テキスト解析処理



テキスト解析では、次の解析モードの指定を変更できます。設定方法については 2.4.3 (1) `tts_AnalyseText` 関数を参照してください。

- ・記号読みのモード指定
- ・数字読みのモード指定
- ・アルファベットの大文字 / 小文字の区別
- ・スペース, TAB, 改行の完全無視

(1) 記号読みのモード指定

辞書中の登録語で、品詞が「記号」と登録されている単語を、読む / 読まないを指定できます。

例 「【】」と入力があった場合

- ・記号読みを指定した場合は、「はじめすみつきかっこ」と読みます。
- ・記号読みを指定しない場合は、「【】」を読みません。

(2) 数字読みのモード指定**(a) 「棒読み」を指定した場合**

「1234」を「いちにさんよん」と読みます。

(b) 「桁付き」を指定した場合

「1234」を「せんにひゃくさんじゅうよん」と読みます。

(3) アルファベットの大文字/小文字の区別**(a) 「アルファベットの大文字/小文字を区別する」を選択した場合**

テキスト解析部へ入力したテキスト中のアルファベットの大文字と小文字を区別します。この場合、「TTS」、「Tts」、「tts」は、すべて異なる単語として解釈します。

(b) 「アルファベットの大文字/小文字を区別しない」を選択した場合

テキスト解析部へ入力したテキストが、まったく同じ表記で辞書に登録されていなかった場合、すべてのテキストを小文字であるとみなし、再び検索します。

例 「tts (てきすととーすぴーち)」と「Tts (てきすとリーだ)」をユーザ辞書に登録している場合

「tts」は、「てきすととーすぴーち」と読みます。

「Tts」は、「てきすとリーだ」と読みます。

「TTS」は、「てきすととーすぴーち」と読みます。

(4) スペース, TAB, 改行の完全無視

解析処理で、入力テキスト中に含まれるスペース, TAB, 改行を無視する/しないを選択できます。

例 「起_床_時_間」を解析した場合(_ はスペース)

・完全無視指定なし:

ki ; yuka ; toki “ ; aida . (「起」, 「床」, 「時」, 「間」と4つの単語として解釈します。)

・完全無視指定あり:

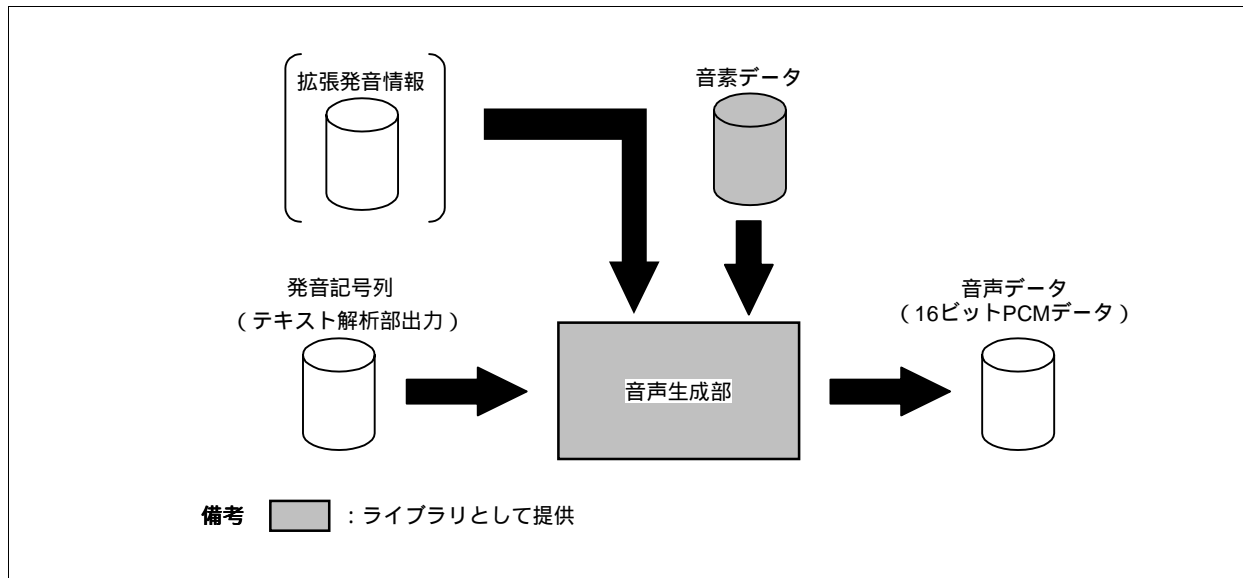
k_isho_ji “ kaN . (「起床時間」の1つの単語として解釈します。)

2.1.3 音声生成部

音声生成部は、発音記号列（テキスト解析部出力）から音声データを生成します。音声生成処理には、必ず音素データが必要です。

音声生成部では、入力される発音記号列に対して、「読み上げ速度」、「アクセント」、「ピッチ」を設定できます。

図2-3 音声生成処理



2.2 ライブラリ概要

ライブラリ関数の一覧を次に示します。

表2-2 ライブラリ関数一覧

処 理	関 数		説 明
テキスト解析処理	テキスト解析処理	tts_AnalyseText	日本語仮名漢字文字列（テキスト）を発音記号に変換します。
	解析処理停止	tts_StopAnalyse	tts_AnalyseText の処理を停止します。
音声生成処理	初期化処理部	tts_Initialize	音声生成処理の初期化を行います。
	終了処理部	tts_Uninitialize	音声生成処理の後処理を行います。
	音声生成処理部	tts_SyntheWave	発音記号を元に音声生成処理を行います。
	音声生成処理停止要求	tts_StopSynthe	音声生成処理を停止します。
	停止要求クリア	tts_ClearStopSynthe	音声生成処理停止要求を解除します。
	音声データ要求処理部	tts_RequestWave	音声データの要求をします。
ユーザ辞書管理処理	ユーザ辞書初期化処理部	tts_DicInitialize	ユーザ辞書管理の初期化を行います。
	ユーザ辞書登録	tts_DicEntry	ユーザ辞書の登録を行います。
その他	バージョン情報出力要求	tts_GetVersion	TTS のバージョン情報、エンジン情報の出力を行います。
ユーザ定義 (コール・バック関数)	音声生成処理終了通知	tts_DataReady ^{注1}	音声生成処理終了後、呼び出されます。
	音声生成処理終了通知	tts_DataReady2 ^{注1}	音声生成処理終了後、呼び出されます。
RTOS インタフェース	フラグ生成	mwc_CreFlg ^{注1,2}	関数内にイベント・フラグを生成するシステム・コールを記述します。
	フラグ削除	mwc_DelFlg ^{注1,2}	関数内にイベント・フラグを削除するシステム・コールを記述します。
	フラグ・セット	mwc_SetFlg ^{注1,2}	関数内にイベント・フラグをセットするシステム・コールを記述します。
	フラグ・クリア	mwc_ClrFlg ^{注1,2}	関数内にイベント・フラグをクリアするシステム・コールを記述します。
	フラグ・チェック	mwc_WaiFlg ^{注1,2}	関数内にイベント・フラグをチェックするシステム・コールを記述します。
	フラグ・チェック	mwc_PolFlg ^{注1,2}	関数内にイベント・フラグをチェックするシステム・コールを記述します。

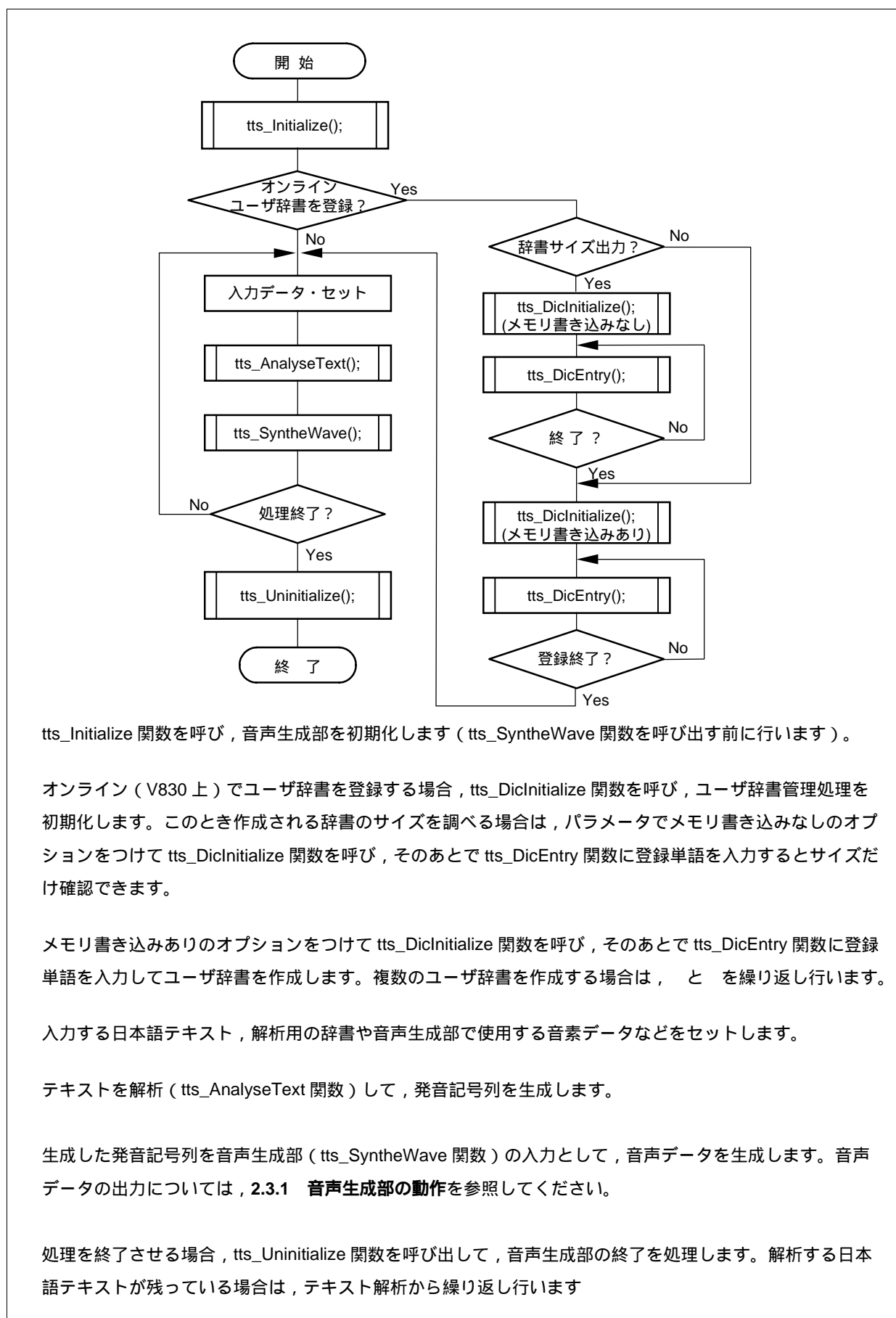
注1. ソースでの提供になります。

2. リアルタイム OS を使用しない場合、NULL 関数になります。

2.3 アプリケーション処理フロー

TTS の使用方法として、テキスト解析を使用して日本語テキストの解析から行う場合と、アクセントや読みの情報をあらかじめ持った発音記号列を使用してテキスト解析処理を省略して行う場合の2通りがあります。それぞれのアプリケーション処理フローを図2-4、2-5に示します。

図2-4 アプリケーション処理フロー（テキスト解析を使用する場合）



tts_Initialize 関数を呼び、音声生成部を初期化します（tts_SyntheWave 関数を呼び出す前に行います）。

オンライン（V830 上）でユーザ辞書を登録する場合、tts_DicInitialize 関数を呼び、ユーザ辞書管理処理を初期化します。このとき作成される辞書のサイズを調べる場合は、パラメータでメモリ書き込みなしのオプションをつけて tts_DicInitialize 関数を呼び、そのあとで tts_DicEntry 関数に登録単語を入力するとサイズだけ確認できます。

メモリ書き込みありのオプションをつけて tts_DicInitialize 関数を呼び、そのあとで tts_DicEntry 関数に登録単語を入力してユーザ辞書を作成します。複数のユーザ辞書を作成する場合は、 と を繰り返し行います。

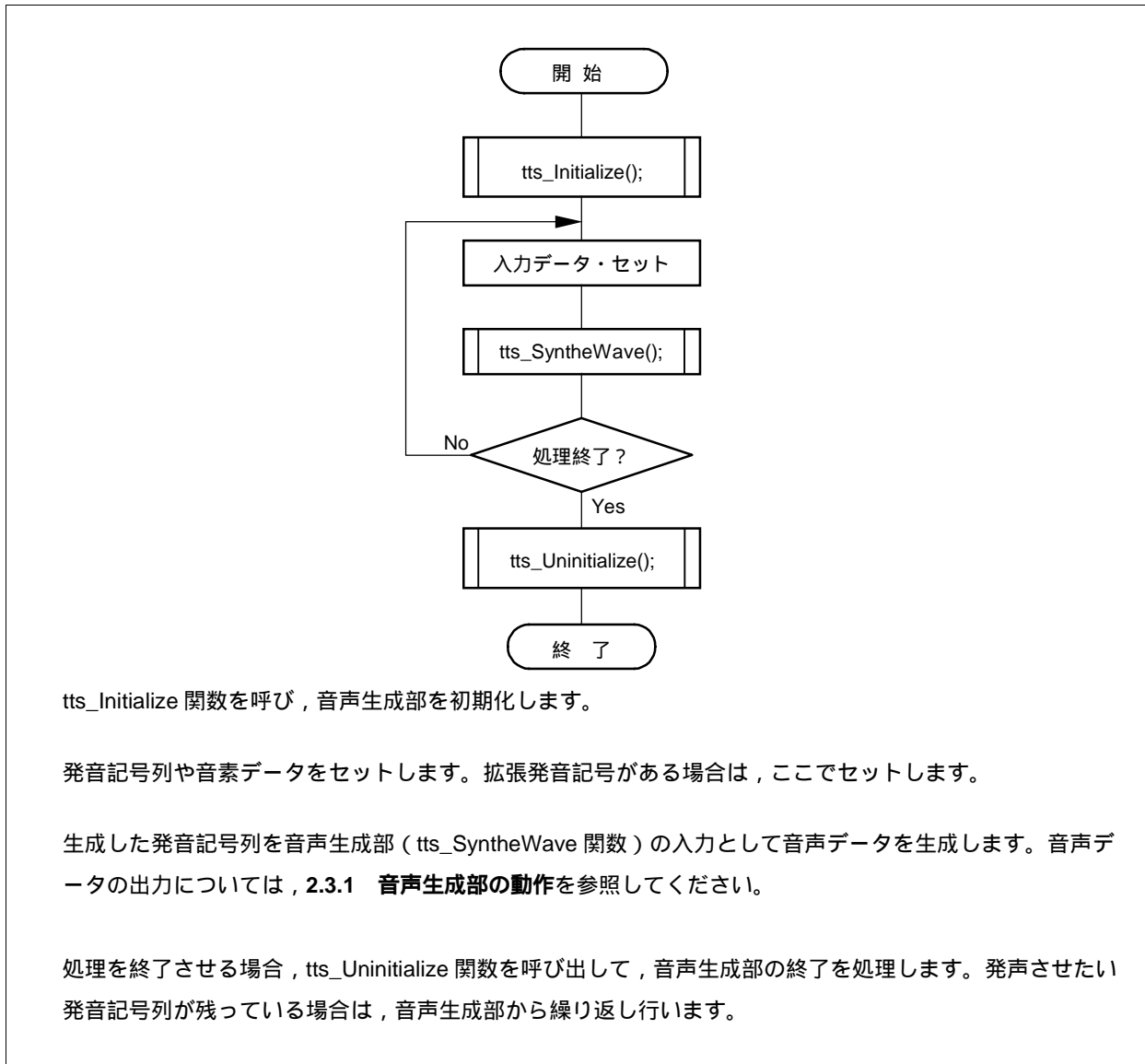
入力する日本語テキスト、解析用の辞書や音声生成部で使用する音素データなどをセットします。

テキストを解析（tts_AnalyseText 関数）して、発音記号列を生成します。

生成した発音記号列を音声生成部（tts_SyntheWave 関数）の入力として、音声データを生成します。音声データの出力については、2.3.1 音声生成部の動作を参照してください。

処理を終了させる場合、tts_Uninitialize 関数を呼び出して、音声生成部の終了を処理します。解析する日本語テキストが残っている場合は、テキスト解析から繰り返し行います

図2-5 アプリケーション処理フロー（テキスト解析を使用しない場合）



2.3.1 音声生成部の動作

音声生成部は、リアルタイム・モードとバッチ・モードの2種類の動作モードがあります。

(1) リアルタイム・モード

音声生成処理と、ユーザが作成する音声出力処理を同期して行います。リアルタイム・モードでは、1ブロック（8Kバイト）単位で音声を合成します。ユーザが作成した音声出力処理と並行に処理することにより、コンパクトな音声出力データ領域での合成処理が可能になります。

リアルタイム・モードにおいてミドルウェアは、パラメータにより指定された出力バッファ領域をリング・バッファとして使用します。図2-6に音声生成部を呼び出したときの音声出力バッファの状態を示します。

すべての入力発音記号列を合成するまで、呼び出しもとのプログラムには戻りません。

リアルタイム・モードでは、最小2ブロック（16Kバイト）の出力があれば動作できます。

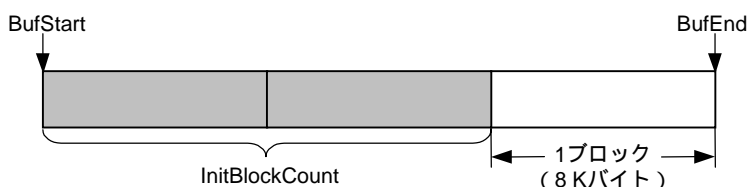
注意1. 生成音声の最後のデータが8Kバイト以下の場合は、残りのバッファには何も出力されないので、前に書き込んだデータが残った状態になります。

2. リアルタイム・モードは、複数の関数（`tts_AnalyseText`、`tts_SyntheWave`）を同時に動作させることができます。その際には、各関数のワーク領域がぶつからないようにしてください。

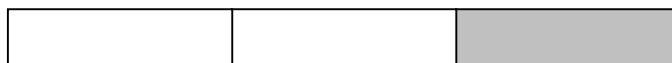
3. リアルタイム・モードに必要な出力バッファ・サイズは、アプリケーション・システムの負荷に依存します

図2-6 出力バッファの使われ方

音声生成部（`tts_SyntheWave`関数）を呼び出すと、パラメータ（`InitBlockCount`）で指定されたブロック数分の音声データを生成して、音声生成処理はウエイト状態になります。




次にユーザが`tts_RequestWave`関数を呼び出すごとに、1ブロック分の音声データを生成していきます。



音声データがバッファの終点（`BufEnd`）までくると、次の音声データはバッファの始点から出力します。このとき音声データは上書きされるので、元のデータが発声終了したことを確認してから`tts_RequestWave`関数を呼び出してください。



 : 生成する音声データ

備考 音素データの出力領域を3ブロック（8K×3）確保し、`InitBlockCount=2`としたものです。

(2) バッチ・モード

入力した発音記号列を、一括して音声データに変換します。生成される音声データよりも出力バッファが小さい場合は、出力バッファ分の音声データを生成して処理を終了します。

2.3.2 ユーザ・OWN・コーディング関数

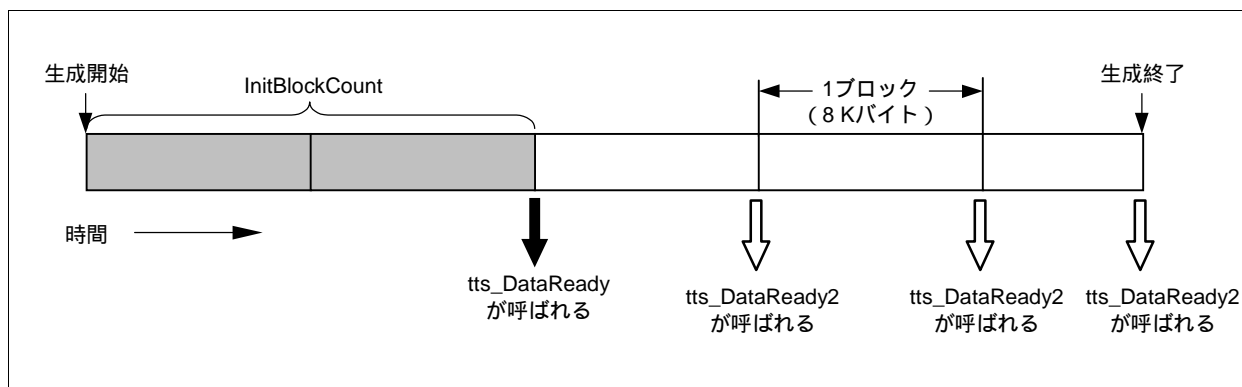
音声生成部は、InitBlockCount 分の音声データを生成すると、tts_DataReady 関数を呼び出してからウエイト状態になります。

その後、同じようにユーザが tts_RequestWave 関数を呼び出して1ブロック分の音声データを生成したあとに tts_DataReady2 関数を呼び出してウエイト状態になります。

tts_DataReady 関数と tts_DataReady2 関数は、TTS が呼び出す関数でソース・プログラムの形で提供しています。提供した状態は、return; だけの記述になっています。この関数を書き換えることにより、出力される音声データを1ブロックずつ転送したりすることができます。

tts_DataReady, tts_DataReady2 が呼ばれるタイミングを図2-7に示します。音声生成部の内部動作については2.4.3(5) tts_SyntheWave 関数を参照してください。

図2-7 tts_DataReady, tts_DataReady2が呼ばれるタイミング



2.4 関数仕様

2.4.1 構造体

このライブラリで使用する構造体、定数について説明します（ttslib.h で定義されています。使用するためには、ttslib.h をインクルードしてください）。

(1) 定数 (参照用)

表2-3 ライブラリ参照定数

定数名	数 値	説 明
tts_BLOCKSIZE	8192	音声バッファのブロック・サイズ
tts_ANALYSEWORK	23x1024	tts_AnalyseText で使用するワーク・サイズ
tts_SYNTHWORK	128x1024	tts_SyntheWave で使用するワーク・サイズ

(2) 構造体 (typedef で宣言)

(a) TTSTEXT

TTSTEXT は、テキスト解析部への入力文字列に関する情報を指定する構造体です。

TextIn は、解析する文字列の先頭アドレスを指定します。

NextText は、テキスト解析部からの出力が返される領域で、次の解析開始アドレスが格納されます（次の解析開始アドレスは、指定した区切り文字の次を指します）。

表2-4 TTSTEXT構造体

メンバ名	型	入出力	内 容
TextIn	char*	入力	解析する日本語かな漢字テキスト格納領域の先頭アドレス
NextText	char*	出力	次のテキスト解析の開始アドレス

(b) TTSPHNBUF

TTSPHNBUF は、tts_AnalyseText によって生成された解析結果を格納するための構造体です。

PhoneData には、解析された発音記号列を格納する領域の先頭アドレスを指定します。

PhoneDataSize には、PhoneData のサイズを指定します。

表2-5 TTSPHNBUF構造体

メンバ名	型	入出力	内 容
PhoneData	char*	入力	解析結果出力領域の先頭アドレス
PhoneDataSize	int	入力	解析結果出力領域のサイズ

(c) TTSDIC

TTSDIC は、使用する辞書に関する情報を指定する構造体です。

DicArea は、辞書領域の先頭アドレスを指定します。

DicState は、DicArea によって指定される辞書を使用するかどうかを指定します。

表2 - 6 TTSDIC構造体

メンバ名	型	入出力	内 容
DicArea	long*	入力	辞書領域の先頭アドレス
DicState	short	入力	辞書状態 0：使用しない 1：使用する

注意 DicArea には、必ず 4 バイト align された領域を指定してください。

(d) TTSOSINFO

TTSOSINFO は、音声生成処理で使用する OS 資源を生成するための情報の構造体です。

表2 - 7 TTSOSINFO構造体

メンバ名	型	入出力	内 容
flg_ID	short	入力	イベント・フラグの ID

(e) TTSMEM

TSMEM は、音声生成処理およびユーザ辞書管理処理で使用するワーク・メモリ領域について指定します。

ReserveArea は、音声生成処理全体またはユーザ辞書管理処理全体で使用する共通のワーク・メモリ領域（リザーブ・エリア）を指定します。

WorkArea は、それぞれの関数が使用するワーク・メモリ領域（ワーク・エリア）を指定します。

表2 - 8 TSMEM構造体

メンバ名	型	入出力	内 容
ReserveArea	long*	入力	リザーブ・エリア（保存用）先頭アドレス
WorkArea	long*	入力	ワーク・エリア（関数ごとに使用）先頭アドレス

注意 ReserveArea および WorkArea には、必ず 4 バイト align された領域を指定してください。

(f) TTSPHON

TTSPHON は、tts_SyntheWave に対して発音情報を指定するための構造体です。

PhoneData は、発音記号データの格納された領域の先頭アドレスを指定します。tts_SyntheWave は、このアドレスから発音記号を読み込み、音声を合成します。

