

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い申し上げます。

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情報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。お客様に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。

# SH7144Fグループ 内蔵インタフェース I<sup>2</sup>Cバスインタフェース編

アプリケーションノート

ルネサスSuperH™ RISC engineファミリ  
SH7144シリーズ

## ご注意

### 安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

### 本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジー製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジーが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジーは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジーは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジー半導体製品のご購入に当たりましては、事前にルネサス テクノロジー、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジーホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジーはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジーは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジー、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジーの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジー、ルネサス販売または特約店までご照会ください。

---

## はじめに

---

現在、周辺インタフェースは、低コスト化や使い勝手向上のニーズにより、各分野において統一化／標準化が進んでいます。本アプリケーションノートで取り扱う I<sup>2</sup>C バスインタフェースも標準インタフェース化されているものの 1 つで、家電製品の制御用 IC のインタフェース、ノート PC のバッテリーパックの制御用インタフェースや PC モニタの制御用インタフェースなどに応用されています。

I<sup>2</sup>C バスはオランダの Philips 社が開発した双方向シリアルバスシステムの規格です。この規格に基づく製品どうしであれば、2 本の配線（クロックライン、データライン）で複数の周辺 IC 間の相互データ通信ができます。

ルネサスオリジナル 32 ビットシングルチップマイクロコンピュータ SH7144F グループに内蔵している I<sup>2</sup>C バスインタフェースは Philips 社の提唱している I<sup>2</sup>C バスインタフェース方式に準拠しており、サブセット機能を備えています（ただし、使用条件により、一部 I<sup>2</sup>C バスインタフェースの仕様を満足しない場合がありますのでご注意ください）。

本アプリケーションノートでは、第 1 章と第 2 章で I<sup>2</sup>C バスの概要と弊社 I<sup>2</sup>C バスインタフェースモジュールの仕様、機能を概説します。第 2 章では SH7144F グループによる I<sup>2</sup>C バスインタフェースの応用例を紹介します。

なお、本アプリケーションノートに掲載されている各タスクのプログラムなどの動作は確認しておりますが、実際にご使用になる場合には改めて動作確認の上ご使用下さいますようお願い致します。

【注】 I<sup>2</sup>C バス : Inter IC Bus

---

# 目次

---

## 第1章 I<sup>2</sup>Cバスの概要

1.1	I <sup>2</sup> Cバスの特徴	1-1
1.1.1	I <sup>2</sup> Cバスの特徴	1-1
1.1.2	シリアルインタフェース(SCI)との相違	1-2
1.1.3	I <sup>2</sup> Cバスの接続形式	1-3
1.2	I <sup>2</sup> Cバスを使用したデータ転送方法	1-4
1.2.1	I <sup>2</sup> Cバスを使用したデータ転送の基本事項	1-4
1.2.2	データの転送手順	1-7
1.3	シングルマスタとマルチマスタの構成	1-8
1.3.1	シングルマスタ	1-8
1.3.2	マルチマスタ	1-8
1.4	通信調整手順	1-9

## 第2章 SH7144F グループ応用例

2.1	SH7144F グループ応用例手引き	2-1
2.1.1	SH7144F グループ応用例の構成	2-1
2.1.2	ベクタテーブル定義ファイル説明	2-3
2.1.3	レジスタ定義ファイル説明	2-7
2.2	シングルマスタ送信	2-7
2.2.1	仕様	2-7
2.2.2	動作説明	2-9
2.2.3	ソフトウェア説明	2-10
2.2.4	フローチャート	2-14
2.2.5	プログラムリスト	2-17
2.3	シングルマスタ受信	2-20
2.3.1	仕様	2-20
2.3.2	動作説明	2-21
2.3.3	ソフトウェア説明	2-22
2.3.4	フローチャート	2-26
2.3.5	プログラムリスト	2-30

## 第3章 付録

3.1	SH7145F レジスタ定義ファイル	3-1
-----	--------------------	-----

---

# 1. I<sup>2</sup>C バスの概要

---

## 1.1 I<sup>2</sup>C バスの特徴

### 1.1.1 I<sup>2</sup>C バスの特徴

I<sup>2</sup>C バスの特徴を以下に示します。

- バスは、シリアルデータライン (SDA)、シリアルクロックライン (SCL) の 2 本のバスラインで構成されます。I<sup>2</sup>C バス装置の拡張が容易です。
- 装置間にはマスタとスレーブという関係が常に成り立ち、各装置は固有のアドレスを持っています。マスタとなる装置が、最初に通信相手の有する固有アドレスを指定することにより通信のバスが形成され、データ通信が可能となります。
- 任意の装置がマスタになることができます (マルチマスタシステムを構築可能)。そのため、I<sup>2</sup>C バスインタフェースは、データ破壊を防ぐためのバス権競合回避のシステムが定義されています。
- データ転送速度は、標準モードで最高 100kbps、高速モードで最高 400kbps です (I<sup>2</sup>C バス仕様書 Ver2.0 では、3.4Mbps まで定義されています)。
- I<sup>2</sup>C バスシステムにおける装置の総数は、システムのバス負荷容量の上限値 400pF で決定されます。
- 応用例として SMBus\*<sup>1</sup>ACCESS.bus\*<sup>2</sup>があります。

【注】 \*1 SMBus (System Management Bus) は、Duracell 社と intel 社が策定したシリアルバスです。

\*2 ACCESS.bus は、Digital Equipment 社が策定したシリアルバスです。

## 1. I<sup>2</sup>C バスの概要

### 1.1.2 シリアルインタフェース(SCI)との相違

ルネサスシリアルインタフェースであるシリアルコミュニケーションインタフェース (SCI) との相違点をまとめます。

表 1.1 に示すように、SCI ではデータラインとして、送信用データラインと受信データラインの 2 本を使用します。データ通信方式は、一般的に 1 対 1 で行われます。一方、I<sup>2</sup>C バスはデータライン 1 本で双方向通信します。通信相手先は、マスタとなる装置が通信相手の固有アドレスを指定することで決定されるため、複数の任意の装置とのデータ送受信が可能です。また I<sup>2</sup>C バスは、バス権の衝突回避メカニズムが定義されているため、任意の装置がマスタとして動作できるマルチマスタシステムに対応します。転送レートは標準モードでは最大 100kbps、高速モードで最大 400kbps となります。

表 1.1 SCI との相違点

	SCI		I <sup>2</sup> C バス
	クロック同期式	調歩同期式	
使用端子	3 線式	2 線式	2 線式
	送信データ出力	送信データ出力	送受信データ入出力
	受信データ入力	受信データ入力	
	シリアルクロック	シリアルクロック (外部クロック使用時)	シリアルクロック
転送レート	100bps~4Mbps	100bps~4Mbps	100kbps(標準モード) 400kbps(高速モード)
複数の IC との送受信	不可	不可	可能、スレーブを アドレスで制御

【注】 I<sup>2</sup>C バス仕様書 Ver.2.0 で定義されている Hs モード (最大転送速度 : 3.4Mbps) はサポートしていません。



1.1.3 I<sup>2</sup>C バスの接続形式

図 1.1 に I<sup>2</sup>C バスインタフェースの接続形式を示します。この図のように I<sup>2</sup>C バスは、クロックライン SCL とデータライン SDA から構成され、それぞれプルアップ抵抗でバス電源 VBB に接続されます。デバイス 1 とデバイス 2 の各 SCL 端子 / SDA 端子はそれぞれ SCL ラインと SDA ラインにワイヤード AND 接続されます。

デバイス 1 が SCL ラインを "Low" にドライブしているとき、デバイス 2 は SCL ラインの状態をモニタすることにより他のデバイスがバスを使用していることを確認します。またワイヤード AND 接続により、デバイス 1 がバスを使用中で SCL ラインをドライブしているとき、デバイス 2 が SCL を "Low" にドライブし、デバイス 1 に対して通信動作を "待ち" 状態にすることができます (詳細は「第 2 章 SH7144F グループ応用例」を参照ください)。

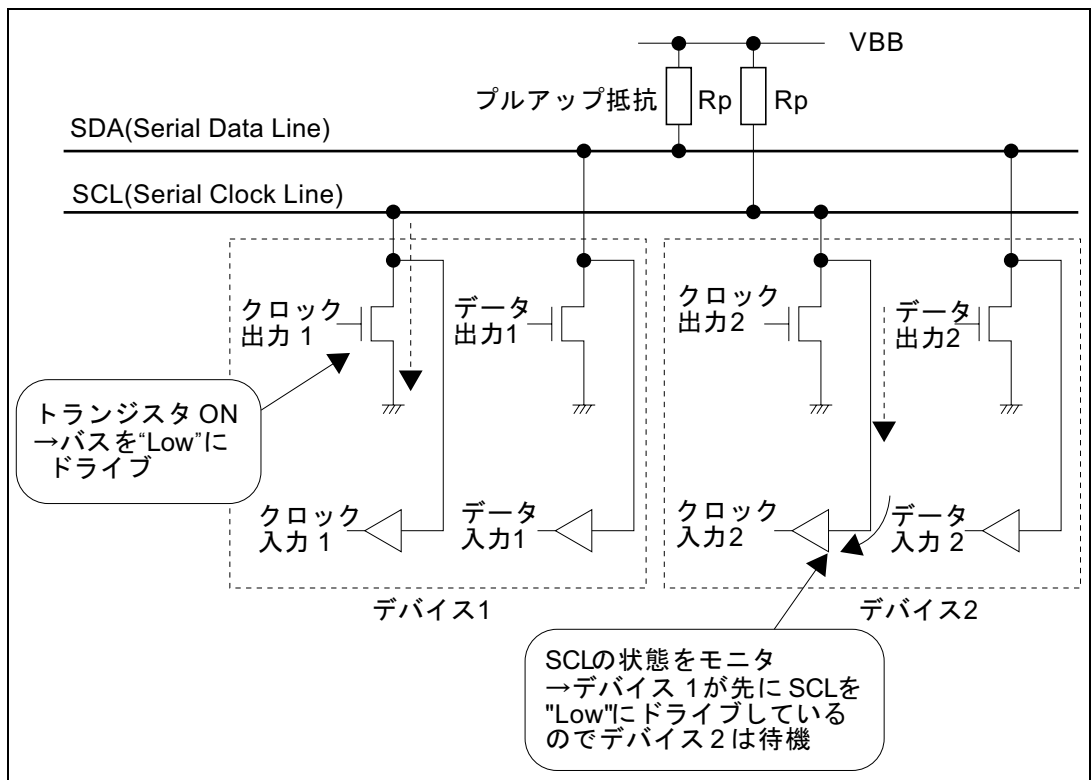


図 1.1 バスインタフェースの接続形式 (デバイス 1 が最初に SCL を "Low" にドライブした場合)

### 1.2 I<sup>2</sup>C バスを使用したデータ転送方法

#### 1.2.1 I<sup>2</sup>C バスを使用したデータ転送の基本事項

はじめに I<sup>2</sup>C バスを使用したデータ転送の基本事項を説明します。

##### (1) マスタデバイス

マスタデバイスは、データ通信を行うための同期化クロックを生成し、データ通信の開始/停止を示す開始条件/停止条件を発行します。

##### (2) スレーブデバイス

スレーブデバイスは、マスタデバイス以外の I<sup>2</sup>C バスデバイスです。マスタデバイスからアドレス指定されます。

##### (3) 送信デバイス

送信デバイスとは、データをバスに送信するデバイスです。マスタデバイスとスレーブデバイスの場合があります。

##### (4) 受信デバイス

受信デバイスとは、データをバスから受信するデバイスです。マスタデバイスとスレーブデバイスの場合があります。

##### (5) 開始条件と停止条件

開始条件とは、図 1.2 のように SCL ラインが"High"のときに、SDA ラインが"High"から"Low"に変化する動作です。これによりデータ通信動作が開始されます。

停止条件とは、図のように SCL ラインが"High"のときに、SDA ラインが"Low"から"High"に変化する動作です。これによりデータ通信動作が停止されます。

開始条件と停止条件は、必ずマスタによって生成されます。開始条件が発生した後は、バスがビジー状態になります。停止条件が生成されると、その後しばらく、バスは再びフリー状態になります

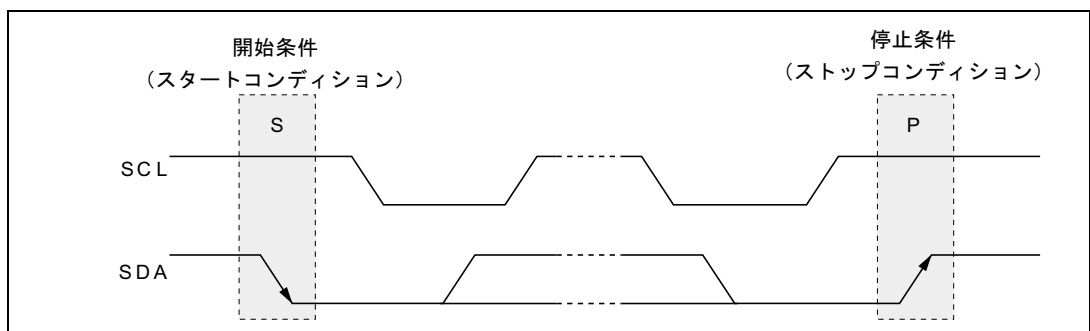


図 1.2 開始条件と停止条件

### (6) データの出力タイミング

図 1.3 のようにデータ出力タイミングは、SCL ラインが"Low"のとき、SDA ライン上のデータが更新され、SCL ラインが"High"のとき SDA ライン上のデータが確定します。SCL ラインが"High"のとき SDA ラインが変化するのは前記「開始条件」および「停止条件」のときのみです。

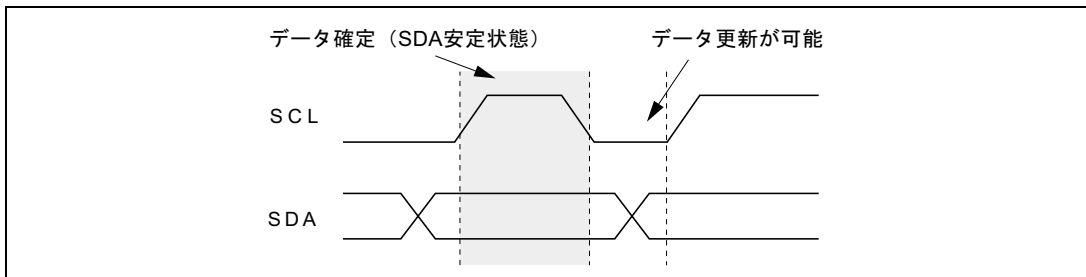


図 1.3 データ出力タイミング

### (7) マスタ送信動作

マスタ送信動作とは、マスタデバイスが送信デバイスの場合の動作です。開始条件発行後のスレーブアドレスの送信やスレーブデバイスへのコマンドなどの送信をする場合があります。

### (8) マスタ受信動作

マスタ受信動作とは、マスタデバイスが受信デバイスの場合の動作です。

### (9) スレーブ送信動作

スレーブ送信動作とは、スレーブデバイスが送信デバイスの場合の動作です。

### (10) スレーブ受信動作

スレーブ受信動作とは、スレーブデバイスが受信デバイスの場合の動作です。開始条件後のマスタデバイスによるスレーブアドレス送信フレームでは、スレーブデバイスは受信動作となります。

### (11) バス解放状態

すべての I<sup>2</sup>C バスデバイスが通信していない状態です。SCL、SDA ラインとも定常的に"High"状態です。

### (12) バス占有状態

バス占有状態とは、I<sup>2</sup>C バスデバイスがデータ通信を行っている状態です。マスタデバイスが停止条件を発行した時点でバス開放状態に戻ります。

## 1. I<sup>2</sup>C バスの概要

### (13) データ転送フォーマット

図 1.4 に I<sup>2</sup>C バスのデータ転送フォーマットを示します。「開始条件」、「停止条件」、SCL クロックはマスタデバイスが生成します。開始条件後の第 1 データはスレーブアドレスとなり、8 ビット目にデータの通信方向を示すビットが付加されます。本ビットは、「0」のとき、第 2 バイト以降のデータ通信がマスタ送信動作であることを示し、「1」のときは、第 2 バイト以降のデータ通信はマスタ受信動作をすることを意味します。スレーブアドレスは 7 ビットで定義され\*1、ユーザが B'0000000 ~ H'1111111 の間で設定しますが、アドレス B'0000000 (ゼネラルコールアドレスと呼ぶ\*2) と一部のアドレスは予約されています。

転送データは 1 バイト (8 ビット) 単位になり、9 ビット目に受信デバイスからの確認応答ビット (アクノリッジビット) が付加されます。例えばマスタがスレーブアドレスを送信した場合には、該当するスレーブは 9 クロック目に SDA を "Low" にドライブし、マスタにアクノリッジを返します。1 回の開始条件-停止条件間に転送できるバイトデータ数には制限はありません。データ通信は、「停止条件」をもって終了します。

【注】 \*1 I<sup>2</sup>C バス仕様書では、10 ビットアドレス指定が定義されています。ルネサスの I<sup>2</sup>C バスインタフェースモジュールでは 10 ビットアドレス指定をサポートしておりません。

\*2 ゼネラルコールアドレス B'0000000 は、バスに接続されているすべてのスレーブアドレスを指定するときに使用します。

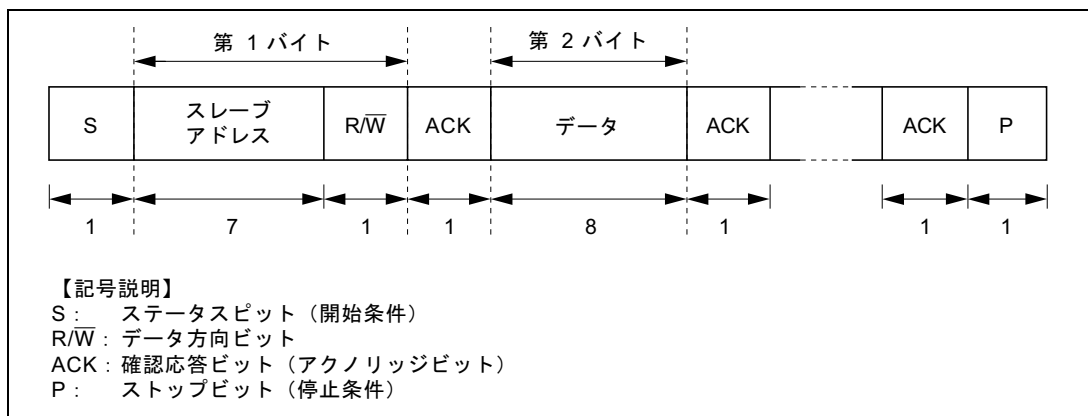


図 1.4 データ転送フォーマット

### 1.2.2 データの転送手順

(例：マスタデバイス＝送信デバイス、スレーブデバイス＝受信デバイス)

図 1.5 にマスタデバイスがスレーブデバイスに 1 バイトのデータを送信する場合の例を示します。まず、マスタデバイスは開始条件を発行し、SCL ラインが"High"のときに、SDA ラインを"High"から"Low"に変化させます。次にマスタは、SCL ライン上にクロックを出力するとともに、SDA ライン上に通信対象となるスレーブのアドレスを出力します。スレーブのアドレスは 7 ビットで定義され、8 ビット目に通信方向を表すビットを付加されます。

マスタデバイスは 9 クロック目に SDA ラインを開放し、スレーブデバイスからのアクノリッジに備えます。スレーブデバイスは、9 クロック目に SDA ラインを"Low"にドライブしアクノリッジを返します。マスタデバイスはスレーブアドレスからのアクノリッジを受信し、次の送信データが準備できるまで、SCL ラインを"Low"に保持します。送信データの準備ができたところでマスタデバイスは、SCL ラインにクロックを出力しながら、データを SDA ラインに出力します。前回と同様に 9 クロック目にスレーブデバイスはマスタデバイスにアクノリッジを返し、正常にデータが受信できたことを通知します。マスタデバイスは、スレーブデバイスからのアクノリッジを受け取ると、SCL ラインを"Low"に保持します。そして停止条件を発行し、SCL ラインが"High"のとき、SDA ラインが"Low"から"High"に変化させます。

データ通信中、もし、スレーブデバイスが他の処理を行っているため、すぐにデータを受信できない場合は、スレーブデバイス側で SCL ラインを"Low"に保持し、マスタデバイスを待ち状態にすることができます。スレーブデバイスが SCL を"Low"にドライブできるタイミングは、マスタデバイスが SCL を"Low"にドライブしているときです。

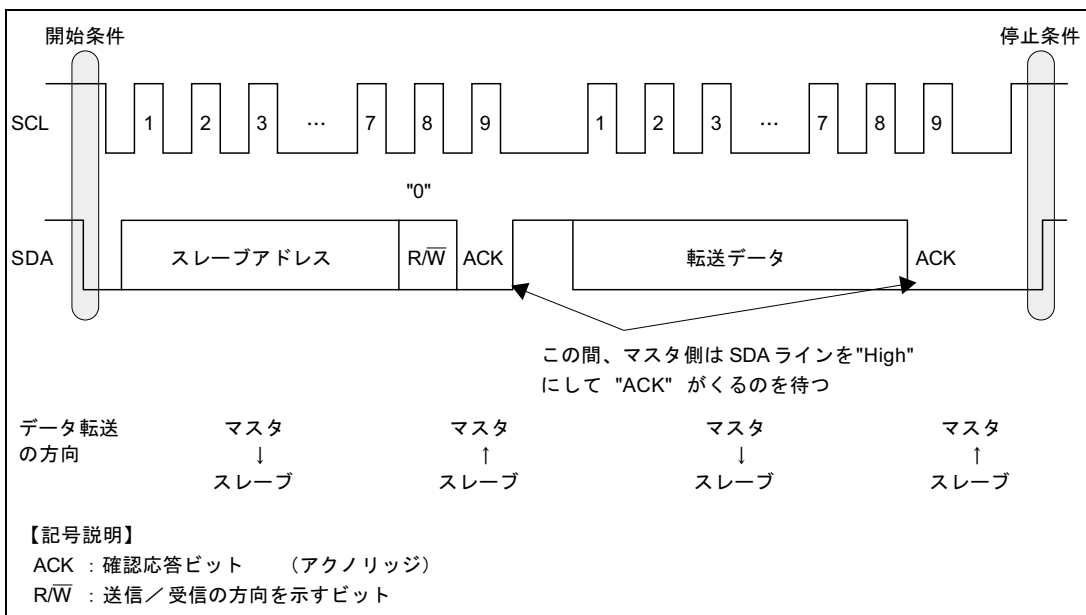


図 1.5 データ転送フォーマット (マスタ＝送信デバイス、スレーブ＝受信デバイスの場合)

## 1.3 シングルマスタとマルチマスタの構成

### 1.3.1 シングルマスタ

マスタデバイスは「開始条件」および「停止条件」を発行し、データ通信を管理します。また SCL ライン上にデータを送受信するための同期化クロックやスレーブアドレスを出力します。マスタデバイスが常に固定されている図 1.6 のようなシステム構成をシングルマスタ構成といいます。

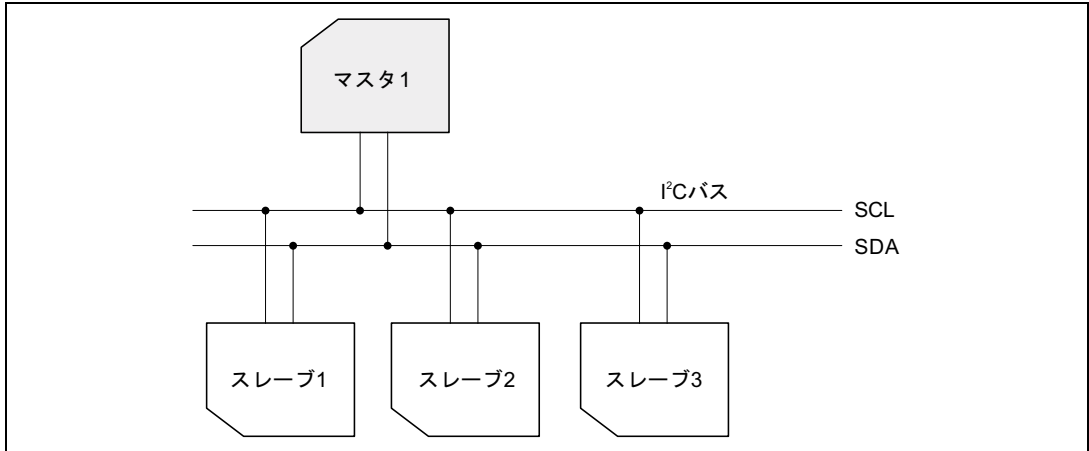


図 1.6 シングルマスタ構成

### 1.3.2 マルチマスタ

図 1.7 のように、1つのシステム内にマスタと成り得るデバイスが 2 個以上存在する構成をマルチマスタ構成といいます。

マスタデバイスはバスが解放状態のときのみデータ転送を開始することができますがマルチマスタ構成の場合、複数のマスタデバイスが同時にデータ転送を開始しようとする可能性があります。つまり、バス権の衝突が生じます。このため、I<sup>2</sup>Cバスの仕様にはバス権の衝突が生じた場合の通信調整手順が規定されています。詳細は「1.4 通信調整手順」を参照してください。

