

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、誤りがな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。

ユーザズ・マニュアル

保守/廃止

μSAP703000-B07

音声認識ミドルウェア Ver.1.1

---

対象デバイス  
V850ファミリ™

(メ モ)

## 目次要約

第1章	概説	...	15
第2章	ライブラリ仕様	...	24
第3章	インストレーション	...	65
第4章	PC版ユーザ辞書作成ツール	...	70
第5章	チェック・サム機能	...	73
第6章	ロケーションおよびシンボル名規約	...	75
付録A	サンプル・ソース(NEC版, V850E/MA1の場合)	...	76
付録B	総合索引	...	116

V850ファミリ, V850E/IA1, V850E/IA2, V850E/MA1, V850E/MA2, V850E/MS1, V850E/MS2, V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SF1, V850/SV1, V853は, 日本電気株式会社の商標です。  
MS-DOSおよび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。  
Green Hills Software, MULTI は, 米国 Green Hills Software, Inc. の商標です。

- 本資料の内容は予告なく変更することがありますので、最新のものとす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**

箇所	内容
p.18	1. 3. 4 サポート・ツールを変更
p.18	1. 3. 5 性能を変更
p.19	1. 3. 6 パッケージ内容を変更
p.24	2. 1 ライブラリ概要を変更
p.25	図 2 - 1 vrg_Recog 関数フローを変更
p.30	2. 4. 1 vrg_MemArea 変数【機能】の記述を変更
p.32	2. 4. 2 vrg_MemAreaSize 変数【機能】の記述を変更
p.33	2. 4. 3 vrg_VoiceBuf 変数を追加
p.34	2. 4. 4 vrg_BufSize 変数【機能】の記述を変更
p.35	2. 4. 5 vrg_VoiceShift 変数を追加
p.37	表 2 - 5 vrg_Recog 関数戻り値一覧を一部追加
p.39	表 2 - 7 認識結果情報一覧を一部追加
p.41	2. 5. 3 vrg_InputWaveData 関数【機能】の記述を変更
p.44	2. 5. 4 vrg_InputWaveDataDMA 関数を追加
p.49	表 2 - 8 vrg_MakeDic 関数戻り値一覧を一部追加
p.54	2. 6. 1 vrg_InputWave 関数【機能】の記述を変更
p.63	2. 6. 5 vrg_InitParam 関数を追加
p.65	第 3 章 インストレーションを変更
p.70	4. 1 動作環境を変更
p.71	図 4 - 3 辞書作成例を変更
p.72	表 4 - 3 エラー・メッセージを一部追加
p.76	付録 A サンプル・ソース (NEC 版, V850E/MA1 の場合) を変更

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、V850 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** V850 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
- ・インストレーション
- ・PC 版ユーザ辞書作成ツール
- ・チェック・サム機能
- ・ロケーションおよびシンボル名規約

- 読み方** このマニュアルでは「 $\mu$  SAP703000-B07」という製品名を「AP703000-B07」に置き換えて説明しています。

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。

V850 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。

V850 ファミリの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** **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 0xXXXX
- 2のべき数を示す接頭語(アドレス空間, メモリ容量):
- K(キ口) ...  $2^{10} = 1024$
- 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。

V850 ファミリに関する資料

品名	資料名	データ・シート	ユーザーズ・マニュアル	
			ハードウェア編	アーキテクチャ編
V853™	μ PD703003A, 703004A, 703025A, 703003A (A), 703025A (A)	U13188J	U10913J	U10243J
	μ PD70F3003A, 70F3025A, 70F3003A (A)	U13189J		
V850/SA1™	μ PD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, 703017A, 703017AY	U14526J	U12768J	
	μ PD70F3015B, 70F3015BY, 70F3017A, 70F3017AY	U14527J		
V850/SB1™	μ PD703031A, 703031AY, 703033A, 703033AY, 70F3033A, 70F3033AY	U14734J	U13850J	
	μ PD703030A, 703030AY, 703032A, 703032AY, 70F3032A, 70F3032AY	U14893J		
V850/SB2™	μ PD703034A, 703034AY, 703035A, 703035AY, 70F3035A, 70F3035AY	U14780J	U13850J	
	μ PD703036A, 703036AY, 703037A, 703037AY, 70F3037A, 70F3037AY	U14894J		
V850/SC1™, V850/SC2™, V850/SC3™	μ PD703068Y, 703069Y, 703088Y, 703089Y, 70F3089Y	作成予定	U15109J	
V850/SF1™	μ PD703078Y, 703079Y, 70F3079Y	U15183J	U14665J	
V850/SV1™	μ PD703038, 703038Y, 703039, 703039Y, 703040, 703040Y, 703041, 703041Y	U13953J	U14462J	
	μ PD70F3038, 70F3038Y, 70F3040, 70F3040Y	U14662J		
V850E/MS1™	μ PD703100-33, 703100-40, 703101-33, 703102-33	U13995J	U12688J	U12197J
	μ PD703100A-33, 703100A-40, 703101A-33, 703102A-33	U14168J		
	μ PD70F3102-33	U13844J		
	μ PD70F3102A-33	U13845J		
V850E/MS2™	μ PD703130	U15390J	U14985J	
V850E/MA1™	μ PD703106, 703107	U14792J	U14359J	U14559J
	μ PD70F3107	U14618J		
V850E/MA2™	μ PD703108	作成予定	U14980J	
V850E/IA1™	μ PD703116	作成予定	U14492J	
	μ PD70F3116, 70F3116(A), 70F3116(A1)	U15299J		
V850E/IA2™	μ PD70F3114, 703114	作成予定	U15195J	

開発ツールに関する資料 (ユーザーズ・マニュアル)

資料名	資料番号	
IE-703002-MC ( V853,V850/SA1,V850/SB1,V850/SB2,V850/SF1,V850/SV1 用インサーキット・エミュレータ )	U11595J	
IE-V850E-MC ( V850E/IA1 用インサーキット・エミュレータ ) , IE-V850E-MC-A ( V850E/MA1 用インサーキット・エミュレータ )	U14487J	
IE-703003-MC-EM1 ( V853 用インサーキット・エミュレータ・オプション・ボード )	U11596J	
IE-703017-MC-EM1 ( V850/SA1 用インサーキット・エミュレータ・オプション・ボード )	U12898J	
IE-703037-MC-EM1 ( V850/SB1,V850/SB2 用インサーキット・エミュレータ・オプション・ボード )	U14151J	
IE-703079-MC-EM1 ( V850/SF1 用インサーキット・エミュレータ・オプション・ボード )	U15447J	
IE-703040-MC-EM1 ( V850/SV1 用インサーキット・エミュレータ・オプション・ボード )	U14337J	
IE-703102-MC ( V850E/MS1 用周辺インサーキット・エミュレータ・オプション・ボード )	U13875J	
IE-703102-MC-EM1, IE-703102-MC-EM1-A ( V850E/MS1 用インサーキット・エミュレータ・オプション・ボード )	U13876J	
IE-703107-MC-EM1 ( V850E/MA1 用インサーキット・エミュレータ・オプション・ボード )	U14481J	
IE-703117-MC-EM1 ( V850E/IA1 用インサーキット・エミュレータ・オプション・ボード )	U14770J	
CA850 ( Ver.2.40 以上 ) ( C コンパイラ・パッケージ )	操作編	U15024J
	C 言語編	U15025J
	プロジェクト・マネージャ編	U15026J
	アセンブリ言語編	U15027J
ID850 ( Ver.2.40 以上 ) ( 統合デバッグ )	操作編 Windows™ ベース	U13716J
SM850 ( Ver.2.40 以上 ) ( システム・シミュレータ )	操作編 Windows ベース	U14217J
SM850 ( Ver.2.40 以上 ) ( システム・シミュレータ )	外部部品ユーザ・オープン・インタフェース仕様編	U14873J
RX850 ( Ver.3.13 以上 ) ( リアルタイム OS )	基礎編	U13430J
	インストレーション編	U13410J
	テクニカル編	U13431J
RX850 Pro ( Ver.3.13 ) ( リアルタイム OS )	基礎編	U13773J
	インストレーション編	U13774J
	テクニカル編	U13772J
RX-NET ( TCP/IP ライブラリ )		U15083J
RD850 ( Ver.3.01 ) ( タスク・デバッグ )		U13737J
RD850 Pro ( Ver.3.01 ) ( タスク・デバッグ )		U13916J
AZ850 ( Ver.3.0 ) ( システム・パフォーマンス・アナライザ )		U14410J
PG-FP3 ( フラッシュ・メモリ・プログラマ )		U13502J

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 音声認識ミドルウェア ... 15
  - 1.2.1 処理概要 ... 16
- 1.3 製品概要 ... 17
  - 1.3.1 仕 様 ... 17
  - 1.3.2 動作対象 CPU ... 17
  - 1.3.3 必要メモリ・サイズ ... 18
  - 1.3.4 サポート・ツール ... 18
  - 1.3.5 性 能 ... 18
  - 1.3.6 パッケージ内容 ... 19

## 第 2 章 ライブラリ仕様 ... 24

- 2.1 ライブラリ概要 ... 24
- 2.2 ライブラリ・ヘッダ・ファイル ... 26
  - 2.2.1 vrglib.h (音声認識ライブラリ・ヘッダ・ファイル) ... 26
  - 2.2.2 mwclib.h (ミドルウェア・ライブラリ共通ヘッダ・ファイル) ... 26
- 2.3 アプリケーション・フロー ... 27
- 2.4 変数仕様 ... 30
  - 2.4.1 vrg\_MemArea 変数 ... 30
  - 2.4.2 vrg\_MemAreaSize 変数 ... 32
  - ★ 2.4.3 vrg\_VoiceBuf 変数 ... 33
  - 2.4.4 vrg\_BufSize 変数 ... 34
  - ★ 2.4.5 vrg\_VoiceShift 変数 ... 35
- 2.5 ライブラリ関数仕様 ... 37
  - 2.5.1 vrg\_Recog 関数 ... 37
  - 2.5.2 vrg\_Stop 関数 ... 40
  - 2.5.3 vrg\_InputWaveData 関数 ... 41
  - ★ 2.5.4 vrg\_InputWaveDataDMA 関数 ... 44
  - 2.5.5 vrg\_MakeDic 関数 ... 49
  - 2.5.6 vrg\_GetVersion 関数 ... 53
- 2.6 ユーザ・OWN・コーディング関数仕様 ... 54
  - 2.6.1 vrg\_InputWave 関数 ... 54
  - 2.6.2 vrg\_IntOn 関数 / vrg\_IntOff 関数 ... 56

- 2. 6. 3 vrg\_RecogReady 関数 ... 61
- 2. 6. 4 vrg\_RecogComplete 関数 ... 62
- ★ 2. 6. 5 vrg\_InitParam 関数 ... 63

### 第 3 章 インストール ... 65

- 3. 1 開発環境へのファイル展開 ... 65
- 3. 2 サンプル・プログラム作成手順 ... 66
  - 3. 2. 1 NEC 製ツールのライブラリとプロジェクト・マネージャを使用する場合 ... 66
  - 3. 2. 2 GHS 社製ツールのライブラリと MULTI のビルダ機能を使用する場合 ... 67
- 3. 3 システム構築方法 ... 68
  - 3. 3. 1 認識単語辞書を PC 版ユーザ辞書作成ツールを使用して作成する場合 ... 68
  - 3. 3. 2 認識単語辞書を vrg\_MakeDic 関数を使用して作成する場合 ... 69

### 第 4 章 PC 版ユーザ辞書作成ツール ... 70

- 4. 1 動作環境 ... 70
- 4. 2 辞書作成方法 ... 70
  - 4. 2. 1 入力フォーマット ... 70
  - 4. 2. 2 PC 版ユーザ辞書作成ツール使用方法 ... 71
  - 4. 2. 3 出力結果 ... 72
  - 4. 2. 4 エラー・コード ... 72

### 第 5 章 チェック・サム機能 ... 73

- 5. 1 チェック・サム概要 ... 73
- 5. 2 チェック・サム使用方法 ... 73

### 第 6 章 ロケーションおよびシンボル名規約 ... 75

- 6. 1 ロケーションの変更 ... 75
- 6. 2 シンボル名規約 ... 75

### 付録 A サンプル・ソース (NEC 版 , V850E/MA1 の場合) ... 76

- A. 1 認識処理サンプル・ソース (sample1.c) ... 76
- A. 2 辞書作成サンプル・ソース (sample2.c) ... 80
- A. 3 認識処理サンプル・ソース (sample3.c) ... 85
- A. 4 スタートアップ・ルーチン (cpu\_init.s) ... 89
- A. 5 スタートアップ・ルーチン (cpu\_init\_dma.s) ... 96
- A. 6 ユーザ・OWN・コーディング・ファイル (vrg\_own.s) ... 105

A.7 ユーザ・OWN・コーディング・ファイル (vrg\_own\_dma.s) ... 109

**付録 B 総合索引** ... 116

B.1 50音で始まる語句の索引 ... 116

B.2 アルファベットで始まる語句の索引 ... 118

## 図の目次

図番号	タイトル, ページ
1 - 1	単語音声認識構成 ... 16
1 - 2	パッケージ内容 ... 19
2 - 1	vrg_Recog 関数フロー ... 25
2 - 2	PC 版辞書作成ツールなどで作成済みの認識単語辞書を 1 つのみ使用して認識を行う場合 ... 27
2 - 3	複数の認識単語辞書を切り替えながら認識を行う場合 ... 28
2 - 4	vrg_MakeDic 関数を用いて作成した認識単語辞書を認識に使用する場合 ... 29
2 - 5	vrg_VoiceShift 変数指定例 ... 35
2 - 6	アプリケーション側での音声書き込み処理 ... 44
2 - 7	DMA 機能を利用した音声入力処理フロー（メイン処理） ... 46
2 - 8	DMA 機能を利用した音声入力処理フロー（割り込み処理） ... 48
2 - 9	入力フォーマット ... 50
2 - 10	バージョン情報 ... 53
2 - 11	マスカブル割り込みのイメージ ... 56
2 - 12	メイン処理 ... 57
2 - 13	vrg_IntOn 関数発行後の処理 ... 59
2 - 14	vrg_IntOff 関数発行後の処理 ... 60
3 - 1	プロジェクト・マネージャ画面 ... 66
3 - 2	MULTI 画面 ... 67
4 - 1	入力フォーマット ... 70
4 - 2	入力ファイル例（smpdic.txt） ... 71
4 - 3	辞書作成例 ... 71

## 表の目次

表番号	タイトル, ページ
1 - 1	音声認識ミドルウェア仕様 ... 17
1 - 2	必要メモリ・サイズ ... 18
1 - 3	ライブラリのファイル一覧 (bin¥vrg850) ... 20
1 - 4	ライブラリのファイル一覧 (dat850¥vrg850) ... 20
1 - 5	ライブラリのファイル一覧 (doc) ... 21
1 - 6	ライブラリのファイル一覧 (inc850) ... 21
1 - 7	ライブラリのファイル一覧 (lib850¥r32) ... 21
1 - 8	ライブラリのファイル一覧 (lib850_ghs¥r32) ... 21
1 - 9	ライブラリのファイル一覧 (smp850¥vrg850) ... 22
1 - 10	ライブラリのファイル一覧 (smp850_ghs¥vrg850) ... 23
2 - 1	ライブラリ・グローバル変数一覧 ... 24
2 - 2	ライブラリ関数一覧 ... 24
2 - 3	ユーザ・OWN・コーディング関数一覧 ... 24
2 - 4	必要ワーク・メモリ・サイズ ... 30
2 - 5	vrg_Recog 関数戻り値一覧 ... 37
2 - 6	単語 ID 例 ... 38
2 - 7	認識結果情報一覧 ... 39
2 - 8	vrg_MakeDic 関数戻り値一覧 ... 49
2 - 9	各入力文字仕様 ... 50
2 - 10	使用可能文字一覧表 ... 51
2 - 11	辞書作成結果情報 ... 51
2 - 12	vrg_InitParam で設定するパラメータの位置 ... 63
4 - 1	各入力文字仕様 ... 70
4 - 2	出力結果 ... 72
4 - 3	エラー・メッセージ ... 72
5 - 1	check_sum 関数戻り値一覧 ... 73
6 - 1	セクション名一覧 ... 75



# 第 1 章 概 説

## 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 音声認識ミドルウェア

発声された日本語音声を認識する機能をもつミドルウェアです。

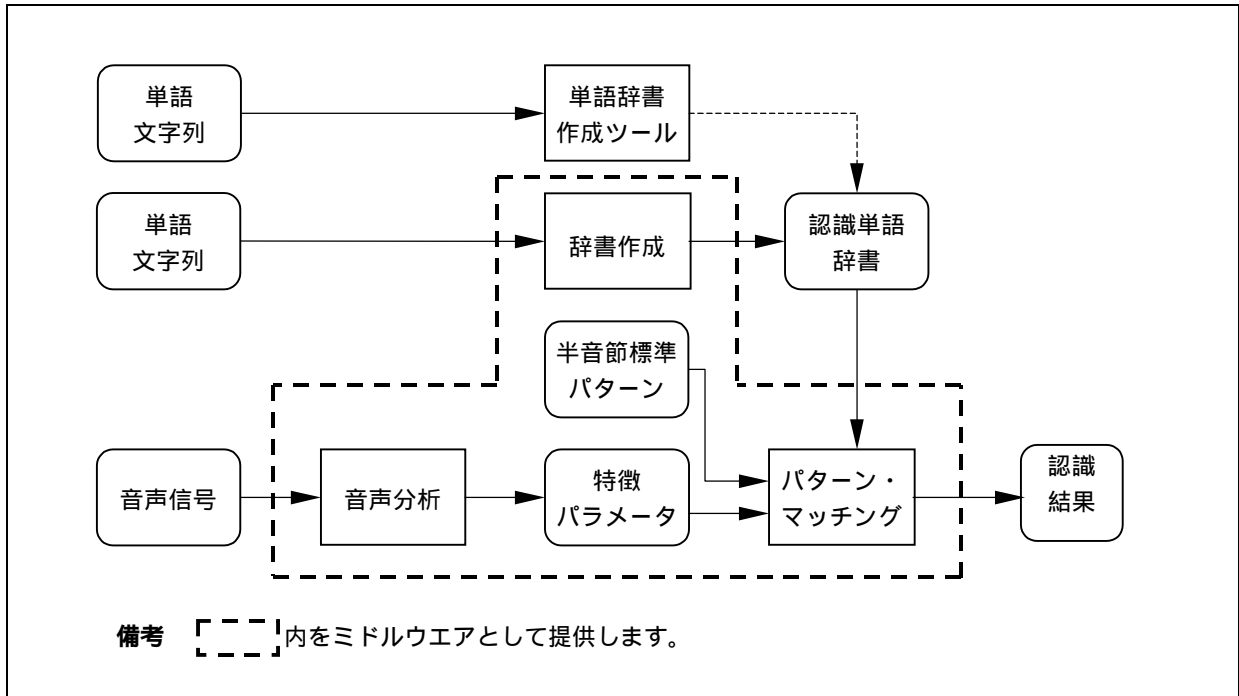
あらかじめ認識単語辞書に登録された単語の中で、発声された日本語音声に最も近いと推定された単語の単語番号を認識結果として出力します。次に特徴を示します。

- ・ 離散単語認識（単語単位で発声された言葉を認識）
- ・ 事前発声不要な不特定話者単語認識
- ・ 認識単語を半角カナ文字により登録するため、簡単に認識単語辞書が作成可能
- ・ リアルタイム（発声終了後 0.4 秒で認識結果出力）での認識処理が可能
- ・ 内蔵 A/D コンバータ（10 ビット分解能）を使用しての認識が可能

### 1.2.1 処理概要

音声認識の構成を次に示します。

図1-1 単語音声認識構成



#### (1) 認識処理

##### (a) 音声信号入力

発声した音声を PCM 信号（16 ビット，8 kHz サンプルング）として入力します。

##### (b) 音声分析

入力された音声信号について、音声分析処理をして特徴パラメータを抽出します。このとき、周囲の定常ノイズによる影響を低減させる処理を行います。

##### (c) パターン・マッチング

音声分析処理により抽出された特徴パラメータに対し、あらかじめ用意した半音節単位の標準パターンと認識単語辞書を使用してパターン・マッチングを行います。

##### (d) 認識結果出力

パターン・マッチングの結果、入力した音声に似ていると推定できる単語を認識結果とします。認識結果として単語 ID とスコアを出力します。候補を最大 3 位まで出力します。単語 ID とスコアについては、2.5.1 vrg\_Recog 関数を参照してください。

## (2) 辞書作成

音声認識処理を行うときに使用する認識単語辞書を作成します。

### (a) ミドルウェア上の辞書作成機能

単語文字列を入力することにより認識単語辞書登録を行います。入力された単語文字列は音声認識処理に必要な形式（認識単語辞書）に変換され、ユーザの確保した領域に格納されます。

### (b) 単語辞書作成ツール

テキスト・ファイルを用いて認識単語を入力し、認識単語辞書をバイナリ・ファイルとして出力します。詳細については、第4章 PC版ユーザ辞書作成ツールを参照してください。

## 1.3 製品概要

### 1.3.1 仕 様

表1-1 音声認識ミドルウェア仕様

項 目	仕 様
音声認識方式	半音節音声認識方式
認識対象言語	単語単位で発声された日本語の音声
認識対象話者	不特定話者（事前の学習，話者登録不要）
外部ノイズ対策	定常ノイズ・キャンセル
入力音声	10-16 ビット・リニア PCM サンプリング周波数：8.0 kHz 帯域：0.3-3.4 kHz 入力：1ch

### 1.3.2 動作対象 CPU

- ・ V850 ファミリ

### 1.3.3 必要メモリ・サイズ

表1-2 必要メモリ・サイズ

ROM/RAM	用 途	容 量〔バイト〕
ROM	プログラム	約 25 K
	テーブル・データ	約 62 K
ROM/RAM 両方可	認識単語辞書（認識文字数に依存） （平均 5 文字 15 単語の場合）	約 0.5 K
RAM	スタック	約 0.2 K
	ワーク（認識単語辞書に依存） （平均 5 文字 15 単語の場合）	約 3.3 K

