

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。

ユーザーズ・マニュアル

保守/廃止

μ SAP30100-B12 μ SAP705100-B12

TrueSpeech™8.5 ミドルウェア

対象デバイス

μ SAP30100-B12 : VR4100 シリーズ™

μ SAP705100-B12 : V830 ファミリ™

保守 / 廃止

[× 毛]

目次要約

第1章	概 説	…	15
第2章	ライブラリ仕様	…	23
第3章	インストール	…	27
第4章	システム例	…	33
付 録	サンプル・ソース・リスト	…	37

V830 ファミリ, V830, V831, VR4100 シリーズ, VR4100, VR4101, VR4102, VR4111 は, 日本電気株式会社
の商標です。

TrueSpeech は, 米国 DSP Group, Inc. 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。

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。

SUN4 は米国 Sun Microsystems, Inc. の商標です。

- 本資料の内容は、後日変更する場合があります。
- 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所有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。

本版で改訂された主な箇所

箇所	内容
全般	VR4100 シリーズ用 TrueSpeech8.5 ミドルウェアを追加

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・コーナーを設けております。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。

はじめに

対象者 このマニュアルは、V830 ファミリ/Vr4100 シリーズの応用システムを設計、開発するユーザーを対象としています。

目的 V830 ファミリ/Vr4100 シリーズ用の応用、開発する際にサポートするミドルウェアを、ユーザー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
- ・インストレーション
- ・システム例

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

→各製品のユーザーズ・マニュアル 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 ユーザーズ・マニュアル アーキテクチャ編を参照してください。

Vr4100 シリーズのハードウェア機能/命令機能を知りたいとき

→各製品のユーザーズ・マニュアルを参照してください。

凡 例 データ表記の重み：左が上位桁、右が下位桁

アクティブ・ロウの表記： $\overline{\text{xxx}}$ (端子、信号名称に上線)

メモリ・マップのアドレス：上部ー上位、下部ー下位

注：本文中に付けた注の説明

注意：気を付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) : $2^{20}=1024^2$

G(ギガ) : $2^{30}=1024^3$

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™	μPD705100	U11483J	U10064J	U12496J
V831™	μPD705101	U12979J	U11483J	

Vr4100 シリーズに関する資料

製品名		データ・シート	ユーザズ・マニュアル	アプリケーション・ノート
愛称	品名			
Vr4100™	μPD30100	U10428J	U10050J	プログラミング・ガイド (U10710J)
Vr4101™	μPD30101	U11846J	U12149J	
Vr4102™	μPD30102	U12543J	U12739J	
Vr4111™	μPD30111	U13211J	U13137J	

V830 ファミリ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
IE-705100-MC-EM1 (V830 用インサートキット・エミュレータ)		U11869J
IE-70000-MC-NW (V831 用インサートキット・エミュレータ)		U12476J
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
ID830 (C ソース・ディバッガ)	操作編 (UNIX ベース)	U12024J
	操作編 (Windows ベース)	U12206J
	インストレーション編 (UNIX ベース)	U12023J
RX830 (リアルタイム OS)	基礎編	U11730J
	インストレーション編	U11731J
	テクニカル編	U11713J

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)アドバンスド データ コントロールズ TEL (03)3576-5351

Cygnus Solutions のサポートするツールに関する資料

Cygnus Solutions のサポートするツールは、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。

日本シグナスソリューションズ TEL (03)3234-3896

保守 / 廃止

[× 毛]

目 次

第1章 概 説	…	15
1.1 ミドルウェア	…	15
1.2 TrueSpeech8.5	…	15
1.2.1 特 徴	…	15
1.3 システム概要	…	19
1.3.1 機 能	…	19
1.3.2 動作環境	…	19
1.3.3 ディレクトリ構成	…	21
1.3.4 性 能	…	22
第2章 ライブラリ仕様	…	23
2.1 ライブラリ概要	…	23
2.2 アプリケーションの処理フロー	…	23
2.3 関数仕様	…	24
2.3.1 外部インタフェース	…	24
第3章 インストレーション	…	27
3.1 リンク手順	…	27
3.2 サンプルの作成手順	…	28
★ 3.2.1 GHS 版 (ELF 版 : Ver.1.8.8 以上)	…	28
★ 3.2.2 GNU 版 (ELF 版 : Ver.2.7 以上)	…	30
3.3 シンボル名規約	…	31
第4章 システム例	…	33
4.1 サンプルの動作概要	…	33
4.2 メモリ・マップ例	…	34
4.2.1 メモリ・マップ例 (V830 ファミリ)	…	34
★ 4.2.2 メモリ・マップ例 (Vr4100 シリーズ)	…	35

付 録 サンプル・ソース・リスト … 37

付.1 サンプル・ソース・リスト (V830 ファミリ) … 37

★ 付.2 サンプル・ソース・リスト (VR4100 シリーズ) … 41

図の目次

図番号	タイトル, ページ
1-1	データ処理のフロー … 16
1-2	タイミング (エンコーダ) … 17
1-3	タイミング (デコーダ) … 17
1-4	ディレクトリ構成 (V830 ファミリ用) … 21
1-5	ディレクトリ構成 (VR4100 シリーズ用) … 21
2-1	アプリケーションの処理フロー例 … 23
4-1	メモリ・マップ例 (V830 ファミリ) … 34
4-2	メモリ・マップ例 (VR4100 シリーズ) … 35

表の目次

表番号	タイトル, ページ
1-1	対象 CPU … 19
1-2	必要メモリの容量 (V830 ファミリ用) … 19
1-3	必要メモリの容量 (VR4100 シリーズ用) … 20
1-4	サポート・ツール … 20
1-5	240 サンプル (30ms) に対する圧縮／伸長処理時間 … 22
2-1	関数一覧 … 23
3-1	セクション名 … 27
3-2	シンボル名 … 31
4-1	ハードウェアの仕様 (V830 ファミリ) … 33

第1章 概 説

この章では、ミドルウェアと TrueSpeech8.5 について説明します。

1.1 ミドルウェア

ミドルウェアは、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です。現在、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、V800 シリーズ/Vr4100 シリーズ用にマルチメディア・システムを実現する要素技術（たとえば、音声コーデック、画像データの圧縮/伸長といったミドルウェア）をタイムリに提供して、ユーザのシステム開発を支援します。