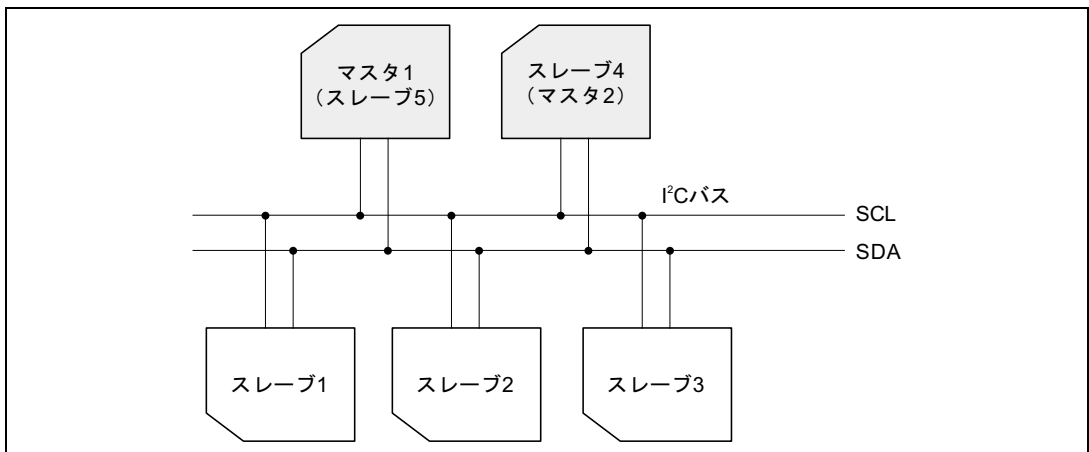


図 1.7 マルチマスタ構成

## 1.4 通信調整手順

I<sup>2</sup>C バスインタフェースでは、バス権の衝突を回避するための通信調整手順が定義されており、マルチマスタ構成のシステムに対応できます。

マスタデバイスは、バスラインをモニタし、バスが開放されていることを確認し、開始条件を発行します。このとき複数のマスタデバイスによる開始条件発行処理が発生する可能性があります。そこで図 1.8 に示すような通信調整手順により、唯一のマスタデバイスが決定されます。

I<sup>2</sup>C バスでは、SCL ラインが"High"期間中に SDA ライン上のデータが確定している必要があります。そこで各デバイスは、開始条件後の SCL ラインの立ち上がりをモニタし、SDA ラインの状態と各デバイス内部のデータ（スレーブアドレス）を比較します。もし、デバイス 1 が SDA を"High"に、デバイス 2 が SDA を"Low"にしていれば、ワイヤード AND 接続により実際の SDA ラインは"Low"になりますので、デバイス 1 は、自分の出そうとしているデータと異なることを確認し、データ出力段をオフにします。本例ではデバイス 2 がマスタデバイスとして動作を継続します。すべてのマスタが同一のスレーブ装置をアドレス指定しようとしている場合にはさらに次の段階に進み、データの比較が行われます。

例えば、図 1.9 のように転送データが H'01、H'02 の場合、データ H'01 の方が"Low"期間が長いのでデータ H'01 が有効になります。したがって、ゼネラルコールアドレス（H'00）が最優先されます。

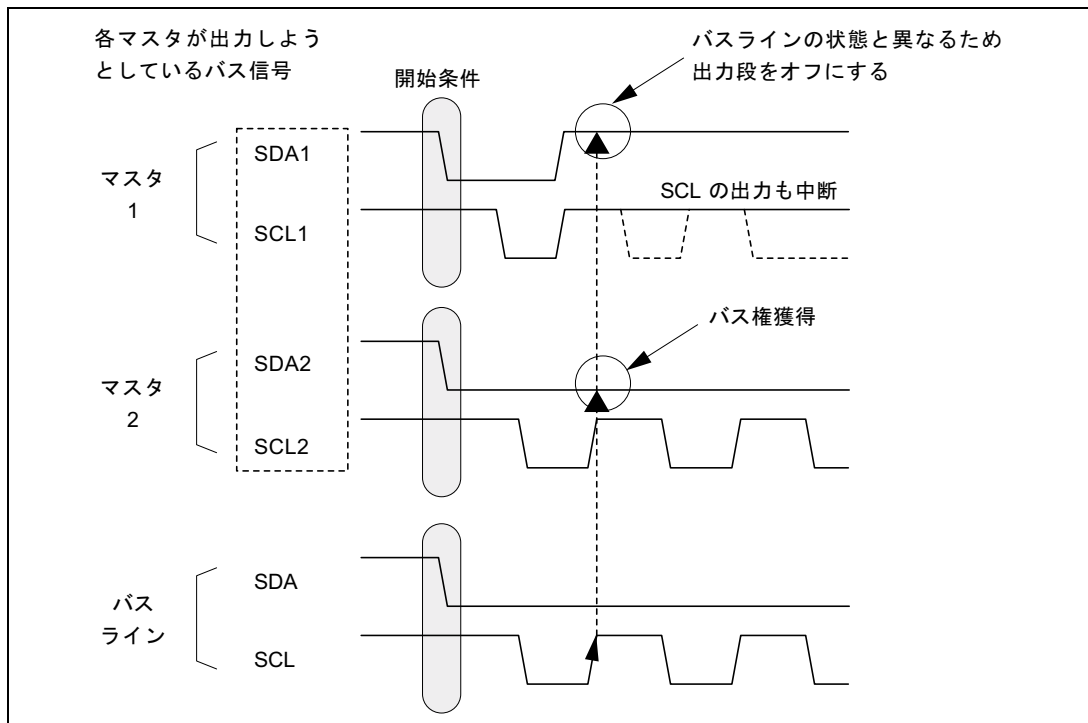


図 1.8 通信調整手順（バスアービトレーションロストの検出）

## 1. I<sup>2</sup>C バスの概要

---

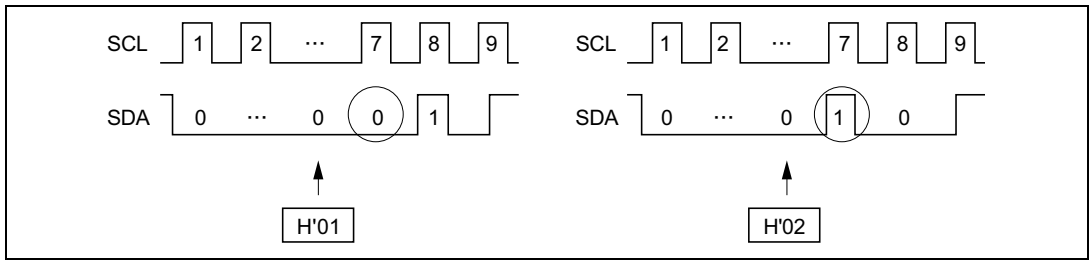


図 1.9 通信調整の具体例



---

## 2. SH7144F グループ応用例

---

### 2.1 SH7144F グループ応用例手引き

#### 2.1.1 SH7144F グループ応用例の構成

SH7144F グループ応用例は図 2.1 に示す構成で SH7144F グループの I<sup>2</sup>C バスインタフェースの使用例について説明しています。デバイスは、SH7145F を使用するものとします。

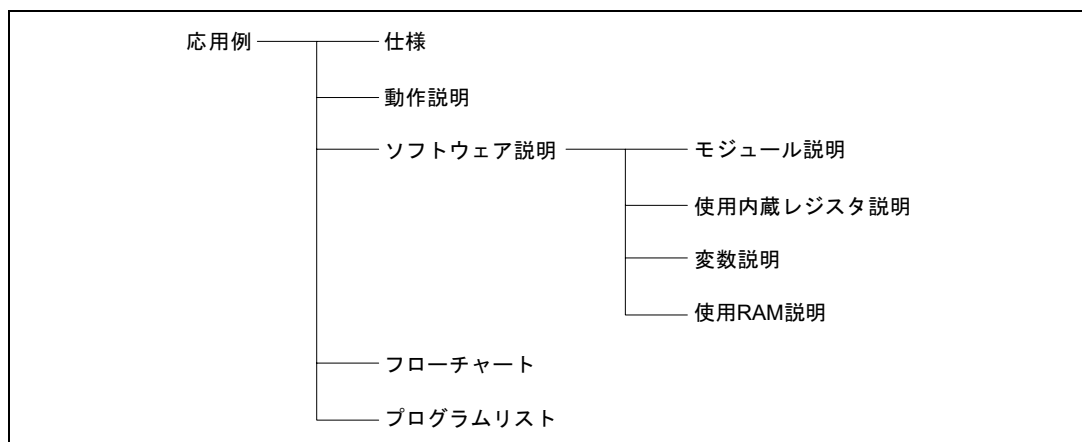


図 2.1 SH7144F グループ応用例の構成

##### (1) 仕様

タスク例のシステム仕様について説明しています。

##### (2) 動作説明

タスク例の動作をタイミングチャートを使用して説明しています。

##### (3) ソフトウェア説明

###### (a) モジュール説明

- タスク例を動作させるソフトウェアのモジュールについて説明しています。

###### (b) 使用内蔵レジスタ説明

- モジュールで設定する I<sup>2</sup>C バスインタフェース、およびその他の内蔵レジスタについて説明しています。

###### (c) 変数説明

- タスク例を動作させるソフトウェアの変数について説明しています。

## 2. SH7144F グループ応用例

---

### (d) 使用 RAM 説明

モジュールで使用する RAM のラベル名および機能について説明しています。

### (4) フローチャート

タスク例を実行するソフトウェアについてフローチャートを使用して説明しています。

### (5) プログラムリスト

タスク例を実行するソフトウェアのプログラムリストを示しています。

## 2.1.2 ベクタテーブル定義ファイル説明

以下に C 言語を使用したベクタテーブルの定義ファイルを示します。図 2.2 に示すように割り込み処理ルーチンの先頭アドレスを確保したファイルを作成します。割り込み処理を使用する場合には、割り込み処理ルーチンの先頭ラベルを割り込みに対応したベクタ位置に記述します。

図 2.2 に main()関数を使用する例を示します。mian()関数、dummy()関数を外部参照します。

パワーオンリセット、マニュアルリセットのベクター番地に関数 mian()を配置します。そのときのスタック領域は、内蔵 RAM の H'FFFFFFFc に割り当ててあります。

その他の割り込みについては、ダミー関数 dummy()を配置してあります。

```

/*****
/* Filename      :   vector.c                               */
/* Written       :   2003/2/1 REV.2.1                       */
/* Purpose       :   SH7145F vector table                   */
*****/

/*----- External Function Definition -----*/
extern void main(void); /* main function */
extern void dummy(void); /* dummy function */

/*-----*/
/*          vector table                               */
/*-----*/
#pragma section VECT
const void (*const vect_tbl[]) (void) =
{
    /* NO.  Offset      Exception Sources */
    main, /* (000) H'00000000 Power-on reset PC */
    (void (*)(void))0xFFFFFfc, /* (001) H'00000004 Power-on Reset SP */
    main, /* (002) H'00000008 Manual reset PC */
    (void (*)(void))0xFFFFFfc, /* (003) H'0000000C Manual reset SP */
    dummy, /* (004) H'00000010 General illegal instruction */
    dummy, /* (005) H'00000014 (Reserved for system use) */
    dummy, /* (006) H'00000018 illegal slot instruction */
    dummy, /* (007) H'0000001C (Reserved for system use) */
    dummy, /* (008) H'00000020 (Reserved for system use) */
    dummy, /* (009) H'00000024 CPU address error */
    dummy, /* (010) H'00000028 DMA address error */
    dummy, /* (011) H'0000002C NMI */
    dummy, /* (012) H'00000030 UBC (User break) */
    dummy, /* (013) H'00000034 (Reserved for system use) */
    dummy, /* (014) H'00000038 H-UDI */
    dummy, /* (015) H'0000003C (Reserved for system use) */
    dummy, /* (016) H'00000040 (Reserved for system use) */
    dummy, /* (017) H'00000044 (Reserved for system use) */
    dummy, /* (018) H'00000048 (Reserved for system use) */
    dummy, /* (019) H'0000004C (Reserved for system use) */
    dummy, /* (020) H'00000050 (Reserved for system use) */
    dummy, /* (021) H'00000054 (Reserved for system use) */
    dummy, /* (022) H'00000058 (Reserved for system use) */
    dummy, /* (023) H'0000005C (Reserved for system use) */
    dummy, /* (024) H'00000060 (Reserved for system use) */
    dummy, /* (025) H'00000064 (Reserved for system use) */
    dummy, /* (026) H'00000068 (Reserved for system use) */
    dummy, /* (027) H'0000006C (Reserved for system use) */
    dummy, /* (028) H'00000070 (Reserved for system use) */
    dummy, /* (029) H'00000074 (Reserved for system use) */
    dummy, /* (030) H'00000078 (Reserved for system use) */
    dummy, /* (031) H'0000007C (Reserved for system use) */
    dummy, /* (032) H'00000080 Trap inst (user vectors) */
    dummy, /* (033) H'00000084 Trap inst (user vectors) */
    dummy, /* (034) H'00000088 Trap inst (user vectors) */
    dummy, /* (035) H'0000008C Trap inst (user vectors) */
}

```

図 2.2 ベクタ定義ファイル

## 2. SH7144F グループ応用例

```

dummy, /* (036) H'00000090 Trap inst (user vectors) */
dummy, /* (037) H'00000094 Trap inst (user vectors) */
dummy, /* (038) H'00000098 Trap inst (user vectors) */
dummy, /* (039) H'0000009C Trap inst (user vectors) */
dummy, /* (040) H'000000A0 Trap inst (user vectors) */
dummy, /* (041) H'000000A4 Trap inst (user vectors) */
dummy, /* (042) H'000000A8 Trap inst (user vectors) */
dummy, /* (043) H'000000AC Trap inst (user vectors) */
dummy, /* (044) H'000000B0 Trap inst (user vectors) */
dummy, /* (045) H'000000B4 Trap inst (user vectors) */
dummy, /* (046) H'000000B8 Trap inst (user vectors) */
dummy, /* (047) H'000000BC Trap inst (user vectors) */
dummy, /* (048) H'000000C0 Trap inst (user vectors) */
dummy, /* (049) H'000000C4 Trap inst (user vectors) */
dummy, /* (050) H'000000C8 Trap inst (user vectors) */
dummy, /* (051) H'000000CC Trap inst (user vectors) */
dummy, /* (052) H'000000D0 Trap inst (user vectors) */
dummy, /* (053) H'000000D4 Trap inst (user vectors) */
dummy, /* (054) H'000000D8 Trap inst (user vectors) */
dummy, /* (055) H'000000DC Trap inst (user vectors) */
dummy, /* (056) H'000000E0 Trap inst (user vectors) */
dummy, /* (057) H'000000E4 Trap inst (user vectors) */
dummy, /* (058) H'000000E8 Trap inst (user vectors) */
dummy, /* (059) H'000000EC Trap inst (user vectors) */
dummy, /* (060) H'000000F0 Trap inst (user vectors) */
dummy, /* (061) H'000000F4 Trap inst (user vectors) */
dummy, /* (062) H'000000F8 Trap inst (user vectors) */
dummy, /* (063) H'000000FC Trap inst (user vectors) */
dummy, /* (064) H'00000100 IRQ0 */
dummy, /* (065) H'00000104 IRQ1 */
dummy, /* (066) H'00000108 IRQ2 */
dummy, /* (067) H'0000010C IRQ3 */
dummy, /* (068) H'00000110 IRQ4 */
dummy, /* (069) H'00000114 IRQ5 */
dummy, /* (070) H'00000118 IRQ6 */
dummy, /* (071) H'0000011C IRQ7 */
dummy, /* (072) H'00000120 DMAC/ DEI0 */
dummy, /* (073) H'00000124 */
dummy, /* (074) H'00000128 */
dummy, /* (075) H'0000012C */
dummy, /* (076) H'00000130 DMAC/ DEI1 */
dummy, /* (077) H'00000134 */
dummy, /* (078) H'00000138 */
dummy, /* (079) H'0000013C */
dummy, /* (080) H'00000140 DMAC/ DEI2 */
dummy, /* (081) H'00000144 */
dummy, /* (082) H'00000148 */
dummy, /* (083) H'0000014C */
dummy, /* (084) H'00000150 DMAC/ DEI3 */
dummy, /* (085) H'00000154 */
dummy, /* (086) H'00000158 */
dummy, /* (087) H'0000015C */
dummy, /* (088) H'00000160 MTU0/TGIA_0 */
dummy, /* (089) H'00000164 MTU0/TGIB_0 */
dummy, /* (090) H'00000168 MTU0/TGIC_0 */
dummy, /* (091) H'0000016C MTU0/TGID_0 */
dummy, /* (092) H'00000170 MTU0/TCIV_0 */
dummy, /* (093) H'00000174 */
dummy, /* (094) H'00000178 */
dummy, /* (095) H'0000017C */
dummy, /* (096) H'00000180 MTU1/TGIA_1 */
dummy, /* (097) H'00000184 MTU1/TGIB_1 */
dummy, /* (098) H'00000188 */
dummy, /* (099) H'0000018C */
dummy, /* (100) H'00000190 MTU1/TCIV_1 */
dummy, /* (101) H'00000194 MTU1/TCIU_1 */
dummy, /* (102) H'00000198 */
dummy, /* (103) H'0000019C */
dummy, /* (104) H'000001A0 MTU2/TGIA_2 */

```

図 2.2 ベクタ定義ファイル (続き)

dummy,	/* (105)	H'000001A4	MTU2/TGIB_2	*/
dummy,	/* (106)	H'000001A8	*/	*/
dummy,	/* (107)	H'000001AC	*/	*/
dummy,	/* (108)	H'000001B0	MTU2/TCIV_2	*/
dummy,	/* (109)	H'000001B4	MTU2/TCIU_2	*/
dummy,	/* (110)	H'000001B8	*/	*/
dummy,	/* (111)	H'000001BC	*/	*/
dummy,	/* (112)	H'000001C0	MTU3/TGIA_3	*/
dummy,	/* (113)	H'000001C4	MTU3/TGIB_3	*/
dummy,	/* (114)	H'000001C8	MTU3/TGIC_3	*/
dummy,	/* (115)	H'000001CC	MTU3/TGID_3	*/
dummy,	/* (116)	H'000001D0	MTU3/TCIV_3	*/
dummy,	/* (117)	H'000001D4	*/	*/
dummy,	/* (118)	H'000001D8	*/	*/
dummy,	/* (119)	H'000001DC	*/	*/
dummy,	/* (120)	H'000001E0	MTU4/TGIA_4	*/
dummy,	/* (121)	H'000001E4	MTU4/TGIB_4	*/
dummy,	/* (122)	H'000001E8	MTU4/TGIC_4	*/
dummy,	/* (123)	H'000001EC	MTU4/TGID_4	*/
dummy,	/* (124)	H'000001F0	MTU4/TCIV_4	*/
dummy,	/* (125)	H'000001F4	*/	*/
dummy,	/* (126)	H'000001F8	*/	*/
dummy,	/* (127)	H'000001FC	*/	*/
dummy,	/* (128)	H'00000200	SCIO/ERI	*/
dummy,	/* (129)	H'00000204	SCIO/RXI	*/
dummy,	/* (130)	H'00000208	SCIO/TXI	*/
dummy,	/* (131)	H'0000020C	SCIO/TEI	*/
dummy,	/* (132)	H'00000210	SCI1/ERI	*/
dummy,	/* (133)	H'00000214	SCI1/RXI	*/
dummy,	/* (134)	H'00000218	SCI1/TXI	*/
dummy,	/* (135)	H'0000021C	SCI1/TEI	*/
dummy,	/* (136)	H'00000220	A/D ADI0	*/
dummy,	/* (137)	H'00000224	A/D ADI1	*/
dummy,	/* (138)	H'00000228	*/	*/
dummy,	/* (139)	H'0000022C	*/	*/
dummy,	/* (140)	H'00000230	DTC/SWDTEND	*/
dummy,	/* (141)	H'00000234	*/	*/
dummy,	/* (142)	H'00000238	*/	*/
dummy,	/* (143)	H'0000023C	*/	*/
dummy,	/* (144)	H'00000240	CMT/CMT0	*/
dummy,	/* (145)	H'00000244	*/	*/
dummy,	/* (146)	H'00000248	*/	*/
dummy,	/* (147)	H'0000024C	*/	*/
dummy,	/* (148)	H'00000250	CMT/CMT1	*/
dummy,	/* (149)	H'00000254	*/	*/
dummy,	/* (150)	H'00000258	*/	*/
dummy,	/* (151)	H'0000025C	*/	*/
dummy,	/* (152)	H'00000260	WDT/ITI	*/
dummy,	/* (153)	H'00000264	(Reserved for system use)	*/
dummy,	/* (154)	H'00000268	*/	*/
dummy,	/* (155)	H'0000026C	*/	*/
dummy,	/* (156)	H'00000270	I/O(MTU)/MTUOEI	*/
dummy,	/* (157)	H'00000274	*/	*/
dummy,	/* (158)	H'00000278	*/	*/
dummy,	/* (159)	H'0000027C	*/	*/
dummy,	/* (160)	H'00000280	*/	*/
dummy,	/* (161)	H'00000284	*/	*/
dummy,	/* (162)	H'00000288	*/	*/
dummy,	/* (163)	H'0000028C	*/	*/
dummy,	/* (164)	H'00000290	(Reserved for system use)	*/
dummy,	/* (165)	H'00000294	(Reserved for system use)	*/
dummy,	/* (166)	H'00000298	(Reserved for system use)	*/
dummy,	/* (167)	H'0000029C	(Reserved for system use)	*/
dummy,	/* (168)	H'000002A0	SCI2/ERI	*/
dummy,	/* (169)	H'000002A4	SCI2/RXI	*/
dummy,	/* (170)	H'000002A8	SCI2/TXI	*/
dummy,	/* (170)	H'000002AC	SCI2/TEI	*/
dummy,	/* (170)	H'000002B0	SCI3/ERI	*/
dummy,	/* (170)	H'000002B4	SCI3/RXI	*/

図 2.2 ベクタ定義ファイル (続き)

## 2. SH7144F グループ応用例

dummy,	/* (170)	H'000002B8	SCI3/TXI	*/
dummy,	/* (170)	H'000002BC	SCI3/TEI	*/
dummy,	/* (170)	H'000002C0	(Reserved for system use)	*/
dummy,	/* (170)	H'000002C4	(Reserved for system use)	*/
dummy,	/* (170)	H'000002C8	(Reserved for system use)	*/
dummy,	/* (170)	H'000002CC	(Reserved for system use)	*/
dummy,	/* (180)	H'000002D0	(Reserved for system use)	*/
dummy,	/* (180)	H'000002D4	(Reserved for system use)	*/
dummy,	/* (180)	H'000002D8	(Reserved for system use)	*/
dummy,	/* (180)	H'000002DC	(Reserved for system use)	*/
dummy,	/* (180)	H'000002E0	(Reserved for system use)	*/
dummy,	/* (180)	H'000002E4	(Reserved for system use)	*/
dummy,	/* (180)	H'000002E8	(Reserved for system use)	*/
dummy,	/* (180)	H'000002EC	(Reserved for system use)	*/
dummy,	/* (180)	H'000002F0	(Reserved for system use)	*/
dummy,	/* (180)	H'000002F4	(Reserved for system use)	*/
dummy,	/* (190)	H'000002F8	(Reserved for system use)	*/
dummy,	/* (191)	H'000002FC	(Reserved for system use)	*/
dummy,	/* (192)	H'00000300	IIC/ICI	*/
dummy,	/* (193)	H'00000304	(Reserved for system use)	*/
dummy,	/* (194)	H'00000308	(Reserved for system use)	*/
dummy,	/* (195)	H'0000030C	(Reserved for system use)	*/
dummy,	/* (196)	H'00000310	(Reserved for system use)	*/
dummy,	/* (197)	H'00000314	(Reserved for system use)	*/
dummy,	/* (198)	H'00000318	(Reserved for system use)	*/
dummy,	/* (199)	H'0000031C	(Reserved for system use)	*/
dummy,	/* (200)	H'00000320	(Reserved for system use)	*/
dummy,	/* (201)	H'00000324	(Reserved for system use)	*/
dummy,	/* (202)	H'00000328	(Reserved for system use)	*/
dummy,	/* (203)	H'0000032C	(Reserved for system use)	*/
dummy,	/* (204)	H'00000330	(Reserved for system use)	*/
dummy,	/* (205)	H'00000334	(Reserved for system use)	*/
dummy,	/* (206)	H'00000338	(Reserved for system use)	*/
dummy,	/* (207)	H'0000033C	(Reserved for system use)	*/
dummy,	/* (208)	H'00000340	(Reserved for system use)	*/
dummy,	/* (209)	H'00000344	(Reserved for system use)	*/
dummy,	/* (210)	H'00000348	(Reserved for system use)	*/
dummy,	/* (211)	H'0000034C	(Reserved for system use)	*/
dummy,	/* (212)	H'00000350	(Reserved for system use)	*/
dummy,	/* (213)	H'00000354	(Reserved for system use)	*/
dummy,	/* (214)	H'00000358	(Reserved for system use)	*/
dummy,	/* (215)	H'0000035C	(Reserved for system use)	*/
dummy,	/* (216)	H'00000360	(Reserved for system use)	*/
dummy,	/* (217)	H'00000364	(Reserved for system use)	*/
dummy,	/* (218)	H'00000368	(Reserved for system use)	*/
dummy,	/* (219)	H'0000036C	(Reserved for system use)	*/
dummy,	/* (220)	H'00000370	(Reserved for system use)	*/
dummy,	/* (221)	H'00000374	(Reserved for system use)	*/
dummy,	/* (222)	H'00000378	(Reserved for system use)	*/
dummy,	/* (223)	H'0000037C	(Reserved for system use)	*/
dummy,	/* (224)	H'00000380	(Reserved for system use)	*/
dummy,	/* (225)	H'00000384	(Reserved for system use)	*/
dummy,	/* (226)	H'00000388	(Reserved for system use)	*/
dummy,	/* (227)	H'0000038C	(Reserved for system use)	*/
dummy,	/* (228)	H'00000390	(Reserved for system use)	*/
dummy,	/* (229)	H'00000394	(Reserved for system use)	*/
dummy,	/* (230)	H'00000398	(Reserved for system use)	*/
dummy,	/* (231)	H'0000039C	(Reserved for system use)	*/
dummy,	/* (232)	H'000003A0	(Reserved for system use)	*/
dummy,	/* (233)	H'000003A4	(Reserved for system use)	*/
dummy,	/* (234)	H'000003A8	(Reserved for system use)	*/
dummy,	/* (235)	H'000003AC	(Reserved for system use)	*/
dummy,	/* (236)	H'000003B0	(Reserved for system use)	*/
dummy,	/* (237)	H'000003B4	(Reserved for system use)	*/
dummy,	/* (238)	H'000003B8	(Reserved for system use)	*/
dummy,	/* (239)	H'000003BC	(Reserved for system use)	*/
dummy,	/* (240)	H'000003C0	(Reserved for system use)	*/
dummy,	/* (241)	H'000003C4	(Reserved for system use)	*/
dummy,	/* (242)	H'000003C8	(Reserved for system use)	*/