**備考** 各メモリ容量の詳細については AP703000-B07 に添付されている資料を参照してください。

### 1.3.4 サポート・ツール

- ・ NEC 製 C コンパイラ・パッケージ

CA850

Ver.2.30 以上

- ・ GHS (Green Hills Software, Inc.) 社製 C コンパイラ・アセンブラ

ccv850, ccv850e

Ver.1.8.9 (pr0448, pr0484 を含む) 以上

**注意** リアルタイム OS とあわせての使用を検討の際は、別途お問い合わせください。

### 1.3.5 性 能

#### (1) V850E/MA1 (50 MHz 動作時) 使用時

- ・ CPU 占有率

20 単語認識時：約 35%

50 単語認識時：約 55%

- ・ 条 件

認識単語辞書に平均 5 音節の単語を使用

内蔵 ROM, 内蔵 RAM 領域を使用

#### (2) V850/SA1 (20 MHz 動作時) 使用時

- ・ CPU 占有率

20 単語認識時：約 85%

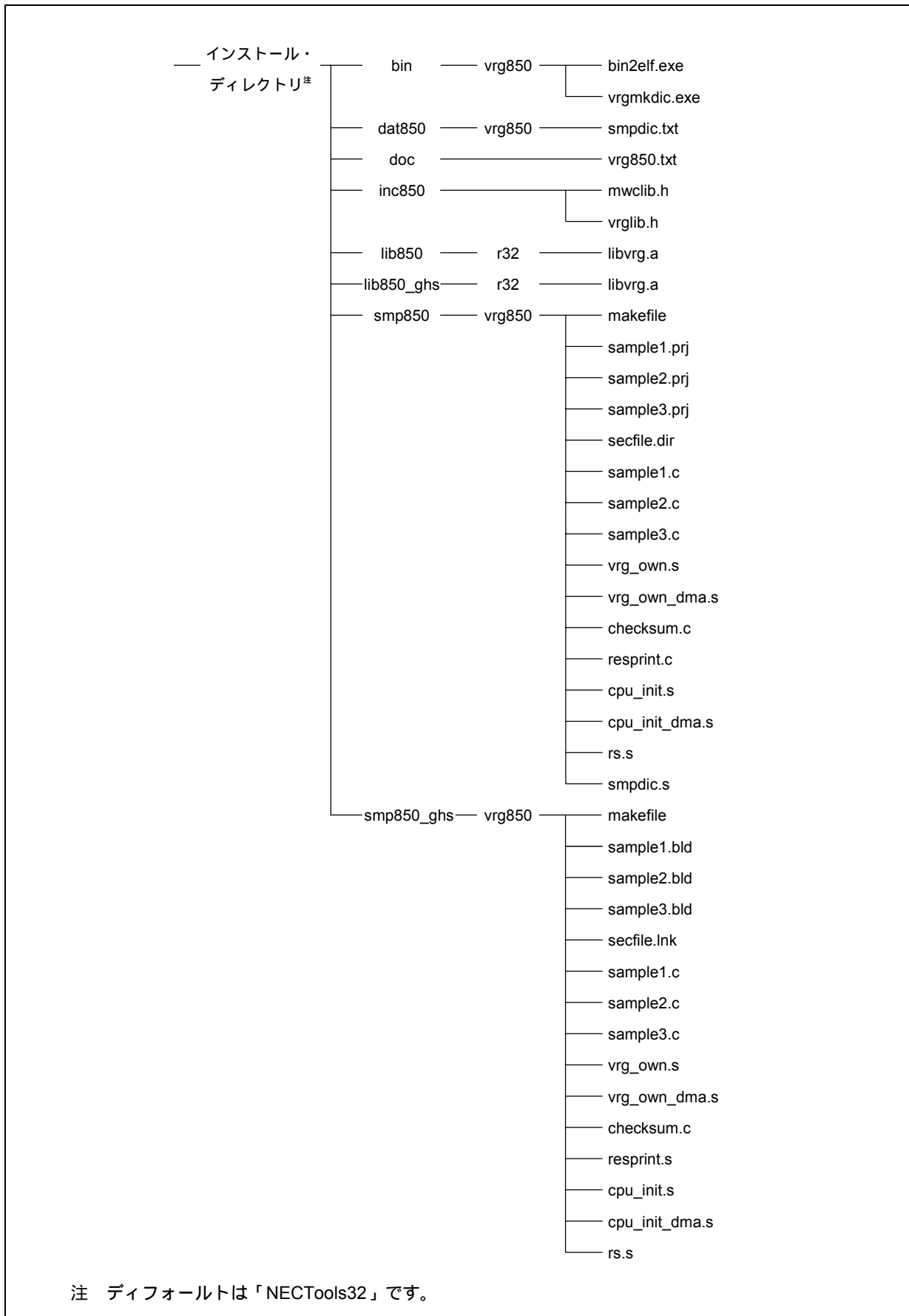
- ・ 条 件

認識単語辞書に平均 5 音節の単語を使用

内蔵 ROM, 内蔵 RAM 領域を使用

1.3.6 パッケージ内容

図1-2 パッケージ内容



次に AP703000-B07 のディレクトリとファイルの説明を示します。

(1) bin¥vrg850

AP703000-B07 が提供するユーティリティ・ツール（辞書作成ツール、バイナリ・ファイル ELF ファイル変換ツール）を格納しています。

表1-3 ライブラリのファイル一覧 (bin¥vrg850)

ディレクトリ	ファイル名	ファイルの説明
bin¥vrg850	bin2elf.exe	<p>バイナリ・ファイル ELF ファイル変換ツール (MS-DOS™ 版)</p> <p>・プログラム起動方法</p> <p>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。</p> <p>例 C:¥NECTools32¥bin¥vrg850&gt;bin2elf</p> <pre>Binary file to ELF file converter Version 1.10 [Mar 23 1999] Copyright (C) NEC Corporation 1997-1999 usage:bin2elf -s &lt;section&gt; -l &lt;label&gt; -o &lt;outfile&gt; &lt;infile&gt;</pre> <p>・オプション説明</p> <p>指定できるオプションは3種類あります。各オプションの説明を次に示します。 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。</p> <p>-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)</p> <p>例 辞書ファイル smpdic.cs1 から, smpdic.o を作成します。 セクションは .data, ラベルは _DicArea とします。</p> <pre>C:¥&gt;bin2elf -s .data -l _DicArea -o smpdic.o smpdic.cs1</pre>
	vrgmkdic.exe	<p>辞書作成ツール (MS-DOS 版)</p> <p>第4章 PC版ユーザ辞書作成ツール参照</p>

(2) dat850¥vrg850

AP703000-B07 のサンプル・プログラムで使用されるサンプル辞書テキスト・ファイルを格納しています。

次に dat850¥vrg850 に含まれるファイルを示します。

表1-4 ライブラリのファイル一覧 (dat850¥vrg850)

ディレクトリ	ファイル名	ファイルの説明
dat850¥vrg850	smpdic.txt	<p>サンプル辞書テキスト・ファイル</p> <p>サンプル・プログラムで使用する辞書の元となるテキスト・ファイルです。</p>

## (3) doc

AP703000-B07 をご使用になる上で必要となる情報を記述したテキスト・ファイルを格納しています。  
次に doc に含まれるファイルを示します。

表1 - 5 ライブラリのファイル一覧 (doc)

ディレクトリ	ファイル名	ファイルの説明
doc	vrg850.txt	ライブラリ関数が動作する上で必要となる情報を記述したテキスト・ファイル ご使用の前に必ずお読みください。

## (4) inc850

AP703000-B07 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。  
次に inc850 に含まれるファイルを示します。

表1 - 6 ライブラリのファイル一覧 (inc850)

ディレクトリ	ファイル名	ファイルの説明
inc850	mwclib.h	ミドルウェア・ライブラリ共通ヘッダ・ファイル ミドルウェア共通の構造体が記述されています。必ずインクルードしてください。
	vrglib.h	音声認識ライブラリ・ヘッダ・ファイル 音声認識ミドルウェア固有のエラー・コード・マクロ定数やライブラリ関数の プロトタイプ宣言が記述されています。必ずインクルードしてください。

## (5) lib850¥r32

AP703000-B07 の NEC コンパイラ版ライブラリを格納しています。  
次に lib850¥r32 に含まれるファイルを示します。

表1 - 7 ライブラリのファイル一覧 (lib850¥r32)

ディレクトリ	ファイル名	ファイルの説明
lib850¥r32	libvrg.a	NEC コンパイラ版音声認識ライブラリ ライブラリ関数ごとにオブジェクト化されています。

## (6) lib850\_ghs¥r32

AP703000-B07 の GHS コンパイラ版ライブラリを格納しています。  
次に lib850\_ghs¥r32 に含まれるファイルを示します。

表1 - 8 ライブラリのファイル一覧 (lib850\_ghs¥r32)

ディレクトリ	ファイル名	ファイルの説明
lib850_ghs¥r32	libvrg.a	GHS コンパイラ版音声認識ライブラリ ライブラリ関数ごとにオブジェクト化されています。

## (7) smp850¥vrg850

AP703000-B07 を使用した NEC 版サンプル・プログラムを格納しています。

smp850¥vrg850 において実行プログラム・ファイルを作成するとサンプル・プログラム ( sample1.out, sample2.out, sample3.out ) が生成されます。

次に smp850¥vrg850 に含まれるファイルを示します。

表1-9 ライブラリのファイル一覧 ( smp850¥vrg850 )

ディレクトリ	ファイル名	ファイルの説明
smp850¥vrg850	makefile	サンプル・プログラム用 make ファイル ( make コマンドを使用する場合 )
	sample1.prj	サンプル・プログラム 1 用プロジェクト・ファイル ( プロジェクト・マネージャを使用する場合 )
	sample2.prj	サンプル・プログラム 2 用プロジェクト・ファイル ( プロジェクト・マネージャを使用する場合 )
	sample3.prj	サンプル・プログラム 3 用プロジェクト・ファイル ( プロジェクト・マネージャを使用する場合 )
	secfile.dir	サンプル・プログラム用リンク・ディレクティブ・ファイル
	sample1.c	サンプル・プログラム 1
	sample2.c	サンプル・プログラム 2
	sample3.c	サンプル・プログラム 3
	vrg_own.s	ユーザ・OWN・コーディング・ファイル 音声取り込み割り込み関連のハードウェア依存部プログラムです。ユーザ・コーディングが必須です。
	vrg_own_dma.s	ユーザ・OWN・コーディング・ファイル DMA を使用する場合の音声取り込み割り込み関連のハードウェア依存部プログラムです。ユーザ・コーディングが必須です。
	checksum.c	認識単語辞書チェック・サム確認ルーチン
	resprint.c	認識結果表示ルーチン ボード上の RS-232-C を使用して PC 上のターミナル・ソフトウェアに認識結果を表示するルーチンです。
	cpu_init.s	スタートアップ・ルーチン 割り込みハンドラ・アドレス設定, スタートアップ・ルーチンを記述したファイルです。サンプル・プログラム 1, 2 で使用します。
	cpu_init_dma.s	スタートアップ・ルーチン 割り込みハンドラ・アドレス設定, スタートアップ・ルーチンを記述したファイルです。サンプル・プログラム 3 で使用します。
rs.s	認識結果表示ルーチン: ハードウェア依存部 resprint.c のハードウェア依存部分です。	
smpdic.s	認識辞書インクルード用ファイル 認識辞書バイナリファイルをサンプル・プログラムにリンクするために使用するファイルです。	



## (8) smp850\_ghs¥vrg850

AP703000-B07 を使用した GHS 版サンプル・プログラムを格納しています。

smp850\_ghs¥vrg850 において実行プログラム・ファイルを作成するとサンプル・プログラム ( sample1.out, sample2.out, sample3.out ) が生成されます。

次に smp850\_ghs¥vrg850 に含まれるファイルを示します。

表1 - 10 ライブラリのファイル一覧 ( smp850\_ghs¥vrg850 )

ディレクトリ	ファイル名	ファイルの説明
smp850_ghs¥vrg850	makefile	サンプル・プログラム用 make ファイル ( make コマンドを使用する場合 )
	sample1.bld	サンプル・プログラム 1 用ビルド・ファイル ( MULTI™ のビルダ機能を使用する場合 )
	sample2.bld	サンプル・プログラム 2 用ビルド・ファイル ( MULTI のビルダ機能を使用する場合 )
	sample3.bld	サンプル・プログラム 3 用ビルド・ファイル ( MULTI のビルダ機能を使用する場合 )
	secfile.lnk	サンプル・プログラム用リンク・ディレクティブ・ファイル
	sample1.c	サンプル・プログラム 1
	sample2.c	サンプル・プログラム 2
	sample3.c	サンプル・プログラム 3
	vrg_own.s	ユーザ・OWN・コーディング・ファイル 音声取り込み割り込み関連のハードウェア依存部プログラムです。ユーザ・コーディングが必須です。
	vrg_own_dma.s	ユーザ・OWN・コーディング・ファイル DMA を使用する場合の音声取り込み割り込み関連のハードウェア依存部プログラムです。ユーザ・コーディングが必須です。
	checksum.c	認識単語辞書チェック・サム確認ルーチン
	resprint.c	認識結果表示ルーチン ボード上の RS-232-C を使用して PC 上のターミナル・ソフトウェアに認識結果を表示するルーチンです。
	cpu_init.s	スタートアップ・ルーチン 割り込みハンドラ・アドレス設定, スタートアップ・ルーチンを記述したファイルです。サンプル・プログラム 1, 2 で使用します。
cpu_init_dma.s	スタートアップ・ルーチン 割り込みハンドラ・アドレス設定, スタートアップ・ルーチンを記述したファイルです。サンプル・プログラム 3 で使用します。	
rs.s	認識結果表示ルーチン : ハードウェア依存部 resprint.c のハードウェア依存部分です。	

## 第 2 章 ライブラリ仕様

### 2.1 ライブラリ概要

ライブラリ・グローバル変数一覧を次に示します。

**表2 - 1 ライブラリ・グローバル変数一覧**


ライブラリ・グローバル変数名	機能概要
vrg_MemArea	ワーク領域の確保を行います。
vrg_MemAreaSize	ワーク領域サイズの指定を行います。
vrg_VoiceBuf	音声バッファの確保を行います。
vrg_BufSize	音声バッファ・サイズの指定を行います。
vrg_VoiceShift	入力音声データ・シフト数を指定します。

ライブラリ関数一覧を次に示します。

**表2 - 2 ライブラリ関数一覧**

ライブラリ関数名	機能概要
vrg_Recog	認識処理を行います。
vrg_Stop	認識処理の強制終了を行います。
vrg_InputWaveData	音声データの格納処理を行います。
vrg_InputWaveDataDMA	音声データ書き込みを認識処理側に通知します。
vrg_MakeDic	辞書作成を行います。
vrg_GetVersion	AP703000-B07 のバージョン情報を出力します。

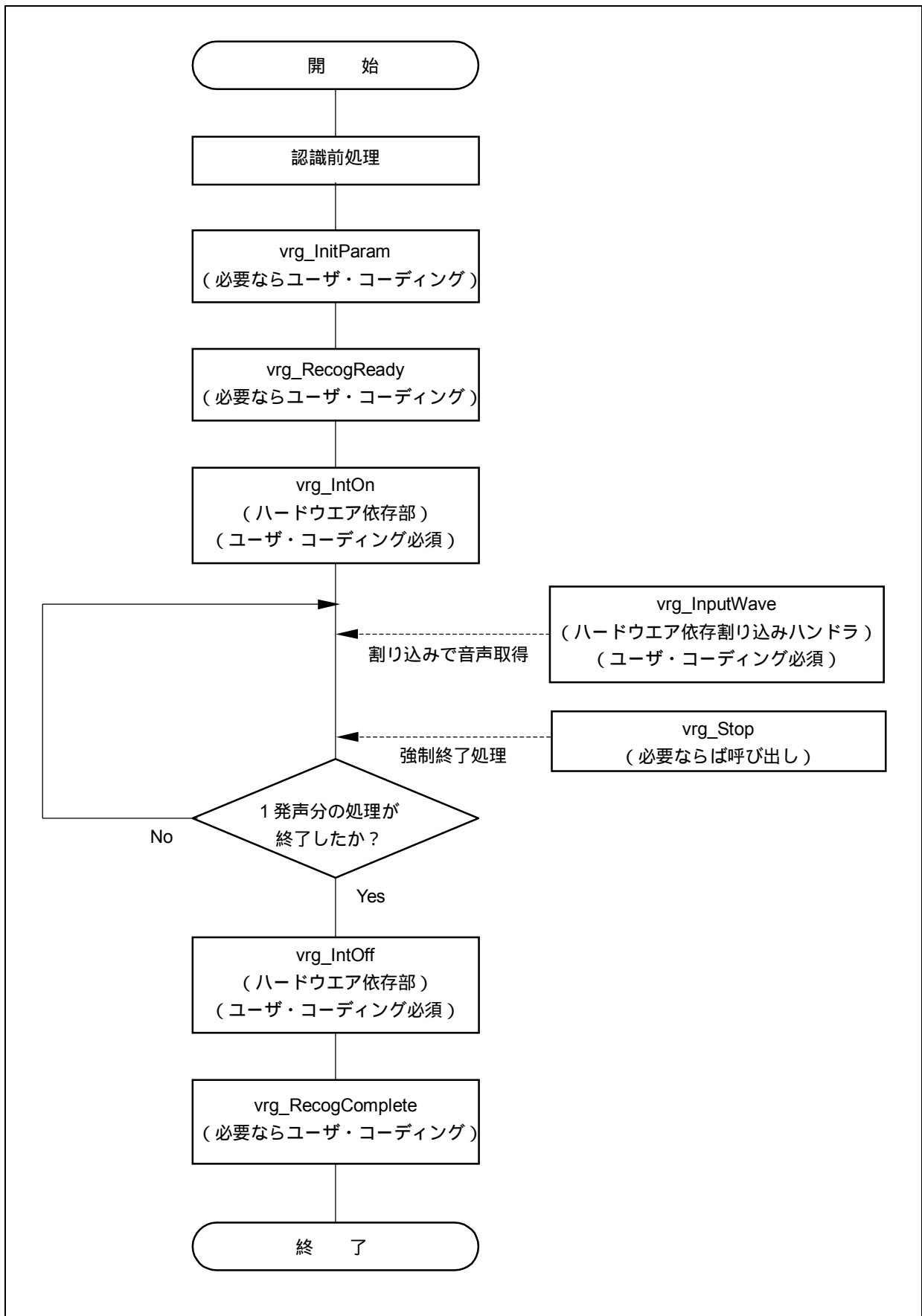
ライブラリ関数とは別に、次のユーザ・OWN・コーディング関数が必要となります。

ユーザ・OWN・コーディング関数は vrg\_Recog 関数内からコールされます。呼び出されるタイミングについては、 2 - 1 を参照してください。

**表2 - 3 ユーザ・OWN・コーディング関数一覧**

関数名	機能概要	ユーザ・コーディング
vrg_InputWave	音声入力処理を行います。	使用するハードウェア環境にあわせて、必ずサンプル・プログラムを変更してください。
vrg_IntOn	音声取り込み割り込みの許可を行います。	
vrg_IntOff	音声取り込み割り込みの禁止を行います。	
vrg_RecogReady	認識前処理終了通知を行います。	サンプル・プログラムを変更して使用できます。
vrg_RecogComplete	認識動作終了通知を行います。	
vrg_InitParam	各初期設定を行います。	必要に応じてコーディングしてください。