TrueSpeech8.5 は、音声コーデック機能を提供するミドルウェアです。

備考 RISC : Reduced Instruction Set Computer

1.2 TrueSpeech8.5

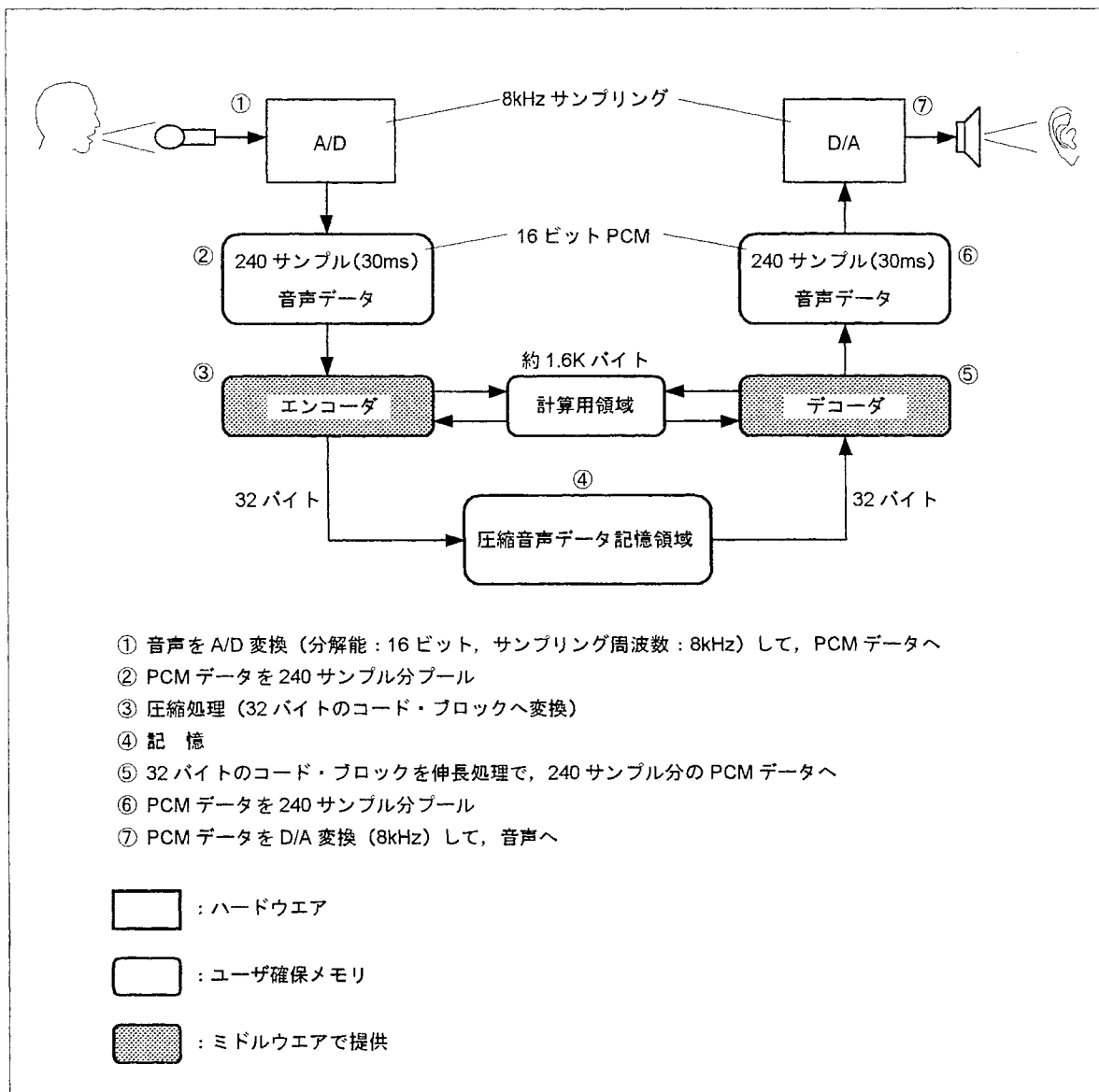
TrueSpeech8.5 は、米国 DSP Group, Inc. よりライセンス供与されている音声コーデック方式です。米国 Microsoft Corp. や米国 Netscape Communications Corp. などを始め多くのメーカーに採用されています。

圧縮データは、Windows95 Microsoft Sound Recorder で再生できます。16 ビット 8kHz でサンプリングされた PCM 音声データを 8.5Kbps で圧縮します。

1.2.1 特 徴

- 米国 DSP Group, Inc. の音声コーデック・アルゴリズムを採用
- 音声入出力データは 16 ビット・リニア・データで、8kHz でサンプリング
- 圧縮レートは 15 : 1 で、1 秒間分（128K ビット）の音声データを 8.5K ビットに圧縮
- 処理単位は、圧縮/伸長ともに 240 サンプル（30ms）単位（30ms の処理遅延が存在）
- OGHS 社製 C コンパイラの C 言語から呼び出し可能（V830 ファミリー/Vr4100 シリーズ）
- ★ ○Cygnus Solutions がサポートする C コンパイラ（gcc）の C 言語から呼び出し可能（Vr4100 シリーズのみ）
- OGHS 対応リアルタイム OS に対応（リエントラント可能）

図1-1 データ処理のフロー



(1) タイミング

図1-2 タイミング (エンコーダ)

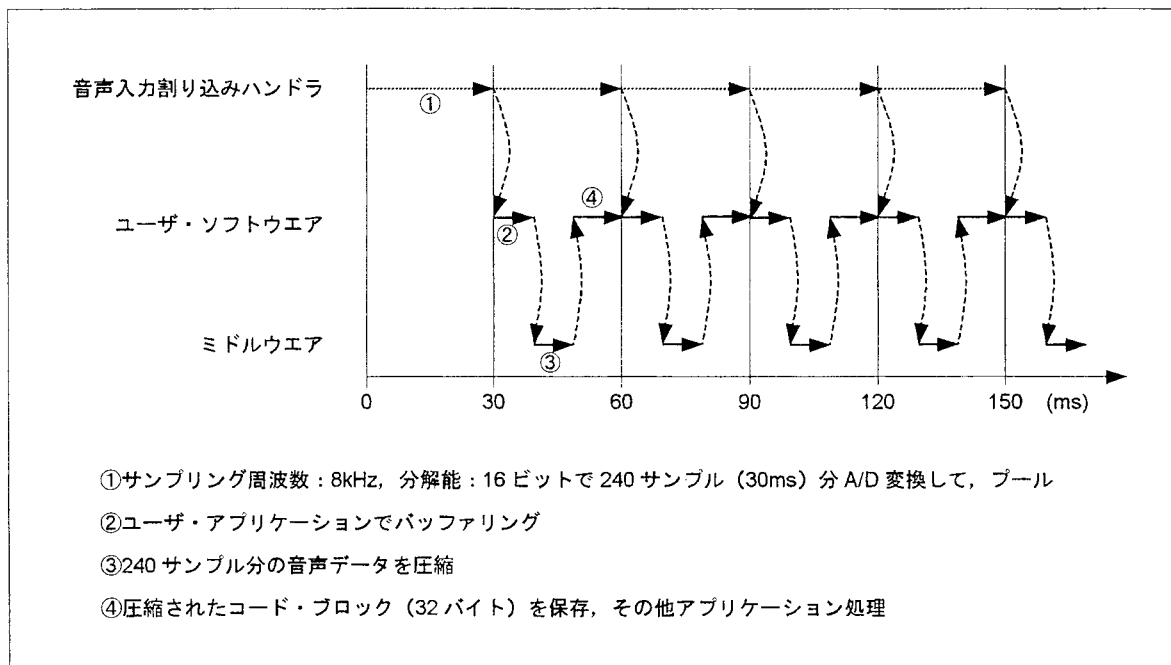
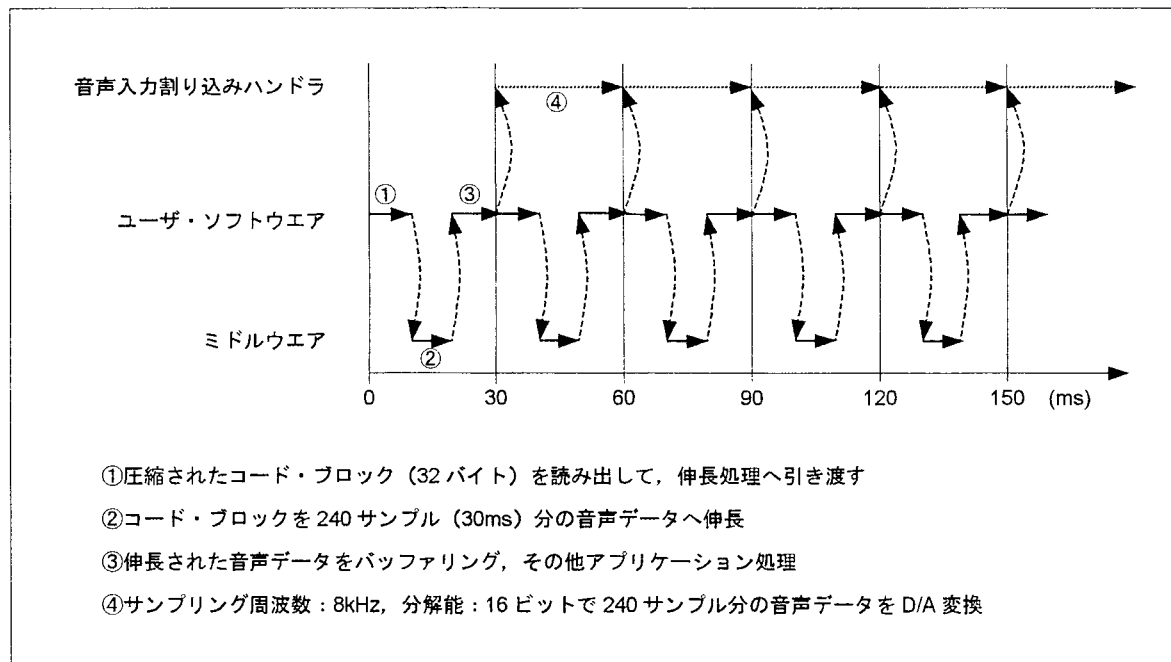
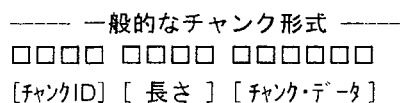