図 2.2 ベクタ定義ファイル (続き)

dummy,	/* (243) H'000003CC	(Reserved for system use)	*/
dummy,	/* (244) H'000003D0	(Reserved for system use)	*/
dummy,	/* (245) H'000003D4	(Reserved for system use)	*/
dummy,	/* (246) H'000003D8	(Reserved for system use)	*/
dummy,	/* (247) H'000003DC	(Reserved for system use)	*/

図 2.2 ベクタ定義ファイル (続き)

### 2.1.3 レジスタ定義ファイル説明

SH7145F のレジスタ定義を「3.1 SH7145F レジスタ定義ファイル」に示します。

## 2.2 シングルマスタ送信

### 2.2.1 仕様

- (1) SH7145FのI<sup>2</sup>Cバスインタフェースのチャンネル0を使用して、EEPROM (HN58X2464、64k bit, 8word×8bit) に10バイトのデータを書き込みます。
- (2) 接続するEEPROMのスレーブアドレスは [1010000] とし、EEPROMメモリアドレスのH'0000番地からH'0009番地にデータを書き込みます。
- (3) 書き込むデータは [H'01,H'02,H'03,H'04,H'05,H'06,H'07,H'08,H'09,H'0A] とします。
- (4) 本システムのI<sup>2</sup>Cバスに接続されているデバイスは、マスタデバイス (SH7145F) 1個、スレーブデバイス (EEPROM) 1個のシングルマスタ構成とします。
- (5) I<sup>2</sup>Cバスのデータ転送クロックの周波数は156kHzとします。
- (6) SH7145F動作周波数は、CPUクロック40MHz、内蔵周辺クロック40MHzとします。
- (7) 図2.3にSH7145FとEEPROMの接続例を示します。

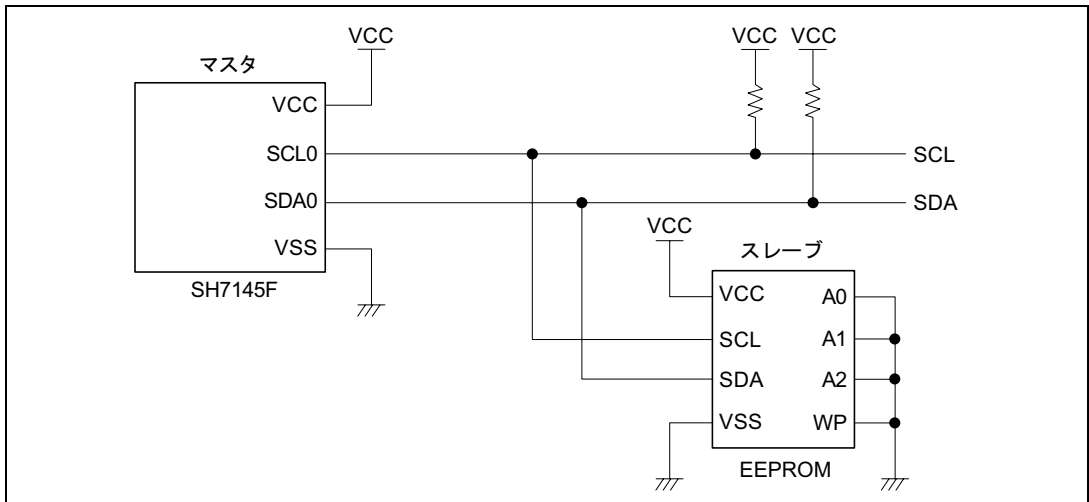


図 2.3 SH7145F と EEPROM との接続例





## 2.2.2 動作説明

図 2.5 に動作原理を示します。

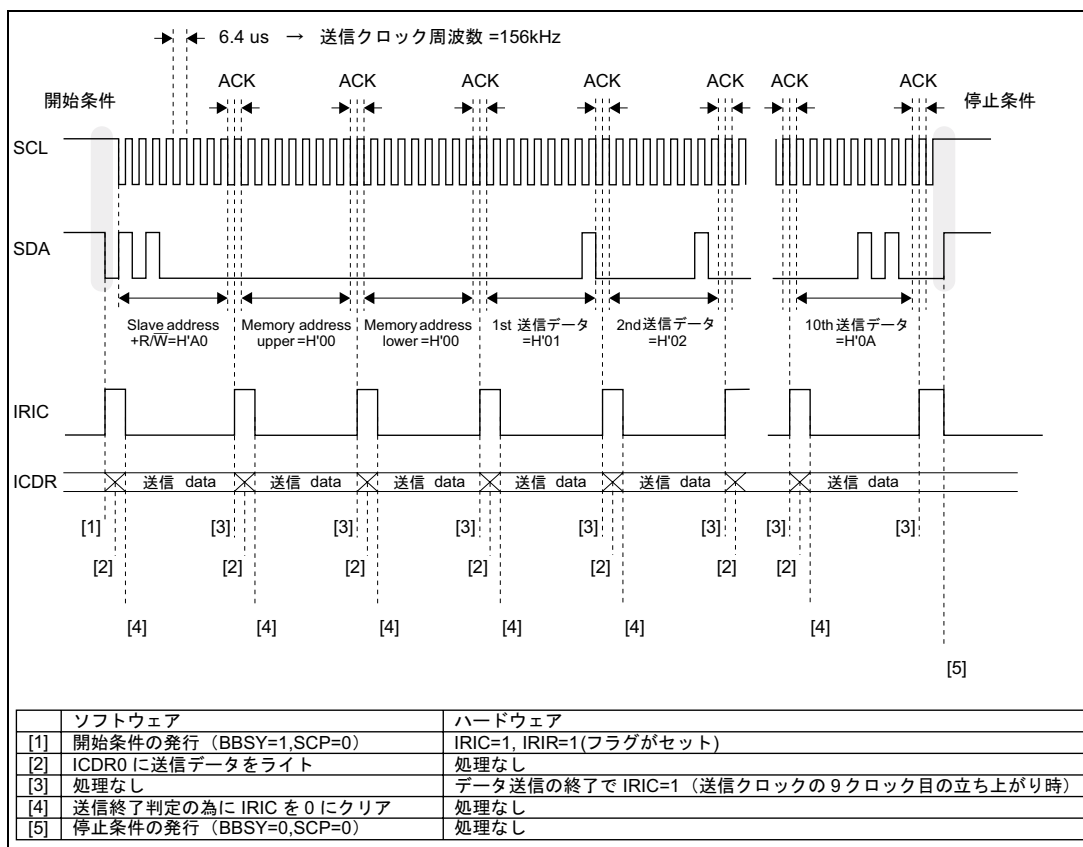


図 2.5 シングルマスタ送信動作原理

## 2.2.3 ソフトウェア説明

## (1) モジュール説明

下表に、本タスク例のモジュールを示します。

表 2.1 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	I <sup>2</sup> C の初期設定、端子設定
ダミー割り込みルーチン	dummy	ダミーの割り込み処理
EEPROM ライトルーチン	Write_page_EEPROM	n バイトの EEPROM 書き込みルーチン
アドレスセットルーチン	Set_adrs_EEPROM	開始条件の生成、スレーブアドレス発行、EEPROM のアドレス設定

## (2) 使用内蔵レジスタ説明

下表に、本タスク例の使用する内部レジスタを示します。

表 2.2 使用内蔵レジスタ説明

レジスタ名		機能	アドレス	設定値
ビット			ビット	
MSTCR1		モジュールスタンバイコントロールレジスタ 1	H'FFFF861C	
	MSTP21	I <sup>2</sup> C モジュールスタンバイ制御ビット :MSTP21=0 のとき、モジュールスタンバイ解除	ビット 5	B'0
PBCR1		ポート B コントロールレジスタ 1 ポート B コントロールレジスタ 2 と組み合わせて、ポート B の端子機能を設定	H'FFFF8398	H'0C00
PBCR2		ポート B コントロールレジスタ 2 ポート B コントロールレジスタ 1 と組み合わせて、ポート B の端子機能を設定、 ポート B3 (PB3) の端子機能を I <sup>2</sup> C の SDA0 入出力端子に設定 ポート B2 (PB2) の端子機能を I <sup>2</sup> C の SCL0 入出力端子に設定	H'FFFF839A	H'0000
ICDR0		I <sup>2</sup> C バスデータレジスタ 8 ビットの読み出し/書き込み可能なレジスタ。送信時は送信用データレジスタとして、受信時は受信用データレジスタとして使用。	H'FFFF880E	-
SAR0		スレーブアドレスレジスタ	H'FFFF880F	H'00
	SVA6-0	スレーブアドレス SVA6~SVA0 ビットには I <sup>2</sup> C バスにつながっているほかのスレーブと異なるユニークなアドレスを設定。	ビット 7-1	
	FS	フォーマットセレクト SARX の FSX ビットとともに、転送フォーマットを選択。 FS=FSX=0 のとき、転送フォーマットは、I <sup>2</sup> C バスフォーマット。	ビット 0	

レジスタ名		機能	アドレス	設定値
ビット			ビット	
SARX0		第2スレーブアドレスレジスタ	H'FFFF880E	H'00
	SVAX6-0	第2スレーブアドレス SVAX6~SVAX0 ビットにはI <sup>2</sup> Cバスにつながっているほかのスレーブと異なるユニークなアドレスを設定。	ビット7-1	
	FSX	フォーマットセレクト SARのFSビットとともに、転送フォーマットを選択。 FS=FSX=0のとき、転送フォーマットは、I <sup>2</sup> Cバスフォーマット。	ビット0	
ICMR0		I <sup>2</sup> Cバスモードレジスタ	H'FFFF880F	H'38
	MLS	MSBファースト/LSBファースト選択。 MLS=0のとき、MSBファースト	ビット7	
	WAIT	ウェイト挿入ビット。 WAIT=0のとき、データとアクノレッジを連続に転送	ビット6	
	CKS2 CKS1 CKS0	転送クロック選択2~0 SCRXレジスタのIICX0ビットと組み合わせて、転送クロックの周波数を設定。IICX=B'1,CKS[2:0]=B'111のとき、156kHz(Pφ=40MHz)	ビット5 ビット4 ビット3	
	BC2 BC1 BC0	ビットカウンタ。 I <sup>2</sup> Cバスフォーマットで次に転送するデータのビット数を9ビット(ACKビットを含む)/フレームに設定。 BC[2:0]=B'000。	ビット2 ビット1 ビット0	
ICCR0		I <sup>2</sup> Cバスコントロールレジスタ	H'FFFF8808	H'89
	ICE	I <sup>2</sup> Cバスインタフェースイネーブル(ICE) ICE=B'1のとき、I <sup>2</sup> Cモジュールは転送動作可能状態 ICMR、ICDRレジスタが有効	ビット7	
	IEIC	I <sup>2</sup> Cバスインタフェース割り込みイネーブル IEIC=B'0のとき、割り込み要求を禁止	ビット6	
	MST	マスタ/スレーブ選択 MST=B'0のとき、スレーブモード	ビット5	
	TRS	送信/受信選択 TES=B'0のとき、受信モード	ビット4	
	ACKE	アクノリッジビット判定選択 ACKE=B'1のとき、アクノレッジビットが1の場合、連続的な転送を中断	ビット3	
	BBSY	バスビジー BBSY=B'0のとき、バス開放状態	ビット2	
	IRIC	I <sup>2</sup> Cバスインタフェース割り込み要求フラグ IRIC=B'1のとき、割り込みが発生	ビット1	
SCP	開始条件/停止条件発行禁止ビット SCP=B'0のとき、BBSYフラグと組み合わせて開始条件、停止条件を発行	ビット0		

## 2. SH7144F グループ応用例

レジスタ名		機能	アドレス	設定値
ビット			ビット	
ICSR0		I <sup>2</sup> C パスステータスレジスタ	H'FFFF8809	-
	ESTP	エラー停止条件検出フラグ	ビット 7	
	STOP	正常停止条件検出フラグ	ビット 6	
	IRTR	I <sup>2</sup> C バスインタフェース連続送受信割り込み要求フラグ	ビット 5	
	AASX	第 2 スレーブアドレス認識フラグ	ビット 4	
	AL	アービトレーションロストフラグ	ビット 3	
	AAS	スレーブアドレス認識フラグ	ビット 2	
	ADZ	ゼネラルコールアドレス認識フラグ	ビット 1	
	ACKB	アクノリッジビット。 アクノリッジデータを格納。	ビット 0	
SCRX		シリアルコントロールレジスタ X	H'FFFF87F0	H'39
	リザーブ	リザーブビットです。 読み出すと 0 が読み出されます。書き込む値も常に 0 としてください	ビット 7,6	
	IICX0	I <sup>2</sup> C トランスファレートセレクト 0 ICMR の CKD[2:0] と組み合わせて、マスタモードでの転送レートを選択。 IICX0=B'1	ビット 5	
	IICE	I <sup>2</sup> C マスタイネーブル IICE=B'1 のとき、I2C バスインタフェースのレジスタアクセスを許可	ビット 4	
	HNDS	ハンドシェイク受信ビット HNDS=B'1 のとき、連続受信動作を禁止。	ビット 3	
	リザーブ	リザーブビットです。 読み出すと 0 が読み出されます。書き込む値も常に 0 としてください。	ビット 2	
	ICDRF0	ICDR に有効な受信データの有無を示す。	ビット 1	
		STOPIM	停止条件検出割り込みマスク。 STOPIM=B'1 のとき、スレーブモードのとき、停止条件を検出した場合でも、IRIC 割り込みの発生を抑制する。	ビット 0

## (3) 変数説明

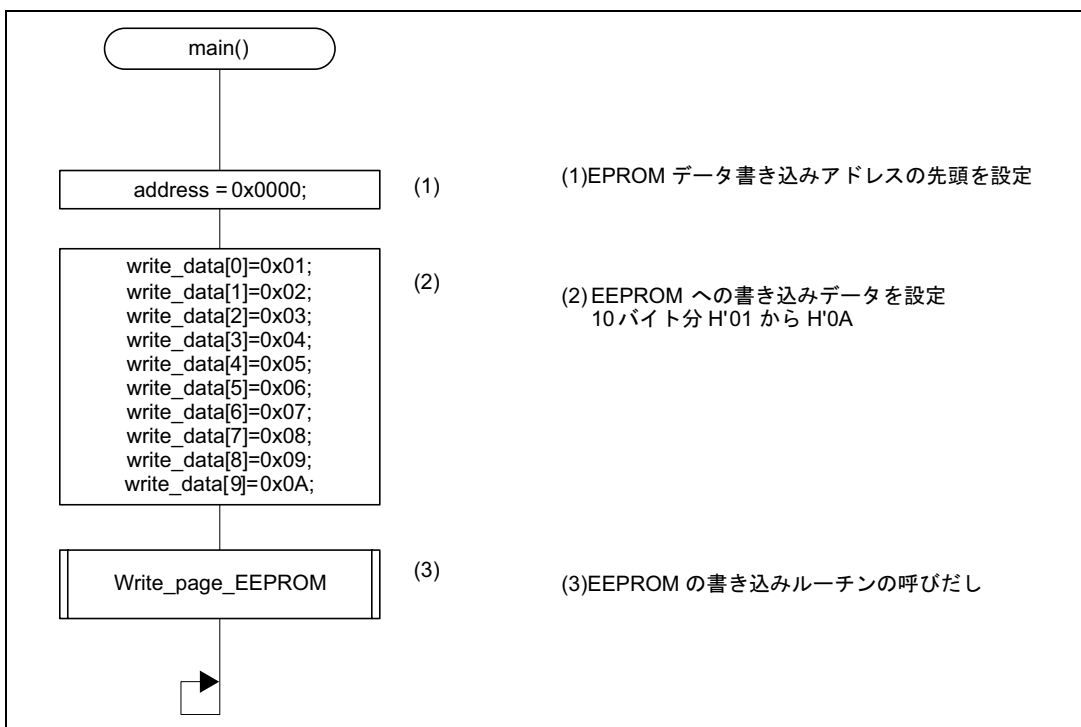
変数	機能	データ長	初期値	使用モジュール名
write_data[0]	1 バイト目送信データ	1 バイト	H'01	メインルーチン
write_data[1]	2 バイト目送信データ	1 バイト	H'02	メインルーチン
write_data[2]	3 バイト目送信データ	1 バイト	H'03	メインルーチン
write_data[3]	4 バイト目送信データ	1 バイト	H'04	メインルーチン
write_data[4]	5 バイト目送信データ	1 バイト	H'05	メインルーチン
write_data[5]	6 バイト目送信データ	1 バイト	H'06	メインルーチン
write_data[6]	7 バイト目送信データ	1 バイト	H'07	メインルーチン
write_data[7]	8 バイト目送信データ	1 バイト	H'08	メインルーチン
write_data[8]	9 バイト目送信データ	1 バイト	H'09	メインルーチン
write_data[9]	10 バイト目送信データ	1 バイト	H'0A	メインルーチン
address	EEPROM 書き込みスタートアドレス	2 バイト	H'0000	メインルーチン
adrs	EEPROM 書き込みスタートアドレスのコピー	2 バイト	-	EEPROM ライトルーチン
num	送信データ数	1 バイト	H'0A	EEPROM ライトルーチン
w_data	送信データ配列変数 write_data へのポインタ変数	4 バイト	-	EEPROM ライトルーチン
ack	アクノレッジ受信判定フラグ	1 バイト	H'01	EEPROM ライトルーチン