図2-1 vrg\_Recog関数フロー



## 2.2 ライブラリ・ヘッダ・ファイル

AP703000-B07 を使用する場合に必要なヘッダ・ファイルの説明を次に示します。

### 2.2.1 vrglib.h (音声認識ライブラリ・ヘッダ・ファイル)

AP703000-B07 固有のエラー・コード・マクロ定数やライブラリ関数のプロトタイプ宣言が記述されています。  
必ずインクルードしてください。

### 2.2.2 mwclib.h (ミドルウェア・ライブラリ共通ヘッダ・ファイル)

ミドルウェア共通の構造体が記述されています。  
必ずインクルードしてください。

## 2.3 アプリケーション・フロー

AP703000-B07 を使用したアプリケーション例を示します。

PC 版辞書作成ツール等で作成済みの認識単語辞書を 1 つのみ使用して認識を行う例を図 2 - 2 に示します。

複数の認識単語辞書を切り替えながら認識を行う場合の例を図 2 - 3 に示します。

vrg\_MakeDic 関数で認識単語辞書を作成して認識に使用する例を図 2 - 4 に示します。

図2 - 2 PC版辞書作成ツールなどで作成済みの認識単語辞書を1つのみ使用して認識を行う場合

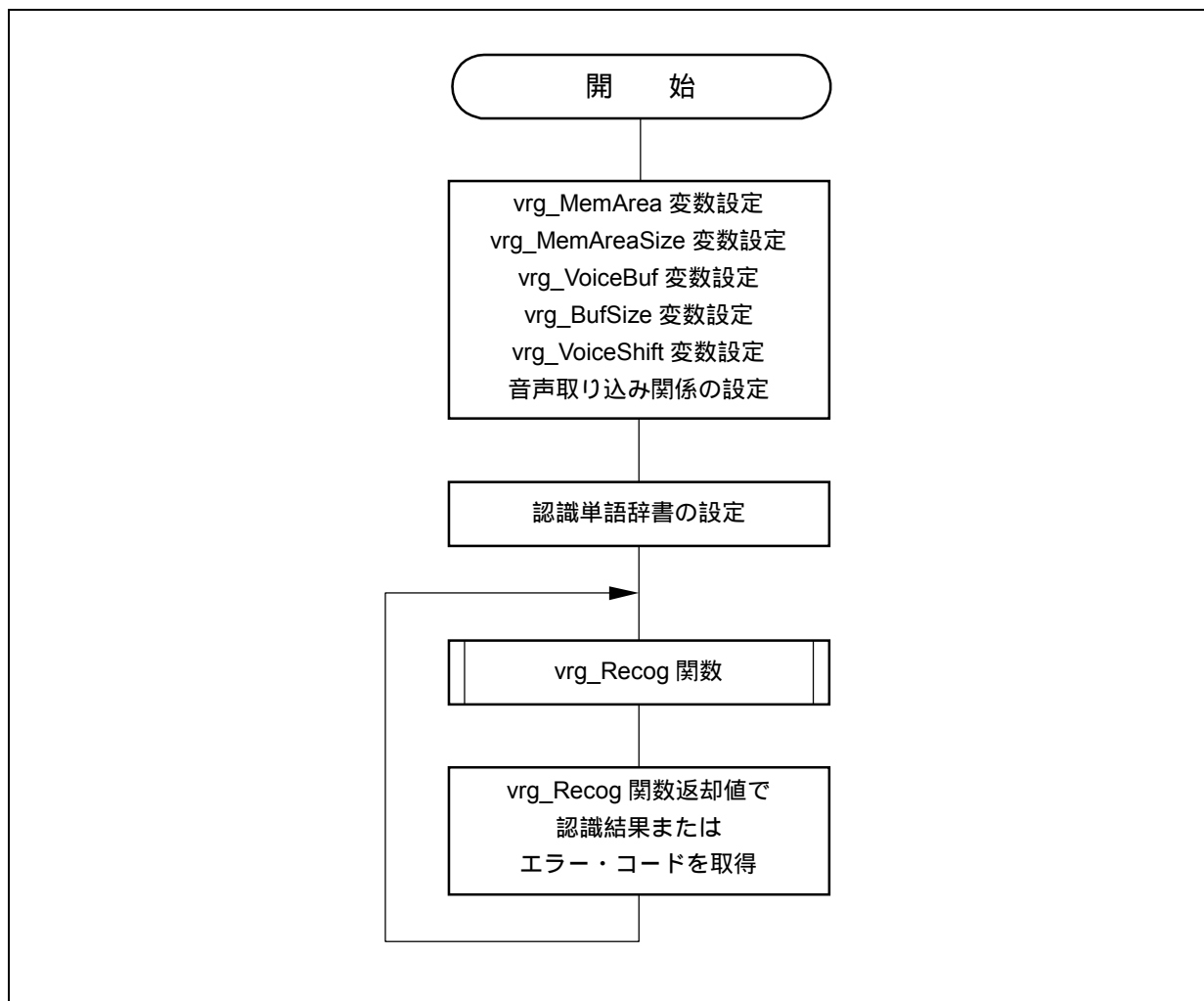


図2-3 複数の認識単語辞書を切り替えながら認識を行う場合

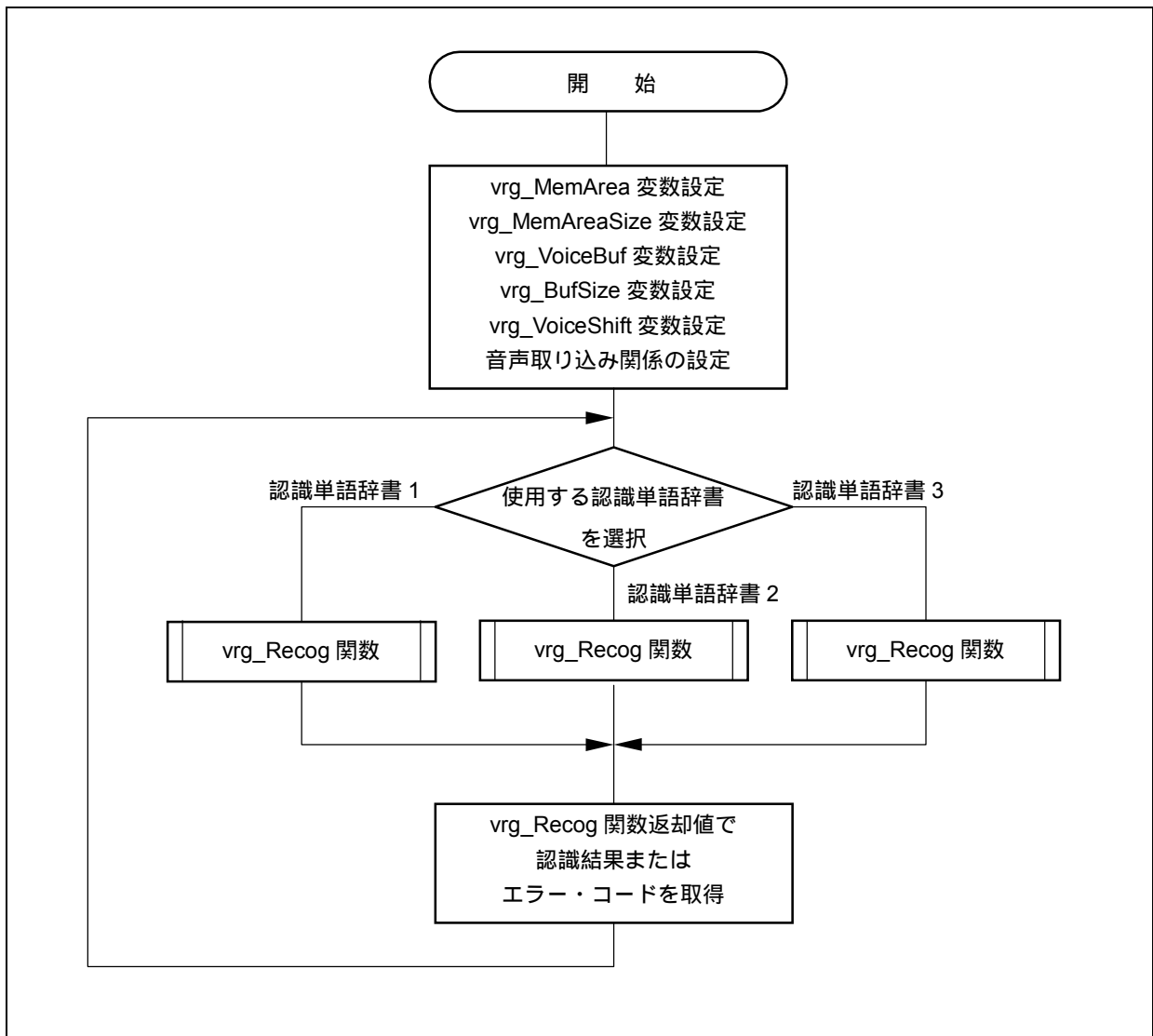
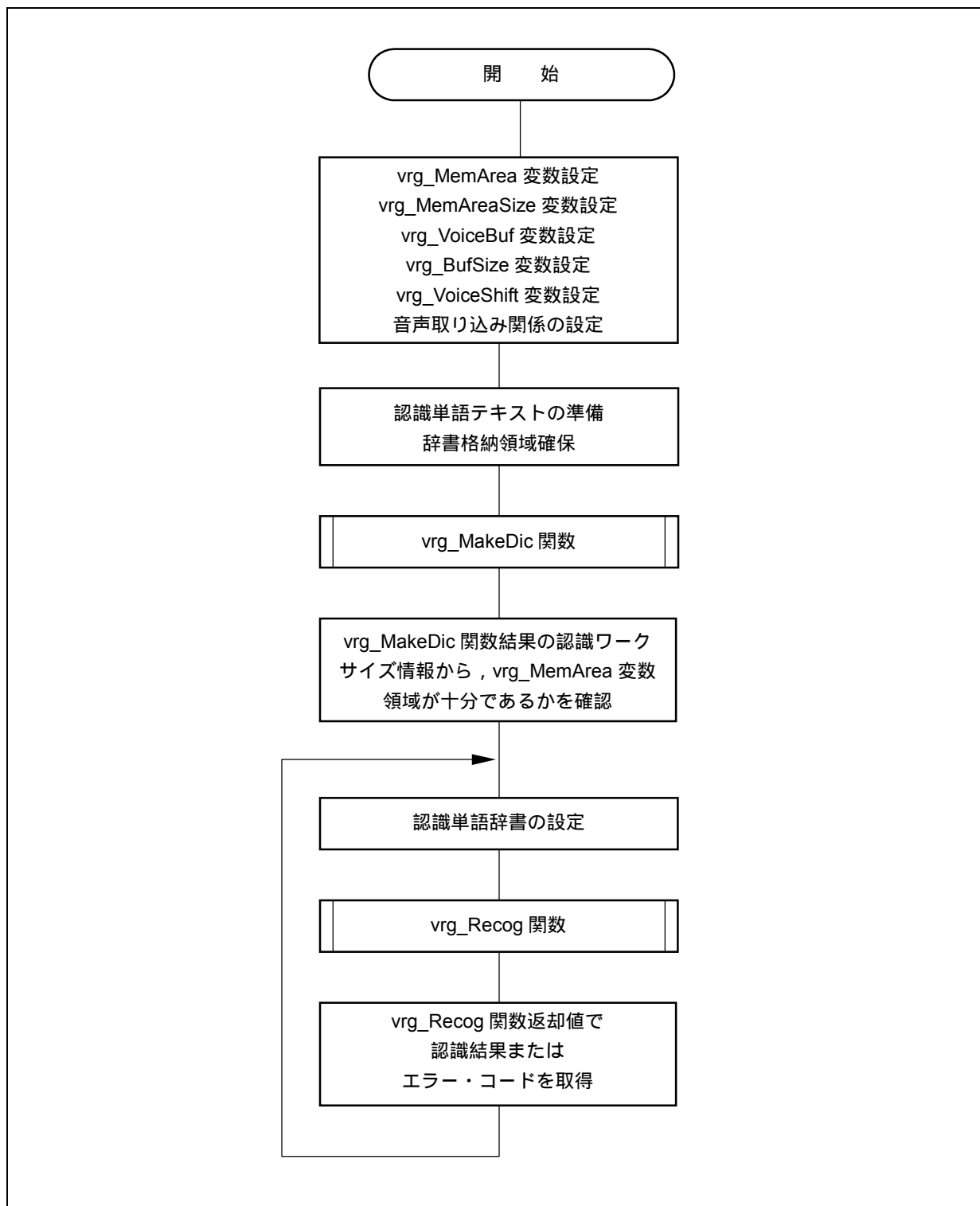


図2 - 4 vrg\_MakeDic関数を用いて作成した認識単語辞書を認識に使用する場合



## 2.4 変数仕様

### 2.4.1 vrg\_MemArea 変数

【変数名】 vrg\_MemArea

【概要】 AP703000-B07 の動作に必要なワーク領域を確保します。

【形式】 long vrg\_MemArea[VRGMEMAREASIZE/sizeof(long)];

【機能】

#### (a) 概要

AP703000-B07 の動作に必要なワーク領域を確保します。必要なワーク領域サイズは、使用する認識単語辞書によって異なります。

表2-4 必要ワーク・メモリ・サイズ

関数名	必要ワーク・メモリ・サイズ
vrg_Recog 関数	VRGMEMAREASIZE (バイト) = 認識ワーク・サイズ (バイト)
vrg_MakeDic 関数	VRGMEMAREASIZE (バイト) = 8 (バイト)

各ライブラリ関数動作中は、この領域を破壊しないよう注意してください。各ライブラリ関数が動作していない時点では、vrg\_MemArea 変数の領域を音声認識以外の処理で使用できます。

#### (b) 認識ワーク・サイズ

認識ワーク・サイズは vrg\_MakeDic 関数またはユーザ辞書作成ツールによって得られる値です。詳細は 2.5.5 vrg\_MakeDic 関数, 4.2.2 PC 版ユーザ辞書作成ツール使用方法を参照してください。



サンプル・プログラムを次に示します。

```
#define VRGMEMAREASIZE   x x x x x /* 認識ワーク・サイズ */
#define VRGBUFSIZE       x x x x x /* 音声バッファ・サイズ */
#define VRGSHIFTNUM     x x x x x /* 音声シフト数 */

long   vrg_MemArea[VRGMEMAREASIZE/sizeof(long)];
int    vrg_MemAreaSize = VRGMEMAREASIZE;
short  vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int    vrg_BufSize = VRGBUFSIZE;
int    vrg_VoiceShift = VRGSHIFTNUM;

main()
{
    :
}
```

- 注意 1. vrg\_MemArea 変数は、ユーザ・オリジナル・セクションに割り付けないでください。
2. ミドルウェア関数動作中に値を書き換えしないでください。
3. vrg\_MemArea 変数は 4 バイト・アラインであることが必須のため、long 型変数で宣言してください。

## 2.4.2 vrg\_MemAreaSize 変数

【変数名】 vrg\_MemAreaSize

【概要】 AP703000-B07 の動作に必要なワーク領域サイズを指定します。

【形式】 int vrg\_MemAreaSize;

【機能】

### (a) 動作

AP703000-B07 の動作に必要なワーク領域サイズを指定します。ここで指定される値は、vrg\_MemArea 変数の領域として確保したサイズ (VRGMEMAREASIZE) と同じ値にしてください。

サンプル・プログラムを次に示します。

```
#define VRGMEMAREASIZE    x x x x x /* 認識ワーク・サイズ */
#define VRGBUFSIZE       x x x x x /* 音声バッファ・サイズ */
#define VRGSHIFTNUM     x x x x x /* 音声シフト数 */

long   vrg_MemArea[VRGMEMAREASIZE/sizeof(long)];
int    vrg_MemAreaSize = VRGMEMAREASIZE;
short  vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int    vrg_BufSize = VRGBUFSIZE;
int    vrg_VoiceShift = VRGSHIFTNUM;

main()
{
    :
}
```

- 注意 1. vrg\_MemAreaSize 変数は、ユーザ・オリジナル・セクションに割り付けないでください。
2. ミドルウェア関数動作中に値を書き換えしないでください。
  3. vrg\_MemArea 変数は 4 バイト・アラインであることが必須のため、long 型変数で宣言します。したがって vrg\_MemAreaSize 変数に設定する値は、4 の整数倍であることが必須となります。
  4. vrg\_MemAreaSize 変数は、vrg\_MemArea 変数のサイズをライブラリに通知するために使用されます。vrg\_MemArea 変数とあわせて設定してください。

## 2.4.3 vrg\_VoiceBuf 変数

【変数名】 vrg\_VoiceBuf

【概要】 AP703000-B07 の動作に必要な音声バッファを確保します。

【形式】 short vrg\_VoiceBuf[VRGBUFSIZE/sizeof(short)];

【機能】

## (a) 動作

AP703000-B07 の動作に必要な音声バッファを確保します。ただし、256 バイトの整数倍かつ 768 バイト以上の領域を確保してください。通常は、768 バイトの領域を確保してください。

サンプル・プログラムを次に示します。

```
#define VRGMEMAREASIZE  x x x x x /* 認識ワーク・サイズ */
#define VRGBUFSIZE      768      /* 音声バッファ・サイズ */
#define VRGSHIFTNUM     x x x x x /* 音声シフト数 */

long   vrg_MemArea[VRGMEMAREASIZE/sizeof(long)];
int    vrg_MemAreaSize = VRGMEMAREASIZE;
short  vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int    vrg_BufSize = VRGBUFSIZE;
int    vrg_VoiceShift = VRGSHIFTNUM;

main()
{
    :
}
```

- 注意 1. vrg\_VoiceBuf 変数は、ユーザ・オリジナル・セクションに割り付けないでください。
2. ミドルウェア関数動作中に値を書き換えしないでください。
  3. vrg\_VoiceBuf 変数は 256 バイトの整数倍かつ 768 バイト以上のサイズであることが必須です。通常は 768 バイト確保してください。
  4. vrg\_Recog 関数 (2.5.1 参照) 以外のライブラリ関数を動作させるときも、vrg\_VoiceBuf 変数の設定は必要です。

## 2.4.4 vrg\_BufSize 変数

【変数名】 vrg\_BufSize

【概要】 AP703000-B07 の動作に必要な音声バッファ・サイズを指定します。

【形式】 int vrg\_BufSize;

【機能】

## (a) 動作

AP703000-B07 の動作に必要な音声バッファ・サイズをバイト単位の値で指定します。ここで指定される値は vrg\_VoiceBuf 変数の領域として確保したサイズ (VRGBUFSIZE) と同じ値にしてください。

サンプル・プログラムを次に示します。

```
#define VRGMEMAREASIZE    x x x x x /* 認識ワーク・サイズ */
#define VRGBUFSIZE       768      /* 音声バッファ・サイズ */
#define VRGSHIFTNUM     x x x x x /* 音声シフト数 */

long   vrg_MemArea[VRGMEMAREASIZE/sizeof(long)];
int    vrg_MemAreaSize = VRGMEMAREASIZE;
short  vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int    vrg_BufSize = VRGBUFSIZE;
int    vrg_VoiceShift = VRGSHIFTNUM;

main()
{
    :
}
```

注意 1. vrg\_BufSize 変数は、ユーザ・オリジナル・セクションに割り付けないでください。

2. ミドルウェア関数動作中に値を書き換えしないでください。

3. vrg\_BufSize 変数に設定する値は vrg\_VoiceBuf 変数の領域として確保したサイズと同じであることが必須です。

4. vrg\_BufSize 変数は、vrg\_VoiceBuf 変数のサイズをライブラリに通知するために使用されます。vrg\_VoiceBuf 変数とあわせて設定してください。

## 2.4.5 vrg\_VoiceShift 変数

【変数名】 vrg\_VoiceShift

【概要】 音声データ・シフト数を指定します。

【形式】 int vrg\_VoiceShift;

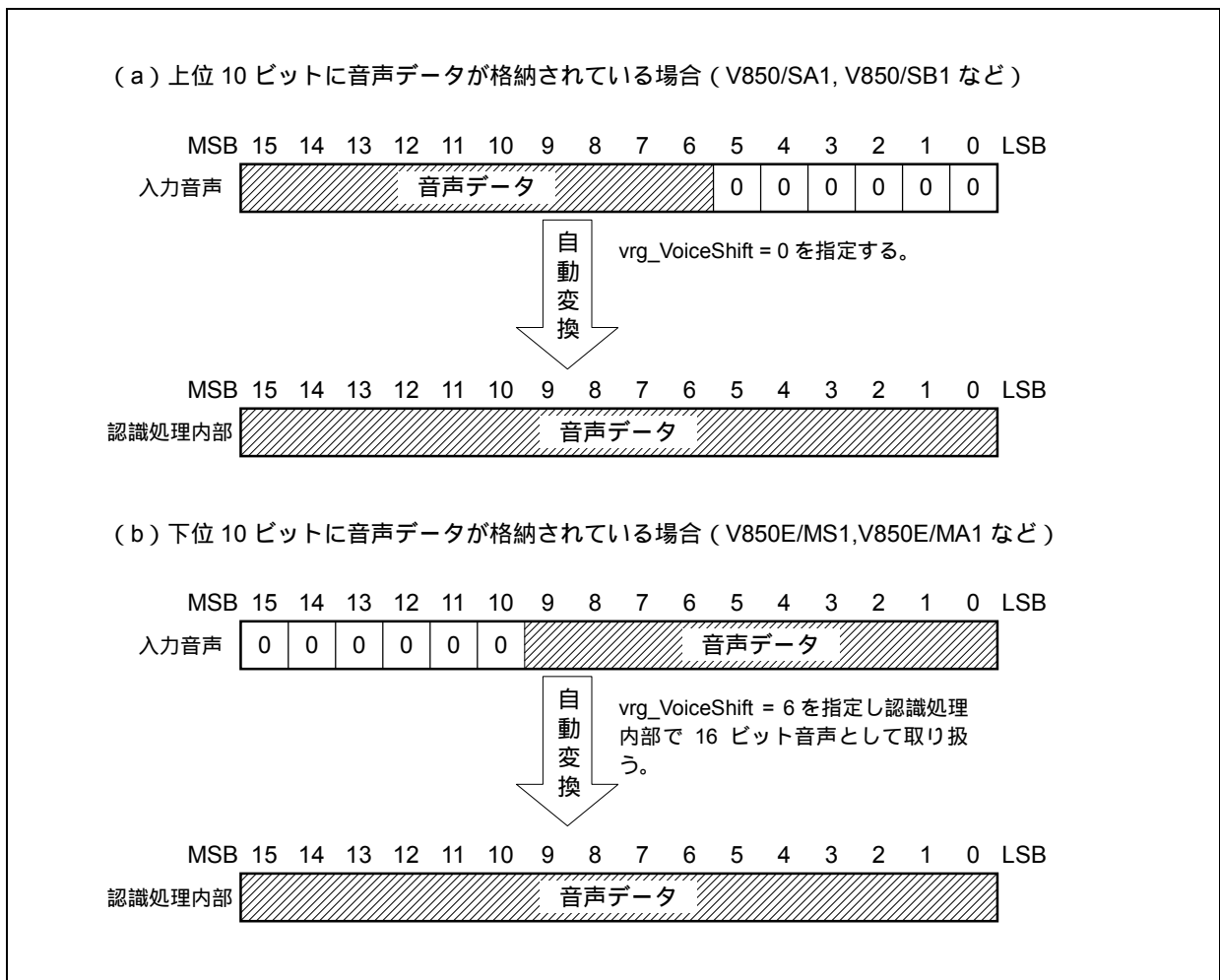
【機能】

## (a) 動作

AP703000-B07 では、入力される音声データは 16 ビット音声データが最適です。入力される A/D コンバータ分解能が 16 ビット未満である場合、vrg\_VoiceShift 変数を指定することによって、認識処理内部で左シフトを自動的に行い、16 ビットの音声に変換し以後の処理を行います。

vrg\_VoiceShift 変数に指定可能な値は、0-15 の範囲の整数です。

図2 - 5 vrg\_VoiceShift変数指定例



V850E/MA1 使用時のサンプル・プログラムを次に示します。

```
#define VRGMEMAREASIZE   x x x x x /* 認識ワーク・サイズ */
#define VRGBUFSIZE       x x x x x /* 音声バッファ・サイズ */
#define VRGSHIFTNUM      6         /* 音声シフト数 */

long   vrg_MemArea[VRGMEMAREASIZE/sizeof(long)];
int    vrg_MemAreaSize = VRGMEMAREASIZE;
short  vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int    vrg_BufSize = VRGBUFSIZE;
int    vrg_VoiceShift = VRGSHIFTNUM;

main()
{
    :
}
```

- 注意 1. vrg\_VoiceShift 変数は、ユーザ・オリジナル・セクションに割り付けないでください。
2. ミドルウェア関数動作中に値を書き換えしないでください。
3. vrg\_VoiceShift 変数に指定可能な値は、0-15 の範囲の整数です。
4. vrg\_Recog 関数 (2. 5. 1 参照) 以外のライブラリ関数を動作させるときも、vrg\_VoiceShift 変数の設定は必要です。

## 2.5 ライブラリ関数仕様

ライブラリ関数とは、AP703000-B07 で用意している関数群です。

### 2.5.1 vrg\_Recog 関数

【関数名】 vrg\_Recog

【概要】 1 単語発声に対して認識を行います。

【形式】 int vrg\_Recog(short \*DicArea\_p);

【引き数】 [ ユーザ入力変数 ] \*DicArea\_p : 認識単語辞書格納領域先頭アドレス

【戻り値】

表2 - 5 vrg\_Recog関数戻り値一覧

戻り値	内 容
0 以上の値	第 1 位認識候補単語 ID (正常終了)
E_VRGENDTIMEOUT [- 0x203 (0xFFFFFDFD)]	約 15 秒以上の音声を検出され続けたため、強制終了しました。
E_VRGTERMINATE [- 0x204 (0xFFFFDFDC)]	認識停止関数 vrg_Stop がコールされ、強制終了しました。
E_VRGOVERWRITE [- 0x205 (0xFFFFDFDB)]	音声バッファが上書きされました。処理がオーバーフローしています。 より小語彙数の認識単語辞書を使用してください。
E_VRGDIFORMAT [- 0x206 (0xFFFFDFDA)]	辞書フォーマットが違います。認識単語辞書格納部のアドレスが間違っているか、破壊されています。
E_VRGMEMAREASIZE [- 0x207 (0xFFFFDFD9)]	vrg_MemArea が不足しています。 必要なワーク領域サイズを確保してください。
E_VRGDICALIGN [- 0x208 (0xFFFFDFD8)]	認識単語辞書が 2 バイト・アラインされていません。
E_VRGVBUFAREASIZE [- 0x209 (0xFFFFDFD7)]	音声バッファ・サイズに規定外の値が指定されました。 256 バイトの整数倍かつ 768 バイト以上の値を指定してください。
E_VRGVOICESHIFT [- 0x20b (0xFFFFDFD5)]	vrg_VoiceShift 変数に規定外の値が指定されました。 0-15 の範囲の整数を指定してください。
E_VRGTIMEOUTVAL [- 0x20d (0xFFFFDFD3)]	タイムアウト設定値が異常です。vrg_InitParam 関数内で正しい値を設定してください。
E_VRGFATAL [- 0x101 (0xFFFFFEFF)]	予期しないエラーが発生しました。

## 【機能】

## (a) 動作

指定された認識単語辞書を使用して、1 単語発声に対して認識を行います。

正常に認識結果が得られた場合は、戻り値として第 1 位認識候補単語 ID が返却されます。

15 秒間音声検出がされ続けた場合は強制終了します。

音声を検出されない場合は、vrg\_Recog 関数の処理は終了しません。

## (b) 認識単語辞書の指定

引き数 DicArea\_p で指定された認識単語辞書を認識対象とします。ここで正常でない認識単語辞書（一部破壊された辞書など）を指定した場合の動作は保証しません。

## (c) 単語 ID

単語 ID は、辞書作成時に入力された単語の順に先頭から 0, 1, 2, ... と割り振られています。

たとえば、「サッポロ」「センダイ」「チハ」「カワサ」「ヨコハマ」「ナゴヤ」「キョウト」「オオサカ」「コウベ」「ヒロシマ」「キタユウシュウ」「フカカ」の順に 12 単語が登録されている場合は表 2 - 6 のようになります。

表2 - 6 単語ID例

登録単語	単語 ID
サッポロ	0
センダイ	1
チハ	2
カワサ	3
ヨコハマ	4
ナゴヤ	5
キョウト	6
オオサカ	7
コウベ	8
ヒロシマ	9
キタユウシュウ	10
フカカ	11



## (d) 認識結果情報

正常終了した場合の戻り値が、第1位認識候補単語IDとなります。

表2-7にあるような第1位認識候補単語ID以外の認識結果情報はvrg\_MemArea変数の先頭20バイトに格納されますので、これらの値を参照することも可能です。参照する際は、vrg\_MemArea変数の先頭アドレスからのオフセット値を考慮し参照してください。

単語スコアは、入力された発声と認識候補単語との差を表します。使用環境や発声話者によって上下します。小さいほど認識結果の信頼性が高い傾向があります。認識結果の確度の目安として使用してください。

表2-7 認識結果情報一覧

vrg_MemArea 先頭からのオフセット値	型	内容
0x0	unsigned short	第1位認識候補単語ID
+0x2	unsigned short	第2位認識候補単語ID
+0x4	unsigned short	第3位認識候補単語ID
+0x6	unsigned short	第1位認識候補単語スコア
+0x8	unsigned short	第2位認識候補単語スコア
+0xa	unsigned short	第3位認識候補単語スコア
+0xc	unsigned short	音声レベル
+0xe	unsigned short	ノイズ相対レベル
+0x10	unsigned short	音声入力時間
+0x12	unsigned short	1フレーム目の音声レベル

## (e) ユーザ・OWN・コーディング関数のコール

vrg\_Recog 関数は、内部でユーザ・OWN・コーディング関数をコールします(図2-1 vrg\_Recog 関数フロー参照)。

**注意** vrg\_Recog 関数は、音声を検出しないと処理を終了しません。音声を検出するためには、「取得した音声データがvrg\_InputWaveData関数(2.5.3参照)から認識処理に引き渡されること」、もしくは「vrg\_InputWaveDataDMA関数(2.5.4参照)によって音声データ128サンプルの入力を通知されること」が必要です。必要なときのみvrg\_InputWave関数の動作を許可するために、vrg\_IntOn関数、vrg\_IntOff関数(2.6.2参照)をコールし、音声取り込みの割り込み要求を制御します。

## 2.5.2 vrg\_Stop 関数

- 【関数名】** vrg\_Stop  
**【概要】** 認識処理の強制終了処理を行います。  
**【形式】** void vrg\_Stop(void);  
**【引き数】** なし  
**【戻り値】** なし  
**【機能】**

### (a) 動作

vrg\_Stop 関数をコールすることにより、認識処理が強制終了されます。その際、vrg\_Recog 関数 (2.5.1 参照) では認識結果情報としてすべて 0x7FFF を格納し、E\_VRGTERMINATE エラー・コードを返却して終了します。

### (b) 呼び出し可能なタイミング

vrg\_Stop 関数が有効に動作する状態は、ユーザ・OWN・コーディング関数の vrg\_RecogReady 関数 (2.6.3 参照) のコール後から vrg\_RecogComplete 関数 (2.6.4 参照) コール前までです。その他の状況でコールされた場合の動作は保証しません。

### (c) 割り込みハンドラからの呼び出し

vrg\_Stop 関数を割り込みハンドラ内でコールする際、vrg\_Stop 関数内で使用するレジスタについては内部で退避を行っています。ユーザ・プログラムで割り込みベクタ内でレジスタを使用するものに対しては退避処理を必ず行ってください。

**注意** vrg\_Stop 関数をコールした場合でも、処理タイミングの問題上 vrg\_Recog 関数 (2.5.1 参照) の戻り値が E\_VRGTERMINATE でない場合があります。

**備考** vrg\_Stop 関数がコールされてから、vrg\_Recog 関数が強制終了するまでに多少応答時間 (数 10 ms 程度) がかかる場合もあります。

### 2.5.3 vrg\_InputWaveData 関数

【関数名】 vrg\_InputWaveData

【概要】 音声データの格納処理を行います。

【形式】 void vrg\_InputWaveData(unsigned short WaveData);