PhoneInfo は拡張発音情報の格納された領域の先頭アドレスを指定します（拡張発音情報を使用しない場合は NULL を指定してください）。

NextData は音声生成部からの出力が返される領域で次の音声処理の開始アドレスが格納されます。

表2 - 9 TTSPHON構造体

メンバ名	型	入出力	内 容
PhoneData	char*	入力	発音記号データ先頭アドレス
PhoneInfo	char*	入力	拡張発音情報先頭アドレス
NextData	char*	出力	次の音声生成開始アドレス

(g) TTSVOICE

TTSVOICE は、tts_SyntheWave によって合成された音声などの情報を格納する構造体です。

BufStart と BufEnd は、合成された音声データを出力する領域の先頭アドレスと終了アドレスを指定します。tts_SyntheWave は、BufStart から BufEnd - 1 までの範囲に音声データを出力します。

InitBlockCount は、音声合成処理が呼び出されてから tts_DataReady 関数が呼び出されるまでの間に、どれくらいの音声データを生成するかを指定します。1 ブロックのサイズは 8 K バイトです (*InitBlockCount × 8 K バイトが出力バッファ・サイズを越えないようにしてください)。

CurrentSize は、tts_SyntheWave が生成した音声データのサイズ (バイト) です。

Status は、tts_SyntheWave の処理状態を返します (図 2 - 9 TTS 音声生成処理内部の動作参照)。

表2 - 10 TTSVOICE構造体

メンバ名	型	入出力	内 容
BufStart	short*	入力	出力バッファ開始アドレス
BufEnd	short*	入力	出力バッファ終了アドレス
InitBlockCount	int	入力	初期生成音声ブロック数
CurrentSize	long	出力	生成音声データ・サイズ
Status	int	出力	生成完了フラグ 0: 音声データが準備できていない 1: 音声データが準備できた 2: 生成処理完了

(h) TTSVOIMOD

TTSVOIMOD は、tts_SyntheWave の動作モードを指定するための構造体です。

Character は、メモリ上の音素データの先頭アドレスを指定します。

Pitch, Speed, Accent は、それぞれ出力音声のピッチ, スピード, アクセントを指定 (- 10 ~ 10 までの整数) します。

表2 - 11 TTSVOIMOD構造体

メンバ名	型	入出力	内 容
Character	long*	入力	音素データ先頭アドレス
Pitch	short	入力	声の高さの指定
Speed	short	入力	読み上げ速度の指定
Accent	short	入力	アクセントの強弱の指定

注意 Character には、必ず 4 バイト align された領域を指定してください。

(i) MWCVERSION

MWCVERSION は、ミドルウェア製品のバージョン情報を格納するための構造体です（この構造体は mwclib.h にて定義しています）。

表 2 - 12 MWCVERSION 構造体

メンバ名	型	入出力	内 容
Version	int	出力	製品のバージョン情報
Serial	char*	出力	製品のシリアル番号
Others	int*	-	使用しません

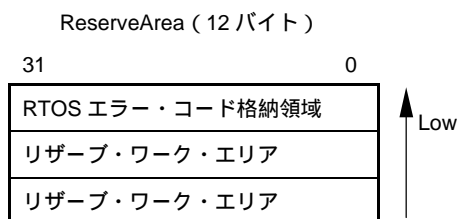
2.4.2 エラー・コード

表2-13 エラー・コード一覧

分類	シンボル	ID	内容
正常系	E_SUCCESS	0x0000	正常終了
異常系	E_COMERR ^{注1}	0x0101	コマンド・エラー
	E_COMSEQ	0x0102	コマンド・シーケンシャル・エラー
	E_PARAMETER	0x0103	パラメータ選択エラー
	E_PARVAL	0x0104	パラメータ設定値エラー
	E_MEMOVER	0x0105	メモリ・オーバ
	E_DATA	0x0106	入力データが不正
	E_CHARNUM	0x0107	文字数エラー
	E_DIC	0x0108	辞書未設定
	E_DICFORMAT	0x0109	辞書フォーマット・エラー
	E_OVERFLOW	0x010A	出力データ・オーバフロー
	E_TIMEOUT	0x010B	タイムアウトにより強制終了
	E_TERMINATE	0x010C	強制終了された
	E_NORUN	0x010D	処理中でない
	E_DONE	0x010E	処理が終了した場合
	E_DUPLI	0x010F	二重起動された
E_OVERWRIT	0x0110	音声バッファが上書きされる可能性あり	
E_WORDOVER	0x0111	単語数オーバ	
OS 関連	E_RTOS ^{注2}	0x0200	RTOS システム・コール・エラー
	E_RTOSTMOUT	0x0201	タイムアウト
	E_RTOSPOLFAIL	0x0202	条件不成立

注1. E_COMERR は、指定した辞書や音素データが壊れているときなどに出る致命的なエラーです。ミドルウェア・ライブラリがこのエラーを返した場合は、メモリの配置や正しいデータを使用しているかなどを調べてください。

2. ミドルウェア・ライブラリが E_RTOS を返した場合、TTSMEM 構造体の ReserveArea の先頭 4 バイトに RTOS システム・コール自身が返すエラー・コードが格納されています。エラー解析の参考にしてください。



2.4.3 外部インタフェース

(1) tts_AnalyseText 関数

【分類】 テキスト解析処理

【関数名】 tts_AnalyseText

【形式】 int Error = tts_AnalyseText(TextInfo, Mode, PhoneDataArea, DicInfoNum, DicInfo, WorkArea);

【引き数】 TTSTEXT *TextInfo 日本語かな漢字テキスト情報の格納領域の先頭アドレス

==TTSTEXT の構造==

```
typedef struct{
    char *TextIn; /*解析する日本語かな漢字テキストの格納領域の先頭アドレス(入力)*/
    char *NextText; /*次回のテキスト解析開始アドレス(出力)*/
}TTSTEXT;
```

int Mode 処理モード指定(入力)

TTSPHNBUF *PhoneDataArea 解析結果出力構造体の先頭アドレス(入力)

==TTSPHNBUF の構造==

```
typedef struct{
    char *PhoneData; /*解析結果出力領域の先頭アドレス(入力)*/
    int PhoneDataSize; /*解析結果出力領域のサイズ(入力)*/
}TTSPHNBUF;
```

int DicInfoNum 指定辞書数(入力)

TTSDIC *DicInfo 辞書情報構造体配列へのポインタ(入力)

==TTSDIC の構造==

```
typedef struct{
    long *DicArea; /*辞書バッファ先頭アドレス(入力)*/
    short DicState; /*辞書状態(0:使用しない, 1:使用する)(入力)*/
}TTSDIC;
```

long *WorkArea ワーク・メモリの先頭アドレス(入力)

【返り値】 E_SUCCESS 正常終了
 E_COMERR コマンド・エラー
 E_MEMOVER 内部で使用しているメモリがオーバーフローした
 E_DATA 文字列の先頭が0
 E_DIC システム辞書が1つも定義されていない
 システム辞書が2つ以上定義された
 E_DICFORMAT ユーザ辞書のフォーマットが不正
 E_OVERFLOW 出力データが音声生成部の入力の限界(512バイト)を越えた
 出力データが PhoneDataSize を越えた
 E_TERMINATE 強制終了された

(a) 動作概要

tts_AnalyseText 関数は、日本語かな漢字テキスト（SJIS コード）の読み、アクセントなどを解析して、発音記号データを生成します。

TextInfo->TextIn から始まる領域に格納されたテキスト文字列を Mode で指定する区切り文字または NULL まで読み込み、テキスト文字列を解析して、解析結果の発音記号データを PhoneDataArea ->PhoneData で指定された領域に格納して返します。

同時に TextInfo->NextText に次のテキスト解析の開始アドレスを格納します。次のテキスト解析開始アドレスは、最初に見つけた区切り文字のアドレス+1 が格納されます。

Mode で与える引き数により、図 2 - 8 に示す指定を行います。

図2 - 8 モードの指定

	31		24	23		16	15		8	7		0	
Mode	R	R	R	R	R	R	R	R	R	R	R	R	R

- a : 記号読みのモード指定 (0: 読まない, 1: 読む)
- b : 数字読みのモード指定 (0: 棒読み, 1: 桁付き)
- c : アルファベットの大文字 / 小文字の区別 (0: 区別する, 1: 区別しない)
- d : 区切り文字の指定 (0: OFF, 1: ON) (リターン)
- e : 区切り文字の指定 (0: OFF, 1: ON) (スペース, タブ)
- f : 区切り文字の指定 (0: OFF, 1: ON) (句点「。」「!」「?」)
- g : 区切り文字の指定 (0: OFF, 1: ON) (読点「、」)
- h : 区切り文字の指定 (0: OFF, 1: ON) (ピリオド)
- i : 区切り文字の指定 (0: OFF, 1: ON) (コロンの、セミコロン)
- j : スペース, TAB, 改行の完全無視 (0: OFF, 1: ON)
- k^注 : Ver.2.1 モード (0: OFF, 1: ON)
- R : 予約領域 (0 を指定してください)

注 Ver.2.2 ではテキスト解析処理におけるポーズの挿入規則を一部変更しております。
Ver.2.1 と同じ規則でテキスト解析処理を行う場合には“1”に設定してください。

(b) 区切り文字について

区切り文字を指定した場合は、半角、全角、ともに区切り文字として扱います。

区切り文字コードを表2-14に示します。

表2-14 区切り文字コード

区切り文字	全角 (2バイト・コード)	半角 (1バイト・コード)
リターン	0x0d0a	0x0a
スペース	0x8140	0x20
タブ	-	0x20
句点「。」	0x8142	0xa1
「！」	0x8149	0x21
「？」	0x8148	0x3f
ピリオド「.」	0x8144	0x2e
コロンの「:」	0x8146	0x3a
セミコロンの「;」	0x8147	0x3b

注意 1. 音声生成部 (tts_SyntheWave) は先頭に文末記号「。」があると E_DATA を返します。したがって、この場合は入力文字列を再び指定して、テキスト解析処理をやり直してください。

2. 入力テキスト文字列の先頭に区切り文字がある場合、その区切りが「スペース、タブ、リターン」の場合は区切り文字をとばして処理を行い、「スペース、タブ、リターン」以外の区切り文字の場合は、発音記号に「.」(0x2e)のみを出力し、次のテキスト解析の開始アドレスは区切り文字の次のアドレスを指します。このときの返り値は E_SUCCESS です。

(c) 辞書

tts_AnalyseText は、複数のユーザ辞書を使用できます。辞書の最大使用数を DicInfoNum で、辞書本体の指定を DicInfo で指定します。

システム辞書とユーザ辞書を混在する場合の検索順序は、ユーザ辞書のあとにシステム辞書を検索します。複数のユーザ辞書での検索の優先順位は、DicInfo 配列に登録した順に検索します。

TTS では次のシステム辞書を提供しています。

- ・システム辞書：tts_SysDic

(d) 注意事項

- ・入力文字列には、SJIS コードを使用してください。
- ・出力される発音記号列は 512 バイト以内になしてください。入力する日本語文字列を 256 バイト以内にするのを推奨します。
- ・入力した日本語かな漢字テキストは、解析中に変更しないでください。
- ・テキスト解析中は、ユーザ辞書の内容を変更しないでください。

- ・ ユーザ辞書のみでテキストを解析することはできません（システム辞書は必要です）。
- ・ tts_AnalyseText 関数は、引き数で指定された作業領域（WorkArea：23 K バイトの連続空間）を使用して動作します。この作業領域は、ユーザのアプリケーションで確保してください。
- ・ TextInfo, PhoneDataArea, DicInfo, WorkArea には、必ず 4 バイト align された領域を指定してください。
- ・ DicInfoNum には、使用するシステム辞書およびユーザ辞書の数を指定します。DicInfoNum に DicInfo 構造体配列の数より大きい値を指定しないでください。

(e) 補 足

入力文字列が非常に長い場合、内部で使用しているメモリがオーバーフローする場合があります。エラー（E_MEMOVER）が返った場合は、区切り文字を変更するなどの処置をして、入力文字列が短くなるようにしてください。入力文字列は、256 バイト以下にすることを推奨します。

(2) tts_StopAnalyse 関数

- 【分 類】 テキスト解析処理停止要求
- 【関 数 名】 tts_StopAnalyse
- 【機能概要】 テキスト解析中の処理を中止します。
- 【形 式】 int Error = tts_StopAnalyse(WorkArea);
- 【引 き 数】 long *WorkArea ワーク・メモリ情報を格納する領域の先頭アドレス（入力）
- 【返 り 値】 E_SUCCESS 正常終了

(a) 動作概要

tts_AnalyseText()の処理を中止します。解析処理の再開はできません。

tts_Analyse()が処理中でないときに、tts_StopAnalyse()を実行した場合、tts_StopAnalyse()は無視されます。その場合、tts_StopAnalyse()の返却値は E_SUCCESS になります。

(b) 注意事項

WorkArea には tts_AnalyseText に渡したのと同じ領域を指定します。

(3) tts_Initialize 関数

【分 類】 初期化処理部

【関 数 名】 tts_Initialize

【形 式】 int Error = tts_Initialize(OsInfo, MemInfo);

【引 き 数】 TTSOSINFO *OsInfo OS 情報を格納した領域の先頭アドレス (入力)

==TTSOSINFO の構造==

```
typedef struct{
    short flg_ID;          /*イベント・フラグの ID (入力)*/
}TTSOSINFO;
```

TTSMEM *MemInfo ワーク・メモリ情報を格納する領域の先頭アドレス (入力)

==TTSMEM の構造==

```
typedef struct{
    long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力)*/
    long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力)*/
}TTSMEM;
```

【返 り 値】 E_SUCCESS 正常終了
 E_COMERR コマンド・エラー
 E_RTOS RTOS システム・コール・エラー

(a) 動作概要

TTS で使用する OS 資源 (イベント・フラグ) の生成処理, ReserveArea の初期化を行う関数です。システム起動時に tts_SyntheWave, tts_StopSynthe, tts_RequestWave, tts_Uninitialize 関数を呼び出す前に一度だけ呼び出します。メモリは 12 バイトの ReserveArea を使用します。

(b) 注意事項

- ・ MemInfo には, 必ず 4 バイト align された領域を指定してください。
- ・ ReserveArea は, tts_Uninitialize が呼び出されるまで壊さないでください。WorkArea に対しては何も行いません。
- ・ OS を使用しない場合は flg_ID (OSInfo のメンバ) の値は使用しません。

(4) tts_Uninitialize 関数

【分 類】 終了処理部

【関 数 名】 tts_Uninitialize

【形 式】 int Error = tts_Uninitialize(MemInfo);

【引 き 数】 TTSMEM *MemInfo ワーク・メモリ情報を格納する領域の先頭アドレス (入力)

==TTSMEM の構造==

```
typedef struct{
    long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力)*/
    long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力)*/
}TTSMEM;
```

【返 り 値】 E_SUCCESS 正常終了

E_COMERR コマンド・エラー

E_RTOS RTOS システム・コール・エラー

(a) 動作概要

TTS で使用する OS 資源 (イベント・フラグ) の削除を行う関数です。システム終了時に呼び出します。

(b) 注意事項

- ・ MemInfo には、必ず 4 バイト align された領域を指定してください。
- ・ ReserveArea は、tts_Initialize に渡したものと同一領域を指定します。WorkArea に対しては何も行いません。

(5) tts_SyntheWave 関数

【分類】 音声生成処理部

【関数名】 tts_SyntheWave

【形式】 int Error = tts_SyntheWave (PhoneInfo, VoiceInfo, VoiceMode, MemInfo);

【引き数】 TTSPHON *PhoneInfo 発音情報領域の先頭アドレス

==TTSPHON の構造==

```
typedef struct{
    char *PhoneData; /*発音記号データ先頭アドレス(入力)*/
    char *PhoneInfo; /*拡張発音情報先頭アドレス(入力)*/
    char *NextData; /*次の音声合成開始アドレス(出力)*/
}TTSPHON;
```

TTSVOICE *VoiceInfo 音声データ出力領域指定構造体のアドレス

==TTSVOICE の構造==

```
typedef struct{
    short *BufStart; /*出力バッファ開始アドレス(入力)*/
    short *BufEnd; /*出力バッファ終了アドレス(入力)*/
    int InitBlockCount; /*初期音声生成ブロック数(入力)*/
    long CurrentSize; /*生成音声データ・サイズ(出力)*/
    int Status; /*生成完了フラグ(出力)*/
}TTSVOICE;
```

TTSVOIMOD *VoiceMode 出力音声指定

==TTSVOIMOD の構造==

```
typedef struct{
    long *Character; /*音素データ領域の先頭アドレス(入力)*/
    short Pitch; /*声の高さ指定(入力)*/
    short Speed; /*読み上げ速度指定(入力)*/
    short Accent; /*アクセントの強弱指定(入力)*/
}TTSVOIMOD;
```

TTSMEM *MemInfo ワーク・メモリ情報を格納する領域の先頭アドレス(入力)

==TTSMEM の構造==

```
typedef struct{
    long *ReserveArea; /*リザーブ・エリア(保存用)先頭アドレス(入力)*/
    long *WorkArea; /*ワーク・エリア(関数ごとに使用)先頭アドレス(入力)*/
}TTSMEM;
```

【返り値】	E_SUCCESS	正常終了
	E_COMERR	コマンド・エラー
	E_PARVAL	パラメータ(ピッチ, スピード, アクセント)の値が不正
	E_MEMOVER	内部で使用しているメモリがオーバーフローした
	E_DATA	不正な発音記号フォーマットが入力された
	E_CHARNUM	入力文字数が512バイトを越えている
	E_OVERFLOW	出力データがオーバーフローした
	E_TERMINATE	tts_StopSyntheにより中断された
	E_RTOS	RTOS システム・コール・エラー

(a) 動作概要

tts_SyntheWave 関数は、tts_AnalyseText で解析した発音記号データを改行コードまたは NULL まで読み込み、音声データを生成します。発音記号は、ローマ字表記の発音記号とカタカナ表記の発音記号の2種類があります。詳細については**付録 A 発音記号インタフェース**を参照してください。

PhoneData により指定されたアドレスから発音記号データを読み込み、音声データを生成して、それを BufStart/BufEnd で指定された領域に書き込みます。同時に生成された音声データのサイズと、読み込んだ発音記号のデータの次のアドレスを指定された領域に書き込みます。出力バッファの大きさは 8 K バイトの整数倍を指定してください。

また、拡張発音情報がある場合には PhonelInfo にその先頭アドレスを入れ、拡張情報がない場合には PhonelInfo に NULL を指定します。

なお、一度に入力できる発音記号データ（ローマ字発音記号）の長さは 512 バイトまでです。カタカナ表記の発音記号を使用する場合は、**表 A - 2 カナ表記発音記号 ローマ字表記発音記号の対応**を参考に対応するローマ字発音記号の合計を 512 バイト以下にしてください。

tts_SyntheWave 関数は、動作上 128 K バイトの作業領域（WorkArea）が必要になります。アプリケーション・プログラムで作業領域を確保して、そのアドレスを引き数として渡します。

ReserveArea には tts_Initialize に渡したものと同一領域を指定します。

TTS の音声生成処理には、リアルタイム・モードとバッチ・モードがあります。リアルタイム・モード使用時は、InitBlockCount に初期音声生成ブロック数を指定します。バッチ・モード使用時は、InitBlockCount に 0 を指定します。

(i) リアルタイム・モード (DEFAULT)

音声生成処理と音声出力処理（ユーザが用意する割り込みハンドラで実現）を同期して行います。リアルタイム・モードを使用すると、音声出力バッファを有効に利用できます。

(ii) バッチ・モード

入力した発音記号列を一括して音声データに変換します。生成される音声データよりも出力バッファが小さいときは、出力バッファ分の音声データを生成して処理を終了します。

(b) モード指定

tts_SyntheWave 関数では、読み上げる音声の声質をパラメータによって変化させることができます。変更できるパラメータを次に示します。

- ・ 男声/女声の切り替え（音素データの切り替え）
- ・ 声の高さ（ピッチ）の変更
- ・ 読み上げ速度の変更
- ・ アクセントの強弱の変更

(i) 男声 / 女声の切り替え (音素データの切り替え)

VoiceMode->Character に音素データの先頭アドレスを指定することにより、発声する音声を切り替えることができます。TTS では表 2 - 15 に示す音素データを提供しています。

表2 - 15 音素データ

サンプリング・レート	Character	シンボル名
8 kHz	男声	tts_male8
	女声	tts_female8
11 kHz	男声	tts_male11
	女声 1	tts_female11
	女声 2	tts_female11_2

(ii) 声の高さ (ピッチ) の変更

VoiceMode->Pitch に、- 10 ~ 10 までの整数を指定することによってピッチを変更できます (最低ピッチは - 10, 最高ピッチは 10, 標準ピッチは 0 を指定します)。

(iii) 読み上げ速度の変更

VoiceMode->Speed に、- 10 ~ 10 までの整数を指定することによって読み上げ速度を変更できます (低速は - 10, 高速は 10, 標準は 0 を指定します)。

(iv) アクセント強弱の変更

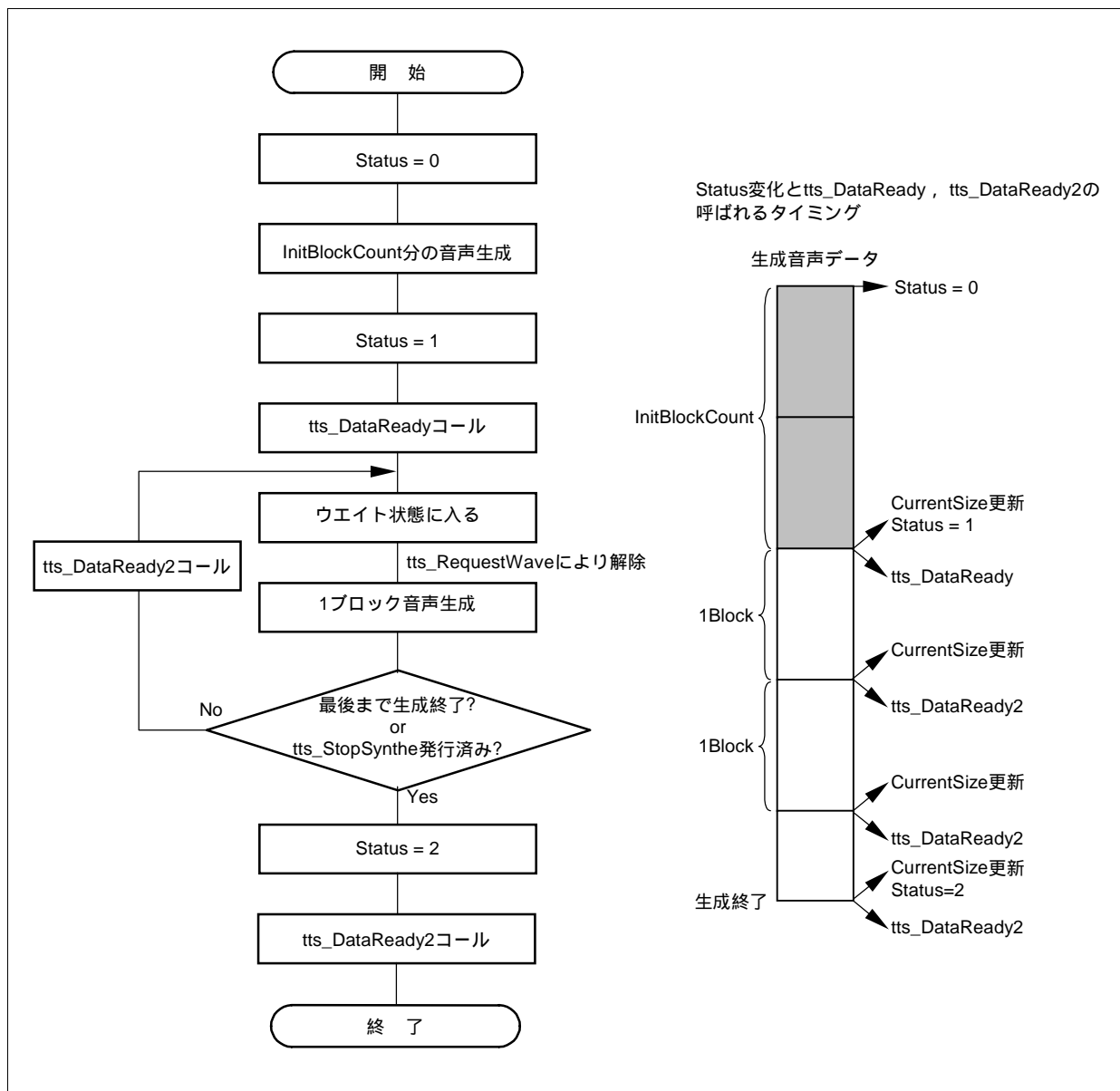
VoiceMode->Accent に、- 10 ~ 10 までの整数を指定することによってアクセントの強弱を変更できます (最弱は - 10, 最強は 10, 標準は 0 を指定します)。

(c) 注意事項

- ・発音記号として不正な文字を入力した場合の動作は保証できません。
- ・発音記号が 512 バイトを越えたとき、音声データは出力されません。
- ・拡張発音情報として不正な文字が入力された場合の動作は保証されてません。
- ・音素データの先頭アドレスを指定するパラメータに音素データがない場合の動作は保証できません。
- ・出力バッファの大きさが 8 K バイトの整数倍でないときの動作は保証できません。
- ・tts_SyntheWave 関数が戻るまでは、発音記号列を変更しないでください。
- ・PhoneInfo, VoiceInfo, VoiceMode, MemInfo には、必ず 4 バイト align された領域を指定してください。
- ・入力する発音記号列の先頭が「.」(0x2e) または NULL の場合、E_DATA を返し終了します。