## (4) 使用 RAM 説明

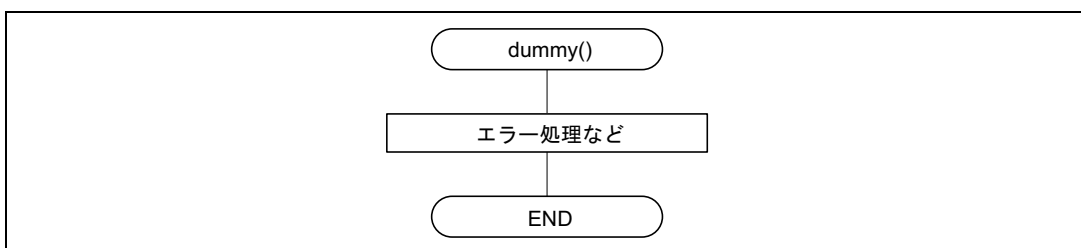
本タスク例では、変数以外の RAM は使用しません。

## 2.2.4 フローチャート

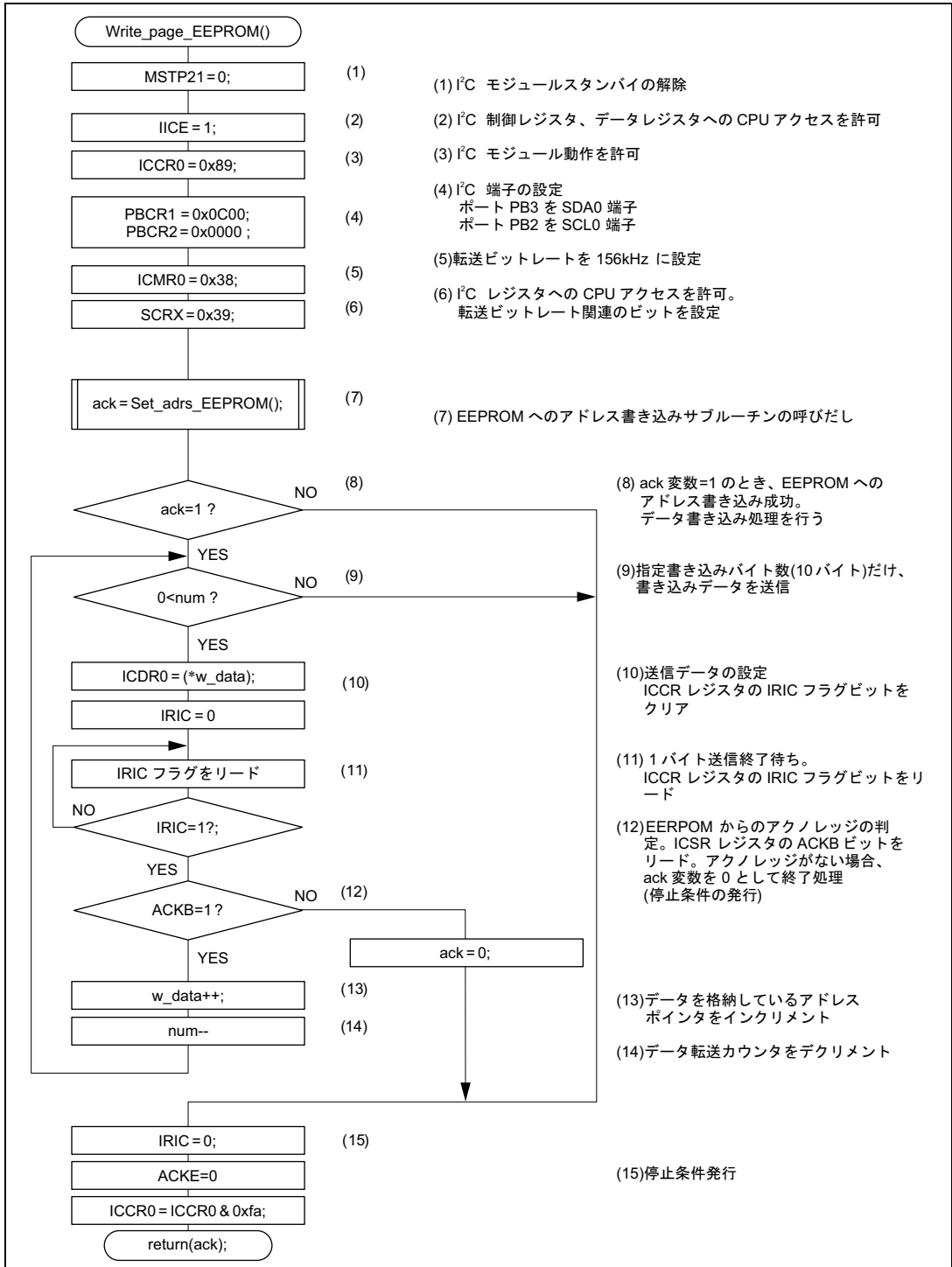
### (1) メインルーチン



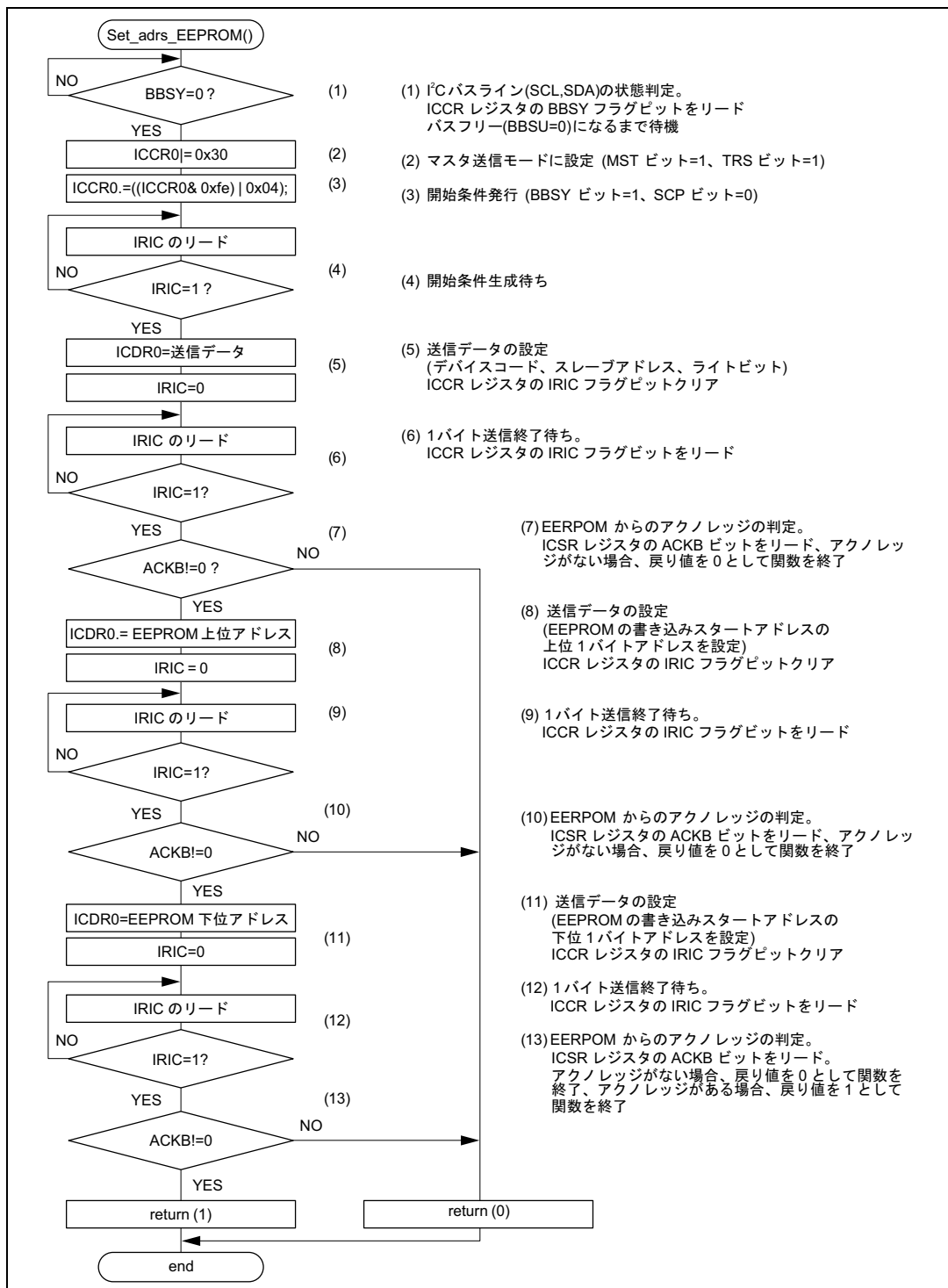
### (2) ダミー割り込みルーチン



(3) EEPROM 書き込みサブルーチン



(4) 開始条件発行、Slave address、EEPROM memory address 送信サブルーチン





## 2.2.5 プログラムリスト

```

/*****
// SH7144F Series -SH7145- I2C-bus Application Note
//   Single master transmit
//   n Byte data write/read 64kbit EEPROM
//   Clock :CPU=40MHz (External input=10MHz)
//         :Peripheral=40MHz
//   I2c bit rate :156kHz
//   Written:2003/2/1 Rev.2.0
/*****
#include <machine.h>
#include "iodefine.h" // SH7145 I/O Register Definition

//----- Symbol Definition -----
#define DEVICE_CODE      0xa0          // EEPROM DEVICE CODE:b'1010
#define SLAVE_ADRS      0x00          // SLAVE ADRS:b'00
#define IIC_DATA_W      0x00          // WRITE DATA:b'0
#define IIC_DATA_R      0x01          // READ DATA:b'1
#define DATA_NUM       10           // data size

//----- Function Definition -----
void main(void);
void dummy(void);

unsigned char Write_page_EEPROM(unsigned short,unsigned char*,unsigned char);
unsigned char Set_adrs_EEPROM(unsigned short);

/*****
//   main
/*****
void main(void)
{
    unsigned short address;          // EEPROM memory address

    address= 0x0000;                  // set EEPROM address

    // set write data
    write_data[0]=0x01;
    write_data[1]=0x02;
    write_data[2]=0x03;
    write_data[3]=0x04;
    write_data[4]=0x05;
    write_data[5]=0x06;
    write_data[6]=0x07;
    write_data[7]=0x08;
    write_data[8]=0x09;
    write_data[9]=0x0a;

    // EEPROM data write
    Write_page_EEPROM(address,write_data,DATA_NUM);

    while(1);
}

/*****
//   dummy interrupt function
/*****

```

## 2. SH7144F グループ応用例

```
#pragma interrupt(dummy)
void dummy(void)
{
    // Interrupt error
}

//*****
// Write_page_EEPROM
//   argument1 ;write address(unsigned short)
//   argument2 ;write data(unsigned char)
//   argument3 ;write data number(unsigned char)
//   return    ;1=OK/0=NG EEPROM NO_ACK(unsigned char)
//*****
unsigned char Write_page_EEPROM(unsigned short adrs,unsigned char* w_data,unsigned char num)
{
    unsigned char ack;          // ACK check flag

    // Set standby mode
    P_STBY.MSTCR1.BIT.MSTP21 = 0; // disable I2C standby mode

    ack = 1;
    P_IIC.SCRX.BIT.IICE = 1;      // Enables CPU access to the register

    P_IIC.ICCR0.BYTE = 0x89;
        // ICE(7)=b'1          Enable I2C bus interface
        // IEIC(6)=b'0        Disables the interrupt
        // MST(5)=b'0         Slave mode
        // TRS(4)=b'0         Receive mode
        // ACKE(3)=b'1        Continuous data transfer is halted
        // BBSY(2)=b'0
        // IRIC(1)=b'0
        // SCP(0)=b'1         Start/stop condition issuance disabling

    // set I2C pin function
    P_PORTB.PBCR1.WORD = 0x0c00;  // SDA0(PB3-32pin@SH7145F),SCL0(PB2-31pin@SH7145F)

    P_PORTB.PBCR2.WORD = 0x0000;

    P_IIC.ICMR0.BYTE = 0x38;
        // MILS(7)=b'0        MSB first
        // WAIT(6)=b'0        A wait state is inserted between DATA and ACK
        // CKS2[2:0](5:3)=b'111 Transfer clock select
        //   156kHz@(@P-fai=40MHz,IICX=1)
        //   39.1kHz@(@P-fai=10MHz,IICX=1)
    P_IIC.SCRX.BYTE = 0x39;
        // IICX(5)=b'1        transfer-rate select, reference CKS bit
        // IICE(4)=b'1        Enables CPU access to the register
        // HNDS(3)=b'1
        // STOPIM(0)=b'1      disables interrupt requests

    // Set device code,EEPROM address
    ack = Set_adrs_EEPROM(adrs);

    // EEPROM write data Transmission (n byte)
    if( ack==1 ){
        for( ; 0<num; num-- ){
            P_IIC.ICDR0.BYTE = (*w_data); // write data set
            P_IIC.ICCR0.BIT.IRIC = 0;    // clear IRIC
            while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait lbyte transmitted
            if( P_IIC.ICSR0.BIT.ACKB != 0 ){ // Test the acknowledge bit
                ack = 0; // no ACK
                break;
            }
        }
    }
}
```

```

        w_data++;           // write data pointer increment
    }
}

// Stop condition issuance
P_IIC.ICCR0.BIT.IRIC = 0;           // clear IRIC
P_IIC.ICCR0.BIT.ACKE = 0;          // set AKCE=0
P_IIC.ICCR0.BYTE = P_IIC.ICCR0.BYTE & 0xfa;    // Stop condition
issuance(BBSY=0,SCP=0)

return(ack);
}

/*****
// Set_adrs_EEPROM
// argument1 ;write address(unsigned short)
// return ;1=OK/0=NG EEPROM NO_ACK(unsigned char)
*****/
unsigned char Set_adrs_EEPROM(unsigned short adrs)
{
    while( P_IIC.ICCR0.BIT.BBSY!=0 );           // BUS FREE?(BBSY==0→Bus Free)

    // Master-Transmission, Generate the start condition.
    P_IIC.ICCR0.BYTE |= 0x30;                   // Select master transmit mode(MST=1,TRS=1)
    P_IIC.ICCR0.BYTE=(P_IIC.ICCR0.BYTE & 0xfe | 0x04); // Generate start
    condition(BBSY=1,SCP=0)
    while( P_IIC.ICCR0.BIT.IRIC==0 );           // Wait for a start condition generation

    // Slave address+W Transmission
    P_IIC.ICDR0.BYTE = (unsigned char)(DEVICE_CODE|SLAVE_ADRS|IIC_DATA_W); // data set
    P_IIC.ICCR0.BIT.IRIC = 0;                   // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 );           // Wait lbyte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){              // Test the acknowledge bit
        return (0);                             // no ACK
    }

    // EEPROM upper address Transmission(lbyte)
    P_IIC.ICDR0.BYTE = (unsigned char)(adrs>>8); // data set
    P_IIC.ICCR0.BIT.IRIC = 0;                   // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 );           // Wait lbyte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){              // Test the acknowledge bit
        return (0);                             // no ACK
    }

    // EEPROM lower address Transmission(lbyte)
    P_IIC.ICDR0.BYTE = (unsigned char)(adrs & 0x00ff); // data set
    P_IIC.ICCR0.BIT.IRIC = 0;                   // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 );           // Wait lbyte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){              // Test the acknowledge bit
        return (0);                             // no ACK
    }

    return (1);                                 // ACK OK
}

```

## 2.3 シングルマスタ受信

### 2.3.1 仕様

- (1) SH7145FのI<sup>2</sup>Cバスインタフェースのチャンネル0を使用して、EEPROM (HN58X2464、64k bit ,8word×8bit) から10バイトのデータを読み込みます。
- (2) 接続するEEPROMのスレーブアドレスは [1010000] とし、EEPROMメモリアドレスのH'0000番地からH'0009番地までのデータを読み込みます。
- (3) 読み込んだデータは、変数の配列に取り込みます。
- (4) 本システムのI<sup>2</sup>Cバスに接続されているデバイスは、マスタデバイス (SH7145F) 1個、スレーブデバイス (EEPROM) 1個のシングルマスタ構成とします。
- (5) I<sup>2</sup>Cバスのデータ転送クロックの周波数は156kHzとします。
- (6) SH7145F動作周波数は、CPUクロック40MHz、内蔵周辺クロック40MHzとします。
- (7) 図2.6にSH7145FとEEPROMの接続例を示します。

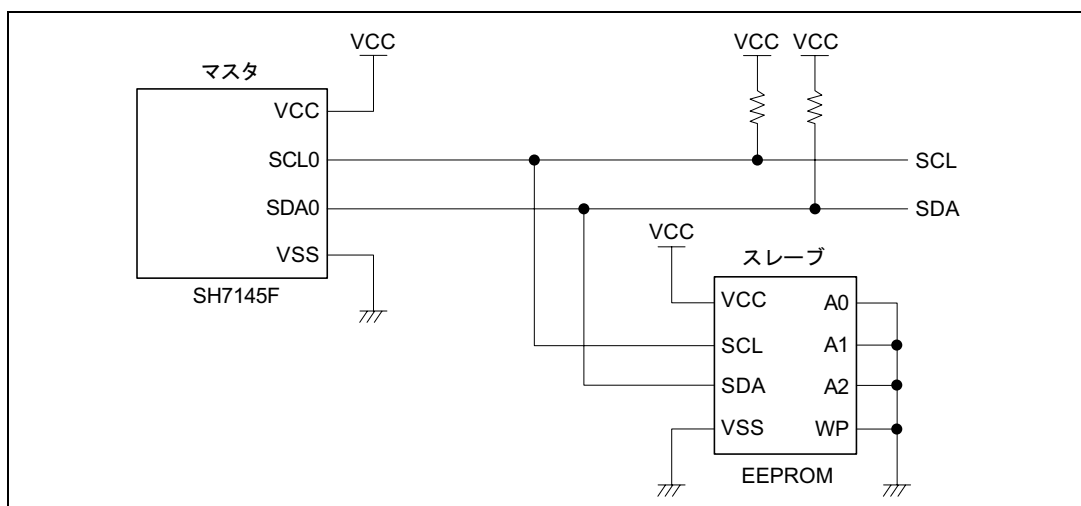


図 2.6 SH7145F と EEPROM との接続例

本タスクで例で使用する I<sup>2</sup>C バスフォーマットを図 2.7 に示します。

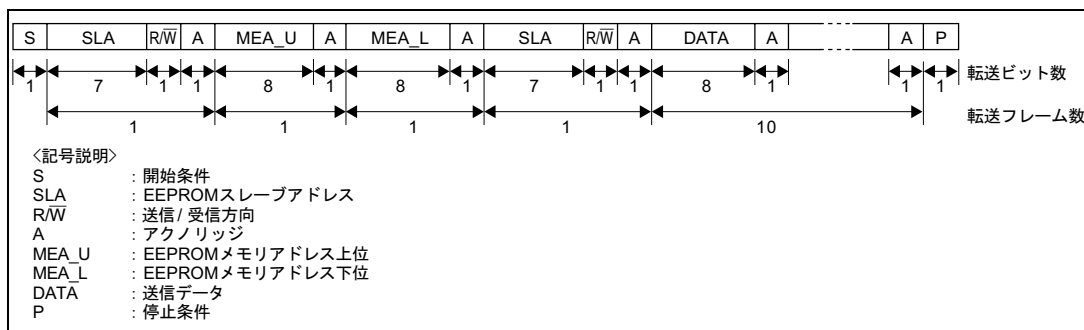


図 2.7 本タスクで使用する転送フォーマット

## 2.3.2 動作説明

図 2.8 に動作原理の概要を示します。

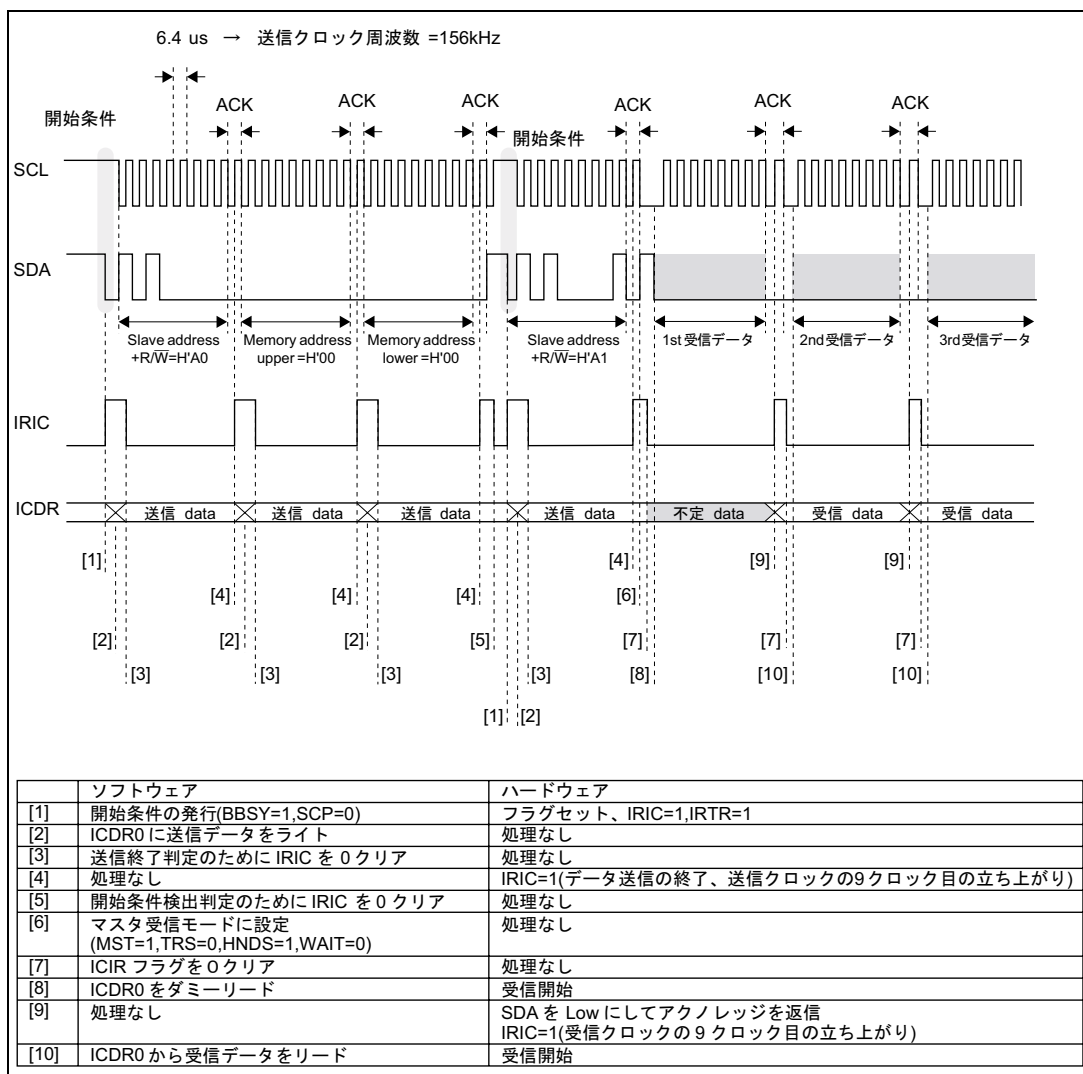


図 2.8 シングルマスタ受信動作原理

## 2.3.3 ソフトウェア説明

## (1) モジュール説明

下表に、本タスク例のモジュールをに示します。

表 2.3 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	I <sup>2</sup> C の初期設定、端子設定
ダミー割り込みルーチン	dummy	ダミーの割り込み処理
EEPROM リードルーチン	Read_page_EEPROM	N バイトの EEPROM 読み出しルーチン(n>1)
アドレスセットルーチン	Set_adrs_EEPROM	開始条件の生成、スレーブアドレス発行、EEPROM のアドレス設定

## (2) 使用内蔵レジスタ説明

下表に、本タスク例の使用する内部レジスタを示します。

表 2.4 使用内蔵レジスタ説明

レジスタ名	ビット	機能	アドレス	設定値
			ビット	
MSTCR1		モジュールスタンバイコントロールレジスタ 1	H'FFFF861C	
	MSTP21	I <sup>2</sup> C モジュールスタンバイ制御ビット :MSTP21=0 のとき、モジュールスタンバイ解除	ビット 5	B'0
PBCR1		ポート B コントロールレジスタ 1 ポート B コントロールレジスタ 2 と組み合わせて、ポート B の端子機能を設定	H'FFFF8398	H'0C00
PBCR2		ポート B コントロールレジスタ 2 ポート B コントロールレジスタ 1 と組み合わせて、ポート B の端子機能を設定、 ポート B3(PB3)の端子機能を I <sup>2</sup> C の SDA0 入出力端子に設定 ポート B2(PB2)の端子機能を I <sup>2</sup> C の SCL0 入出力端子に設定	H'FFFF839A	H'0000
ICDR0		I <sup>2</sup> C バスデータレジスタ 8 ビットの読み出し/書き込み可能なレジスタ。送信時は送信用データレジスタとして、受信時は受信用データレジスタとして使用。	H'FFFF880E	-
SAR0		スレーブアドレスレジスタ	H'FFFF880F	H'00
	SVA6-0	スレーブアドレス SVA6~SVA0 ビットには I <sup>2</sup> C バスにつながっているほかのスレーブと異なるユニークなアドレスを設定。	ビット 7-1	
	FS	フォーマットセレクト SARX の FSX ビットとともに、転送フォーマットを選択。 FS=FSX=0 のとき、転送フォーマットは、I <sup>2</sup> C バスフォーマット。	ビット 0	

レジスタ名		機能	アドレス	設定値
ビット			ビット	
SARX0		第2スレーブアドレスレジスタ	H'FFFF880E	H'00
	SVAX6-0	第2スレーブアドレス SVAX6~SVAX0ビットにはI <sup>2</sup> Cバスにつながっているほかのスレーブと異なるユニークなアドレスを設定。	ビット7-1	
	FSX	フォーマットセレクト SARのFSビットとともに、転送フォーマットを選択。 FS=FSX=0のとき、転送フォーマットは、I <sup>2</sup> Cバスフォーマット。	ビット0	
ICMR0		I <sup>2</sup> Cバスモードレジスタ	H'FFFF880F	H'38
	MLS	MSBファースト/LSBファースト選択。 MLS=0のとき、MSBファースト	ビット7	
	WAIT	ウェイト挿入ビット。 WAIT=0のとき、データとアクノレッジを連続に転送	ビット6	
	CKS2 CKS1 CKS0	転送クロック選択2~0 SCRXレジスタのIICX0ビットと組み合わせて、転送クロックの周波数を設定。IICX=B'1,CKS[2:0]=B'111のとき、156kHz(Pφ=40MHz)	ビット5 ビット4 ビット3	
	BC2 BC1 BC0	ビットカウンタ。 I <sup>2</sup> Cバスフォーマットで次に転送するデータのビット数を9ビット(ACKビットを含む)/フレームに設定。 BC[2:0]=B'000。	ビット2 ビット1 ビット0	
ICCR0		I <sup>2</sup> Cバスコントロールレジスタ	H'FFFF8808	H'89
	ICE	I <sup>2</sup> Cバスインタフェースイネーブル(ICE) ICE=B'1のとき、I <sup>2</sup> Cモジュールは転送動作可能状態ICMR、ICDRレジスタが有効	ビット7	
	IEIC	I <sup>2</sup> Cバスインタフェース割り込みイネーブル IEIC=B'0のとき、割り込み要求を禁止	ビット6	
	MST	マスタ/スレーブ選択 MST=B'0のとき、スレーブモード	ビット5	
	TRS	送信/受信選択 TES=B'0のとき、受信モード	ビット4	
	ACKE	アクノリッジビット判定選択 ACKE=B'1のとき、アクノレッジビットが1の場合、連続的な転送を中断	ビット3	
	BBSY	バスビジー BBSY=B'0のとき、バス開放状態	ビット2	
	IRIC	I <sup>2</sup> Cバスインタフェース割り込み要求フラグ IRIC=B'1のとき、割り込みが発生	ビット1	
SCP	開始条件/停止条件発行禁止ビット SCP=B'0のとき、BBSYフラグと組み合わせて開始条件、停止条件を発行	ビット0		

## 2. SH7144F グループ応用例

レジスタ名		機能	アドレス	設定値
	ビット		ビット	
ICSR0		I <sup>2</sup> C バスステータスレジスタ	H'FFFF8809	-
	ESTP	エラー停止条件検出フラグ	ビット 7	
	STOP	正常停止条件検出フラグ	ビット 6	
	IRTR	I <sup>2</sup> C バスインタフェース連続送受信割り込み要求フラグ	ビット 5	
	AASX	第 2 スレーブアドレス認識フラグ	ビット 4	
	AL	アービトレーションロストフラグ	ビット 3	
	AAS	スレーブアドレス認識フラグ	ビット 2	
	ADZ	ゼネラルコールアドレス認識フラグ	ビット 1	
	ACKB	アクノリッジビット。 アクノリッジデータを格納。	ビット 0	
SCRX		シリアルコントロールレジスタ X	H'FFFF87F0	H'39
	リザーブ	リザーブビットです。 読み出すと 0 が読み出されます。書き込む値も常に 0 としてください。	ビット 7,6	
	IICX0	I <sup>2</sup> C トランスファレートセレクト 0 ICMR の CKD[2:0] と組み合わせて、マスタモードでの転送レートを選択。 IICX0=B'1	ビット 5	
	IICE	I <sup>2</sup> C マスタイネーブル IICE=B'1 のとき、I <sup>2</sup> C バスインタフェースのレジスタアクセスを許可	ビット 4	
	HNDS	ハンドシェイク受信ビット HNDS=B'1 のとき、連続受信動作を禁止。	ビット 3	
	リザーブ	リザーブビットです。 読み出すと 0 が読み出されます。書き込む値も常に 0 としてください。	ビット 2	
	ICDRF0	ICDR に有効な受信データの有無を示す。	ビット 1	
		STOPIM	停止条件検出割り込みマスク。 STOPIM=B'1 のとき、スレーブモードのとき、停止条件を検出した場合でも、IRIC 割り込みの発生を抑制する。	ビット 0