【引き数】 [ ユーザ入力変数 ] WaveData : 入力音声データ

【戻り値】 なし

【機能】

#### (a) 動作

vrg\_Recog 関数 (2.5.1 参照) 内で使用された音声バッファへ 1 サンプル分の音声データを格納します。

#### (b) 音声データ・フォーマット

入力される音声データ・フォーマットは 16 ビット音声データです。

内蔵 10 ビット A/D コンバータを使用する場合、vrg\_VoiceShift 変数 (2.4.5 参照) を指定することで 16 ビットにスケールアップされた値として処理することが可能です。符号なしハーフワード・データ (0-65535) となるように vrg\_VoiceShift 変数を指定してください。

vrg\_VoiceShift 変数の指定によって入力音声データの振幅が 16 ビットの範囲を越えないようご注意ください。

- 注意 1. vrg\_InputWaveData 関数は、vrg\_InputWave 関数 (2.6.1 参照) からコールしてください。
2. 内蔵 10 ビット A/D コンバータを使用する場合、vrg\_VoiceShift 変数 (2.4.5 参照) を指定することで 16 ビットにスケールアップされた値として処理することが可能です。A/D コンバータの仕様については、使用されている V850 ファミリのデバイスのユーザーズ・マニュアル ハードウェア編を参照してください。

次に V850E/MA1 における処理の一例を示します。

V850E/MA1 内蔵 A/D コンバータは A/D 変換結果レジスタの下位 10 ビットに A/D 変換結果が格納されるため、`vrg_VoiceShift = 6` を指定することで、16 ビット音声データとして認識処理を行います。

```

:
#define   VRGSHIFTNUM   6           /* 音声シフト数 */
int      vrg_VoiceShift = VRGSHIFTNUM;
:
main()
{
:
}

:
.globl  _vrg_InputWave
_vrg_InputWave:
:
/* A/D コンバータから音声を取り込みます。 */
ld.h   _ADCR7[r0],   r6

/*
 * vrg_InputWaveData() をコールして、音声 1 サンプルを
 * vrg_Recog() に引き渡します。
 */
jarl   _vrg_InputWaveData, lp
:
reti
```

次に V850/SA1 における処理の一例を示します。

V850/SA1 内蔵 A/D コンバータは A/D 変換結果レジスタの上位 10 ビットに A/D 変換結果が格納されるため、vrg\_VoiceShift = 0 を指定し、A/D 変換結果を無加工で 16 ビット音声データとして認識処理を行います。

```

:
#define VRGSHIFTNUM      0      /* 音声シフト数 */
int    vrg_VoiceShift = VRGSHIFTNUM;
:
main()
{
:
}

.globl _vrg_InputWave
_vrg_InputWave:

:
/* A/D コンバータから音声を取り込みます。 */
ld.h   _ADCR7[r0],    r6

/*
* vrg_InputWaveData() をコールして、音声 1 サンプルを
* vrg_Recog() に引き渡します。
*/
jarl   _vrg_InputWaveData,lp
:
reti

```

### (c) 呼び出すタイミング

vrg\_InputWaveData 関数を使用して音声を入力する場合、vrg\_InputWave 関数 (2.6.1 参照) は 125  $\mu$ s ごと (サンプリング周波数: 8 kHz の場合) に起動されるように設定し、その中で 1 サンプル分の音声データを取得して vrg\_InputWaveData 関数の引き数に設定してください。

- 注意 1.** vrg\_Recog 関数 (2.5.1 参照) 内で使用される音声バッファに音声データを格納する動作をするため、vrg\_Recog 関数動作中でないときに、vrg\_InputWave 関数をコールした場合の動作は保証しません。
- 2.** vrg\_InputWaveData 関数と vrg\_InputWaveDataDMA 関数は、排他的に使用してください。
- 3.** 1 回の発声に対する認識 (1 度の vrg\_Recog 関数のコール) 中に vrg\_InputWaveData 関数と vrg\_InputWaveDataDMA 関数の双方をコールした場合の動作は保証しません。

## 2.5.4 vrg\_InputWaveDataDMA 関数

- 【関数名】 vrg\_InputWaveDataDMA  
 【概要】 128 サンプルの音声データ入力を音声認識処理に通知します。  
 【形式】 void vrg\_InputWaveDataDMA(void);  
 【引き数】 なし  
 【戻り値】 なし  
 【機能】

## (a) 動作

音声バッファ (vrg\_VoiceBuf 領域) へ 128 サンプルの音声データ書き込みを行ったことを vrg\_Recog 関数に通知するために使用する関数です。

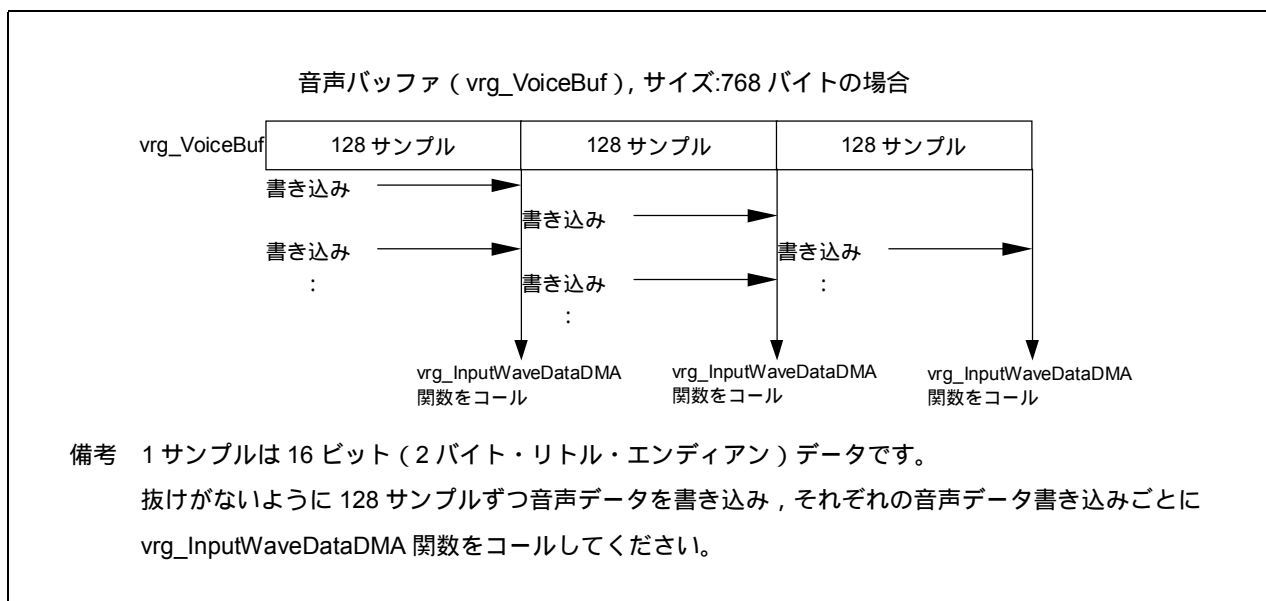
特に DMA 機能を使用して内蔵 A/D コンバータから音声バッファへの転送を行うことを想定していません。この場合、1 サンプルずつ (125  $\mu$ s ごと) 割り込みをコールしなければならない vrg\_InputWaveData 関数使用時に比べて、128 サンプルごと (16 ms ごと) の割り込み処理で音声入力が可能となり割り込み処理時間の短縮が図れます。

vrg\_InputWaveData 関数 (2.5.3 参照) を使用せず、アプリケーション側で音声バッファに音声データを直接書き込む方式で音声データを入力する場合に使用します。音声データ 128 サンプルの書き込みごとに、この関数を 1 回コールしてください。

アプリケーション側で音声バッファに音声を書き込む場合、音声バッファをリング・バッファとして先頭から抜けなく音声データを書き込む必要があります。特に、繰り返し認識処理を行うような場合は注意してください。

次に例を示します。

図2-6 アプリケーション側での音声書き込み処理



**(b) 音声データ・フォーマット**

入力される音声データ・フォーマットは 16 ビット音声データです。

内蔵 10 ビット A/D コンバータを使用する場合、vrg\_VoiceShift 変数 (2.4.5 参照) を指定することで 16 ビットにスケールアップされた値として処理することが可能です。符号なしハーフワード・データ (0-65535) となるように vrg\_VoiceShift 変数を指定してください。

vrg\_VoiceShift 変数の指定によって入力音声データの振幅が 16 ビットの範囲を越えないように注意してください。

**(c) 呼び出すタイミング**

vrg\_InputWaveDataDMA 関数を使用して音声を入力する場合、DMA 機能などを使用して音声バッファ (vrg\_VoiceBuf 領域) に 128 サンプルの音声データを書き込み、vrg\_InputWaveDataDMA 関数を 1 度コールしてください。DMA 転送回数を 128 回に設定し、DMA 転送終了割り込みから vrg\_InputWave 関数をコールし、vrg\_InputWave 関数内から vrg\_InputWaveDataDMA 関数をコールするようなコーディングを推奨します。この場合、vrg\_InputWaveDataDMA 関数コール後に次の DMA 転送を開始するコーディングをしてください。

- 注意**
1. vrg\_InputWaveDataDMA 関数は、vrg\_InputWave 関数 (2.6.1 参照) からコールしてください。
  2. 内蔵 10 ビット A/D コンバータを使用する場合、vrg\_VoiceShift 変数 (2.4.5 参照) を指定することで 16 ビットにスケールアップされた値として処理することが可能です。A/D コンバータの仕様については、使用されている V850 ファミリのデバイスのユーザーズ・マニュアル ハードウェア編を参照してください。

次に V850E/MA1 における処理の一例を示します。

V850E/MA1 内蔵 A/D コンバータは A/D 変換結果レジスタの下位 10 ビットに A/D 変換結果が格納されるため、vrg\_VoiceShift = 6 を指定することで、16 ビット音声データとして認識処理を行います。

図2 - 7 DMA機能を利用した音声入力処理フロー（メイン処理）（1/2）

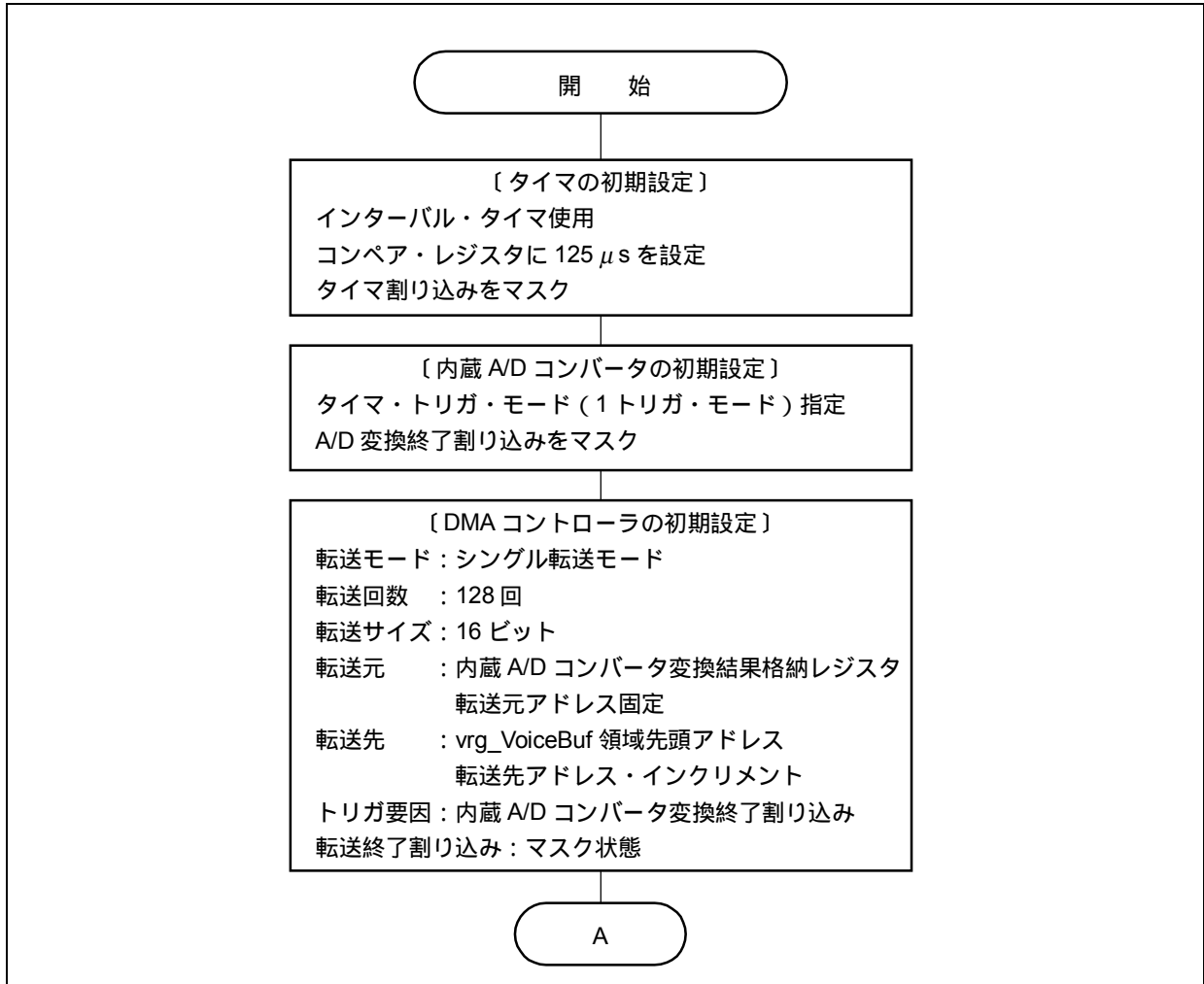




図2-7 DMA機能を利用した音声入力処理フロー（メイン処理）（2/2）

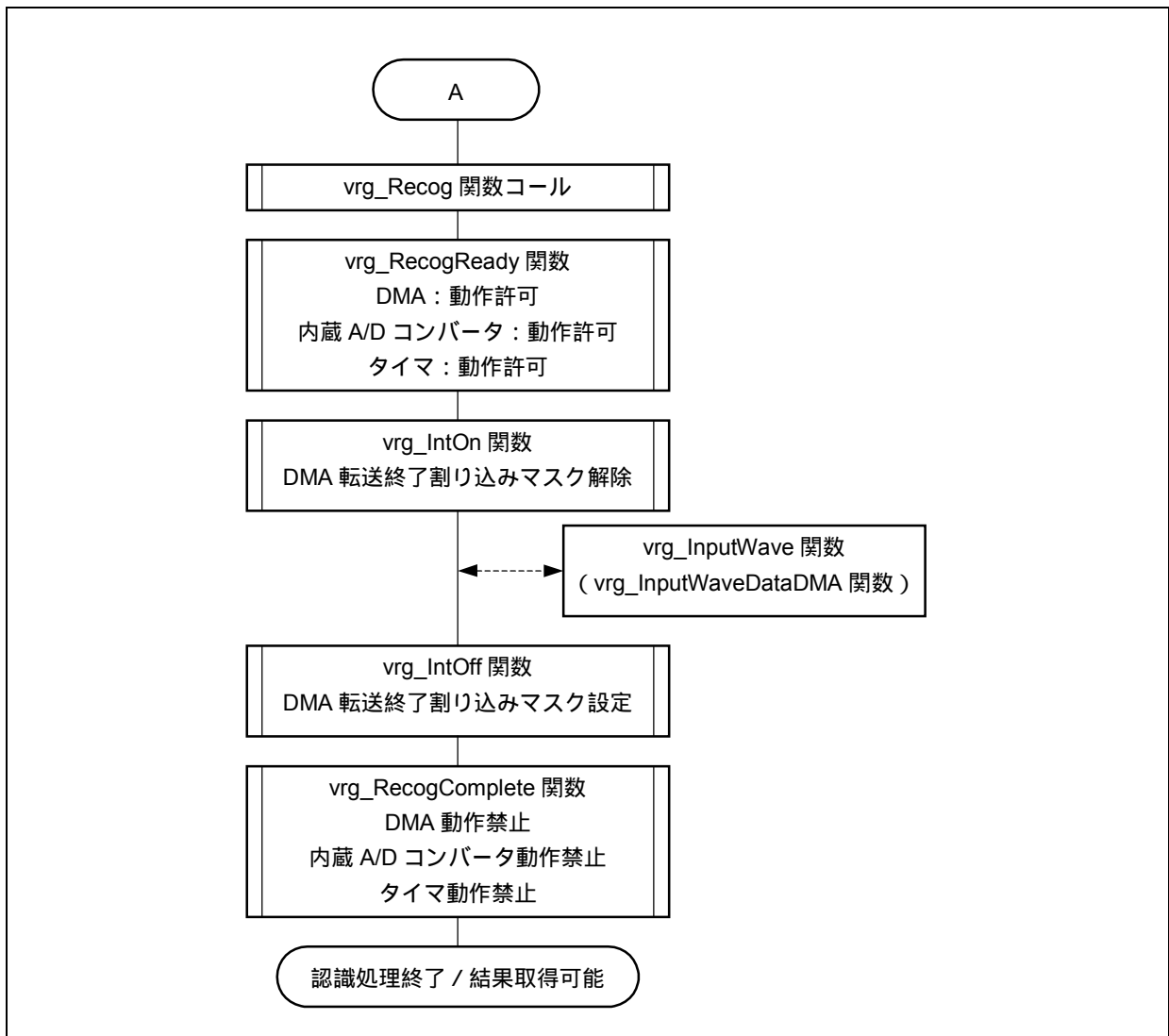
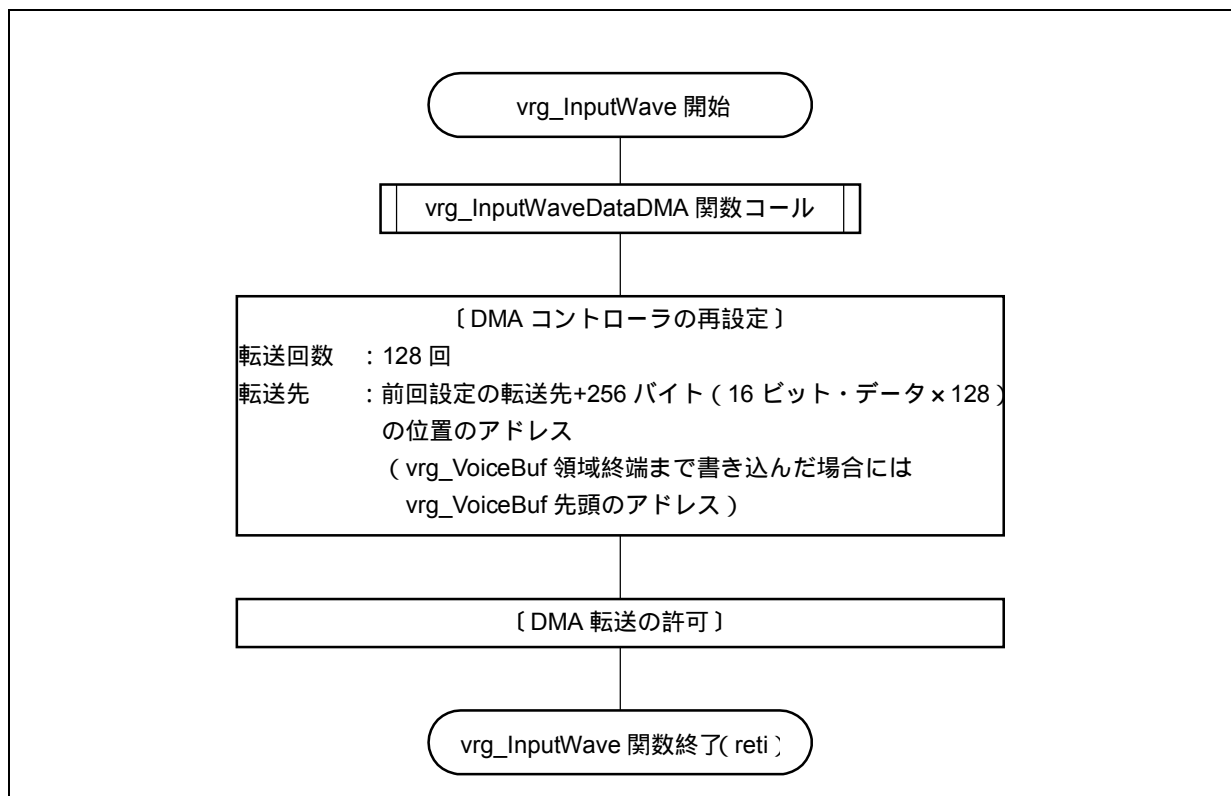


図2 - 8 DMA機能を利用した音声入力処理フロー（割り込み処理）



## 2.5.5 vrg\_MakeDic 関数

【関数名】 vrg\_MakeDic

【概要】 認識に使用する認識単語辞書を作成します。

【形式】 int vrg\_MakeDic(char \*DicText\_p, short \*DicArea\_p, short DicAreaSize);

【引き数】 [ ユーザ入力変数 ] \*DicText\_p : 登録単語列の先頭アドレス

[ ユーザ入力変数 ] \*DicArea\_p : 認識単語辞書格納領域先頭アドレス

[ ユーザ入力変数 ] DicAreaSize : 認識単語辞書格納領域サイズ (バイト)

【戻り値】

表2-8 vrg\_MakeDic関数戻り値一覧

戻り値	内 容
0以上の値	作成された認識単語辞書のサイズ (正常終了)
E_VRGDICAREASIZE [- 0x501 (0xFFFFFAFF)]	認識単語辞書格納領域が不足しました。
E_VRGCHAR [- 0x502 (0xFFFFFAFE)]	不正文字が使用されたか、または有効な文字列が検出されないまま NULL が検出されました。
E_VRGCHARNUM [- 0x503 (0xFFFFFAFD)]	1単語の最大文字数 (20文字) を越えた単語が登録されています。
E_VRGOVERFLOW [- 0x504 (0xFFFFFAFC)]	登録単語数、文字数が多過ぎます (仕様範囲外です)。
E_VRGDICAREAALIGN [- 0x505 (0xFFFFFAFB)]	認識単語辞書出力先が2バイト・アラインされていません。
E_VRGDICMEMSIZE [- 0x506 (0xFFFFFAFA)]	vrg_MemArea が不足しています。必要なワーク・サイズ領域を確保して ください。
E_VRGFATAL [- 0x101 (0xFFFFFEFF)]	予期しないエラーが発生しました。

【機能】

(a) 動作

AP703000-B07 はカナ文字列を直接認識に使用できません。vrg\_MakeDic 関数もしくは PC 版辞書作成ツール (第4章 PC版ユーザー辞書作成ツール参照) を使用して専用バイナリ形式に変換したものを使用します。この専用バイナリ形式に変換したものを「認識単語辞書」と呼び、「専用バイナリ形式に変換する」ことを「辞書作成」と呼びます。

半角カナ文字によって指定された文字列から認識単語辞書を作成し、認識単語辞書格納領域に出力します。半角カナ文字列の先頭アドレスを第一引き数 DicText\_p に設定してください。

この文字列から作成した辞書バイナリ・データを、第二引き数 DicArea\_p で示される領域に格納します。認識単語辞書格納領域を別に確保し、その先頭アドレスを引き渡してください。

(b) 入力フォーマット

入力フォーマットを図2-9に示します。図2-9内の各入力文字の様子は、表2-9に従ってください。

## (c) 認識単語辞書格納領域

第二引き数 DicArea\_p で指定した認識単語辞書格納領域のサイズを第三引き数 DicAreaSize に設定してください。作成される認識単語辞書のサイズが DicAreaSize で指定したサイズよりも大きい場合、認識単語辞書格納領域不足エラーが返却されます。この場合、vrg\_MemArea 先頭に格納される認識単語辞書作成結果情報から作成される認識単語辞書のサイズが得られますので、その容量の認識単語辞書格納領域を確保したあと、再度辞書作成を実行してください。

**注意** 認識単語辞書格納領域は vrg\_MemArea 変数とは別領域として確保する必要があります。

図2-9 入力フォーマット



表2-9 各入力文字仕様

フォーマット	仕 様
単語	「半角カナ文字」で指定してください。 1単語の最大文字数は20文字です。 使用可能文字は表2-10に示すとおりです。
区切り	「半角スペース(0x20)」で指定してください。
終端	「NULL(0x00もしくは'¥0')」で指定してください。

