

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

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社が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。

SH7145 グループ

シリアル EEPROM (I²C EEPROM) のリード・ライト応用例

要旨

I²C モジュールを用いて、2 線式シリアル方式 (I²C バス) の EEPROM に対して、リード・ライトを行う応用例です。

動作確認デバイス

SH7145F

目次

1. I ² C バスの概要	2
2. SH7144F シリーズ応用例	11
3. 付録	62

1. I²C バスの概要

1.1 I²C バスの特長

1.1.1 I²C バスの特長

I²C バスの特長を以下に示します。

- 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。I²C 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パスが形成され、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、I²C 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 (I²C バス仕様書 Ver2.0 では、3.4Mbps まで定義されています)。
- I²C 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*1ACCESS. bus*2 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。

1.1.2 シリアルインタフェース(SCI)との相違

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用データラインの 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、I²C 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 I²C 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 SCI との相違点

	SCI		I ² 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】 1* I²C 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 I²C バスの接続形式

図 1 に I²C 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 I²C 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ても、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。

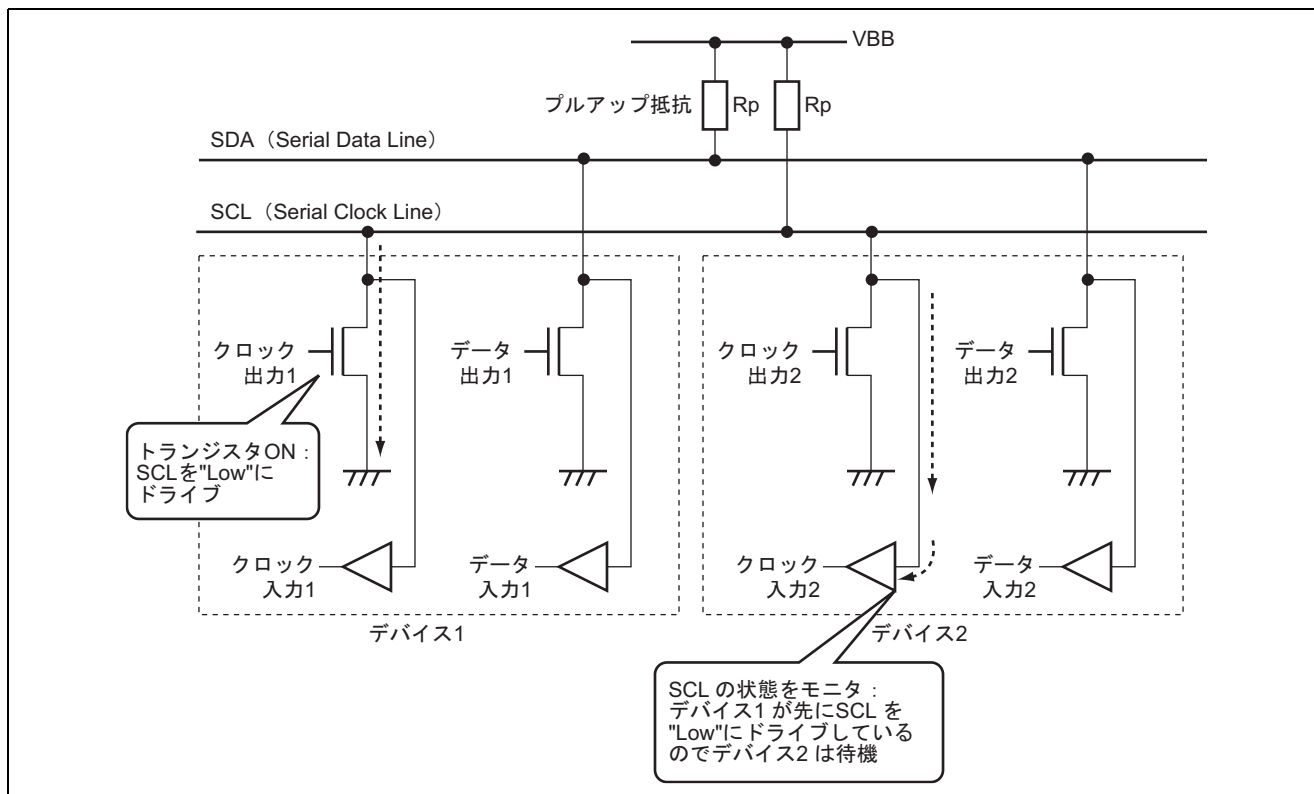


図 1 バスインタフェースの接続形式 (デバイス 1 が最初に SCL を“Low”にドライブした場合)

1.2 I²C バスを使用したデータ転送方法

1.2.1 I²C バスを使用したデータ転送の基本事項

はじめに I²C 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 I²C 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 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。

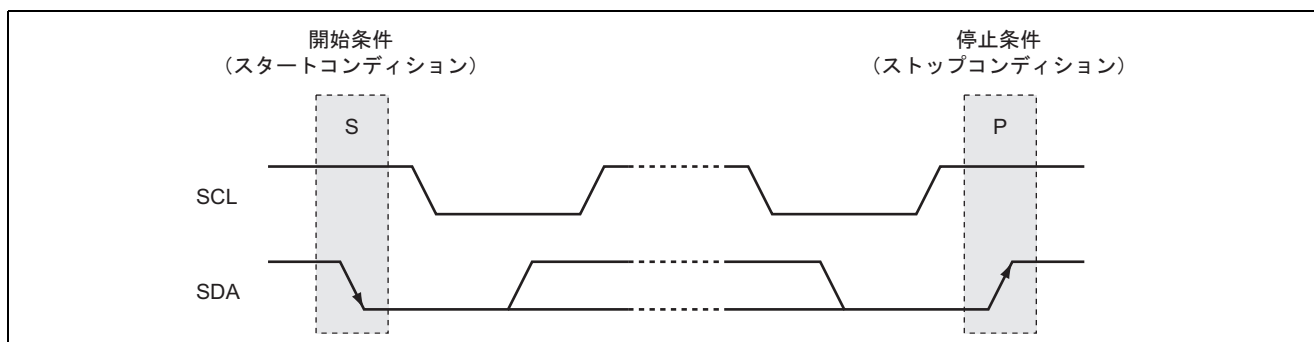


図 2 開始条件と停止条件

(6) データの出力タイミング

図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。

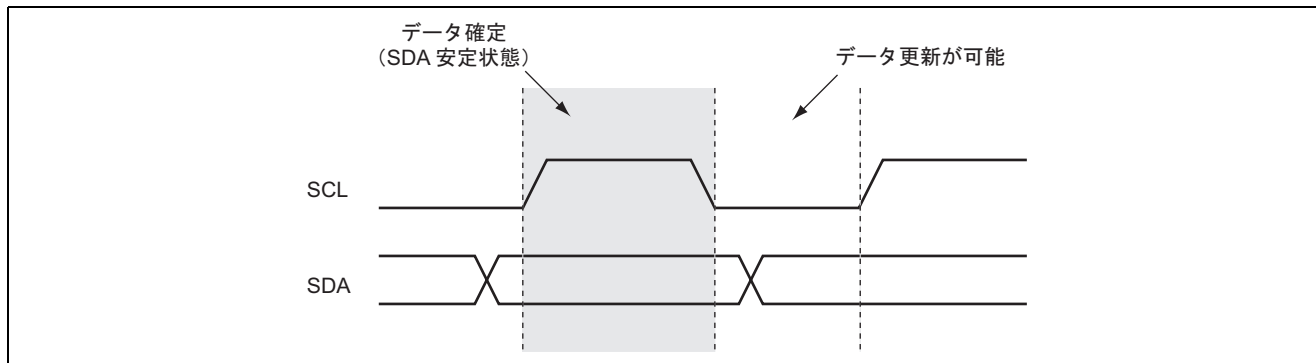


図3 データ出力タイミング

(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I²C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、I²C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。

(13) データ転送フォーマット

図 4 に I²C 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'00000000～H'11111111 の間で設定しますが、アドレス B'00000000 (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 I²C 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 I²C 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'00000000 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。

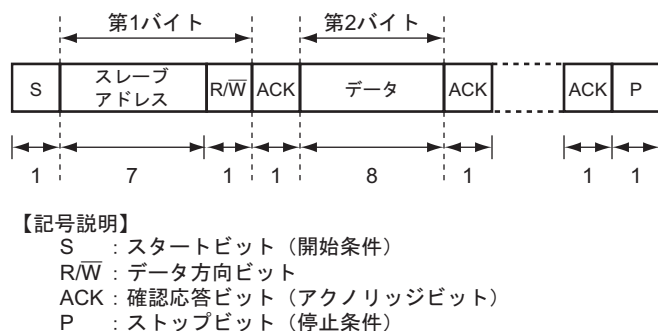


図 4 データ転送フォーマット

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 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せん。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。

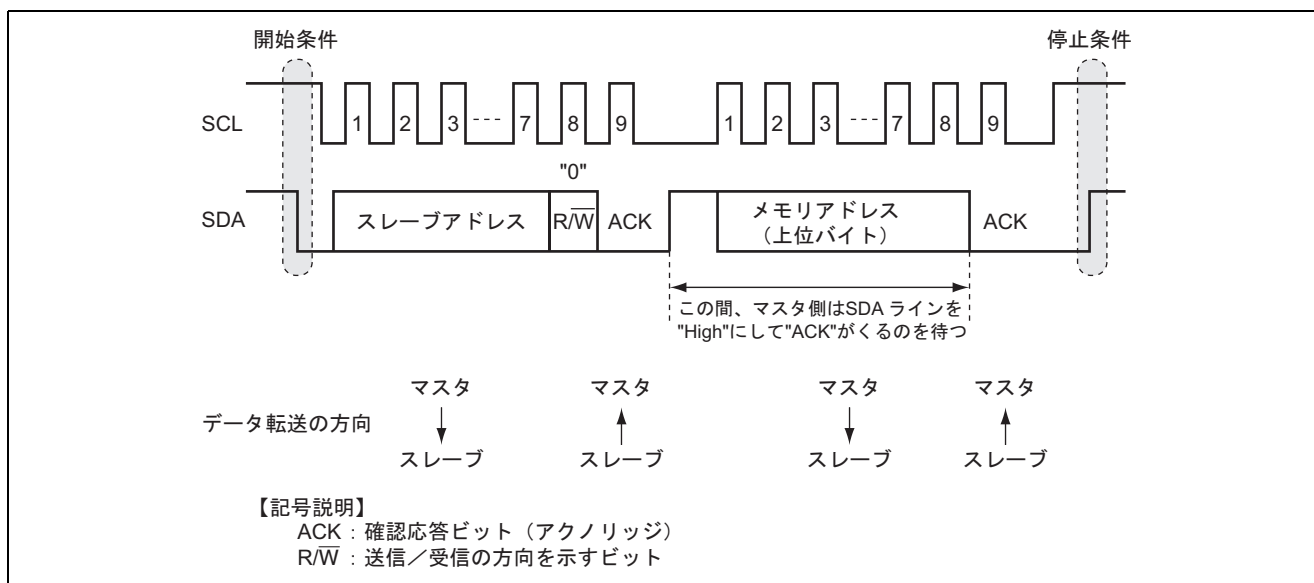


図 5 データ転送フォーマット (マスタ=送信デバイス、スレーブ=受信デバイスの場合)

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 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。

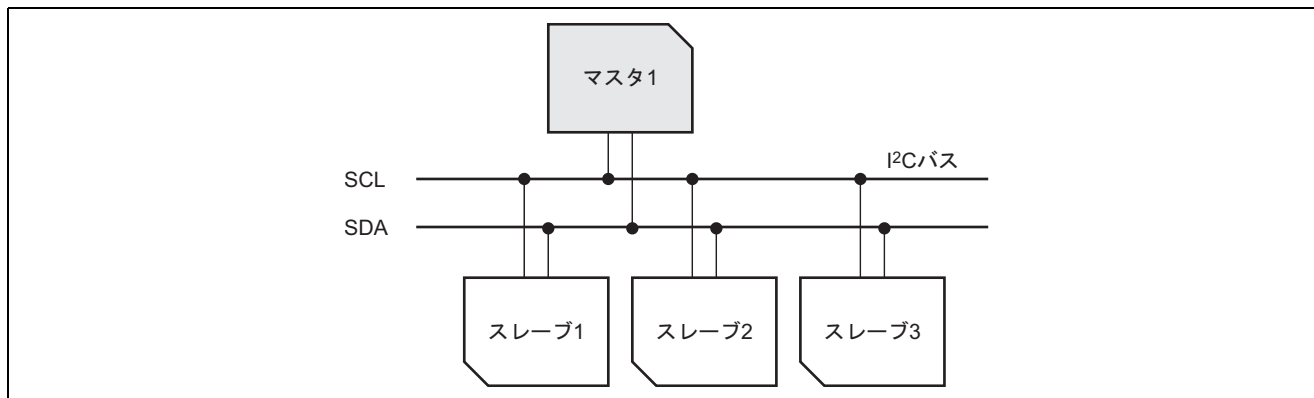


図 6 シングルマスタ構成

1.3.2 マルチマスタ

図 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、I²C 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。

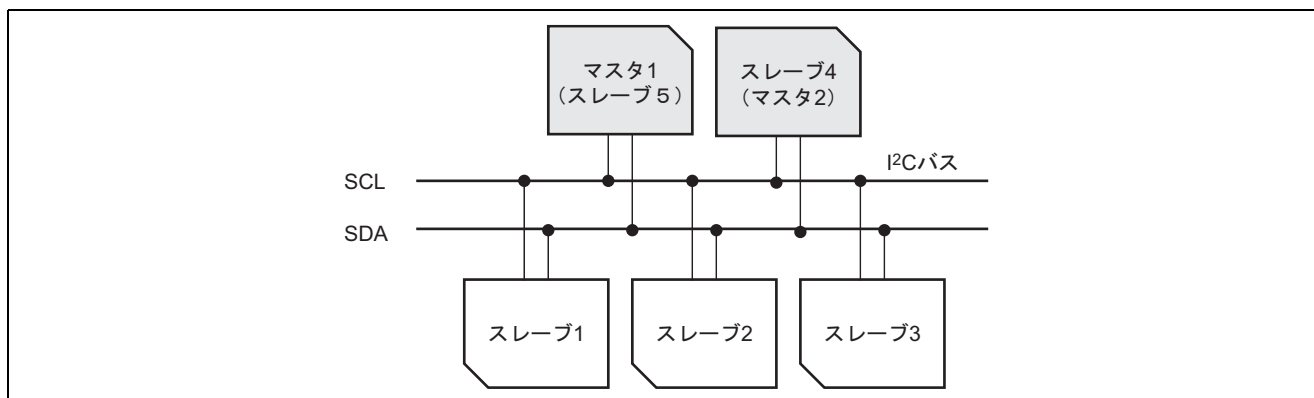


図 7 マルチマスタ構成

1.4 通信調整手順

I²C 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 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。

I²C 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 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。

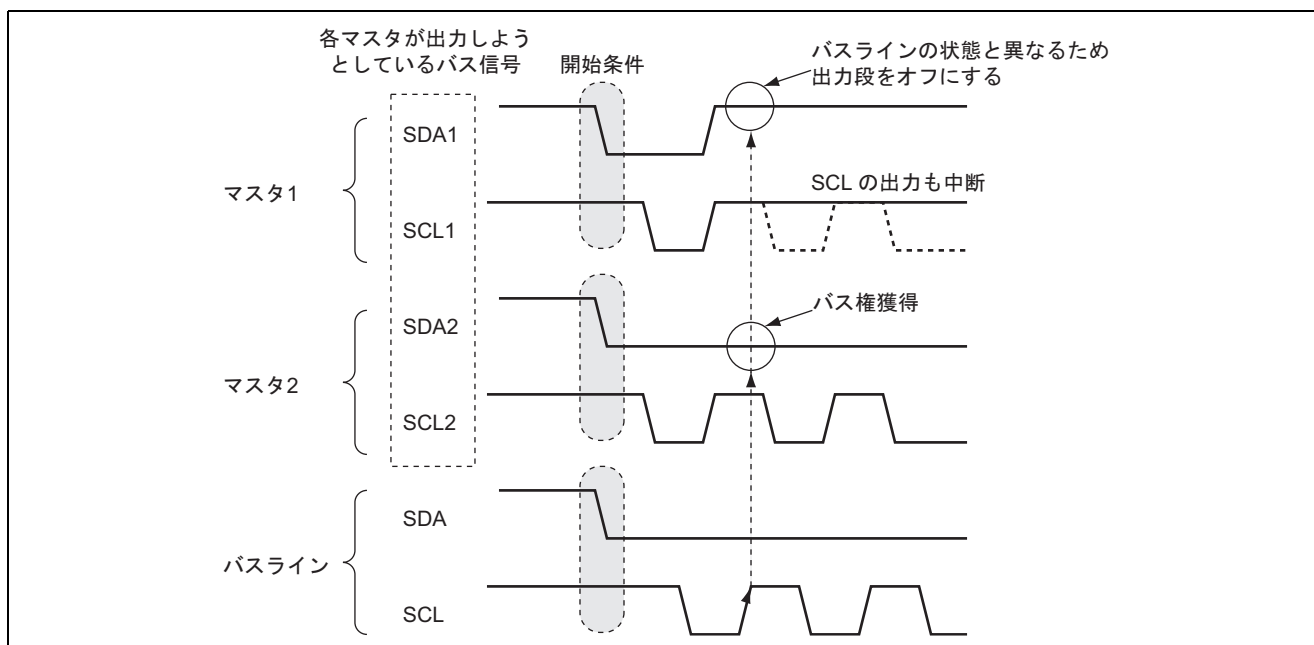


図 8 通信調整手順（バスアービトレーションロストの検出）

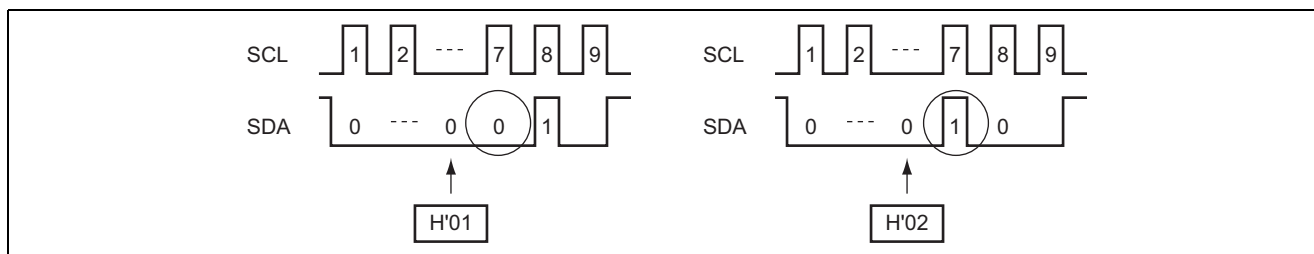


図 9 通信調整の具体例

2. SH7144F シリーズ応用例

2.1 SH7144F シリーズ応用例手引き

2.1.1 SH7144F シリーズ応用例の構成

SH7144F シリーズ応用例は図 10 に示す構成で SH7144F シリーズの I²C 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。