図1-3 タイミング (デコーダ)



(2) ヘッダ情報

TrueSpeech8.5 のヘッダ情報は、Windows の Wave ファイルで RIFF 形式です。RIFF (Resource Interchange File Format) は本質的にあらゆるデータを構造化形式で保存できます。



byte offset	byte size	説 明
0x0000	*4	チャンク・タイプ : "RIFF"
0x0004	*4	合計ファイル・サイズ - 8
0x0008	*4	形式名 : "WAVE"
0x000C	*4	チャンク・タイプ : "fmt "
0x0010	*4	形式チャンク・データ長 = 0x32
0x0014	50	形式チャンク・データ
	*2	wFormatTag = 0x22 (WAVE FORMAT TrueSpeech)
	*2	nChannels = 1
	*4	nSamplesPerSec = 8000
	*4	nAvgBytePerSec 1067 1 秒間のデータ量
	*2	nBlockAlign = 0x20
	*2	wBitPerSample = 1
	2	cbSize = 0x20 (size of extension area following)
	2	wRevision = 1 (For TrueSpeech8.5)
	2	nSamplesPerBlock = 240
	28	abReserved[28] (reserved area)
0x0042	4	チャンク・タイプ : "fact"
0x0046	4	形式チャンク・データ長 = 4
0x004A	4	dwSampleLength nBlock × Frame = Size_of_Wave_data
0x004E	*4	チャンク・タイプ : "data"
0x0052	*4	サウンド・データ・サイズ(bytes)
0x0056	*n	実際のサウンド・サンプリング

* : PCM Format と同じです。

注意 ハーフワード境界にワード型変数があるために、アラインメントされたアドレスしかアクセスできない CPU 上でヘッダ解析関数などを作成する場合は、ハーフワード単位でアクセスするようにしてください。

1.3 システム概要

1.3.1 機 能

(1) 圧縮処理

8kHz でサンプリングした音声データ（16 ビット・リニア・データ：240 サンプル分）を入力して、32 バイトの圧縮データに変換します。

(2) 伸長処理

32 バイトの圧縮データを、音声データ（16 ビット・リニア・データ）に変換します。

1.3.2 動作環境

★ (1) 対象 CPU

表 1-1 対象 CPU

対象 CPU	動作モード	エンディアン
V830 ファミリ	—	リトル・エンディアン
Vr4100 シリーズ	32 ビット・モード	

★ (2) 必要メモリ

表 1-2 必要メモリの容量 (V830 ファミリ用)

メモリ	用 途	サイズ	
ROM	プログラム	約 20K バイト	
	テーブル	約 2.5K バイト	
RAM	入出力音声データ・バッファ	480 バイト (240 サンプル分)	
	コード・ブロック	32 バイト (240 サンプル分)	
	変数領域	圧縮処理	約 2.5K バイト
		伸長処理	約 1K バイト
	スタック領域		約 256 バイト
計算用領域		約 1.6K バイト (圧縮/伸長処理共通)	

表1-3 必要メモリの容量 (Vr4100 シリーズ用)

メモリ	用 途		サイズ
ROM	プログラム	GHS 版	約 25K バイト
		GNU 版	約 20K バイト
	テーブル		約 1.6K バイト
RAM	入出力音声データ・バッファ		480 バイト (240 サンプル分)
	コード・ブロック		32 バイト (240 サンプル分)
	変数領域	圧縮処理	約 2.5K バイト
		伸長処理	約 1K バイト
	スタック領域		約 256 バイト
	計算用領域		約 1.6K バイト (圧縮/伸長処理共通)

(3) 必要 A/D スペック, D/A スペック

A/D : 1 チャンネル, D/A : 1 チャンネル, 16 ビット分解能, 8kHz サンプルング周波数

★ (4) サポート・ツール

表1-4 サポート・ツール

対象 CPU	サポート・ツール		
V830 ファミリ	GHS 社製 C コンパイラ/アセンブラ	CC830 (Windows 版, SUN4™ 版)	Ver.1.8.8 以上
Vr4100 シリーズ	GHS 社製 C コンパイラ/アセンブラ	CC-MIPS (Windows 版, SUN4 版)	Ver.1.8.8 以上
	Cygnus Solutions がサポートする Vr4100 シリーズ用 (ELF 形式) C コンパイラ/アセンブラ	gcc (Windows 版, SUN4 版)	Ver.2.7 以上

★ 1.3.3 ディレクトリ構成

V830 ファミリー用のディレクトリ構成を図 1-4 に、Vr4100 シリーズ用のディレクトリ構成を図 1-5 に示します。

図 1-4 ディレクトリ構成 (V830 ファミリー用)