## (3) 変数説明

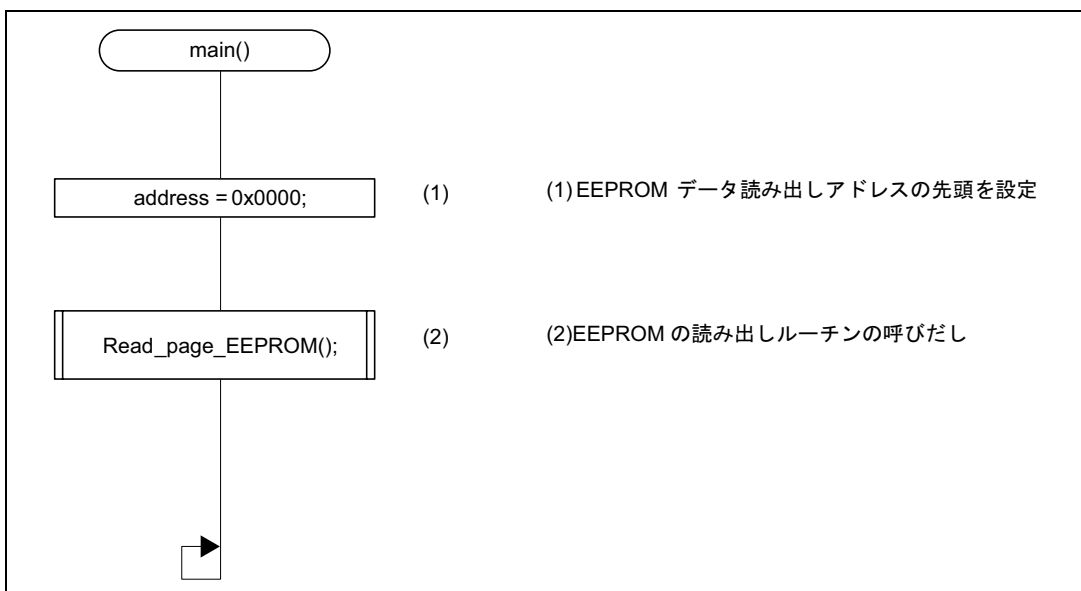
変数	機能	データ長	初期値	使用モジュール名
read_data[0]	1 バイト目受信データ	1 バイト	-	メインルーチン
read_data[1]	2 バイト目受信データ	1 バイト	-	メインルーチン
read_data[2]	3 バイト目受信データ	1 バイト	-	メインルーチン
read_data[3]	4 バイト目受信データ	1 バイト	-	メインルーチン
read_data[4]	5 バイト目受信データ	1 バイト	-	メインルーチン
read_data[5]	6 バイト目受信データ	1 バイト	-	メインルーチン
read_data[6]	7 バイト目受信データ	1 バイト	-	メインルーチン
read_data[7]	8 バイト目受信データ	1 バイト	-	メインルーチン
read_data[8]	9 バイト目受信データ	1 バイト	-	メインルーチン
read_data[9]	10 バイト目受信データ	1 バイト	-	メインルーチン
address	EEPROM 読み込みスタートアドレス	2 バイト	H'0000	メインルーチン
adrs	EEPROM 読み込みスタートアドレスのコピー	2 バイト	-	EEPROM リードルーチン
num	受信データ数	1 バイト	H'0A	EEPROM リードルーチン
r_data	受信データ配列変数 read_data へのポインタ変数	4 バイト	-	EEPROM リードルーチン
dummy	ダミーリードデータ	1 バイト	-	EEPROM リードルーチン
ack	アクノレッジ受信判定フラグ	1 バイト	H'01	EEPROM リードルーチン

## (4) 使用 RAM 説明

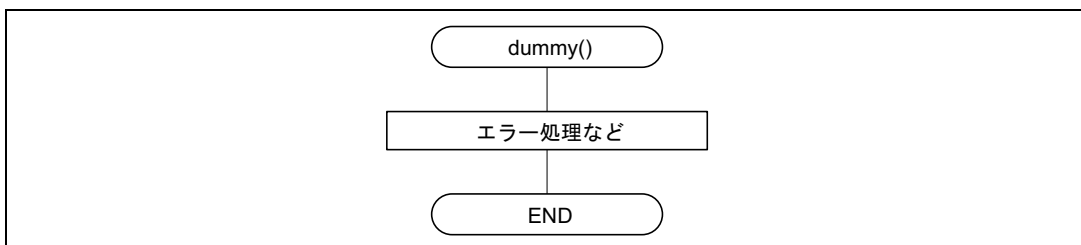
本タスク例では、変数以外の RAM は使用しません。

### 2.3.4 フローチャート

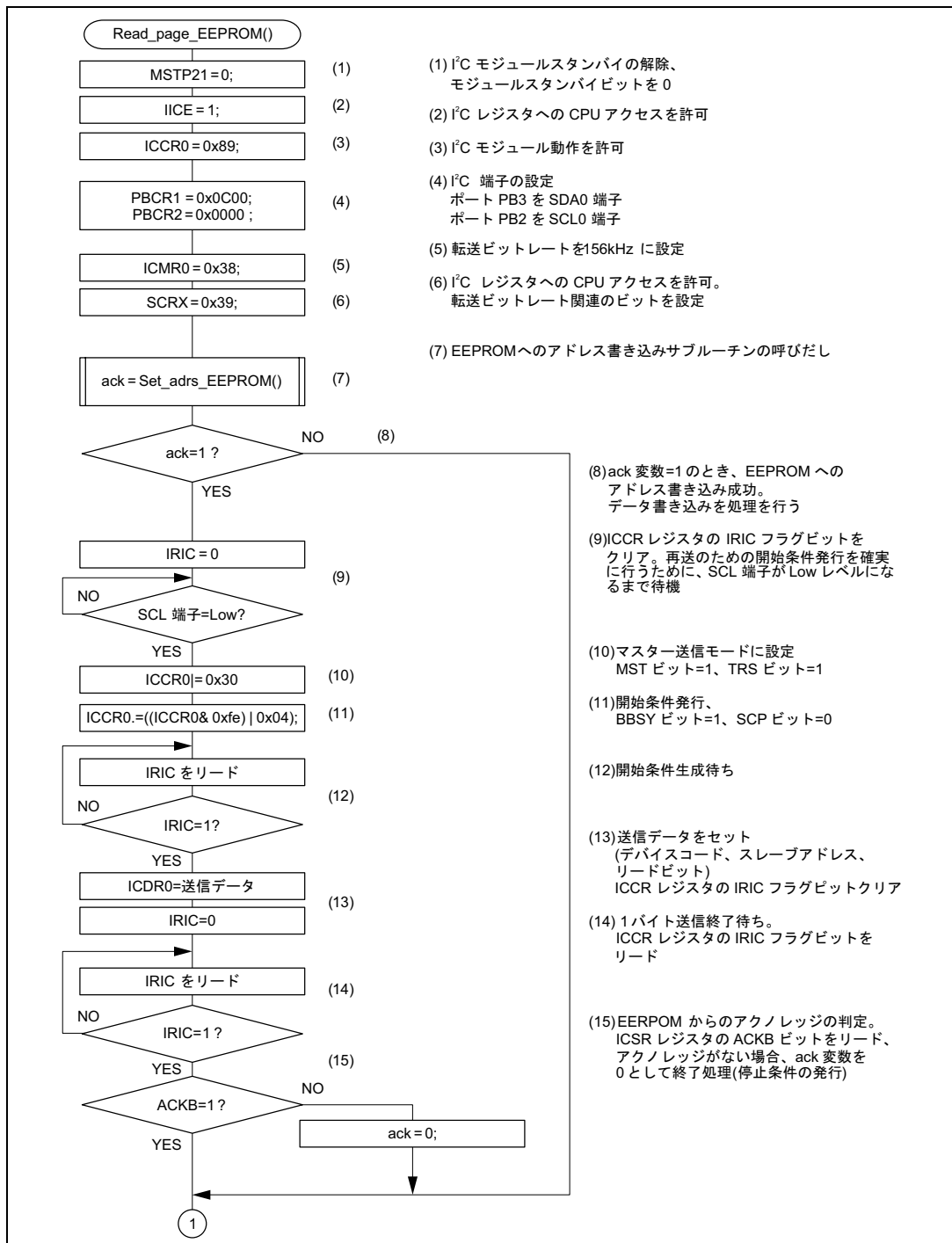
#### (1) メインルーチン



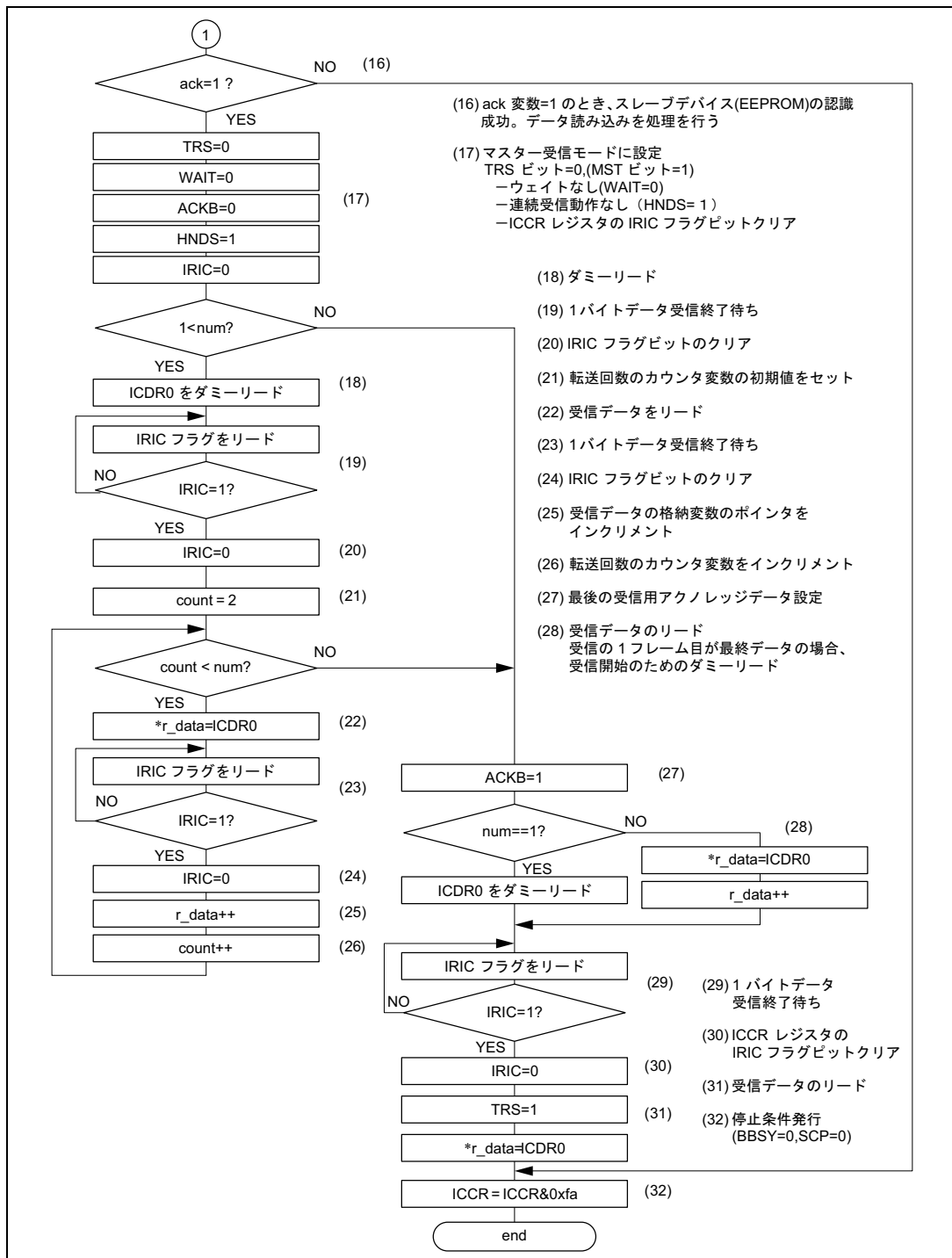
#### (2) ダミー割り込みルーチン



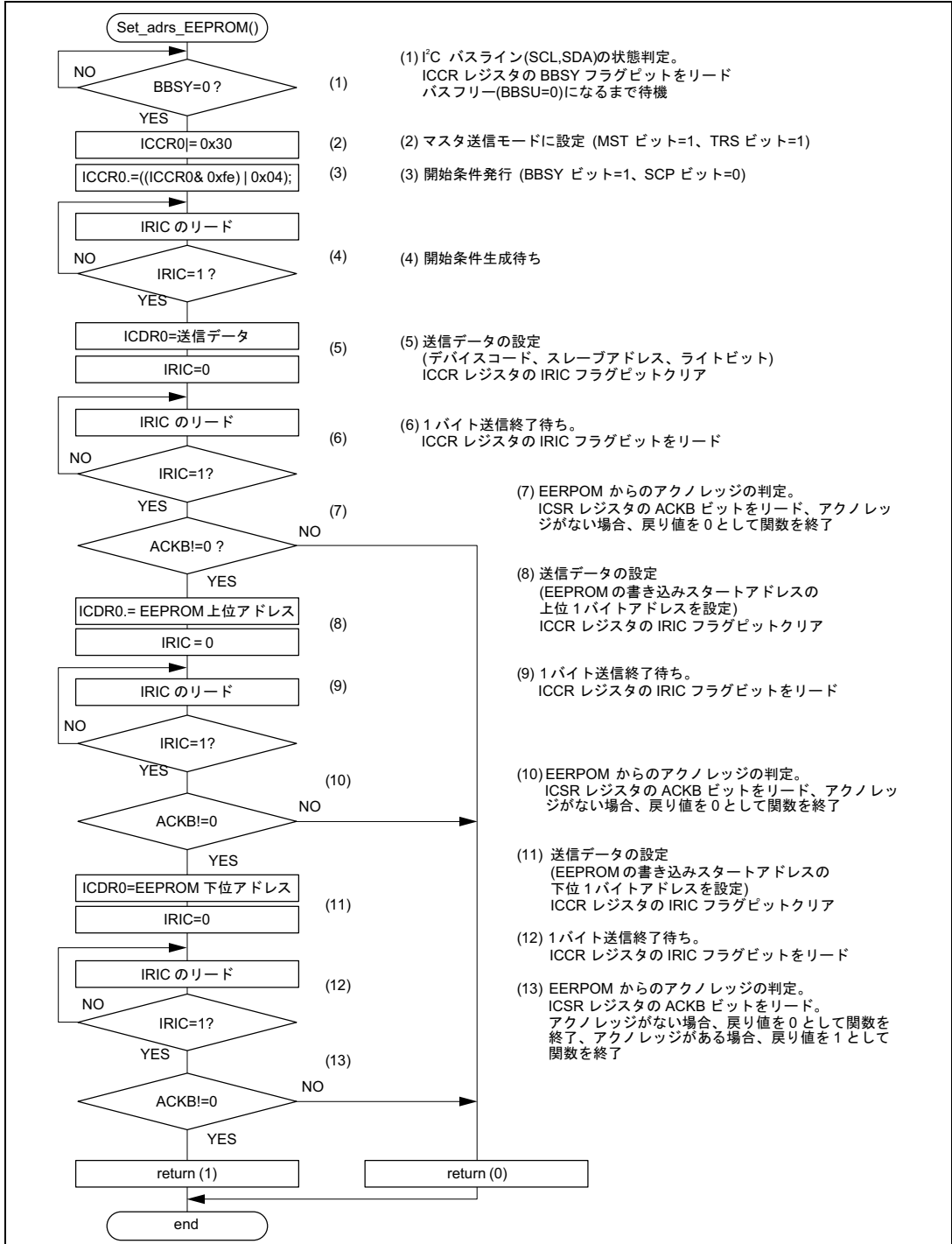
## (3) EEPROM 読み出しサブルーチン



## 2. SH7144F グループ応用例



(4) 開始条件発行、Slave address、EEPROM memory address 送信サブルーチン



## 2.3.5 プログラムリスト

```
/** *****  
// SH7144F Series -SH7145- I2C-bus Application Note  
// Single master receive  
// n Byte data write/read 64kbit EEPROM  
// Clock :CPU=40MHz (External input=10MHz)  
// :Peripheral=40MHz  
// I2c bit rate :156kHz  
// Written:2003/2/1 Rev.2.0  
/** *****  
#include <machine.h>  
#include "iodefine.h"  
  
//----- Symbol Definition -----  
#define DEVICE_CODE 0xa0 // EEPROM DEVICE CODE:b'1010  
#define SLAVE_ADRS 0x00 // SLAVE ADRS:b'000  
#define IIC_DATA_W 0x00 // WRITE DATA:b'0  
#define IIC_DATA_R 0x01 // READ DATA:b'1  
#define DATA_NUM 10 // data size  
  
//----- Function Definition -----  
void main(void);  
void dummy(void);  
  
unsigned char Read_page_EEPROM(unsigned short,unsigned char*,unsigned char);  
unsigned char Set_adrs_EEPROM(unsigned short);  
  
/** *****  
// main  
/** *****  
void main(void)  
{  
    unsigned short address; // EEPROM memory address  
    unsigned char read_data[DATA_NUM]; // read data  
  
    address= 0x0000; // set EEPROM address  
  
    // EEPROM data read  
    Read_page_EEPROM(address,read_data,DATA_NUM);  
  
    while(1);  
}  
  
/** *****  
// dummy interrupt function  
/** *****  
#pragma interrupt(dummy)  
void dummy(void)  
{  
    // Interrput error  
}
```

```

//*****
//      Read_page_EEPROM
//      argument1 ;read address(unsigned short)
//      argument2 ;read data(unsigned char)
//      argument2 ;read data number (unsigned char)
//      return    ;1=OK/0=NG EEPROM_NO_ACK(unsigned char)
//*****
unsigned char Read_page_EEPROM(unsigned short adrs,unsigned char* r_data,unsigned char num)
{
    unsigned char ack;    // ACK flag
    unsigned char count;  // read data number
    unsigned char dummy;  // dummy data

    // Set standby mode
    P_STBY.MSTCR1.BIT.MSTP21 = 0; // disable I2C standby mode

    ack =1;
    P_IIC.SCRX.BIT.IICE = 1;    // Enables CPU access to the register
    P_IIC.ICCR0.BYTE = 0x89;
        // ICE(7)=b'1    Enable I2C bus interface
        // IEIC(6)=b'0   Disables the interrupt
        // MST(5)=b'0    Slave mode
        // TRS(4)=b'0    Receive mode
        // ACKE(3)=b'1   Continuous data transfer is halted
        // BBSY(2)=b'0
        // IRIC(1)=b'0
        // SCP(0)=b'1    Start/stop condition issuance disabling

    // set I2C pin function
    P_PORTB.PBCR1.WORD = 0x0c00; // SDA0(PB3-32pin@SH7145F),SCL0(PB2-31pin@SH7145F)
    P_PORTB.PBCR2.WORD = 0x0000;

    P_IIC.ICMR0.BYTE = 0x38;
        // MILS(7)=b'0    MSB first
        // WAIT(6)=b'0    A wait state is inserted between DATA and ACK
        // CKS2[2:0](5:3)=b'111    Transfer clock select
        //                                156kHz@(@P-fai40MHz,IICX=1)
        //                                39.1kHz@(@P-fail0MHz,IICX=1)
    P_IIC.SCRX.BYTE = 0x39;
        // IICX(5)=b'1    transfer-rate select,reference CKS bit
        // IICE(4)=b'1    Enables CPU access to the register
        // HNDS(3)=b'1    Set this bit to 1
        // STOPIM(0)=b'1    disables interrupt requests

    // Set device code,EEPROM address
    ack = Set_adrs_EEPROM(adrs);    // set device code,EEPROM address

    if( ack==1){

        P_IIC.ICCR0.BIT.IRIC = 0;    // clear IRIC
        while(P_PORTB.PBDR.BIT.PB2DR!=0);    // check SCL0 pin state == low?

        // Master-Transmission, Generate the start condition.
        P_IIC.ICCR0.BYTE |= 0x30;    // Select master transmit
mode(MST=1,TRS=1)
        P_IIC.ICCR0.BYTE=((P_IIC.ICCR0.BYTE & 0xfe)|0x04); // Generate start
condition(BBSY=1,SCP=0)
        while( P_IIC.ICCR0.BIT.IRIC==0 );    // Wait for a start condition
generation
    }
}

```

## 2. SH7144F グループ 応用例

```
// Slave address+R Transmission
P_IIC.ICDR0.BYTE = (unsigned char)(DEVICE_CODE|SLAVE_ADRS|IIC_DATA_R); // data set
P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    ack = 0; // no ACK
}
}

if( ack==1 ){
    // Master receive operation (HNDS=1, WAIT=0)
    P_IIC.ICCR0.BIT.TRIS = 0; // Select receive mode(TRS=0)
    P_IIC.ICMR0.BIT.WAIT = 0; // set wait=0
    P_IIC.ICSR0.BIT.ACKB = 0; // set ACK data =0
    P_IIC.SCRX.BIT.HNDS = 1; // set HNDS bit =1

    P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC

    // Start data receiving
    if(num>1){ // case nByte data read (n>1)
        dummy = P_IIC.ICDR0.BYTE; // dummy read
        while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for 1 byte to be received
        P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
        for( count=2; count<num; count++ ){ // (num-2)byte read
            *r_data = P_IIC.ICDR0.BYTE; // read receive data
            while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for 1 byte to be received
            P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
            r_data++;
        }

        P_IIC.ICSR0.BIT.ACKB = 1; // set ACK data =1

        if(num==1){ // case 1Byte read
            dummy = P_IIC.ICDR0.BYTE; // dummy read
        }else{ // case nByte data read (n>1)
            *r_data = P_IIC.ICDR0.BYTE; // read receive data(n-1)
            r_data++;
        }
        while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for 1 byte to be received
        P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC

        // End data receiving
        P_IIC.ICCR0.BIT.TRIS = 1; // Select transmit mode
        *r_data = P_IIC.ICDR0.BYTE; // read END receive data
    }

    // Stop condition issuance
    P_IIC.ICCR0.BYTE = P_IIC.ICCR0.BYTE & 0xfa; // Stop condition issuance(BBSY=0,SCP=0)

    return(ack);
}

//*****
// Set_adrs_EEPROM
// argument1 ;write address(unsigned short)
// return ;1=OK/0=NG EEPROM_NO_ACK(unsigned char)
//*****
unsigned char Set_adrs_EEPROM(unsigned short adrs)
{
    while( P_IIC.ICCR0.BIT.BBSY!=0 ); // BUS FREE?(BBSY==0→Bus Free)
```



```
// Master-Transmission, Generate the start condition.
P_IIC.ICCR0.BYTE |= 0x30; // Select master transmit mode(MST=1,TRS=1)

P_IIC.ICCR0.BYTE=((P_IIC.ICCR0.BYTE & 0xfe) | 0x04); // Generate start
condition(BBSY=1,SCP=0)
while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for a start condition generation

// Slave address+W Transmission
P_IIC.ICDR0.BYTE = (unsigned char)(DEVICE_CODE|SLAVE_ADRS|IIC_DATA_W); // data set

P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait lbyte transmitted
if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    return (0); // no ACK
}
// EEPROM upper address Transmission(lbyte)
P_IIC.ICDR0.BYTE = (unsigned char)(adrs>>8); // data set
P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait lbyte transmitted
if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    return (0); // no ACK
}
// EEPROM lower address Transmission(lbyte)
P_IIC.ICDR0.BYTE = (unsigned char)(adrs & 0x00ff); // data set
P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait lbyte transmitted
if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    return (0); // no ACK
}
return (1); // ACK OK
}
```



## 3. 付録

### 3.1 SH7145F レジスタ定義ファイル

以下に SH7145F のレジスタ定義ファイルを示します。

```
/* ***** */
/* FILE      :iodefine.h                               */
/* DATE      :Tue, Oct 02, 2001                       */
/* DESCRIPTION:Definition of I/O Register              */
/* CPU TYPE   :SH7145F                                */
/* This file is generated by Hitachi Project Generator (Ver.1.2). */
/* ***** */

/* ***** */
/* 7145 Include File                                  */
/* ***** */
struct st_sci0 {
    union {
        unsigned char BYTE;
        struct {
            unsigned char CA:1;
            unsigned char CHR:1;
            unsigned char PE:1;
            unsigned char OE:1;
            unsigned char STOP:1;
            unsigned char MP:1;
            unsigned char CKS:2;
        } BIT;
    } SMR_0;
    unsigned char BRR_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TIE:1;
            unsigned char RIE:1;
            unsigned char TE:1;
            unsigned char RE:1;
            unsigned char MPIE:1;
            unsigned char TEIE:1;
            unsigned char CKE:2;
        } BIT;
    } SCR_0;
    unsigned char TDR_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TDRE:1;
            unsigned char RDRF:1;
            unsigned char ORER:1;
            unsigned char FER:1;
            unsigned char PER:1;
            unsigned char TEND:1;
            unsigned char MPB:1;
            unsigned char MPBT:1;
        } BIT;
    } SSR_0;
    unsigned char RDR_0;
    union {
```