(d) 注意事項

図2-9 TTS音声生成処理内部の動作



tts_SyntheWave 関数は処理を開始すると Status=0 になり、InitBlockCount で指定したブロック数 (1 ブロック : 8 K バイト) 分の音声データを作成して、ウェイト状態になります (tts_RequestWave 関数により解除)。その際に Status=1 になり、TTS は tts_DataReady 関数を呼び出します。

その後、tts_SyntheWave 関数は、tts_RequestWave 関数が呼び出されるごとに 1 ブロックずつデータを生成し tts_DataReady2 関数を呼び出して、ウェイト状態になります。

tts_SyntheWave を呼び出してから tts_DataReady または tts_DataReady2 が呼ばれるまでに生成された音声データのサイズ (単位 : バイト) は CurrentSize に格納され、すべてのデータの生成が終了すると、Status=2 になります。

tts_DataReady, tts_DataReady2 関数は TTS が呼び出す関数で、ソース・プログラムの形で提供されます。関数の内容は、デフォルトでは return();のみ記述されていますが、tts_DataReady 関数内に割り込みを許可する命令などを記述すると、音声データが生成されるまで割り込みを禁止したり、また tts_DataReady2 関数を用い、割り込みを使用しないで1ブロックずつ音声データを転送したりできます。

```
void tts_DataReady(void)
{
    return;          /* デフォルトでは return();のみ */
}
```

```
void tts_DataReady2(void)
{
    return;          /* デフォルトでは return();のみ */
}
```

(e) 補 足

- E_MEMOVER について

女声 11 kHz サンプリングの音声を、pitch=(9 or 10)かつ speed=(- 9 or - 10)のパラメータで使用する場合、入力発音記号列が 400 バイト以上あると内部で使用しているメモリがオーバーフローする場合があります。上記のパラメータで使用する場合、入力発音記号列を 400 バイト未満にすることを推奨します（母音が非常に多い発音記号列の場合、400 バイト未満でもエラー（E_MEMOVER）を返すことがあります）。

- E_DATA について

入力信号例が「.」（0x2e）のみなど、音声データがまったく生成されない不正な発音記号である場合、E_DATA のエラーを返します。この場合、出力領域には無音データ（0 データ）が1サンプル（2 バイト）分出力されます。このとき CurrentSize=2 となります。

(6) tts_StopSynthe 関数

- 【分類】 処理停止要求
- 【関数名】 tts_StopSynthe
- 【機能概要】 音声生成中の処理を中止します。
- 【形式】 int Error = tts_StopSynthe (MemInfo);
- 【引き数】 TTSMEM *MemInfo ワーク・メモリ情報を格納する領域の先頭アドレス (入力)
- ==TTSMEM の構造==
- ```
typedef struct{
 long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力)*/
 long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力)*/
}TTSMEM;
```
- 【返り値】 E\_SUCCESS            正常終了  
           E\_COMERR          コマンド・エラー  
           E\_RTOS             RTOS システム・コール・エラー

## (a) 動作概要

tts\_SyntheWave()の処理を中止します。発声の再開はできません。

tts\_SyntheWave()が実行中でないときに音声生成処理停止要求 ( tts\_StopSynthe ) を発行した場合もストップ要求は有効になります。このとき、次に tts\_SyntheWave()を発行したときに tts\_SyntheWave()が E\_TERMINATE を返し終了します。

ストップ要求は「tts\_SyntheWave()が E\_TERMINATE を返す」または「tts\_ClearStopSynthe()を発行する」まで有効となります。

また tts\_StopSynthe()発行後は、生成したデータ・サイズは更新されずに、tts\_SyntheWave()は tts\_DataReady または tts\_DataReady2 を呼び出したあと終了します。

## (b) 注意事項

- MemInfo には、必ず 4 バイト align された領域を指定してください。
- ReserveArea に tts\_Initialize に渡したのと同じ領域を指定します。WorkArea は必要ありません。



## (7) tts\_ClearStopSynthe 関数

【分類】 停止要求クリア

【関数名】 tts\_ClearStopSynthe

【機能概要】 音声生成中の処理停止要求を解除します。

【形式】 int Error = tts\_ClearStopSynthe ( MemInfo );

【引き数】 TTSMEM \*MemInfo      ワーク・メモリ情報を格納する領域の先頭アドレス (入力)

==TTSMEM の構造==

```
typedef struct{
 long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力)*/
 long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力)*/
}TTSMEM;
```

【返り値】 E\_SUCCESS            正常終了

E\_COMERR            コマンド・エラー

E\_RTOS            RTOS システム・コール・エラー

## (a) 動作概要

tts\_SyntheWave()による音声生成停止要求を解除します。

音声生成処理停止要求が有効でないときに発行された場合は無視されます。このときの戻り値は E\_SUCCESS となります。

## (b) 注意事項

- ・ MemInfo には、必ず 4 バイト align された領域を指定してください。
- ・ ReserveArea に tts\_Initialize に渡したのと同じ領域を指定します。WorkArea は必要ありません。

**(8) tts\_RequestWave 関数**

【分 類】 音声データ要求処理部

【関 数 名】 tts\_RequestWave

【機能概要】 1 ブロック分の音声データの生成を音声生成部に要求します。

【形 式】 int Error = tts\_RequestWave ( MemInfo );

【引 き 数】 TTSMEM \*MemInfo      ワーク・メモリ情報を格納する領域の先頭アドレス ( 入力 )

==TTSMEM の構造==

```
typedef struct{
 long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力)*/
 long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力)*/
}TTSMEM;
```

【返 り 値】 E\_SUCCESS      正常終了

E\_COMERR      コマンド・エラー

E\_RTOS      RTOS システム・コール・エラー

**(a) 動作概要**

音声生成部に対して, 1 ブロック分 ( 8 K バイト ) の音声データの出力を要求します。

**(b) 注意事項**

- ・ MemInfo には, 必ず 4 バイト align された領域を指定してください。
- ・ ReserveArea に tts\_Initialize に渡したのと同じ領域を指定します。WorkArea は必要ありません。

## (9) tts\_DicInitialize 関数

- 【分類】 ユーザ辞書初期化処理部
- 【関数名】 tts\_DicInitialize
- 【機能概要】 ユーザ辞書作成処理を初期化します。
- 【形式】 `int Error = tts_DicInitialize ( WordNum, DicSize, WriteOpt, DicMemInfo );`
- 【引き数】 `short WordNum` ユーザ辞書登録文字列数 (入力)
- `int DicSize` 確保した辞書領域のサイズ (入力)
- `short WriteOpt` メモリ書き込みオプション (入力)
- `TTSMEM *DicMemInfo` ワーク・メモリ情報を格納する領域の先頭アドレス (入力)
- ==TTSMEM の構造==
- ```
typedef struct{
    long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力)*/
    long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力)*/
}TTSMEM;
```
- 【戻り値】 `E_SUCCESS` 正常終了
- `E_COMERR` コマンド・エラー

(a) 動作概要

ユーザ辞書登録処理を初期化します。WriteOpt (メモリ書き込みオプション) の指定によって、tts_DicEntry() で辞書情報をメモリに書き込む / 書き込まないを指定します。TTS ミドルウェアは、「メモリに書き込む」を指定したときだけ、辞書が作成されます。

WriteOpt = 0 : メモリに書き込まない

WriteOpt = 1 : メモリに書き込む

DicSize には、確保した辞書領域のサイズを指定します。

作成される辞書のサイズは、tts_DicInitialize を WriteOpt = 0 で発行したあと、tts_DicEntry を発行すると確認できます。この場合は、ユーザ辞書は作成されません。

(b) 注意事項

メモリは ((登録数 × 8) + 100) バイトの ReserveArea を使用します。ReserveArea は辞書登録が終了するまで壊さないでください。WorkArea は必要ありません。

(10) tts_DicEntry 関数

【分類】	ユーザ辞書管理
【関数名】	tts_DicEntry
【機能概要】	ユーザ辞書に文字列をセットします。
【形式】	int Error = tts_DicEntry (InputWord, DicArea, DicAreaSize, DicMemInfo);
【引き数】	char *InputWord 入力単語文字列の先頭アドレス (入力)
	long *DicArea ユーザ辞書領域 (出力)
	int *DicAreaSize 辞書情報のサイズ (出力)
	TTSMEM *DicMemInfo ワーク・メモリ情報を格納する領域の先頭アドレス (入力)
	==TTSMEM の構造==
	typedef struct{
	long *ReserveArea; /*リザーブ・エリア (保存用) 先頭アドレス (入力) */
	long *WorkArea; /*ワーク・エリア (関数ごとに使用) 先頭アドレス (入力) */
	}TTSMEM;
【返り値】	E_SUCCESS 正常終了
	E_COMERR コマンド・エラー
	E_COMSEQ コマンド・シーケンシャル・エラー
	tts_DicInitialize()が未コール
	E_MEMOVER メモリ・オーバ
	E_DATA 不正な文字列を入れた
	E_WORDOVER tts_DicInitialize()で指定した登録数を越えた

(a) 動作概要

指定した文字列を基に辞書情報を作成して、DicArea で指定した領域に書き込みます。

tts_DicInitialize()で WriteOpt = 0 (メモリに書き込まない) を指定した場合は、辞書情報は DicArea に書き込まれず、使用する DicArea のサイズだけが DicAreaSize で与えられた領域に書き込まれます。このとき、DicArea で指定される領域に対しては何も行われません。

連続した作業領域 (WorkArea) は、200 バイト必要です。アプリケーション・プログラムで作業領域を確保して、そのアドレスを引き数として渡してください。また、ReserveArea には tts_DicInitialize に渡したのと同じ領域を指定します。

ユーザ辞書文字列は複数に分けて入力することができます。入力文字列については 3.3 入力フォーマットを参照してください。最終行のあとには必ず NULL を入れてください。

最終行の改行コードは省略できます。

(b) 注意事項

- ・ 0x1a の EOF コードは使用できません。入力単語文字列の最後に 0x1a がある場合、E_WORDOVER になります。
- ・ DicArea には必ず 4 バイト align された領域を指定してください。

== tts_DicEntry 入力文字列フォーマット ==

品詞コード	表記	読み[ret]
品詞コード	表記	読み[ret]
⋮		
NULL		

備考 : スペース (半角を使用してください)

品詞番号には半角数字を使用してください。入力文字列中に記述する品詞番号を表 2 - 16 に示します。

表2 - 16 品詞コード

品詞コード	品 詞
0	連体詞
1	感動詞
2	接続詞
3	副詞
4	形容詞
5	形容動詞
6	サ変名詞
7	普通名詞
8	固有名詞
9	カ行五段活用動詞
10	ガ行五段活用動詞
11	サ行五段活用動詞
12	タ行五段活用動詞
13	ナ行五段活用動詞
14	バ行五段活用動詞
15	マ行五段活用動詞
16	ラ行五段活用動詞
17	ア,ワ行五段活用動詞
18	上一段活用動詞
19	下一段活用動詞
20	記号

(11) tts_GetVersion 関数

【分類】 バージョン情報出力要求

【関数名】 tts_GetVersion

【機能概要】 TTS のバージョン情報を出力します。

【形式】 int Error = tts_GetVersion (VersionPacket);

【引き数】 MWCVERSION *VersionPacket バージョン情報へのポインタを格納する領域の
アドレス (出力)

==MWCVERSION の構造==

```
typedef struct{
    int Version;          /*製品のバージョン情報(出力)*/
    char *Serial;        /*製品のシリアル番号(出力)*/
    int *Others;         /*(使用しません)*/
}MWCVERSION;
```

【戻り値】 E_SUCCESS 正常終了

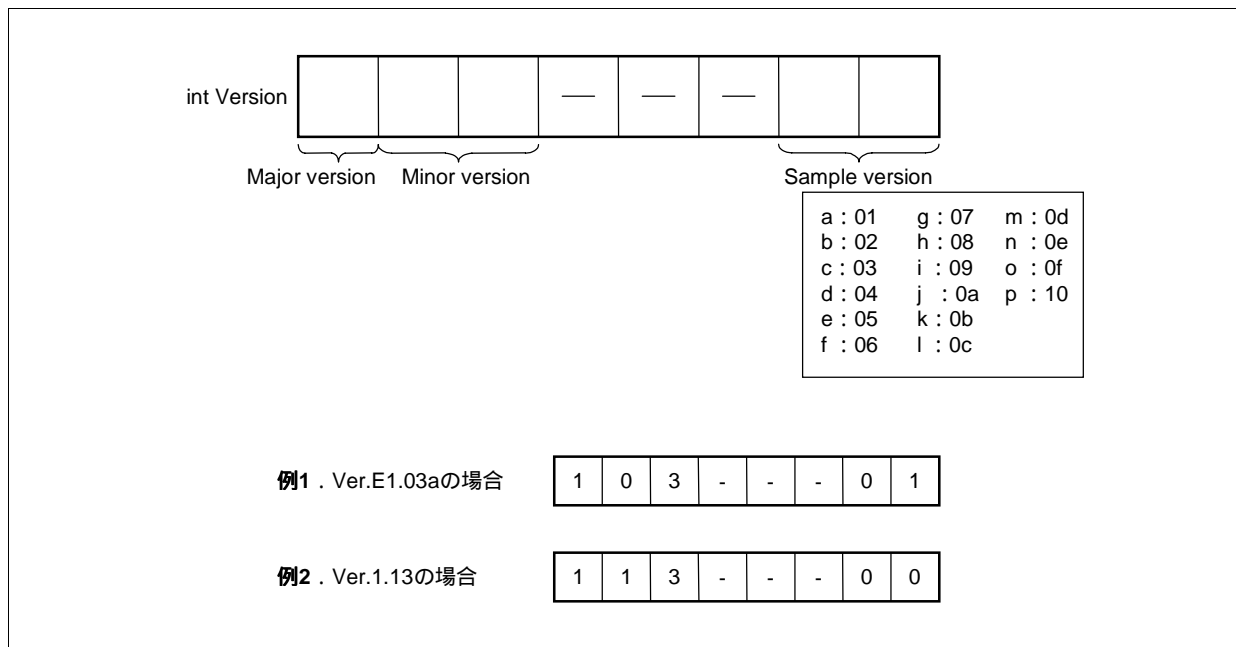
(a) 動作概要

TTS のバージョン情報が格納された領域のアドレスを VersionPacket に出力します。

tts_GetVersion では、作業領域は必要ありません。

Version は、製品バージョンが図 2 - 10 に示すフォーマットで格納されます。

図2 - 10 バージョン番号のフォーマット



(12) tts_DataReady 関数

- 【分 類】 音声生成処理終了通知 (ユーザ定義コール・バック)
- 【関 数 名】 tts_DataReady
- 【機能概要】 (ユーザ定義)
- 【形 式】 void tts_DataReady();
- 【引 数】 なし
- 【返 り 値】 なし

(a) 動作概要

tts_SyntheWave 関数は音声生成処理を開始し、InitBlockCount で指定したブロック数 (1 ブロック : 8 K バイト) 分の音声データを生成した後、ウェイト状態になります。そのとき、音声生成処理部は tts_DataReady 関数を呼び出します。

(b) 注意事項

tts_DataReady 関数は、ソース・プログラムの形で提供されます。関数の内容は、デフォルトでは return();のみ記述されています。

(13) tts_DataReady2 関数

- 【分 類】 音声生成処理終了通知 (ユーザ定義コール・バック)
- 【関 数 名】 tts_DataReady2
- 【機能概要】 (ユーザ定義)
- 【形 式】 void tts_DataReady2();
- 【引 数】 なし
- 【返 り 値】 なし

(a) 動作概要

音声生成処理部は tts_RequestWave 関数が呼び出されるごとに 1 ブロックずつデータを生成します。1 ブロック分の音声データを生成終了後、ウェイト状態になります。そのとき、音声生成処理部は tts_DataReady2 関数を呼び出します。

(b) 注意事項

tts_DataReady2 関数は、ソース・プログラムの形で提供されます。関数の内容は、デフォルトでは return();のみ記述されています。

(14) mwc_CreFlg 関数

- 【分 類】 RTOS インタフェース部
- 【関 数 名】 mwc_CreFlg
- 【機能概要】 イベント・フラグを生成します。
- 【形 式】 int Error = mwc_CreFlg (short ID , int *AccAdr , int *ErrorCode);
- 【引 き 数】 short ID イベント・フラグの ID <入力>
int *AccAdr イベント・フラグ・アクセス・アドレスを格納する領域 <出力>
int *ErrorCode エラー・コード格納領域 <出力>
- 【返 り 値】 E_SUCCESS 正常終了
E_RTOS システム・コール・エラー
- 【機 能】 イベント・フラグを生成し、*AccAdr で示される領域にイベント・フラグ・アクセス・アドレスを格納します。

(15) mwc_DelFlg 関数

- 【分 類】 RTOS インタフェース部
- 【関 数 名】 mwc_DelFlg
- 【機能概要】 イベント・フラグを削除します。
- 【形 式】 int Error = mwc_DelFlg (int AccAdr , int *ErrorCode);
- 【引 き 数】 int AccAdr イベント・フラグ・アクセス・アドレス<入力>
int *ErrorCode エラー・コード格納領域 <出力>
- 【返 り 値】 E_SUCCESS 正常終了
E_RTOS システム・コール・エラー
- 【機 能】 AccAdr で示されるイベント・フラグを削除します。

(16) mwc_SetFlg 関数

- 【分 類】 RTOS インタフェース部
- 【関 数 名】 mwc_SetFlg
- 【機能概要】 イベント・フラグをセットします。
- 【形 式】 int Error = mwc_SetFlg (int AccAdr , unsigned int setptn , int *ErrorCode);
- 【引 き 数】 int AccAdr イベント・フラグ・アクセス・アドレス<入力>
unsigned int setptn 設定するビット・パターン<入力>
int *ErrorCode エラー・コード格納領域 <出力>
- 【返 り 値】 E_SUCCESS 正常終了
E_RTOS システム・コール・エラー
- 【機 能】 AccAdr で示されるイベント・フラグを指定したビット・パターンでセットします。

(17) mwc_ClrFlg 関数

- 【分 類】 RTOS インタフェース部
- 【関 数 名】 mwc_ClrFlg
- 【機能概要】 イベント・フラグをクリアします。
- 【形 式】 `int Error = mwc_ClrFlg (int AccAdr , unsigned int clrptn , int *ErrorCode);`
- 【引 き 数】 `int AccAdr` イベント・フラグ・アクセス・アドレス<入力>
`unsigned int clrptn` クリアするビット・パターン<出力>
`int *ErrorCode` エラー・コード格納領域 <出力>
- 【返 り 値】 `E_SUCCESS` 正常終了
`E_RTOS` システム・コール・エラー
- 【機 能】 `AccAdr` で示されるイベント・フラグを指定したビット・パターンでクリアします。

(18) mwc_WaiFlg 関数

- 【分 類】 RTOS インタフェース部
- 【関 数 名】 mwc_WaiFlg
- 【機能概要】 イベント・フラグのチェックを行います。
- 【形 式】 `int Error = mwc_ClrFlg (int AccAdr , unsigned int waiptn , unsigned int wfmode ,
int *p_prtn , int *ErrorCode);`
- 【引 き 数】 `int AccAdr` イベント・フラグ・アクセス・アドレス<入力>
`unsigned int waiptn` 要求するビット・パターン<入力>
`unsigned int wfmode` 命令オプション<入力>
`wfmode = 0` AND 待ち (リセット指定なし)
`wfmode = 1` AND 待ち (リセット指定あり)
`wfmode = 2` OR 待ち (リセット指定なし)
`wfmode = 3` OR 待ち (リセット指定あり)
`int *p_prtn` 条件成立時のビット・パターンを格納するアドレス<出力>
`int *ErrorCode` エラー・コード格納領域 <出力>
- 【返 り 値】 `E_SUCCESS` 正常終了
`E_RTOS` システム・コール・エラー
- 【機 能】 `AccAdr` で示されるイベント・フラグを指定したビット・パターンをチェックし, `waiptn` で指定したビット・パターンになるまで Wait 状態になります。

(19) mwc_PolFlg 関数

- 【分類】 RTOS インタフェース部
- 【関数名】 mwc_PolFlg
- 【機能概要】 イベント・フラグのチェックを行います。
- 【形式】 `int Error = mwc_PolFlg (int AccAdr , unsigned int waiptn , unsigned int wfmode ,
int *p_prtn , int *ErrorCode);`
- 【引き数】 `int AccAdr` イベント・フラグ・アクセス・アドレス<入力>
`unsigned int waiptn` 要求するビット・パターン<入力>
`unsigned int wfmode` 命令オプション<入力>
`wfmode = 0` AND 待ち (リセット指定なし)
`wfmode = 1` AND 待ち (リセット指定あり)
`wfmode = 2` OR 待ち (リセット指定なし)
`wfmode = 3` OR 待ち (リセット指定あり)
- `int *p_prtn` 条件成立時のビット・パターンを格納するアドレス<出力>
`int *ErrorCode` エラー・コード格納領域 <出力>
- 【戻り値】 `E_SUCCESS` 正常終了
`E_RTOS` システム・コール・エラー
- 【機能】 `AccAdr` で示されるイベント・フラグのビット・パターンが `waiptn` で指定したビット・パターンと同じかどうかチェックします。

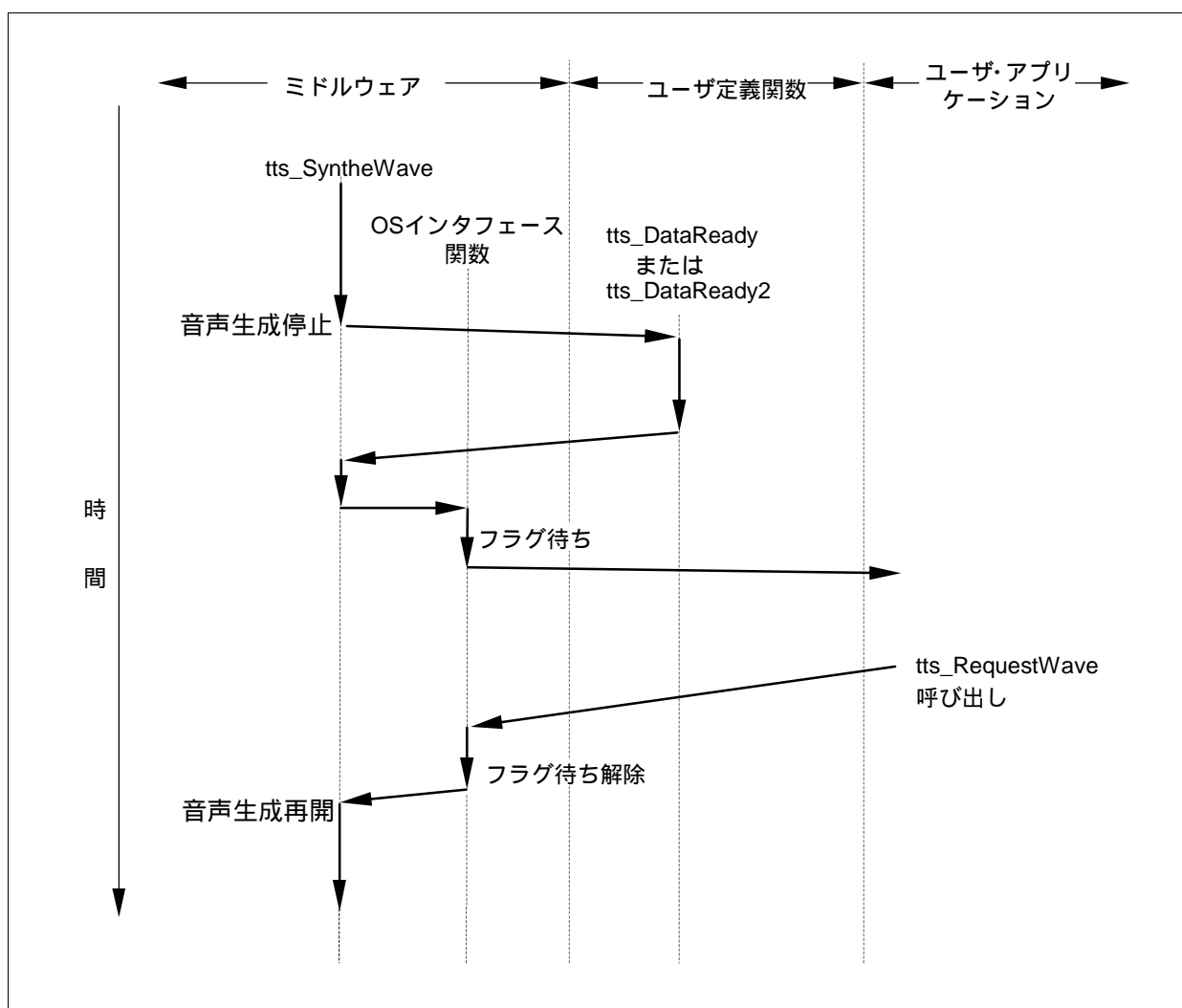
(a) 注意事項

OS インタフェース関数は OS を使用する / しないに関わらず、次のようにリンクしてください。

- ・ RX830 (ITRON1) を使用する場合 mwcrcx02.c
- ・ RX830 (μ ITRON Ver.3.0) を使用する場合 mwcrcx01.c
- ・ OS を使用しない場合 mwcnoos.c
- ・ RX830 以外の OS を使用する場合 ユーザ定義