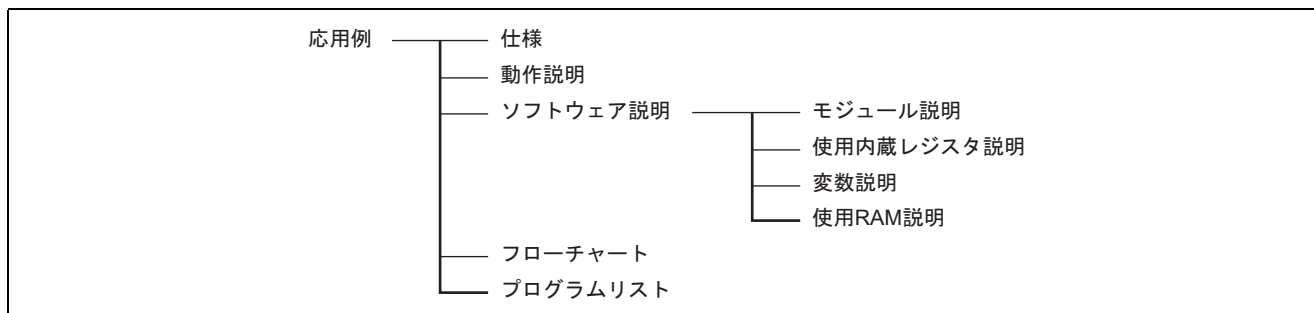


図 10 SH7144F シリーズ応用例の構成

(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 I²C 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。
- (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。割り込み処理ルーチンの先頭アドレスを記述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。

main()関数を使用する例を示します。main()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 main()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) =
{
    main,                  /* NO.   Offset      Exception Sources      */
    (void (*)(void))0xfffffc, /* (001) H'00000004  Power-on Reset SP      */
    main,                  /* (002) H'00000008  Manual reset PC       */
    (void (*)(void))0xfffffc, /* (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)	*/
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	*/
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	SCI0/ERI	*/
dummy,	/* (129) H'00000204	SCI0/RXI	*/
dummy,	/* (130) H'00000208	SCI0/TXI	*/
dummy,	/* (131) H'0000020C	SCI0/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	*/
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)	*/
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)	*/
};			

2.1.3 レジスタ定義ファイル説明

SH7145F のレジスタ定義を付録の 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 I²C 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 I²C 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)I²C 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 11 に SH7145F と EEPROM の接続例を示します。

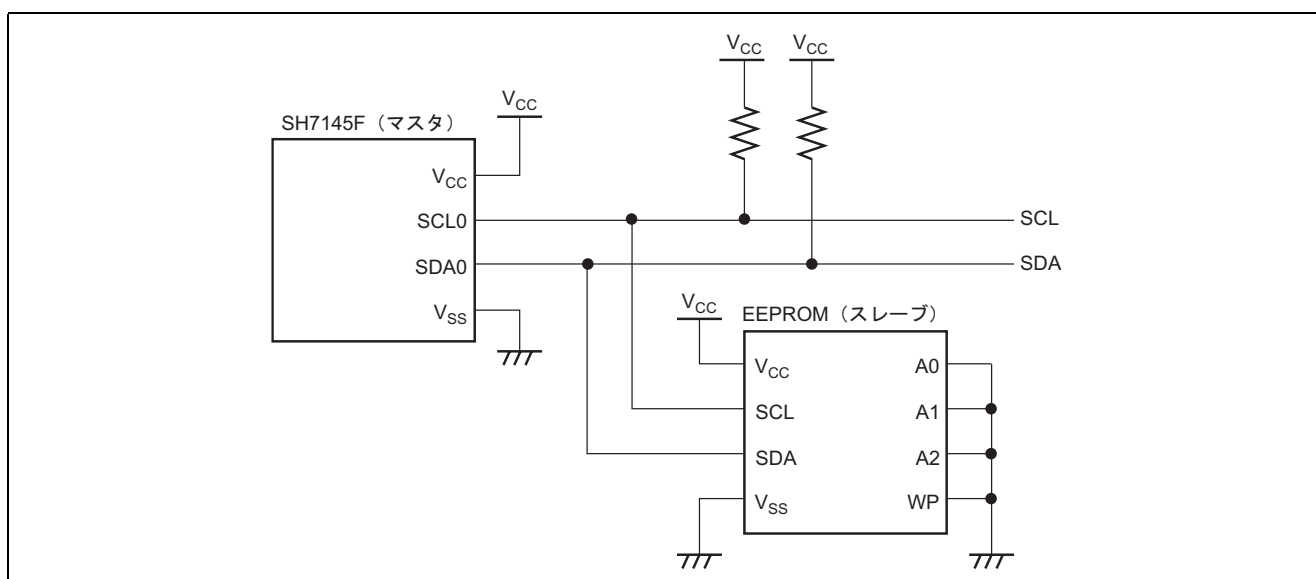


図 11 SH7145F と EEPROM との接続例

本タスクで例で使用する I²C バスフォーマットを図 12 に示します。

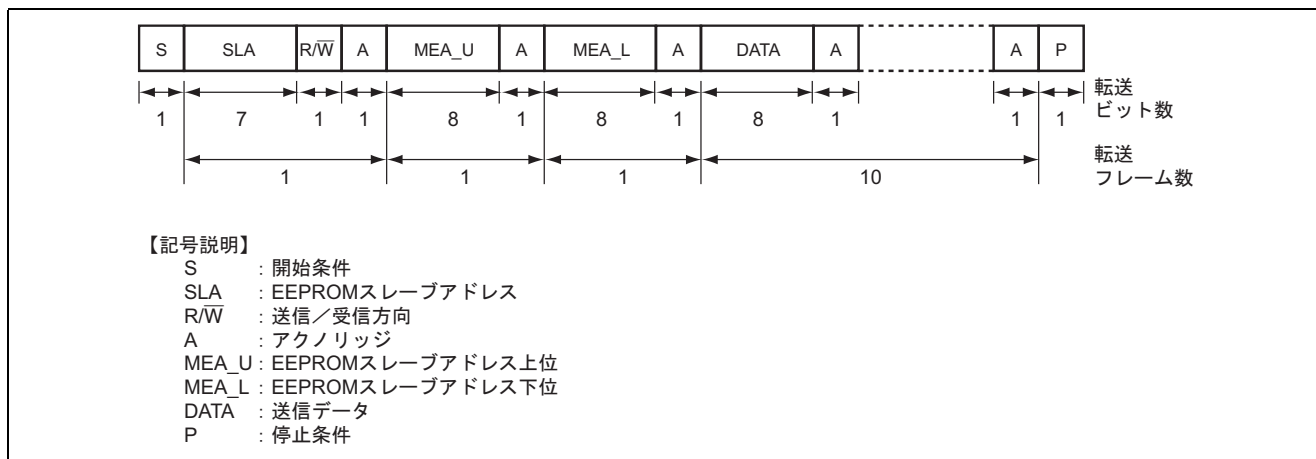
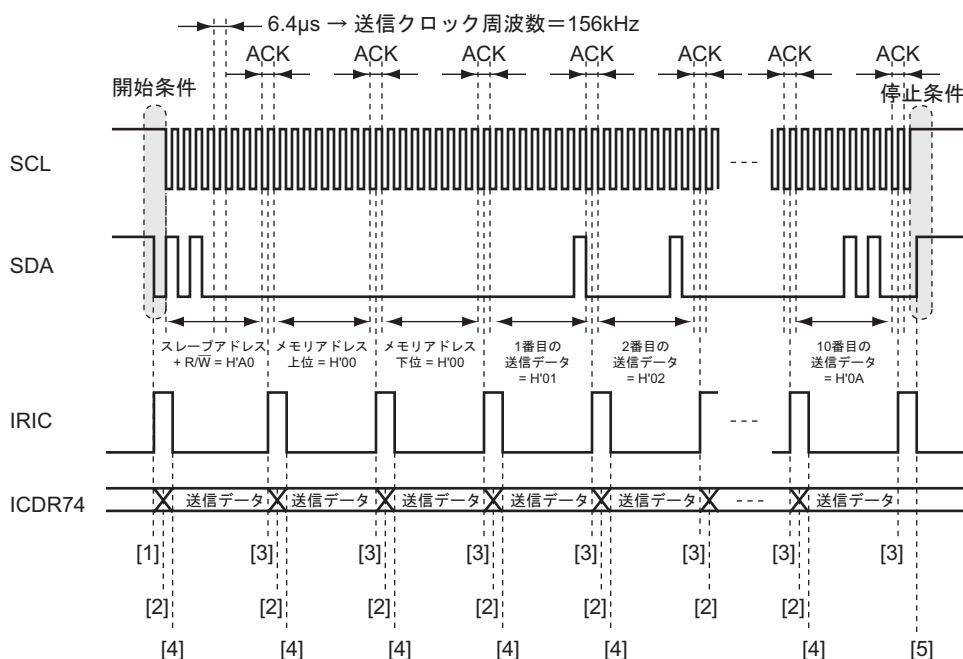


図 12 本タスクで使用する転送フォーマット

2.2.2 動作説明

図 13 に動作を示します。



	ソフトウェア	ハードウェア
[1]	開始条件の発行 (BBSY=1、SCP=0)	IRIC=1、IRTR= 1 (フラグがセット)
[2]	ICDRに送信データをライト	処理なし
[3]	処理なし	データ送信の終了でIRIC=1 (送信クロックの9クロック目の立ち上がり時)
[4]	送信終了判定のためにIRICを0にクリア	処理なし
[5]	停止条件の発行 (BBSY=0、SCP=0)	処理なし

図 13 シングルマスタ送信動作

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 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 ² 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 3 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 ² 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 ² C の SDA0 入出力端子に設定 ポート B2(PB2)の端子機能を I ² C の SCL0 入出力端子に設定	H'FFFF839A	H'0000
ICDR	I ² C バス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	-
SAR	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 ² C バスにつながっている他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 ² C バスフォーマット。	ビット 0	
SARX	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 ² C バスにつながっている他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 ² C バスフォーマット。	ビット 0	

レジスタ名 ビット	機能	アドレス	設定値
		ビット	
ICMR	I ² 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 ² 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 ² C バスコントロールレジスタ	H'FFFF8808	H'B9
ICE	I ² C バスインタフェースイネーブル (ICE) ICE=B'1 のとき、I ² 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 ² 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'1 のとき、マスタモード	ビット 5	
TRS	送信/受信選択 TES=B'1 のとき、送信モード	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	2C 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	
ICSR	I ² 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 ² 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	アービトラ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	

レジスタ名 ビット	機能	アドレス	設定値
		ビット	
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	
IICX	I ² C トランスファレートセレクト 0 ICMR の CKS[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 ² C マスタイネーブル IICE=B'1 のとき、I ² 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	
ICDRF	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

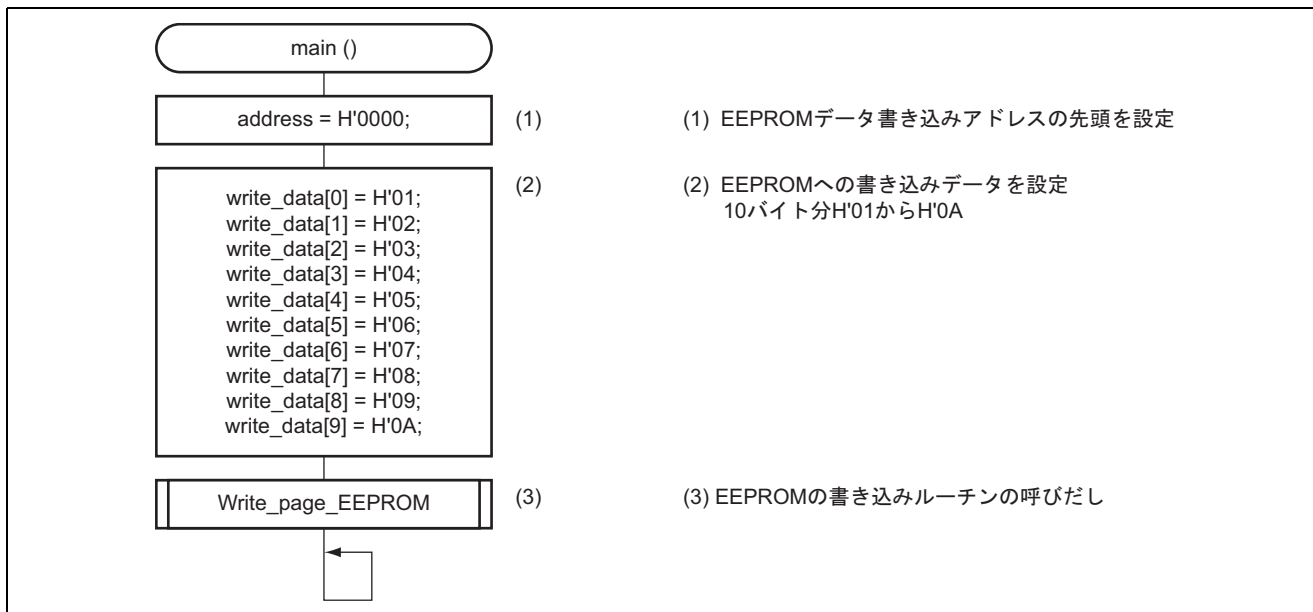
変数	機能	データ長	初期値	使用モジュール名
write_data[0]	1 バイト目送信データ	1 バイト	H'01	メインルーチン
write_data[1]	2 バイト目送信データ	1 バイト	H'02	メインルーチン
write_data[2]	3 バイト目送信データ	1 バイト	H'03	メインルーチン
write_data[3]	4 バイト目送信データ	1 バイト	H'04	メインルーチン
write_data[4]	5 バイト目送信データ	1 バイト	H'05	メインルーチン
write_data[5]	6 バイト目送信データ	1 バイト	H'06	メインルーチン
write_data[6]	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