表2 - 10 使用可能文字一覧表

使用方法およびその可否	文 字	
使用制限なし	ア イ ウ エ オ カ キ ク ケ コ サ シ ス セ ソ タ チ ツ テ ト ナ ニ ヌ ネ ノ ハ ヒ フ ヘ ホ マ ミ ム メ モ ヤ ユ ヨ ラ リ ル レ ロ ワ ヲ ガ キ グ ゲ コ ザ ジ ズ セ ソ ダ チ ツ デ ド パ ビ プ ペ ホ バ ビ プ ペ ホ ヲ	
単語の先頭では使用不可	- ソ ッ	
特殊な文字との組合せでのみ 使用可	文字	下記のいずれかの文字のあとでのみ使用可能
	ア	フ ウ
	イ	ウ テ フ ウ デ
	ウ	ト フ ウ ド
	エ	ウ シ チ ツ フ ヲ ジ ェ
	オ	ウ フ ヲ
	ヤ ヨ	キ シ チ ニ ヒ ミ リ キ ジ ェ ビ ビ ヲ
	ユ	キ シ チ テ ニ ヒ ミ リ キ ジ ェ デ ビ ビ ヲ
「ッ」のあとで使用不可	ア イ ウ エ オ ナ ニ ヌ ネ ノ マ ミ ム メ モ ヤ ユ ヨ ラ リ ル レ ロ ワ ヲ ソ ッ ウ エ ヲ ナ ニ ヌ ミ ヲ ヲ ヲ	
末尾で使用不可	ッ	
「-」の前では使用不可	ソ ッ	
その他，使用不可の組み合わせ	ソ ッ	

(d) 辞書作成結果の出力

「辞書サイズ」情報および「認識ワーク・サイズ」情報を vrg\_MemArea 変数 (2.4.1 参照) によって確保された領域の先頭 8 バイトに表 2 - 11 のフォーマットに従って格納します。「認識ワーク・サイズ」情報に従って vrg\_MemArea を確保して認識に使用してください。

辞書作成部では，チェック・サムが常に認識単語辞書に組み込まれます。

表2 - 11 辞書作成結果情報

オフセット	型	内 容
0	long	vrg_MakeDic 関数が正常終了，または認識単語辞書格納領域不足エラー (E_VRGDICAREASIZE) 発生の場合，作成された (作成される) 辞書サイズ (バイト) を格納します。 その他の場合，0x7FFFFFFF が格納されています。
+4	long	作成された認識単語辞書による認識を行う際に必要な認識ワーク・サイズ (バイト) を格納します。なお，vrg_MakeDic 関数が正常終了でない場合 0x7FFFFFFF が格納されています。

## (e) 認識単語辞書データのチェック・サム

認識単語辞書データ内に格納されたチェック・サムを用いて、データが破壊されていないかを調べることができます。

たとえば、認識単語辞書データのダウンロード時に破壊されていないかを調べることが可能となります。

ただし、ライブラリ側ではチェック・サムの確認は行っていません。ユーザ・プログラム内で必要に応じて確認を行ってください。

チェック・サム仕様については、第5章 チェック・サム機能を参照してください。

サンプル・プログラムを次に示します。

```
#include "vrglib.h"

#define VRGMEMAREASIZE 0x0000 /* 認識ワーク・サイズ */
#define VRGBUFSIZE 0x0000 /* 音声バッファ・サイズ */
#define VRGSHIFTNUM 0x0000 /* 音声シフト数 */
#define DICAREASIZE 0x0000 /* 辞書格納領域サイズ */

long vrg_MemArea[VRGMEMAREASIZE/sizeof(long)];
int vrg_MemAreaSize = VRGMEMAREASIZE;
short vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int vrg_BufSize = VRGBUFSIZE;
int vrg_VoiceShift = VRGSHIFTNUM;
short DicArea[DICAREASIZE/sizeof(short)]; /* 辞書格納領域*/

charDicText[]
={ "サッポロ センダイ 豚 加サキ ヨコハマ コノヤ キョウト 材カカ コウハ` ヒロシマ 杉キウシユウ フカカ¥0" };

void
main( void )
{
    int ret;

    ret = vrg_MakeDic(DicText, DicArea, DICAREASIZE);

    :

}
```

**注意** 正常終了しない場合でも、vrg\_MemArea 変数 (2.4.1 参照) で指定されたワーク領域の先頭 8 バイトまでに対し書き込み処理が発生します。

## 2.5.6 vrg\_GetVersion 関数

- 【関数名】 vrg\_GetVersion
- 【概要】 AP703000-B07 のバージョン情報を出力します。
- 【形式】 int vrg\_GetVersion ( MWCVERSION \*VersionPacket );
- 【引き数】 [ ミドルウェア出力変数 ] \*VersionPacket
- 【戻り値】 常に「0」値が返却されます。
- 【構造体】 次の構造体は、ヘッダ・ファイル mwclib.h に記述されています。

```
typedef struct {
    int    Version; /* バージョン・ナンバー */
    char  *Serial; /* シリアル・ナンバー */
    int    *Others; /* 未使用 */
}MWCVERSION;
```

## 【機能】

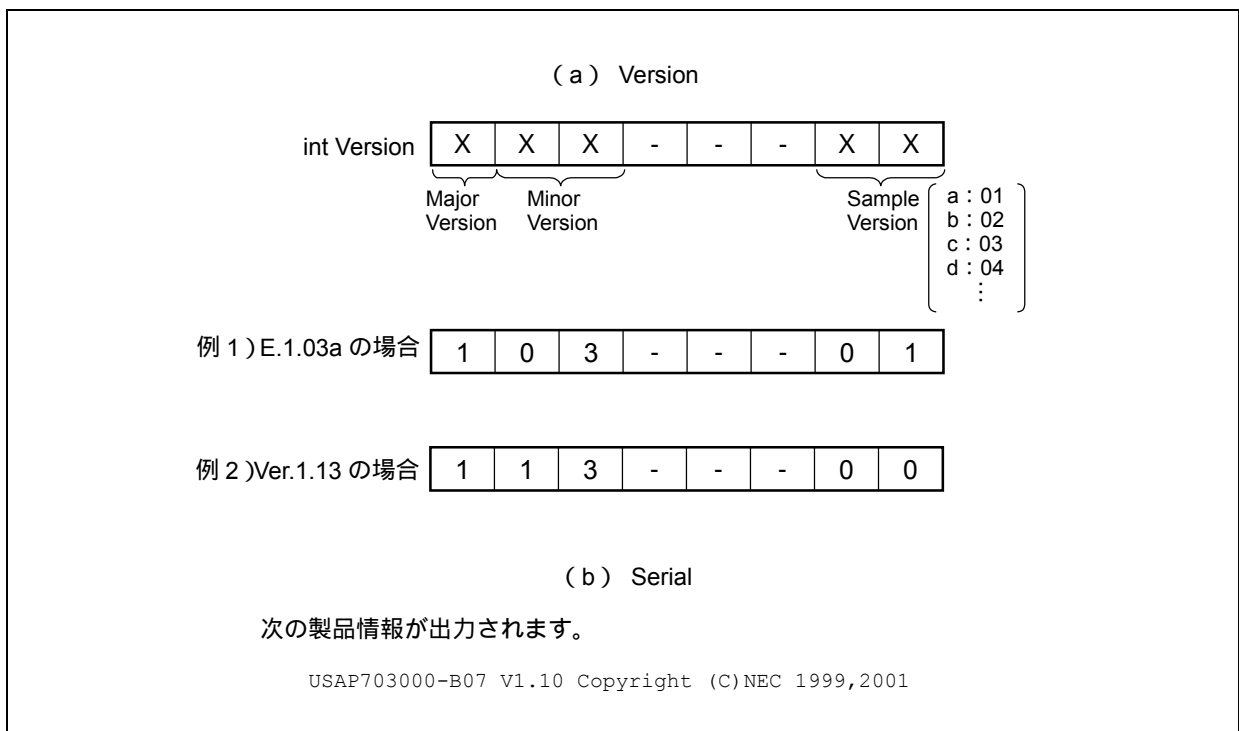
## (a) 動作

AP703000-B07 のバージョン情報を出力します。出力のフォーマットは (b) バージョン情報のとおりです。

MWCVERSION 構造体型変数を確保して、そのアドレスを第一引き数に設定してください。

## (b) バージョン情報

図2 - 10 バージョン情報



## 2.6 ユーザ・OWN・コーディング関数仕様

ユーザ・OWN・コーディング関数とは、ご使用になる環境にあわせて変更する関数群です。

### 2.6.1 vrg\_InputWave 関数

ハードウェア環境にあわせた、ユーザ・コーディングが必須です。

【関数名】 vrg\_InputWave

【概要】 vrg\_InputWaveData 関数 (2.5.3 参照) を通して音声を vrg\_Recog 関数 (2.5.1 参照) 内で使用される音声バッファに引き渡す、もしくは、vrg\_InputWaveDataDMA 関数 (2.5.4 参照) を通して 128 サンプルの音声が入力されたことを vrg\_Recog 関数に通知する割り込みハンドラです。

【形式】 void vrg\_InputWave(void);

【引き数】 なし

【戻り値】 なし

【機能】

#### (a) 動作

vrg\_InputWaveData 関数 (2.5.3 参照) を通して vrg\_Recog 関数 (2.5.1 参照) 内で使用される音声バッファに音声を格納する、もしくは、vrg\_InputWaveDataDMA 関数 (2.5.4 参照) を通して音声が入力されたことを vrg\_Recog 関数に通知する割り込みハンドラです。

vrg\_InputWave 関数を対応する割り込みのハンドラ・アドレスにセットするためには、次のような jr 命令を、使用する割り込みのハンドラ・アドレスに対応させて記述する必要があります。

```
jr    _vrg_InputWave
```

#### (b) 呼び出すタイミング

- vrg\_InputWaveData 関数を使用する場合

vrg\_InputWave 関数を 125  $\mu$ s (サンプリング周波数: 8 kHz の場合) ごとに起動するように設定し、その中で 1 サンプル分の音声データを取得して vrg\_InputWaveData 関数の引き数に設定してください。

- vrg\_InputWaveDataDMA 関数を使用する場合

vrg\_InputWave 関数を 128 サンプルの音声データ書き込みが終了するごとに起動するように設定し、その中で vrg\_InputWaveDataDMA 関数をコールしてください。

#### (c) 音声データ・フォーマット

音声データのフォーマットは、符号なしハーフワード・データ (0-65535) 音声入力を前提としています。内蔵 A/D コンバータなどをご使用の場合は、vrg\_VoiceShift 変数の設定が必要となる場合がありますのでご注意ください。



- 注意 1. ハンドラ・アドレスについては、使用されている V850 ファミリのデバイスのユーザーズ・マニュアル ハードウェア編を参考にしてください。
2. 割り込みハンドラの処理を記述する際は、「レジスタの退避/復帰処理」、「スタックの切り替え処理」、「割り込みハンドラからの復帰処理」などに注意してください。また多重割り込みを行うときは CPU の「EIPC, EIPSW」の扱いに充分注意してください。
3. vrg\_InputWaveData 関数 (2. 5. 3 参照) または vrg\_InputWaveDataDMA 関数 (2. 5. 4 参照) が、vrg\_Recog 関数 (2. 5. 1 参照) 動作中のみコールされるために vrg\_IntOn 関数および vrg\_IntOff 関数 (2. 6. 2 参照) のコーディングが必須です。vrg\_IntOn 関数コールで、初めて vrg\_InputWave 関数が起動するように設定し、同様に vrg\_IntOff 関数で vrg\_InputWave 関数が起動しなくなるように割り込み周辺の設定をしてください。

## 2.6.2 vrg\_IntOn 関数 / vrg\_IntOff 関数

ハードウェア環境にあわせた、ユーザ・コーディングが必須です。

音声取り込みのための割り込み要求の許可 / 禁止を制御する関数です。

## (1) マスカブル割り込みの概説 (vrg\_IntOn 関数 / vrg\_IntOff 関数を使用する前に)

V850 ファミリでは、マスカブル割り込みに対して次の2段階の制御を行います。

- ・割り込み全体に対する許可 / 禁止<sup>※</sup> (ei/di 命令の発行など)
- ・各割り込み要因に対するマスク (割り込み制御レジスタ)

あるマスカブル割り込み処理を受け付けるためには、すべての割り込みを許可し、かつ、受け付けようとする割り込み要因に対するマスクを解除する必要があります。

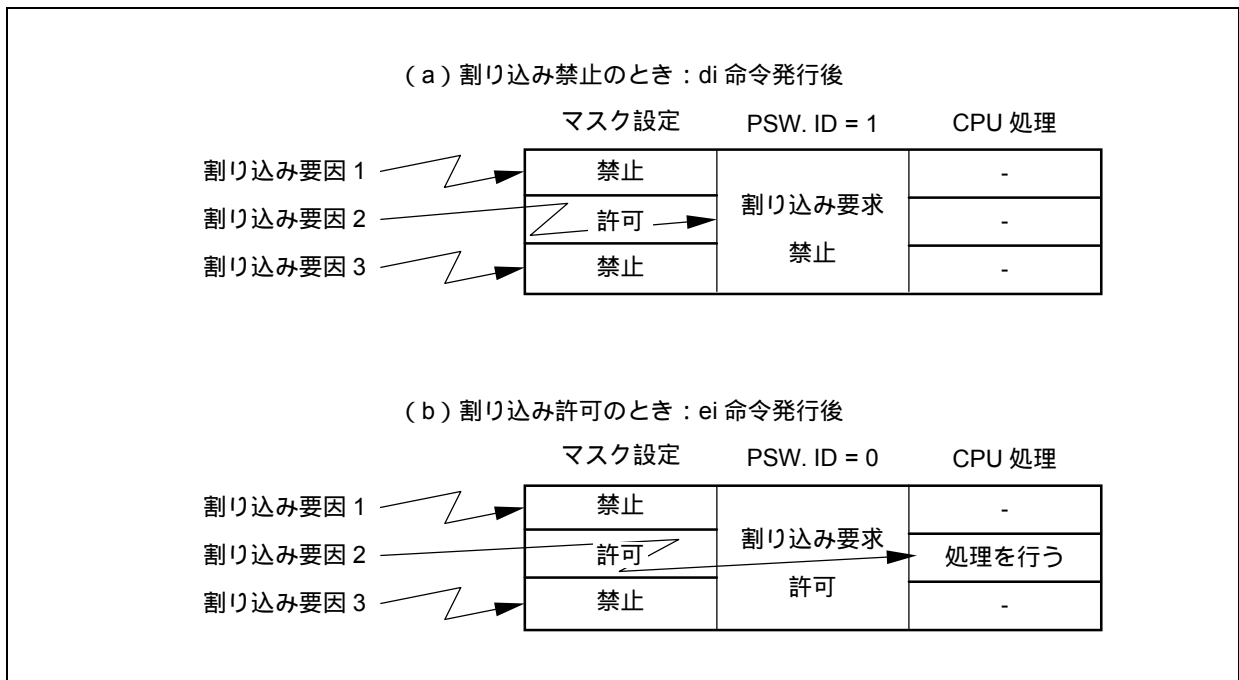
マスカブル割り込みについての詳細は、使用されている V850 ファミリのデバイスのユーザズ・マニュアル **ハードウェア編**を参照してください。

注 V850 ファミリのプログラム・ステータス・ワード (PSW) で示されます。

PSW. ID = 0 割り込み許可

PSW. ID = 1 割り込み禁止

図2 - 11 マスカブル割り込みのイメージ



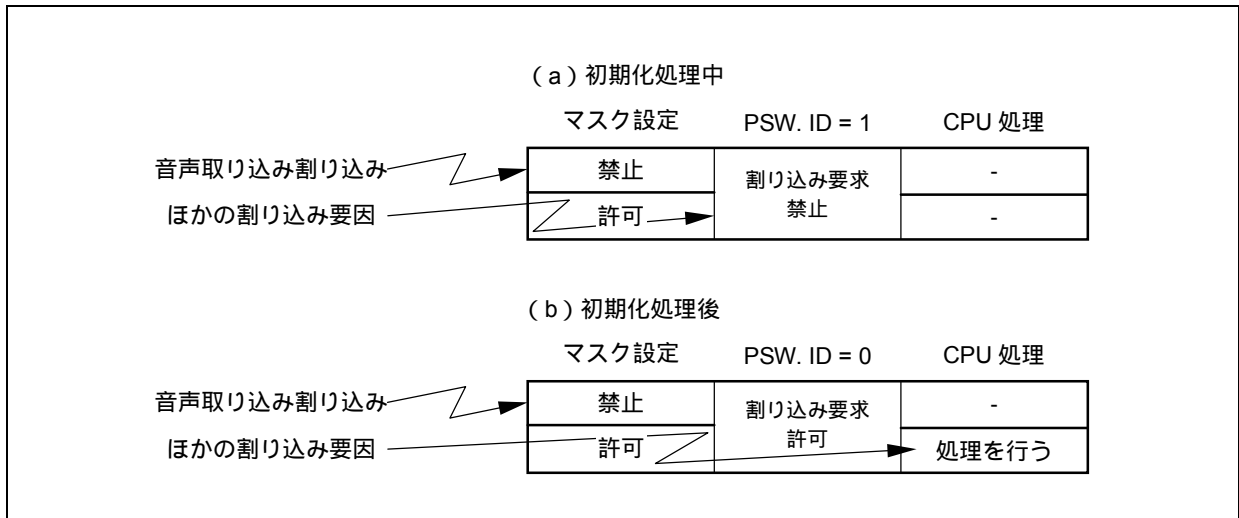
## (2) vrg\_Recog 関数の呼び出し前の設定

次に、割り込み制御レジスタを用いて割り込みマスクを操作する場合の割り込み許可方法の例を示します。

## (a) コーディング例

割り込み設定時にマスクされた状態を初期状態とし、その後 vrg\_Recog 関数 (2.5.1 参照) を呼び出す前までに ei 命令を発行しておきます。この時点では、ほかの割り込みは動作できますが、音声取り込みのための割り込みは動作しません。

図2-12 メイン処理



メイン処理のサンプル・プログラムを次に示します。

```
main()
{
    :
    /* 初期化のため割り込み禁止 */
    __asm("di");
    /* ADC・タイマ・割り込み等の初期化 */
    InputInit();
    /* 認識処理前に割り込み許可 */
    __asm("ei");
    :
    /* 1 発声分の認識処理を行います。 */
    vrg_Recog(DicArea_p);
    :
}
```

割り込み初期化のサンプル・プログラムを次に示します。

```
.globl _InputInit
_InputInit:
    _xxICn .set 0xfffffXXX /* xx 割り込み制御レジスタ定義 */

:

/* 音声取り込みのための xx 割り込みは
   マスクされた状態を初期状態とする*/
mov    0xZZ, r6
st.b   r6, _xxICn[r0]

:

jmp   [lp]
```

### (3) vrg\_IntOn 関数

- 【関数名】 vrg\_IntOn
- 【概要】 音声取り込みのための割り込みを許可します。
- 【形式】 void vrg\_IntOn ( void );
- 【引き数】 なし
- 【戻り値】 なし
- 【機能】

#### (a) 動作

音声認識処理中 (vrg\_Recog 関数 : 2. 5. 1 参照) に呼び出され, 音声取り込みのための割り込みを受け付け可能にします。

#### (b) 関数内で記述する内容

割り込み受け付けを可能とするために必要な命令を vrg\_IntOn 関数内に記述してください。

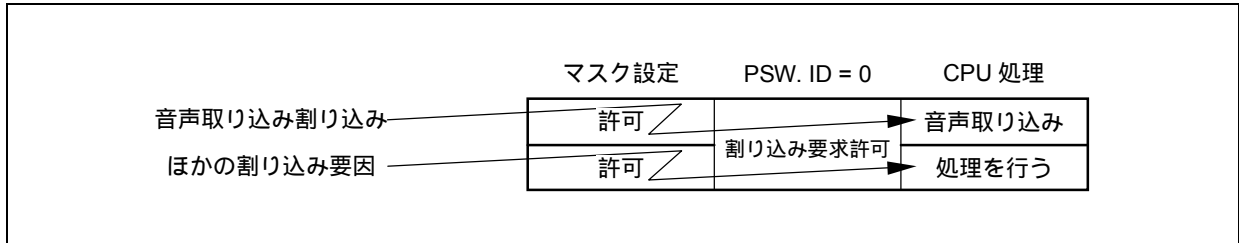
vrg\_IntOn 関数がコールされて初めて音声取り込み割り込みが動作するようなコーディングが必須です。これは, vrg\_Recog 関数動作中のみに音声格納処理 (2. 5. 3 vrg\_InputWaveData 関数, 2. 5. 4 vrg\_InputWaveDataDMA 関数参照) を行うためです。

## (c) コーディング例

vrg\_IntOn 関数でマスクを解除し、音声取り込みのための割り込みが動作します。

図 2 - 14 で、図 2 - 13 に対応した vrg\_IntOff 関数の例を示しています。

図2 - 13 vrg\_IntOn関数発行後の処理



サンプル・プログラムを次に示します。

```

        .globl _vrg_IntOn
_vrg_IntOn:
    _xxICn    .set 0xfffffXXX    /* xx 割り込み制御レジスタ定義 */
    clr1     xxMKn, _xxICn[r0] /* xx 割り込みマスク解除 */
    jmp      [lp]

```

## (4) vrg\_IntOff 関数

- 【関数名】 vrg\_IntOff  
 【概要】 音声取り込みのための割り込みを禁止します。  
 【形式】 void vrg\_IntOff ( void );  
 【引き数】 なし  
 【戻り値】 なし  
 【機能】

## (a) 動作

音声認識処理中 (vrg\_Recog 関数 : 2.5.1 参照) に呼び出され、音声取り込みのための割り込みの受け付けを禁止します。vrg\_Recog 関数は、vrg\_IntOff 関数がコールされた状態で終了します。

## (b) 関数内で記述する内容

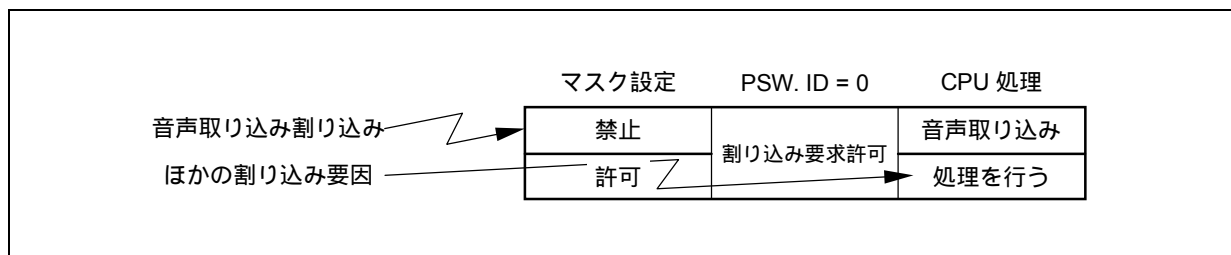
割り込み受け付けを禁止するために必要な命令を vrg\_IntOff 関数内に記述してください。

## (c) コーディング例

次に、2.6.2 (3) vrg\_IntOn 関数で示した例に対応した、割り込み禁止方法の例を示します。

割り込み制御レジスタで割り込みをマスクしています。再度 vrg\_IntOn 関数がコールされると、マスクが解除され再度音声取り込み割り込みが動作できます。

図 2 - 14 vrg\_IntOff 関数発行後の処理



サンプル・プログラムを次に示します。

```
.globl _vrg_IntOff
_vrg_IntOff:
    _xxICn    .set 0xfffffXXX    /* xx 割り込み制御レジスタ定義 */
    setl     xxMKn, _xxICn[r0] /* xx 割り込みにマスクをかける */
    jmp      [lp]
```

**注意** vrg\_IntOff 関数がコールされたあとで、再度 vrg\_IntOn 関数がコールされた場合、再度音声取り込み割り込みが動作できるようなコーディングをしてください。

### 2.6.3 vrg\_RecogReady 関数

- 【関数名】 vrg\_RecogReady  
【概要】 認識前処理終了の通知を行います。  
【形式】 void vrg\_RecogReady ( void );  
【引き数】 なし  
【戻り値】 なし  
【機能】

#### (a) 動作

vrg\_Recog 関数 (2.5.1 参照) 内からコールされます。vrg\_RecogReady 関数がコールされたことにより、認識前処理が終了し、音声データの取得が可能な状態になったことを通知します。

必要に応じてユーザ・コーディングしてください。

vrg\_Recog 関数がコールされてから、実際に音声を取り込み始めるまで、認識をする前処理を行います。したがって厳密には、vrg\_Recog 関数コール直後に発声を始めてしまうと音声の先頭を取りこぼす恐れがあります。これを回避するために、vrg\_RecogReady 関数を利用して発声開始タイミングを知ることができます。

- 注意 1. 前処理段階でエラーが発生した場合には、vrg\_RecogReady 関数はコールされません。すなわち、vrg\_Recog 関数の戻り値が「E\_VRGDIFORMAT」、 「E\_VRGMEMAREASIZE」、 「E\_VRGDICALIGN」、 「E\_VRGVBUFAREASIZE」、 「E\_VRGVOICESHIFT」となった場合、vrg\_RecogReady 関数はコールされません。
2. vrg\_Stop 関数 (2.5.2 参照) を使用して vrg\_Recog 関数を強制終了する場合には、vrg\_RecogReady 関数コールから vrg\_RecogComplete 関数 (2.6.4 参照) コールまでの間にコールしなければなりません。それ以外のタイミングで vrg\_Stop 関数がコールされた場合の動作は保証できません。