OS を使用する場合、システムコールを OS インタフェース関数内に記述します。OS インタフェース関数は TTS から呼び出され、フラグ処理を行います。

図2 - 11 tts_SyntheWave()内でのOSインタフェース関数の呼び出し



[メ モ]

第3章 ユーザ辞書の作成ツール

3.1 概要

ユーザ辞書は、システム辞書に登録されていない単語などを登録するために使用します。TTS ミドルウェアでは、オンボード上 (V830 ファミリ) でのユーザ辞書作成ライブラリのほかにホスト・マシン上で動作するツールも用意しています。この辞書作成ツールは、テキスト形式の登録データを、TTS ミドルウェアで使用できる辞書 (バイナリ・データ) に変換するプログラムです。

3.2 コマンド

ユーザ辞書作成ツールの使用方法を説明します。実行方法を次に示します。

```
% udicmake input output [ret]
```

udicmake : コマンド名
input : 入力するテキスト・ファイル名
output : 出力される辞書のファイル名

3.3 入力フォーマット

入力テキストは「品詞コード」, 「表記」, 「読み」の順にスペースを挟んで記述します。記述する際は次に示す規則に従ってください。

1 行に 1 項目ずつ記述してください。

行末に改行コードを入れてください。

表記は最大 36 バイト, 読みは最大 72 バイトまでにしてください。

日本語文字コードは SJIS を使用してください。

読みは全角ひらがなで記述して, 「'」でアクセントの位置を, 「,」でポーズの位置を指定してください。

アクセントの位置, ポーズの位置は, 半角での記述もできます。

最終行は改行のあとに「¥0」を付けてください。tts_DicEntry()を使用する場合, 最終行の改行は省略できます。

品詞コードは半角で記述してください (品詞コードには全角を使用しないでください)。

「品詞コード」, 「表記」, 「読み」の区切りに入るスペースは半角を使用してください。

1 つの辞書には 32767 個までの単語を登録できます。これを越える数の単語を登録する場合は, 複数の辞書データに分割してください。

「読み」を記述するときに, 濁音 (「が」など), 撥音 (「ば」など) を使用する場合は「`」, (0x814a), 「°」(0x814b) を使用しないでください。

例: 「読み」に「かばん」と記述する場合

正しい記述: かばん (0x82a9, 0x82ce, 0x82fl)

不正な記述: かは°ん (0x82a9, 0x82cd, 0x814a, 0x82fl)

次にフォーマットの例を示します（ はスペースを示します：半角を使用してください）。

品詞コード 表記 読み

- 例 8 TTS てきすととーすびーち [ret]
 8 NEC えぬ/いーしー [ret]
 8 日本電気 えぬ/いーしー [ret]

上記の例では「TTS」という文字列を「てきすととーすびーち」という読み方で、「NEC」、「日本電気」という文字列を「えぬ/いーしー」という読み方で登録しています。

「/」は、直前の音節にアクセントがあることを表します。1単語中にアクセントが複数ある場合は、アクセントの区切りに「/」を入力します。

ユーザ辞書に記述する品詞コードを表 3-1 に示します。品詞番号には必ず半角数字を使用してください。

表3-1 品詞コード

品詞コード	品 詞	品詞コード	品 詞	品詞コード	品 詞
0	連体詞	7	普通名詞	14	八行五段活用動詞
1	感動詞	8	固有名詞	15	マ行五段活用動詞
2	接続詞	9	カ行五段活用動詞	16	ラ行五段活用動詞
3	副詞	10	ガ行五段活用動詞	17	ア,ワ行五段活用動詞
4	形容詞	11	サ行五段活用動詞	18	上一段活用動詞
5	形容動詞	12	タ行五段活用動詞	19	下一段活用動詞
6	サ変名詞	13	ナ行五段活用動詞	20	記号

3.4 出力データ

次に示すデータが出力されます。

- ・ ユーザ辞書 バイナリ・ファイル
- ・ 登録数 標準出力
- ・ 辞書サイズ [バイト] 標準出力

3.5 エラー

エラーが発生した場合、エラーが発生した時点で次に示すメッセージを出力して、停止します。登録番号は、不正な文字が何番目の登録にあるかを示しています。

Error : エラー・メッセージ (登録番号)

次にエラー・メッセージを示します。

表3-2 エラー・メッセージ

エラー・メッセージ	意味	備考
E_HINSI	品詞番号が不正です。	
E_HYOUSOU	表層のフォーマットが不正です。	
E_YOMI	読みのフォーマットが不正です。	
E_WORDNUM	登録数が最大値を越えています。	最大値は 32767 個です。
E_MEMOVER	メモリ・オーバ	登録番号は出力されません。
E_FILEOPEN	ファイル・オープン・エラー	登録番号は出力されません。

[メ モ]

第 4 章 スピーチ・デザイナー対応機能

4.1 概 要

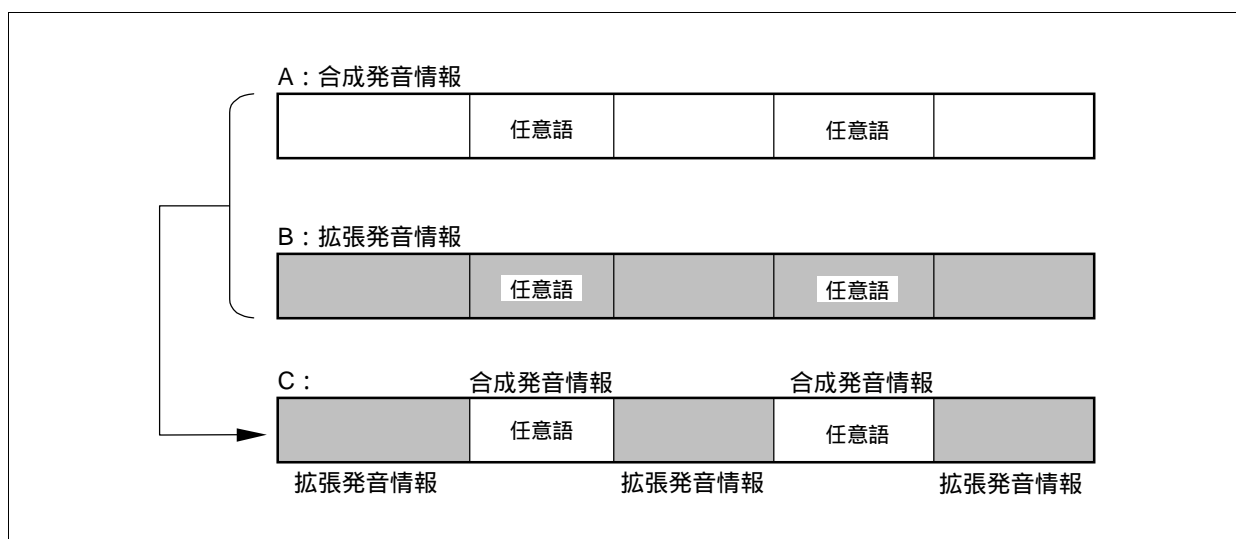
スピーチ・デザイナーとは、あらかじめデザインした情報（拡張発音情報）を通常音声合成中に作成されるものと入れ換えることにより、より自然な音声をつくり出す機能です。

4.2 動作概要

音声生成部は、拡張発音情報がパラメータで指定されていると、PhoneData で指定された任意語（入れ換え単語）と拡張発音情報の文章から通常の合成発音情報（図 4 - 1 の「A:」）を作ります。また、拡張発音情報には NEC によって作成された合成発音情報と同じフォーマットの情報（図 4 - 1 の「B:」）が含まれています。

音声生成部は、この拡張発音情報の任意語部分と合成発音情報の任意語部分を入れ換えたもの（図 4 - 1 の「C:」）を作成しこれを処理対象の発音情報とします。

図4 - 1 動作概要



4.3 仕 様

拡張発音情報を使用する場合、PhoneData には入れ換える任意語の発音記号を次のような形で記述します。（文章の発音記号は拡張発音情報中に含まれているので任意語のみの記述にしてください。）

（発音記号 1）（発音記号 2） ... NULL

また、拡張発音情報中の任意語をそのまま使用するときには次のような指定をしてください。

- ・ 任意語の入れ換える必要がない場所は（ ）の中には何も入れない。
- ・ 任意語の入れ換えをしない場合は NULL を入れる。
- ・ NULL の前にくる（ ）は省略可能。

表4-1に、oyoso;(ichikirome-toru)sa"ki;(shiya"k_usho)des_u.という文の拡張発音情報に対するの任意語入れ替えの例をいくつか示します。

表4-1 拡張発音情報に対するの任意語入れ替え例

PhoneDataに入れる発音記号	音声生成対象発音記号
(i"tteN/nanakirome-toru) (tocho) NULL	oyoso;(i"tteN/nanakirome-toru)sa"ki;(tocho)des_u.
(イ) ッテンナナキロメートル (トチョー) NULL	oyoso;(i"tteN/nanakirome-toru)sa"ki;(tocho)des_u.
() (tocho) NULL	oyoso;(ichikirome-toru)sa"ki;(tocho)des_u.
(i"tteN/nanakirome-toru) () NULL	oyoso;(i"tteN/nanakirome-toru)sa"ki;(shiya"k_usho)des_u.
NULL	oyoso;(ichikirome-toru)sa"ki;(shiya"k_usho)des_u.

拡張発音情報中には合成音のスピード、ピッチ、アクセントの情報も含まれていますが、指定する任意語によっては自然につながらないものがあります。その場合は任意語の発音記号の中(かっこの中)に表4-2に示す制御記号を埋め込み、手動で指定してください。

表4-2 制御記号

ピッチ	L1, ... , L9	やや低く ~ 低く
	H1, ... , H9	やや高く ~ 高く
速度	S1, ... , S9	やや遅く ~ 遅く
	F1, ... , F9	やや速く ~ 速く
アクセント	W1, ... , W9	やや弱く ~ 弱く
	A1, ... , A9	やや強く ~ 強く

注意 制御記号には半角を使用してください。

例： (F2H4A1tocho)

表4-2の値は、拡張発音情報中の合成音のスピード、ピッチ、アクセント情報からの相対値となります。また、表4-2の制御記号は1つの拡張発音記号でそれぞれ1度の使用に限られています。したがって任意語が2つ以上ある場合、任意語ごとにピッチなどを変えるということではできません。(1つの拡張発音記号で2度以上制御記号を使用した場合、最後に指定した制御記号が有効になります。)

備考 拡張発音情報は11 kHz サンプリングの音声のみサポートします。

第 5 章 ビープ音生成機能

5.1 概 要

TTS の音声生成部 (tts_SyntheWave()関数) にビープ音生成用の記号列を入力することにより、ビープ音を生成できます。

5.2 仕 様

tts_SyntheWave()の引き数 PhoneData にビープ音記号列の先頭アドレスを指定します。ビープ音記号列を NULL まで読み込み、ビープ音生成処理を行います。ビープ音記号列は次のように記述します。

- ・ビープ音記号列の先頭には、「%」(0x25)をつけてください。
- ・ビープ音記号列の終端には NULL をつけてください。
- ・ビープ音記号列の長さの上限はありません。
- ・ビープ音記号は、発音記号と混在させて使用できません。

```
%(ビープ音記号 1) (ビープ音記号 2) ... NULL
```

(1) ビープ音記号について

ビープ音記号は、音の種類と音の時間を組み合わせて作成します。音の種類は、ド (262 Hz) ~ シ (1975 Hz) まで 3 オクターブの音を半音刻みで指定可能です。音の長さは、1 K (1024) サンプル (0.09 sec [11 kHz] , 0.13 sec [8 kHz]) を 1 単位として、0-99 まで指定可能です。

また、ビープ音記号列中にポーズが挿入可能です。ポーズ長は、1 K (1024) サンプルを 1 単位として、0-99 まで指定可能です。

アルファベット、数字は半角を使用してください。

表5 - 1 ビープ音の種類

ド	レ	ミ	ファ	ソ	ラ	シ	ド#	レ#	ファ#	ソ#	ラ#
C1 (262)	D1 (294)	E1 (330)	F1 (349)	G1 (392)	A1 (440)	B1 (494)	c1 (277)	d1 (311)	f1 (367)	g1 (415)	a1 (466)
C2 (523)	D2 (587)	E2 (659)	F2 (698)	G2 (784)	A2 (880)	B2 (988)	c2 (554)	d2 (622)	f2 (740)	g2 (831)	a2 (932)
C3 (1047)	D3 (1175)	E3 (1318)	F3 (1396)	G3 (1567)	A3 (1760)	B3 (1975)	c3 (1109)	d3 (1244)	f3 (1479)	g3 (1661)	a3 (1864)
ポーズ Pmn (mn は 00-99)											

備考 () は周波数 [Hz] を示します。

(2) ビープ音の長さ Tmn (mn は 00-99)

例

- ・ドミソと1Kサンプルずつ発音する場合 %C2T01E2T01G2T01
- ・上記の例でミとソの間にポーズを2Kサンプル分挿入する場合 %C2T01E2T01P02G2T01

(3) 注意事項

- ・ビープ音生成中は、入力のビープ音記号列を変更しないでください。
- ・PhoneInfo には、NULL を指定してください。
- ・ビープ音生成機能を使用する場合、tts_SyntheWave()の次の引き数は使用しません。未使用の引き数は、設定された値を無視します。

TTSVOIMOD のメンバ

short Pitch; /* 声の高さ指定 (入力) */ (未使用)

short speed; /* 読み上げ速度指定 (入力) */ (未使用)

short Accent; /* アクセントの強弱指定 (入力) */ (未使用)

- ・音素領域の先頭アドレスを示す long *Character (TTSVOIMOD のメンバ) はサンプリング周波数の判定に使用します。正しい音素のアドレスを設定してください。
- ・上記以外の引き数は、発音記号から音声データを生成する場合と同じです。2.4.3 (5) **tts_SyntheWave 関数**を参照してください。
- ・ビープ音生成処理中の内部動作は、発音記号列から音声データを生成する場合と同じです。2.4.3 (5) **tts_SyntheWave 関数**を参照してください。

第 6 章 インストール

この章では、TTS の提供媒体からファイルを展開して、サンプル・プログラムを作成する手順を示します。

6.1 提供形態

TTS では、NEC 製または GHS 社製ツールを使用してアプリケーションを開発するためのライブラリを 2 種類（NEC 製ツール用と GHS 社製ツール用）提供しています。

提供形態について図 6 - 1 に示します。

図6-1 TTSの提供形態(1/2)

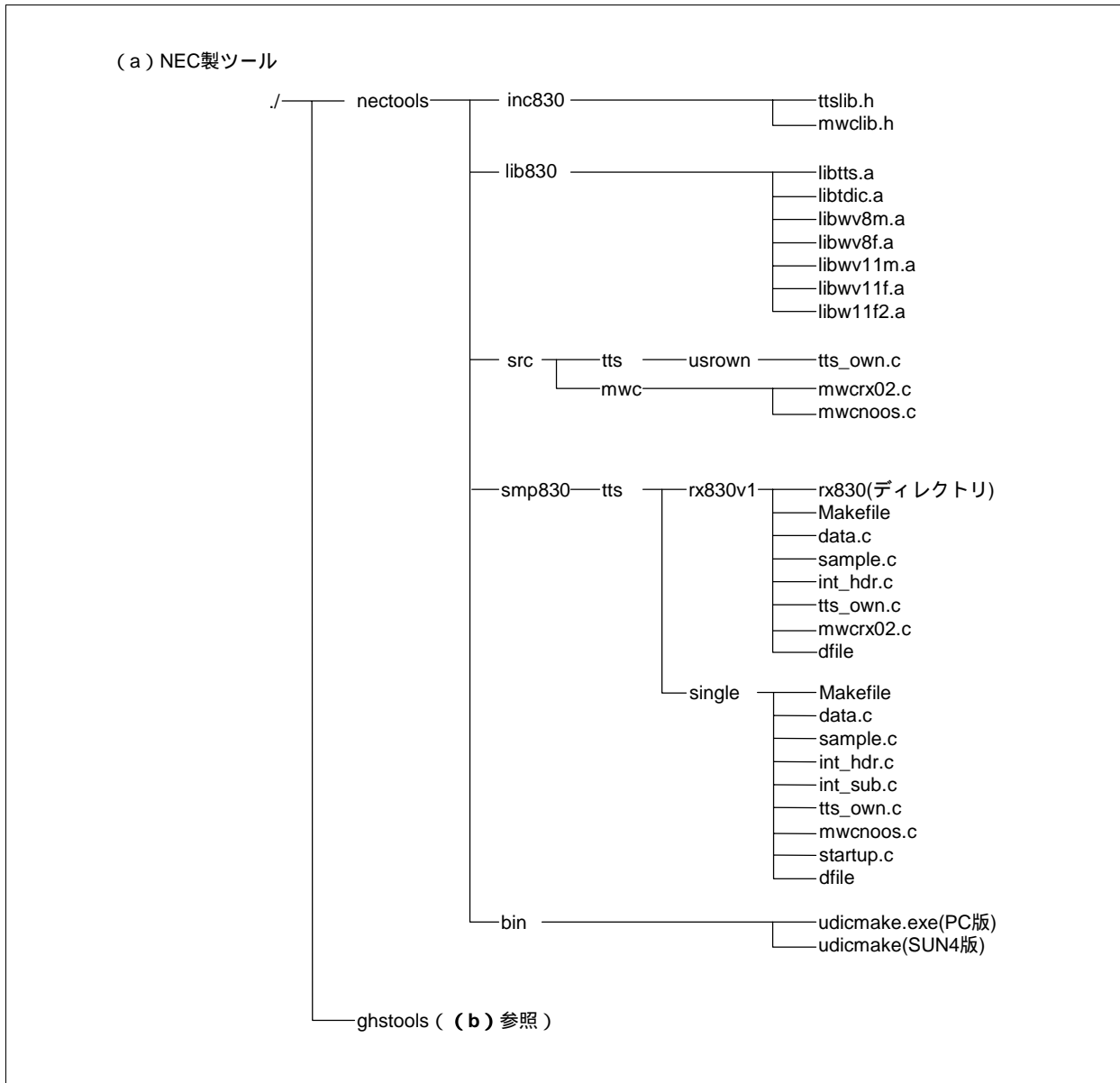
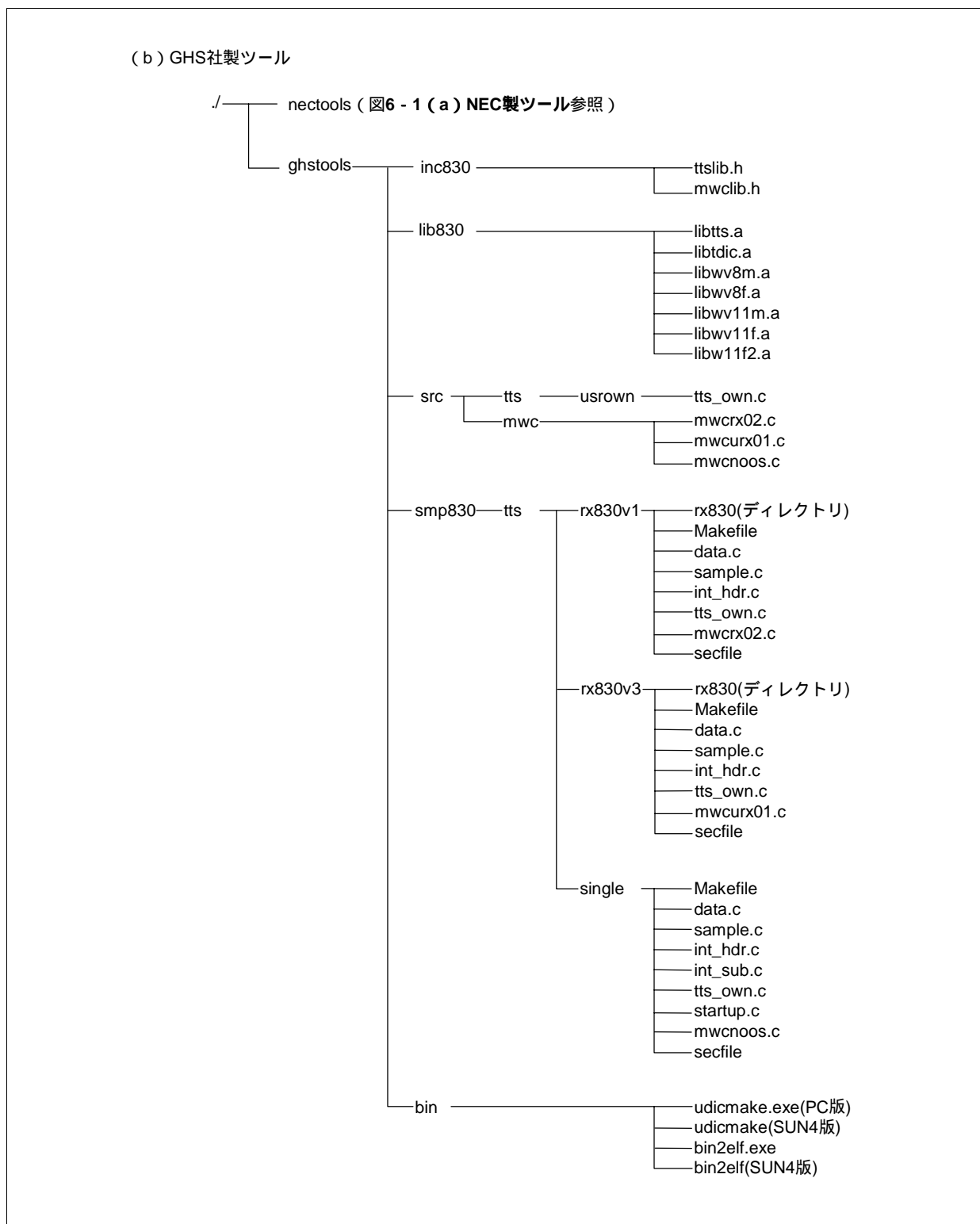


図6-1 TTSの提供形態(2/2)



6.1.1 ディレクトリとファイル

各ディレクトリおよびファイルについて説明します。

(1) nectools

NEC 製ツールを使用して、アプリケーションを開発する場合の TTS プログラムを格納しています。

(2) ghstools

GHS 社製ツールを使用して、アプリケーションを開発する場合の TTS プログラムを格納しています。

(3) inc830

TTS で使用する構造体やエラー・コードを定義しているヘッダ・ファイルを格納しています。

(4) lib830

TTS のライブラリを格納しています。

libtts.a

(テキスト解析部ライブラリ, 音声生成部ライブラリ, ユーザ辞書管理部ライブラリ)

libtdic.a

(テキスト解析部(システム辞書)ライブラリ)

libwv8m.a

(音声生成部(8 kHz サンプリング男声波形データ)ライブラリ)

libwv8f.a

(音声生成部(8 kHz サンプリング女声波形データ)ライブラリ)

libwv11m.a

(音声生成部(11 kHz サンプリング男声波形データ)ライブラリ)

libwv11f.a

(音声生成部(11 kHz サンプリング女声 1 波形データ)ライブラリ)

libwv11f2.a

(音声生成部(11 kHz サンプリング女声 2 波形データ)ライブラリ)

(5) src/mwc

ソース・プログラムの形で提供される関数のソース・ファイルを格納しています。

mwcrx02.c

(RX830 (ITRON1) を使用する場合の OS インタフェース関数ソース・ファイル)

mwcux01.c

(RX830 (μITRON Ver.3.0) を使用する場合の OS インタフェース関数ソース・ファイル)

mwcnoos.c

(OS を使用しない場合の OS インタフェース関数ソース・ファイル)

(6) src/tts/usrown

ソース・プログラムの形で提供される関数のソース・ファイルが格納されています。

tts_own.c

(TTS が呼び出すユーザ定義関数のソース・ファイル)

(7) smp830/tts

TTS を使用したサンプル・プログラム、サンプル・データおよびサンプルの音声出力制御ルーチンを格納しています。ハードウェア依存部分は、RTE-V831-PC (マイダス・ラボ製) で動作するように作成されています。

(8) smp830/tts/single

OS を使用しない場合のサンプル・プログラムを格納しています。

(9) smp830/tts/rx830v1, smp830/tts/rx830v3

OS およびシステムの初期化のサンプル・ファイルを格納しています。

- ・ rx830v1 : RX830 (ITRON1) を使用する場合
- ・ rx830v3 : RX830 (μITRON Ver.3.0) を使用する場合

(10) bin

ユーザ辞書作成ツールを格納しています。

バイナリ ELF ファイル変換プログラムは、ghstools だけ格納しています。バイナリ ELF ファイル変換プログラム (bin2elf) の使用方法について次に説明します。

(a) バイナリ ELF ファイル変換プログラムの操作

bin2elf(.exe)は、バイナリ・ファイルから GHS 社製ツールのリンクに入力できる ELF ファイルを生成するプログラムです。

(i) プログラムの起動方法

プログラム bin2elf(.exe)をオプションなしで起動すると、簡単な操作説明を表示します。

例

```
C:¥>bin2elf
Binary file to ELF file converter Version 1.00 [JUN 6 1997]
Copyright(C)NEC Corporation 1997

usage:bin2elf -s<section> -l<label> -o<outfile> <infile>
```

(ii) オプション

プログラム実行時に指定できるオプションは次に示す 3 種類があります。

-s : セクション名を指定します。(デフォルト=.data)
 -l : ラベル名を指定します(デフォルト=_dummy)
 -o : 出力ファイル名を指定します(デフォルト=a.o)
 オプションを指定しないとデフォルトの名前になります。