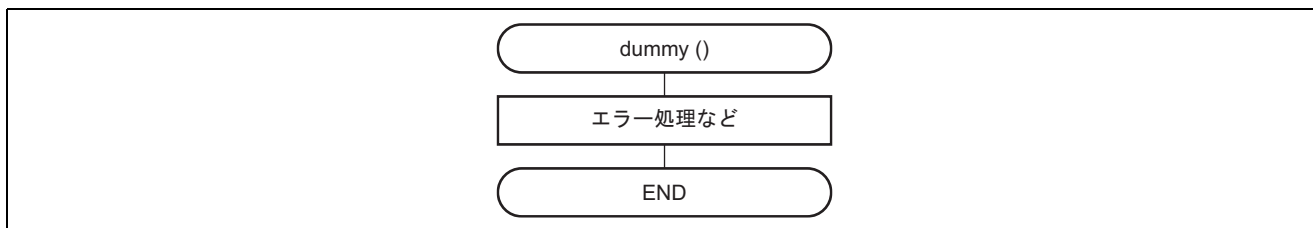
本タスク例では、変数以外の RAM は使用しません。

2.2.4 フローチャート

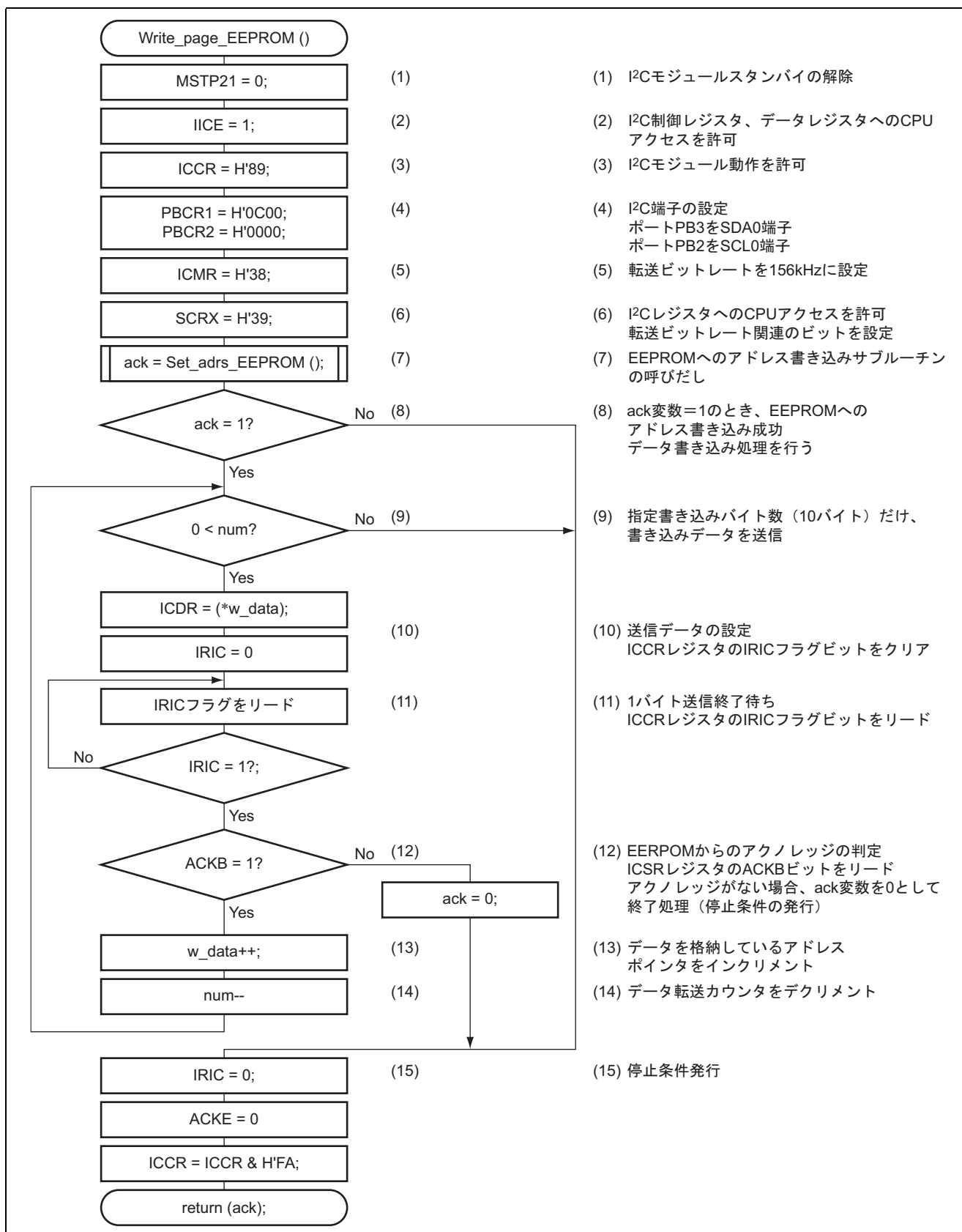
(1) メインルーチン



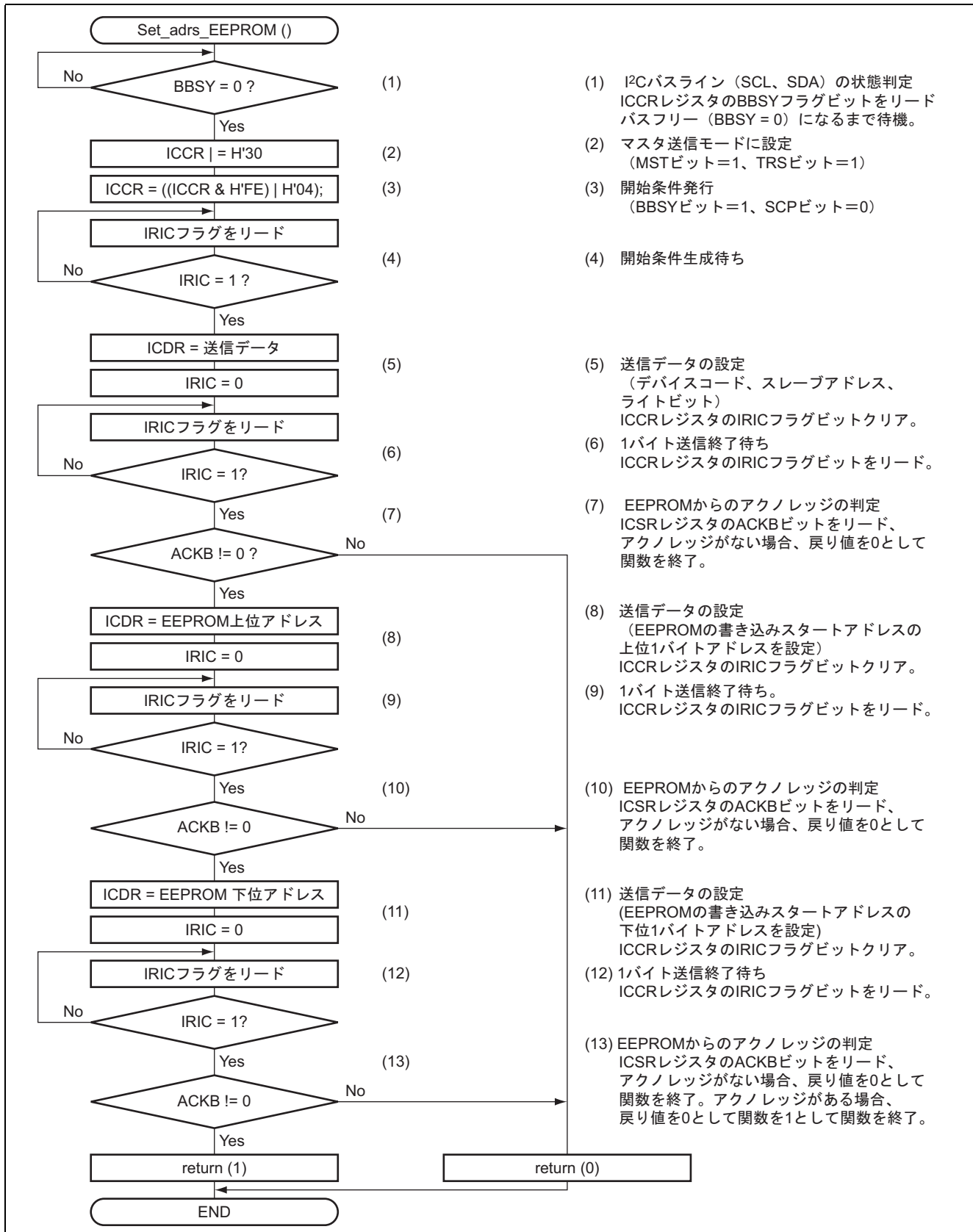
(2) ダミー割り込みルーチン



(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 trasmit
/* 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 "iodef.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 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
    unsigned char write_data[DATA_NUM]; /* write data

    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
/*****
#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 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 1byte 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 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){ /* Test the acknowledge bit
        return (0); /* no ACK
    }
    /* EEPROM upper address Transmission(1byte)
    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 1byte
transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){               /* Test the acknowledge bit
        return (0);                             /* no ACK
    }
    /* EEPROM lower address Transmission(1byte)
    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 1byte
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 I²C 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 I²C 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)I²C 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 14 に SH7145F と EEPROM の接続例を示します。

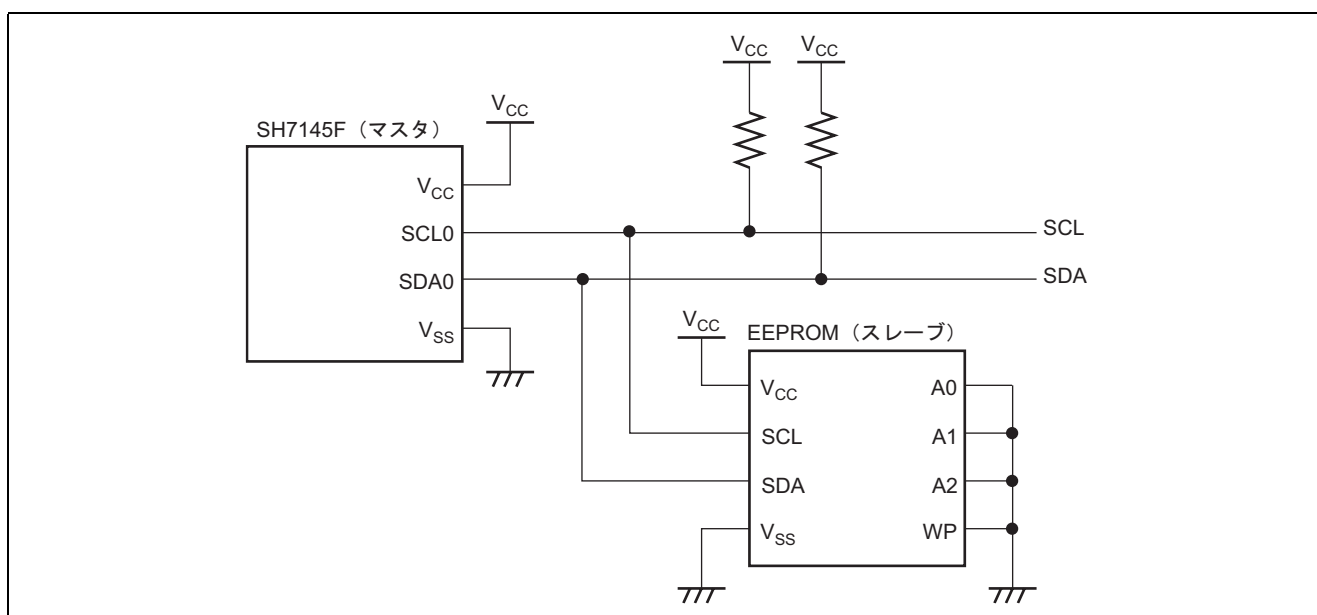


図 14 SH7145F と EEPROM との接続例

本タスクで例で使用する I²C バスフォーマットを図 15 に示します。

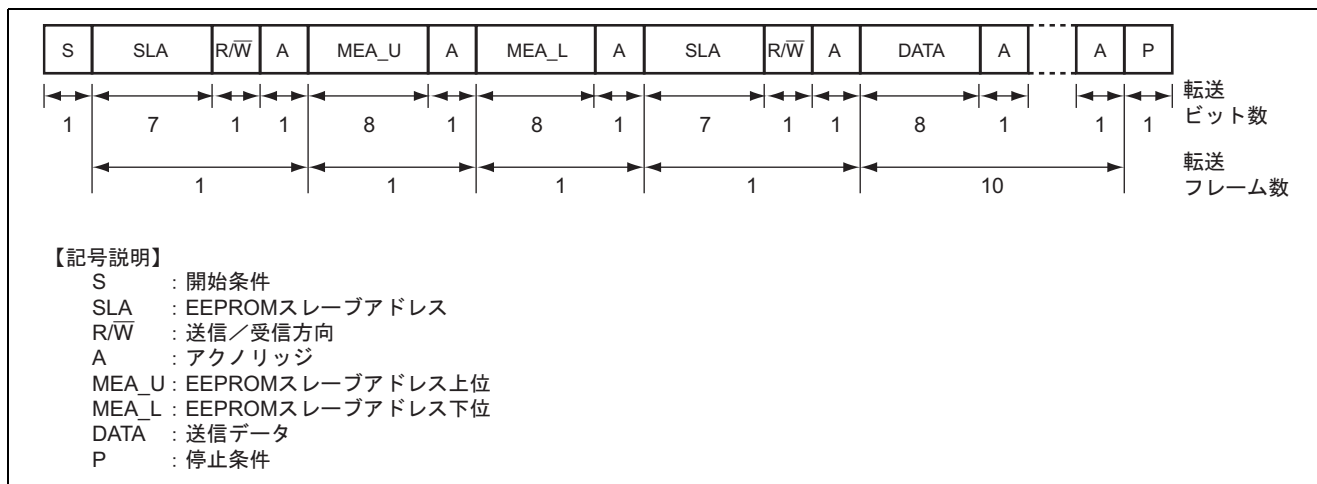
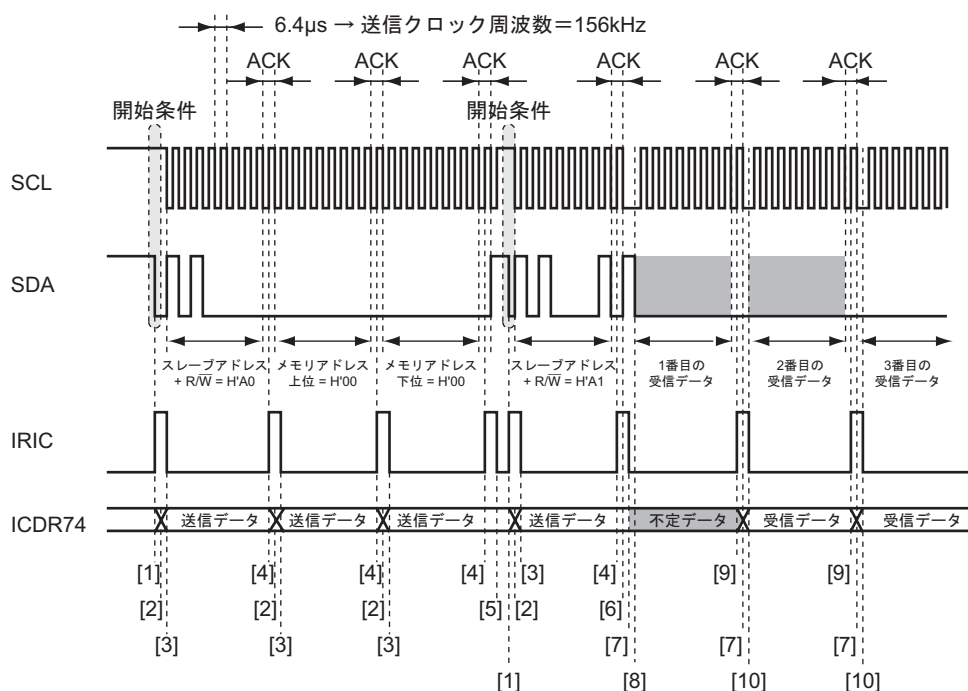


図 15 本タスクで使用する転送フォーマット

2.3.2 動作説明

図 16 に動作の概要を示します。



	ソフトウェア	ハードウェア
[1]	開始条件の発行 (BBSY=1、SCP=0)	IRIC=1、IRTR= 1 (フラグがセット)
[2]	ICDRに送信データをライト	処理なし
[3]	処理なし	データ送信の終了でIRIC=1 (送信クロックの9クロック目の立ち上がり時)
[4]	送信終了判定のためにIRICを0にクリア	処理なし
[5]	停止条件の発行 (BBSY=0、SCP=0)	処理なし
[6]	マスタ受信モードに設定 (MST=1、TRS=0、HNDS=1、WAIT=0)	処理なし
[7]	ICIRフラグを0クリア	処理なし
[8]	ICDR0をダミーリード	受信開始
[9]	処理なし	SDAをLowにしてアクノレッジを返信 IRIC=1 (受信クロックの9クロック目の立ち上がり)
[10]	ICDRから受信データをリード	受信開始

図 16 シングルマスタ受信動作

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示します。

表 4 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 ² 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 5 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 ² 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 ² C の SDA0 入出力端子に設定 ポート B2(PB2)の端子機能を I ² C の SCL0 入出力端子に設定	H'FFFF839A	H'0000
ICDR		I ² 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	-
SAR		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 ² C バスにつながっている他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 ² C バスフォーマット。	ビット 0	
SARX		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 ² C バスにつながっている他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 ² C バスフォーマット。	ビット 0	

レジスタ名 ビット	機能	アドレス	設定値
		ビット	
ICMR	I ² 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 ² 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	
ICCR	I ² C バスコントロールレジスタ	H'FFFF8808	H'89
ICE	I ² C バスインタフェースイネーブル (ICE) ICE=B'1 のとき、I ² 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 ² 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	2C 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	
ICSR	I ² 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 ² 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	

レジスタ名 ビット	機能	アドレス	設定値
		ビット	
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 ² C トランスファレートセレクト 0 ICMR の CKS[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 ² C マスタイネーブル IICE=B'1 のとき、I ² 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

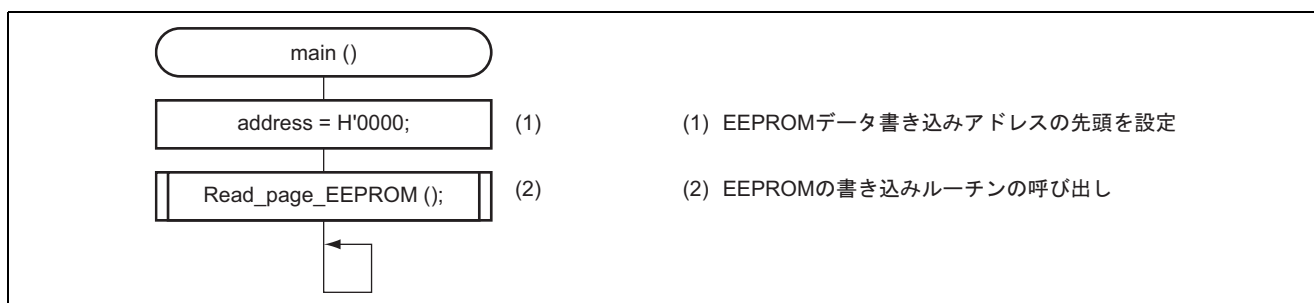
変数	機能	データ長	初期値	使用モジュール名
read_data[0]	1 バイト目受信データ	1 バイト	-	メインルーチン
read_data[1]	2 バイト目受信データ	1 バイト	-	メインルーチン
read_data[2]	3 バイト目受信データ	1 バイト	-	メインルーチン
read_data[3]	4 バイト目受信データ	1 バイト	-	メインルーチン
read_data[4]	5 バイト目受信データ	1 バイト	-	メインルーチン
read_data[5]	6 バイト目受信データ	1 バイト	-	メインルーチン
read_data[6]	7 バイト目受信データ	1 バイト	-	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