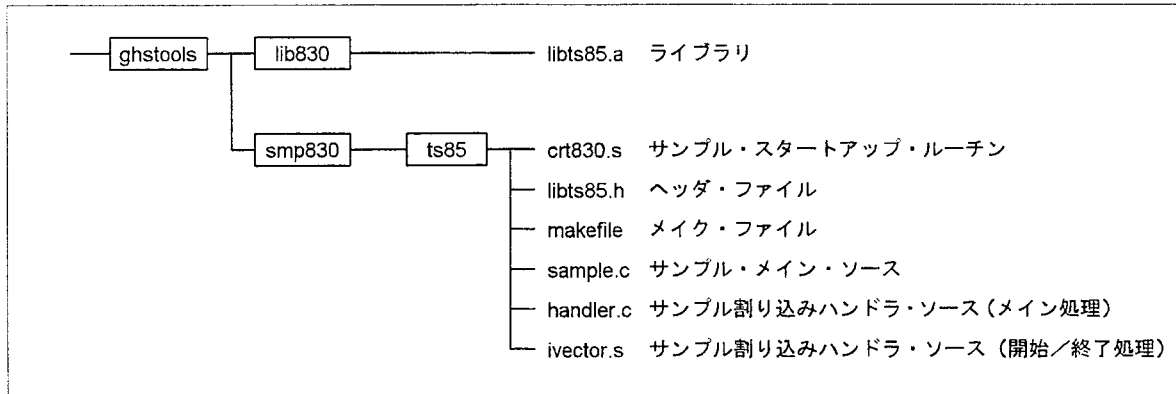
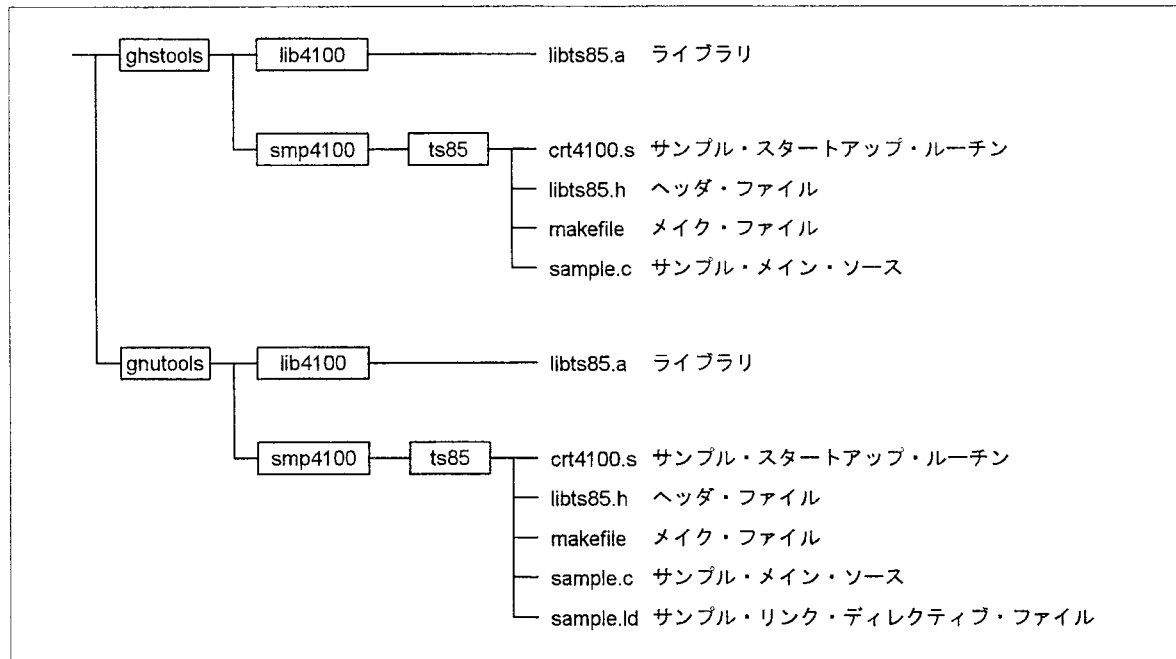


図 1-5 ディレクトリ構成 (Vr4100 シリーズ用)



(1) ghstools/ gnutools

GHS 社製ツール/Cygnus Solutions がサポートするツールを使用して、アプリケーションを開発する場合の TrueSpeech8.5 プログラムを格納しています。

(2) lib830, lib4100

TrueSpeech8.5 のライブラリを格納しています。

(3) smp830/ts85, smp4100/ts85

サンプル・プログラムのソース・ファイルを格納しています。サンプル・プログラムについてはユーザの動作環境に依存するため、保証することができません。

★ 1.3.4 性 能

240 サンプル (30ms) に対する圧縮/伸長処理時間を表 1-5 に示します。

表 1-5 240 サンプル (30ms) に対する圧縮/伸長処理時間

CPU	条 件	コンパイラ	性 能	
			圧縮処理時間	伸長処理時間
V830	100MHz (外部: 33MHz) 32ビット・バス 命令キャッシュ: 4Kバイト 内蔵データ RAM: 4Kバイト	GHS 社製 C コンパイラ	約 8.5ms	約 1.5ms
Vr4111	80MHz (外部: 40MHz) 32ビット・バス 命令キャッシュ: 16Kバイト データ・キャッシュ: 8Kバイト	GHS 社製 C コンパイラ	約 10ms	約 1.6ms
		Cygnus Solutions がサポートする C コンパイラ	約 16ms	約 2.0ms

第2章 ライブラリ仕様

2.1 ライブラリ概要

TrueSpeech8.5 では表 2-1 に示す関数があります。

表 2-1 関数一覧

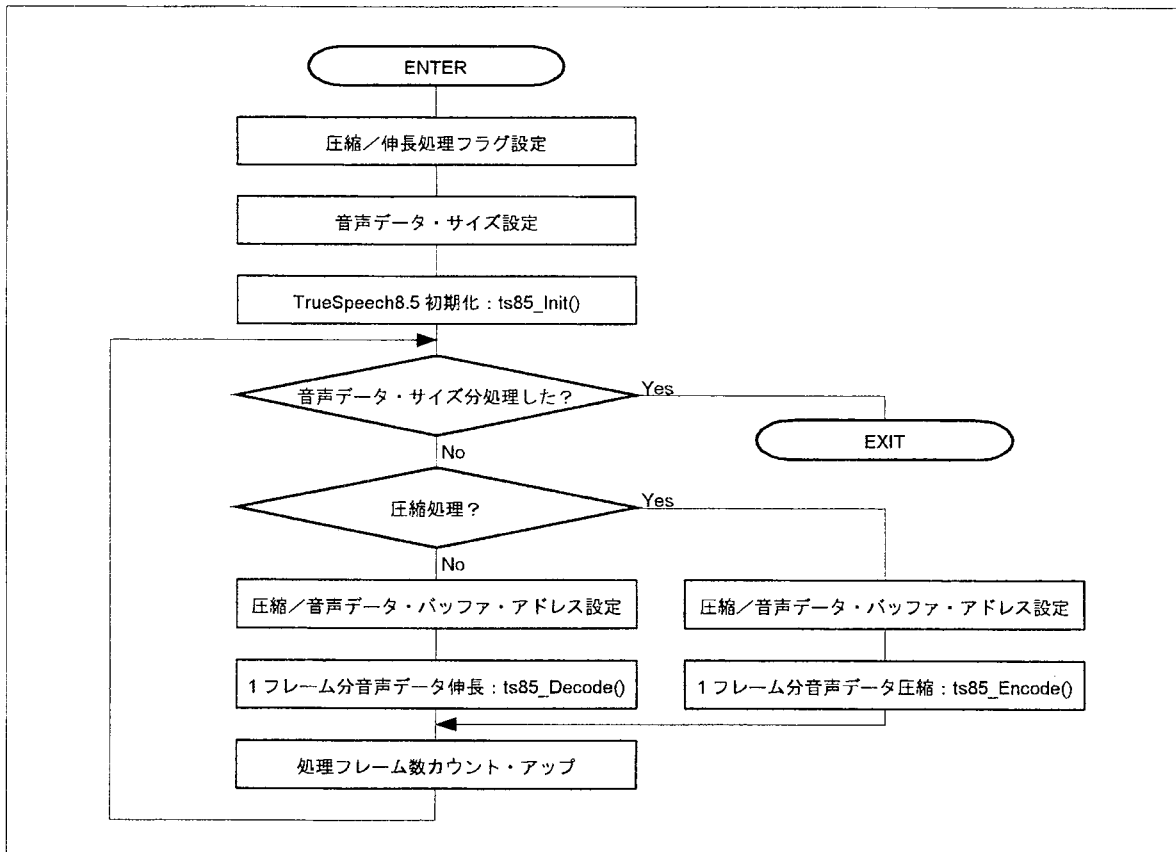
関数名	機 能
ts85_Init()	TrueSpeech8.5 初期化
ts85_Encode()	圧縮処理
ts85_Decode()	伸長処理

2.2 アプリケーションの処理フロー

TrueSpeech8.5 を使用したアプリケーションの処理フロー例を図 2-1 に示します。

割り込みハンドラの音声データ入出力処理部は、ターゲット・システムのハードウェアに依存する処理なので、ユーザがターゲット・システムにあわせて設計してください。

図 2-1 アプリケーションの処理フロー例



2.3 関数仕様

各ライブラリを呼び出す際の仕様（C 言語記述）を次に示します。

2.3.1 外部インタフェース

(1) ts85_Init 関数

【分 類】 TrueSpeech8.5 初期化処理

【関 数 名】 ts85_Init()