例 辞書ファイル (sample.dic) から sample.o を生成します。セクションは.jisho , ラベルは _chimei とします。

```
C:¥>bin2elf -s .jisho -l _chimei -o sample.o sample.dic
```

6.2 ホスト・マシンへのファイル展開

提供媒体からホスト・マシン上にファイル群を転送する手順を, UNIX 版 (SUN4) と Windows 版に分けて説明します。

6.2.1 UNIX 版 (CGMT)

UNIX 版の提供媒体は, CGMT だけです (フロッピー・ディスクでは提供していません)。

媒体には, tar 形式で TTS のファイル群を格納しています。ホスト・マシンへのインストールの手順を次に示します。

TTS をインストールするためのディレクトリを作成します。ここでは tts830 という名前のディレクトリを作成します。

```
% mkdir tts830 [ret]
```

作成したディレクトリに移動します。

```
% cd tts830 [ret]
```

提供媒体を磁気テープ装置にセットします。

tar コマンドを実行して、ファイル群をディスク上に展開します。なお、ホスト・マシンにより指定するスペシャル・ファイル名は異なります。ここでは/dev/rst8 であるとして、NEC 製と GHS 社製の実行方法をそれぞれ次に示します。

NEC 製ツール用

```
% tar -xvof /dev/rst8 nectools [ret]
```

GHS 社製ツール用

```
% tar -xvof /dev/rst8 ghstools [ret]
```

ファイルがインストールされたことを確認します。各ディレクトリについては **6.1 提供形態** を参照してください。

```
% ls -CFR [ret]
```

6.2.2 Windows 版

Windows 版の提供媒体は、CD-ROM とフロッピー・ディスク (3.5 インチ) で提供しています。ホスト・マシンへのインストールの手順を次に示します。

(1) CD-ROM

CD-ROM には、図 6 - 1 に示した提供形態と同じ構成を格納しています。インストール先のディレクトリを作成して、NEC 製ツールを使用する場合は nectools ディレクトリを、GHS 社製ツールを使用する場合は ghstools ディレクトリをそのままコピーしてください。

(2) フロッピー・ディスク

TTS をインストールするためのディレクトリを作成します。ここでは A ドライブに tts830 という名前のディレクトリを作成します。

```
A> md tts830 [ret]
```

作成したディレクトリに移動します。

```
A> cd tts830 [ret]
```

NEC 製ツール用を使用する場合は DISK#1-DISK#4 を、GHS 社製ツール用を使用する場合は DISK#5-DISK#8 を使用します。ここではフロッピー・ディスク・ドライブは C ドライブとしています。次に NEC 製ツール用と GHS 社製ツール用の実行方法を示します。

NEC 製ツール用

```
DISK#1 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥NECDISK1[ret]  
DISK#2 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥NECDISK2[ret]  
DISK#3 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥NECDISK3[ret]  
DISK#4 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥NECDISK4[ret]
```

GHS 社製ツール用

```
DISK#5 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥GHSDISK1[ret]  
DISK#6 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥GHSDISK2[ret]  
DISK#7 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥GHSDISK3[ret]  
DISK#8 をフロッピー・ディスク・ドライブにセットして、次のコマンドを実行します。  
A> c:¥GHSDISK4[ret]
```

ファイルがインストールされたことを確認します。各ディレクトリについては **6.1 提供形態**を参照してください。

NEC 製ツール用

```
A> dir a:¥nectools [ret]
```

GHS 社製ツール用

```
A> dir a:¥ghstools [ret]
```

6.3 サンプル・プログラムの作成

smp830 ディレクトリには、音声出力制御部のサンプル・プログラム、TTS を使用した簡単なアプリケーション・プログラムと発声させる日本語かな漢字テキストを、C 言語の配列に変換したものが格納されています。

サンプル・プログラムの内容は、data.c のファイルに格納されたテキスト（C 言語の配列の形で格納）から音声を生成して、オーディオ・デバイスに出力するものです。

なお、サンプルとして提供している音声出力制御部（int_hdr.c、int_sub.c）と初期化処理部は、マイダス・ラボ製 V831 評価ボード（RTE-V831-PC）を対象として作成されています。リアルタイム OS の呼び出しルーチン（mwcrx02.c）およびリアルタイム OS の初期化（rx830/*）は、NEC 製 RX830 を対象として作成されています。ほかのターゲット・ボードやリアルタイム OS 上で動作させる場合は、これらのファイルを修正して使用してください。

次に、サンプル・プログラムの make について例を示します（NEC 製ツールのライブラリと、OS に RX830 (ITRON1) を使用して、UNIX 版を使用した場合）。

サンプル・プログラムを格納しているディレクトリに移動します。

```
% cd tts830/nectools/smp830/tts/rx830v1 [ret]
```

エディタを使用して、Makefile、int_sub.c、int_hdr.c、tts_own.c、mwcrx01.c、rx830/* をターゲットにあわせて変更します。Makefile の TOOLS にはコンパイラのあるパスを、OSTOOLS にはリアルタイム OS のあるパスを指定します。

make コマンドを実行します。

```
% make [ret]
```

tts.elf が生成されたことを確認します。

```
% ls -l *.elf [ret]
```

インサーキット・エミュレータなどを使用して、ターゲットにダウンロードします。

ダウンロード後、PC (プログラム・カウンタ) を「start」(または「_boot」) に設定します (ブレークポイントは必要に応じていくつか設定してください)。この設定を終了させてからプログラムを実行してください。

6.4 ロケーションの変更

TTS は、次に示すセクション名が付けられています。ユーザのターゲットにあわせて、ロケーションを変更できます。

表6-1 セクション名

セクション名	内 容
.TTSTEXT	TTS プログラム
.TTSWAV	TTS 波形データ ROM
.TTSDIC	TTS 辞書データ ROM
.TTSROM	TTS データ ROM (辞書, 波形データ以外)

V830 の jal 命令で指定できるアドレスが 26 ビット・ディスプレイメントのため、ミドルウェアとミドルウェアを直接呼び出すプログラムは、26 ビット以上離さないください。

表 6-1 で示したセクションは、必ず 4 バイト align された場所に配置してください。

6.5 シンボル名規約

TTS ライブラリ内のグローバル・シンボルには、すべて先頭に “tts” または “mwc”（アセンブラでは “_tts” または “_mwc”）が付いています。ユーザ・アプリケーションの中のシンボル名と重複しないようにしてください。

6.6 ユーザ・OWN・コーディング関数

サンプル・プログラムの中の `tts_own.c`, `mwcrx02.c`, `mwcurx01.c`, `mwcnos.c` には、ユーザ・OWN・コーディング関数が入っています。必要に応じて書き換えてください。`mwcrx02.c`, `mwcurx01.c`, `mwcnos.c` は TTS から呼び出される関数を定義しています。どれか1つを必ずリンクしてください。

使用する OS によってファイルが異なるので注意してください。

RX830 (ITRON1) を使用	: mwcrx02.c
RX830 (μ ITRON Ver.3.0) を使用	: mwcurx01.c
上記以外の OS を使用	: mwcnos.c (ユーザにより定義)
OS を使用しない	: mwcnos.c

付録 A 発音記号インタフェース

音声生成部単体で動作するときに必要な発音記号（ローマ字表記発音記号，カナ表記発音記号）のインタフェースについて説明します。

TTS では，音声生成部だけを動作させることもできます。

A.1 ローマ字表記発音記号とかな文字の対応

発音記号とかな文字の対応を表 A - 1 に，アクセントなどを表す補助発音記号を **A.3 補助発音記号** に示します。これらの発音記号と補助発音記号を組み合わせて発音記号列を作成します。

発音記号と発音記号の間にポーズ記号（@mn）を挿入することにより，ポーズ（無音区間）が挿入可能です。ポーズ時間は mn（00-99）で指定し，1 単位は 10 ms です。10 ms-990 ms まで指定可能です。

例) 30 ms のポーズを挿入する場合 koNnichiwa@03sayo-nara

表 A - 1 ローマ字表記発音記号 かな文字の対応 (1/2)

表記	発声	表記	発声	表記	発声	表記	発声	表記	発声
a	あ	i	い	u	う	e	え	o	お
ka	か	ki	き	ku	く	ke	け	ko	こ
		ki-	きい			kye	きえ		
kya	きゃ			kyu	きゅ			kyo	きょ
ga	が	gi	ぎ	gu	ぐ	ge	げ	go	ご
		gi-	ぎい			gye	ぎえ		
gya	ぎゃ			gyu	ぎゅ			gyo	ぎょ
Ga	か°	Gi	き°	Gu	く°	Ge	げ°	Go	ご°
						Gye	ぎえ		
Gya	き°ゃ			Gyu	き°ゅ			Gyo	き°ょ
kwa	くあ	kwi	くい			kwe	くえ	kwo	くお
kwa	くわ								
gwa	ぐあ	gwi	ぐい			gwe	ぐえ	gwo	ぐお
gwa	ぐわ								
sa	さ	shi	し	su	す	se	せ	so	そ
		si	すい			se	すえ		
za	ざ	ji	じ	zu	ず	ze	ぜ	zo	ぞ
		zi	ずい			ze	ずえ		
sha	しゃ	shi-	しい	shu	しゅ	she	しえ	sho	しょ
ja	じゃ	ji-	じい	ju	じゅ	je	じえ	jo	じょ
ta	た	chi	ち	tsu	つ	te	て	to	と
		ti	てい	tu	とう	tye	てえ		
tya	てゃ			tyu	てゅ			tyo	てょ
da	だ	ji	ぢ	zu	づ	de	で	do	ど

表A - 1 ローマ字表記発音記号 かな文字の対応 (2/2)

表記	発声	表記	発声	表記	発声	表記	発声	表記	発声
		di	でい	du	どう	dye	でえ		
dya	でや			dyu	でゆ			dyo	でよ
cha	ちゃ	chi-	ちい	chu	ちゆ	che	ちえ	cho	ちょ
ja	じゃ	ji-	じい	ju	じゆ	je	じえ	jo	じょ
tsha	つあ	tsi	つい			tse	つえ	tso	つお
na	な	ni	に	nu	ぬ	ne	ね	no	の
		ni-	にい			nye	にえ		
nya	にや			nyu	にゆ			nyo	にょ
ha	は	hi	ひ	hu	ふ	he	へ	ho	ほ
		hi-	ひい			hye	ひえ		
hya	ひや			hyu	ひゆ			hyo	ひょ
ba	ば	bi	び	bu	ぶ	be	べ	bo	ぼ
		bi-	びい			bye	びえ		
bya	びや			byu	びゆ			byo	びょ
pa	ぱ	pi	ぴ	pu	ぷ	pe	ぺ	po	ぽ
		pi-	ぴい			pye	ぴえ		
pya	ぴや			pyu	ぴゆ			pyo	ぴょ
fa	ふあ	fi	ふい			fe	ふえ	fo	ふお
fya	ふや			fyu	ふゆ			fyo	ふょ
ma	ま	mi	み	mu	む	me	め	mo	も
						mye	みえ		
mya	みや			myu	みゆ			myo	みょ
ya	や			yu	ゆ			yo	よ
		ye	いえ						
ra	ら	ri	り	ru	る	re	れ	ro	ろ
						rye	りえ		
rya	りや			ryu	りゆ			ryo	りょ
wa	わ							o	を
wi	うい					we	うえ	wo	うお
va	ヴァ	vi	ヴィ	vu	ヴ	ve	ヴェ	vo	ヴォ
va	ヴァ	vi	ヴィ	vu	ヴ	ve	ヴェ	vo	ヴォ
vya	ヴァ			vyu	ヴユ			vyo	ヴヨ
vya	ヴァ			vyu	ヴユ			vyo	ヴヨ
N	ん								
wa	(私)は		子音を重ねる	っ		e	(山)へ		
ポーズ	@mn (mn は 00 ~ 99)		時間は mn × 10 (ms)						

A.2 カナ表記発音記号とローマ字表記発音記号の対応

音声生成部では、カナ表記の発音記号列が入力された場合、ローマ字表記発音記号内部で変換し、音声を生成します。カナ表記発音記号とローマ字表記発音記号の対応を表 A - 2 に示します。

表A - 2 カナ表記発音記号 ローマ字表記発音記号の対応 (1/2)

ア	a	イ	i	ウ	u	エ	e	オ	o
カ	ka	キ	ki	ク	ku	ケ	ke	コ	ko
		キィ	ki-			キェ	kye		
キャ	kya			キュ	kyu			キョ	kyo
ガ	ga	ギ	gi	グ	gu	ゲ	ge	ゴ	go
		ギィ	gi-			ギェ	gye		
ギャ	gya			ギユ	gyu			ギョ	gyo
クァ	kwa	クィ	kwi			クェ	kwe	クォ	kwo
クゥ	kwa								
グァ	gwa	グィ	gwi			グェ	gwe	グォ	gwo
グゥ	gwa								
サ	sa	シ	shi	ス	su	セ	se	ソ	so
		スイ	si			スェ	se		
ザ	za	ジ	ji	ズ	zu	ゼ	ze	ゾ	zo
		ズィ	zi			ズェ	ze		
シャ	sha	シィ	shi-	シュ	shu	シェ	she	ショ	sho
ジャ	ja	ジィ	ji-	ジュ	ju	ジェ	je	ジョ	jo
タ	ta	チ	chi	ツ	tsu	テ	te	ト	to
		ティ	ti	トゥ	tu	テェ	tye		
テャ	tya			テュ	tyu			テョ	tyo
ダ	da	ヂ	ji	ヅ	zu	デ	de	ド	do
		ヂィ	di	ドウ	du	デェ	dye		
デャ	dya			デュ	dyu			デョ	dyo
チャ	cha	チィ	chi-	チュ	chu	チェ	che	チョ	cho
チャ	ja	ヂィ	ji-	ヂュ	ju	ヂェ	je	ヂョ	jo
ツァ	tsa	ツイ	tsi			ツェ	tse	ツォ	tso
ナ	na	ニ	ni	ヌ	nu	ネ	ne	ノ	no
		ニィ	ni-			ニェ	nye		
ニャ	nya			ニュ	nyu			ニョ	nyo
ハ	ha	ヒ	hi	フ	hu	ヘ	he	ホ	ho
		ヒィ	hi-			ヒェ	hye		
ヒャ	hya			ヒュ	hyu			ヒョ	hyo
バ	ba	ビ	bi	ブ	bu	ベ	be	ボ	bo
		ビィ	bi-			ビェ	bye		
ビャ	bya			ビュ	byu			ビョ	byo

表A - 2 カナ表記発音記号 ローマ字表記発音記号の対応 (2/2)

パ	pa	ピ	pi	プ	pu	ペ	pe	ポ	po
		ピィ	pi-			ピェ	pye		
ピャ	pya			ピュ	pyu			ピョ	pyo
ファ	fa	フィ	fi			フェ	fe	フォ	fo
フャ	fya			フュ	fyu			フョ	fyo
マ	ma	ミ	mi	ム	mu	メ	me	モ	mo
						ミェ	mye		
ミャ	mya			ミュ	myu			ミョ	myo
ヤ	ya			ユ	yu			ヨ	yo
		イエ	ye						
ラ	ra	リ	ri	ル	ru	レ	re	ロ	ro
						リエ	rye		
リャ	rya			リュ	ryu			リョ	ryo
ワ	wa							ヲ	o
		ウィ	wi			ウエ	we	ウォ	wo
ヴァ	va	ヴィ	vi	ヴ	vu	ヴェ	ve	ヴォ	vo
ヴァ	vya			ヴュ	vyu			ヴョ	vyo
ン	N								
				ッ	子音を重ねる				
-									

A.3 補助発音記号

表A-3 補助発音記号

補助発音記号	発音	例
- (マイナス)	直前の母音を長母音化	格子 ko-shi (cf. 子牛 koushi)
_ (アンダスコア)	直後の母音を無声化	明日 ash_ita
" (ダブルクォーテーション)	直前の音節に対するアクセント核	日本 niho"N
」 (全角)		
/	アクセント句境界 (ポーズなし, 強結合)	
//	アクセント句境界 (ポーズなし, 弱結合)	兄の黒いかばん a"nino//kuro"i/kabaN
・ (全角)		
, (カンマ)	アクセント句境界 (短ポーズ)	
。 (全角)		
;(セミコロン)	アクセント句境界 (中ポーズ)	
:(コロン)	アクセント句境界 (長ポーズ)	
.(ピリオド)	文末 (平叙文)	
?	文末 (疑問文)	
??	文末 (強い疑問文)	
!	文末 (命令文)	
!!	文末 (強い命令文)	
@	可変長ポーズ (無音期間) (@mn : mn = 00-99 ポーズ時間は mn × 10 ms)	

[メ モ]

付録 B 補足資料

B.1 テキスト解析処理部

- ・解析処理部に入力するコードは次の範囲を使用してください。

< 解析可能コード範囲 >

- 1 バイト・コード : 0x20-0x7e, 0xa1-0xdf
- 2 バイト・コード : 第 1 バイト 0x81-0x9f, 0xe0-0xfc
: 第 2 バイト 0x40-0x7e, 0x80-0xfc

上記以外のコード（解析不能コードと呼びます）が入力テキスト文字列中にあった場合、そのコードの発音記号は出力されません。その場合、解析不能コードのあるところが、意味の切れ目になります。

例) 東京 to-kyo- (「東京」と解釈。)
東 (解析不能コード) 京 higashikyo"- (「東」と「京」と解釈。)

詳細は B.4.1 TTS SJIS コード表を参照してください。

注意 解析可能コードは、システム辞書に登録されているコードを示すものではありません。B.4.1 TTS SJIS コード表で空欄になっているコードは、システム辞書に登録されていません。したがって、発音記号は出力されません。

- ・ユーザ定義文字は、解析可能コードの範囲内で定義してください。
ユーザ定義文字を割り当てる場合は、解析可能コードの範囲内でコードを割り当ててください。
- ・文字列が制御コードのみの場合、エラーが返ります。
解析処理部に入力するテキスト文字列（コード列）が制御コードのみで構成されている場合、テキスト解析処理部はエラー（E_DATA）を返します。
制御コード : 0x00-0x1a, 0x1c-0x20
- ・文字列がスペースのみの場合、エラーが返ります。
解析処理部に入力するテキスト文字列（コード列）がスペース（半角 0x20, 全角 0x8140）のみで構成されている場合、テキスト解析部はエラー（E_DATA）を返します。
- ・文字列の先頭が 0x00 の場合、エラーが返ります。
入力テキストの先頭が 0x00 の場合、そのあとに続く文字列にかかわらず、テキスト解析処理はエラー（E_DATA）を返します。

- ・ 「。(0x8142)」「?(0x8148, 0x3f)」「!(0x8149, 0x21)」は文末を示します。
記号を読まないモードで、入力文字列中に「。(0x8142)」「?(0x8149)」「!(0x8148)」がある場合、文末を表す補助発音記号「.」（平叙文）「?」（疑問文）「!」（命令文）に変換されます。
「。」「?」「!」が複数個連続している場合は、1個のみ有効になり、2個目以降は無視されます。
記号読むモードでは、それぞれ「くてん」「ぎもんぷ」「かんとんぷ」と記号の回数分だけ読まれます。

- ・ 長音記号は「ー(0x815b, 0xb0)」を使用してください。
長音記号に「-」を使用すると、「まいなす」と読まれます。

例) ハーイ ha"-i. (0x815b を使用。長音を表す。)
 ハ - イ ha//mainasu//i". (0x817c を使用。「まいなす」と読まれる。)
 ハーイ ha"-i. (0xb0 を使用。長音を表す。)
 ハ-イ hamaina"sui. (0x2d を使用。「まいなす」と読まれる。)

- ・ 1音の文字でひらがなとカタカナは混在させないでください。
大文字と小文字の2つの文字で構成される文字で、ひらがなとカタカナを混在して使用すると、正しく解析されません。

例) 「しゃ」 sha.
 「しゃ」 shi//ya. (「し」と「ャ」を別々に解釈)

- ・ 促音「っ」、長音「ー」が複数個連続しても、促音、長音の発声時間は変わりません。
入力文字列中に「っ」「ー」が複数個連続していても促音、長音の発声時間は個数に応じて変わりません。(解析処理より出力される発音記号列を音声生成部に入力すると、生成データ・サイズ、アクセントが異なります。詳細は B.2 音声生成処理部を参照してください。)

例) かつこ ka"kko.
 かつっっこ ka""Qkko. (「かつこ」を解析 合成した場合と同じ音声データを生成。)
 スペース s_upe"-su.
 スパーーース s_upe"--su. (「スペース」を解析 合成した場合とアクセントが異なる。)

- ・ 漢字の送り仮名はひらがなを使用してください。
カタカナは1つの意味のあるかたまりとして解釈するため、漢字の送り仮名にカタカナを使用すると、正しく解析されないことがあります。

例) 来る ku"ru. (送り仮名として解釈される。)
 来ル rai//ru. (送り仮名として解釈されない。)

- ・小文字「っ」「ッ」を単独で入力すると発音記号は出力されません。「わ」「ワ」を単独で入力すると、「あ」の読みになります。

表B - 1 小文字の単独発音

入力文字	コード	出力(発音記号)	読み	入力文字	コード	出力(発音記号)	読み
あ	0x829f	a.	あ	ア	0x8340	a.	あ
い	0x82a1	i.	い	イ	0x8342	i.	い
う	0x82a3	u.	う	ウ	0x8344	u.	う
え	0x82a5	e.	え	エ	0x8346	e.	え
お	0x82a7	o.	お	オ	0x8348	o.	お
っ	0x82c1	.	(発声しない)	ッ	0x8362	.	(発声しない)
や	0x82e1	ya.	や	ヤ	0x8383	ya.	や
ゆ	0x82e3	yu.	ゆ	ユ	0x8385	yu.	ゆ
よ	0x82e5	yo.	よ	ヨ	0x8387	yo.	よ
わ	0x82ec	a	あ	ワ	0x838e	a	あ
				カ	0x8395	ka.	か
				ケ	0x8396	ka.	か

- ・数字は、数字または漢数字のみで入力し、混在させないでください。

数字と漢数字を混在させると、正しく桁読みできません。

- 例) 1234 se"N/nihyak_usa"Nju-/yo"N.
 ー 23 四 ichi"/ni"ju-/saN/yo"N. (「1」,「23」,「4」と解釈。)
 1 二三四 ichi"/nihyak_usa"Nju-/yo"N. (「1」,「234」と解釈。)

- ・単位はひらがな/カタカナ/記号文字を使用してください。

単位は、ひらがな/カタカナ/記号文字を使用してください。アルファベット等を用いて記述すると、辞書に登録されていない場合、正しく解析できません。

- 例) 1.5 キロメートル先 i"tteN/gokirome-torusa"ki.
 1.5km 先 i"tteN/gokirome-torusa"ki. (「k」(0x6b) + 「m」(0x6d)を使用。)
 1.5KM 先 i"tteN/go"/ke-e"mu//saki. (「K」(0x4b) + 「M」(0x4d)を使用。)
 1.5 km先 i"tteN/gokirome-torusa"ki. (「km」(0x8771)を使用。)

注意「KM」はシステム辞書に登録されていないので、正しく読めません。

- ・撥音記号「°」をひらがなと組み合わせて使用しないでください。

撥音記号「°」はカタカナと組み合わせて使用してください。ひらがなと組み合わせた場合、正しく解析できません。

- 例) 「ハ°ン」 pa"N.
 「は°ん」 ha//N. (「は」と「°」が別々に解釈されている。)

- ・数字読み=ON の場合、読点「、」、カンマ「,」、ピリオド「.」の読みに注意してください。

数字読み=ON の場合、カンマ「,」、ピリオド「.」は数字として解釈されます。

数字読み=ON の場合、数字の直前に「.」がある場合、記号読みモードに関係なく「てん」と読み上げられます。

半角と全角は同じ解釈になります。

例 1：数字読み = ON，記号読み = ON の場合)

- | | | |
|--------|------------------------------|---------------------|
| 12、 34 | ju-ni":to"-teN:sa"Nju-/yo"N. | (「12」と「34」と解釈。) |
| 12, 34 | se"N/nihyak_usa"Nju-/yo"N. | (「1234」と解釈。) |
| 12. 34 | ju-ni"-teN/saNyo"N. | (「12.34」と解釈。) |
| 12、 | ju-ni":to"-teN:. | (「、」を「とうてん」と読み上げる。) |
| 12, | ju-ni". | (「,」は読み上げられない。) |
| 12. | ju-ni". | (「.」は読み上げられない。) |
| 、 34 | :to"-teN:sa"Nju-/yo"N. | (「、」を「とうてん」と読み上げる。) |
| , 34 | :ko"Nma:sa"Nju-/yo"N. | (「,」を「こんま」と読み上げる。) |
| . 34 | te"N/saNyo"N. | (「.」を「てん」と読み上げる。) |

例 2：数字読み = ON，記号読み = OFF の場合)

- | | | |
|--------|----------------------------|-------------------|
| 12、 34 | ju-ni";sa"Nju-/yo"N. | (「12」と「34」と解釈。) |
| 12, 34 | se"N/nihyak_usa"Nju-/yo"N. | (「1234」と解釈。) |
| 12. 34 | ju-ni"-teN/saNyo"N. | (「12.34」と解釈。) |
| 12、 | ju-ni";. | (「、」は読み上げられない。) |
| 12, | ju-ni". | (「,」は読み上げられない。) |
| 12. | ju-ni". | (「.」は読み上げられない。) |
| 、 34 | sa"Nju-/yo"N. | (「、」は読み上げられない。) |
| , 34 | sa"Nju-/yo"N. | (「,」は読み上げられない。) |
| . 34 | te"N/saNyo"N. | (「.」を「てん」と読み上げる。) |