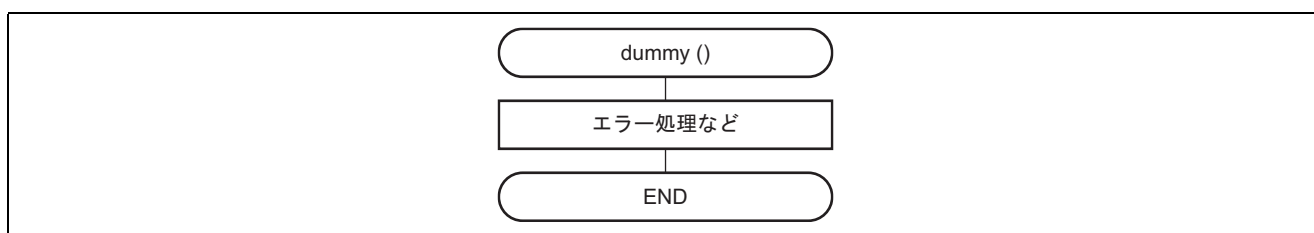
本タスク例では、変数以外の RAM は使用しません。

2.3.4 フローチャート

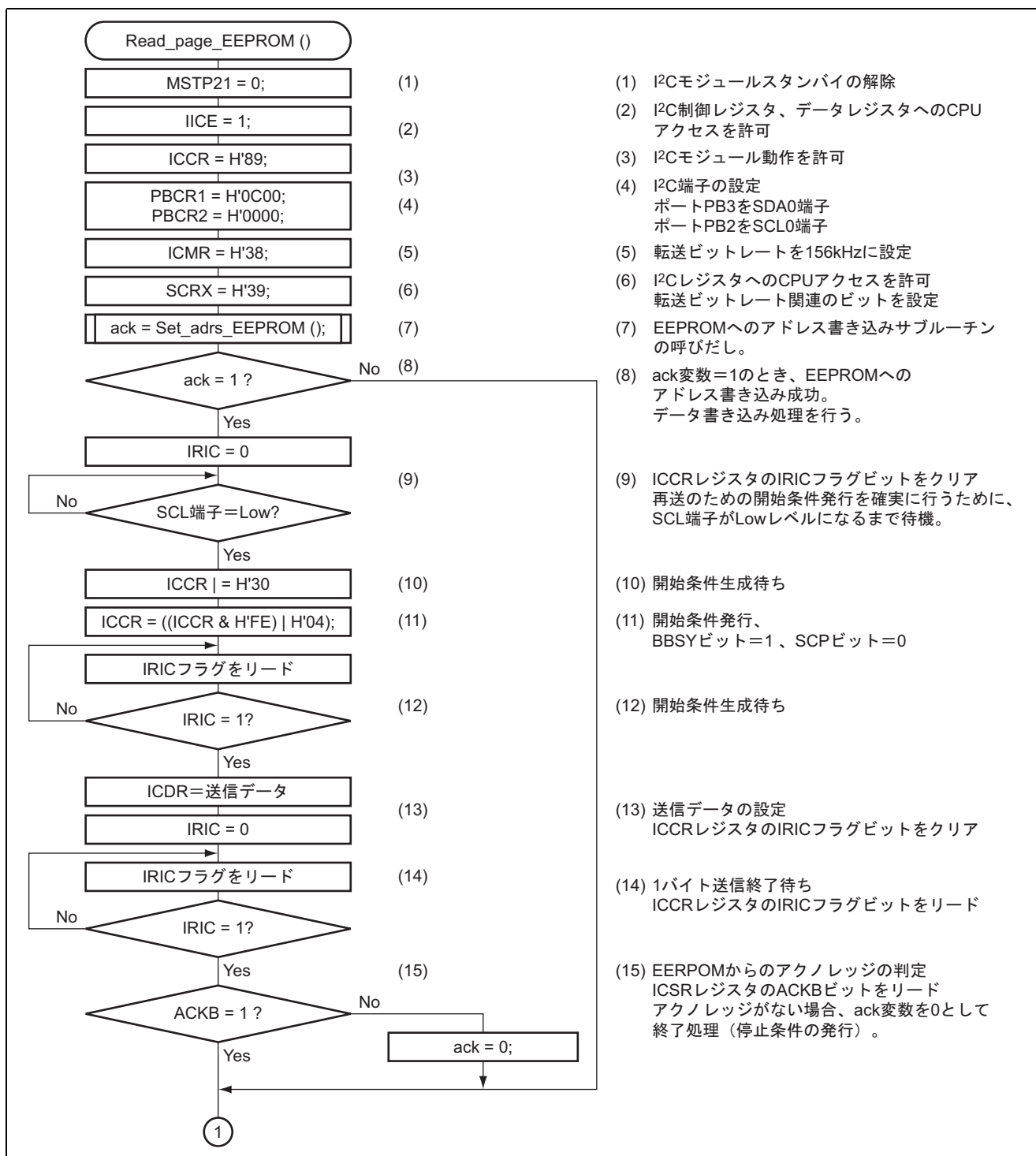
(1) メインルーチン

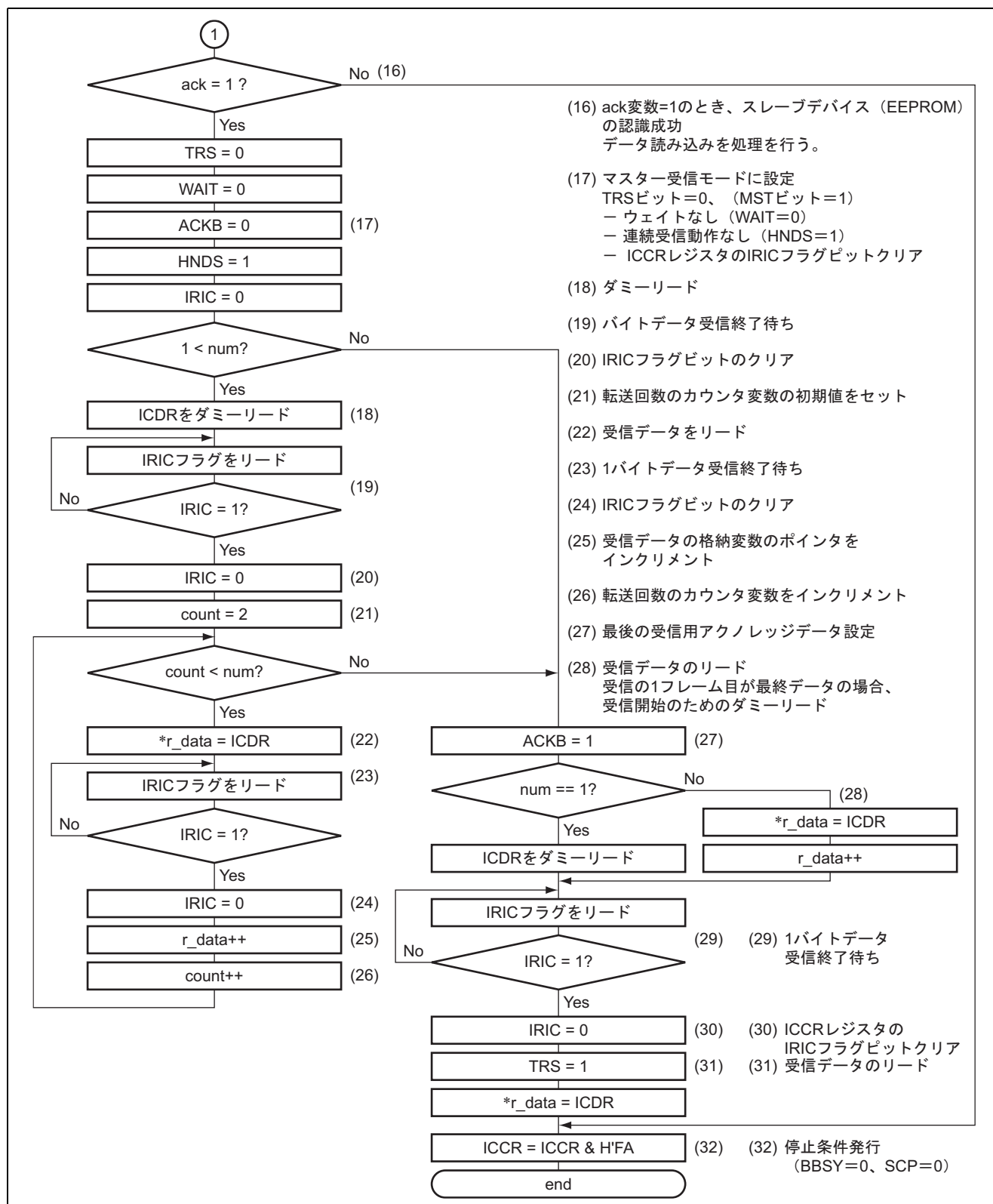


(2) ダミー割り込みルーチン

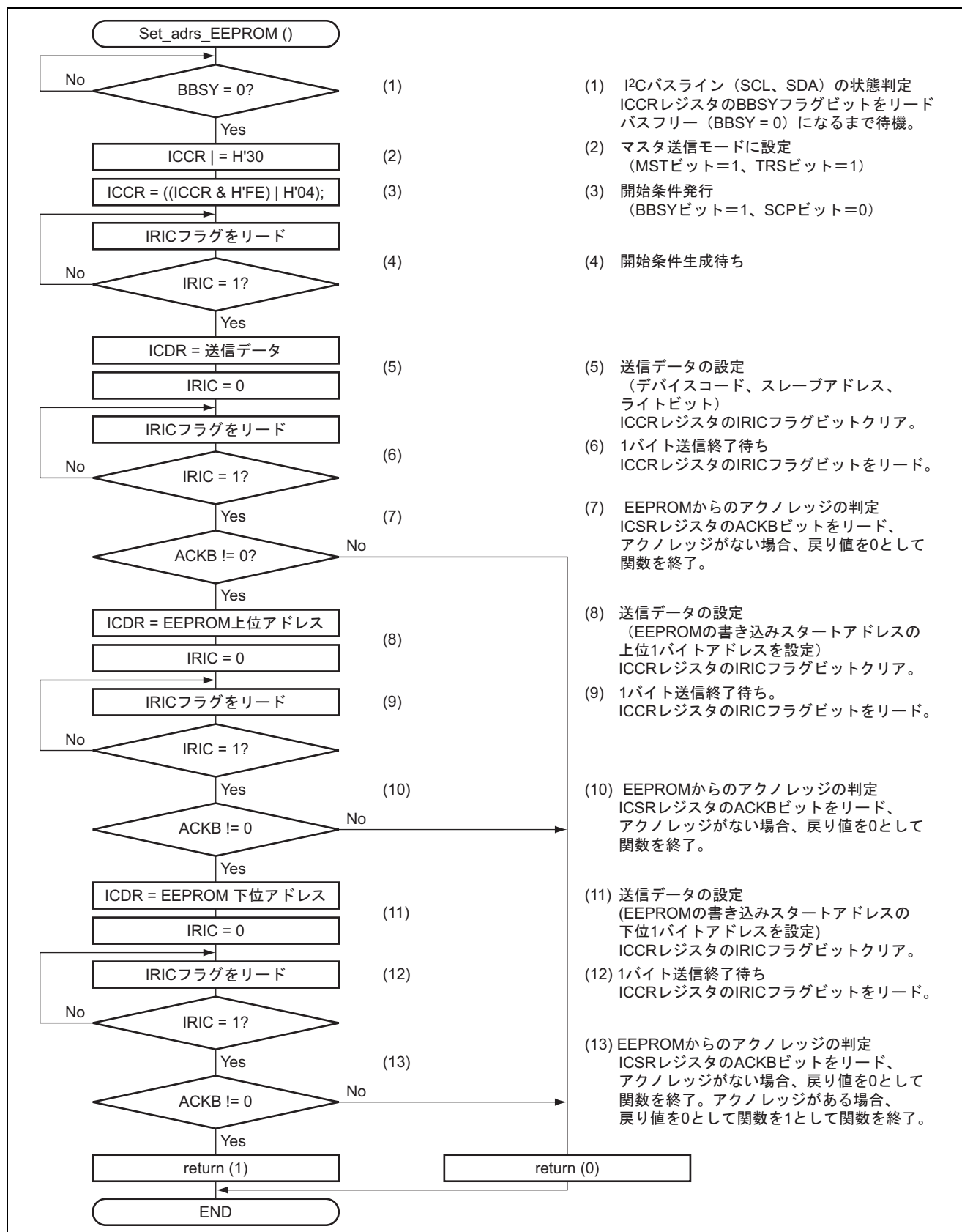


(3)EEPROM 読み出しサブルーチン





(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 interrput 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-fai10MHz,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