## 2.6.4 vrg\_RecogComplete 関数

- 【関数名】 vrg\_RecogComplete  
【概要】 認識動作終了通知を行います。  
【形式】 void vrg\_RecogComplete ( void );  
【引き数】 なし  
【戻り値】 なし  
【機能】

### (a) 動作

vrg\_Recog 関数 (2.5.1 参照) 内でコールされます。認識動作自体は終了したことを通知します。  
vrg\_Stop 関数 (2.5.2 参照) コール可能な範囲は、vrg\_RecogComplete 関数コール時までです。  
必要に応じてユーザ・コーディングしてください。

**注意** 前処理段階でエラーが発生した場合には、vrg\_RecogComplete 関数はコールされません。すなわち、vrg\_Recog 関数の戻り値が「E\_VRGDIFORMAT」, 「E\_VRGMEMAREASIZE」, 「E\_VRGDICALIGN」, 「E\_VRGVBUFAREASIZE」, 「E\_VRGVOICESHIFT」となった場合、vrg\_RecogComplete 関数はコールされません。



## 2.6.5 vrg\_InitParam 関数

- 【関数名】 vrg\_InitParam  
 【概要】 各初期設定を行います。  
 【形式】 void vrg\_InitParam ( void );  
 【引き数】 なし  
 【戻り値】 なし  
 【機能】

## (a) 動作

各初期設定を行うための関数です。

vrg\_Recog 関数内で、設定値を変更すべきタイミングでコールされます。

設定は vrg\_MemArea 変数内の値を直接書き換えることで行います。変更可能な設定とその位置を表 2 - 12 に示します。

表2 - 12 vrg\_InitParamで設定するパラメータの位置

パラメータ	デフォルト値	オフセット	書き込み サイズ	書き込み位置マクロ	指定可能 範囲
連続無音状態 タイムアウト時間	0 (なし)	+ 56 バイト	2 バイト	PARA_VRGTIMEOUT (((short*)vrg_MemArea)[28])	0-1000
連続有音状態 タイムアウト時間	937 (15 秒)	+ 58 バイト	2 バイト	PARA_VRGENDTIMEOUT (((short*)vrg_MemArea)[29])	0-1000

**(b) 連続無音状態タイムアウト設定領域**

連続無音状態タイムアウト機能とは、vrg\_Recog 関数がよばれてからユーザーによって指定された時間まで音声を検出されなかった場合、vrg\_Recog 関数を強制終了する機能のことです。連続無音状態タイムアウト時間の指定は、vrglib.h 内に定義されている PARA\_VRGTIMEOUT に設定することで行います。設定値は 0-1000 整数値で行い、16 ms 周期で設定値まで計測します。そのため最大連続無音状態タイムアウト時間は 16 s (16 ms×1000) となります。

このパラメータを設定する場合、vrg\_InitParam 関数内で vrg\_MemArea 変数の先頭から 56 バイト目に 2 バイトのサイズで設定値を書き込んでください。この書き込み位置とサイズは、vrglib.h ヘッダ・ファイルに PARA\_VRGTIMEOUT として定義してあります。

**(c) 連続有音状態タイムアウト設定領域**

連続有音状態タイムアウト機能とは、音声の始端が検出されてからユーザーによって指定された時間まで音声を検出され続けた場合、vrg\_Recog 関数を強制終了する機能のことです。連続有音状態タイムアウト時間の指定は、vrglib.h 内に定義されている PARA\_VRGENDTIMEOUT に設定することで行います。設定値は 0-1000 整数値で行い、16 ms 周期で設定値まで計測します。そのため最大連続有音状態タイムアウト時間は 16 s (16 ms×1000) となります。

このパラメータを設定する場合、vrg\_InitParam 関数内で vrg\_MemArea 変数の先頭から 58 バイト目に 2 バイトのサイズで設定値を書き込んでください。この書き込み位置とサイズは、vrglib.h ヘッダ・ファイルに PARA\_VRGENDTIMEOUT として定義してあります。

```
#include vrglib.h /* PARA_VRGTIMEOUT & PARA_VRGENDTIMEOUT 定義 */

#define VRGMEMAREASIZE XXXX /* 認識ワークサイズ */
#define VRGBUFSIZE XXXX /* 音声バッファサイズ */
#define VRGSHIFTNUM XXXX /* 音声シフト数 */

long vrg_MemArea[VRGMEMAREASIZE / sizeof(long)];
int vrg_MemAreaSize = VRGMEMAREASIZE;
short vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int vrg_BufSize = VRGBUFSIZE;
int vrg_VoiceShift = VRGSHIFTNUM;

void
vrg_InitParam()
{
    /* 連続無音状態タイムアウト:3 秒 (3*1000/16) */
    PARA_VRGTIMEOUT = 188;

    /* 連続有音状態タイムアウト:4 秒 (4*1000/16) */
    PARA_VRGENDTIMEOUT = 250;
}

main()
{
    .....
}
```

## 第3章 インストール

### 3.1 開発環境へのファイル展開

次の作業を MS-DOS プロンプトなどで行ってください。

- (1) 提供媒体をセットします。ここでは A ドライブにセットしたとします。
- (2) AP703000-B07 のファイル群は自己解凍形式です。Ap850b07.exe を実行し、ファイル群を展開します。  
デフォルトでは C:\NECTools32 ディレクトリに展開する形式となっています。  
C:\>A:\Ap850b07.exe <CR>
- (3) ファイルがインストールされたことを確認します。  
ディレクトリ構成や各ファイル名については 1.3.6 **パッケージ内容**を参照してください。

## 3.2 サンプル・プログラム作成手順

サンプル・プログラム ( sample1.c, sample2.c, sample3.c ) の make 方法を説明します。

### 3.2.1 NEC 製ツールのライブラリとプロジェクト・マネージャを使用する場合

(1) プロジェクト・マネージャを起動します。

Windows のスタートボタンから「プログラム」 「NECTools32」 「プロジェクト・マネージャ」を実行してください。

(2) プロジェクト・マネージャのメニューから「プロジェクト」 「開く」を選択し、sample1.prj、sample2.prj あるいは sample3.prj ファイルを開いてください。

(3) ウィンドウ上部のタイトル・バーで、sample1.prj、sample2.prj あるいは sample3.prj が読み込まれたことを確認します。

sample1.prj

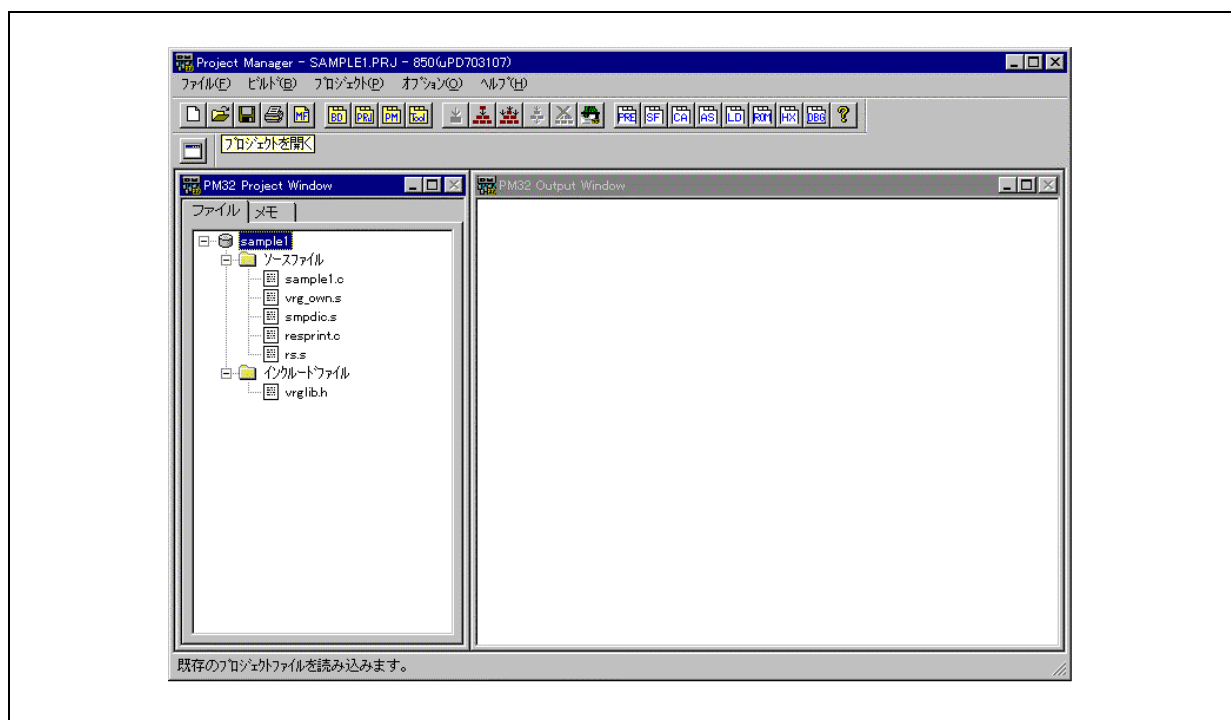
sample2.prj

sample3.prj

(4) プロジェクト・マネージャのメニューから「ビルド」 「0 vrgmkdic」を選択し、サンプル・プログラム ( sample1.out, sample3.out ) に使用する辞書を作成します。

(5) プロジェクト・マネージャのメニューから「ビルド」 「ビルド」を選択しプログラム・ファイル ( sample1.out, sample2.out あるいは sample3.out ) を生成します。

図3-1 プロジェクト・マネージャ画面



## 3.2.2 GHS 社製ツールのライブラリと MULTI のビルダ機能を使用する場合

(1) MULTI を起動します。

multi.exe を実行してください。

(2) MULTI のメニューから「ファイル」「読み込み」を選択し、sample1.bld、sample2.bld あるいは sample3.bld ファイルを開いてください。

(3) プロジェクト・ボタン横のボックスで、sample1.bld、sample2.bld あるいは sample3.bld が読み込まれたことを確認します。

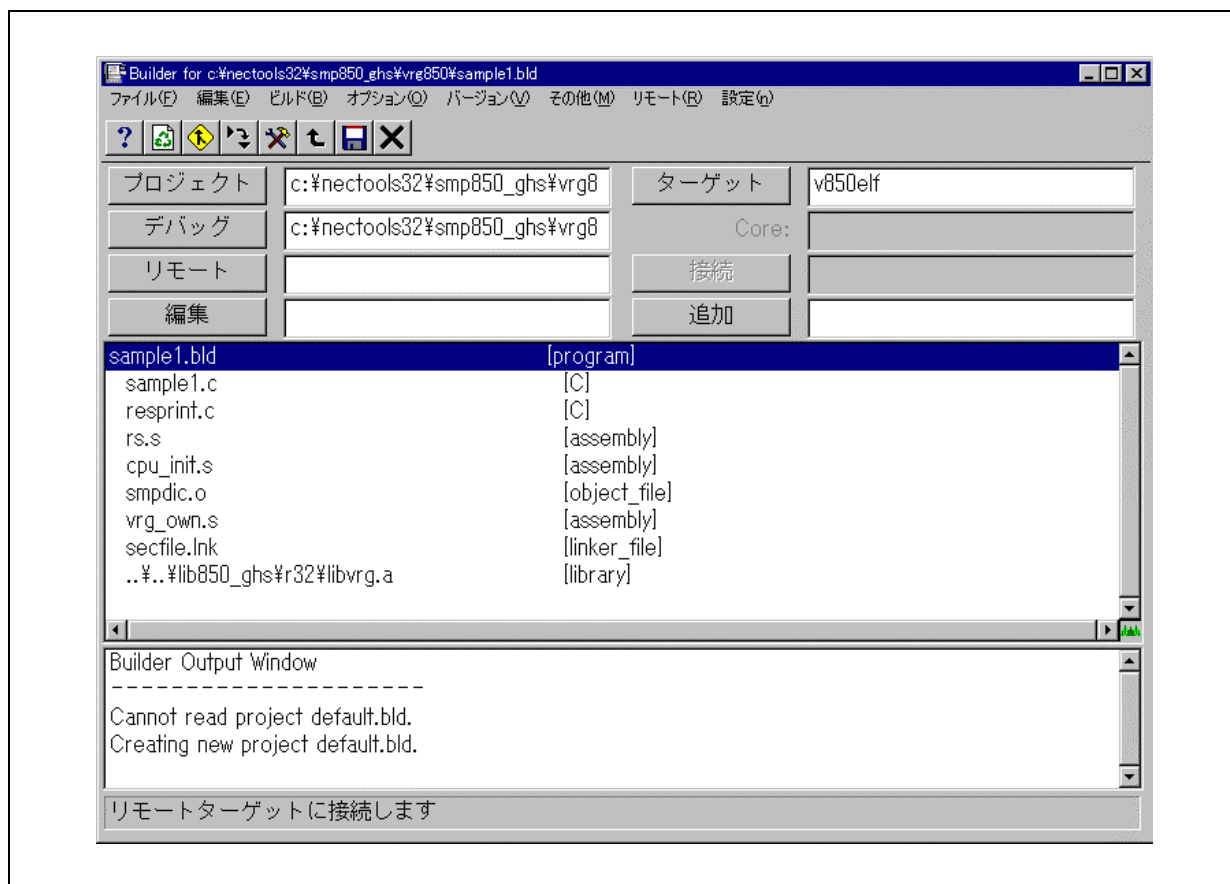
C:\NECTools32\smp850\_ghs\vrg850\sample1.bld

C:\NECTools32\smp850\_ghs\vrg850\sample2.bld

C:\NECTools32\smp850\_ghs\vrg850\sample3.bld

(4) MULTI のメニューから「ビルド」「ビルド」を選択しプログラム・ファイル (sample1.out、sample2.bld あるいは sample3.bld) を生成します。

図3 - 2 MULTI画面



## 3.3 システム構築方法

### 3.3.1 認識単語辞書を PC 版ユーザ辞書作成ツールを使用して作成する場合

#### (1) 認識単語辞書オブジェクト・ファイルの作成

PC 版辞書作成ツールを用いて辞書ファイルを出力します（第4章 PC 版ユーザ辞書作成ツール参照）。

出力された辞書ファイルをリンクに入力できるファイル形式に変換します。変換方法を次に示します。

- ・ NEC 製ツールの場合：アセンブラ擬似命令（.bininclude）を使用
- ・ GHS 製ツールの場合：バイナリ・ファイル ELF ファイル変換ツール（表 1-3 ライブラリのファイル一覧（bin¥vrg850）参照）を使用

#### (2) ユーザ・OWN・コーディング関数の作成

vrg\_InputWave 関数（2.6.1 参照），vrg\_IntOn 関数，vrg\_IntOff 関数（2.6.2 参照）を，使用するハードウェア環境にあわせて作成します。

V850E/MA1 上で動作するサンプル・プログラムを vrg\_own.s として添付しています。

vrg\_RecogReady 関数（2.6.3 参照），vrg\_RecogComplete 関数（2.6.4 参照）を必要に応じて記述します。

#### (3) メイン・プログラムの作成

システムで実現する処理を記述したメイン・プログラムを作成します。添付のファイル（sample1.c）を参照してください。

システム起動に必要となるブート処理プログラムを作成します。添付のファイル（cpu\_init.s）を参照してください。

#### (4) アドレス割り付けの決定

(1) から (3) のプログラムおよびオブジェクトなどのアドレス割り付けを決め，リンク・ディレクティブ・ファイルを作成します。添付のファイル（secfile.dir または secfile.lnk）を参照してください。

#### (5) 実行プログラム・ファイルの生成

(1) から (3) のプログラムおよびオブジェクトなどをコンパイルまたはアセンブルしたあと，AP703000-B07 のライブラリとリンクして実行プログラム・ファイルを生成します。

### 3.3.2 認識単語辞書を vrg\_MakeDic 関数を使用して作成する場合

#### (1) ユーザ・OWN・コーディング関数の作成

vrg\_InputWave 関数, vrg\_IntOn 関数, vrg\_IntOff 関数を, 使用するハードウェア環境にあわせて作成します。

V850E/MA1 上で動作するサンプル・プログラムを vrg\_own.s として添付しています。

vrg\_RecogReady 関数, vrg\_RecogComplete 関数を必要に応じて変更します。

#### (2) メイン・プログラムの作成

システムで実現する処理を記述したメイン・プログラムを作成します。登録する認識単語文字列を vrg\_MakeDic 関数 (2.5.5 参照) に入力する処理を記述します。添付のファイル (sample2.c) を参照してください。

システム起動に必要となるブート処理プログラムを作成します。添付のファイル (cpu\_init.s) を参照してください。

#### (3) アドレス割り付けの決定

(1), (2) のプログラムおよびオブジェクトなどのアドレス割り付けを決め, リンク・ディレクティブ・ファイルを作成します。添付のファイル (secfile.dir または secfile.lnk) を参照してください。

#### (4) 実行プログラム・ファイルの生成

(1), (2) のプログラムおよびオブジェクトなどをコンパイルまたはアセンブルしたあと, AP703000-B07 のライブラリとリンクして実行プログラム・ファイルを生成します。

**注意** スタック領域を確保する際は, メイン・プログラムで使用するもの以外に, ミドルウェアで必要となるスタック量および割り込みハンドラで必要となるスタック量が必ず含まれるようにしてください。

## 第 4 章 PC 版ユーザ辞書作成ツール

PC 版ユーザ辞書作成ツールの使用方法について説明します。

### 4.1 動作環境

Windows95/98/2000 (日本語版のみ) が動作する環境

### 4.2 辞書作成方法

PC 版ユーザ辞書作成ツールを使用する場合、あらかじめ入力フォーマットに従って認識登録単語を記述したファイルが必要です。このファイルを PC 版ユーザ辞書作成ツールにかけると認識単語辞書が作成されます。

次に PC 版ユーザ辞書作成ツールを用いた場合の辞書作成方法を「入力フォーマット」、「PC 版ユーザ辞書作成ツール使用方法」、「出力結果」の順に説明します。

#### 4.2.1 入力フォーマット

半角カナ文字によって指定された文字列を入力することにより、認識単語辞書を作成します。入力フォーマットは図 4 - 1 のようにしてください。

図4 - 1 入力フォーマット

単語	区切り	単語	区切り	...	単語	終端
----	-----	----	-----	-----	----	----

なお、各入力文字の仕様は表 4 - 1 に従ってください。

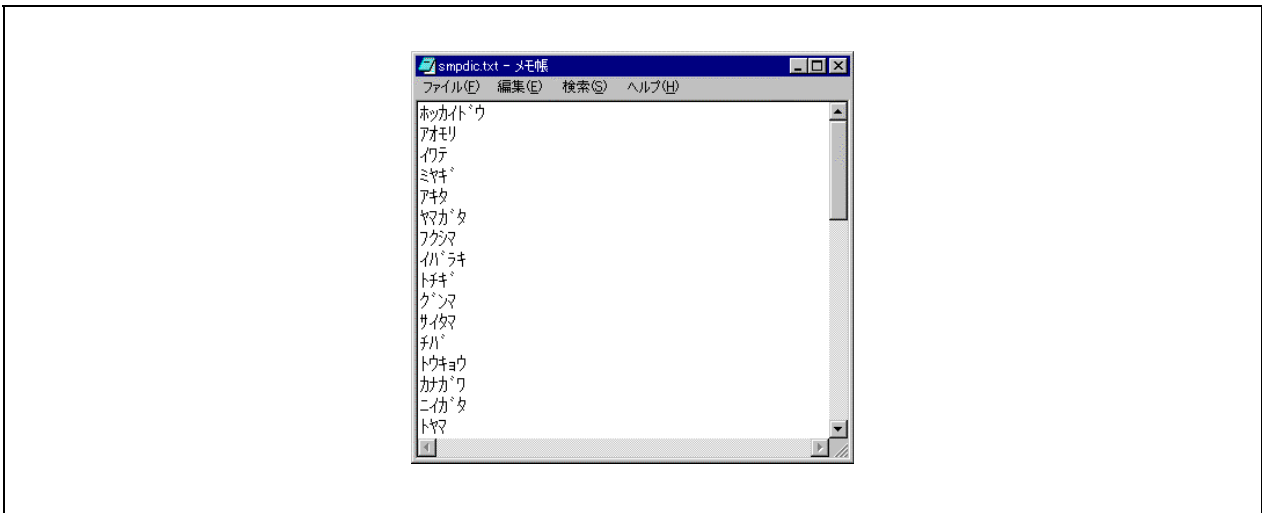
表4 - 1 各入力文字仕様

フォーマット	仕 様
単語	「半角カナ文字」で指定してください。 1 単語の最大文字数は 20 文字です。 使用可能文字については表 2 - 10 を参照してください。
区切り	「半角スペース」または「改行コード」で指定してください。
終端	「EOF」で指定してください (ファイル保存で自動的に付きます)。 なお、直前は「単語」および「区切り」のどちらでも可能です。



次に入力ファイル例を示します。

図4-2 入力ファイル例 (smpdic.txt)



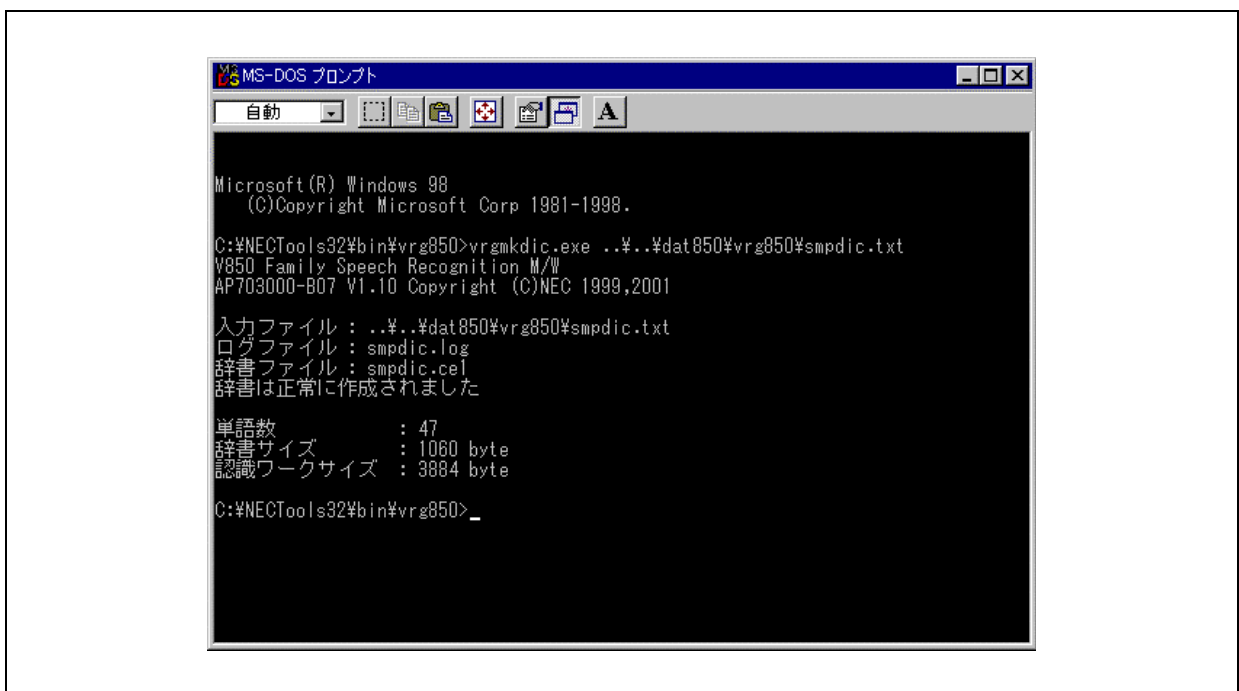
#### 4.2.2 PC版ユーザ辞書作成ツール使用方法

PC版ユーザ辞書作成ツールによる辞書作成は、MS-DOSプロンプト上で次のように入力してください。

```
c:%dictool> vrgmkdic 入力ファイル名
```

次に入力ファイル (smpdic.txt) を図4-2とした場合の実行例を示します。

図4-3 辞書作成例



### 4.2.3 出力結果

辞書作成が成功すると「ログ・ファイル」と「辞書ファイル」が作成されます。ログ・ファイルには「単語数」、「辞書サイズ」、「認識ワーク・サイズ」が出力されますので認識動作時の参考にしてください。

作成される「ログ・ファイル」、「辞書ファイル」は、入力ファイルの拡張子を変更（拡張子がない場合は付加）したファイル名となります。それぞれ、「ログ・ファイル」は「.log」、「辞書ファイル」は「.ce1」の拡張子となります。