【機能概要】 TrueSpeech8.5 で使用する RAM 領域を初期化します。

【形 式】 #include "libts85.h"

```
void ts85_Init(void *StaticDataObject)
```

【引 き 数】 void *StaticDataObject

TrueSpeech8.5 圧縮/伸長処理に必要な計算領域の先頭アドレス（入力）。

ワード・アライメントしたアドレスを指定してください。

約 1.6K バイトの実サイズを確保してください。

【返 り 値】 なし

【機 能】 TrueSpeech8.5 圧縮/伸長処理に必要な計算領域として、指定された StaticDataObject より約 1.6K バイトを初期化します。ts85_Init 関数は、TrueSpeech8.5 圧縮/伸長で共通に使用します。

注意 ts85_Init 関数を呼び出したあとは、音声圧縮/伸長処理が終了するまで RAM 領域を壊さないようにしてください。壊れた場合の動作は保証できません。

(2) ts85_Encode 関数

【分 類】 TrueSpeech8.5 圧縮処理

【関 数 名】 ts85_Encode()

【機能概要】 指定された音声データ (240 サンプル分) を 32 バイトのコードに圧縮します。

【形 式】 #include "libts85.h"

```
void ts85_Encode( void *StaticDataObject, void *InFrame, void *OutCodeBlock,  
void *EncVal )
```

【引 き 数】 void *StaticDataObject : 圧縮処理に必要な計算領域の先頭アドレス

ワード・アライメントしたアドレスを指定してください。

void *InFrame : 音声データ (240 サンプル) の先頭アドレス

ハーフワード・アライメントしたアドレスを指定してください。

void *OutCodeBlock : コード・ブロック (32 バイト) の先頭アドレス

ワード・アライメントしたアドレスを指定してください。

void *EncVal : 圧縮処理変数領域の先頭アドレス

ワード・アライメントしたアドレスを指定してください。

約 2.5K バイトの実サイズを確保してください。

【返 り 値】 なし

【機 能】 *InFrame で指定した 240 サンプルの音声データに対して、*StaticDataObject 領域を計算用ワークとして、32 バイトのコード・ブロックに圧縮して、*OutCodeBlock に書き込んで返します。返り値はありません。

*EncVal は、圧縮処理変数領域として使用します。

注意 *StaticDataObject に設定するアドレスは、ts85_Init 関数で*StaticDataObject に設定したアドレスと同じ値を設定してください。違うアドレスを設定した場合の動作は保証できません。また、*EncVal に設定するアドレスは、ts85_Encode 関数を使用して圧縮処理を続ける場合、同じアドレスを設定してください。

(3) ts85_Decode 関数

【分類】 TrueSpeech8.5 伸長処理

【関数名】 ts85_Decode()

【機能概要】 指定された圧縮データ (32 バイト) を音声データ (240 サンプル分) に伸長します。

【形式】 #include "libts85.h"

```
void ts85_Decode( void *StaticDataObject, void *InCodeBlock, void *OutFrame,  
void *DecVal )
```

【引き数】 void *StaticDataObject : 伸長処理に必要な計算領域の先頭アドレス
ワード・アライメントしたアドレスを指定してください。

void *InCodeBlock : コード・ブロック (32 バイト) の先頭アドレス

void *OutFrame : 音声データ (240 サンプル) の先頭アドレス
ワード・アライメントしたアドレスを指定してください。

void *DecVal : 伸長処理変数領域の先頭アドレス
ワード・アライメントしたアドレスを指定してください。
約 1K バイトの実サイズを確保してください。

【返り値】 なし

【機能】 *InCodeBlock で指定した 32 バイトのコード・ブロックに対して、*StaticDataObject 領域を計算用ワークとして、240 サンプルの音声データに伸長して、*OutFrame に書き込んで返します。返り値はありません。

*DecVal は、伸長処理変数領域として使用します。

注意 *StaticDataObject に設定するアドレスは、ts85_Init 関数で*StaticDataObject に設定したアドレスと同じ値を設定してください。違うアドレスを設定した場合の動作は保証できません。また、*DecVal に設定するアドレスは、ts85_Decode 関数を使用して伸長処理を続ける場合、同じアドレスを設定してください。

第 3 章 インストレーション

3.1 リンク手順

このライブラリ内で使用しているセクション名を表 3-1 に示します。

★

表 3-1 セクション名

セクション名	属 性	内 容
.ts85TEXT	text	プログラム
.ts85DATA	data	テーブル

リンク手順を次に示します。

GHS 版 (ELF 版 : Ver.1.8.8 以上) (V830 ファミリ/Vr4100 シリーズ共通)

```
lx -o <出力ファイル> -sec <マップ・ファイル> <..オブジェクト・ファイル> -L<dir> -lts85
```

★

GNU 版 (ELF 版 : Ver.2.7 以上) (Vr4100 シリーズ)

```
ld -EL -o <出力ファイル> -T <マップ・ファイル> <..オブジェクト・ファイル> -L<dir> -lts85
```

注意 コンパイラの仕様変更 / バージョン・アップなどにより、オプション・コマンド・ラインなどが変更になる場合があります。

3.2 サンプルの作成手順

★ 3.2.1 GHS 版 (ELF 版 : Ver.1.8.8 以上)

(1) V830 ファミリ

次に GHS 版 (V830 ファミリ) の makefile の実行を示します。

```
CC = ccv830
AS = as800
LD = lx

OBJECTS = crt830.o ivector.o sample.o handler.o

sample.elf: $(OBJECTS) makefile
    $(LD) -o sample.elf -e __start -sec { .text 0x10000000 : .ts85TEXT :
    .syscall : .itext 0xfe000010 : .ts85DATA 0x10007000 : .data : .sdata :
    .sbss : .bss : .rodata } $(OBJECTS) -L../lib830 -lts85 -lansi -lsys
    -M

sample.o: sample.c makefile
    $(CC) -c -OA -ansi -G sample.c

handler.o: handler.c makefile
    $(CC) -c -OA -ansi -G handler.c

crt830.o: crt830.s
    $(AS) -elf -cpu=V830 -o crt830.o crt830.s

ivector.o: ivector.s
    $(AS) -elf -cpu=V830 -o ivector.o ivector.s
```

(a) リンカのオプション

-o<ファイル名>

生成される実行ファイル名の指定を行います。

-sec { セクション アドレス [: セクション アドレス...] }

セクション (.text, .data, ...) の先頭アドレスを設定します。

各セクションの指定は “:” で区切ります。

アドレスを省略した場合、直前に指定したセクションに連続します。

(b) サンプル・メイン・ソースのコンパイル

最適化オプション `-OA` を指定して、コンパイルを行っています。

```
例  ccv830 -c -OA -G sample.c
```

└─ Multi C ソース・ディバッガ
└─ 最適化 (アルゴリズム)
└─ コンパイルのみ

詳細は GHS のリンク、コンパイラのユーザーズ・マニュアルを参照してください。

(2) Vr4100 シリーズ

次に GHS 版 (Vr4100 シリーズ) の makefile の実行を示します。

```
CC = ccmipel
AS = asmips -elf -cpu=r4100
LD = lx
LIB2 = ../../lib4100
OBJECTS = crt4100.o sample.o
sample.elf: $(OBJECTS) makefile
    $(LD) -o sample.elf -e __start -sec {.text 0x80020000 : .ts85TEXT :
        .syscall : .data 0x80030000 : .sdata : .bss : .sbss : .rodata :
        .ts85DATA } $(OBJECTS) -L$(LIB) -L$(LIB2) -L -lts85 -lansi -lsys
        -M
sample.o: sample.c makefile
    $(CC) -c -OA -ansi -G sample.c
crt4100.o: crt4100.s
    $(AS) -b0 -o crt4100.o crt4100.s
```