/* 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.TRS = 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.TRS = 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 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){           /* Test the acknowledge bit
        return (0);                        /* no ACK
    }
    /* EEPROM upper address Transmission(1byte)
    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 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){           /* Test the acknowledge bit
        return (0);                        /* no ACK
    }
    /* EEPROM lower address Transmission(1byte)
    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 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){           /* Test the acknowledge bit
        return (0);                        /* no ACK
    }
    return (1);                            /* ACK OK
}

```

2.4 EEPROM のバス状態の初期化

2.4.1 仕様

- (1)SH7145F のポートを使用して開始条件発行し、ダミーのスレーブアドレス [1111111] を送信後、停止条件を発行して強制的に EEPROM (HN58X2464、64k bit ,8word×8bit) のバス状態の初期化を行います。
(この初期化処理は、EEPROM からのデータ受信中に通信が中断した等の理由で、EEPROM の SDA バスが出力状態のままになっていて、マスターデバイスからの受信処理ができないときに、強制的に、EEPROM の SDA バスを入力状態にするために使用します)
- (2)本システムの I²C 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。
- (3)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。
- (4)図 17 に SH7145F と EEPROM の接続例を示します。

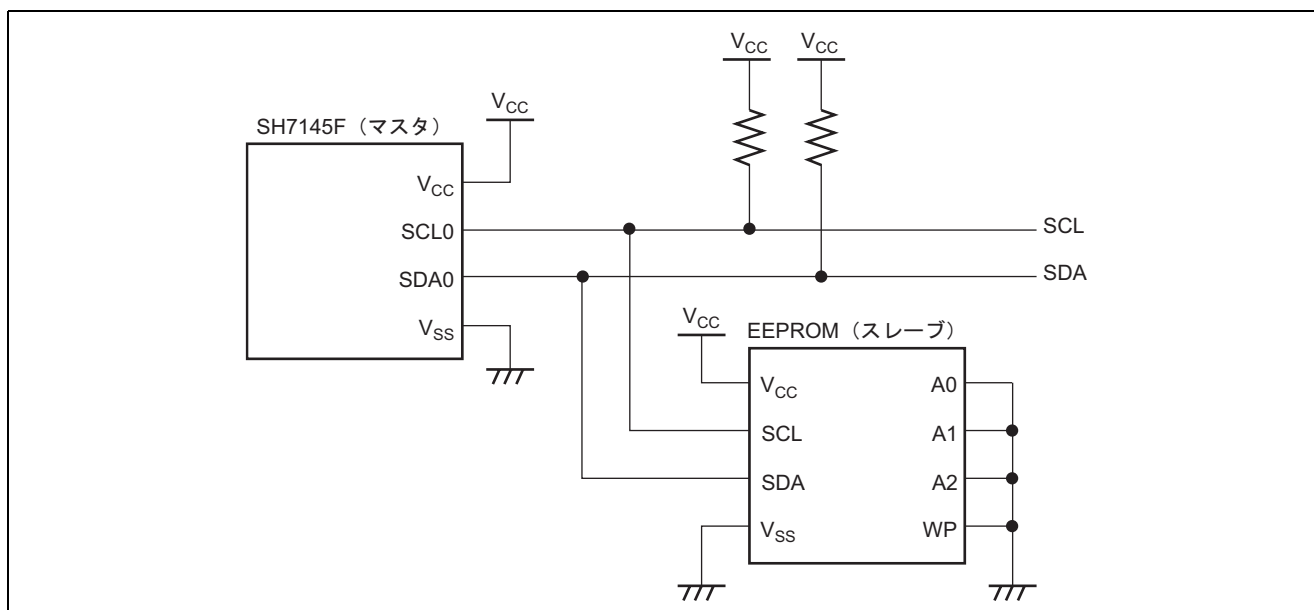


図 17 SH7145F と EEPROM との接続例

本タスクで例で使用する I²C バスフォーマットを図 18 示します。

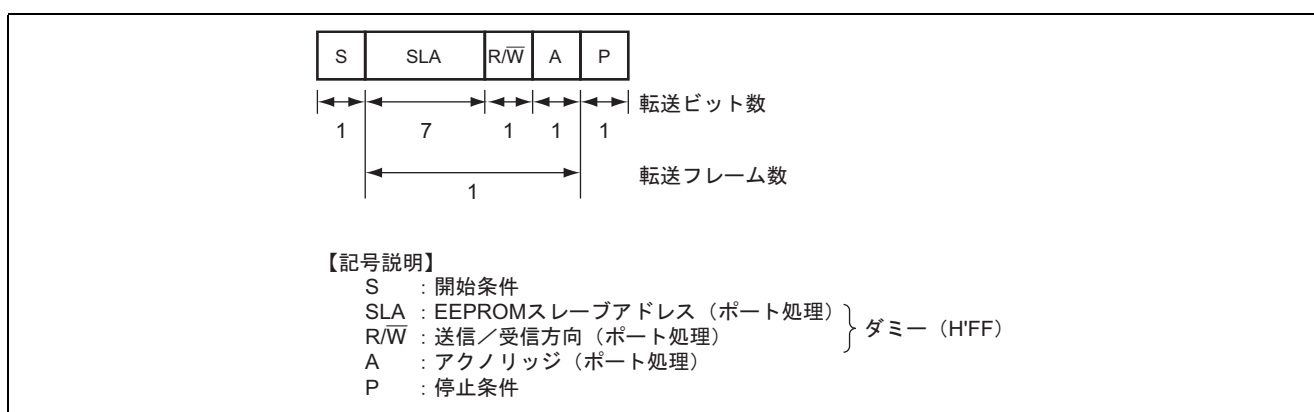


図 18 本タスクで使用する転送フォーマット

2.4.2 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 6 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 ² 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
I ² C バスインタフェース チェックルーチン	Check_i2c	I ² C I/F モジュールの動作状態を機能停止状態 (SCL0/SDA0 端子は、ポート機能) に設定
開始条件発行ルーチン	I2c_start	I ² C デバイスに開始条件を発行 (ポート処理)
バイトデータ送信ルーチン	I2c_bytesend	I ² C デバイスにバイトデータを送信 (ポート処理)
アクノリッジルーチン	I2c_ackck	I ² C デバイスからアクノリッジを得る (ポート処理)
停止条件発行ルーチン	I2c_stop	I ² C デバイスに停止条件を発行 (ポート処理)
ウェイトタイマルーチン	Wait_timer	ウェイトタイマ

(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 7 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 ² 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)の端子機能をポート機能 (PB3) に設定 ポート B2(PB2)の端子機能をポート機能 (PB2) に設定	H'FFFF839A	H'0000
PBDR	ポート B コントロールレジスタ 読み出し／書き込み可能な 16 ビットのレジスタでポート B を格納。端子機能が汎用出力の場合には、PBDR に値を書き込むと端子からその値が出力されます。	H'FFFF8390	H'000C
PBIOR	ポート B・IO レジスタ 読み出し／書き込み可能な 16 ビットのレジスタで、ポート B にある端子の入出力方向を選択	H'FFFF8394	H'000C

レジスタ名 ビット	機能	アドレス	設定値
		ビット	
ICCR	I ² C バスコントロールレジスタ	H'FFFF8808	H'89
ICE	I ² C バスインタフェースイネーブル (ICE) ICE=B'1 のとき、I ² 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 ² 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 ² 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	

(3)変数説明

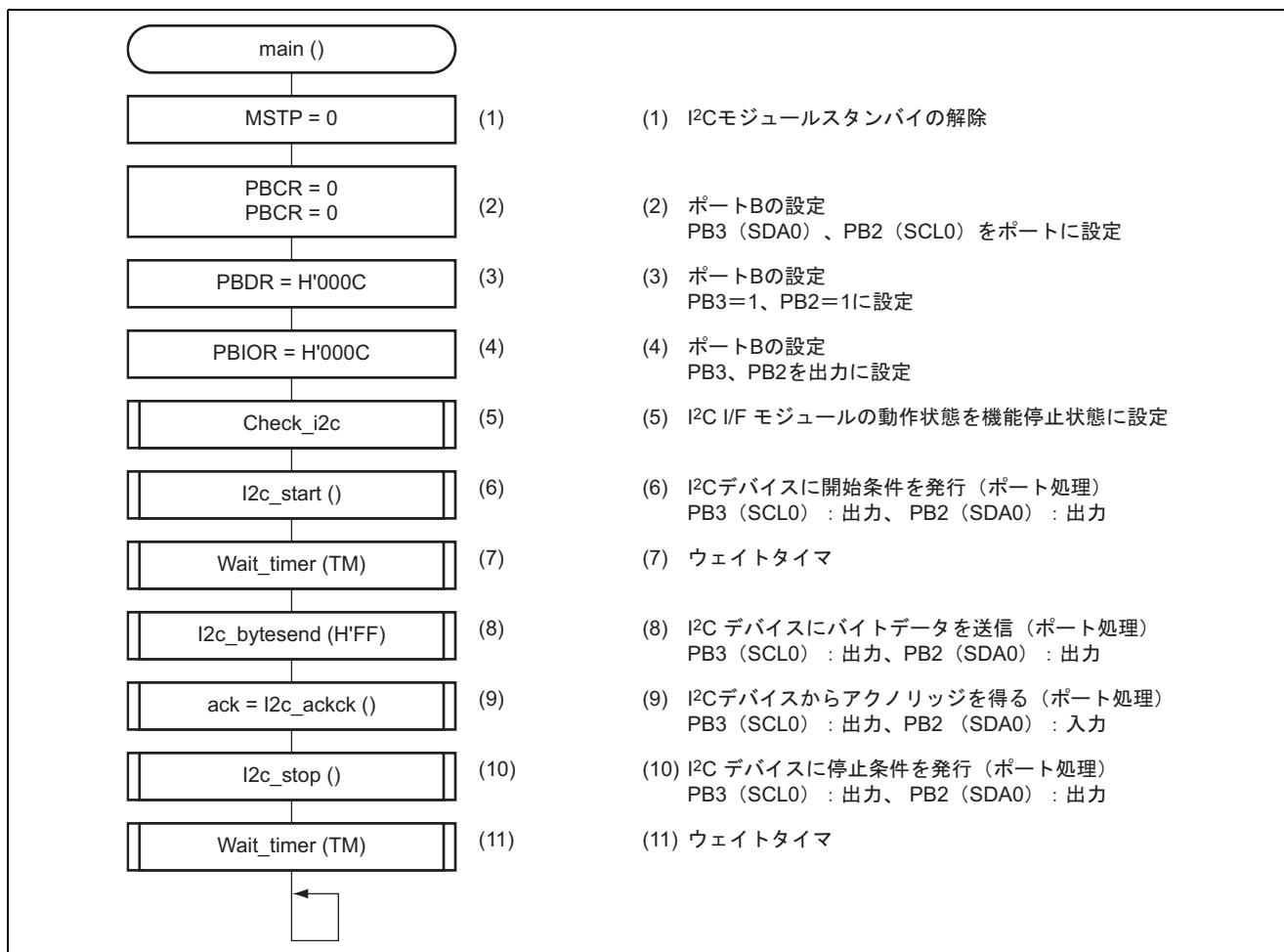
変数	機能	データ長	初期値	使用モジュール名
ack	アクノレッジ判定フラグ	1 バイト	-	main
cnt	ウェイトタイマカウンタ	2 バイト	0	Wait_timer
Ckbit	送信データチェック	1 バイト	H'80	I2c_bytesend
Ack_flag	アクノレッジ判定フラグ	1 バイト	-	I2c_ackck
Data	送信ビットデータチェック	1 バイト	0	I2c_set
Bit_data	送信ビットデータ	1 バイト	-	I2c_bitsend

(4)使用 RAM 説明

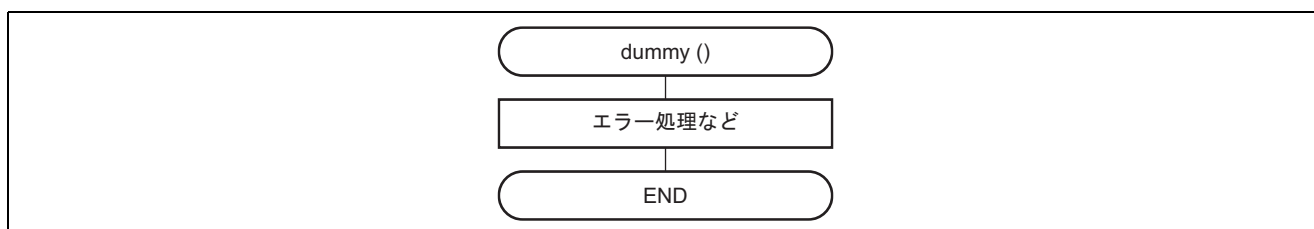
本タスク例では、変数以外の RAM は使用しません。

2.4.3 フローチャート

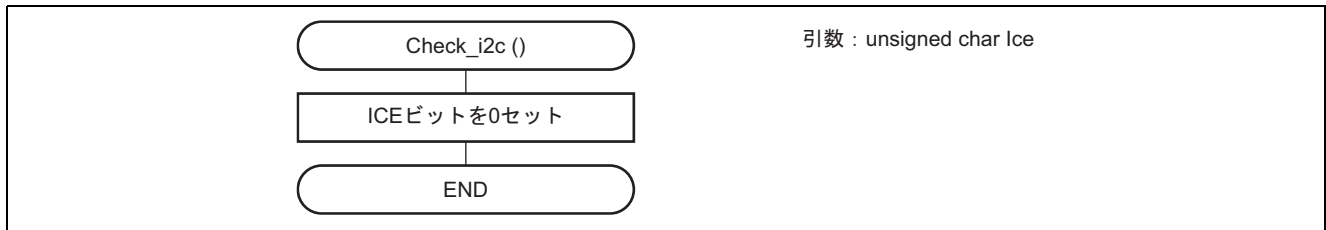
(1) メインルーチン



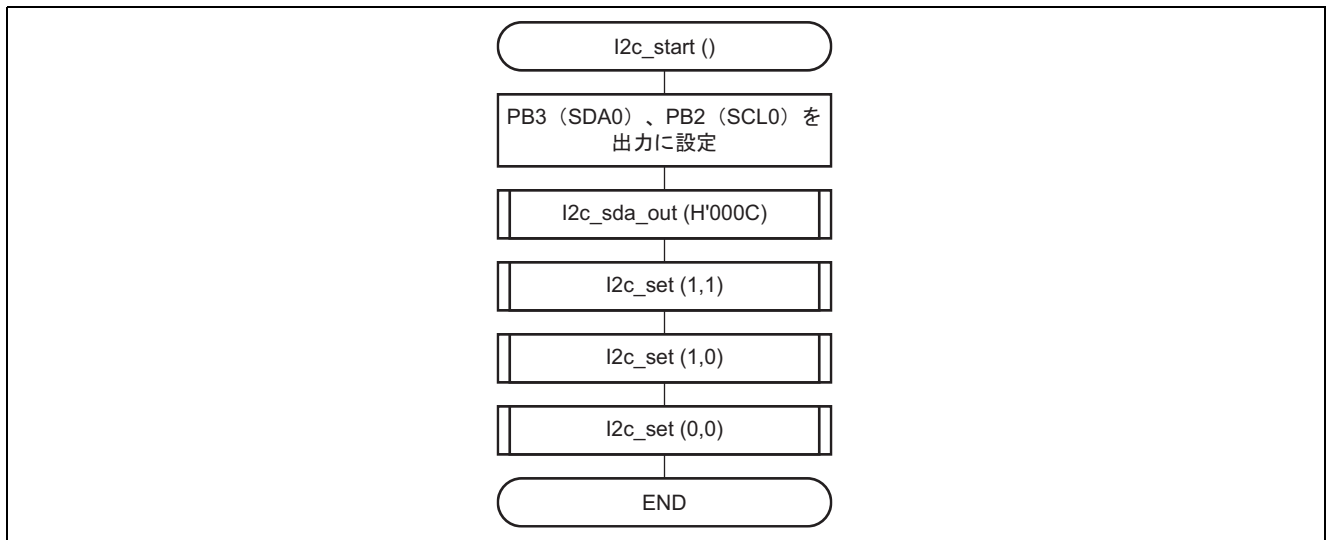
(2) ダミー割り込みルーチン



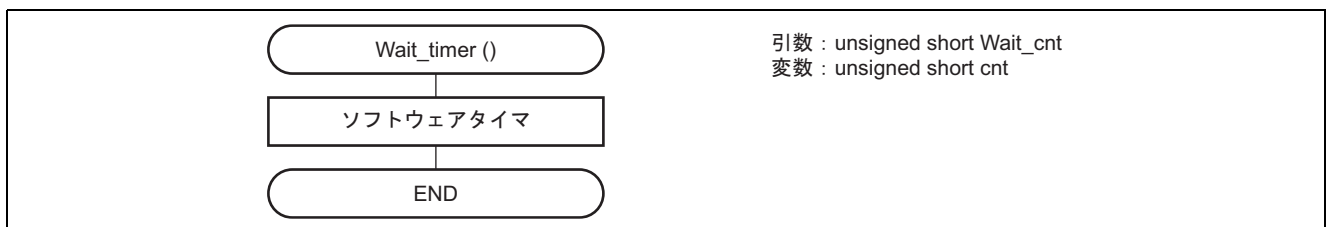
(3) Check_i2c ルーチン



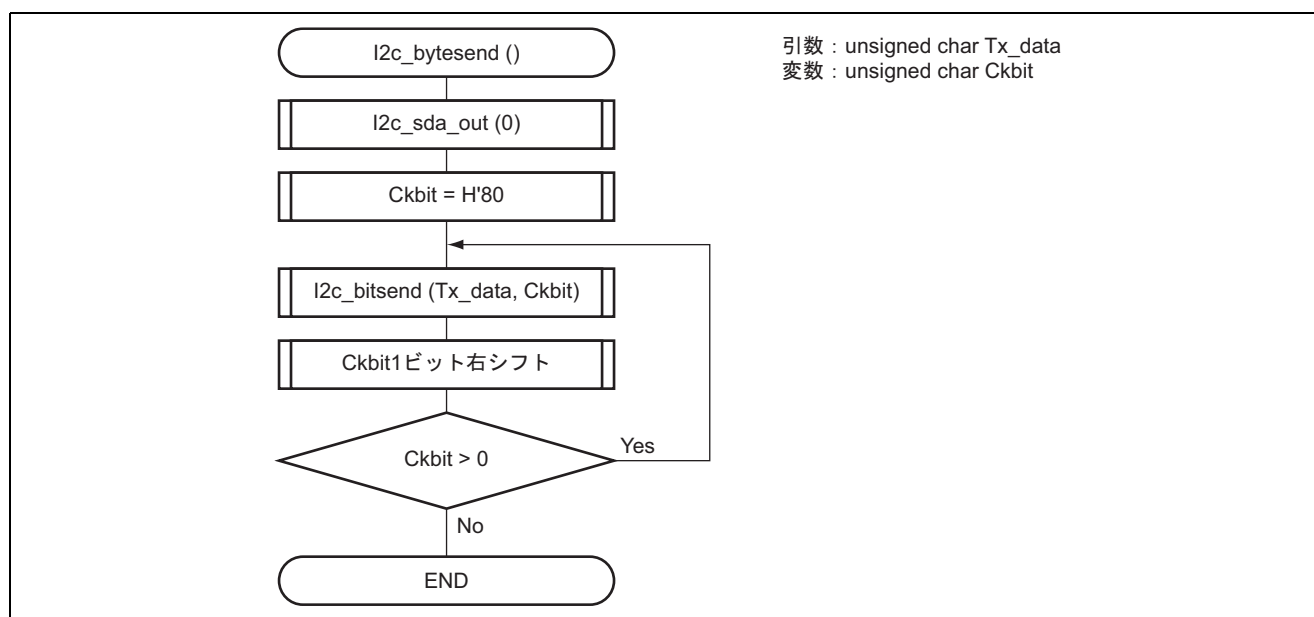
(4) I2c_start ルーチン



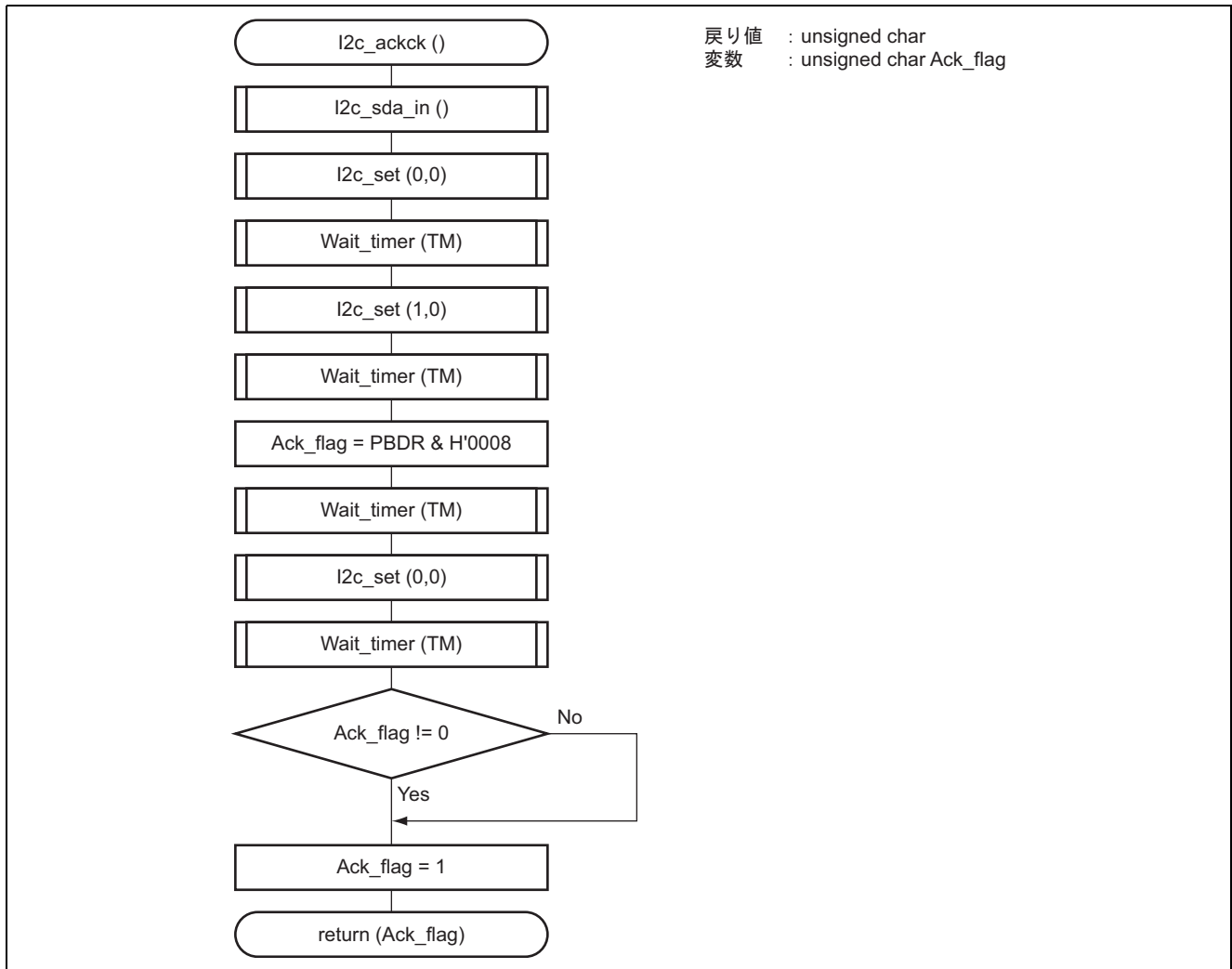
(5) Wait_timer ルーチン



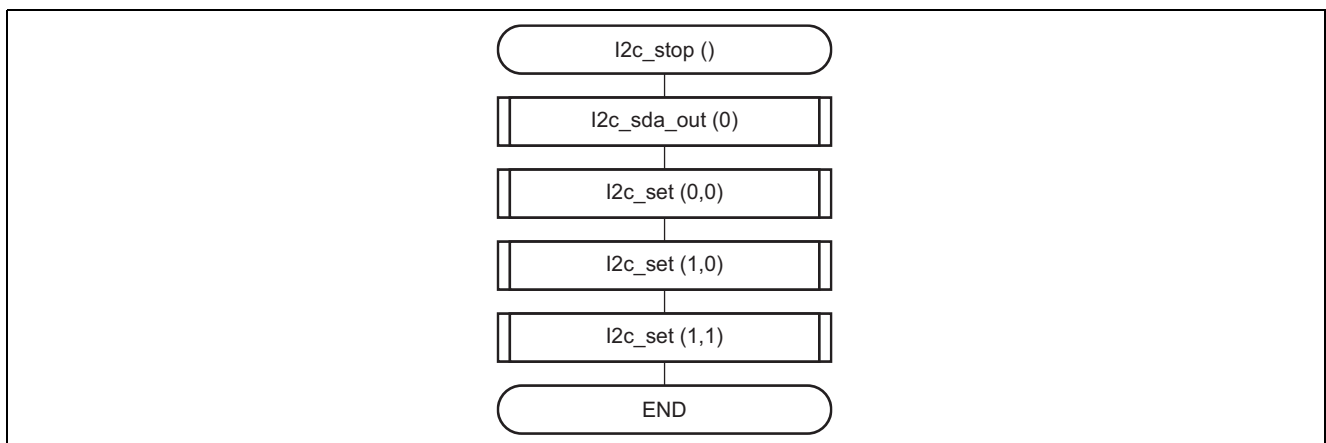
(6) I2c_bytesend ルーチン



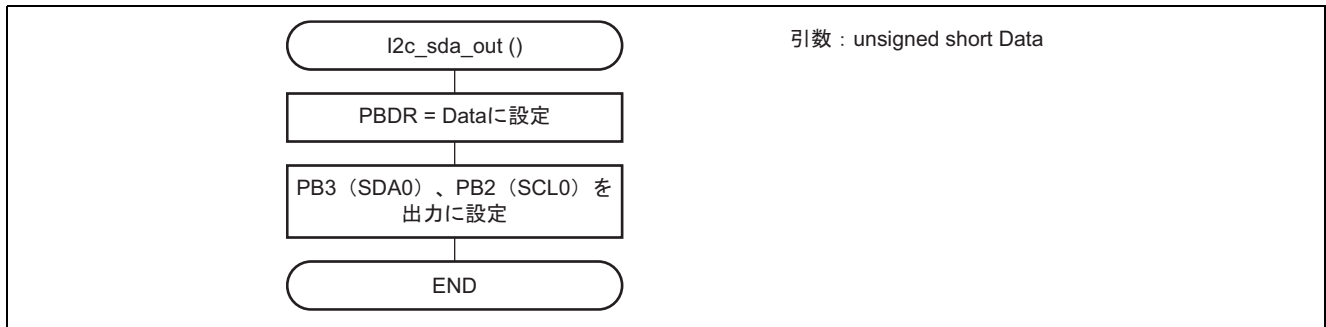
(7) I2c_ackck ルーチン



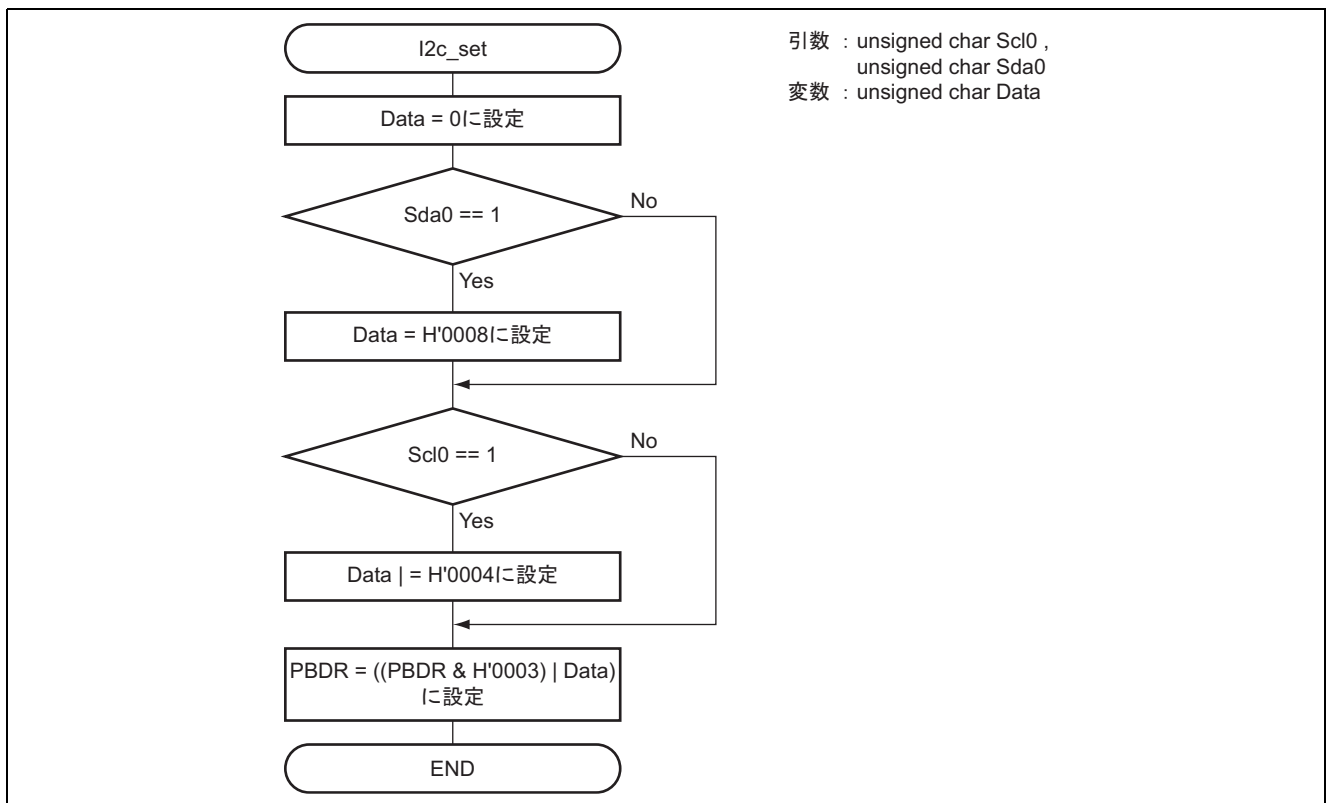
(8) I2c_stop ルーチン



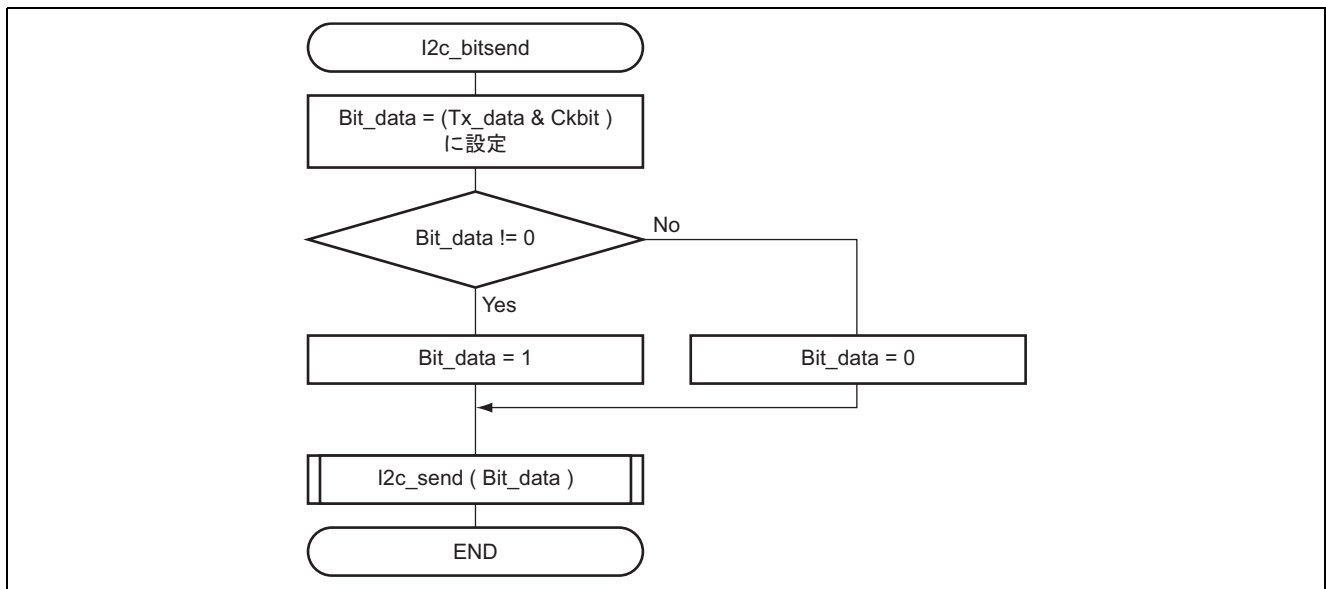
(9) I2c_sda_out ルーチン



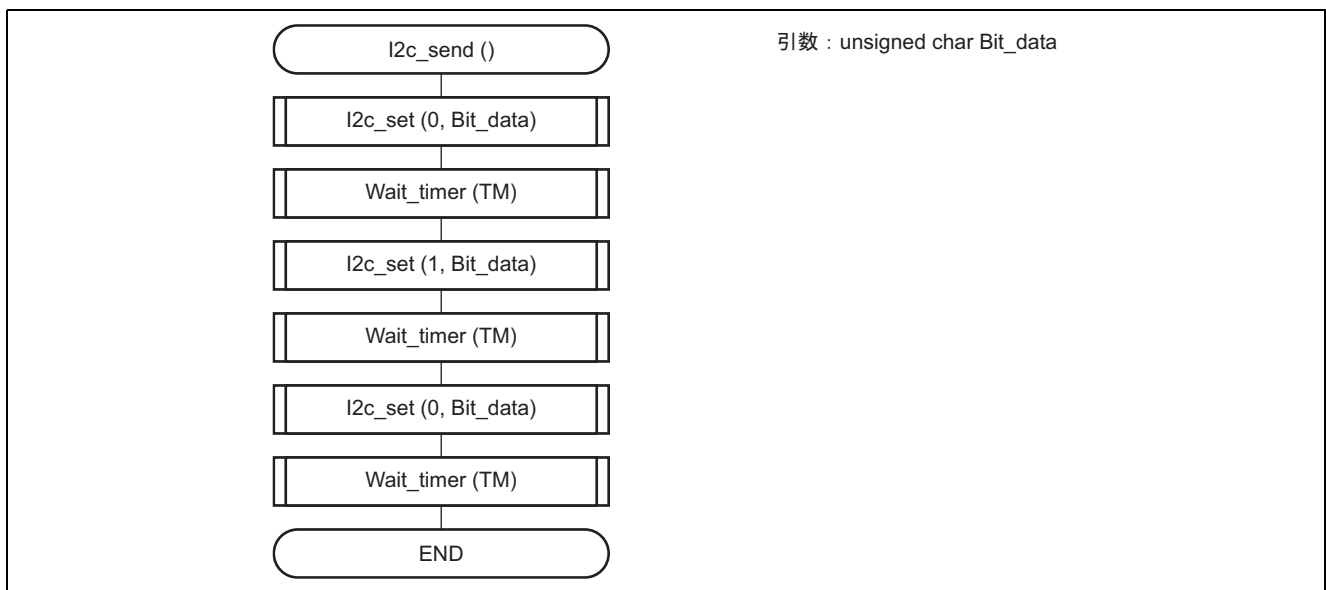
(10) I2c_set ルーチン



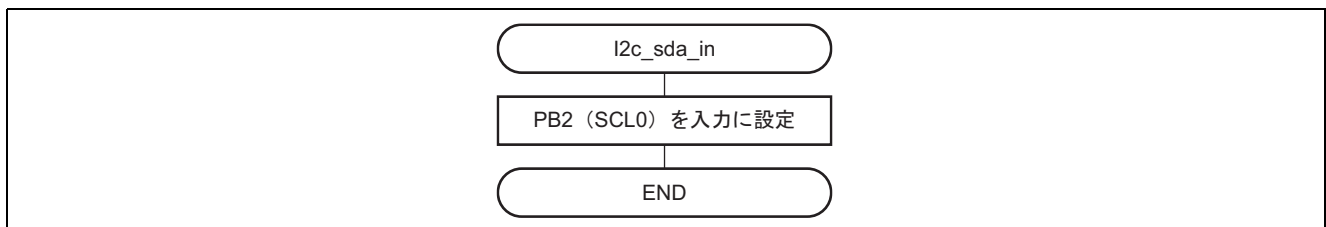
(11)I2c_bit send ルーチン



(12)I2c_send ルーチン



(13)I2c_sda_in ルーチン



2.4.4 プログラムリスト