**注意** 入力ファイル名に「.log」、「.ce1」の拡張子のファイルを指定しないでください。これらの拡張子がついたファイル名を指定した場合、異常終了し、ファイル内容が上書きされてしまいます。

表4-2 出力結果

出 力	内 容
ログ・ファイル (*.log)	テキスト形式のファイルです。 辞書作成結果のログが書き込まれています。なお、ログ・ファイルはカレント・ディレクトリに出力されます。
辞書ファイル (*.ce1)	バイナリ形式のファイルです。 辞書データが格納されています。なお、辞書ファイルはカレント・ディレクトリに出力されます。
単語数	登録単語数
辞書サイズ	辞書ファイル (*.ce1) のサイズ (バイト)
認識ワーク・サイズ	vrg_MemArea に必要な認識ワーク・サイズ

### 4.2.4 エラー・コード

入力された文字列に異常がある場合、認識単語辞書を正常に作成できません。次のようなフォーマットでエラー・メッセージが表示されます。

ID: {エラーが発生した単語の ID 番号} {エラー・メッセージ}

単語の ID 番号の規約については、表 2-6 を参照してください。

エラー発生時のメッセージとその原因を表 4-3 に示します。

表4-3 エラー・メッセージ

メッセージ	原 因
使用されている文字が不正です	半角カナ以外の文字コードが発見されました。
単語文字数が規定数を超過しています	1単語の文字数が20文字を越えています。
認識に使用できない大きさの認識単語辞書を作成しようとした	登録単語数、文字数が多すぎます (仕様範囲外です)。
予測されないエラーが発生しました	通常は発生しません。
辞書領域が不足しています	通常は発生しません。

## 第5章 チェック・サム機能

認識単語辞書中に組み込まれているチェック・サムについて説明します。

### 5.1 チェック・サム概要

辞書データ内に格納されたチェック・サムを用いて、データが破壊されていないかを調べることができます。たとえばチェック・サムを使用することにより認識単語辞書データがダウンロード時に破壊されていないかを確認できます。

ただし、ライブラリ側ではチェック・サムの確認は行っていません。ユーザ・プログラム内で必要に応じて確認を行ってください。

### 5.2 チェック・サム使用方法

チェック・サムを使用する場合は、次のコードを用いてチェックしてください。

- 【関数名】 check\_sum
- 【概要】 認識単語辞書のデータ破壊をチェックします（処理内容を公開しています）。
- 【形式】 int check\_sum(short \*DicArea\_p, long DicSize);
- 【引き数】 [ ユーザ入力変数 ] \*DicArea\_p : 認識単語辞書格納領域先頭アドレス  
[ ユーザ入力変数 ] DicSize : 認識単語辞書サイズ（バイト）
- 【戻り値】

表5 - 1 check\_sum関数戻り値一覧

戻り値	内 容
0	正常終了
1	認識単語辞書データが破壊されています。 または引き数が誤っています。

## 【コード】

```
#define SUCCESS      0x0
#define ERROR        0x1

int
check_sum(short *DicArea_p, long DicSize)
{
    long    i;
    long    count;
    short   checksum;

    count = DicSize / sizeof(short);

    checksum = 0;

    for (i=0; i<count; i++){
        checksum ^= *DicArea_p++;
    }

    if (checksum == 0){
        return SUCCESS;
    }
    else{
        return ERROR;
    }
}
```

## 第6章 ロケーションおよびシンボル名規約

### 6.1 ロケーションの変更

AP703000-B07 では、次のセクション名が付けられています。

表6 - 1 セクション名一覧

セクション名	タイプ	内 容
.vrg_text	.text	プログラム
.vrg_const	.const (.rodata)	テーブル・データ
.bss	.bss	RAM 領域データ

AP703000-B07 では、グローバル変数としていくつかの変数を持ちます。グローバル変数参照時には、同一セクション名でないとお互いに参照できません。そのため、AP703000-B07 の RAM 領域データはすべてセクション名が .bss となるように設定してください。

### 6.2 シンボル名規約

AP703000-B07 で使用するグローバル・シンボル名は、すべて先頭が vrg\_ (アセンブリ言語では \_vrg\_) から始まるように命名されています。ほかのアプリケーションを組み合わせで使用するときには、重複しないようにしてください。

## 付録 A サンプル・ソース (NEC 版, V850E/MA1 の場合)

## A.1 認識処理サンプル・ソース (sample1.c)

```
/*
 * Sample for Ultalker-C
 *
 * Copyright (C) NEC Corporation 2001
 * All rights reserved by NEC Corporation.
 *
 * This is a sample program.
 * NEC assumes no responsibility for any losses incurred by customers or
 * third parties arising from the use of this program.
 *
 * sample1.c
 */

/* 音声認識ライブラリのヘッダをインクルードします。 */
#include "vrglib.h"

/*
 * 「認識ワークサイズ」
 * 辞書作成ツールから得られる認識ワークサイズ以上確保してください。
 * (ここでリンクする辞書"smpdic.ce1"の場合, 認識ワークサイズは 3884byte です。)
 */
#define VRGMEMAREASIZE 4000

/*
 * 「音声バッファサイズ」
 * 通常は 768byte としてください。
 */
#define VRGBUFSIZE 768

/*
 * 「音声シフト数」
 * V850E/MA1 内蔵 A/D コンバータを使用する場合は, 6 ビット左にシフトする必要があります。
 */
#define VRGVOICESHIFT 6

/* vrg_MemArea[]の宣言 */
long vrg_MemArea[ VRGMEMAREASIZE/sizeof(long) ];
int vrg_MemAreaSize = VRGMEMAREASIZE;

/* 音声バッファの宣言 */
short vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int vrg_BufSize = VRGBUFSIZE;
```

```

/* 音声シフト数 */
int      vrg_VoiceShift = VRGVOICESHIFT;

/*
 * 辞書格納領域
 * bin2elf.exe を使用して辞書作成ツールで作成した辞書をリンクします。
 */
extern const short DicArea[];

/* =====
 * (1) 関数名      : application()
 * (2) 処理内容   : 認識結果を使用するアプリケーション
 * (3) 引数       : int      ret      : vrg_Recog() の返却値
 *                : short *result : vrg_MemArea 先頭アドレス
 *                :                short* にキャスト)
 * (4) 戻り値    : なし
 * ===== */
void application(int ret, short *result)
{
#ifdef __RS232C__
    /* 認識動作の結果を表示します。 */
    result_print(ret, result);
#endif

    return;
}

/* =====
 * (1) 関数名      : vrg_RecogReady()
 * (2) 処理内容   : 認識前処理終了を示すフック関数
 *                : (vrg_Recog() 内からコールされます。)
 * (3) 引数       : なし
 * (4) 戻り値    : なし
 * ===== */
void vrg_RecogReady(void)
{
#ifdef __RS232C__
    /* RS-232C を使用して, 「発声をどうぞ」と表示します。 */
    recog_ok();
#endif
}

/* =====
 * (1) 関数名      : main()
 * (2) 処理内容   : main 関数
 * (3) 引数       : なし
 * (4) 戻り値    : 異常終了の場合のみ-1 を返却します。
 * ===== */
int main(void)
{

```

```

int ret;

/* 割り込み禁止 */
__asm("di");

#ifdef __RS232C__
/*
 * 表示のための RS-232C を初期化します。
 * 受信側を以下のように設定してください。
 *
 * 送受信状態      : 送受信許可
 * パリティビット  : パリティなし拡張ビット動作
 * キャラクタ長    : 8 ビット
 * ストップビット長 : 1 ビット
 * シリアルクロック : ポーレートジェネレータ出力
 */
rsinit(38400);
#endif

/*
 * A/D コンバータおよびタイマの初期化
 *
 * 音声取り込みに使用する以下の資源の初期化を行います。
 * ・タイマ      TMC0, CCC00
 * ・A/D コンバータ ADCR0
 * ・割り込み    INTP000
 *
 * 注) 割り込みはマスク状態で初期化していますので, ei 命令によって割り込みを許可しても,
 *     vrg_InputWave() がいきなり動作することはありません。
 *     vrg_Recog() 内からコールされる vrg_IntOn() ・ vrg_IntOff() によってマスク状態を変化
 *     させることによって, vrg_InputWave をミドルウェアから制御します。
 *     ・vrg_IntOn() : マスクを解除 (割り込み許可)
 *     ・vrg_IntOff() : マスクをかける (割り込み禁止)
 */
InputInit();
/* 割り込み許可 */
__asm("ei");

/* 認識を繰り返します。 */
while(1){

    /* 1 発声分の認識処理を行います。 */
    ret = vrg_Recog((short*)DicArea);
    if (ret < 0){
        /* vrg_Recog 異常終了 */
    }
}

#ifdef __RS232C__
/* エラーコード表示 */
result_print(ret, (short *)vrg_MemArea);
#endif

/* タイムアウト以外のエラーの場合, ループを抜け終了します。 */
if (ret != E_VRGENDTIMEOUT){

```



```
        return -1;
    }

}

else{
    /* 認識結果を使用するアプリケーション */
    /* 認識結果は, vrg_Recog() の返却値または, vrg_MemArea[] の先頭 20byte から取り出せます。 */
    application(ret, (short*)vrg_MemArea);
}

}

}
```

## A.2 辞書作成サンプル・ソース (sample2.c)

```

/*
 * Sample for Ultalker-C
 *
 * Copyright (C) NEC Corporation 2001
 * All rights reserved by NEC Corporation.
 *
 * This is a sample program.
 * NEC assumes no responsibility for any losses incurred by customers or
 * third parties arising from the use of this program.
 *
 * sample2.c
 */

/* 音声認識ライブラリのヘッダをインクルードします。 */
#include "vrglib.h"

/*
 * 「認識ワークサイズ」
 * 辞書作成ツールから得られる認識ワークサイズ以上確保してください。
 * (ここで作成する辞書の場合、認識ワークサイズは 3884byte です。)
 */
#define VRGMEMAREASIZE 4000

/*
 * 「音声バッファサイズ」
 * 通常は 768byte としてください。
 */
#define VRGBUFSIZE 768

/*
 * 「音声シフト数」
 * V850E/MA1 内蔵 A/D コンバータを使用する場合は、6 ビット左にシフトする必要があります。
 */
#define VRGVOICESHIFT 6

/*
 * 「辞書格納領域サイズ」
 * 下記の DicText[] の内容で辞書を作成すると、作成される辞書のサイズは 1060 バイトになります。
 * ここでは 1200byte 確保します。
 */
#define DICAREASIZE 1200

/* vrg_MemArea[] の宣言 */
long vrg_MemArea[ VRGMEMAREASIZE/sizeof(long) ];
int vrg_MemAreaSize = VRGMEMAREASIZE;

/* 音声バッファの宣言 */
short vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];

```

```

Int      vrg_BufSize = VRGBUFSIZE;

/* 音声シフト数 */
int      vrg_VoiceShift = VRGVOICESHIFT;

/* この内容で辞書を作成し、認識に使用します。 */
const char DicText[] = {"カキドウアオリ イテ ミヤキ アキ ヤガタ フクシ 八`ネ トキ` ゲ`ム サイ`ス 八` トキヨウ カガ`リ ニガ`タ トマ`イ
加` フキ` ヤマシ` ガ`ノ`ギ`フ`シ`カ`ア`イ`ミ`シ`キ`ヨウ`オ`カ`カ`ヒョウ`ラ`ワ`カ`マ`ト`リ`シ`ネ`オ`カ`マ`ヒ`ロ`シ`ヤ`マ`チ`ト`ク`シ`カ`ガ`リ`エ`ム`コ`ウ`フ`ク`カ`サ`ガ`
サ`ク`マ`ト`オ`イ`タ`ミ`ガ`キ`カ`シ`サ`サ`サ`0"};

/*
 * 「辞書格納領域」
 * この領域に辞書を作成します。
 */
short    DicArea[DicText/sizeof(short)];

/* =====
 * (1)関数名      : application()
 * (2)処理内容    : 認識結果を使用するアプリケーション
 * (3)引数        : int ret      : vrg_Recog()の返却値
 *                : short *result : vrg_MemArea 先頭アドレス
 *                : short*にキャスト)
 * (4)戻り値      : なし
 * ===== */
void application(int ret, short *result)
{
#ifdef __RS232C__
    /* 認識動作の結果を表示します。 */
    result_print(ret, result);
#endif

    return;
}

/* =====
 * (1)関数名      : vrg_RecogReady()
 * (2)処理内容    : 認識前処理終了を示すフック関数
 *                ( vrg_Recog() 内からコールされます。 )
 * (3)引数        : なし
 * (4)戻り値      : なし
 * ===== */
void vrg_RecogReady(void)
{
#ifdef __RS232C__
    /* RS-232C を使用して、「発声をどうぞ」と表示します。 */
    recog_ok();
#endif
}

```

```

/* =====
* (1)関数名      :main()
* (2)処理内容   :main 関数
* (3)引数       :なし
* (4)戻り値     :異常終了の場合のみ-1を返却します。
* ===== */
int main(void)
{
    int      ret;
    long     dicsize,needrecogworksize;

    /* 割り込み禁止 */
    __asm("di");

#ifdef __RS232C__
    /*
     * 表示のための RS-232C を初期化します。
     * 受信側を以下のように設定してください。
     *
     * 送受信状態      :送受信許可
     * パリティビット  :パリティなし拡張ビット動作
     * キャラクタ長     :8ビット
     * ストップビット長 :1ビット
     * シリアルクロック :ボーレートジェネレータ出力
     */
    rsinit(38400);
#endif

    /*
     * A/D コンバータおよびタイマの初期化
     *
     * 音声取り込みに使用する以下の資源の初期化を行います。
     * ・タイマ      TMC0,CCC00
     * ・A/D コンバータ ADCR0
     * ・割り込み    INTP000
     * 注) 割り込みはマスク状態で初期化していますので、ei 命令によって割り込みを許可しても、
     *     vrg_InputWave() がいきなり動作することはありません。
     *     vrg_Recog() 内からコールされる vrg_IntOn()・vrg_IntOff() によってマスク状態を変化
     *     させることによって、vrg_InputWave をミドルウェアから制御します。
     *     ・vrg_IntOn() :マスクを解除(割り込み許可)
     *     ・vrg_IntOff():マスクをかける(割り込み禁止)
     */
    InputInit();

    /* 割り込み許可 */
    __asm("ei");

    /* 辞書作成 */
    ret = vrg_MakeDic((char*)DicText,DicArea,DICAREASIZE);
    if (ret<0){

```

```

/* 辞書作成失敗 */

#ifdef __RS232C__
/* 辞書作成結果を表示します。 */
MakeDic_Result_Print(ret, vrg_MemArea);
#endif

/* 終了します。 */
return -1;
}

#ifdef __RS232C__
/* 辞書作成結果を表示します。 */
MakeDic_Result_Print(ret, vrg_MemArea);
#endif

/* vrg_MemArea 先頭に格納されている辞書作成結果の情報を取得します。 */
dicsize = vrg_MemArea[0];
needrecogworksize = vrg_MemArea[1];

/*
* 辞書は RAM 上に配置されているので、認識の前にチェックサムを使用して
* 辞書が正常であることを確認します。
*/
ret = check_sum(DicArea, dicsize);
if (ret != 0) {
/* チェックサム計算結果がエラー */
#ifdef __RS232C__
/* チェック・サム計算結果がエラーである旨を通知します。 */
checksum_error();
#endif
/* 終了します。 */
return -1;
}

/* 認識を繰り返します。 */
while(1) {

/* 1 発声分の認識処理を行います。 */
ret = vrg_Recog(DicArea);
if (ret < 0) {
/* vrg_Recog 異常終了 */

#ifdef __RS232C__
/* エラーコード表示 */
result_print(ret, (short *)vrg_MemArea);
#endif

/* タイムアウト以外のエラーの場合、ループを抜け終了します。 */
if (ret != E_VRGENDTIMEOUT) {
return -1;
}
}
}
}

```

```
    }  
  
    }  
else{  
    /* 認識結果を使用するアプリケーション */  
    /* 認識結果は, vrg_Recog() の返却値または, vrg_MemArea[] の先頭 20byte から取り出せます。 */  
    application(ret, (short*)vrg_MemArea);  
}  
}  
}
```

## A. 3 認識処理サンプル・ソース (sample3.c)

```

/*
 * Sample for Ultalker-C
 *
 * Copyright (C) NEC Corporation 2001
 * All rights reserved by NEC Corporation.
 *
 * This is a sample program.
 * NEC assumes no responsibility for any losses incurred by customers or
 * third parties arising from the use of this program.
 *
 * sample3.c
 */

/* 音声認識ライブラリのヘッダをインクルードします。 */
#include "vrglib.h"

/*
 * 「認識ワークサイズ」
 * 辞書作成ツールから得られる認識ワークサイズ以上確保してください。
 * (ここでリンクする辞書"smpdic.cel"の場合、認識ワークサイズは 3884byte です。)
 */
#define VRGMEMAREASIZE 4000

/*
 * 「音声バッファサイズ」
 * 通常は 768byte としてください。
 */
#define VRGBUFSIZE 768

/*
 * 「音声シフト数」
 * V850E/MA1 内蔵 A/D コンバータを使用する場合は、6 ビット左にシフトする必要があります。
 */
#define VRGVOICESHIFT 6

/* vrg_MemArea[]の宣言 */
long vrg_MemArea[ VRGMEMAREASIZE/sizeof(long) ];
int vrg_MemAreaSize = VRGMEMAREASIZE;

/* 音声バッファの宣言 */
short vrg_VoiceBuf[VRGBUFSIZE/sizeof(short)];
int vrg_BufSize = VRGBUFSIZE;

/* 音声シフト数 */
int vrg_VoiceShift = VRGVOICESHIFT;

/*
 * 辞書格納領域
 * bin2elf.exe を使用して辞書作成ツールで作成した辞書をリンクします。

```

```

*/
extern const short DicArea[];

/* =====
* (1)関数名      : application()
* (2)処理内容    : 認識結果を使用するアプリケーション
* (3)引数        : int ret      : vrg_Recog() の返却値
*                  : short *result : vrg_MemArea 先頭アドレス
*                  :                  (short*にキャスト)
* (4)戻り値      : なし
* ===== */
void application(int ret, short *result)
{
#ifdef __RS232C__
    /* 認識動作の結果を表示します。 */
    result_print(ret, result);
#endif

    return;
}

/* =====
* (1)関数名      : vrg_RecogReady()
* (2)処理内容    : 認識前処理終了を示すフック関数
*                  : (vrg_Recog() 内からコールされます。)
* (3)引数        : なし
* (4)戻り値      : なし
* ===== */
void vrg_RecogReady(void)
{

    /* DMA 初期化 */
    DMAInit();

    /*
    * タイマスタート
    * A/D 変換はタイマ・トリガ・モードで行います。
    * A/D 変換が終了すると DMA により音声は 1 サンプル転送されます。
    */

    StartTimer();

#ifdef __RS232C__
    /* RS-232C を使用して、「発声をどうぞ」と表示します。 */
    recog_ok();
#endif
}

```



```

/* =====
* (1) 関数名      : vrg_RecogComplete()
* (2) 処理内容   : 認識後処理終了を示すフック関数
*                ( vrg_Recog() 内からコールされます。 )
* (3) 引数      : なし
* (4) 戻り値    : なし
* ===== */
void vrg_RecogComplete(void)
{
    /* タイマストップ */
    StopTimer();
}

/* =====
* (1) 関数名      : vrg_InitParam()
* (2) 処理内容   : パラメータをセットします。
*                ( vrg_Recog からコールされます。 )
* (3) 引数      : なし
* (4) 戻り値    : なし
* ===== */
void vrg_InitParam(void)
{
    PARA_VRGTIMEOUT = 188; /* 3sec timeout (3*1000/16 = 188) */
    PARA_VRGENDTIMEOUT = 250; /* 4sec timeout (4*1000/16 = 250) */
}

/* =====
* (1) 関数名      : main()
* (2) 処理内容   : main 関数
* (3) 引数      : なし
* (4) 戻り値    : 異常終了の場合のみ-1 を返却します。
* ===== */
int main(void)
{
    int ret;

    /* 割り込み禁止 */
    __asm("di");

#ifdef __RS232C__
    /*
     * 表示のための RS-232C を初期化します。
     * 受信側を以下のように設定してください。
     *
     * 送受信状態      : 送受信許可
     * パリティビット  : パリティなし拡張ビット動作
     * キャラクタ長    : 8 ビット
     * ストップビット長 : 1 ビット
     * シリアルクロック : ボーレートジェネレータ出力
     */

```

```

rsinit(38400);
#endif

/*
 *   A/D コンバータおよびタイマの初期化
 *
 *   音声取り込みに使用する以下の資源の初期化を行います。
 *   ・ タイマ           TMC0, CCC00
 *   ・ A/D コンバータ  ADCR0
 *   ・ 割り込み         DMAIC0
 *   注) 割り込みはマスク状態で初期化していますので, ei 命令によって割り込みを許可しても,
 *       vrg_InputWave() がいきなり動作することはありません。
 *       vrg_Recog() 内からコールされる vrg_IntOn() ・ vrg_IntOff() によってマスク状態を変化
 *       させることによって, vrg_InputWave をミドルウェアから制御します。
 *       ・ vrg_IntOn()   : マスクを解除 (割り込み許可)
 *       ・ vrg_IntOff()  : マスクをかける (割り込み禁止)
 */
InputInit();

/* 割り込み許可 */
__asm("ei");

/* 認識を繰り返します。 */
while(1){

    /* 1 発声分の認識処理を行います。 */
    ret = vrg_Recog((short*)DicArea);
    if (ret < 0){
        /* vrg_Recog 異常終了 */

#ifdef __RS232C__
        /* エラーコード表示 */
        result_print(ret, (short *)vrg_MemArea);
#endif

        /* タイムアウト以外のエラーの場合, ループを抜け終了します。 */
        if (ret != E_VRGENDTIMEOUT){
            return -1;
        }
    }
    else{
        /* 認識結果を使用するアプリケーション */
        /* 認識結果は, vrg_Recog() の返却値または, vrg_MemArea[] の先頭 20byte から取り出せます。 */
        application(ret, (short*)vrg_MemArea);
    }
}
}

```

## A.4 スタートアップ・ルーチン (cpu\_init.s)

```
--
-- Sample for Ultalker-C
--
-- Copyright (C) NEC Corporation 2001
-- All rights reserved by NEC Corporation.
--
-- This is a sample program.
-- NEC assumes no responsibility for any losses incurred by customers or
-- third parties arising from the use of this program.
--
-- cpu_init.s
--

-- interrupt vectors

.section ".INTRPT",text
.align 4
.globl __reset
__reset:
    jr    __start

.org 0x10
.globl __nmi
__nmi:
    reti

.org 0x40
.globl __trap00
__trap00:
    reti

.org 0x50
.globl __trap10
__trap10:
    reti

.org 0x60
.globl __ilgop
__ilgop:
    reti

.org 0x80
.globl __intov00
__intov00:
    reti

.org 0x90
.globl __intov01
__intov01:
```

```
    reti

    .org 0xa0
    .globl __intov02
__intov02:
    reti

    .org 0xb0
    .globl __intov03
__intov03:
    reti

    .org 0xc0
    .globl __intp000
__intp000:
    jr _vrg_InputWave

    .org 0xd0
    .globl __intp001
__intp001:
    reti

    .org 0xe0
    .globl __intp010
__intp010:
    reti

    .org 0xf0
    .globl __intp011
__intp011:
    reti

    .org 0x100
    .globl __intp020
__intp020:
    reti

    .org 0x110
    .globl __intp021
__intp021:
    reti

    .org 0x120
    .globl __intp030
__intp030:
    reti

    .org 0x130
    .globl __intp031
__intp031:
    reti
```

```
.org 0x140
.globl __intp100
__intp100:
    reti

.org 0x150
.globl __intp101
__intp101:
    reti

.org 0x160
.globl __intp102
__intp102:
    reti

.org 0x170
.globl __intp103
__intp103:
    reti

.org 0x180
.globl __intp110
__intp110:
    reti

.org 0x190
.globl __intp111
__intp111:
    reti

.org 0x1a0
.globl __intp112
__intp112:
    reti

.org 0x1b0
.globl __intp113
__intp113:
    reti

.org 0x1c0
.globl __intp120
__intp120:
    reti

.org 0x1d0
.globl __intp121
__intp121:
    reti

.org 0x1e0
.globl __intp122
```

```
__intp122:
    reti

    .org 0x1f0
    .globl __intp123
__intp123:
    reti

    .org 0x200
    .globl __intp130
__intp130:
    reti

    .org 0x210
    .globl __intp131
__intp131:
    reti

    .org 0x220
    .globl __intp132
__intp132:
    reti

    .org 0x230
    .globl __intp133
__intp133:
    reti

    .org 0x240
    .globl __intcmd0
__intcmd0:
    reti

    .org 0x250
    .globl __intcmd1
__intcmd1:
    reti

    .org 0x260
    .globl __intcmd2
__intcmd2:
    reti

    .org 0x270
    .globl __intcmd3
__intcmd3:
    reti

    .org 0x280
    .globl __intdma0
__intdma0:
    reti
```

```
.org 0x290
.globl __intdma1
__intdma1:
    reti

.org 0x2a0
.globl __intdma2
__intdma2:
    reti

.org 0x2b0
.globl __intdma3
__intdma3:
    reti

.org 0x2c0
.globl __intcsi0
__intcsi0:
    reti

.org 0x2d0
.globl __intser0
__intser0:
    reti

.org 0x2e0
.globl __intsr0
__intsr0:
    reti

.org 0x2f0
.globl __intst0
__intst0:
    reti

.org 0x300
.globl __intcsi1
__intcsi1:
    reti

.org 0x310
.globl __intser1
__intser1:
    reti

.org 0x320
.globl __intsr1
__intsr1:
    reti

.org 0x330
```

```

    .globl  __intst1
__intst1:
    reti

    .org 0x340
    .globl  __intcsi2
__intcsi2:
    reti

    .org 0x350
    .globl  __intser2
__intser2:
    reti

    .org 0x360
    .globl  __intsr2
__intsr2:
    reti

    .org 0x370
    .globl  __intst2
__intst2:
    reti

    .org 0x380
    .globl  __intad
__intad:
    reti

-- =====
-- function : __start
-- =====

.section ".START",text
.align 2

-- stack memory size
.set  _StackSize, 0x200

.globl  __start
.globl  __startend
.extern _main
__start:

.set  _PHCMD, 0xfffff800
.set  _CKC, 0xfffff822
.set  _VSWC, 0xfffff06e

mov  0x13, r6
st.b r6,  _VSWC[r0]  -- set VSWC (50MHz)