B.2 音声生成処理部

- ・音声データを生成する記号は表 A - 1 から表 A - 3 に示してありますので参照してください。
 - 表 A - 1 は、半角小文字アルファベット (0x61-0x7a) , 半角「@」(0x40) , 半角数字 (0x30-0x39) を使用します。
 - 表 A - 2 は、SJIS 全角カタカナ (0x8340-0x8396) , 「ー」(0x815b) を使用しています。
 - 表 A - 3 は、半角「-」(0x2d) , 「_」(0x5f) , 「"」(0x22) , 「/」(0x2f) , 「,」(0x2c) , 「;」(0x3b) , 「:」(0x3a) , 「.」(0x2e) , 「?」(0x3f) , 「!」(0x21) を使用しています。
- ・入力する発音記号列の先頭が 0x00 の場合、エラー (E_DATA) が返ります。
- ・表 A - 1 から表 A - 3 に示す記号以外のコードが入力発音記号列中に入った場合、そのコードは無視されません。
- ・表 A - 1 , 表 A - 2 に示す組み合わせが 1 つもない場合 (入力記号がすべて無視された場合) , エラー (E_DATA) を返します。
- ・入力発音記号列が表 A - 3 に示す補助発音記号のみの場合、エラー (E_DATA) を返します。
- ・促音文字「ッ」は、ローマ字発音記号に変換したときに、次に続く文字の繰り返しになります。
- ・入力発音記号列の先頭が 1 つ以上の長音記号「-」の場合、エラー (E_DATA) を返します。
 - 例)** 「--kakko-」を入力した場合、E_DATA が返ります。
- ・入力発音記号列中に 2 個以上の長音記号「-」が連続していた場合、長音記号は 1 個のみ有効になります。
 - 例)** 「kakko---」を入力した場合、生成される音声データは「kakko-」を入力した場合と同じになります。
- ・子音が 3 個以上連続している場合、2 個のみ有効になり、促音を生成します。
 - 例)** 「kakkko-」は「kakko-」と同じ音声データを生成します。
- ・発音記号列中にアクセント記号「"」が複数個連続している場合、1 個のみ有効になり、2 個目以降は無視されます。
 - 例)** 「ho"kkaido-」と「ho""""kkaido-」は同じ発音になります。
- ・先頭が「.」(0x2e) の場合、エラーになります。
 - 入力発音記号列の先頭が「.」(0x2e) である場合、エラー (E_DATA) を返します。

B.3 ユーザ辞書作成時の注意事項

- ・記号文字を登録する場合は注意が必要です。

記号文字を「固有名詞」として登録する場合と「記号」として登録する場合とでは、解析結果が異なることがあります。

例) 「 」の読みを「まるいち」と登録し、文字列「 はみがきこ」をテキスト解析処理に入力する。

【「固有名詞」として登録した場合】

「まるいちわみがきこ」 (「は」を「わ」と読む。「 は、磨き粉」の意味になる。)

【「記号」として登録した場合】

「まるいち はみがきこ」 (「は」を「は」と読む。「 歯磨き粉」の意味になる。)

「 」を「固有名詞」として登録し、「まるいち はみがきこ」と読ませるためには、読みを「まるいち / 」としてください。「 / 」はアクセント区切りを表し、同時に意味の切れ目の意味も持ちます。

- ・複数のユーザ辞書を使用する場合は、注意が必要です。

複数のユーザ辞書を使用する場合、ユーザ辞書の検索順は、辞書情報構造体配列に登録した順になります。ユーザ辞書はシステム辞書よりも優先して検索されます。

ユーザ辞書検索時に、登録してある表記にヒットしてしまうと、その後のユーザ辞書は検索されません。複数のユーザ辞書に同じ表記を含む単語を登録する場合、注意が必要です。

例) ユーザ辞書 1, ユーザ辞書 2 に次の単語を登録します。

ユーザ辞書 1: 8 東京 かんとー

8 京都 かんさい

ユーザ辞書 2: 8 東京都 しゅと

辞書情報構造体配列は次のように設定します。

DicInfo[0]: システム辞書 (検索順 3)

DicInfo[1]: ユーザ辞書 1 (検索順 1)

DicInfo[2]: ユーザ辞書 2 (検索順 2)

”東京都”を解析すると、読みは「かんとーと」になります。(ユーザ辞書 1 に登録されている「東京」にヒットしてしまうため、ユーザ辞書 2 は検索されません。)

辞書情報構造体配列の設定を次のように変更します。

DicInfo[0]: システム辞書 (検索順 3)

DicInfo[1]: ユーザ辞書 2 (検索順 1)

DicInfo[2]: ユーザ辞書 1 (検索順 2)

”東京都”を解析すると”しゅと”となります。

B.4 記号コードの読みについて

記号コード（表 B-2 から表 B-5）の読み方について説明します。

コード : SJIS コードです。すべて 16 進表示です。

記号 : 各コードがあらわす記号を示しています。

読み : 記号をユーザ辞書に登録する場合の「読み」を示しています。

表 B-2 から表 B-5 のように「読み」を登録すると、表に示す「発音記号」を出力するユーザ辞書が作成されます。

発音記号 : TTS ミドルウェアの解析処理部に入力した場合に出力する発音記号です。

(1) 記号読みモード=1 (読む) の場合のみ、発音記号を出力するコード。

表 B-2 半角記号の読み (「記号読む」の場合のみ)

コード	記号	読み	発音記号
20	sp	すべ'ーす	s_upe"-su
21	!	か'んた'んふ	kaNta"Nhu
22	"	だ'ぶるこ'ーて'ーしょん	daburuko-te"-shoN
27	'	し'んぐるこ'ーて'ーしょん	shiNguruko-te"-shoN
28	(は'じめしょ'ーか'っこ	hajimesho-ka"kko
29)	お'わりしょ'ーか'っこ	owarisho-ka"kko
2c	,	こ'んま	ko"Nma
2e	.	ぴ'り'おど	piri"odo
2f	/	しゃ'せん	sha"seN
3a	:	こ'ろん	ko"roN
3b	;	せ'みこ'ろん	semiko"roN
3f	?	ぎ'も'んふ	gimo"Nhu
40	@	あ'つとま'ーく	attoma"-ku
5b	[は'じめだい'か'っこ	hajimedaika"kko
5c	¥	え'んま'ーく	eNma"-ku
5d]	お'わりだい'か'っこ	owaridaika"kko
5e	^	あ'くさんし'るこ'んふれ'くず	ak_usaNshirukoNhure"k_usu
5f	_	あ'んだ'ーら'いん	aNda-ra"iN
7b	{	か'っこ	ka"kko
7c		た'て'ぼ'ー	tate"bo-
7d	}	か'っこ	ka"kko
7e	~	ち'るだ	chi"ruda
a1	。	く'てん	k_uteN
a2	「	は'じめか'ぎ'か'っこ	hajimekagika"kko
a3	」	お'わりか'ぎ'か'っこ	owarikagika"kko
a4	、	と'ーてん	to"-teN
a5	・	ち'ゆ'ーてん	chu"-teN
de	°	だ'くてん	dak_uteN
df	°	は'んだ'く'てん	haNdaku"teN

表B-3 全角記号の読み（「記号読む」の場合のみ）（1/2）

コード	記号	読み	発音記号
8140	sp	すべ'ーす	s_upe"-su
8141	、	と'ーてん	to"-teN
8142	。	くてん	k_uteN
8143	、	こ'んま	ko"Nma
8144	・	ぴ'りおど	pi"riodo
8145	・	ちゅ'うてん	chu"uteN
8146	:	こ'ろん	ko"roN
8147	;	せみころ'ん	semiko"roN
8148	?	ぎも'んふ	gimo"Nhu
8149	!	かんだ'んふ	kaNta"Nhu
814a	°	だく'てん	daku"teN
814b	°	はんだく'てん	haNdaku"teN
814c	´	あくさ'んていぎゅう	ak_usa"Ntigyū
814d	`	あくさんぐら'ーぶ	ak_usaNgura"-bu
814e	¨	う'むらうと	u"murauto
814f	^	あくさんしるこんふれ'つくす	ak_usaNshirukoNhure"kkus_u
8150		おーばーら'いん	o-ba-ra"iN
8151	—	あんだーら'いん	aNda-ra"iN
8152	ゝ	くりかえし	kurikaeshi
8153	ゞ	だくてんくりかえし	dak_uteNkurikaeshi
8154	ゞ	くりかえし	kurikaeshi
8155	ゞ	だくてんくりかえし	dak_uteNkurikaeshi
8156	〃	くりかえし	kurikaeshi
8157	全	くりかえし	kurikaeshi
8158	々 ^注	くりかえし	kurikaeshi
815c		だ'っしゅ	da"sshu
815d		は'いふん	ha"ihuN
815e	/	しゃせん	shaseN
815f	\	ぎやくしゃ'せん	gyak_usha"seN
8160	~	なみだ'っしゅ	namida"sshu
8161		へいこー	heiko-
8162		たてせん	tateseN
8163	...	さんてんり'ーだ	saNteNri"-da
8164		にてんり'ーだ	niteNri"-da
8165	‘	ひだりいんよ'ーふ	hidariiNyo"-hu
8166	’	みぎいんよ'ーふ	migiiNyo"-hu
8167	“	ひだりにじゅーいんよ'ーふ	hidariniju-iNyo"-hu

注 「々」の直前に読むを持つ文字がある場合、「々」は直前の読みの繰り返しになります。

例)「村々」を解析した場合、読みは「むらむら」になります。

表B - 3 全角記号の読み（「記号読む」の場合のみ）（2/2）

コード	記号	読み	発音記号
8168	”	みぎにじゅーいんよ’ーふ	miginiju-iNyo"-hu
8169	（	はじめしょーか’っこ	hajimesho-ka"kko
816a	）	おわりしょーか’っこ	owarisho-ka"kko
816b	〔	はじめきっこーか’っこ	hajimek_ikko-ka"kko
816c	〕	おわりきっこーか’っこ	owarik_ikko-ka"kko
816d	[はじめだいか’っこ	hajimedaika"kko
816e]	おわりだいか’っこ	owaridaika"kko
816f	{	はじめちゅーか’っこ	hajimechu-ka"kko
8170	}	おわりちゅーか’っこ	owarichu-ka"kko
8171		はじめやまか’っこ	hajimeyamaka"kko
8172		おわりやまか’っこ	owariyamaka"kko
8173	《	はじめにじゅーか’っこ	hajimeniju-ka"kko
8174	》	おわりにじゅーか’っこ	owariniju-ka"kko
8175	「	はじめかぎか’っこ	hajimekagika"kko
8176	」	おわりかぎか’っこ	owarikagika"kko
8177	『	はじめにじゅーかぎか’っこ	hajimeniju-kagika"kko
8178	』	おわりにじゅーかぎか’っこ	owariniju-kagika"kko
8179	【	はじめすみつきか’っこ	hajimesumits_ukika"kko
817a	】	おわりすみつきか’っこ	owarisumits_ukika"kko
81ac	＝	げたま’ーく	getama"-ku
8780	”	き’ごー	ki"go-
8781	“	き’ごー	ki"go-

(2) 読みモードに関係なく常に発音記号を出力するコード。

表B - 4 半角記号の読み

コード	記号	読み	発音記号
23	#	い'げた	i"geta
24	\$	ど'る	do"ru
25	%	ぱ-せ'んと	pa-se"Nto
26	&	あんぱさ'んど	aNpasa"Ndo
2a	*	あすたり'すく	as_utari"s_uku
2b	+	ぷ'らす	pu"ras_u
2d	-	まいなす	mainas_u
3c	<	よりちいさ'い	yorichiisa"i
3d	=	いこ'ーる	iko"-ru
3e	>	よりおおき'い	yoriooki"i

表B - 5 全角記号の読み (1/10)

コード	記号	読み	発音記号
8159	ㄥ	しめ'	shime"
815a	○	ぜ'ろ	ze"ro
817b	+	ぷ'らす	pu"ras_u
817c	-	まいなす	mainas_u
817d	±	ぷらすま'いなす	purasuma"inas_u
817e	×	かけ'る	kake"ru
8180	÷	わる	waru
8181	=	いこ'ーる	iko"-ru
8182		ひとしくな'い	h_itosh_ikuna"i
8183	<	よりちいさ'い	yorichiisa"i
8184	>	よりおおき'い	yoriooki"i
8185		よりちいさ'いか/また'わひとしい	yorichiisa"ika/mata"wah_itoshii
8186		よりおおき'いか/また'わひとしい	yoriooki"ika/mata"wah_itoshii
8187		むげんだい	mugeNdai
8188		ゆえ'に	yue"ni
8189		おすき'ごー	os_uki"go-
818a		めすき'ごー	mes_uki"go-
818c		ふ'ん	hu"N
818d		びょ'う	byo"u
818e		せ'っし	se"sshi
818f	¥	えんま'ーく	eNma"-ku
8190	\$	ど'る	do"ru
8191	¢	せ'んと	se"Nto
8192	£	ぽ'んど	po"Ndo

表B-5 全角記号の読み(2/10)

コード	記号	読み	発音記号
8193	%	ぱーせ'んと	pa-se"Nto
8194	#	いげた	igeta
8195	&	あんばさ'んど	aNpasa"Ndo
8196	*	あすたり'すく	as_utari"s_uku
8197	@	あっとま'ーく	attoma"-ku
8198	§	せ'くしょん	se"k_ushoN
8199		しろ'ぼし	shiro"boshi
819a		くろ'ぼし	kuro"boshi
819b		しろまる	shiromaru
819c		くろまる	kuromaru
819d		にじゅう'まる	nijuu"maru
819e		ひしがた	h_ishigata
819f		くろひ'しがた	kurohi"shigata
81a0		しかく	sh_ikaku
81a1		くろし'かく	kuroshi"kaku
81a2		さ'んかく	sa"Nkaku
81a3		くろさ'んかく	kurosa"Nkaku
81a4		ぎやくさ'んかく	gyak_usa"Nkaku
81a5		くろぎやくさ'んかく	kurogyak_usa"Nkaku
81a6		こめじ'るし	komeji"rushi
81a7	〒	ゆーびん	yu-biN
81a8		みぎむきやじ'るし	migimukiyaji"rushi
81a9		ひだりむきやじ'るし	hidarimukiyaji"rushi
81aa		うえむきやじ'るし	uemukiyaji"rushi
81ab		したむきやじ'るし	sh_itamukiyaji"rushi
81b8		ぞくす'る	zok_usu"ru
81b9		ぞくす'る	zok_usu"ru
81ba		ぶいぶんしゅ'ーごー	bubuNshu"-go-
81bb		ぶいぶんしゅ'ーごー	bubuNshu"-go-
81bc		しんぶいぶんしゅ'ーごー	shiNbubuNshu"-go-
81bd		しんぶいぶんしゅ'ーごー	shiNbubuNshu"-go-
81be		がっぺーしゅ'ーごー	gappe-shu"-go-
81bf		きょーつーしゅ'ーごー	kyo-tsu-shu"-go-
81c8		ろんり'せき	roNri"seki
81c9		ろんり'わ	roNri"wa
81ca	ㇿ	ひてい	h_itei
81cb		が'んい	ga"Ni
81cc		どーとー	do-to-
81cd		ぜんしょーさよ'ーそ	zeNsho-sayo"-so

表B - 5 全角記号の読み (3/10)

コード	記号	読み	発音記号
81ce		そんざいさよ'ーそ	soNzaisayo"-so
81da		か'く	ka"ku
81db		ちよっこー	chokko-
81dc		しゃ'とー	sha"to-
81dd		らうんどでい'ー	rauNdodi"-
81de		な'ぶら	na"bura
81df		ごーどーき'ごー	go-do-ki"go-
81e0		ほと'んど/ひとし'い	hoto"Ndo/h_itoshi"i
81e1		きわ'めて/しょ'ー	kiwa"mete/sho"-
81e2		きわ'めて/だ'い	kiwa"mete//da"i
81e3		る'ーと	ru"-to
81e4		そーじき'ごー	so-jiki"go-
81e5		ひれーき'ごー	hire-ki"go-
81e6		なぜならき'ごー	nazenaraki"go-
81e7		せきぶんき'ごー	sekibuNki"go-
81e8		にじゅー/せきぶんきごー	niju-//sekibuNki"go-
81f0		おんぐすとろ'ーむ	oNgus_utoro"-mu
81f1	‰	ぱーみる	pa-miru
81f2		しゃ'ーぷ	sha"-pu
81f3		ふら'っと	hura"tto
81f4		はちぶんお'んぷ	hachibuNo"Npu
81f5	†	たんけ'んぷ	taNke"Nhu
81f6	‡	にじゅーたんけ'んぷ	niju-taNke"Nhu
81f7	¶	ぴ'ー	pi"-
81fc		しろまる	shiromaru
8740		まる'いち	maru"ichi
8741		ま'るに	ma"runi
8742		まる'さん	maru"saN
8743		まる'よん	maru"yoN
8744		ま'るご	ma"rugo
8745		まる'ろく	maru"roku
8746		まる'なな	maru"nana
8747		まる'はち	maru"hachi
8748		まる'きゅー	maru"kyu-
8749		まる'じゅー	maru"ju-
874a		まるじゅ'ーいち	maruju"-ichi
874b		まるじゅ'ーに	maruju"-ni
874c		まるじゅ'ーさん	maruju"-saN
874d		まるじゅ'ーよん	maruju"-yoN

表B - 5 全角記号の読み (4/10)

コード	記号	読み	発音記号
874e		まるじゅ'ーご	maruju"-go
874f		まるじゅ'ーろく	maruju"-roku
8750		まるじゅ'ーなな	maruju"-nana
8751		まるじゅ'ーはち	maruju"-hachi
8752		まるじゅ'ーきゅー	maruju"-kyu-
8753		まるに'じゅー	maruni"ju-
8754		いち	ichi
8755		に	ni
8756		さん	saN
8757		よ'ん	yo"N
8758		ご	go
8759		ろく	roku
875a		な'な	na"na
875b		はち	hachi
875c		きゅ'ー	kyu"-
875d		じゅ'ー	ju"-
875f	ミリ	み'り	mi"ri
8760	キロ	き'ろ	ki"ro
8761	センチ	せ'んち	se"Nchi
8762	メートル	め'ーとる	me"-toru
8763	グラム	ぐ'らむ	gu"ramu
8764	トン	と'ん	to"N
8765	アル	あ'ーる	a"-ru
8766	ヘクタール	へくた'ーる	hek_uta"-ru
8767	リットル	り'っとる	ri"ttoru
8768	ワット	わ'っと	wa"tto
8769	カリ	か'るりー	ka"rori-
876a	ドル	ど'る	do"ru
876b	セント	せ'んと	se"Nto
876c	パーセント	ぱーせ'んと	pa-se"Nto
876d	ミリメートル	みりば'ーる	miriba"-ru
876e	ジー	ぺ'ーじ	pe"-ji
876f	mm	みりめ'ーとる	mirime"-toru
8770	cm	せんちめ'ーとる	seNchime"-toru
8771	km	きろめ'ーとる	kirome"-toru
8772	mg	みりぐ'らむ	mirigu"ramu
8773	kg	きろぐ'らむ	kirogu"ramu
8774	cc	し'ーしー	shi"-shi-
8775	m ²	へいほーめ'ーとる	heiho-me"-toru

表B - 5 全角記号の読み (5/10)

コード	記号	読み	発音記号
877e	穢	へ'ーせー	he"-se-
8782		な'んばー	na"Nba-
8783	KK	け'ーけー	ke"-ke-
8784		て'る	te"ru
8785	⊕	じょ'う	jo"u
8786	⊕	ちゅ'う	chu"u
8787	⊕	げ	ge
8788	⊕	ひ'だり	hi"dari
8789	⊕	みぎ	migi
878a	(株)	かぶ	kabu
878b	(有)	ゆ'ー	yu"-
878c	(代)	だ'い	da"i
878d	聒	め'いじ	me"iji
878e	炬	たい'しょー	tai"sho-
878f	聒	しょ'ーわ	sho"-wa
8790		ほと'んど/ひと'しい	hoto"Ndo/h_ito"shii
8791		ご'ーどー	go"-do-
8792		せきぶんき'ごー	sekibuNki"go-
8793		せきぶんき'ごー	sekibuNki"go-
8794		し'ぐま	shi"guma
8795		る'ーと	ru"-to
8796		すい'ちよく	sui"choku
8797		か'く	ka"ku
8798		き'ごー	ki"go-
8799		き'ごー	ki"go-
879a		なぜな'らば	nazena"raba
879b		がっぺーき'ごー	gappe-ki"go-
879c		きょーつーしゅ'ーごー	kyo-tsu-shu"-go-
8260	A	え'ー	e"-
8261	B	び'ー	bi"-
8262	C	し'ー	shi"-
8263	D	でい'ー	di"-
8264	E	い'ー	i"-
8265	F	え'ふ	e"hu
8266	G	じ'ー	ji"-
8267	H	え'っち	e"cchi
8268	I	あ'い	a"i
8269	J	じえ'ー	je"-
826a	K	け'ー	ke"-

表B - 5 全角記号の読み (6/10)

コード	記号	読み	発音記号
826b	L	え'る	e"ru
826c	M	え'む	e"mu
826d	N	え'ぬ	e"nu
826e	O	お'ー	o"-
826f	P	ぴ'ー	pi"-
8270	Q	きゅ'ー	kyu"-
8271	R	あ'ーる	a"-ru
8272	S	え'す	e"s_u
8273	T	てい'ー	ti"-
8274	U	ゆ'ー	yu"-
8275	V	ヴ'い	vu"i
8276	W	だ'ぶりゅ	da"buryu
8277	X	え'っくす	e"kkus_u
8278	Y	わ'い	wa"i
8279	Z	ぜ'っと	ze"tto
8281	a	え'ー	e"-
8282	b	び'ー	bi"-
8283	c	し'ー	shi"-
8284	d	でい'ー	di"-
8285	e	い'ー	i"-
8286	f	え'ふ	e"hu
8287	g	じ'ー	ji"-
8288	h	え'っち	e"cchi
8289	i	あ'い	a"i
828a	j	じえ'ー	je"-
828b	k	け'ー	ke"-
828c	l	え'る	e"ru
828d	m	え'む	e"mu
828e	n	え'ぬ	e"nu
828f	o	お'ー	o"-
8290	p	ぴ'ー	pi"-
8291	q	きゅ'ー	kyu"-
8292	r	あ'ーる	a"-ru
8293	s	え'す	e"s_u
8294	t	てい'ー	ti"-
8295	u	ゆ'ー	yu"-
8296	v	ヴ'い	vu"i
8297	w	だ'ぶりゅ	da"buryu
8298	x	え'っくす	e"kkus_u

表B-5 全角記号の読み(7/10)

コード	記号	読み	発音記号
8299	y	わ'い	wa"i
829a	z	ぜ'っと	ze"tto
824f	0	ぜ'ろ	ze"ro
8250	1	いち'	ichi"
8251	2	に'	ni"
8252	3	さん	saN
8253	4	よ'ん	yo"N
8254	5	ご'	go"
8255	6	ろく'	roku"
8256	7	な'な	na"na
8257	8	はち'	hachi"
8258	9	きゅ'ー	kyu"-
839f		あ'るふあ	a"rufa
83a0		べ'ーた	be"-ta
83a1		が'んま	ga"Nma
83a2		で'るた	de"ruta
83a3		い'ぶしろん	i"p_ushiroN
83a4		つえ'ーた	tse"-ta
83a5		い'ーた	i"-ta
83a6		し'ーた	shi"-ta
83a7		い'おた	i"ota
83a8		か'っぱ	ka"ppa
83a9		ら'むだ	ra"muda
83aa		みゅ'ー	myu"-
83ab		にゅ'ー	nyu"-
83ac		くさ'い	k_usa"i
83ad		お'みくろん	o"mikuroN
83ae		ぱ'い	pa"i
83af		ろ'ー	ro"-
83b0		し'ぐま	shi"guma
83b1		た'う	ta"u
83b2		い'ぶしろん	i"p_ushiroN
83b3		ふあ'い	fa"i
83b4		か'い	ka"i
83b5		ぶさ'い	p_usa"i
83b6		お'めが	o"mega
83bf		あ'るふあ	a"rufa
83c0		べ'ーた	be"-ta
83c1		が'んま	ga"Nma

表B - 5 全角記号の読み (8/10)

コード	記号	読み	発音記号
83c2		で'るた	de"ruta
83c3		い'ぶしろん	i"p_ushiroN
83c4		つえ'-た	tse"-ta
83c5		い'-た	i"-ta
83c6		し'-た	shi"-ta
83c7		い'おた	i"ota
83c8		か'っぱ	ka"ppa
83c9		ら'むだ	ra"muda
83ca	μ	みゆ'-	myu"-
83cb		にゆ'-	nyu"-
83cc		くさ'い	k_usa"i
83cd		お'みくろん	o"mikuroN
83ce		ぱ'い	pa"i
83cf		ろ'-	ro"-
83d0		し'ぐま	shi"guma
83d1		た'う	ta"u
83d2		ゆ'ぶしろん	yu"p_ushiroN
83d3		ふあ'い	fa"i
83d4		か'い	ka"i
83d5		ふさ'い	p_usa"i
83d6		お'めが	o"mega
8440		あ'-	a"-
8441		べ'-	be"-
8442		ヴェ'-	ve"-
8443		げ'-	ge"-
8444		で'-	de"-
8445		いえ'-	ie"-
8446		いよ'-	iyo"-
8447		じえ'-	je"-
8448		ぜ'-	ze"-
8449		い'-	i"-
844a		い'-くら-とこえ	i"-kura-tokoe
844b		か'-	ka"-
844c		え'る	e"ru
844d		え'む	e"mu
844e		え'ぬ	e"nu
844f		お'-	o"-
8450		べ'-	pe"-
8451		え'る	e"ru