```

/*****
/* SH7144F Series -SH7145- I2C-bus Application Note
/* Initialization of the bus state of EEPROM
/*   Clock :CPU=40MHz      (External input=10MHz)
/*   :Peripheral=40MHz
/*   Written :2004/3/24 Rev.2.1
*****/
#include <machine.h>
#include "iodefine.h"

/*----- Symbol Definition -----
#define TM      200      /* wait timer      */

/*----- Function Definition -----
void main( void );
void Check_i2c ( unsigned char Ice );
void I2c_start ( void );
void I2c_sda_out ( unsigned short Data );
void I2c_set ( unsigned char Scl , unsigned char Sda );
void Wait_timer ( unsigned short Wait_cnt );
void I2c_bytesend ( unsigned char Tx_data );
void I2c_bitsend ( unsigned char Tx_data , unsigned char Ckbit );
void I2c_send ( unsigned char Bit_data );
unsigned char I2c_ackck ( void );
void I2c_sda_in ( void );
void I2c_stop ( void );
void dummy(void);

/*****
/*   main
*****/
void main( void ){

    unsigned char ack;

    P_STBY.MSTCR1.BIT.MSTP21 = 0; /* disable I2C standby mode
    P_PORTB.PBCR1.WORD = 0;
    P_PORTB.PBCR2.WORD = 0;      /* PBmode =>PB3,PB2 select
    P_PORTB.PBDR.WORD = 0x000c; /* PB3(SDA0)=1,PB2(SCL0)=1
    P_PORTB.PBIOR.WORD = 0x000c; /* PB3(SDA0),PB2(SCL0) output
    Check_i2c (0);              /* IIC I/F module outage
                                /* (SDA0/SCL0=>PORT)set
    I2c_start ();               /* issue start-up condition (PORT processing)
    Wait_timer (TM);           /* lay over
    I2c_bytesend ( 0xff );     /* slave adress(dummy data) transmission (PORT
processing)
    ack = I2c_ackck();         /* acknowledge signal(PORT processing)
    I2c_stop ();              /* issue end condition (PORT processing)
    Wait_timer (TM);          /* lay over

    while(1);
}

```

```

/*****
/* Check_i2c
/*      argument1      ;I2C bus I/F check (unsigned char)
/*      --0:outage(SDA0/SCL0=>PORT)
/*      --1:driving(SDA0/SCL0=>bus drive)
/*      return          ;none
/*****
void Check_i2c ( unsigned char Ice ){
    Ice &= 0x01;
    if ( Ice == 1 ){
        Ice = 0x80;
    }else{
        Ice = 0x00;
    }
    P_IIC.ICCR0.BYTE = ((P_IIC.ICCR0.BYTE & 0x7f) | Ice); /* ICE bit set
}

/*****
/* I2c_start
/*      argument1      ;none
/*      return          ;none
/*****
void I2c_start ( void ){

    I2c_sda_out( 0x000c ); /* PB3(SDA0),PB2(SCL0) output
                          /* PB3(SDA0)=1,PB2(SCL0)=1
    I2c_set (1,1); /* PB2(SCL0)=1 , PB3(SDA0)=1 output
    I2c_set (1,0); /* PB2(SCL0)=1 , PB3(SDA0)=0 output
    I2c_set (0,0); /* PB2(SCL0)=0 , PB3(SDA0)=0 output
}

/*****
/* Wait_timer
/*      argument1      ;wait counter (unsigned short)
/*      return          ;none
/*****
void Wait_timer ( unsigned short Wait_cnt ){

    unsigned short cnt;

    for (cnt = 0; cnt < Wait_cnt; cnt ++){
    }
}

/*****
/* I2c_bytesend
/*      argument1      ;Transmit Data(unsigned char)
/*      return          ;none
/*****
void I2c_bytesend ( unsigned char Tx_data ){

    unsigned char Ckbit;

    I2c_sda_out( 0 ); /* PB3(SDA0),PB2(SCL0) output

```

```

        /* PB3(SDA0)=0,PB2(SCL0)=0
for (Ckbit = 0x80; Ckbit > 0; Ckbit >=> 1){          /* 8bit data amount
    I2c_bitsend ( Tx_data , Ckbit ); /* MSB first
}
}

/*****
/* I2c_ackck
/*     argument1    ;none
/*     return       ;Ack_flag(unsigned char)
/*         --0:normal
/*         --1:error
/*****
unsigned char I2c_ackck ( void ){

    unsigned char Ack_flag;

    I2c_sda_in();          /* PB2(SCL0) output,PB3(SDA0) input
    I2c_set (0,0);
    Wait_timer (TM);          /* lay over
    I2c_set (1,0);
    Wait_timer (TM);          /* lay over
    Ack_flag = P_PORTB.PBDR.WORD & 0x0008; /* flag check
    Wait_timer (TM);          /* lay over
    I2c_set (0,0);
    Wait_timer (TM);          /* lay over
    if (Ack_flag !=
        Ack_flag = 1;          /* PB3(SDA0)=1 ==> Ack_flag=1
    }
    return (Ack_flag);
}

/*****
/* I2c_stop
/*     argument1    ;none
/*     return       ;none
/*****
void I2c_stop ( void ){
    I2c_sda_out( 0 );          /* PB3(SDA0),PB2(SCL0) output
                                /* PB3(SDA0)=0,PB2(SCL0)=0
    I2c_set (0,0);          /* PB2(SCL0)=0 , PB3(SDA0)=0 output
    I2c_set (1,0);          /* PB2(SCL0)=1 , PB3(SDA0)=0 output
    I2c_set (1,1);          /* PB2(SCL0)=1 , PB3(SDA0)=1 output
}