図 3.1 SH7145F レジスタ定義

### 3. 付録

```

    unsigned char BYTE;
    struct {
        unsigned char :4;
        unsigned char DIR:1;
        unsigned char :3;
    } BIT;
    } SDCR_0;
};
struct st_scil {
    union {
        unsigned char BYTE;
        struct {
            unsigned char CA:1;
            unsigned char CHR:1;
            unsigned char PE:1;
            unsigned char OE:1;
            unsigned char STOP:1;
            unsigned char MP:1;
            unsigned char CKS:2;
        } BIT;
    } SMR_1;
    unsigned char BRR_1;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TIE:1;
            unsigned char RIE:1;
            unsigned char TE:1;
            unsigned char RE:1;
            unsigned char MPE:1;
            unsigned char TEIE:1;
            unsigned char CKE:2;
        } BIT;
    } SCR_1;
    unsigned char TDR_1;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TDRE:1;
            unsigned char RDRF:1;
            unsigned char ORER:1;
            unsigned char FER:1;
            unsigned char PER:1;
            unsigned char TEND:1;
            unsigned char MPB:1;
            unsigned char MPBT:1;
        } BIT;
    } SSR_1;
    unsigned char RDR_1;
    union {
        unsigned char BYTE;
        struct {
            unsigned char :4;
            unsigned char DIR:1;
            unsigned char :3;
        } BIT;
    } SDCR_1;
};
struct st_sci2 {
    union {
        unsigned char BYTE;
        struct {
            unsigned char CA:1;
            unsigned char CHR:1;
            unsigned char PE:1;
            unsigned char OE:1;
            unsigned char STOP:1;
            unsigned char MP:1;
        } BIT;
    } SMR_2;
    unsigned char BRR_2;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TIE:1;
            unsigned char RIE:1;
            unsigned char TE:1;
            unsigned char RE:1;
            unsigned char MPE:1;
            unsigned char TEIE:1;
            unsigned char CKE:2;
        } BIT;
    } SCR_2;
    unsigned char TDR_2;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TDRE:1;
            unsigned char RDRF:1;
            unsigned char ORER:1;
            unsigned char FER:1;
            unsigned char PER:1;
            unsigned char TEND:1;
            unsigned char MPB:1;
            unsigned char MPBT:1;
        } BIT;
    } SSR_2;
    unsigned char RDR_2;
    union {
        unsigned char BYTE;
        struct {
            unsigned char :4;
            unsigned char DIR:1;
            unsigned char :3;
        } BIT;
    } SDCR_2;
};

```

図 3.1 SH7145F レジスタ定義 (続き)

```

        unsigned char CKS:2;          /* CKS */
    } BIT;                            /* */
} SMR_2;                              /* */
unsigned char BRR_2;                  /* BRR_2 */
union {                                /* SCR_2 */
    unsigned char BYTE;               /* Byte Access */
    struct {                           /* Bit Access */
        unsigned char TIE:1;          /* TIE */
        unsigned char RIE:1;          /* RIE */
        unsigned char TE:1;           /* TE */
        unsigned char RE:1;           /* RE */
        unsigned char MPIE:1;         /* MPIE */
        unsigned char TEIE:1;         /* TEIE */
        unsigned char CKE:2;          /* CKE */
    } BIT;                             /* */
} SCR_2;                              /* */
unsigned char TDR_2;                  /* TDR_2 */
union {                                /* SSR_2 */
    unsigned char BYTE;               /* Byte Access */
    struct {                           /* Bit Access */
        unsigned char TDRE:1;         /* TDRE */
        unsigned char RDRF:1;         /* RDRF */
        unsigned char ORER:1;         /* ORER */
        unsigned char FER:1;          /* FER */
        unsigned char PER:1;          /* PER */
        unsigned char TEND:1;         /* TEND */
        unsigned char MPB:1;          /* MPB */
        unsigned char MPBT:1;         /* MPBT */
    } BIT;                             /* */
} SSR_2;                              /* */
unsigned char RDR_2;                  /* RDR_2 */
union {                                /* SDCR_2 */
    unsigned char BYTE;               /* Byte Access */
    struct {                           /* Bit Access */
        unsigned char :4;              /* */
        unsigned char DIR:1;          /* DIR */
        unsigned char :3;              /* */
    } BIT;                             /* */
} SDCR_2;                              /* */
};
struct st_sci3 {
    union {
        unsigned char BYTE;           /* Byte Access */
        struct {                       /* Bit Access */
            unsigned char CA:1;        /* C/A */
            unsigned char CHR:1;       /* CHR */
            unsigned char PE:1;        /* PE */
            unsigned char OE:1;        /* O/E */
            unsigned char STOP:1;      /* STOP */
            unsigned char MP:1;        /* MP */
            unsigned char CKS:2;       /* CKS */
        } BIT;                         /* */
    } SMR_3;                            /* */
    unsigned char BRR_3;                /* BRR_3 */
    union {
        unsigned char BYTE;           /* Byte Access */
        struct {                       /* Bit Access */
            unsigned char TIE:1;       /* TIE */
            unsigned char RIE:1;       /* RIE */
            unsigned char TE:1;        /* TE */
            unsigned char RE:1;        /* RE */
            unsigned char MPIE:1;      /* MPIE */
            unsigned char TEIE:1;      /* TEIE */
            unsigned char CKE:2;       /* CKE */
        } BIT;                         /* */
    } SCR_3;                            /* */
    unsigned char TDR_3;                /* TDR_3 */
    union {
        unsigned char :4;              /* */
        unsigned char DIR:1;          /* DIR */
        unsigned char :3;              /* */
    } BIT;                             /* */
} SDCR_3;                              /* */
};

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

    unsigned char BYTE;
    struct {
        unsigned char TDRE:1;
        unsigned char RDRF:1;
        unsigned char ORER:1;
        unsigned char FER:1;
        unsigned char PER:1;
        unsigned char TEND:1;
        unsigned char MPB:1;
        unsigned char MPBT:1;
    } BIT;
} SSR_3;
unsigned char RDR_3;
union {
    unsigned char BYTE;
    struct {
        unsigned char :4;
        unsigned char DIR:1;
        unsigned char :3;
    } BIT;
} SDCR_3;
};
struct st_mtu34 {
    union {
        unsigned char BYTE;
        struct {
            unsigned char CCLR:3;
            unsigned char CKEG:2;
            unsigned char TPSC:3;
        } BIT;
    } TCR_3;
    union {
        unsigned char BYTE;
        struct {
            unsigned char CCLR:3;
            unsigned char CKEG:2;
            unsigned char TPSC:3;
        } BIT;
    } TCR_4;
    union {
        unsigned char BYTE;
        struct {
            unsigned char :2;
            unsigned char BFB:1;
            unsigned char BFA:1;
            unsigned char MD:4;
        } BIT;
    } TMDR_3;
    union {
        unsigned char BYTE;
        struct {
            unsigned char :2;
            unsigned char BFB:1;
            unsigned char BFA:1;
            unsigned char MD:4;
        } BIT;
    } TMDR_4;
    union {
        unsigned char BYTE;
        struct {
            unsigned char IOB:4;
            unsigned char IOA:4;
        } BIT;
    } TIORH_3;
    union {
        unsigned char BYTE;
        struct {
            unsigned char IOD:4;

```

図 3.1 SH7145F レジスタ定義 (続き)

```

        unsigned char IOC:4;          /* IOC          */
        } BIT;                        /*              */
    } TIORL_3;                          /*              */
union {                                /* TIORH_4      */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char IOB:4;          /* IOB          */
        unsigned char IOA:4;          /* IOA          */
        } BIT;                        /*              */
    } TIORH_4;                          /*              */
union {                                /* TIORL_4      */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char IOD:4;          /* IOD          */
        unsigned char IOC:4;          /* IOC          */
        } BIT;                        /*              */
    } TIORL_4;                          /*              */
union {                                /* TIER_3       */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char TTGE:1;         /* TTGE         */
        unsigned char :2;             /*              */
        unsigned char TCIEV:1;        /* TCIEV        */
        unsigned char TGIED:1;        /* TGIED        */
        unsigned char TGIEC:1;        /* TGIEC        */
        unsigned char TGIEB:1;        /* TGIEB        */
        unsigned char TGIEA:1;        /* TGIEA        */
        } BIT;                        /*              */
    } TIER_3;                          /*              */
union {                                /* TIER_4       */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char TTGE:1;         /* TTGE         */
        unsigned char :2;             /*              */
        unsigned char TCIEV:1;        /* TCIEV        */
        unsigned char TGIED:1;        /* TGIED        */
        unsigned char TGIEC:1;        /* TGIEC        */
        unsigned char TGIEB:1;        /* TGIEB        */
        unsigned char TGIEA:1;        /* TGIEA        */
        } BIT;                        /*              */
    } TIER_4;                          /*              */
union {                                /* TOER         */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char :2;             /*              */
        unsigned char OE4D:1;         /* OE4D         */
        unsigned char OE4C:1;         /* OE4C         */
        unsigned char OE3D:1;         /* OE3D         */
        unsigned char OE4B:1;         /* OE4B         */
        unsigned char OE4A:1;         /* OE4A         */
        unsigned char OE3B:1;         /* OE3B         */
        } BIT;                        /*              */
    } TOER;                            /*              */
union {                                /* TOCR         */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char :1;             /*              */
        unsigned char PSYE:1;         /* PSYE         */
        unsigned char :4;             /*              */
        unsigned char OLSN:1;         /* OLSN         */
        unsigned char OLSP:1;         /* OLSP         */
        } BIT;                        /*              */
    } TOCR;                            /*              */
unsigned char wk0[1];                  /*              */
union {                                /* TGCR         */
    unsigned char BYTE;                /* Byte Access */
    struct {                            /* Bit Access  */
        unsigned char :1;             /*              */
    }

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned char BDC:1;          /* BDC */
        unsigned char N:1;           /* N */
        unsigned char P:1;           /* P */
        unsigned char FB:1;          /* FB */
        unsigned char WF:1;          /* WF */
        unsigned char VF:1;          /* VF */
        unsigned char UF:1;          /* UF */
        } BIT;                       /* */
    } TGCR;                           /* */
    unsigned char wk1[2];             /* */
    unsigned short TCNT_3;           /* TCNT_3 */
    unsigned short TCNT_4;           /* TCNT_4 */
    unsigned short TCDR;             /* TCDR */
    unsigned short TDDR;             /* TDDR */
    unsigned short TGRA_3;           /* TGRA_3 */
    unsigned short TGRB_3;           /* TGRB_3 */
    unsigned short TGRA_4;           /* TGRA_4 */
    unsigned short TGRB_4;           /* TGRB_4 */
    unsigned short TCNTS;            /* TCNTS */
    unsigned short TCBR;             /* TCBR */
    unsigned short TGRC_3;           /* TGRC_3 */
    unsigned short TGRD_3;           /* TGRD_3 */
    unsigned short TGRC_4;           /* TGRC_4 */
    unsigned short TGRD_4;           /* TGRD_4 */
    union {                           /* TSR_3 */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char TCFD:1;    /* TCFD */
            unsigned char :2;        /* */
            unsigned char TCFV:1;    /* TCFV */
            unsigned char TGFD:1;    /* TGFD */
            unsigned char TGFC:1;    /* TGFC */
            unsigned char TGFB:1;    /* TGFB */
            unsigned char TGFA:1;    /* TGFA */
        } BIT;                       /* */
    } TSR_3;                           /* */
    union {                           /* TSR_4 */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char TCFD:1;    /* TCFD */
            unsigned char :2;        /* */
            unsigned char TCFV:1;    /* TCFV */
            unsigned char TGFD:1;    /* TGFD */
            unsigned char TGFC:1;    /* TGFC */
            unsigned char TGFB:1;    /* TGFB */
            unsigned char TGFA:1;    /* TGFA */
        } BIT;                       /* */
    } TSR_4;                           /* */
    unsigned char wk2[18];           /* */
    union {                           /* TSTR */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char CST4:1;    /* CST4 */
            unsigned char CST3:1;    /* CST3 */
            unsigned char :3;        /* */
            unsigned char CST:3;     /* CST */
        } BIT;                       /* */
    } TSTR;                           /* */
    union {                           /* TSYR */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char SYNC4:1;   /* SYNC4 */
            unsigned char SYNC3:1;   /* SYNC3 */
            unsigned char :3;        /* */
            unsigned char SYNC:3;    /* SYNC */
        } BIT;                       /* */
    } TSYR;                           /* */
};                                     /* */

```

図 3.1 SH7145F レジスタ定義 (続き)



```

struct st_mtu0 {
    union {
        unsigned char BYTE;
        struct {
            unsigned char CCLR:3;
            unsigned char CKEG:2;
            unsigned char TPSC:3;
        } BIT;
    } TCR_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char :2;
            unsigned char BFB:1;
            unsigned char BFA:1;
            unsigned char MD:4;
        } BIT;
    } TMDR_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char IOB:4;
            unsigned char IOA:4;
        } BIT;
    } TIORH_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char IOD:4;
            unsigned char IOC:4;
        } BIT;
    } TIORL_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TTGE:1;
            unsigned char :2;
            unsigned char TCIEV:1;
            unsigned char TGIED:1;
            unsigned char TGIEC:1;
            unsigned char TGIEB:1;
            unsigned char TGIEA:1;
        } BIT;
    } TIER_0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char :3;
            unsigned char TCFV:1;
            unsigned char TGFD:1;
            unsigned char TGFC:1;
            unsigned char TGFB:1;
            unsigned char TGFA:1;
        } BIT;
    } TSR_0;
    unsigned short TCNT_0;
    unsigned short TGRA_0;
    unsigned short TGRB_0;
    unsigned short TGRC_0;
    unsigned short TGRD_0;
};
struct st_mtu1 {
    union {
        unsigned char BYTE;
        struct {
            unsigned char :1;
            unsigned char CCLR:2;
            unsigned char CKEG:2;
        } BIT;
    } TCR_1;
};

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned char TPSC:3;          /* TPSC */
    } BIT;                             /* */
} TCR_1;                               /* */
union {                                /* TMDR_1 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char :4;             /* */
        unsigned char MD:4;          /* MD */
    } BIT;                             /* */
} TMDR_1;                              /* */
union {                                /* TIOR_1 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char IOB:4;         /* IOB */
        unsigned char IOA:4;         /* IOA */
    } BIT;                             /* */
} TIOR_1;                              /* */
unsigned char wk0[1];                 /* */
union {                                /* TIER_1 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char TTGE:1;        /* TTGE */
        unsigned char :1;            /* */
        unsigned char TCIEU:1;       /* TCIEU */
        unsigned char TCIEV:1;       /* TCIEV */
        unsigned char :2;            /* */
        unsigned char TGIEB:1;       /* TGIEB */
        unsigned char TGIEA:1;       /* TGIEA */
    } BIT;                             /* */
} TIER_1;                              /* */
union {                                /* TSR_1 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char TCFD:1;        /* TCFD */
        unsigned char :1;            /* */
        unsigned char TCFU:1;        /* TCFU */
        unsigned char TCFV:1;        /* TCFV */
        unsigned char :2;            /* */
        unsigned char TGFB:1;        /* TGFB */
        unsigned char TGFA:1;        /* TGFA */
    } BIT;                             /* */
} TSR_1;                              /* */
unsigned short TCNT_1;                 /* TCNT_1 */
unsigned short TGRA_1;                 /* TGRA_1 */
unsigned short TGRB_1;                 /* TGRB_1 */
};
struct st_mtu2 {                       /* struct MTU2 */
    union {                            /* TCR_2 */
        unsigned char BYTE;          /* Byte Access */
        struct {                    /* Bit Access */
            unsigned char :1;        /* */
            unsigned char CCLR:2;    /* CCLR */
            unsigned char CKEG:2;    /* CKEG */
            unsigned char TPSC:3;    /* TPSC */
        } BIT;                       /* */
    } TCR_2;                          /* */
    union {                            /* TMDR_2 */
        unsigned char BYTE;          /* Byte Access */
        struct {                    /* Bit Access */
            unsigned char :4;        /* */
            unsigned char MD:4;      /* MD */
        } BIT;                       /* */
    } TMDR_2;                          /* */
    union {                            /* TIOR_2 */
        unsigned char BYTE;          /* Byte Access */
        struct {                    /* Bit Access */
            unsigned char IOB:4;     /* IOB */
            unsigned char IOA:4;     /* IOA */
        } BIT;                       /* */
    } TIOR_2;                          /* */
};

```

図 3.1 SH7145F レジスタ定義 (続き)

```

        } BIT;
    } TIOR_2;
    unsigned char wk0[1];
    union {
        unsigned char BYTE;
        struct {
            unsigned char TTGE:1;
            unsigned char :1;
            unsigned char TCIEU:1;
            unsigned char TCIEV:1;
            unsigned char :2;
            unsigned char TGIEB:1;
            unsigned char TGIEA:1;
        } BIT;
    } TIER_2;
    union {
        unsigned char BYTE;
        struct {
            unsigned char TCFD:1;
            unsigned char :1;
            unsigned char TCFU:1;
            unsigned char TCFV:1;
            unsigned char :2;
            unsigned char TGFB:1;
            unsigned char TGFA:1;
        } BIT;
    } TSR_2;
    unsigned short TCNT_2;
    unsigned short TGRA_2;
    unsigned short TGRB_2;
};
struct st_intc {
    union {
        unsigned short WORD;
        struct {
            unsigned short IRQ0:4;
            unsigned short IRQ1:4;
            unsigned short IRQ2:4;
            unsigned short IRQ3:4;
        } BIT;
    } IPRA;
    union {
        unsigned short WORD;
        struct {
            unsigned short IRQ4:4;
            unsigned short IRQ5:4;
            unsigned short IRQ6:4;
            unsigned short IRQ7:4;
        } BIT;
    } IPRB;
    union {
        unsigned short WORD;
        struct {
            unsigned short DMAC0:4;
            unsigned short DMAC1:4;
            unsigned short DMAC2:4;
            unsigned short DMAC3:4;
        } BIT;
    } IPRC;
    union {
        unsigned short WORD;
        struct {
            unsigned short MTU0:8;
            unsigned short MTU1:8;
        } BIT;
    } IPRD;
    union {
        unsigned short WORD;
        struct {
            unsigned short :8;
        } BIT;
    } IPRE;
};

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

    struct {
        unsigned short MTU2:8; /* Bit Access */
        unsigned short MTU3:8; /* MTU2 */
    } BIT; /* MTU3 */
} IPRE; /* */
union { /* IPRF */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short MTU4:8; /* MTU4 */
        unsigned short SCI0:4; /* SCI0 */
        unsigned short SCI1:4; /* SCI1 */
    } BIT; /* */
} IPRF; /* */
union { /* IPRG */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short AD01:4; /* A/D0,1 */
        unsigned short DTC:4; /* DTC */
        unsigned short CMT0:4; /* CMT0 */
        unsigned short CMT1:4; /* CMT1 */
    } BIT; /* */
} IPRG; /* */
union { /* IPRH */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short WDT:4; /* WDT */
        unsigned short IOMTU:4; /* I/O(MTU) */
        unsigned short :8; /* */
    } BIT; /* */
} IPRH; /* */
union { /* ICR1 */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short NMIL:1; /* NMIL */
        unsigned short :6; /* */
        unsigned short NMIE:1; /* NMIE */
        unsigned short IRQ0S:1; /* IRQ0S */
        unsigned short IRQ1S:1; /* IRQ1S */
        unsigned short IRQ2S:1; /* IRQ2S */
        unsigned short IRQ3S:1; /* IRQ3S */
        unsigned short IRQ4S:1; /* IRQ4S */
        unsigned short IRQ5S:1; /* IRQ5S */
        unsigned short IRQ6S:1; /* IRQ6S */
        unsigned short IRQ7S:1; /* IRQ7S */
    } BIT; /* */
} ICR1; /* */
union { /* ISR */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short :8; /* */
        unsigned short IRQ0F:1; /* IRQ0F */
        unsigned short IRQ1F:1; /* IRQ1F */
        unsigned short IRQ2F:1; /* IRQ2F */
        unsigned short IRQ3F:1; /* IRQ3F */
        unsigned short IRQ4F:1; /* IRQ4F */
        unsigned short IRQ5F:1; /* IRQ5F */
        unsigned short IRQ6F:1; /* IRQ6F */
        unsigned short IRQ7F:1; /* IRQ7F */
    } BIT; /* */
} ISR; /* */
union { /* IPRI */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short SCI2:4; /* SCI2 */
        unsigned short SCI3:4; /* SCI3 */
        unsigned short :8; /* */
    } BIT; /* */
} IPRI; /* */

```

図 3.1 SH7145F レジスタ定義 (続き)

```

union {
    unsigned short WORD; /* IPRJ */
    struct { /* Word Access */
        unsigned short :8; /* Bit Access */
        unsigned short IIC:4; /* IIC */
        unsigned short :4; /* */
    } BIT; /* */
    } IPRJ; /* */
unsigned char wk0[6]; /* */
union { /* ICR2 */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short IRQ0ES:2; /* IRQ0ES */
        unsigned short IRQ1ES:2; /* IRQ1ES */
        unsigned short IRQ2ES:2; /* IRQ2ES */
        unsigned short IRQ3ES:2; /* IRQ3ES */
        unsigned short IRQ4ES:2; /* IRQ4ES */
        unsigned short IRQ5ES:2; /* IRQ5ES */
        unsigned short IRQ6ES:2; /* IRQ6ES */
        unsigned short IRQ7ES:2; /* IRQ7ES */
    } BIT; /* */
    } ICR2; /* */
}; /* */
struct st_porta { /* struct PORTA */
    union { /* PADRH */
        unsigned short WORD; /* Word Access */
        struct { /* Bit Access */
            unsigned short :8; /* */
            unsigned short PA23DR:1; /* PA23DR */
            unsigned short PA22DR:1; /* PA22DR */
            unsigned short PA21DR:1; /* PA21DR */
            unsigned short PA20DR:1; /* PA20DR */
            unsigned short PA19DR:1; /* PA19DR */
            unsigned short PA18DR:1; /* PA18DR */
            unsigned short PA17DR:1; /* PA17DR */
            unsigned short PA16DR:1; /* PA16DR */
        } BIT; /* */
        } PADRH; /* */
    union { /* PADRL */
        unsigned short WORD; /* Word Access */
        struct { /* Bit Access */
            unsigned short PA15DR:1; /* PA15DR */
            unsigned short PA14DR:1; /* PA14DR */
            unsigned short PA13DR:1; /* PA13DR */
            unsigned short PA12DR:1; /* PA12DR */
            unsigned short PA11DR:1; /* PA11DR */
            unsigned short PA10DR:1; /* PA10DR */
            unsigned short PA9DR:1; /* PA9DR */
            unsigned short PA8DR:1; /* PA8DR */
            unsigned short PA7DR:1; /* PA7DR */
            unsigned short PA6DR:1; /* PA6DR */
            unsigned short PA5DR:1; /* PA5DR */
            unsigned short PA4DR:1; /* PA4DR */
            unsigned short PA3DR:1; /* PA3DR */
            unsigned short PA2DR:1; /* PA2DR */
            unsigned short PA1DR:1; /* PA1DR */
            unsigned short PA0DR:1; /* PA0DR */
        } BIT; /* */
        } PADRL; /* */
    union { /* PAIORH */
        unsigned short WORD; /* Word Access */
        struct { /* Bit Access */
            unsigned short :8; /* */
            unsigned short PA23IOR:1; /* PA23IOR */
            unsigned short PA22IOR:1; /* PA22IOR */
            unsigned short PA21IOR:1; /* PA21IOR */
            unsigned short PA20IOR:1; /* PA20IOR */
            unsigned short PA19IOR:1; /* PA19IOR */
        } BIT; /* */
        } PAIORH; /* */
};

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned short PA18IOR:1;      /* PA18IOR */
        unsigned short PA17IOR:1;      /* PA17IOR */
        unsigned short PA16IOR:1;      /* PA16IOR */
    } BIT;
    /* */
} PAIORH;
/* */
union {
    /* PAIORL */
    unsigned short WORD;               /* Word Access */
    struct {
        /* Bit Access */
        unsigned short PA15IOR:1;      /* PA15IOR */
        unsigned short PA14IOR:1;      /* PA14IOR */
        unsigned short PA13IOR:1;      /* PA13IOR */
        unsigned short PA12IOR:1;      /* PA12IOR */
        unsigned short PA11IOR:1;      /* PA11IOR */
        unsigned short PA10IOR:1;      /* PA10IOR */
        unsigned short PA9IOR:1;       /* PA9IOR */
        unsigned short PA8IOR:1;       /* PA8IOR */
        unsigned short PA7IOR:1;       /* PA7IOR */
        unsigned short PA6IOR:1;       /* PA6IOR */
        unsigned short PA5IOR:1;       /* PA5IOR */
        unsigned short PA4IOR:1;       /* PA4IOR */
        unsigned short PA3IOR:1;       /* PA3IOR */
        unsigned short PA2IOR:1;       /* PA2IOR */
        unsigned short PA1IOR:1;       /* PA1IOR */
        unsigned short PA0IOR:1;       /* PA0IOR */
    } BIT;
    /* */
} PAIORL;
/* */
union {
    /* PACRH */
    unsigned short WORD;               /* Word Access */
    struct {
        /* Bit Access */
        unsigned short :1;             /* */
        unsigned short PA23MD:1;       /* PA23MD */
        unsigned short :1;             /* */
        unsigned short PA22MD:1;       /* PA22MD */
        unsigned short :1;             /* */
        unsigned short PA21MD:1;       /* PA21MD */
        unsigned short :1;             /* */
        unsigned short PA20MD:1;       /* PA20MD */
        unsigned short PA19MD:2;       /* PA19MD */
        unsigned short PA18MD:2;       /* PA18MD */
        unsigned short PA17MD:2;       /* PA17MD */
        unsigned short PA16MD:2;       /* PA16MD */
    } BIT;
    /* */
} PACRH;
/* */
unsigned char wk0[2];
union {
    /* PACRL1 */
    unsigned short WORD;               /* Word Access */
    struct {
        /* Bit Access */
        unsigned short PA15MD:2;       /* PA15MD */
        unsigned short PA14MD:2;       /* PA14MD */
        unsigned short PA13MD:2;       /* PA13MD */
        unsigned short PA12MD:2;       /* PA12MD */
        unsigned short PA11MD:2;       /* PA11MD */
        unsigned short PA10MD:2;       /* PA10MD */
        unsigned short PA9MD:2;        /* PA9MD */
        unsigned short PA8MD:2;        /* PA8MD */
    } BIT;
    /* */
} PACRL1;
/* */
union {
    /* PACRL2 */
    unsigned short WORD;               /* Word Access */
    struct {
        /* Bit Access */
        unsigned short PA7MD:2;        /* PA7MD */
        unsigned short PA6MD:2;        /* PA6MD */
        unsigned short PA5MD:2;        /* PA5MD */
        unsigned short PA4MD:2;        /* PA4MD */
        unsigned short PA3MD:2;        /* PA3MD */
        unsigned short PA2MD:2;        /* PA2MD */
        unsigned short PA1MD:2;        /* PA1MD */
        unsigned short PA0MD:2;        /* PA0MD */
    }