(a) リンカのオプション

`-o` <ファイル名>

生成される実行ファイル名の指定を行います。

`-sec { セクション アドレス [: セクション アドレス...] }`

セクション (.text, .data, ...) の先頭アドレスを設定します。

各セクションの指定は “:” で区切ります。

アドレスを省略した場合、直前に指定したセクションに連続します。

(b) サンプル・メイン・ソースのコンパイル

最適化オプション `-OA` を指定して、コンパイルを行っています。

```
例 ccmipel -c -OA -G sample.c
```

Multi C ソース・ディバッガ
最適化 (アルゴリズム)
コンパイルのみ

詳細は GHS のリンク、コンパイラのユーザーズ・マニュアルを参照してください。

(c) make 時の注意

make を実行するときは、GHS のライブラリである `libansi.a` が存在するディレクトリを指定してください。

例 `make LIB = (libansi.a があるディレクトリ)`

★ 3.2.2 GNU 版 (ELF 版 : Ver.2.7 以上)

(1) Vr4100 シリーズ

次に GNU 版 (Vr4100 シリーズのみ) の makefile の実行を示します。

```
CC = gcc -mcpu=r4100 -EL
AS = as -mcpu=r4100 -EL
LD = ld -EL
LIB2 = ../../lib4100
OBJECTS = crt4100.o sample.o
sample.elf: $(OBJECTS) makefile
$(LD) -Map sample.map -T sample.ld -o ts85.elf $(OBJECTS) ¥
-L$(LIB2) -lts85 -L$(LIB) -lc

sample.o: sample.c makefile
$(CC) -c -ansi -g sample.c
crt4100.o: crt4100.s
$(AS) -o crt4100.o crt4100.s
```

(a) リンカのオプション

`-o<ファイル名>`

生成される実行ファイル名の指定を行います。

-Map<ファイル名>

ファイル名の指定によりマップ・ファイルが作成されます。

-T<ファイル名>

リンク・ディレクティブ・ファイルの指定を行います。

(b) サンプル・メイン・ソースのコンパイル

```
例 gcc -mcpu=r4100 -EL -c -ansi -g sample.c
```

デバッグ・オプション

リトル・エンディアンの指定

CPUの指定

詳細は GNU のリンカ, コンパイラのユーザズ・マニュアルを参照してください。

(c) make 時の注意

make を実行するときは, gcc のライブラリである libc.a が存在するディレクトリを指定してください。

例 make LIB = (libc.a があるディレクトリ)

3.3 シンボル名規約

このライブラリで使用するシンボル名などは, 次に示す規約に従っています。ユーザがアプリケーション内でシンボルなどを定義する際は, 重複しないように注意してください。

表 3-2 シンボル名

分類	規約
関数	先頭に "ts85_" を付けています。
変数	先頭に "ts85_" を付けています。

[メ 毛]

第 4 章 システム例

この章では TrueSpeech8.5 によるシステム例を示します。システム例のソースについては付録 サンプル・ソース・リストを参照してください。

★

4.1 サンプルの動作概要

外部変数 (record_flag) の設定により、20 秒分の音声データを圧縮/伸長します。処理フローについては、2.2 アプリケーションの処理フローを参照してください。

V830 ファミリの TrueSpeech8.5 によるシステム例では、表 4-1 に示すハードウェアの仕様を想定しています。

表 4-1 ハードウェアの仕様 (V830 ファミリ)

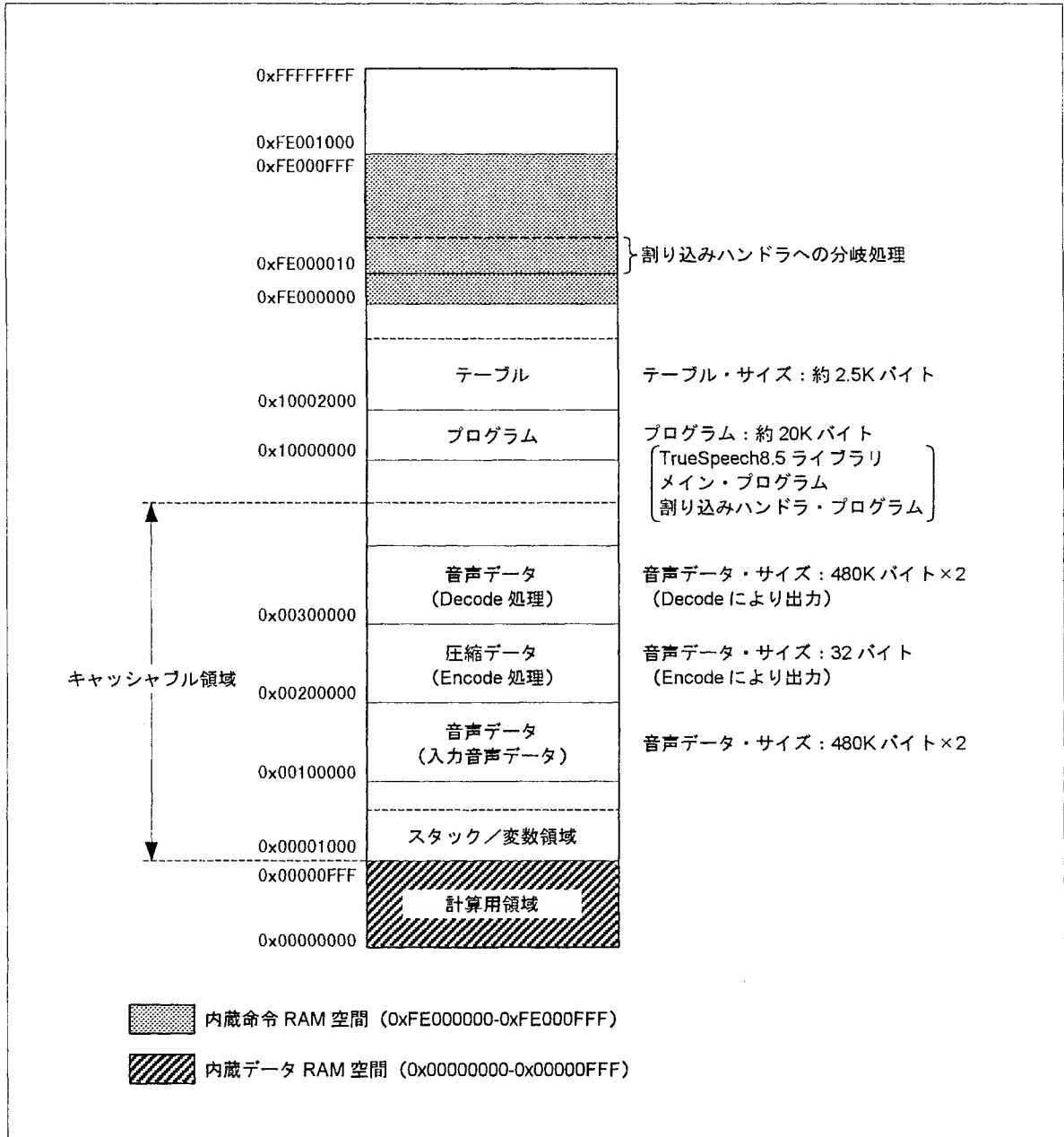
I/O アドレス	IN/OUT	機 能
0x10044	OUT	ラッチ・クリア ビット1 0 レベル1 (A/D 割り込み)
0x10048	OUT	割り込みマスク ビット1 0 レベル1 (A/D 割り込み)
0x10050	IN/OUT	下位 16 ビット 右チャンネル A/D 変換データ (音声入力データ) 右チャンネル D/A 変換データ (音声出力データ)
0x10054	OUT	0x04 出力で 8kHz の A/D サンプルング設定