表B - 5 全角記号の読み (9/10)

コード	記号	読み	発音記号
8452		え'す	e"s_u
8453		て'ー	te"-
8454		う'ー	u"-
8455		え'ふ	e"hu
8456		は'ー	ha"-
8457		つえ'ー	tse"-
8458		ちえ'ー	che"-
8459		しゃ'ー	sha"-
845a		ししゃ'ー	shisha"-
845b		つぼるでいーずな'ーく	tsuborudi-zuna"-ku.
845c		うい'ー	ui"-
845d		みやーふいーずな'ーく	mya-fi-zuna"-ku.
845e		え'ー	e"-
845f		ゆ'ー	yu"-
8460		や'ー	ya"-
8470		あ'ー	a"-
8471		べ'ー	be"-
8472		ヴェ'ー	ve"-
8473		げ'ー	ge"-
8474		で'ー	de"-
8475		いえ'ー	ie"-
8476		いよ'ー	iyo"-
8477		じえ'ー	je"-
8478		ぜ'ー	ze"-
8479		い'ー	i"-
847a		い'ーくら-とこえ	i"-kura-tokoe
847b		か'ー	ka"-
847c		え'る	e"ru
847d		え'む	e"mu
847e		え'ぬ	e"nu
8480		お'ー	o"-
8481		べ'ー	pe"-
8482		え'る	e"ru
8483		え'す	e"s_u
8484		て'ー	te"-
8485		う'ー	u"-
8486		え'ふ	e"hu
8487		は'ー	ha"-
8488		つえ'ー	tse"-

表B - 5 全角記号の読み (10/10)

コード	記号	読み	発音記号
8489		ちえ'ー	che"-
848a		しゃ'ー	sha"-
848b		ししゃ'ー	shisha"-
848c		つぼるでいーずな'ーく	tsuborudi-zuna"-ku.
848d		うい'ー	ui"-
848e		みやーふいーずな'ーく	mya-fi-zuna"-ku.
848f		え'ー	e"-
8490		ゆ'ー	yu"-
8491		や'ー	ya"-

B. 4. 1 TTS SJIS コード表

SJIS	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
8140	SP	、	。	、	・	：	；	？	！	ゝ	°	´	、	”	^	
8150		—	、	、	、	”	全	々	々	〇	—			/	\	
8160	~			...	‘	’	“	”	()	[]	[]	{	
8170	}			《	》	「	」	『	』	【	】	+	-	±	x	
8180	÷	=		<	>							°				¥
8190	\$	¢	£	%	#	&	*	@	§							
81a0								〒								
81b0																
81c0																
81d0																
81e0																
81f0		%				†	‡	¶								
8240																0
8250	1	2	3	4	5	6	7	8	9							
8260	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
8270	Q	R	S	T	U	V	W	X	Y	Z						
8280		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
8290	p	q	r	s	t	u	v	w	x	y	z					あ
82a0	あ	い	い	う	う	え	え	お	お	か	が	き	ぎ	く	ぐ	け
82b0	げ	こ	こ	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た	だ	ち
82c0	ぢ	っ	っ	づ	て	で	と	ど	な	に	ぬ	ね	の	は	ば	ば
82d0	ひ	び	び	ふ	ぶ	ぶ	へ	べ	べ	ほ	ぼ	ぼ	ま	み	む	め
82e0	も	ゃ	ゃ	ゅ	ゅ	ょ	ょ	ら	り	る	れ	ろ	わ	わ	ゐ	ゑ
82f0	を	ん														
8340	ア	ア	イ	イ	ウ	ウ	エ	エ	オ	オ	カ	ガ	キ	ギ	ク	グ
8350	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ	ダ
8360	チ	ヂ	ツ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ
8370	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ	
8380	ム	メ	モ	ヤ	ヤ	ユ	ユ	ヨ	ヨ	ラ	リ	ル	レ	ロ	ワ	ワ
8390	ヰ	ヱ	ヲ	ン	ヴ	カ	ケ									
83a0																
83b0																
83c0																
83d0																
83e0																
83f0																
8440																
8450																
8460																
8470																
8480																
8490																

84a0																
84b0																
84c0																
84d0																
84e0																
84f0																
8640																
8650	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
8660	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
8670	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
8680	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
8690																
86a0			・	・	・	・	・	・	・	・	・	・	・	・	・	・
86b0	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
86c0	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
86d0	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
86e0	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
86f0																
8740																
8750																ミリ
8760	キロ	キ	メ	グ	ト	ア	キ	リ	ワ	カ	ド	セ	シ	ギ		mm
8770	cm	km	mg	kg	cc	m ²								靴		
8780	”	”		KK		⊕	⊕	⊕	⊕	⊕	(株)	(有)	(代)	職	証	職
8790																
87a0																
87b0																
87c0																
87d0																
87e0																
87f0																
8840																
8850																
8860																
8870																
8880																
8890																亜
88a0	唾	娃	阿	哀	愛	挨	始	逢	葵	茜	穉	惡	握	渥	旭	葦
88b0	芦	鯨	梓	庄	幹	扱	宛	姐	虻	飴	絢	綾	鮎	或	粟	裕
88c0	安	庵	按	暗	案	闇	鞍	杏	以	伊	位	依	偉	困	夷	委
88d0	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃	菱	衣
88e0	謂	違	遺	医	井	亥	域	育	郁	磯	一	巷	溢	逸	稻	茨
88f0	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭			
8940	院	陰	隱	韻	吋	右	宇	烏	羽	迂	雨	卯	鵝	窺	丑	碓
8950	臼	渦	噓	唄	鬱	蔚	鰻	姥	厩	浦	瓜	閏	噲	云	運	雲



8960	荏	餌	叡	嘗	嬰	影	映	曳	榮	永	泳	洩	瑛	盈	穎	穎
8970	英	衛	詠	銳	液	疫	益	馭	悅	謁	越	閱	榎	厭	円	
8980	園	堰	奄	宴	延	怨	掩	援	沿	演	炎	焰	煙	燕	猿	縁
8990	艷	苑	菴	遠	鉛	鴛	塩	於	汚	甥	凹	央	奧	往	応	押
89a0	旺	橫	歐	毆	王	翁	襖	篤	鷗	甥	岡	沖	荻	億	屋	憶
89b0	臆	桶	牡	乙	俺	卸	恩	温	穩	音	岡	化	仮	何	伽	億
89c0	佳	加	可	嘉	夏	嫁	家	寡	科	暇	果	架	仮	河	伽	億
89d0	禍	禾	稼	箇	花	苛	茄	荷	華	菓	蝦	課	嘩	火	伽	億
89e0	霞	蚊	俄	峨	我	牙	画	臥	芽	菓	賀	課	嘩	火	伽	億
89f0	解	回	塊	壞	迴	快	怪	悔	恢	懷	戒	拐	改	介	會	會
8a40	魁	晦	械	海	灰	界	皆	繪	芥	蟹	開	階	貝	凱	幼	外
8a50	咳	害	崖	慨	概	涯	碍	蓋	街	該	鎖	骸	湮	馨	蛙	垣
8a60	柿	蛻	鈎	劃	嚇	各	廓	拈	攪	格	核	殼	獲	確	穫	覺
8a70	角	赫	較	郭	閣	隔	革	括	岳	樂	額	顎	掛	笠	櫛	
8a80	檀	梶	鯁	渴	割	喝	恰	括	活	渴	滑	葛	褐	轄	且	鯉
8a90	叶	花	樺	鞆	株	兜	竈	蒲	釜	鎌	嚙	鴨	栢	茅	萱	粥
8aa0	刈	苜	瓦	乾	侃	冠	寒	刊	勘	勸	卷	喚	堪	姦	完	官
8ab0	寬	干	幹	患	感	慣	憾	換	敢	柑	桓	棺	款	歡	汗	漢
8ac0	潤	漚	環	甘	監	看	竿	管	簡	緩	缶	翰	肝	艦	莞	觀
8ad0	諫	貫	還	鑑	間	閑	閑	陷	韓	館	缶	丸	含	艦	莞	觀
8ae0	癩	眼	岩	翫	贗	雁	頑	顏	願	館	伎	危	喜	岸	巖	玩
8af0	嬉	寄	岐	希	幾	忌	揮	机	旗	企	期	棋	棄	器	巖	奇
8b40	機	歸	毅	氣	汽	畿	祈	季	稀	紀	徽	規	記	貴	起	軌
8b50	輝	飢	騎	鬼	龜	偽	儀	妓	宜	戲	技	擬	欺	儀	疑	祇
8b60	義	蟻	誼	議	掬	菊	鞠	吉	吃	喫	桔	橘	詰	砧	杵	黍
8b70	却	客	脚	虐	逆	丘	久	仇	休	及	吸	宮	弓	急	救	
8b80	朽	求	汲	泣	灸	球	究	窮	笈	級	糾	給	旧	牛	去	居
8b90	巨	拒	拋	泣	渠	虛	許	距	鋸	漁	禦	魚	亨	享	京	供
8ba0	俠	僑	兇	泣	共	凶	協	匡	錫	叫	喬	境	峽	強	疆	怯
8bb0	恐	恭	挾	教	橋	況	狂	狹	矯	胸	脅	興	蕎	鄉	鏡	響
8bc0	饗	驚	仰	凝	堯	曉	業	局	曲	極	玉	桐	糝	僅	勤	均
8bd0	巾	錦	斤	欣	欽	琴	禁	禽	筋	緊	芹	菌	衿	襟	謹	近
8be0	金	吟	銀	九	俱	句	区	狗	玖	矩	苦	軀	驅	駟	駒	具
8bf0	愚	虞	喰	空	偶	寓	遇	隅	串	櫛	釧	屑	屈			
8c40	掘	窟	沓	靴	轡	窪	熊	隈	柔	栗	繰	桑	鍬	勲	君	薰
8c50	訓	群	軍	郡	卦	袞	祁	係	傾	刑	兄	啓	圭	珪	型	契
8c60	形	徑	患	慶	慧	憩	揭	携	敬	景	桂	溪	畦	稽	系	經
8c70	繼	繫	罴	荳	荊	蚩	計	詣	警	輕	頸	鷄	芸	迎	鯨	
8c80	劇	戟	擊	激	隙	桁	傑	欠	決	潔	穴	結	芸	訣	月	件
8c90	俟	倦	健	兼	券	劍	喧	圈	堅	嫌	建	憲	懸	拳	捲	檢
8ca0	權	牽	犬	獻	研	硯	絹	梟	肩	見	謙	賢	軒	遣	鍵	險
8cb0	頭	驗	鯨	元	原	廠	幻	弦	減	源	玄	現	絃	肱	言	諺
8cc0	限	乎	個	古	呼	固	姑	孤	己	庫	弧	戶	故	枯	湖	狐
8cd0	糊	袴	股	胡	菰	虎	誇	跨	鈷	雇	顧	鼓	互	互	伍	午
8ce0	吳	吾	娛	後	御	悟	梧	檣	瑚	暮	語	誤	護	乞	鯉	鯉



8cf0	交	佼	侯	候	倅	光	公	功	効	勾	厚	口	向				
8d40	后	喉	坑	垢	好	孔	孝	宏	工	巧	巷	幸	広	庚	康	弘	
8d50	恒	慌	抗	拘	控	攻	昂	晃	更	杭	校	梗	構	江	洪	浩	
8d60	港	溝	甲	皇	硬	稿	糠	紅	紘	絞	綱	耕	考	肯	肱	腔	
8d70	膏	航	荒	行	衡	講	貢	購	郊	酵	鉞	砘	鋼	閤	降		
8d80	項	香	高	鴻	剛	劫	号	合	壕	拷	濠	豪	轟	鞠	克	刻	
8d90	告	国	穀	酷	鵠	黑	獄	漉	腰	甌	忽	惚	骨	狢	込	此	
8da0	頃	今	困	坤	壘	婚	恨	懇	昏	昆	根	梱	混	痕	紺	良	
8db0	魂	些	佐	又	唆	嵯	左	差	查	沙	磋	砂	詐	鎖	裳	坐	
8dc0	座	挫	債	催	再	最	哉	塞	妻	宰	彩	才	採	裁	歲	濟	
8dd0	災	采	犀	碎	砦	祭	齋	細	菜	裁	載	際	剖	在	材	罪	
8de0	財	冴	坂	阪	堺	柵	肴	咲	崎	埼	碕	鷺	作	削	咋	擗	
8df0	昨	朔	柵	窄	策	索	錯	桜	鮭	筵	匙	冊	刷				
8e40	察	撈	撮	擦	札	殺	薩	雜	皐	鯖	捌	鑄	鮫	皿	晒	三	
8e50	傘	參	山	慘	撒	散	棧	燦	珊	産	算	纂	蚕	讚	贛	酸	
8e60	餐	斬	暫	残	仕	仔	伺	使	刺	司	史	嗣	四	士	始	姉	
8e70	姿	子	屍	市	師	志	思	指	支	孜	斯	施	旨	枝	止		
8e80	死	氏	獅	祉	私	糸	紙	紫	肢	脂	至	視	詞	詩	試	誌	
8e90	諮	資	賜	雌	飼	齒	事	似	侍	兒	字	寺	慈	持	時	次	
8ea0	滋	治	爾	璽	痔	磁	示	而	耳	自	蒔	辞	汐	鹿	式	識	
8eb0	鳴	竺	軸	穴	零	七	叱	執	失	嫉	室	悉	湿	漆	疾	質	
8ec0	実	葩	篠	悃	柴	芝	屢	蕊	緝	舍	写	射	捨	赦	斜	煮	
8ed0	社	紗	者	謝	車	遮	蛇	邪	借	勺	尺	杓	灼	爵	酌	积	
8ee0	錫	若	寂	弱	惹	主	取	守	手	朱	殊	狩	珠	腫	趣		
8ef0	酒	首	儒	受	呪	寿	授	樹	綬	需	囚	収	周				
8f40	宗	就	州	修	愁	拾	洲	秀	秋	終	繡	習	臭	舟	蒐	衆	
8f50	襲	讐	蹴	輯	週	酋	酬	集	醜	什	住	充	十	從	戎	柔	
8f60	汁	洪	獸	縱	重	銃	叔	夙	宿	淑	祝	縮	肅	塾	熟	出	
8f70	術	洪	俊	峻	春	瞬	竣	舜	駿	准	循	旬	楯	殉	淳		
8f80	準	潤	盾	純	巡	遵	醇	順	処	初	所	暑	曙	渚	庶	緒	
8f90	署	書	薯	譜	諸	助	叙	女	序	徐	恕	鋤	除	傷	償	勝	
8fa0	匠	升	召	哨	商	唱	嘗	奨	妾	娼	宵	將	少	尚	庄		
8fb0	床	廠	彰	承	抄	招	掌	捷	昇	昌	昭	晶	梢	樟	樵		
8fc0	沼	消	涉	湘	燒	焦	照	症	省	硝	礁	祥	章	粧	粧		
8fd0	紹	肖	菖	蔣	蕉	衝	裳	訟	証	詔	詳	象	醬	笑	錠		
8fe0	鐘	障	鞘	上	丈	丞	乘	冗	剌	城	場	壤	常	情	擾		
8ff0	条	杖	淨	狀	量	穰	蒸	讓	釀	錠	囑	墳	飾				
9040	拭	植	殖	燭	織	職	色	觸	食	蝕	辱	尻	伸	信	侵	唇	
9050	娠	寢	審	心	慎	振	新	晉	森	榛	浸	深	申	疹	真	神	
9060	秦	紳	臣	芯	薪	親	診	身	辛	進	針	震	人	仁	刃	塵	
9070	壬	尋	甚	尽	腎	訊	迅	陣	韌	筈	諷	須	酢	厨			
9080	逗	吹	垂	帥	推	水	炊	睡	粹	翠	衰	遂	醉	錘	隨		
9090	瑞	髓	崇	嵩	数	枢	趨	離	据	杉	相	菅	頗	雀	澄		
90a0	摺	寸	世	瀨	畝	是	淒	制	勢	姓	征	性	成	政	整	星	

90b0	晴	棲	栖	正	清	牲	生	盛	精	聖	声	製	西	誠	誓	請
90c0	逝	醒	青	静	齐	税	脆	隻	席	惜	戚	斥	昔	析	石	積
90d0	籍	績	脊	責	赤	跡	蹟	碩	切	拙	接	撰	折	設	窃	節
90e0	說	雪	絕	舌	蝉	仙	先	千	占	宣	專	尖	川	戟	扇	撰
90f0	栓	栴	泉	浅	洗	染	潜	煎	煽	旋	穿	箭	線			
9140	織	羨	腺	舛	船	薦	詮	賤	踐	選	遷	錢	銑	閃	鮮	前
9150	善	漸	然	全	禅	繕	膳	糗	贈	塑	岨	措	曾	曾	楚	狙
9160	疏	疎	礎	祖	租	粗	素	組	蘇	訴	阻	邈	鼠	僧	創	双
9170	叢	倉	喪	壯	奏	爽	宋	層	匠	忽	想	搜	掃	挿	搔	
9180	操	早	曹	巢	槍	槽	漕	燥	争	瘦	相	窓	掃	綜	綜	聰
9190	草	莊	葬	蒼	藻	裝	走	送	遭	鎗	霜	騷	像	增	憎	臟
91a0	葳	贈	造	促	側	則	即	息	捉	束	測	足	速	俗	属	賊
91b0	族	統	卒	袖	其	揃	存	孫	尊	損	村	遜	他	多	太	汰
91c0	訖	唾	墮	妥	情	打	柁	舵	椅	陀	馱	驛	体	堆	对	耐
91d0	岱	帶	待	怠	態	戴	替	泰	滯	胎	腿	苔	袋	貸	退	逮
91e0	隊	黛	鯛	代	台	大	第	醜	題	鷹	淹	卓	卓	宅	宅	托
91f0	扞	拓	沢	濯	琢	託	鐸	濁	諾	苴	胤	蛸	只			
9240	叩	但	達	辰	奪	脫	巽	豎	辿	棚	谷	狸	鱈	樽	誰	丹
9250	单	嘆	坦	担	探	旦	歎	淡	湛	炭	短	端	筆	綻	耽	胆
9260	蛋	誕	鍛	団	壇	彈	断	暖	檀	段	男	談	值	知	地	弛
9270	恥	智	池	痴	稚	置	致	蚺	遲	馳	築	畜	竹	筑	蓄	
9280	逐	秩	室	茶	嫡	着	中	仲	宙	忠	抽	昼	柱	注	虫	衷
9290	註	耐	鏄	駐	檣	漭	猪	苧	著	貯	丁	兆	凋	喋	龍	帖
92a0	帳	庁	弔	張	彫	徵	懲	挑	暢	朝	潮	牒	町	眺	聽	脹
92b0	腸	蝶	調	謀	超	跳	黽	長	頂	朝	勅	抄	直	朕	沈	珍
92c0	賃	鎮	陳	津	墜	椎	槌	追	鎚	痛	通	塚	梅	掴	楓	佃
92d0	漬	柘	辻	薦	綴	鏢	樁	潰	坪	壺	孀	紬	爪	吊	釣	鶴
92e0	亭	低	停	偵	剃	貞	呈	堤	定	帝	底	庭	廷	弟	悌	抵
92f0	挺	提	梯	汀	碇	禎	程	締	艇	訂	諦	蹄	遞			
9340	邸	鄭	釘	鼎	泥	摘	擢	敵	滴	的	笛	適	適	溺	哲	徹
9350	撤	轍	迭	鉄	典	填	天	展	店	添	纏	甜	貼	軋	顛	点
9360	伝	殿	澱	田	電	兎	吐	堵	塗	妬	屠	徒	斗	杜	渡	登
9370	菟	賭	途	都	鍍	砥	砺	努	度	土	奴	怒	倒	党	冬	
9380	凍	刀	唐	塔	塘	套	宕	鳥	嶋	悼	投	搭	東	桃	椿	棟
9390	盜	淘	湯	涛	灯	燈	当	痘	禱	等	答	筒	糖	統	到	董
93a0	蕩	藤	討	膳	豆	踏	逃	透	鐙	陶	頭	騰	鬪	働	動	同
93b0	堂	導	懂	撞	洞	瞳	童	胴	萄	道	銅	峠	鶉	得	得	德
93c0	澆	特	督	禿	篤	毒	独	読	柝	橡	凸	突	届	篤	苦	那
93d0	寅	酉	瀨	噸	屯	惇	敦	沌	豚	遁	頓	吞	曇	鈍	那	難
93e0	内	乍	凧	薙	謎	灘	捺	鍋	槽	馴	繩	嚼	楠	軟	軟	
93f0	汝	二	尼	弍	迓	勺	賑	肉	虹	甘	日	乳	入			
9440	如	尿	菲	任	妊	忍	認	濡	襦	祢	寧	葱	猫	熱	年	念
9450	捻	撚	燃	粘	乃	迺	之	埜	囊	惱	濃	納	能	腦	膿	農
9460	覘	蚤	巴	把	播	霸	杷	波	派	琶	破	婆	罵	芭	馬	俳



9470	糜	拌	排	敗	杯	盃	牌	背	肺	輩	配	倍	培	媒	梅	
9480	煤	煤	排	買	壳	賠	陪	這	蠅	秤	矧	荻	伯	剥	博	拍
9490	柏	泊	狼	買	壳	賠	陪	這	蠅	秤	矧	荻	伯	剥	博	拍
94a0	箱	裕	白	箔	粕	舶	薄	迫	曝	漠	爆	縛	莫	駁	麥	函
94b0	伐	罰	箸	肇	筍	櫨	幡	肌	烟	阜	八	鉢	澆	駁	醜	髮
94c0	搬	斑	拔	筏	筍	鳩	晰	塙	蛤	隼	伴	判	半	駁	叛	帆
94d0	頒	斑	板	汜	汎	版	犯	班	畔	繁	般	藩	販	駁	采	煩
94e0	扉	批	挽	晚	番	盤	磐	蕃	蛮	匪	卑	否	妃	庇	彼	悲
94f0	避	非	飛	斐	比	泌	疲	皮	碑	秘	緋	罷	肥	被	誹	費
9540	鼻	柶	稗	匹	疋	髭	彦	膝	菱	肘	弼	必	畢	筆	逼	桧
9550	姬	媛	紐	百	謬	儀	彪	標	冰	漂	瓢	票	表	評	豹	廟
9560	描	病	秒	苗	錨	鋌	蒜	蛭	鱮	品	彬	斌	浜	瀕	貧	竇
9570	頻	敏	瓶	不	付	埠	夫	婦	富	富	布	府	怖	扶	敷	
9580	斧	普	浮	父	符	腐	膚	芙	富	負	賦	赴	阜	附	侮	撫
9590	武	舞	葡	蕪	部	封	楓	風	葺	蔭	伏	副	復	幅	服	福
95a0	腹	複	覆	淵	弗	弘	沸	佻	物	鮒	分	吻	噴	墳	憤	扮
95b0	焚	奮	粉	糞	紛	雰	文	聞	丙	併	兵	塀	幣	平	弊	柄
95c0	並	蔽	閉	陛	米	頁	僻	壁	癖	碧	別	警	蔑	篋	偏	變
95d0	片	篇	編	辺	返	遍	便	勉	婉	弁	鞭	保	舖	圃	捕	包
95e0	步	甫	補	輔	穗	募	墓	戊	暮	母	放	簿	菩	倖		
95f0	呆	報	奉	宝	峰	峯	崩	庖	抱	捧	方	朋	朋	倖		
9640	法	泡	烹	砲	縫	胞	芳	萌	蓬	蜂	褒	訪	豐	邦	鋒	飽
9650	鳳	鵬	乏	亡	傍	剖	坊	妨	帽	忘	忙	房	暴	望	某	棒
9660	冒	紡	肪	膨	謀	貌	貿	鉞	防	吠	頰	北	僕	卜	墨	撲
9670	朴	牧	睦	穆	釦	勃	沒	殆	堀	幌	奔	本	翻	凡	盆	
9680	摩	磨	魔	麻	埋	妹	昧	枚	每	哩	槓	幕	膜	枕	鮪	枉
9690	鱒	榭	亦	侯	又	抹	末	沫	迄	俛	繭	磨	萬	慢	滿	漫
96a0	蔓	味	未	魅	已	箕	岬	密	蜜	湊	蓑	稔	脈	妙	耗	民
96b0	眠	務	夢	無	牟	矛	霧	鷓	棕	媚	娘	冥	名	命	明	盟
96c0	迷	銘	鳴	姪	牝	滅	免	棉	綿	緬	面	麵	摸	模	茂	妄
96d0	孟	毛	猛	盲	網	耗	蒙	儲	木	默	目	杳	勿	餅	尤	戾
96e0	勑	賞	問	悶	紋	門	匆	也	冶	夜	爺	耶	野	餅	戾	厄
96f0	役	約	葉	訊	躍	靖	柳	藪	鑊	愉	愈	油	癒	彌		
9740	諭	輸	唯	佑	優	勇	友	宥	幽	悠	憂	揖	有	袖	湧	涌
9750	猶	猷	由	祐	裕	誘	遊	邑	郵	雄	融	夕	予	余	與	譽
9760	輿	預	傭	幼	妖	容	庸	揚	搖	擁	曜	楊	樣	洋	溶	譽
9770	用	窳	羊	耀	葉	蓉	要	謠	踊	遙	陽	養	慾	抑	欲	
9780	沃	浴	翌	翼	淀	羅	螺	裸	來	萊	賴	雷	洛	絡	落	酪
9790	乱	卵	嵐	欄	濫	藍	蘭	覽	利	吏	履	李	梨	理	璃	痢
97a0	裏	裡	里	離	陸	律	率	立	律	掠	略	劉	流	溜	琉	留
97b0	硫	粒	隆	竜	龍	侶	慮	旅	虜	了	亮	僚	兩	凌	寮	料
97c0	梁	涼	獵	療	瞭	稜	糧	良	諒	遼	量	陵	領	力	綠	倫
97d0	厘	林	淋	憐	琳	臨	輪	隣	麟	鈴	瑠	壘	淚	累	類	令
97e0	伶	例	冷	勵	嶺	伶	玲	禮	苓	鈴	隸	零	靈	麗	齡	曆
97f0	歷	列	劣	烈	裂	廉	戀	憐	漣	煉	簾	練	聯			