```

図 3.1 SH7145F レジスタ定義 (続き)

```

        } BIT;
    } PACRL2;
};
struct st_portb {
    union {
        unsigned short WORD;
        struct {
            unsigned short :6;
            unsigned short PB9DR:1;
            unsigned short PB8DR:1;
            unsigned short PB7DR:1;
            unsigned short PB6DR:1;
            unsigned short PB5DR:1;
            unsigned short PB4DR:1;
            unsigned short PB3DR:1;
            unsigned short PB2DR:1;
            unsigned short PB1DR:1;
            unsigned short PB0DR:1;
        } BIT;
    } PBDR;
    unsigned char wk0[2];
    union {
        unsigned short WORD;
        struct {
            unsigned short :6;
            unsigned short PB9IOR:1;
            unsigned short PB8IOR:1;
            unsigned short PB7IOR:1;
            unsigned short PB6IOR:1;
            unsigned short PB5IOR:1;
            unsigned short PB4IOR:1;
            unsigned short PB3IOR:1;
            unsigned short PB2IOR:1;
            unsigned short PB1IOR:1;
            unsigned short PB0IOR:1;
        } BIT;
    } PBIOR;
    unsigned char wk1[2];
    union {
        unsigned short WORD;
        struct {
            unsigned short :4;
            unsigned short PB3MD2:1;
            unsigned short PB2MD2:1;
            unsigned short :6;
            unsigned short PB9MD:2;
            unsigned short PB8MD:2;
        } BIT;
    } PBCR1;
    union {
        unsigned short WORD;
        struct {
            unsigned short PB7MD:2;
            unsigned short PB6MD:2;
            unsigned short PB5MD:2;
            unsigned short PB4MD:2;
            unsigned short PB3MD:2;
            unsigned short PB2MD:2;
            unsigned short PB1MD:2;
            unsigned short PB0MD:2;
        } BIT;
    } PBCR2;
};
struct st_portc {
    union {
        unsigned short WORD;
        struct {
            unsigned short PC15DR:1;

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned short PC14DR:1;      /* PC14DR */
        unsigned short PC13DR:1;      /* PC13DR */
        unsigned short PC12DR:1;      /* PC12DR */
        unsigned short PC11DR:1;      /* PC11DR */
        unsigned short PC10DR:1;      /* PC10DR */
        unsigned short PC9DR:1;       /* PC9DR */
        unsigned short PC8DR:1;       /* PC8DR */
        unsigned short PC7DR:1;       /* PC7DR */
        unsigned short PC6DR:1;       /* PC6DR */
        unsigned short PC5DR:1;       /* PC5DR */
        unsigned short PC4DR:1;       /* PC4DR */
        unsigned short PC3DR:1;       /* PC3DR */
        unsigned short PC2DR:1;       /* PC2DR */
        unsigned short PC1DR:1;       /* PC1DR */
        unsigned short PC0DR:1;       /* PC0DR */
    } BIT;
} PCDR;
unsigned char wk0[2];
union {
    unsigned short WORD;
    struct {
        unsigned short PC15IOR:1;     /* PC15IOR */
        unsigned short PC14IOR:1;     /* PC14IOR */
        unsigned short PC13IOR:1;     /* PC13IOR */
        unsigned short PC12IOR:1;     /* PC12IOR */
        unsigned short PC11IOR:1;     /* PC11IOR */
        unsigned short PC10IOR:1;     /* PC10IOR */
        unsigned short PC9IOR:1;      /* PC9IOR */
        unsigned short PC8IOR:1;      /* PC8IOR */
        unsigned short PC7IOR:1;      /* PC7IOR */
        unsigned short PC6IOR:1;      /* PC6IOR */
        unsigned short PC5IOR:1;      /* PC5IOR */
        unsigned short PC4IOR:1;      /* PC4IOR */
        unsigned short PC3IOR:1;      /* PC3IOR */
        unsigned short PC2IOR:1;      /* PC2IOR */
        unsigned short PC1IOR:1;      /* PC1IOR */
        unsigned short PC0IOR:1;      /* PC0IOR */
    } BIT;
} PCIOR;
unsigned char wk1[4];
union {
    unsigned short WORD;
    struct {
        unsigned short PC15MD:1;      /* PC15MD */
        unsigned short PC14MD:1;      /* PC14MD */
        unsigned short PC13MD:1;      /* PC13MD */
        unsigned short PC12MD:1;      /* PC12MD */
        unsigned short PC11MD:1;      /* PC11MD */
        unsigned short PC10MD:1;      /* PC10MD */
        unsigned short PC9MD:1;       /* PC9MD */
        unsigned short PC8MD:1;       /* PC8MD */
        unsigned short PC7MD:1;       /* PC7MD */
        unsigned short PC6MD:1;       /* PC6MD */
        unsigned short PC5MD:1;       /* PC5MD */
        unsigned short PC4MD:1;       /* PC4MD */
        unsigned short PC3MD:1;       /* PC3MD */
        unsigned short PC2MD:1;       /* PC2MD */
        unsigned short PC1MD:1;       /* PC1MD */
        unsigned short PC0MD:1;       /* PC0MD */
    } BIT;
} PCCR;
};
struct st_portd {
    union {
        unsigned short WORD;
        struct {
            unsigned short PD31DR:1;  /* PD31DR */
            unsigned short PD30DR:1;  /* PD30DR */

```

図 3.1 SH7145F レジスタ定義 (続き)



```

        unsigned short PD29DR:1;      /* PD29DR */
        unsigned short PD28DR:1;      /* PD28DR */
        unsigned short PD27DR:1;      /* PD27DR */
        unsigned short PD26DR:1;      /* PD26DR */
        unsigned short PD25DR:1;      /* PD25DR */
        unsigned short PD24DR:1;      /* PD24DR */
        unsigned short PD23DR:1;      /* PD23DR */
        unsigned short PD22DR:1;      /* PD22DR */
        unsigned short PD21DR:1;      /* PD21DR */
        unsigned short PD20DR:1;      /* PD20DR */
        unsigned short PD19DR:1;      /* PD19DR */
        unsigned short PD18DR:1;      /* PD18DR */
        unsigned short PD17DR:1;      /* PD17DR */
        unsigned short PD16DR:1;      /* PD16DR */
    } BIT;
} PDDRH;
union {
    unsigned short WORD;
    struct {
        unsigned short PD15DR:1;      /* PD15DR */
        unsigned short PD14DR:1;      /* PD14DR */
        unsigned short PD13DR:1;      /* PD13DR */
        unsigned short PD12DR:1;      /* PD12DR */
        unsigned short PD11DR:1;      /* PD11DR */
        unsigned short PD10DR:1;      /* PD10DR */
        unsigned short PD9DR:1;       /* PD9DR */
        unsigned short PD8DR:1;       /* PD8DR */
        unsigned short PD7DR:1;       /* PD7DR */
        unsigned short PD6DR:1;       /* PD6DR */
        unsigned short PD5DR:1;       /* PD5DR */
        unsigned short PD4DR:1;       /* PD4DR */
        unsigned short PD3DR:1;       /* PD3DR */
        unsigned short PD2DR:1;       /* PD2DR */
        unsigned short PD1DR:1;       /* PD1DR */
        unsigned short PD0DR:1;       /* PD0DR */
    } BIT;
} PDDRL;
union {
    unsigned short WORD;
    struct {
        unsigned short PD31IOR:1;     /* PD31IOR */
        unsigned short PD30IOR:1;     /* PD30IOR */
        unsigned short PD29IOR:1;     /* PD29IOR */
        unsigned short PD28IOR:1;     /* PD28IOR */
        unsigned short PD27IOR:1;     /* PD27IOR */
        unsigned short PD26IOR:1;     /* PD26IOR */
        unsigned short PD25IOR:1;     /* PD25IOR */
        unsigned short PD24IOR:1;     /* PD24IOR */
        unsigned short PD23IOR:1;     /* PD23IOR */
        unsigned short PD22IOR:1;     /* PD22IOR */
        unsigned short PD21IOR:1;     /* PD21IOR */
        unsigned short PD20IOR:1;     /* PD20IOR */
        unsigned short PD19IOR:1;     /* PD19IOR */
        unsigned short PD18IOR:1;     /* PD18IOR */
        unsigned short PD17IOR:1;     /* PD17IOR */
        unsigned short PD16IOR:1;     /* PD16IOR */
    } BIT;
} PDIORH;
union {
    unsigned short WORD;
    struct {
        unsigned short PD15IOR:1;     /* PD15IOR */
        unsigned short PD14IOR:1;     /* PD14IOR */
        unsigned short PD13IOR:1;     /* PD13IOR */
        unsigned short PD12IOR:1;     /* PD12IOR */
        unsigned short PD11IOR:1;     /* PD11IOR */
        unsigned short PD10IOR:1;     /* PD10IOR */
        unsigned short PD9IOR:1;      /* PD9IOR */
    } BIT;
} PDIORL;

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned short PD8IOR:1;      /* PD8IOR */
        unsigned short PD7IOR:1;      /* PD7IOR */
        unsigned short PD6IOR:1;      /* PD6IOR */
        unsigned short PD5IOR:1;      /* PD5IOR */
        unsigned short PD4IOR:1;      /* PD4IOR */
        unsigned short PD3IOR:1;      /* PD3IOR */
        unsigned short PD2IOR:1;      /* PD2IOR */
        unsigned short PD1IOR:1;      /* PD1IOR */
        unsigned short PD0IOR:1;      /* PD0IOR */
    } BIT;
} PDIORL;
union {
    unsigned short WORD;
    struct {
        unsigned short PD31MD:2;      /* PD31MD */
        unsigned short PD30MD:2;      /* PD30MD */
        unsigned short PD29MD:2;      /* PD29MD */
        unsigned short PD28MD:2;      /* PD28MD */
        unsigned short PD27MD:2;      /* PD27MD */
        unsigned short PD26MD:2;      /* PD26MD */
        unsigned short PD25MD:2;      /* PD25MD */
        unsigned short PD24MD:2;      /* PD24MD */
    } BIT;
} PDCRH1;
union {
    unsigned short WORD;
    struct {
        unsigned short PD23MD:2;      /* PD23MD */
        unsigned short PD22MD:2;      /* PD22MD */
        unsigned short PD21MD:2;      /* PD21MD */
        unsigned short PD20MD:2;      /* PD20MD */
        unsigned short PD19MD:2;      /* PD19MD */
        unsigned short PD18MD:2;      /* PD18MD */
        unsigned short PD17MD:2;      /* PD17MD */
        unsigned short PD16MD:2;      /* PD16MD */
    } BIT;
} PDCRH2;
union {
    unsigned short WORD;
    struct {
        unsigned short PD15MD0:1;      /* PD15MD0 */
        unsigned short PD14MD0:1;      /* PD14MD0 */
        unsigned short PD13MD0:1;      /* PD13MD0 */
        unsigned short PD12MD0:1;      /* PD12MD0 */
        unsigned short PD11MD0:1;      /* PD11MD0 */
        unsigned short PD10MD0:1;      /* PD10MD0 */
        unsigned short PD9MD0:1;       /* PD9MD0 */
        unsigned short PD8MD0:1;       /* PD8MD0 */
        unsigned short PD7MD0:1;       /* PD7MD0 */
        unsigned short PD6MD0:1;       /* PD6MD0 */
        unsigned short PD5MD0:1;       /* PD5MD0 */
        unsigned short PD4MD0:1;       /* PD4MD0 */
        unsigned short PD3MD0:1;       /* PD3MD0 */
        unsigned short PD2MD0:1;       /* PD2MD0 */
        unsigned short PD1MD0:1;       /* PD1MD0 */
        unsigned short PD0MD0:1;       /* PD0MD0 */
    } BIT;
} PDCRL1;
union {
    unsigned short WORD;
    struct {
        unsigned short PD15MD1:1;      /* PD15MD1 */
        unsigned short PD14MD1:1;      /* PD14MD1 */
        unsigned short PD13MD1:1;      /* PD13MD1 */
        unsigned short PD12MD1:1;      /* PD12MD1 */
        unsigned short PD11MD1:1;      /* PD11MD1 */
        unsigned short PD10MD1:1;      /* PD10MD1 */
        unsigned short PD9MD1:1;       /* PD9MD1 */
    }

```

図 3.1 SH7145F レジスタ定義 (続き)

```

        unsigned short PD8MD1:1;      /* PD8MD1 */
        unsigned short PD7MD1:1;      /* PD7MD1 */
        unsigned short PD6MD1:1;      /* PD6MD1 */
        unsigned short PD5MD1:1;      /* PD5MD1 */
        unsigned short PD4MD1:1;      /* PD4MD1 */
        unsigned short PD3MD1:1;      /* PD3MD1 */
        unsigned short PD2MD1:1;      /* PD2MD1 */
        unsigned short PD1MD1:1;      /* PD1MD1 */
        unsigned short PD0MD1:1;      /* PD0MD1 */
    } BIT;
} PDCRL2;
};
struct st_porte {
    union {
        unsigned short WORD;
        struct {
            unsigned short PE15DR:1;   /* PE15DR */
            unsigned short PE14DR:1;   /* PE14DR */
            unsigned short PE13DR:1;   /* PE13DR */
            unsigned short PE12DR:1;   /* PE12DR */
            unsigned short PE11DR:1;   /* PE11DR */
            unsigned short PE10DR:1;   /* PE10DR */
            unsigned short PE9DR:1;    /* PE9DR */
            unsigned short PE8DR:1;    /* PE8DR */
            unsigned short PE7DR:1;    /* PE7DR */
            unsigned short PE6DR:1;    /* PE6DR */
            unsigned short PE5DR:1;    /* PE5DR */
            unsigned short PE4DR:1;    /* PE4DR */
            unsigned short PE3DR:1;    /* PE3DR */
            unsigned short PE2DR:1;    /* PE2DR */
            unsigned short PE1DR:1;    /* PE1DR */
            unsigned short PE0DR:1;    /* PE0DR */
        } BIT;
    } PEDRL;
    unsigned char wk0[2];
    union {
        unsigned short WORD;
        struct {
            unsigned short PE15IOR:1;  /* PE15IOR */
            unsigned short PE14IOR:1;  /* PE14IOR */
            unsigned short PE13IOR:1;  /* PE13IOR */
            unsigned short PE12IOR:1;  /* PE12IOR */
            unsigned short PE11IOR:1;  /* PE11IOR */
            unsigned short PE10IOR:1;  /* PE10IOR */
            unsigned short PE9IOR:1;   /* PE9IOR */
            unsigned short PE8IOR:1;   /* PE8IOR */
            unsigned short PE7IOR:1;   /* PE7IOR */
            unsigned short PE6IOR:1;   /* PE6IOR */
            unsigned short PE5IOR:1;   /* PE5IOR */
            unsigned short PE4IOR:1;   /* PE4IOR */
            unsigned short PE3IOR:1;   /* PE3IOR */
            unsigned short PE2IOR:1;   /* PE2IOR */
            unsigned short PE1IOR:1;   /* PE1IOR */
            unsigned short PE0IOR:1;   /* PE0IOR */
        } BIT;
    } PEIORL;
    unsigned char wk1[2];
    union {
        unsigned short WORD;
        struct {
            unsigned short PE15MD:2;   /* PE15MD */
            unsigned short PE14MD:2;   /* PE14MD */
            unsigned short PE13MD:2;   /* PE13MD */
            unsigned short PE12MD:2;   /* PE12MD */
            unsigned short PE11MD:2;   /* PE11MD */
            unsigned short PE10MD:2;   /* PE10MD */
            unsigned short PE9MD:2;    /* PE9MD */
            unsigned short PE8MD:2;    /* PE8MD */
        }
    }
};

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        } BIT;
    } PECRL1;
union {
    unsigned short WORD;
    struct {
        unsigned short PE7MD:2;
        unsigned short PE6MD:2;
        unsigned short PE5MD:2;
        unsigned short PE4MD:2;
        unsigned short PE3MD:2;
        unsigned short PE2MD:2;
        unsigned short PE1MD:2;
        unsigned short PE0MD:2;
    } BIT;
    } PECRL2;
};
struct st_portf {
    union {
        unsigned short WORD;
        struct {
            unsigned short :8;
            unsigned short PF7DR:1;
            unsigned short PF6DR:1;
            unsigned short PF5DR:1;
            unsigned short PF4DR:1;
            unsigned short PF3DR:1;
            unsigned short PF2DR:1;
            unsigned short PF1DR:1;
            unsigned short PF0DR:1;
        } BIT;
        } PFDR;
};
struct st_mtu {
    union {
        unsigned short WORD;
        struct {
            unsigned short POE3F:1;
            unsigned short POE2F:1;
            unsigned short POE1F:1;
            unsigned short POE0F:1;
            unsigned short :3;
            unsigned short PIE:1;
            unsigned short POE3M:2;
            unsigned short POE2M:2;
            unsigned short POE1M:2;
            unsigned short POE0M:2;
        } BIT;
        } ICSR1;
    union {
        unsigned short WORD;
        struct {
            unsigned short OSF:1;
            unsigned short :5;
            unsigned short OCE:1;
            unsigned short OIE:1;
            unsigned short :8;
        } BIT;
        } OCSR;
};
struct st_cmt {
    union {
        unsigned short WORD;
        struct {
            unsigned short :14;
            unsigned short STR:2;
        } BIT;
        } CMSTR;
    union {
        /* CMCSR_0 */

```

図 3.1 SH7145F レジスタ定義 (続き)

```

    unsigned short WORD;          /* Word Access */
    struct {                      /* Bit Access */
        unsigned short :8;       /*          */
        unsigned short CMF:1;    /* CMF      */
        unsigned short CMIE:1;   /* CMIE     */
        unsigned short :4;       /*          */
        unsigned short CKS:2;    /* CKS      */
    } BIT;                        /*          */
    } CMCSR_0;                    /*          */
    unsigned short CMCNT_0;       /* CMCNT_0  */
    unsigned short CMCOR_0;       /* CMCOR_0  */
    union {                       /* CMCSR_1  */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short :8;   /*          */
            unsigned short CMF:1; /* CMF      */
            unsigned short CMIE:1; /* CMIE     */
            unsigned short :4;   /*          */
            unsigned short CKS:2; /* CKS      */
        } BIT;                   /*          */
    } CMCSR_1;                    /*          */
    unsigned short CMCNT_1;       /* CMCNT_1  */
    unsigned short CMCOR_1;       /* CMCOR_1  */
};
struct st_ad {                   /* struct A/D */
    union {                       /* ADDR0    */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short AD:10; /* AD       */
            unsigned short :6;   /*          */
        } BIT;                   /*          */
    } ADDR0;                      /*          */
    union {                       /* ADDR1    */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short AD:10; /* AD       */
            unsigned short :6;   /*          */
        } BIT;                   /*          */
    } ADDR1;                      /*          */
    union {                       /* ADDR2    */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short AD:10; /* AD       */
            unsigned short :6;   /*          */
        } BIT;                   /*          */
    } ADDR2;                      /*          */
    union {                       /* ADDR3    */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short AD:10; /* AD       */
            unsigned short :6;   /*          */
        } BIT;                   /*          */
    } ADDR3;                      /*          */
    union {                       /* ADDR4    */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short AD:10; /* AD       */
            unsigned short :6;   /*          */
        } BIT;                   /*          */
    } ADDR4;                      /*          */
    union {                       /* ADDR5    */
        unsigned short WORD;     /* Word Access */
        struct {                 /* Bit Access */
            unsigned short AD:10; /* AD       */
            unsigned short :6;   /*          */
        } BIT;                   /*          */
    } ADDR5;                      /*          */
    union {                       /* ADDR6    */

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned short WORD;                /* Word Access */
        struct {                            /* Bit Access */
            unsigned short AD:10;           /* AD */
            unsigned short :6;              /* */
        } BIT;                               /* */
    } ADDR6;                                /* ADDR7 */
union {                                    /* ADDR7 */
    unsigned short WORD;                  /* Word Access */
    struct {                              /* Bit Access */
        unsigned short AD:10;           /* AD */
        unsigned short :6;              /* */
    } BIT;                               /* */
} ADDR7;
unsigned char wk0[80];                    /* */
union {                                    /* ADCSR_0 */
    unsigned char BYTE;                  /* Byte Access */
    struct {                              /* Bit Access */
        unsigned char ADF:1;           /* ADF */
        unsigned char ADIE:1;         /* ADIE */
        unsigned char :1;              /* */
        unsigned char ADM:1;          /* ADM */
        unsigned char :2;              /* */
        unsigned char CH:2;           /* CH */
    } BIT;                               /* */
} ADCSR_0;
union {                                    /* ADCSR_1 */
    unsigned char BYTE;                  /* Byte Access */
    struct {                              /* Bit Access */
        unsigned char ADF:1;           /* ADF */
        unsigned char ADIE:1;         /* ADIE */
        unsigned char :1;              /* */
        unsigned char ADM:1;          /* ADM */
        unsigned char :2;              /* */
        unsigned char CH:2;           /* CH */
    } BIT;                               /* */
} ADCSR_1;
unsigned char wk1[6];                    /* */
union {                                    /* ADCR_0 */
    unsigned char BYTE;                  /* Byte Access */
    struct {                              /* Bit Access */
        unsigned char TRGE:1;          /* TRGE */
        unsigned char CKS:2;           /* CKS */
        unsigned char ADST:1;         /* ADST */
        unsigned char ADCS:1;         /* ADCS */
        unsigned char :3;              /* */
    } BIT;                               /* */
} ADCR_0;
union {                                    /* ADCR_1 */
    unsigned char BYTE;                  /* Byte Access */
    struct {                              /* Bit Access */
        unsigned char TRGE:1;          /* TRGE */
        unsigned char CKS:2;           /* CKS */
        unsigned char ADST:1;         /* ADST */
        unsigned char ADCS:1;         /* ADCS */
        unsigned char :3;              /* */
    } BIT;                               /* */
} ADCR_1;
unsigned char wk2[874];                  /* */
union {                                    /* ADTSR */
    unsigned char BYTE;                  /* Byte Access */
    struct {                              /* Bit Access */
        unsigned char :4;              /* */
        unsigned char TRG1S:2;        /* TRG1S */
        unsigned char TRG0S:2;        /* TRG0S */
    } BIT;                               /* */
} ADTSR;
};
struct st_flash {                        /* struct FLASH */