4.2 メモリ・マップ例

4.2.1 メモリ・マップ例 (V830 ファミリ)

V830 ファミリ用サンプル・プログラムのメモリ・マップ例を図 4-1 に示します。

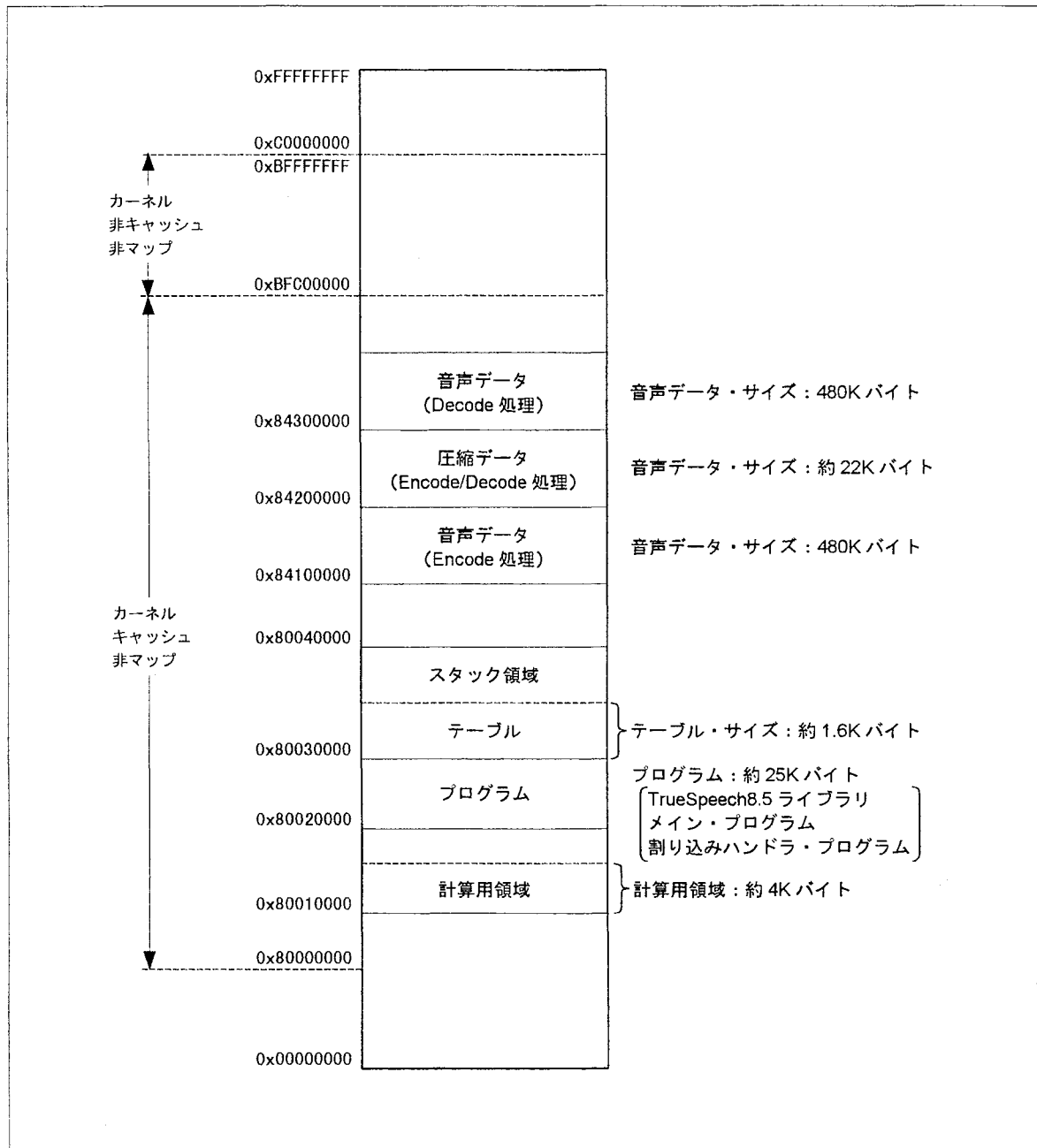
図 4-1 メモリ・マップ例 (V830 ファミリ)



★ 4.2.2 メモリ・マップ例 (Vr4100 シリーズ)

Vr4100 シリーズ用サンプル・プログラムのメモリ・マップ例を図4-2に示します。

図4-2 メモリ・マップ例 (Vr4100 シリーズ)



[× 毛]

★ 付 録 サンプル・ソース・リスト

付. 1 サンプル・ソース・リスト (V830 ファミリ)

(1) サンプル・メイン・ソース (sample.c)

```
/* TrueSpeech8.5 sample main source (main routine) */

#include <stdlib.h>
#include <string.h>
#include "libts85.h"

#define E_R_BUFF 0x100000 /* Encode Read Buff Address */
#define E_W_BUFF 0x200000 /* Encode Write Buff Address */
#define D_R_BUFF 0x200000 /* Decode Read Buff Address */
#define D_W_BUFF 0x300000 /* Decode Write Buff Address */
#define WORK_AREA 0x2000
#define ENC_AREA 0x25F0 /* WORK_AREA + 0x5F0 */
#define DEC_AREA 0x25F0 /* WORK_AREA + 0x5F0 */
#define SEC 15 /* For Record Time [sec] */
#define ON 1
#define OFF 0
#define True 1
#define False 0
#define Frame 240

void set_para();
void please_wait();
void set_time();
int byou;

unsigned int record_flag;

void main( void )
{
    int *CodeBlock;
    short *Data;
    long FrameCount=0; /* Counter */
    long amari;
    long data_size;

    /* Set Parameter */
    set_para(); /* ON: Encode / OFF: Decode */
    set_time();
    please_wait();

    /* Loop Count cal */
    data_size = 8000 * SEC;
    amari = data_size % 240;
```

```
data_size =( data_size - amari) / 240;

/* Initialize encoder/decoder */
ts85_Init((void *)WORK_AREA);

/* MAIN LOOP */

while(FrameCount < data_size)
{
    if( record_flag != ON ) /* For Decode */
    {
        CodeBlock = (int*)(D_R_BUFF + FrameCount * 32 );
        Data = (short*)(D_W_BUFF + FrameCount * Frame * 2);
        ts85_Decode
            ((void *)WORK_AREA, (void*)CodeBlock, (void*)Data ,
            (void *)DEC_AREA);
    }
    else /* For Encode */
    {
        CodeBlock = (int*)(E_W_BUFF + FrameCount * 32 );
        Data = (short*)(E_R_BUFF + FrameCount * Frame * 2);
        ts85_Encode
            ((void *)WORK_AREA, (void*)Data, (void*)CodeBlock ,
            (void *)ENC_AREA);
    }
    FrameCount++;
}

void set_para()
{
    record_flag = ON; /* ON: Encode / OFF: Decode */
    return;
}

void please_wait()
{
    return;
}

void set_time()
{
    byou = SEC ;
    return;
}
```