9840	蓮	連	鍊	呂	魯	櫓	炉	賂	路	露	勞	婁	廊	弄	朗	樓
9850	榔	浪	漏	牢	狼	篋	老	贖	蠟	郎	六	麓	祿	肋	錄	論
9860	倭	和	話	歪	賄	脇	惑	梓	驚	互	巨	鱒	詫	藁	蕨	椀
9870	湾	碗	腕													
9880																
9890																式
98a0	丐	丕	个	卍	丶	井	丿	乂	乖	乘	亂	丿	豫	爭	舒	式
98b0	于	亞	亟	一	亢	京	毫	亶	从	仍	仄	仆	仂	仗	仞	仍
98c0	仟	价	伉	侏	估	佛	佝	佗	佇	佖	侈	侏	侗	佻	佩	佖
98d0	侑	伴	來	侑	儘	倪	俟	俎	俘	俛	俑	俚	侗	佻	俚	倚
98e0	倨	偃	倪	侗	倅	倅	倅	倡	倩	倅	倅	倅	們	們	偃	假
98f0	會	偕	修	偈	倅	倅	倅	倅	倅	倅	倅	倅	倅	倅		
9940	僉	僊	傳	僂	僂	僂	僂	僂	僂	僂	僂	僂	儉	僂	儂	儉
9950	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉	儉
9960	兪	兮	冀	冂	冂	册	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂
9970	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂
9980	鳳	冂	函	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂	冂
9990	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓
99a0	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓	劓
99b0	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆	匆
99c0	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅	卅
99d0	廠	厶	參	纂	雙	叟	曼	變	叮	叨	叭	叭	吁	吁	呻	呻
99e0	吭	吼	吮	吮	吮	吮	吮	吮	吮	吮	吮	吮	呻	呻	呻	呻
99f0	咀	嗽	咄	咄	咄	咄	咄	咄	咄	咄	咄	咄	咨	咨		
9a40	咫	晒	咤	咤	咤	咤	咤	咤	咤	咤	咤	咤	哭	咤	咤	咤
9a50	唯	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣
9a60	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣	啣
9a70	噴	嘔	嗽	噴	噴	噴	噴	噴	噴	噴	噴	噴	噴	噴	噴	噴
9a80	噫	噤	嘯	噤	噤	噤	噤	噤	噤	噤	噤	噤	噤	噤	噤	噤
9a90	嚼	囁	囁	囁	囁	囁	囁	囁	囁	囁	囁	囁	囁	囁	囁	囁
9aa0	國	圍	圓	團	圖	圖	圖	圖	圖	圖	圖	圖	圖	圖	圖	圖
9ab0	垩	坡	坩	坩	垩	垩	垩	垩	垩	垩	垩	垩	垩	垩	垩	垩
9ac0	坩	埤	埤	埤	埤	埤	埤	埤	埤	埤	埤	埤	埤	埤	埤	埤
9ad0	墟	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩	壩
9ae0	壘	壯	壺	壺	壺	壺	壺	壺	壺	壺	壺	壺	壺	壺	壺	壺
9af0	夸	夾	奇	奕	奕	奕	奕	奕	奕	奕	奕	奕	奕	奕	奕	奕
9b40	奸	妁	妝	佞	佞	妣	妣	姆	姨	姜	妍	妊	姚	娥	娟	娑
9b50	娜	娉	娉	姍	姍	婉	婉	媪	媪	婪	媚	媼	媼	媼	媼	媽
9b60	媽	嫗	嫗	嫩	嫗	嫗	嫗	嬌	嬋	嬋	嬋	嬋	嬋	嬋	嬋	嬋
9b70	嫗	孀	子	孕	孚	孛	孛	孩	孰	孳	孳	孳	孳	孳	孳	孳
9b80	它	宦	宸	寃	寇	崔	寃	寃	寃	寃	寃	寃	寃	寃	寃	寃
9b90	寶	尅	將	專	對	尔	尅	尅	尅	尅	尅	尅	尅	尅	尅	尅
9ba0	屏	孱	屬	屮	屮	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌
9bb0	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌	岌



9bc0	崑	崔	崢	峻	崙	崘	崚	崛	崜	崝	崞	崟	崠	崡	崢	崣
9bd0	嶢	嶣	嶤	嶥	嶦	嶧	嶨	嶩	嶪	嶽	嶼	嶽	嶽	嶽	嶽	嶽
9be0	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮	卮
9bf0	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣	幣
9c40	廖	廣	廡	廚	廛	廢	廡	廡	廡	廡	廡	廡	廡	廡	廡	廡
9c50	弃	井	彝	彝	弋	弋	弋	弋	弋	弋	弋	弋	弋	弋	弋	弋
9c60	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖	彖
9c70	徘	徠	徠	徠	徠	徠	徠	徠	徠	徠	徠	徠	徠	徠	徠	徠
9c80	怙	恂	怙	恂	恂	恂	恂	恂	恂	恂	恂	恂	恂	恂	恂	恂
9c90	協	恆	恍	恣	恃	恤	恂	恂	恂	恂	恂	恂	恂	恂	恂	恂
9ca0	俊	悖	悒	悒	悒	悒	悒	悒	悒	悒	悒	悒	悒	悒	悒	悒
9cb0	愠	愕	愠	愠	愠	愠	愠	愠	愠	愠	愠	愠	愠	愠	愠	愠
9cc0	愨	愧	慊	慊	愨	愨	愨	愨	愨	愨	愨	愨	愨	愨	愨	愨
9cd0	懼	傷	慄	慄	慄	慄	慄	慄	慄	慄	慄	慄	慄	慄	慄	慄
9ce0	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫	憫
9cf0	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺	懺
9d40	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛	戛
9d50	抉	找	抒	抓	抖	拔	扞	扞	扞	扞	扞	扞	扞	扞	扞	扞
9d60	拜	拌	拊	拂	拇	拋	拉	拈	拈	拈	拈	拈	拈	拈	拈	拈
9d70	挾	捍	搜	捏	掖	掎	掀	掀	掀	掀	掀	掀	掀	掀	掀	掀
9d80	捺	掾	揩	揀	揆	揣	揉	插	擲	揄	搖	搯	撿	撿	撿	撿
9d90	攝	搗	搗	搗	搗	搗	搗	搗	搗	搗	搗	搗	搗	搗	搗	搗
9da0	擒	擅	擇	撻	擘	搯	搯	搯	搯	搯	搯	搯	搯	搯	搯	搯
9db0	擴	擲	擺	攀	攪	攘	攜	攢	攤	攢	攢	攢	攢	攢	攢	攢
9dc0	斲	效	敖	敕	敍	敍	敍	敍	敍	敍	敍	敍	敍	敍	敍	敍
9dd0	斷	旃	旃	旃	旃	旃	旃	旃	旃	旃	旃	旃	旃	旃	旃	旃
9de0	杳	昵	昶	昶	昶	昶	昶	昶	昶	昶	昶	昶	昶	昶	昶	昶
9df0	晰	晁	暈	暈	暈	暈	暈	暈	暈	暈	暈	暈	暈	暈	暈	暈
9e40	曄	瞭	曖	曖	曠	眈	曠	曠	曠	曠	曠	曠	曠	曠	曠	曠
9e50	霸	朮	束	朮	朮	朮	朮	朮	朮	朮	朮	朮	朮	朮	朮	朮
9e60	杼	杪	杪	杪	杪	杪	杪	杪	杪	杪	杪	杪	杪	杪	杪	杪
9e70	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝	柝
9e80	梳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳
9e90	梵	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳	栳
9ea0	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧	棧
9eb0	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣
9ec0	棟	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣
9ed0	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷	榷
9ee0	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅	樅
9ef0	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣
9f40	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣
9f50	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣	檣
9f60	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃	歃
9f70	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫	殫

9f80	麾	氈	氓	气	氛	氲	氣	汞	汕	汙	汪	沂	沅	沚	沁	沛
9f90	汾	汨	汭	沒	沐	泄	決	泓	沽	泗	汜	沂	沮	沚	沁	沛
9fa0	泛	泯	汭	汨	洩	衍	洶	洳	洽	洸	洑	洵	洳	洳	洳	洳
9fb0	涓	涇	浚	浹	浙	涎	洩	洳	洳	洳	洳	洳	洳	洳	洳	洳
9fc0	涓	涇	浚	浹	浙	涎	洩	洳	洳	洳	洳	洳	洳	洳	洳	洳
9fd0	渙	浚	淞	淌	淨	淒	淒	淒	淒	淒	淒	淒	淒	淒	淒	淒
9fe0	游	浚	淞	渾	渣	湫	淒	淒	淒	淒	淒	淒	淒	淒	淒	淒
9ff0	溟	潁	溉	灌	滄	滄	滄	滄	滄	滄	滄	滄	滄	滄	滄	滄
e040	漾	漓	涵	澆	滂	潛	澁	澀	澀	潛	潛	潭	澁	澁	潘	澎
e050	澆	漓	涵	澆	滂	潛	澁	澀	澀	潛	潛	潭	澁	澁	潘	澎
e060	濮	濛	瀉	瀆	濇	瀑	澆	澆	澆	澆	澆	澆	澆	澆	澆	澆
e070	瀾	濛	瀉	灣	炙	炒	炯	炯	炬	炸	炳	炮	烟	烟	烟	烟
e080	烙	焉	烽	焜	焙	煥	熙	熙	煦	煖	煖	煖	煖	煖	煖	煖
e090	煩	熨	熬	爛	熹	熾	熾	熾	熾	熾	熾	熾	熾	熾	熾	熾
e0a0	耀	爍	爐	爛	爨	爭	爬	爨	爨	爨	爨	爨	爨	爨	爨	爨
e0b0	牴	牾	犁	犁	犇	犇	犇	犇	犇	犇	犇	犇	犇	犇	犇	犇
e0c0	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝	狝
e0d0	獎	獾	默	獾	獾	獨	獾	獾	獾	獾	獾	獾	獾	獾	獾	獾
e0e0	珥	珥	珥	珥	珥	瑯	瑯	瑯	瑯	瑯	瑯	瑯	瑯	瑯	瑯	瑯
e0f0	瑩	瑰	瑣	瑪	瑤	瑾	璋	璞	璧	瓊	瓊	瓊	瓊	瓊	瓊	瓊
e140	瓠	瓣	舂	舂	瓮	甌	甌	甌	甌	瓷	甄	甌	甌	甌	甌	甌
e150	瓠	瓣	舂	舂	瓮	甌	甌	甌	甌	瓷	甄	甌	甌	甌	甌	甌
e160	畫	睺	畸	當	疆	疇	疇	疇	疇	疇	疇	疇	疇	疇	疇	疇
e170	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖	疖
e180	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼	痼
e190	瘰	癭	癭	癭	癭	癭	癭	癭	癭	癭	癭	癭	癭	癭	癭	癭
e1a0	夬	癸	發	皂	兒	販	咬	咬	咬	咬	咬	咬	咬	咬	咬	咬
e1b0	皺	孟	盍	盍	盒	盞	盞	盞	盞	盞	盞	盞	盞	盞	盞	盞
e1c0	昵	眞	皆	眦	昧	眷	睇	睇	睇	睇	睇	睇	睇	睇	睇	睇
e1d0	瞎	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋	瞋
e1e0	矜	矣	矮	矸	砌	砒	礦	砒	礪	砒	砒	砒	砒	砒	砒	砒
e1f0	碣	碩	碓	碓	碓	碓	碓	碓	碓	碓	碓	碓	碓	碓	碓	碓
e240	磧	磚	磽	磽	磽	磽	磽	磽	磽	磽	磽	磽	磽	磽	磽	磽
e250	被	祺	祿	禊	禊	禊	禊	禊	禊	禊	禊	禊	禊	禊	禊	禊
e260	菽	秣	稈	稍	稍	稍	稍	稍	稍	稍	稍	稍	稍	稍	稍	稍
e270	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡	穡
e280	囊	竅	竅	竅	竅	竅	竅	竅	竅	竅	竅	竅	竅	竅	竅	竅
e290	竦	謁	謁	筧	筧	筧	筧	筧	筧	筧	筧	筧	筧	筧	筧	筧
e2a0	笄	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍
e2b0	箠	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍	筍
e2c0	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠	箠
e2d0	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞	簞
e2e0	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝
e2f0	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝	糝

e340	紂	紘	紕	紊	綱	紘	紕	紕	紕	紕	紕	紕	紕	紕	紕	紕	紕
e350	絮	紉	紉	經	綉	條	紕	紕	紕	紕	紕	紕	紕	紕	紕	紕	紕
e360	總	網	絢	縣	綸	絛	絛	絛	絛	絛	絛	絛	絛	絛	絛	絛	絛
e370	縣	綽	絳	縱	縟	縟	縟	縟	縟	縟	縟	縟	縟	縟	縟	縟	縟
e380	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵
e390	辦	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵	縵
e3a0	嬰	嬰	嬰	罐	网	罕	罔	罔	罔	罔	罔	罔	罔	罔	罔	罔	罔
e3b0	羈	羈	羈	羌	羔	羞	羈	羈	羈	羈	羈	羈	羈	羈	羈	羈	羈
e3c0	翅	翠	翊	翕	翕	翕	翕	翕	翕	翕	翕	翕	翕	翕	翕	翕	翕
e3d0	耙	耜	耒	耒	耒	耒	耒	耒	耒	耒	耒	耒	耒	耒	耒	耒	耒
e3e0	聰	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶	聶
e3f0	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱	肱
e440	隋	腴	脾	腓	腑	胼	胼	胼	胼	胼	胼	胼	胼	胼	胼	胼	胼
e450	膠	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈	膈
e460	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍	臍
e470	舊	舍	舐	舖	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩	舩
e480	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦	艦
e490	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜
e4a0	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜	苜
e4b0	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢
e4c0	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢
e4d0	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢
e4e0	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢
e4f0	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢	莢
e540	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁
e550	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁
e560	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁	蕁
e570	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪	蚪
e580	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
e590	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮	蝮
e5a0	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
e5b0	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
e5c0	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟	蛟
e5d0	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊
e5e0	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊
e5f0	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊	裊
e640	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦	襦
e650	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲	覲
e660	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣
e670	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣
e680	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣
e690	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣	誣
e6a0	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬	譬
e6b0	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰	踰
e6c0	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊	猊



e6d0	賻	贄	贄	贄	贄	贄	贄	贄	贄	贄	贄	贄	贄	赧	赧	赧
e6e0	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧
e6f0	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧	赧
e740	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇
e750	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇
e760	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇	蹇
e770	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻
e780	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻	輻
e790	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳
e7a0	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳	迳
e7b0	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁	邁
e7c0	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈	酈
e7d0	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪	醪
e7e0	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞	鈞
e7f0	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜	銜
e840	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙
e850	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙	錙
e860	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻
e870	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻	鐻
e880	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼	閼
e890	關	關	關	關	關	關	關	關	關	關	關	關	關	關	關	關
e8a0	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟	陟
e8b0	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳	佳
e8c0	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏	霏
e8d0	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺	靺
e8e0	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞	鞞
e8f0	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡	頡
e940	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛	顛
e950	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘	餘
e960	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒	饒
e970	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱
e980	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱	駱
e990	體	體	體	體	體	體	體	體	體	體	體	體	體	體	體	體
e9a0	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆	鬆
e9b0	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏	魏
e9c0	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨
e9d0	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨	鯨
e9e0	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲	鰲
e9f0	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩	鳩
ea40	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠	鵠
ea50	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓
ea60	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓	鷓
ea70	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩	麩
ea80	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽	徽

ea90 𪗇 𪗈 𪗉 𪗊 𪗋 𪗌 𪗍 𪗎 𪗏 𪗐 𪗑 𪗒 𪗓 𪗔 𪗕 𪗖 𪗗
eaa0 榧 遙 瑤 凜 熙

[メ モ]

付録 C 総合索引

C.1 50音で始まる語句の索引

【あ】

アプリケーション処理フロー ... 25
アルファベットの大文字/小文字の区別 ... 23

【い】

インストラクション ... 67

【え】

エラー ... 61
エラー・コード ... 34

【お】

音声生成処理部 ... 87
音声生成部 ... 16, 24
音声生成部の動作 ... 28
音声データ ... 17
音素データ ... 17, 43

【か】

外部インタフェース ... 35
拡張発音情報 ... 17
カナ表記発音記号とローマ字表記発音記号の対応 ... 79
関数仕様 ... 30

【き】

記号コードの読みについて ... 89
記号読みのモード指定 ... 22

【く】

区切り文字コード ... 37

【こ】

構造体 ... 30
コマンド ... 59

【さ】

サポート・ツール ... 20
サンプル・プログラムの作成 ... 74

【し】

システム概要 ... 18
システム構成 ... 16
システム辞書 ... 16
出力データ ... 60
シンボル名規約 ... 76

【す】

数字読みのモード指定 ... 23
スピーチ・デザイナー対応機能 ... 63
スペース, TAB, 改行の完全無視... 23

【せ】

性能 ... 20

【て】

提供形態 ... 67
定数(参照用) ... 30
ディレクトリとファイル ... 70
テキスト解析部 ... 19, 22, 83

【と】

登録単語データ ... 17

【に】

入力データ ... 17
入力フォーマット ... 59

【は】

発音記号インタフェース ... 77

発音記号データ ... 17

バッチ・モード ... 29

【ひ】

ビープ音生成機能 ... 65

必要メモリ ... 19

品詞コード ... 51, 60

【ほ】

補助発音記号 ... 81

ホスト・マシンへのファイル展開 ... 72

【み】

ミドルウェア ... 15

【ゆ】

ユーザ・OWN・コーディング関数 ... 29, 76

ユーザ辞書 ... 17

ユーザ辞書管理部 ... 17, 21

ユーザ辞書作成時の注意事項 ... 88

ユーザ辞書の作成ツール ... 59

【ら】

ライブラリ概要 ... 25

【り】

リアルタイム・モード ... 28

【ろ】

ロ - マ字表記発音記号とかな文字の対応 ... 77

ロケーションの変更 ... 75

C.2 アルファベットで始まる語句の索引

- 【A】**
Accent ... 32
- 【B】**
bin ... 71
BufEnd ... 32
BufStart ... 32
- 【C】**
Character ... 32
CurrentSize ... 32
- 【D】**
DicArea ... 31
DicState ... 31
- 【E】**
E_CHARNUM ... 34
E_COMERR ... 34
E_COMSEQ ... 34
E_DATA ... 34
E_DIC ... 34
E_DICFORMAT ... 34
E_DONE ... 34
E_DUPLI ... 34
E_FILEOPEN ... 61
E_HINSI ... 61
E_HYOUSOU ... 61
E_MEMOVER ... 34, 61
E_NORUN ... 34
E_OVERFLOW ... 34
E_OVERWRIT ... 34
E_PARAMETER ... 34
E_PARVAL ... 34
E_RTOS ... 34
E_RTOSPOLFAIL ... 34
E_RTOSTMOUT ... 34
E_SUCCESS ... 34
E_TERMINATE ... 34
E_TIMEOUT ... 34
E_WORDNUM ... 61
E_WORDOVER ... 34
E_YOMI ... 61
- 【F】**
flg_ID ... 31
- 【G】**
ghstools ... 70
- 【I】**
inc830 ... 70
InitBlockCount ... 32
- 【L】**
lib830 ... 70
- 【M】**
mwc_ClrFlg 関数 ... 55
mwc_CreFlg 関数 ... 54
mwc_DelFlg 関数 ... 54
mwc_PolFlg 関数 ... 56
mwc_SetFlg 関数 ... 54
MWCVERSION ... 33
mwc_WaiFlg 関数 ... 55
- 【N】**
nectools ... 70
NextData ... 31
NextText ... 30
- 【O】**
OS 資源 ... 19
Others ... 33

【P】

PhoneData ... 30, 31
 PhoneDataSize ... 30
 PhoneInfo ... 31
 Pitch ... 32

【R】

ReserveArea ... 31

【S】

Serial ... 33
 smp830/tts ... 71
 smp830/tts/rx830v1 , smp830/tts/rx830v3 ... 71
 smp830/tts/single ... 71
 src/mwc ... 70
 src/tts/usrown ... 71
 Speed ... 32
 Status ... 32

【T】

TextIn ... 30
 tts_AnalyseText 関数 ... 35
 tts_ANALYSEWORK ... 30
 tts_BLOCKSIZE ... 30
 tts_ClearStopSynthe 関数 ... 47
 tts_DataReady 関数 ... 53
 tts_DataReady2 関数 ... 53
 TTSDIC ... 30
 tts_DicEntry 関数 ... 50
 tts_DicInitialize 関数 ... 49
 tts_GetVersion 関数 ... 52
 tts_Initialize 関数 ... 39
 TTSMEM ... 31
 TTSOSINFO ... 31
 TTSPHNBUF ... 30
 TTSPHON ... 31
 tts_RequestWave 関数 ... 48
 TTS SJIS コード表 ... 102

tts_StopAnalyse 関数 ... 38
 tts_StopSynthe 関数 ... 46
 tts_SyntheWave 関数 ... 41
 tts_SYNTHWORK ... 30
 TTSTEXT ... 30
 TTS (Text To Speech) ... 15
 tts_Uninitialize 関数 ... 40
 TTSVOICE ... 32
 TTSVOIMOD ... 32

【U】

UNIX 版 (CGMT) ... 72

【V】

Version ... 33

【W】

Windows 版 ... 73
 WorkArea ... 31

[メ モ]

— お問い合わせ先 —

【技術的なお問い合わせ先】

N E C 半導体テクニカルホットライン（インフォメーションセンター）
 （電話：午前 9:00～12:00，午後 1:00～5:00）

電話 : 044-548-8899
 FAX : 044-548-7900
 E-mail : s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

半導体第一販売事業部 半導体第二販売事業部 半導体第三販売事業部	〒108-8001 東京都港区芝5-7-1（日本電気本社ビル）	(03)3454-1111			
中部支社 半導体第一販売部 半導体第二販売部	〒460-8525 愛知県名古屋市中区錦1-17-1（日本電気中部ビル）	(052)222-2170 (052)222-2190			
関西支社 半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540-8551 大阪府大阪市中央区城見1-4-24（日本電気関西ビル）	(06)6945-3178 (06)6945-3200 (06)6945-3208			
北海道支社 東北支社 岩手支店 郡山支店 いわき支店 長岡支店 水戸支店 土浦支店 群馬支店 太田支店	札幌 (011)251-5599 仙台 (022)267-8740 盛岡 (019)651-4344 郡山 (024)923-5511 いわき (0246)21-5511 長岡 (0258)36-2155 水戸 (029)226-1717 土浦 (0298)23-6161 高崎 (027)326-1255 太田 (0276)46-4011	宇都宮支店 小山支店 甲府支店 長野支社 静岡支社 立川支社 埼玉支社 千葉支社 神奈川支社 三重支店	宇都宮 (028)621-2281 小山 (0285)24-5011 甲府 (055)224-4141 松本 (0263)35-1662 静岡 (054)254-4794 立川 (042)526-5981,6167 大宮 (048)649-1415 千葉 (043)238-8116 横浜 (045)682-4524 津 (059)225-7341	北陸支社 京都支社 神戸支社 中国支社 鳥取支店 岡山支店 松山支店 九州支社	金沢 (076)232-7303 京都 (075)344-7824 神戸 (078)333-3854 広島 (082)242-5504 鳥取 (0857)27-5311 岡山 (086)225-4455 松山 (089)945-4149 福岡 (092)261-2806

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] μSAP705100 - B08 ユーザーズ・マニュアル
(U11757JJ4V0UM00 (第4版))

[お名前など] (さしつかえのない範囲で)

御社名(学校名, その他) ()
ご住所 ()
お電話番号 ()
お仕事の内容 ()
お名前 ()

1. ご評価 (各欄に をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他 ()					
()					

2. わかりやすい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 []

3. わかりにくい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは
NEC販売員, 特約店販売員, NEC半導体ソリューション技術本部員,
その他 ()

ご協力ありがとうございました。
下記あてにFAXで送信いただくか、最寄りの販売員にコピーをお渡しください。