```

図 3.1 SH7145F レジスタ定義 (続き)

```

union {
    unsigned char BYTE;
    struct {
        unsigned char FWE:1;
        unsigned char SWE:1;
        unsigned char ESU:1;
        unsigned char PSU:1;
        unsigned char EV:1;
        unsigned char PV:1;
        unsigned char E:1;
        unsigned char P:1;
    } BIT;
} FLMCR1;
union {
    unsigned char BYTE;
    struct {
        unsigned char FLER:1;
        unsigned char :7;
    } BIT;
} FLMCR2;
union {
    unsigned char BYTE;
    struct {
        unsigned char EB:8;
    } BIT;
} EBR1;
union {
    unsigned char BYTE;
    struct {
        unsigned char :4;
        unsigned char EB11:1;
        unsigned char EB10:1;
        unsigned char EB9:1;
        unsigned char EB8:1;
    } BIT;
} EBR2;
unsigned char wk0[164];
union {
    unsigned short WORD;
    struct {
        unsigned short :12;
        unsigned short RAMS:1;
        unsigned short RAM:3;
    } BIT;
} RAMER;
};
struct st_abc {
    union {
        unsigned short WORD;
        struct {
            unsigned short UBA31:1;
            unsigned short UBA30:1;
            unsigned short UBA29:1;
            unsigned short UBA28:1;
            unsigned short UBA27:1;
            unsigned short UBA26:1;
            unsigned short UBA25:1;
            unsigned short UBA24:1;
            unsigned short UBA23:1;
            unsigned short UBA22:1;
            unsigned short UBA21:1;
            unsigned short UBA20:1;
            unsigned short UBA19:1;
            unsigned short UBA18:1;
            unsigned short UBA17:1;
            unsigned short UBA16:1;
        } BIT;
    } UBARH;
};
/* FLMCR1 */
/* Byte Access */
/* Bit Access */
/* FWE */
/* SWE */
/* ESU */
/* PSU */
/* EV */
/* PV */
/* E */
/* P */
/* */
/* */
/* FLMCR2 */
/* Byte Access */
/* Bit Access */
/* FLER */
/* */
/* */
/* EBR1 */
/* Byte Access */
/* Bit Access */
/* EB */
/* */
/* */
/* EBR2 */
/* Byte Access */
/* Bit Access */
/* EB11 */
/* EB10 */
/* EB9 */
/* EB8 */
/* */
/* */
/* RAMER */
/* Word Access */
/* Bit Access */
/* RAMS */
/* RAM */
/* */
/* */
/* struct UBC */
/* UBARH */
/* Word Access */
/* Bit Access */
/* UBA31 */
/* UBA30 */
/* UBA29 */
/* UBA28 */
/* UBA27 */
/* UBA26 */
/* UBA25 */
/* UBA24 */
/* UBA23 */
/* UBA22 */
/* UBA21 */
/* UBA20 */
/* UBA19 */
/* UBA18 */
/* UBA17 */
/* UBA16 */
/* */
/* */

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

union {
    unsigned short WORD;
    struct {
        unsigned short UBA15:1;
        unsigned short UBA14:1;
        unsigned short UBA13:1;
        unsigned short UBA12:1;
        unsigned short UBA11:1;
        unsigned short UBA10:1;
        unsigned short UBA9:1;
        unsigned short UBA8:1;
        unsigned short UBA7:1;
        unsigned short UBA6:1;
        unsigned short UBA5:1;
        unsigned short UBA4:1;
        unsigned short UBA3:1;
        unsigned short UBA2:1;
        unsigned short UBA1:1;
        unsigned short UBA0:1;
    } BIT;
} UBARL;
/* UBARL
/* Word Access */
/* Bit Access */
/* UBA15 */
/* UBA14 */
/* UBA13 */
/* UBA12 */
/* UBA11 */
/* UBA10 */
/* UBA9 */
/* UBA8 */
/* UBA7 */
/* UBA6 */
/* UBA5 */
/* UBA4 */
/* UBA3 */
/* UBA2 */
/* UBA1 */
/* UBA0 */
/*
/*
union {
    unsigned short WORD;
    struct {
        unsigned short UBM31:1;
        unsigned short UBM30:1;
        unsigned short UBM29:1;
        unsigned short UBM28:1;
        unsigned short UBM27:1;
        unsigned short UBM26:1;
        unsigned short UBM25:1;
        unsigned short UBM24:1;
        unsigned short UBM23:1;
        unsigned short UBM22:1;
        unsigned short UBM21:1;
        unsigned short UBM20:1;
        unsigned short UBM19:1;
        unsigned short UBM18:1;
        unsigned short UBM17:1;
        unsigned short UBM16:1;
    } BIT;
} UBAMRH;
/* UBAMRH
/* Word Access */
/* Bit Access */
/* UBM31 */
/* UBM30 */
/* UBM29 */
/* UBM28 */
/* UBM27 */
/* UBM26 */
/* UBM25 */
/* UBM24 */
/* UBM23 */
/* UBM22 */
/* UBM21 */
/* UBM20 */
/* UBM19 */
/* UBM18 */
/* UBM17 */
/* UBM16 */
/*
/*
union {
    unsigned short WORD;
    struct {
        unsigned short UBM15:1;
        unsigned short UBM14:1;
        unsigned short UBM13:1;
        unsigned short UBM12:1;
        unsigned short UBM11:1;
        unsigned short UBM10:1;
        unsigned short UBM9:1;
        unsigned short UBM8:1;
        unsigned short UBM7:1;
        unsigned short UBM6:1;
        unsigned short UBM5:1;
        unsigned short UBM4:1;
        unsigned short UBM3:1;
        unsigned short UBM2:1;
        unsigned short UBM1:1;
        unsigned short UBM0:1;
    } BIT;
} UBAMRL;
/* UBAMRL
/* Word Access */
/* Bit Access */
/* UBM15 */
/* UBM14 */
/* UBM13 */
/* UBM12 */
/* UBM11 */
/* UBM10 */
/* UBM9 */
/* UBM8 */
/* UBM7 */
/* UBM6 */
/* UBM5 */
/* UBM4 */
/* UBM3 */
/* UBM2 */
/* UBM1 */
/* UBM0 */
/*
/*
union {
    unsigned short WORD;
    struct {
        unsigned short :8;
        unsigned short CP:2;
    }
} UBBR;
/* UBBR
/* Word Access */
/* Bit Access */
/*
/* CP

```

図 3.1 SH7145F レジスタ定義 (続き)



```

        unsigned short ID:2;          /* ID          */
        unsigned short RW:2;         /* RW          */
        unsigned short SZ:2;         /* SZ          */
    } BIT;
} UBRR;
/*          */
union {
    unsigned short WORD;             /* UBCR        */
    struct {                          /* Word Access */
        unsigned short :15;          /* Bit Access  */
        unsigned short UBID:1;      /* UBID        */
    } BIT;
} UBCR;
/*          */
};
/*          */
struct st_wdt {
    union {
        unsigned char BYTE;          /* TCSR        */
        struct {                      /* Byte Access */
            unsigned char OVF:1;     /* Bit Access  */
            unsigned char WTIT:1;    /* OVF         */
            unsigned char TME:1;     /* WT/IT      */
            unsigned char :2;        /* TME        */
            unsigned char CKS:3;     /* CKS        */
        } BIT;
    } TCSR;
    unsigned char TCNT;              /* TCNT        */
    union {
        unsigned char BYTE;          /* RSTCSR      */
        struct {                      /* Byte Access */
            unsigned char WOVF:1;    /* Bit Access  */
            unsigned char RSTE:1;    /* WOVF       */
            unsigned char RSTS:1;    /* RSTE       */
            unsigned char :5;        /* RSTS       */
        } BIT;
    } RSTCSR;
}
/*          */
};
/*          */
struct st_stby {
    union {
        unsigned char BYTE;          /* SBYCR       */
        struct {                      /* Byte Access */
            unsigned char SBY:1;     /* Bit Access  */
            unsigned char HIZ:1;     /* SBY         */
            unsigned char :4;        /* HIZ        */
            unsigned char IRQEH:1;   /* IRQEH      */
            unsigned char IRQEL:1;   /* IRQEL      */
        } BIT;
    } SBYCR;
    unsigned char wk0[3];
    union {
        unsigned char BYTE;          /* SYSCR       */
        struct {                      /* Byte Access */
            unsigned char :6;        /* Bit Access  */
            unsigned char AUDSRST:1; /* AUDSRST    */
            unsigned char RAME:1;    /* RAME        */
        } BIT;
    } SYSCR;
    unsigned char wk1[3];
    union {
        unsigned short WORD;         /* MSTCR1      */
        struct {                      /* Word Access */
            unsigned short :4;       /* Bit Access  */
            unsigned short MSTP27:1; /* MSTP27     */
            unsigned short MSTP26:1; /* MSTP26     */
            unsigned short MSTP25:1; /* MSTP25     */
            unsigned short MSTP24:1; /* MSTP24     */
            unsigned short :2;       /* MSTP23     */
            unsigned short MSTP21:1; /* MSTP21     */
            unsigned short :1;       /* MSTP20     */
            unsigned short MSTP19:1; /* MSTP19     */
        }
    }
}

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        unsigned short MSTP18:1;      /* MSTP18 */
        unsigned short MSTP17:1;      /* MSTP17 */
        unsigned short MSTP16:1;      /* MSTP16 */
    } BIT;
} MSTCR1;
/* */
union {
    unsigned short WORD;              /* Word Access */
    struct {
        unsigned short :2;            /* Bit Access */
        unsigned short MSTP13:1;      /* MSTP13 */
        unsigned short MSTP12:1;      /* MSTP12 */
        unsigned short :6;            /* */
        unsigned short MSTP5:1;        /* MSTP5 */
        unsigned short MSTP4:1;        /* MSTP4 */
        unsigned short MSTP3:1;        /* MSTP3 */
        unsigned short MSTP2:1;        /* MSTP2 */
        unsigned short :1;            /* */
        unsigned short MSTP0:1;        /* MSTP0 */
    } BIT;
} MSTCR2;
/* */
};
/* */
struct st_bsc {
    union {
        unsigned short WORD;          /* BCR1 Word Access */
        struct {
            unsigned short :2;         /* Bit Access */
            unsigned short MTURWE:1;   /* MTURWE */
            unsigned short :5;         /* */
            unsigned short A3LG:1;     /* A3LG */
            unsigned short A2LG:1;     /* A2LG */
            unsigned short A1LG:1;     /* A1LG */
            unsigned short A0LG:1;     /* A0LG */
            unsigned short A3SZ:1;     /* A3SZ */
            unsigned short A2SZ:1;     /* A2SZ */
            unsigned short A1SZ:1;     /* A1SZ */
            unsigned short A0SZ:1;     /* A0SZ */
        } BIT;
    } BCR1;
    union {
        unsigned short WORD;          /* BCR2 Word Access */
        struct {
            unsigned short IW3:2;      /* Bit Access IW3 */
            unsigned short IW2:2;      /* IW2 */
            unsigned short IW1:2;      /* IW1 */
            unsigned short IW0:2;      /* IW0 */
            unsigned short CW3:1;      /* CW3 */
            unsigned short CW2:1;      /* CW2 */
            unsigned short CW1:1;      /* CW1 */
            unsigned short CW0:1;      /* CW0 */
            unsigned short SW3:1;      /* SW3 */
            unsigned short SW2:1;      /* SW2 */
            unsigned short SW1:1;      /* SW1 */
            unsigned short SW0:1;      /* SW0 */
        } BIT;
    } BCR2;
    union {
        unsigned short WORD;          /* WCR1 Word Access */
        struct {
            unsigned short W3:4;       /* Bit Access W3 */
            unsigned short W2:4;       /* W2 */
            unsigned short W1:4;       /* W1 */
            unsigned short W0:4;       /* W0 */
        } BIT;
    } WCR1;
    union {
        unsigned short WORD;          /* WCR2 Word Access */
        struct {
            unsigned short :12;        /* Bit Access */
        } BIT;
    } WCR2;
};
/* */

```

図 3.1 SH7145F レジスタ定義 (続き)

```

        unsigned short DSW:4;          /* DSW */
    } BIT;                             /* */
    } WCR2;                             /* */
};                                     /* */
struct st_dmac {                       /* struct DMAC */
    union {                             /* DMAOR */
        unsigned short WORD;          /* Word Access */
        struct {                      /* Bit Access */
            unsigned short :6;        /* */
            unsigned short PR:2;     /* PR */
            unsigned short :5;       /* */
            unsigned short AE:1;     /* AE */
            unsigned short NMIF:1;   /* NMIF */
            unsigned short DME:1;    /* DME */
        } BIT;                       /* */
        } DMAOR;                     /* */
};                                     /* */
struct st_dmac0 {                      /* struct DMAC0 */
    unsigned long SAR0;                /* SAR0 */
    unsigned long DAR0;                /* DAR0 */
    unsigned long DMATCR0;             /* DMATCR0 */
    union {                            /* CHCR0 */
        unsigned long LONG;          /* Long Access */
        struct {                     /* Bit Access */
            unsigned long :13;       /* */
            unsigned long RL:1;      /* RL */
            unsigned long AM:1;      /* AM */
            unsigned long AL:1;      /* AL */
            unsigned long DM:2;      /* DM */
            unsigned long SM:2;      /* SM */
            unsigned long RS:4;      /* RS */
            unsigned long :1;        /* */
            unsigned long DS:1;      /* DS */
            unsigned long TM:1;      /* TM */
            unsigned long TS:2;      /* TS */
            unsigned long IE:1;      /* IE */
            unsigned long TE:1;      /* TE */
            unsigned long DE:1;      /* DE */
        } BIT;                       /* */
        } CHCR0;                     /* */
};                                     /* */
struct st_dmac1 {                      /* struct DMAC1 */
    unsigned long SAR1;                /* SAR1 */
    unsigned long DAR1;                /* DAR1 */
    unsigned long DMATCR1;             /* DMATCR1 */
    union {                            /* CHCR1 */
        unsigned long LONG;          /* Long Access */
        struct {                     /* Bit Access */
            unsigned long :13;       /* */
            unsigned long RL:1;      /* RL */
            unsigned long AM:1;      /* AM */
            unsigned long AL:1;      /* AL */
            unsigned long DM:2;      /* DM */
            unsigned long SM:2;      /* SM */
            unsigned long RS:4;      /* RS */
            unsigned long :1;        /* */
            unsigned long DS:1;      /* DS */
            unsigned long TM:1;      /* TM */
            unsigned long TS:2;      /* TS */
            unsigned long IE:1;      /* IE */
            unsigned long TE:1;      /* TE */
            unsigned long DE:1;      /* DE */
        } BIT;                       /* */
        } CHCR1;                     /* */
};                                     /* */
struct st_dmac2 {                      /* struct DMAC2 */
    unsigned long SAR2;                /* SAR2 */
    unsigned long DAR2;                /* DAR2 */

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

unsigned long DMATCR2;
union {
    unsigned long LONG;
    struct {
        unsigned long :12;
        unsigned long RO:1;
        unsigned long :3;
        unsigned long DM:2;
        unsigned long SM:2;
        unsigned long RS:4;
        unsigned long :2;
        unsigned long TM:1;
        unsigned long TS:2;
        unsigned long IE:1;
        unsigned long TE:1;
        unsigned long DE:1;
    } BIT;
} CHCR2;
};
struct st_dmac3 {
    unsigned long SAR3;
    unsigned long DAR3;
    unsigned long DMATCR3;
    union {
        unsigned long LONG;
        struct {
            unsigned long :11;
            unsigned long DI:1;
            unsigned long :4;
            unsigned long DM:2;
            unsigned long SM:2;
            unsigned long RS:4;
            unsigned long :2;
            unsigned long TM:1;
            unsigned long TS:2;
            unsigned long IE:1;
            unsigned long TE:1;
            unsigned long DE:1;
        } BIT;
    } CHCR3;
};
struct st_dtc {
    union {
        unsigned char BYTE;
        struct {
            unsigned char DTEA:8;
        } BIT;
    } DTEA;
    union {
        unsigned char BYTE;
        struct {
            unsigned char DTEB:8;
        } BIT;
    } DTEB;
    union {
        unsigned char BYTE;
        struct {
            unsigned char DTEC:8;
        } BIT;
    } DTEC;
    union {
        unsigned char BYTE;
        struct {
            unsigned char DTED:8;
        } BIT;
    } DTED;
    unsigned char wk0[2];
    union {

```

図 3.1 SH7145F レジスタ定義 (続き)

```

unsigned short WORD;          /* Word Access */
struct {                      /* Bit Access */
    unsigned short :5;       /* */
    unsigned short NMIF:1;   /* NMIF */
    unsigned short AE:1;     /* AE */
    unsigned short SWDTE:1;  /* SWDTE */
    unsigned short DTVEC7:1; /* DTVEC7 */
    unsigned short DTVEC6:1; /* DTVEC6 */
    unsigned short DTVEC5:1; /* DTVEC5 */
    unsigned short DTVEC4:1; /* DTVEC4 */
    unsigned short DTVEC3:1; /* DTVEC3 */
    unsigned short DTVEC2:1; /* DTVEC2 */
    unsigned short DTVEC1:1; /* DTVEC1 */
    unsigned short DTVEC0:1; /* DTVEC0 */
} BIT;                        /* */
} DTCSR;                       /* DTBR */
unsigned short DTBR;          /* */
unsigned char wk1[6];        /* DTEE */
union {                       /* Byte Access */
    unsigned char BYTE;      /* Bit Access */
    struct {
        unsigned char :2;   /* */
        unsigned char DTEE5:1; /* DTEE5 */
        unsigned char :1;   /* */
        unsigned char DTEE3:1; /* DTEE3 */
        unsigned char DTEE2:1; /* DTEE2 */
        unsigned char DTEE1:1; /* DTEE1 */
        unsigned char DTEE0:1; /* DTEE0 */
    } BIT;                  /* */
    } DTEE;                 /* */
unsigned char wk2[1];        /* DTEG */
union {                       /* Byte Access */
    unsigned char BYTE;      /* Bit Access */
    struct {
        unsigned char DTEG7:1; /* DTEG7 */
        unsigned char :7;     /* */
    } BIT;                  /* */
    } DTEG;                 /* */
};
struct st_iic {
    union {
        unsigned char BYTE; /* Byte Access */
        struct {
            unsigned char :2; /* Bit Access */
            unsigned char IICX0:1; /* IICX0 */
            unsigned char IICE:1; /* IICE */
            unsigned char HNDS:1; /* HNDS */
            unsigned char :1; /* */
            unsigned char ICDRF0:1; /* ICDRF0 */
            unsigned char STOPIM:1; /* STOPIM */
        } BIT; /* */
        } SCRX; /* */
    unsigned char wk0[23]; /* ICCR0 */
    union {
        unsigned char BYTE; /* Byte Access */
        struct {
            unsigned char ICE:1; /* ICE */
            unsigned char IEIC:1; /* IEIC */
            unsigned char MST:1; /* MST */
            unsigned char TRS:1; /* TRS */
            unsigned char ACKE:1; /* ACKE */
            unsigned char BBSY:1; /* BBSY */
            unsigned char IRIC:1; /* IRIC */
            unsigned char SCP:1; /* SCP */
        } BIT; /* */
        } ICCR0; /* */
    } ICSR0; /* ICSR0 */
    unsigned char BYTE; /* Byte Access */
};

```

図 3.1 SH7145F レジスタ定義 (続き)

### 3. 付録

```

        struct {
            unsigned char ESTP:1;
            unsigned char STOP:1;
            unsigned char IRTR:1;
            unsigned char AASX:1;
            unsigned char AL:1;
            unsigned char AAS:1;
            unsigned char ADZ:1;
            unsigned char ACKB:1;
        } BIT;
    } ICSR0;
    unsigned char wk1[4];
    union {
        unsigned char BYTE;
        struct {
            unsigned char ICDR:8;
        } BIT;
    } ICDR0;
    union {
        unsigned char BYTE;
        struct {
            unsigned char MLS:1;
            unsigned char WAIT:1;
            unsigned char CKS:3;
            unsigned char BC:3;
        } BIT;
    } ICMR0;
};

struct st_hudi {
    union {
        unsigned short WORD;
        struct {
            unsigned short TS:4;
            unsigned short :12;
        } BIT;
    } SDIR;
    union {
        unsigned short WORD;
        struct {
            unsigned short :15;
            unsigned short SDTRF:1;
        } BIT;
    } SDRS;
    unsigned short SDDRH;
    unsigned short SDDRL;
};

#define P_SCI0 (*(volatile struct st_sci0 *)0xFFFF81A0)/* SCI0 Address */
#define P_SCI1 (*(volatile struct st_sci1 *)0xFFFF81B0)/* SCI1 Address */
#define P_SCI2 (*(volatile struct st_sci2 *)0xFFFF81C0)/* SCI2 Address */
#define P_SCI3 (*(volatile struct st_sci3 *)0xFFFF81D0)/* SCI3 Address */
#define P_MTU34 (*(volatile struct st_mtu34 *)0xFFFF8200)/* MTU34 Address */
#define P_MTU0 (*(volatile struct st_mtu0 *)0xFFFF8260)/* MTU0 Address */
#define P_MTU1 (*(volatile struct st_mtu1 *)0xFFFF8280)/* MTU1 Address */
#define P_MTU2 (*(volatile struct st_mtu2 *)0xFFFF82A0)/* MTU2 Address */
#define P_INTC (*(volatile struct st_intc *)0xFFFF8348)/* INTC Address */
#define P_PORTA (*(volatile struct st_porta *)0xFFFF8380)/* PORTA Address */
#define P_PORTB (*(volatile struct st_portb *)0xFFFF8390)/* PORTB Address */
#define P_PORTC (*(volatile struct st_portc *)0xFFFF8392)/* PORTC Address */
#define P_PORTD (*(volatile struct st_portd *)0xFFFF83A0)/* PORTD Address */
#define P_PORTE (*(volatile struct st_porte *)0xFFFF83B0)/* PORTE Address */
#define P_PORTF (*(volatile struct st_portf *)0xFFFF83B2)/* PORTF Address */
#define P_MTU (*(volatile struct st_mtu *)0xFFFF83C0) /* MTU Address */
#define P_CMT (*(volatile struct st_cmt *)0xFFFF83D0) /* CMT Address */
#define P_AD (*(volatile struct st_ad *)0xFFFF8420) /* A/D Address */
#define P_FLASH (*(volatile struct st_flash *)0xFFFF8580)/* FLASH Address */
#define P_UBC (*(volatile struct st_abc *)0xFFFF8600) /* UBC Address */
#define P_WDT (*(volatile struct st_wdt *)0xFFFF8610) /* WDT Address */
#define P_STBY (*(volatile struct st_stby *)0xFFFF8614)/* STBY Address */

```

図 3.1 SH7145F レジスタ定義 (続き)

```
#define P_BSC (*(volatile struct st_bsc *)0xFFFF8620) /* BSC Address */
#define P_DMACH (*(volatile struct st_dmac *)0xFFFF86B0) /* DMACH Address */
#define P_DMACH0 (*(volatile struct st_dmac0 *)0xFFFF86C0) /* DMACH0 Address */
#define P_DMACH1 (*(volatile struct st_dmac1 *)0xFFFF86D0) /* DMACH1 Address */
#define P_DMACH2 (*(volatile struct st_dmac2 *)0xFFFF86E0) /* DMACH2 Address */
#define P_DMACH3 (*(volatile struct st_dmac3 *)0xFFFF86F0) /* DMACH3 Address */
#define P_DTC (*(volatile struct st_dtc *)0xFFFF8700) /* DTC Address */
#define P_IIC (*(volatile struct st_iic *)0xFFFF87F0) /* IIC Address */
#define P_HUDI (*(volatile struct st_hudi *)0xFFFF8A50) /* H-UDI Address */
```

図 3.1 SH7145F レジスタ定義 (続き)





---

SH7144Fグループ 内蔵インタフェース I<sup>2</sup>Cバスインタフェース編  
アプリケーションノート

発行年月 2003年 6月 27日 Rev.1.00

発行 株式会社ルネサス テクノロジ 営業企画統括部  
〒100-0004 東京都千代田区大手町 2-6-2

編集 株式会社ルネサス小平セミコン 技術ドキュメント部

---

©2003 Renesas Technology Corp. All rights reserved. Printed in Japan.



<http://www.renesas.com>

営業お問合せ窓口

ルネサス販売本社	〒100-0005	千代田区丸の内1-8-2 (第二鉄鋼ビル)	(03) 3215-8600
京浜支社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西東京支社	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
札幌支店	〒060-0002	札幌市中央区北二条西4-1 (札幌三井ビル5F)	(011) 210-8717
東北支社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア13F)	(022) 221-1351
いわき支店	〒970-8026	いわき市平小太郎町4-9 (損保ジャパンいわき第二ビル3F)	(0246) 22-3222
茨城支社	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田1F)	(029) 271-9411
新潟支店	〒950-0087	新潟市東大通1-4-2 (新潟三井物産ビル3F)	(025) 241-4361
松本支社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中部営業本部	〒460-0008	名古屋市中区栄3-13-20 (栄センタービル4F)	(052) 261-3000
浜松支店	〒430-7710	浜松市板屋町111-2 (浜松アクタワー10F)	(053) 451-2131
西部営業本部	〒541-0044	大阪市中央区伏見町4-1-1 (大阪明治生命館ランドアクシスタワー10F)	(06) 6233-9500
北陸支社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
中松支社	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング8F)	(082) 244-2570
鳥取支店	〒790-0003	松山市三番町4-4-6 (GEエジソンビル松山2号館3F)	(089) 933-9595
九州支社	〒680-0822	鳥取市今町2-251 (日本生命鳥取駅前ビル)	(0857) 21-1915
鹿児支店	〒812-0011	福岡市博多区博多駅前2-17-1 (ヒロカネビル本館5F)	(092) 481-7695
	〒890-0053	鹿児島市中央町12-2 (明治生命西鹿児島ビル2F)	(099) 256-9021

■ 技術的なお問合せおよび資料のご請求は下記へどうぞ。  
 総合お問合せ窓口：カスタマサポートセンタ E-Mail: [csc@renesas.com](mailto:csc@renesas.com)

# SH7144F グループ 内蔵インタフェース I<sup>2</sup>C バスインタフェース編 アプリケーションノート



ルネサスエレクトロニクス株式会社  
神奈川県川崎市中原区下沼部1753 〒211-8668

RJJ05B0092-0100H