/*****
/* I2c_sda_out
/*     argument1    ;Data (unsigned short)
/*         --bit3:SDA0
/*         --bit2:SCL0
/*     return       ;none
/*****
void I2c_sda_out ( unsigned short Data ){
    P_PORTB.PBDR.WORD = Data;

```

```

P_PORTB.PBIOR.WORD = 0x000c;    /* PB3,PB2 output
}

/*****
/* I2c_set
/*      argument1    ;Scl0 (unsigned char)
/*          --0: PB2(SCL0)=0
/*          --1: PB2(SCL0)=1
/*      argument2    ;Sda0 (unsigned char)
/*          --0: SDA0(PB3)=0
/*          --1: SDA0(PB3)=1
/*      return        ;none
*****/
void I2c_set ( unsigned char Scl0 , unsigned char Sda0 ){
    unsigned char Data;
    Data = 0;                      /* initialize PB3(SDA0),PB2(SCL0) output value
    if ( Sda0 == 1 ){
        Data = 0x0008;              /* PB3=1
    }
    if ( Scl0 == 1 ){
        Data |= 0x0004;              /* PB2=1
    }
    P_PORTB.PBDR.WORD = ((P_PORTB.PBDR.WORD & 0x0003) | Data); /*
PB3(SDA0),PB2(SCL0) set
}

/*****
/* I2c_bit send
/*      argument1    ;Transmit Data(unsigned char)
/*      argument2    ;bit position information of Transmit bit data(unsigned char)
/*      return        ;none
*****/
void I2c_bit send ( unsigned char Tx_data , unsigned char Ckbit ){

    unsigned char Bit_data;

    Bit_data = ( Tx_data & Ckbit );          /* bit position information
    if ( Bit_data != 0 ){                    /* check bit position information
        Bit_data = 1;
    }else {
        Bit_data = 0;
    }
    I2c_send ( Bit_data );                  /* Transmit bit data
}

/*****
/* I2c_send
/*      argument1    ;Transmit bit data(unsigned char)
/*      return        ;none
*****/
void I2c_send ( unsigned char Bit_data ){

    I2c_set ( 0,Bit_data);                  /* PB2(SCL0)=0,PB3(SDA0)=Bit_data output
    Wait_timer (TM);                        /* lay over

```

```

    I2c_set (1, Bit_data);          /* PB2(SCL0)=1, PB3(SDA0)=Bit_data output
    Wait_timer (TM);                /* lay over
    I2c_set (0, Bit_data);          /* PB2(SCL0)=0, PB3(SDA0)=Bit_data output
    Wait_timer (TM);                /* lay over
}

/*****
/* I2c_sda_in
/*     argument1    ;none
/*     return       ;none
*****/
void I2c_sda_in ( void ){
    P_PORTB.PBIOR.WORD = 0x0004;    /* PB2(SCL0) output, PB3(SDA0) input
}

#pragma interrupt(dummy)
void dummy(void){
}

```

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 Renesas Project Generator (Ver.1.2). */
/*                                     */
/*****/

/*****/

/*      7144 Include File              */
/*****/

struct st_sci0 {                      /* struct SCI0 */
    union {                          /* SMR_0 */
        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_0;                         /* */
    unsigned char BRR_0;             /* BRR_0 */
    union {                          /* SCR_0 */
        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_0;                         /* */
    unsigned char TDR_0;             /* TDR_0 */
    union {                          /* SSR_0 */
        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_0;                         /* */
    unsigned char RDR_0;             /* RDR_0 */
    union {                          /* SDCR_0 */
        unsigned char BYTE;          /* Byte Access */
        struct {                    /* Bit Access */
            unsigned char :4;        /* */
            unsigned char DIR:1;     /* DIR */
            unsigned char :3;        /* */
        } BIT;                      /* */
    } SDCR_0;                       /* */
};                                  /* */

struct st_sci1 {                   /* struct SCI1 */
    union {                        /* SMR_1 */
        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_1;                          /*
unsigned char BRR_1;                  /* BRR_1
union {                              /* SCR_1
    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_1;                          /*
unsigned char TDR_1;                  /* TDR_1
union {                              /* SSR_1
    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_1;                          /*
unsigned char RDR_1;                  /* RDR_1
union {                              /* SDCR_1
    unsigned char BYTE;              /* Byte Access */
    struct {                          /* Bit Access */

```

```

        unsigned char :4;                /*          */
        unsigned char DIR:1;             /*   DIR    */
        unsigned char :3;                /*          */
    } BIT;                                /*          */
} SDCR_1;                                /*          */
};                                        /*          */
struct st_sci2 {                          /* struct SCI2 */
    union {                               /* SMR_2      */
        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_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 {                    /* struct SCI3 */
    union {                          /* SMR_3 */
        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 {                          /* SCR_3 */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char TIE:1;      /* TIE */
            unsigned char RIE:1;      /* RIE */
            unsigned char TE:1;       /* TE */
        } BIT;                       /* */
    } SCR_3;                         /* */
};

```

```

        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 {                           /* SSR_3 */
        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_3;                          /* */
    unsigned char RDR_3;              /* RDR_3 */
    union {                           /* SDCR_3 */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char :4;         /* */
            unsigned char DIR:1;      /* DIR */
            unsigned char :3;         /* */
        } BIT;                       /* */
    } SDCR_3;                         /* */
};                                    /* */

struct st_mtu34 {                    /* struct MTU34 */
    union {                           /* TCR_3 */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char CCLR:3;     /* CCLR */
            unsigned char CKEG:2;     /* CKEG */
            unsigned char TPSC:3;     /* TPSC */
        } BIT;                       /* */
    }

```

```

    } TCR_3; /* */
union { /* TCR_4 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char CCLR:3; /* CCLR */
        unsigned char CKEG:2; /* CKEG */
        unsigned char TPSC:3; /* TPSC */
    } BIT; /* */
} TCR_4; /* */
union { /* TMDR_3 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char :2; /* */
        unsigned char BFB:1; /* BFB */
        unsigned char BFA:1; /* BFA */
        unsigned char MD:4; /* MD */
    } BIT; /* */
} TMDR_3; /* */
union { /* TMDR_4 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char :2; /* */
        unsigned char BFB:1; /* BFB */
        unsigned char BFA:1; /* BFA */
        unsigned char MD:4; /* MD */
    } BIT; /* */
} TMDR_4; /* */
union { /* TIORH_3 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char IOB:4; /* IOB */
        unsigned char IOA:4; /* IOA */
    } BIT; /* */
} TIORH_3; /* */
union { /* TIORL_3 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char IOD:4; /* IOD */
    } BIT; /* */
} TIORL_3; /* */

```

```

        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        */
    } BIT;                             /*              */
}

```

```

        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;                /*           */
        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; /* */
}; /* */

struct st_mtu0 { /* struct MTU0 */
    union { /* TCR_0 */
        unsigned char BYTE; /* Byte Access */
        struct { /* Bit Access */
            unsigned char CCLR:3; /* CCLR */
            unsigned char CKEG:2; /* CKEG */
            unsigned char TPSC:3; /* TPSC */
        } BIT; /* */
    } TCR_0; /* */
    union { /* TMDR_0 */
        unsigned char BYTE; /* Byte Access */
        struct { /* Bit Access */
            unsigned char :2; /* */
            unsigned char BFB:1; /* BFB */
            unsigned char BFA:1; /* BFA */
            unsigned char MD:4; /* MD */
        }
    }

```

```

        } BIT; /* */
    } TMDR_0; /* */
union { /* TIORH_0 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char IOB:4; /* IOB */
        unsigned char IOA:4; /* IOA */
    } BIT; /* */
} TIORH_0; /* */
union { /* TIORL_0 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char IOD:4; /* IOD */
        unsigned char IOC:4; /* IOC */
    } BIT; /* */
} TIORL_0; /* */
union { /* TIER_0 */
    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_0; /* */
union { /* TSR_0 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char :3; /* */
        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_0;                                /* */
unsigned short TCNT_0;                          /* TCNT_0 */
unsigned short TGRA_0;                          /* TGRA_0 */
unsigned short TGRB_0;                          /* TGRB_0 */
unsigned short TGRC_0;                          /* TGRC_0 */
unsigned short TGRD_0;                          /* TGRD_0 */
};                                                /* */
struct st_mtu1 {                                /* struct MTU1 */
    union {                                    /* TCR_1 */
        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_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; /* */
unsigned char wk0[1]; /* */
union { /* TIER_2 */
    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_2; /* */
union { /* TSR_2 */
    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_2; /* */
unsigned short TCNT_2; /* TCNT_2 */
unsigned short TGRA_2; /* TGRA_2 */
unsigned short TGRB_2; /* TGRB_2 */
}; /* */
struct st_intc { /* struct 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 MTU2:8;
    } BIT;
} IPRE;

```

```

        unsigned short MTU3:8;          /* MTU3 */
    } BIT;                             /* */
} IPRE;                               /* */
union {                               /* IPRF */
    unsigned short WORD;               /* Word Access */
    struct {                           /* Bit Access */
        unsigned short MTU4:8;        /* MTU4 */
        unsigned short SCIO:4;        /* SCIO */
        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;                               /*           */
union {                                 /*  IPRJ     */
    unsigned short WORD;                 /*  Word Access */
    struct {                             /*  Bit Access */
        unsigned short :8;               /*           */
        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 {                         /* PADDRH */
        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;                     /* */
    } PADDRH;                       /* */
    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 */
        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 */
        } BIT;                                /* */
    } PACRL2;                                /* */
};                                               /* */
struct st_portb {                               /* struct PORTB */
    union {                                     /* PBDR */
        unsigned short WORD;                    /* Word Access */
        struct {                                /* Bit Access */
            unsigned short :6;                  /* */
            unsigned short PB9DR:1;            /* PB9DR */
            unsigned short PB8DR:1;            /* PB8DR */
            unsigned short PB7DR:1;            /* PB7DR */
            unsigned short PB6DR:1;            /* PB6DR */
            unsigned short PB5DR:1;            /* PB5DR */
            unsigned short PB4DR:1;            /* PB4DR */
            unsigned short PB3DR:1;            /* PB3DR */
            unsigned short PB2DR:1;            /* PB2DR */
            unsigned short PB1DR:1;            /* PB1DR */
            unsigned short PB0DR:1;            /* PB0DR */
            } BIT;                            /* */
        } PBDR;                                /* */
    unsigned char wk0[2];                      /* */
    union {                                     /* PBIOR */
        unsigned short WORD;                    /* Word Access */
        struct {                                /* Bit Access */
            unsigned short :6;                  /* */
            unsigned short PB9IOR:1;            /* PB9IOR */

```

```

        unsigned short PB8IOR:1;          /* PB8IOR */
        unsigned short PB7IOR:1;          /* PB7IOR */
        unsigned short PB6IOR:1;          /* PB6IOR */
        unsigned short PB5IOR:1;          /* PB5IOR */
        unsigned short PB4IOR:1;          /* PB4IOR */
        unsigned short PB3IOR:1;          /* PB3IOR */
        unsigned short PB2IOR:1;          /* PB2IOR */
        unsigned short PB1IOR:1;          /* PB1IOR */
        unsigned short PB0IOR:1;          /* PB0IOR */
    } BIT;                                /* */
} PBIOR;                                /* */
unsigned char wk1[2];                    /* */
union {                                  /* PBCR1 */
    unsigned short WORD;                  /* Word Access */
    struct {                              /* Bit Access */
        unsigned short :4;                /* */
        unsigned short PB3MD2:1;          /* PB3MD2 */
        unsigned short PB2MD2:1;          /* PB2MD2 */
        unsigned short :6;                /* */
        unsigned short PB9MD:2;           /* PB9MD */
        unsigned short PB8MD:2;           /* PB8MD */
    } BIT;                                /* */
} PBCR1;                                /* */
union {                                  /* PBCR2 */
    unsigned short WORD;                  /* Word Access */
    struct {                              /* Bit Access */
        unsigned short PB7MD:2;           /* PB7MD */
        unsigned short PB6MD:2;           /* PB6MD */
        unsigned short PB5MD:2;           /* PB5MD */
        unsigned short PB4MD:2;           /* PB4MD */
        unsigned short PB3MD:2;           /* PB3MD */
        unsigned short PB2MD:2;           /* PB2MD */
        unsigned short PB1MD:2;           /* PB1MD */
        unsigned short PB0MD:2;           /* PB0MD */
    } BIT;                                /* */
} PBCR2;                                /* */
};                                       /* */
struct st_portc {                        /* struct PORTC */

```

```

union {
    unsigned short WORD;
    struct {
        unsigned short PC15DR:1;
        unsigned short PC14DR:1;
        unsigned short PC13DR:1;
        unsigned short PC12DR:1;
        unsigned short PC11DR:1;
        unsigned short PC10DR:1;
        unsigned short PC9DR:1;
        unsigned short PC8DR:1;
        unsigned short PC7DR:1;
        unsigned short PC6DR:1;
        unsigned short PC5DR:1;
        unsigned short PC4DR:1;
        unsigned short PC3DR:1;
        unsigned short PC2DR:1;
        unsigned short PC1DR:1;
        unsigned short PC0DR:1;
    } BIT;
} PCDR;
unsigned char wk0[2];
union {
    unsigned short WORD;
    struct {
        unsigned short PC15IOR:1;
        unsigned short PC14IOR:1;
        unsigned short PC13IOR:1;
        unsigned short PC12IOR:1;
        unsigned short PC11IOR:1;
        unsigned short PC10IOR:1;
        unsigned short PC9IOR:1;
        unsigned short PC8IOR:1;
        unsigned short PC7IOR:1;
        unsigned short PC6IOR:1;
        unsigned short PC5IOR:1;
        unsigned short PC4IOR:1;
        unsigned short PC3IOR:1;
    }

```

```

        unsigned short PC2IOR:1;          /* PC2IOR */
        unsigned short PC1IOR:1;          /* PC1IOR */
        unsigned short PC0IOR:1;          /* PC0IOR */
    } BIT;                                /*
} PCIOR;                                /*
unsigned char wk1[4];                    /*
union {                                  /* PCCR */
    unsigned short WORD;                 /* Word Access */
    struct {                             /* Bit Access */
        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 {                       /* struct PORTD */
    union {                             /* PDDRH */
        unsigned short WORD;             /* Word Access */
        struct {                         /* Bit Access */
            unsigned short PD31DR:1;      /* PD31DR */
            unsigned short PD30DR:1;      /* PD30DR */
            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 {                                /* PDDRL */
    unsigned short WORD;                /* Word Access */
    struct {                            /* Bit Access */
        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 {                                /* PDIORH */
    unsigned short WORD;                /* Word Access */
    struct {                            /* Bit Access */
        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 {                                  /* PDIORL */
    unsigned short WORD;                  /* Word Access */
    struct {                               /* Bit Access */
        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 */
        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 {                                  /* PDCRH1 */

```

```

unsigned short WORD; /* Word Access */
struct { /* Bit Access */
    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 { /* PDCRH2 */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        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 { /* PDCRL1 */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        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 {                                  /* PDCRL2 */
    unsigned short WORD;                  /* Word Access */
    struct {                               /* Bit Access */
        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 */
        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 {                         /* struct PORTE */
    union {                               /* PEDRL */
        unsigned short WORD;              /* Word Access */
        struct {                           /* Bit Access */
            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 {                                  /* PEIORL */
    unsigned short WORD;                 /* Word Access */
    struct {                             /* Bit Access */
        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 {                                  /* PECRL1 */

```

```

unsigned short WORD;                                /* Word Access */
struct {                                              /* Bit Access */
    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 */
} BIT;                                              /* */
} PECRL1;                                          /* */
union {                                              /* PECRL2 */
    unsigned short WORD;                            /* Word Access */
    struct {                                          /* Bit Access */
        unsigned short PE7MD:2;                    /* PE7MD */
        unsigned short PE6MD:2;                    /* PE6MD */
        unsigned short PE5MD:2;                    /* PE5MD */
        unsigned short PE4MD:2;                    /* PE4MD */
        unsigned short PE3MD:2;                    /* PE3MD */
        unsigned short PE2MD:2;                    /* PE2MD */
        unsigned short PE1MD:2;                    /* PE1MD */
        unsigned short PE0MD:2;                    /* PE0MD */
    } BIT;                                          /* */
} PECRL2;                                          /* */
};                                              /* */

struct st_portf {                                    /* struct PORTF */
    union {                                          /* PFDR */
        unsigned short WORD;                      /* Word Access */
        struct {                                    /* Bit Access */
            unsigned short :8;                     /* */
            unsigned short PF7DR:1;                /* PF7DR */
            unsigned short PF6DR:1;                /* PF6DR */
            unsigned short PF5DR:1;                /* PF5DR */
            unsigned short PF4DR:1;                /* PF4DR */
            unsigned short PF3DR:1;                /* PF3DR */
            unsigned short PF2DR:1;                /* PF2DR */
            unsigned short PF1DR:1;                /* PF1DR */
        };
    };
};

```

```

        unsigned short PF0DR:1;          /* PF0DR */
    } BIT;                                /*
} PFDR;                                  */
};                                       /*
struct st_mtu {                          /* struct MTU */
    union {                              /* ICSR1 */
        unsigned short WORD;            /* Word Access */
        struct {                         /* Bit Access */
            unsigned short POE3F:1;      /* POE3F */
            unsigned short POE2F:1;      /* POE2F */
            unsigned short POE1F:1;      /* POE1F */
            unsigned short POE0F:1;      /* POE0F */
            unsigned short :3;           /*
            unsigned short PIE:1;        /* PIE */
            unsigned short POE3M:2;      /* POE3M */
            unsigned short POE2M:2;      /* POE2M */
            unsigned short POE1M:2;      /* POE1M */
            unsigned short POE0M:2;      /* POE0M */
        } BIT;                          /*
    } ICSR1;                             /*
    union {                              /* OCSR */
        unsigned short WORD;            /* Word Access */
        struct {                         /* Bit Access */
            unsigned short OSF:1;        /* OSF */
            unsigned short :5;           /*
            unsigned short OCE:1;        /* OCE */
            unsigned short OIE:1;        /* OIE */
            unsigned short :8;           /*
        } BIT;                          /*
    } OCSR;                             /*
};                                       /*
struct st_cmt {                          /* struct CMT */
    union {                              /* CMSTR */
        unsigned short WORD;            /* Word Access */
        struct {                         /* Bit Access */
            unsigned short :14;          /*
            unsigned short STR:2;        /* STR */
        } BIT;                          /*