```



```

mov 0x000000a0,r6
ldsr r6, 5 -- set PSW NP 1,ID 1
mov 0x07, r6 -- CKC value 0x07
st.b r6, _PHCMD[r0] -- CKC protection OFF
st.b r6, _CKC[r0] -- set CKC
nop -- dummy NOP
nop -- dummy NOP
nop -- dummy NOP
nop -- dummy NOP
nop -- dummy NOP
mov 0x00000020,r6
ldsr r6, 5 -- reset PSW ID 1

----- sp set -----
mov (#_Stack+_StackSize), sp

----- gp set -----
mov #__gp_DATA, gp

----- tp set -----
mov #__tp_TEXT, tp

jarl _main, lp

-- =====
-- function : __startend
-- =====
__startend:
halt
jr __startend

-- =====
-- variable : _Stack
-- =====
.bss
.section ".stack",bss
.globl _Stack
.lcomm _Stack,_StackSize,4

```

## A.5 スタートアップ・ルーチン (cpu\_init\_dma.s)

```
--
-- Sample for Ultalker-C
--
-- Copyright (C) NEC Corporation 2001
-- All rights reserved by NEC Corporation.
--
-- This is a sample program.
-- NEC assumes no responsibility for any losses incurred by customers or
-- third parties arising from the use of this program.
--
-- cpu_init_dma.s for V850E/MA1
--

-- interrupt vectors

.section ".INTRPT",text
.align 4
.globl __reset
__reset:
    jr    __start

.org 0x10
.globl __nmi
__nmi:
    reti

.org 0x40
.globl __trap00
__trap00:
    reti

.org 0x50
.globl __trap10
__trap10:
    reti

.org 0x60
.globl __ilgop
__ilgop:
    reti
```

```
.org 0x80
.globl __intov00
__intov00:
    reti

.org 0x90
.globl __intov01
__intov01:
    reti

.org 0xa0
.globl __intov02
__intov02:
    reti

.org 0xb0
.globl __intov03
__intov03:
    reti

.org 0xc0
.globl __intp000
__intp000:
    reti

.org 0xd0
.globl __intp001
__intp001:
    reti

.org 0xe0
.globl __intp010
__intp010:
    reti

.org 0xf0
.globl __intp011
__intp011:
    reti

.org 0x100
.globl __intp020
```

```
__intp020:
    reti

    .org 0x110
    .globl __intp021
__intp021:
    reti

    .org 0x120
    .globl __intp030
__intp030:
    reti

    .org 0x130
    .globl __intp031
__intp031:
    reti

    .org 0x140
    .globl __intp100
__intp100:
    reti

    .org 0x150
    .globl __intp101
__intp101:
    reti

    .org 0x160
    .globl __intp102
__intp102:
    reti

    .org 0x170
    .globl __intp103
__intp103:
    reti

    .org 0x180
    .globl __intp110
__intp110:
    reti
```

```
.org 0x190
.globl __intp111
__intp111:
    reti

.org 0x1a0
.globl __intp112
__intp112:
    reti

.org 0x1b0
.globl __intp113
__intp113:
    reti

.org 0x1c0
.globl __intp120
__intp120:
    reti

.org 0x1d0
.globl __intp121
__intp121:
    reti

.org 0x1e0
.globl __intp122
__intp122:
    reti

.org 0x1f0
.globl __intp123
__intp123:
    reti

.org 0x200
.globl __intp130
__intp130:
    reti

.org 0x210
.globl __intp131
__intp131:
```

```
    reti

    .org 0x220
    .globl __intp132
__intp132:
    reti

    .org 0x230
    .globl __intp133
__intp133:
    reti

    .org 0x240
    .globl __intcmd0
__intcmd0:
    reti

    .org 0x250
    .globl __intcmd1
__intcmd1:
    reti

    .org 0x260
    .globl __intcmd2
__intcmd2:
    reti

    .org 0x270
    .globl __intcmd3
__intcmd3:
    reti

    .org 0x280
    .globl __intdma0
__intdma0:
    jr _vrg_InputWave

    .org 0x290
    .globl __intdma1
__intdma1:
    reti

    .org 0x2a0
```

```
.globl __intdma2
__intdma2:
    reti

.org 0x2b0
.globl __intdma3
__intdma3:
    reti

.org 0x2c0
.globl __intcsi0
__intcsi0:
    reti

.org 0x2d0
.globl __intser0
__intser0:
    reti

.org 0x2e0
.globl __intsr0
__intsr0:
    reti

.org 0x2f0
.globl __intst0
__intst0:
    reti

.org 0x300
.globl __intcsi1
__intcsi1:
    reti

.org 0x310
.globl __intser1
__intser1:
    reti

.org 0x320
.globl __intsr1
__intsr1:
    reti
```

```
.org 0x330
.globl __intst1
__intst1:
    reti

.org 0x340
.globl __intcsi2
__intcsi2:
    reti

.org 0x350
.globl __intser2
__intser2:
    reti

.org 0x360
.globl __intsr2
__intsr2:
    reti

.org 0x370
.globl __intst2
__intst2:
    reti

.org 0x380
.globl __intad
__intad:
    reti

-- =====
-- function : __start
-- =====

.section ".START",text
.align 2

-- stack memory size
.set    __StackSize, 0x200

.globl __start
```



```

.globl __startend
.extern _main
__start:

.set _PHCMD, 0xfffff800
.set _CKC, 0xfffff822
.set _VSWC, 0xfffff06e

mov 0x13, r6
st.b r6, _VSWC[r0] -- set VSWC (50MHz)

mov 0x000000a0,r6
ldsr r6, 5 -- set PSW NP 1, ID 1
mov 0x07, r6 -- CKC value 0x07
st.b r6, _PHCMD[r0] -- CKC protection OFF
st.b r6, _CKC[r0] -- set CKC
nop -- dummy NOP
nop -- dummy NOP
nop -- dummy NOP
nop -- dummy NOP
nop -- dummy NOP
mov 0x00000020,r6
ldsr r6, 5 -- reset PSW ID 1

----- sp set -----
mov (#_Stack+_StackSize), sp

----- gp set -----
mov #__gp_DATA, gp

----- tp set -----
mov #__tp_TEXT, tp

jarl _main, lp

-----
-- function : __startend
-----

__startend:
halt
jr __startend

```

```
-- =====  
-- variable : _Stack  
-- =====  
  
.bss  
.section ".stack",bss  
.globl _Stack  
.lcomm _Stack,_StackSize,4
```

## A.6 ユーザ・OWN・コーディング・ファイル (vrg\_own.s)

```

--
-- Sample for Ultalker-C
--
-- Copyright (C) NEC Corporation 2001
-- All rights reserved by NEC Corporation.
--
-- This is a sample program.
-- NEC assumes no responsibility for any losses incurred by customers or
-- third parties arising from the use of this program.
--
-- vrg_own.s
-- (handler initialize & handler body for V850E/MA1)
--

.set _P00IC0, 0xffffffff118
.set _ADM0, 0xffffffff200
.set _ADM1, 0xffffffff201
.set _ADM2, 0xffffffff202
.set _ADCRO, 0xffffffff210
.set _CCC00, 0xffffffff602
.set _TMCC00, 0xffffffff606
.set _TMCC01, 0xffffffff608
.set _SESC1, 0xffffffff619
.set _INTM0, 0xffffffff880
.set _INTM1, 0xffffffff882
.set _INTM2, 0xffffffff884
.set _INTM3, 0xffffffff886
.set _INTM4, 0xffffffff888

-- =====
-- Func name      : _InputInit
-- Description    : Input devices initialize
-- Arguments      : none
-- Ret values     : none
-- Stack         : none
-- =====

.text
.align 2
.globl _InputInit
_InputInit:

-- disable interrupt
di

-- init timer interrupt (with mask)
mov 0x41, r6

```

```

st.b r6,      _P00IC0[r0]

mov  0x01,    r6
st.b r6,      _INTM0[r0] -- NMI up

mov  0x00,    r6
st.b r6,      _INTM1[r0] -- non use

mov  0x00,    r6
st.b r6,      _INTM2[r0] -- non use

mov  0x00,    r6
st.b r6,      _INTM3[r0] -- non use

mov  0x00,    r6
st.b r6,      _INTM4[r0] -- non use

-- init A/D converter
-- reset A/D --
st.b r0,      _ADM2[r0]

-- set ADM2 --
mov  1,       r6
st.b r6,      _ADM2[r0]

-- set ADM0 --
mov  0x10,    r6    -- BS = 0: 1Buffer mode
                    -- MS = 1: Select mode
                    -- Use ANI0
st.b r6,      _ADM0[r0]

-- set ADM1 --
mov  0x06,    r6    -- A/D Trigger Mode, Frequency:9.60us
st.b r6,      _ADM1[r0]

-- init timer
-- set TMCC00 --
mov  0x01,    r6    -- CS10-CS12 = 0
                    -- CE1 = 0 (Timer Stop)
                    -- CAE = 1
st.b r6,      _TMCC00[r0]

-- set TMCC01 --
mov  0x0b,    r6    -- OST  = 0
                    -- ENT11 = 0
                    -- ALV1 = 0
                    -- ETI1 = 0
                    -- CCLR = 1
                    -- (=0)
                    -- CMS11 = 1
                    -- CMS10 = 1

```

```

st.b r6,    _TMCC01[r0]

-- set CCC00 --
mov 1562,   r6    -- Sampling:8kHz,CPU:50MHz : 50*10^6/4/8000
st.h r6,    _CCC00[r0]

-- start A/D convert
ld.b _ADM0[r0], r6
ori 0x80,r6,r6
st.b r6,_ADM0[r0]  -- ADCE = 1:

-- start timer
ld.b _TMCC00[r0], r6
ori 0x02,r6,r6
st.b r6,_TMCC00[r0] -- timer permit

jmp [lp]

-- =====
-- Func name      : _vrg_InputWave
-- Description    : Handler (Inport voice data)
-- Arguments      : none
-- Ret values     : none
-- Stack          : 8+20 bytes
-- =====
        .text
        .align 2
        .globl _vrg_InputWave
_vrg_InputWave:

add -8,sp
st.w lp,0[sp]
st.w r6,4[sp]

-- VoiceGet:
ld.h _ADCR0[r0], r6

-- ADCStart:
setl 7, _ADM0[r0]  -- ADCE = 1:

jarl _vrg_InputWaveData,lp

ld.w 0[sp],lp
ld.w 4[sp],r6
add 8,sp

reti

-- =====
-- Func name      : _vrg_IntOn

```

```

-- Description      : Handler enable
-- Arguments        : none
-- Ret values       : none
-- Stack            : 4 bytes
-- =====
        .text
        .align      2
        .globl      _vrg_IntOn
_vrg_IntOn:

        add         -4, sp
        st.w        r6, 0[sp]

-- enable timer interrupt
        ld.b        _P00IC0[r0], r6
        andi        0xbf, r6, r6
        st.b        r6, _P00IC0[r0]

        ld.w        0[sp], r6
        add         4, sp

        jmp         [lp]

-- =====
-- Func name        : _vrg_IntOff
-- Description      : Handler disable
-- Arguments        : none
-- Ret values       : none
-- Stack            : 4 bytes
-- =====
        .text
        .align      2
        .globl      _vrg_IntOff
_vrg_IntOff:

        add         -4, sp
        st.w        r6, 0[sp]

-- disable timer interrupt
        ld.b        _P00IC0[r0], r6
        ori         0x40, r6, r6
        st.b        r6, _P00IC0[r0]

        ld.w        0[sp], r6
        add         4, sp

        jmp         [lp]

```

## A.7 ユーザ・OWN・コーディング・ファイル (vrg\_own\_dma.s)

```
--
-- Sample for Ultalker-C
--
-- Copyright (C) NEC Corporation 2001
-- All rights reserved by NEC Corporation.
--
-- This is a sample program.
-- NEC assumes no responsibility for any losses incurred by customers or
-- third parties arising from the use of this program.
--
-- cpu_init_dma.s for V850E/MA1
--

-- interrupt vectors

.section ".INTRPT",text
.align 4
.globl __reset
__reset:
jr    __start

.org 0x10
.globl __nmi
__nmi:
reti

.org 0x40
.globl __trap00
__trap00:
reti

.org 0x50
.globl __trap10
__trap10:
reti

.org 0x60
.globl __ilgop
__ilgop:
reti

.org 0x80
.globl __intov00
__intov00:
reti

.org 0x90
.globl __intov01
```

```
__intov01:
    reti

    .org 0xa0
    .globl __intov02
__intov02:
    reti

    .org 0xb0
    .globl __intov03
__intov03:
    reti

    .org 0xc0
    .globl __intp000
__intp000:
    reti

    .org 0xd0
    .globl __intp001
__intp001:
    reti

    .org 0xe0
    .globl __intp010
__intp010:
    reti

    .org 0xf0
    .globl __intp011
__intp011:
    reti

    .org 0x100
    .globl __intp020
__intp020:
    reti

    .org 0x110
    .globl __intp021
__intp021:
    reti

    .org 0x120
    .globl __intp030
__intp030:
    reti

    .org 0x130
    .globl __intp031
__intp031:
    reti
```



```
.org 0x140
.globl __intp100
__intp100:
    reti

.org 0x150
.globl __intp101
__intp101:
    reti

.org 0x160
.globl __intp102
__intp102:
    reti

.org 0x170
.globl __intp103
__intp103:
    reti

.org 0x180
.globl __intp110
__intp110:
    reti

.org 0x190
.globl __intp111
__intp111:
    reti

.org 0x1a0
.globl __intp112
__intp112:
    reti

.org 0x1b0
.globl __intp113
__intp113:
    reti

.org 0x1c0
.globl __intp120
__intp120:
    reti

.org 0x1d0
.globl __intp121
__intp121:
    reti

.org 0x1e0
```

```
.globl __intp122
__intp122:
    reti

.org 0x1f0
.globl __intp123
__intp123:
    reti

.org 0x200
.globl __intp130
__intp130:
    reti

.org 0x210
.globl __intp131
__intp131:
    reti

.org 0x220
.globl __intp132
__intp132:
    reti

.org 0x230
.globl __intp133
__intp133:
    reti

.org 0x240
.globl __intcmd0
__intcmd0:
    reti

.org 0x250
.globl __intcmd1
__intcmd1:
    reti

.org 0x260
.globl __intcmd2
__intcmd2:
    reti

.org 0x270
.globl __intcmd3
__intcmd3:
    reti

.org 0x280
.globl __intdma0
__intdma0:
```

```
jr _vrg_InputWave

.org 0x290
.globl __intdma1
__intdma1:
    reti

.org 0x2a0
.globl __intdma2
__intdma2:
    reti

.org 0x2b0
.globl __intdma3
__intdma3:
    reti

.org 0x2c0
.globl __intcsi0
__intcsi0:
    reti

.org 0x2d0
.globl __intser0
__intser0:
    reti

.org 0x2e0
.globl __intsr0
__intsr0:
    reti

.org 0x2f0
.globl __intst0
__intst0:
    reti

.org 0x300
.globl __intcsi1
__intcsi1:
    reti

.org 0x310
.globl __intser1
__intser1:
    reti

.org 0x320
.globl __intsr1
__intsr1:
    reti
```

```

.org 0x330
.globl __intst1
__intst1:
    reti

.org 0x340
.globl __intcsi2
__intcsi2:
    reti

.org 0x350
.globl __intser2
__intser2:
    reti

.org 0x360
.globl __intsr2
__intsr2:
    reti

.org 0x370
.globl __intst2
__intst2:
    reti

.org 0x380
.globl __intad
__intad:
    reti

-- =====
-- function : __start
-- =====

.section ".START",text
.align 2

-- stack memory size
.set    _StackSize, 0x200

.globl  __start
.globl  __startend
.extern _main
__start:

.set    _PHCMD, 0xfffff800
.set    _CKC, 0xfffff822
.set    _VSWC, 0xfffff06e

mov     0x13, r6
st.b   r6, _VSWC[r0]  -- set VSWC (50MHz)

```

```

mov     0x000000a0,r6
ldsr   r6, 5          -- set PSW NP 1,ID 1
mov     0x07, r6      -- CKC value 0x07
st.b   r6, _PHCMD[r0] -- CKC protection OFF
st.b   r6, _CKC[r0]  -- set CKC
nop
nop
nop
nop
nop
mov     0x00000020,r6
ldsr   r6, 5          -- reset PSW ID 1

----- sp set -----
mov     (#_Stack+_StackSize), sp

----- gp set -----
mov     #__gp_DATA, gp

----- tp set -----
mov     #__tp_TEXT, tp

jarl   _main, lp

-- =====
-- function : __startend
-- =====
__startend:
    halt
    jr   __startend

-- =====
-- variable : _Stack
-- =====

.bss
.section ".stack",bss
.globl  _Stack
.lcomm  _Stack,_StackSize,4

```

## 付録 B 総合索引

### B.1 50音で始まる語句の索引

#### 【あ】

アプリケーション・フロー ... 27

#### 【い】

インストレーション ... 65

#### 【え】

エラー・コード ... 72

#### 【お】

音声信号入力 ... 16

音声認識ミドルウェア ... 15

音声認識ライブラリ・ヘッダ・ファイル ... 26

音声分析 ... 16

#### 【か】

開発環境へのファイル展開 ... 65

#### 【さ】

サポート・ツール ... 18

サンプル・ソース ... 76

サンプル・プログラム作成手順 ... 66

#### 【し】

辞書作成 ... 17

辞書作成サンプル・ソース ... 80

辞書作成方法 ... 70

システム構築方法 ... 68

出力結果 ... 72

処理概要 ... 16

シンボル名規約 ... 75

#### 【す】

スタートアップ・ルーチン ... 89, 96

#### 【せ】

製品概要 ... 17

性能 ... 18

#### 【ち】

チェック・サム概要 ... 73

チェック・サム機能 ... 73

チェック・サム使用方法 ... 73

#### 【と】

動作環境 ... 70

動作対象 CPU ... 17

#### 【に】

入力フォーマット ... 70

認識結果出力 ... 16

認識処理サンプル・ソース ... 76, 85

認識単語辞書を PC 版ユーザ辞書作成ツールを使用して作成する場合 ... 68

認識単語辞書を vrg\_MakeDic 関数を使用して作成する場合 ... 69

#### 【の】

ノイズ相対レベル ... 39

#### 【は】

パターン・マッチング ... 16

パッケージ内容 ... 19

#### 【ひ】

必要メモリ・サイズ ... 18

#### 【へ】

変数仕様 ... 30

**【み】**

ミドルウェア ... 15  
ミドルウェア・ライブラリ共通ヘッダ・ファイル  
... 26

**【ゆ】**

ユーザ・OWN・コーディング関数仕様 ... 54  
ユーザ・OWN・コーディング・ファイル  
... 105, 109

**【ら】**

ライブラリ関数仕様 ... 37  
ライブラリ仕様 ... 24  
ライブラリ・ヘッダ・ファイル ... 26

**【ろ】**

ロケーションおよびシンボル名規約 ... 75  
ロケーションの変更 ... 75

## B.2 アルファベットで始まる語句の索引

## 【B】

bin ... 20  
bin2elf.exe ... 20

## 【C】

checksum.c ... 22, 23  
cpu\_init.s ... 22, 23, 89  
cpu\_init\_dma.s ... 22, 23, 96

## 【D】

dat850 ... 20  
doc ... 21

## 【G】

GHS 社製ツールのライブラリと MULTI のビルダ機能を使用する場合 ... 67

## 【I】

inc850 ... 21

## 【L】

lib850 ... 21  
lib850\_ghs ... 21  
libvrg.a ... 21

## 【M】

makefile ... 22, 23  
mwclib.h ... 21, 26

## 【N】

NEC 製ツールのライブラリとプロジェクト・マネージャを使用する場合 ... 66

## 【P】

PC 版ユーザ辞書作成ツール ... 70  
PC 版ユーザ辞書作成ツール使用方法 ... 71

## 【R】

r32 ... 21  
resprint.c ... 22, 23  
rs.s ... 22, 23

## 【S】

sample1.bld ... 23  
sample1.c ... 22, 23, 76  
sample1.prj ... 22  
sample2.bld ... 23  
sample2.c ... 22, 23, 80  
sample2.prj ... 22  
sample3.bld ... 23  
sample3.c ... 22, 23, 85  
sample3.prj ... 22  
secfile.dir ... 22  
secfile.lnk ... 23  
smp850 ... 22  
smp850\_ghs ... 23  
smpdic.s ... 22  
smpdic.txt ... 20

## 【V】

vrg850 ... 20, 21  
vrg\_BufSize 変数 ... 34  
vrg\_GetVersion 関数 ... 53  
vrg\_InitParam 関数 ... 63  
vrg\_InputWaveDataDMA 関数 ... 44  
vrg\_InputWaveData 関数 ... 41  
vrg\_InputWave 関数 ... 54  
vrg\_IntOff 関数 ... 60  
vrg\_IntOn 関数 ... 58  
vrglib.h ... 21, 26  
vrg\_MakeDic 関数 ... 49  
vrg\_MemAreaSize 変数 ... 32  
vrg\_MemArea 変数 ... 30  
vrgmkdic.exe ... 20  
vrg\_own\_dma.s ... 22, 23, 109  
vrg\_own.s ... 22, 23, 105  
vrg\_RecogComplete 関数 ... 62  
vrg\_RecogReady 関数 ... 61  
vrg\_Recog 関数 ... 37  
vrg\_Stop 関数 ... 40  
vrg\_VoiceBuf 変数 ... 33  
vrg\_VoiceShift 変数 ... 35



(× 毛)

— 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  
(電話 : 午前 9:00 ~ 12:00 , 午後 1:00 ~ 5:00)

電話 : 044-435-9494  
FAX : 044-435-9608  
E-mail : info@lsi.nec.co.jp

**【営業関係お問い合わせ先】**

第一販売事業部

東京 (03)3798-6106, 6107,  
6108  
大阪 (06)6945-3178, 3200,  
3208, 3212  
広島 (082)242-5504  
仙台 (022)267-8740  
郡山 (024)923-5591  
千葉 (043)238-8116

第二販売事業部

東京 (03)3798-6110, 6111,  
6112  
立川 (042)526-5981, 6167  
松本 (0263)35-1662  
静岡 (054)254-4794  
金沢 (076)232-7303  
松山 (089)945-4149

第三販売事業部

東京 (03)3798-6151, 6155, 6586,  
1622, 1623, 6156  
水戸 (029)226-1702  
前橋 (027)243-6060  
鳥取 (0857)27-5313  
太田 (0276)46-4014  
名古屋 (052)222-2170, 2190  
福岡 (092)261-2806

**【資料の請求先】**

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

**【NECエレクトロニクス デバイス ホームページ】**

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.nec.co.jp/>

**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] μSAP703000-B07 ユーザーズ・マニュアル  
(U14836JJ3V0UM00 (第3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, 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。

キ  
ン  
ト  
ン