(2) 割り込みハンドラ・サンプル・ソース (handler.c)

```
/* TrueSpeech8.5 sample source (ADC interrupt routine) */

/* Hardware */
#define ADCFIFO      0x10050
#define ADCFREQ     0x10054
#define EDGECLR     0x10044
#define INTMASK     0x10048

#define ON          1

extern short *out_buf;
extern short *out_buf_max;
extern unsigned int record_flag;          /* Output Data flag */
extern unsigned int end_flag;

void outpw( unsigned long adr, unsigned char data)
{
    __asm("out.w r7,0[r6]");
}

void outpb( unsigned long adr, unsigned long data )
{
    __asm("out.b r7,0[r6]");
}

void outph( unsigned long adr, unsigned short data )
{
    __asm("out.h r7,0[r6]");
}

long inpw( unsigned long adr )
{
    __asm("in.w 0[r6],r10");
}
```

```
void voice_handler()
{
    long data;
    short out_data;

    if( out_buf < out_buf_max )
    {
        if( record_flag != ON )    /* Play */
        {
            out_data = *out_buf;
            outph( ADCFIFO , out_data );
        }
        else
        {
            data = inpw(ADCFIFO);
            *out_buf =(short)data & 0xffff;
        }
    }
    else
    {
        end_flag = ON;
    }

    out_buf++;

    return;
}

void init_handler()
{
    __asm("mov    1, r1");
    __asm("ldsr  r1, 31");           /* INT1 to 0xfe000010 */
    outpw( ADCFREQ, 0x04);          /* 8kHz sampling */
    outpb( INTMASK, 0x02 );        /* INT1 enable */
    outpb( EDGECLR, 0x1d );        /* INT1 edge clear */
}
```

★ 付. 2 サンプル・ソース・リスト (VR4100 シリーズ)

(1) サンプル・メイン・ソース (sample.c)

```

/*-----*/
/*      File Information      <TrueSpeech8.5>      */
/*-----*/
/*      Name      : sample.c      */
/*      Type      : C Programming Language Header  */
/*      Version   : V1.00      */
/*      Date      : 1997 Dec 15      */
/*      CPU       : VR4100 Series ( VR411x )      */
/*      Compiler  : Green Hills Software, Inc / Cygnus Solutions gcc      */
/*      About     : NEC VR4100 Series Middle-Ware Common Header      */
/*-----*/
/*      Copyright(C) NEC Corporation 1997 , 1998      */
/*      NEC CONFIDENTIAL AND PROPRIETARY      */
/*      All rights reserved by NEC Corporation.      */
/*      Use of copyright notice does not evidence publication      */
/*-----*/

#include <stdio.h>
#include <stdlib.h>
#include "libts85.h"

void set_para();
void please_wait();
void set_time();
int byou;
unsigned int record_flag;

void main( void )
{
    int *CodeBlock;
    short *Data;
    long FrameCount=0;          /* Counter */
    long data_size;

    /* Set Parameter */
    set_para();                /* ON: Encode / OFF: Decode */
    set_time();
    please_wait();

    /* Loop Count cal */
    data_size = 8000 * byou / 240;

    /* Initialize encoder/decoder */
    ts85_Init((void *)WORK_AREA);

```

```
/* MAIN LOOP */

while(FrameCount < data_size)
{
    if( record_flag != ON ) /* For Decode */
    {
        CodeBlock = (int*)(D_R_BUFF + FrameCount * 32 );
        Data = (short*)(D_W_BUFF + FrameCount * Frame * 2);

        ts85_Decode
            ((void *)WORK_AREA, (void*)CodeBlock, (void*)Data ,
            (void *)DEC_AREA);
    }
    else /* For Encode */
    {
        CodeBlock = (int*)(E_W_BUFF + FrameCount * 32 );
        Data = (short*)(E_R_BUFF + FrameCount * Frame * 2);
        ts85_Encode
            ((void *)WORK_AREA, (void*)Data, (void*)CodeBlock ,
            (void *)ENC_AREA);
    }
    FrameCount++;
    please_wait();
}

void set_para()
{
    record_flag = ON; /* ON: Encode / OFF: Decode */
    return;
}

void please_wait()
{
    return;
}

void set_time()
{
    byou = SEC ;
    return;
}
```

(2) サンプル・ヘッダ (libts85.h)

```
/*-----*/
/*      File Information      <TrueSpeech8.5>      */
/*-----*/
/*      Name      : libts85.h      */
/*      Type      : C Programming Language Header  */
/*      Version   : V1.00 (E1.00a)  */
/*      Date      : 1997 Oct 22    */
/*      CPU       : VR4100 Series ( VR41xx )      */
/*      Compiler  : Green Hills Software, Inc / Cygnus Solutions gcc  */
/*      About     : NEC VR4100 Series TrueSpeech8.5  */
/*-----*/
/*      Copyright(C) NEC Corporation 1997      */
/*      NEC CONFIDENTIAL AND PROPRIETARY      */
/*      All rights reserved by NEC Corporation.  */
/*      Use of copyright notice does not evidence publication  */
/*-----*/

/* Echo Cancellor sample source (ADC interrupt routine) */
#define ON      1
#define OFF     0
#define True    1
#define False   0

#define BUFF_SIZE      16      /* 1Encode = 16 * 2 Byte */
#define Frame          240     /* 1Frame = 240 Sample */
#define Frame_W        480     /* 1Frame = 480 Byte */
#define SEC             20     /* For Record Time [sec] */

/* TrueSpeech And Voice Data Address */
#define E_R_BUFF 0x84100000 /* Encode Read Buff Address */
#define E_W_BUFF 0x84200000 /* Encode Write Buff Address */
#define D_R_BUFF 0x84200000 /* Decode Read Buff Address */
#define D_W_BUFF 0x84300000 /* Decode Write Buff Address */

/* TrueSpeech Work Area */

#define WORK_AREA 0x80010000
#define ENC_AREA 0x800105F0 /* WORK_AREA + 0x5F0 */
#define DEC_AREA 0x800105F0 /* WORK_AREA + 0x5F0 */
```

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC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 : 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	〒108-8001	東京都港区芝5-7-1	(日本電気本社ビル)				(03)3454-1111	
半導体第三販売事業部								
中部支社	半導体第一販売部	〒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	
	半導体第二販売部						(052)222-2190	
関西支社	半導体第一販売部	〒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) 945-3178	
	半導体第二販売部						(06) 945-3200	
	半導体第三販売部						(06) 945-3208	
北海道支社	札幌	(011)231-0161	宇都宮支店	宇都宮	(028)621-2281	北陸支社	金沢	(076)232-7303
東北支社	仙台	(022)267-8740	小山支店	小山	(0285)24-5011	富山支店	富山	(0764)31-8461
岩手支店	盛岡	(019)651-4344	甲府支店	甲府	(0552)24-4141	福井支店	福井	(0776)22-1866
郡山支店	郡山	(0249)23-5511	長野支社	松本	(0263)35-1662	京都支社	京都	(075)344-7824
いわき支店	いわき	(0246)21-5511	静岡支社	静岡	(054)254-4794	神戸支社	神戸	(078)333-3854
長岡支店	長岡	(0258)36-2155	立川支社	立川	(042)526-5981,6167	中国支社	広島	(082)242-5504
水戸支店	水戸	(029)226-1717	埼玉支社	大宮	(048)649-1415	鳥取支店	鳥取	(0857)27-5311
土浦支店	土浦	(0298)23-6161	千葉支社	千葉	(043)238-8116	岡山支店	岡山	(086)225-4455
群馬支店	高崎	(027)326-1255	神奈川支社	横浜	(045)682-4524	松山支店	松山	(089)945-4149
太田支店	太田	(0276)46-4011	三重支店	津	(059)225-7341	九州支社	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] μSAP30100-B12, μSAP705100-B12 ユーザーズ・マニュアル
(U13124JJ2V0UM00 (第2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。

NEC 半導体テクニカルホットライン
FAX : (044) 548-7900

キ
リ
ト
リ

保守 / 廃止