```

```

    } CMSTR;                                /* */
union {                                    /* CMCSR_0 */
    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      */
    unsigned short WORD;           /* Word Access */
    struct {                       /* Bit Access  */
        unsigned short AD:10;      /* AD          */
        unsigned short :6;         /*          */
    } BIT;                         /*          */
} ADDR6;                          /*          */

```

```

union {
    unsigned short WORD;
    struct {
        unsigned short AD:10;
        unsigned short :6;
    } BIT;
} ADDR7;
unsigned char wk0[80];
union {
    unsigned char BYTE;
    struct {
        unsigned char ADF:1;
        unsigned char ADIE:1;
        unsigned char :1;
        unsigned char ADM:1;
        unsigned char :2;
        unsigned char CH:2;
    } BIT;
} ADCSR_0;
union {
    unsigned char BYTE;
    struct {
        unsigned char ADF:1;
        unsigned char ADIE:1;
        unsigned char :1;
        unsigned char ADM:1;
        unsigned char :2;
        unsigned char CH:2;
    } BIT;
} ADCSR_1;
unsigned char wk1[6];
union {
    unsigned char BYTE;
    struct {
        unsigned char TRGE:1;
        unsigned char CKS:2;
        unsigned char ADST:1;
        unsigned char ADCS:1;
    } BIT;
}

```

```

        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 */
    union {                              /* FLMCR1 */
        unsigned char BYTE;              /* Byte Access */
        struct {                         /* Bit Access */
            unsigned char FWE:1;          /* FWE */
            unsigned char SWE:1;          /* SWE */
            unsigned char ESU:1;          /* ESU */
            unsigned char PSU:1;          /* PSU */
            unsigned char EV:1;           /* EV */
            unsigned char PV:1;           /* PV */
            unsigned char E:1;            /* E */
            unsigned char P:1;            /* P */
        } BIT;                          /* */
    } FLMCR1;                            /* */
    union {                              /* FLMCR2 */

```

```

unsigned char BYTE; /* Byte Access */
struct { /* Bit Access */
    unsigned char FLER:1; /* FLER */
    unsigned char :7; /* */
} BIT; /* */
} FLMCR2; /* */
union { /* EBR1 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char EB:8; /* EB */
    } BIT; /* */
} EBR1; /* */
union { /* EBR2 */
    unsigned char BYTE; /* Byte Access */
    struct { /* Bit Access */
        unsigned char :4; /* */
        unsigned char EB11:1; /* EB11 */
        unsigned char EB10:1; /* EB10 */
        unsigned char EB9:1; /* EB9 */
        unsigned char EB8:1; /* EB8 */
    } BIT; /* */
} EBR2; /* */
unsigned char wk0[164]; /* */
union { /* RAMER */
    unsigned short WORD; /* Word Access */
    struct { /* Bit Access */
        unsigned short :12; /* */
        unsigned short RAMS:1; /* RAMS */
        unsigned short RAM:3; /* RAM */
    } BIT; /* */
} RAMER; /* */
}; /* */

struct st_abc { /* struct ABC */
    union { /* UBARH */
        unsigned short WORD; /* Word Access */
        struct { /* Bit Access */
            unsigned short UBA31:1; /* UBA31 */
            unsigned short UBA30:1; /* UBA30 */

```

```

        unsigned short UBA29:1;          /* UBA29 */
        unsigned short UBA28:1;          /* UBA28 */
        unsigned short UBA27:1;          /* UBA27 */
        unsigned short UBA26:1;          /* UBA26 */
        unsigned short UBA25:1;          /* UBA25 */
        unsigned short UBA24:1;          /* UBA24 */
        unsigned short UBA23:1;          /* UBA23 */
        unsigned short UBA22:1;          /* UBA22 */
        unsigned short UBA21:1;          /* UBA21 */
        unsigned short UBA20:1;          /* UBA20 */
        unsigned short UBA19:1;          /* UBA19 */
        unsigned short UBA18:1;          /* UBA18 */
        unsigned short UBA17:1;          /* UBA17 */
        unsigned short UBA16:1;          /* UBA16 */
    } BIT;                               /* */

} UBARH;                               /* */

union {                                /* UBARL */
    unsigned short WORD;                /* Word Access */
    struct {                             /* Bit Access */
        unsigned short UBA15:1;          /* UBA15 */
        unsigned short UBA14:1;          /* UBA14 */
        unsigned short UBA13:1;          /* UBA13 */
        unsigned short UBA12:1;          /* UBA12 */
        unsigned short UBA11:1;          /* UBA11 */
        unsigned short UBA10:1;          /* UBA10 */
        unsigned short UBA9:1;           /* UBA9 */
        unsigned short UBA8:1;           /* UBA8 */
        unsigned short UBA7:1;           /* UBA7 */
        unsigned short UBA6:1;           /* UBA6 */
        unsigned short UBA5:1;           /* UBA5 */
        unsigned short UBA4:1;           /* UBA4 */
        unsigned short UBA3:1;           /* UBA3 */
        unsigned short UBA2:1;           /* UBA2 */
        unsigned short UBA1:1;           /* UBA1 */
        unsigned short UBA0:1;           /* UBA0 */
    } BIT;                               /* */
} UBARL;                               /* */

union {                                /* UBAMRH */

```

```

unsigned short WORD;                                /* Word Access */
struct {                                             /* Bit Access */
    unsigned short UBM31:1;                        /* UBM31 */
    unsigned short UBM30:1;                        /* UBM30 */
    unsigned short UBM29:1;                        /* UBM29 */
    unsigned short UBM28:1;                        /* UBM28 */
    unsigned short UBM27:1;                        /* UBM27 */
    unsigned short UBM26:1;                        /* UBM26 */
    unsigned short UBM25:1;                        /* UBM25 */
    unsigned short UBM24:1;                        /* UBM24 */
    unsigned short UBM23:1;                        /* UBM23 */
    unsigned short UBM22:1;                        /* UBM22 */
    unsigned short UBM21:1;                        /* UBM21 */
    unsigned short UBM20:1;                        /* UBM20 */
    unsigned short UBM19:1;                        /* UBM19 */
    unsigned short UBM18:1;                        /* UBM18 */
    unsigned short UBM17:1;                        /* UBM17 */
    unsigned short UBM16:1;                        /* UBM16 */
} BIT;                                              /*
} UBAMRH;                                          /*
union {                                           /* UBAMRL */
    unsigned short WORD;                        /* Word Access */
    struct {                                     /* Bit Access */
        unsigned short UBM15:1;                /* UBM15 */
        unsigned short UBM14:1;                /* UBM14 */
        unsigned short UBM13:1;                /* UBM13 */
        unsigned short UBM12:1;                /* UBM12 */
        unsigned short UBM11:1;                /* UBM11 */
        unsigned short UBM10:1;                /* UBM10 */
        unsigned short UBM9:1;                 /* UBM9 */
        unsigned short UBM8:1;                 /* UBM8 */
        unsigned short UBM7:1;                 /* UBM7 */
        unsigned short UBM6:1;                 /* UBM6 */
        unsigned short UBM5:1;                 /* UBM5 */
        unsigned short UBM4:1;                 /* UBM4 */
        unsigned short UBM3:1;                 /* UBM3 */
        unsigned short UBM2:1;                 /* UBM2 */
        unsigned short UBM1:1;                 /* UBM1 */
    };

```

```

        unsigned short UBM0:1;          /*   UBM0   */
    } BIT;                               /*           */
} UBAMRL;                               /*           */
union {                                 /* UBBR      */
    unsigned short WORD;                /*   Word Access   */
    struct {                             /*   Bit Access   */
        unsigned short :8;              /*           */
        unsigned short CP:2;            /*   CP           */
        unsigned short ID:2;            /*   ID           */
        unsigned short RW:2;            /*   RW           */
        unsigned short SZ:2;            /*   SZ           */
    } BIT;                               /*           */
} UBBR;                                 /*           */
union {                                 /* UBCR      */
    unsigned short WORD;                /*   Word Access   */
    struct {                             /*   Bit Access   */
        unsigned short :15;             /*           */
        unsigned short UBID:1;          /*   UBID         */
    } BIT;                               /*           */
} UBCR;                                 /*           */
};                                     /*           */
struct st_wdt {                         /* struct WDT   */
    union {                             /* TCSR        */
        unsigned char BYTE;             /*   Byte Access   */
        struct {                         /*   Bit Access   */
            unsigned char OVF:1;         /*   OVF          */
            unsigned char WTIT:1;        /*   WT/IT        */
            unsigned char TME:1;         /*   TME          */
            unsigned char :2;            /*           */
            unsigned char CKS:3;         /*   CKS          */
        } BIT;                           /*           */
    } TCSR;                             /*           */
    unsigned char TCNT;                  /*   TCNT         */
    union {                             /* RSTCSR      */
        unsigned char BYTE;             /*   Byte Access   */
        struct {                         /*   Bit Access   */
            unsigned char WOVF:1;        /*   WOVF         */
            unsigned char RSTE:1;        /*   RSTE         */
        }
    }

```

```

        unsigned char RSTS:1;          /* RSTS */
        unsigned char :5;              /* */
    } BIT;                             /* */
} RSTCSR;                             /* */
};                                     /* */
struct st_stby {                      /* struct STBY */
    union {                           /* SBYCR */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char SBY:1;      /* SBY */
            unsigned char HIZ:1;      /* HIZ */
            unsigned char :4;         /* */
            unsigned char IRQEH:1;    /* IRQEH */
            unsigned char IRQEL:1;    /* IRQEL */
        } BIT;                       /* */
    } SBYCR;                          /* */
    unsigned char wk0[3];             /* */
    union {                           /* SYSCR */
        unsigned char BYTE;          /* Byte Access */
        struct {                     /* Bit Access */
            unsigned char :6;         /* */
            unsigned char AUDSRST:1;  /* AUDSRST */
            unsigned char RAME:1;     /* RAME */
        } BIT;                       /* */
    } SYSCR;                          /* */
    unsigned char wk1[3];             /* */
    union {                           /* MSTCR1 */
        unsigned short WORD;         /* Word Access */
        struct {                     /* Bit Access */
            unsigned short :4;        /* */
            unsigned short MSTP27:1;  /* MSTP27 */
            unsigned short MSTP26:1;  /* MSTP26 */
            unsigned short MSTP25:1;  /* MSTP25 */
            unsigned short MSTP24:1;  /* MSTP24 */
            unsigned short :2;        /* */
            unsigned short MSTP21:1;  /* MSTP21 */
            unsigned short :1;        /* */
            unsigned short MSTP19:1;  /* MSTP19 */
        }
    }

```

```

        unsigned short MSTP18:1;          /* MSTP18 */
        unsigned short MSTP17:1;          /* MSTP17 */
        unsigned short MSTP16:1;          /* MSTP16 */
    } BIT;                                /* */
} MSTCR1;                                /* */
union {                                  /* MSTCR2 */
    unsigned short WORD;                  /* Word Access */
    struct {                               /* Bit Access */
        unsigned short :2;                /* */
        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 {                           /* struct BSC */
    union {                               /* BCR1 */
        unsigned short WORD;              /* Word Access */
        struct {                           /* Bit Access */
            unsigned short :2;              /* */
            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;
    struct {
        unsigned short IW3:2;
        unsigned short IW2:2;
        unsigned short IW1:2;
        unsigned short IW0:2;
        unsigned short CW3:1;
        unsigned short CW2:1;
        unsigned short CW1:1;
        unsigned short CW0:1;
        unsigned short SW3:1;
        unsigned short SW2:1;
        unsigned short SW1:1;
        unsigned short SW0:1;
    } BIT;
} BCR2;

union {
    unsigned short WORD;
    struct {
        unsigned short W3:4;
        unsigned short W2:4;
        unsigned short W1:4;
        unsigned short W0:4;
    } BIT;
} WCR1;

union {
    unsigned short WORD;
    struct {
        unsigned short :12;
        unsigned short DSW:4;
    } BIT;
} WCR2;

};

struct st_dmac {
    union {
        unsigned short WORD;
        struct {

```

```

        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   */
    unsigned long DMATCR2;                           /* DMATCR2 */
    union {                                          /* CHCR2   */
        unsigned long LONG;                        /* Long Access */
        struct {                                    /* Bit Access */
            unsigned long :12;                     /*          */
            unsigned long RO:1;                     /*   RO   */
            unsigned long :3;                       /*          */
            unsigned long DM:2;                     /*   DM   */
            unsigned long SM:2;                     /*   SM   */
            unsigned long RS:4;                     /*   RS   */
            unsigned long :2;                       /*          */
            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;
    } 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;
/* DTEC */
/* Byte Access */
/* Bit Access */
/* DTEC */
/* */

union {
    unsigned char BYTE;
    struct {
        unsigned char DTED:8;
    } BIT;
} DTED;
/* DTED */
/* Byte Access */
/* Bit Access */
/* DTED */
/* */

unsigned char wk0[2];
/* */

union {
    unsigned short WORD;
    struct {
        unsigned short :5;
        unsigned short NMIF:1;
        unsigned short AE:1;
        unsigned short SWDTE:1;
        unsigned short DTVEC7:1;
        unsigned short DTVEC6:1;
        unsigned short DTVEC5:1;
        unsigned short DTVEC4:1;
        unsigned short DTVEC3:1;
        unsigned short DTVEC2:1;
        unsigned short DTVEC1:1;
        unsigned short DTVEC0:1;
    } BIT;
} DTCSR;
/* DTCSR */
/* Word Access */
/* Bit Access */
/* */
/* NMIF */
/* AE */
/* SWDTE */
/* DTVEC7 */
/* DTVEC6 */
/* DTVEC5 */
/* DTVEC4 */
/* DTVEC3 */
/* DTVEC2 */
/* DTVEC1 */
/* DTVEC0 */
/* */

unsigned short DTBR;
/* DTBR */

unsigned char wk1[6];
/* */

union {
    unsigned char BYTE;
    struct {
        unsigned char :2;
        unsigned char DTEE5:1;
        unsigned char :1;
    } BIT;
} DTEE;
/* DTEE */
/* Byte Access */
/* Bit Access */
/* */
/* DTEE5 */
/* */

```

```

        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];                   /* */
union {                                 /* DTEG */
    unsigned char BYTE;                 /* Byte Access */
    struct {                             /* Bit Access */
        unsigned char DTEG7:1;          /* DTEG7 */
        unsigned char :7;               /* */
    } BIT;                               /* */
} DTEG;                                 /* */
};                                       /* */
struct st_iic {                          /* struct IIC */
    union {                              /* SCRX */
        unsigned char BYTE;             /* Byte Access */
        struct {                         /* Bit Access */
            unsigned char :2;            /* */
            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];               /* */
    union {                              /* ICCR0 */
        unsigned char BYTE;             /* Byte Access */
        struct {                         /* Bit Access */
            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 */
        } BIT;                           /* */
    } ICCR0;

```

```

        unsigned char IRIC:1;          /* IRIC */
        unsigned char SCP:1;          /* SCP */
    } BIT;                             /* */
} ICCR0;                             /* */
union {                               /* ICSR0 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char ESTP:1;         /* ESTP */
        unsigned char STOP:1;         /* STOP */
        unsigned char IRTR:1;         /* IRTR */
        unsigned char AASX:1;         /* AASX */
        unsigned char AL:1;           /* AL */
        unsigned char AAS:1;          /* AAS */
        unsigned char ADZ:1;          /* ADZ */
        unsigned char ACKB:1;         /* ACKB */
    } BIT;                           /* */
} ICSR0;                             /* */
unsigned char wk1[4];                 /* */
union {                               /* ICDR0 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char ICDR:8;         /* ICDR */
    } BIT;                           /* */
} ICDR0;                             /* */
union {                               /* ICMR0 */
    unsigned char BYTE;               /* Byte Access */
    struct {                          /* Bit Access */
        unsigned char MLS:1;          /* MLS */
        unsigned char WAIT:1;         /* WAIT */
        unsigned char CKS:3;          /* CKS */
        unsigned char BC:3;           /* BC */
    } BIT;                           /* */
} ICMR0;                             /* */
};                                    /* */
struct st_hudi {                     /* struct H-UDI */
    union {                           /* SDIR */
        unsigned short WORD;          /* Word Access */
        struct {                      /* Bit Access */

```

```

        unsigned short TS:4;                /*    TS        */
        unsigned short :12;                /*            */
    } BIT;                                  /*            */
} SDIR;                                    /*            */
union {                                    /* SDSA        */
    unsigned short WORD;                    /* Word Access */
    struct {                                /* Bit Access  */
        unsigned short :15;                /*            */
        unsigned short SDTRF:1;            /* SDTRF       */
    } BIT;                                  /*            */
    } SDSA;                                  /*            */
    unsigned short SDDRH;                    /* SDDRH       */
    unsigned short SDDRL;                    /* 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_ubic *)0xFFFF8600) /* UBC Address */
#define P_WDT (*(volatile struct st_wdt *)0xFFFF8610) /* WDT Address */
#define P_STBY (*(volatile struct st_stby *)0xFFFF8614)/* STBY Address */
#define P_BSC (*(volatile struct st_bsc *)0xFFFF8620) /* BSC Address */
#define P_DMAL (*(volatile struct st_dmac *)0xFFFF86B0)/* DMAL 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 */
```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.09.15	—	初版発行

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、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、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、ルネサス販売または特約店までご照会ください。