

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

H8/300H Tiny シリーズ

ROM コレクション

要旨

ROM コレクションとは、ROM の不具合修正や仕様の変更を行うものです。実行アドレスが設定したアドレスになった場合に割り込み処理を発生するアドレスブレイク機能と、EEPROM などの外付けメモリを使用して実現します。

動作確認デバイス

H8/3664

HN58X2464I (EEPROM)

目次

1. 仕様	2
2. 使用機能説明	3
3. 動作説明	6
4. ソフトウェア説明	10
5. フローチャート	17
6. プログラムリスト	27

1. 仕様

- 本タスク例における ROM コレクションの仕様を図 1 に示します。

「ROM コレクション機能」

通常マスク投入後に不具合が発見された場合は、マスク再投入とともに不具合の回収、そして、不良在庫へと時間面においてもコスト面においても多大なロスが生じます。H8/3664 では、実行アドレスをソフトウェアでチェックする「アドレスブレイク割り込み機能」を内蔵しました。この「アドレスブレイク割り込み機能」と EEPROM などの外付けメモリを使用することにより、より簡単に ROM の不具合修正や、仕様の変更を行うことができます。これを ROM コレクション機能と呼んでいます。

- プログラムを起動すると、P75 を出力機能に設定し、P75 のポート出力処理をします。
- P75 の切り換え動作中に IRQ1 割り込みが発生すると、外付け EEPROM から修正プログラムをダウンロードします。
- ダウンロード後、アドレスブレイク機能を用いてダウンロードした修正プログラムデータのある内蔵 RAM へ実行アドレスを移動します。
- 修正プログラムは、P75 を入力機能、P74 を出力機能に設定します。P75 から P74 のポート出力処理に修正されます。
- 修正終了後、修正プログラムで設定したアドレスへ復帰します。
- P75、P74 共に出力機能に設定し、Low に切り換え、終了します。

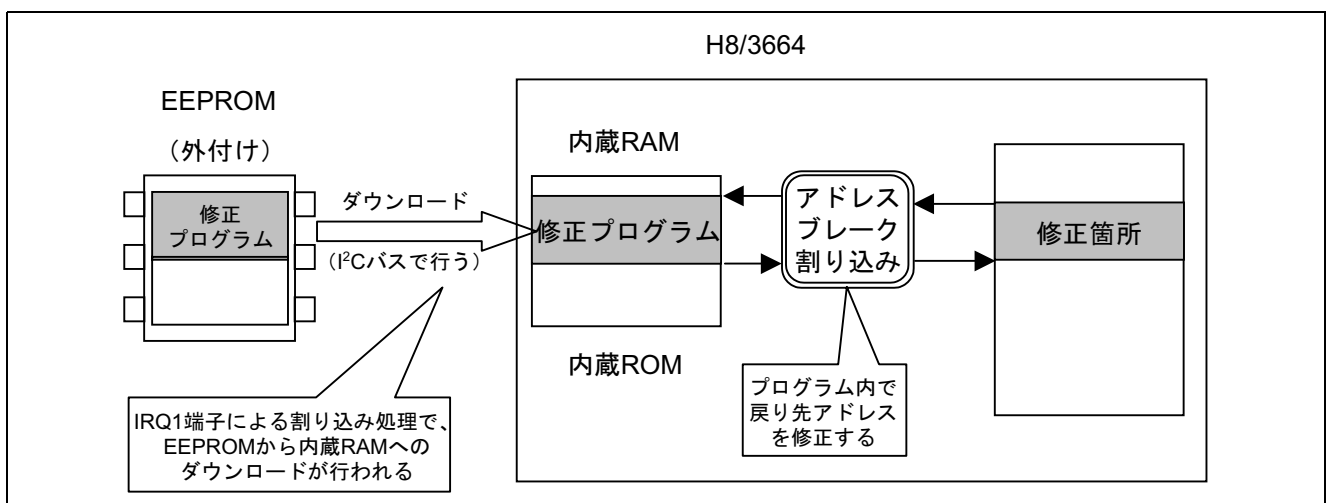


図 1 ROM コレクションの仕様

2. 使用機能説明

2.1 I²C バスインターフェース

- I²C バスインターフェースのブロック図を図 2に示します。以下に、I²C バスインターフェースのブロック図の説明をします。
 - I²C バスコントロールレジスタ (ICCR)
 - I²C バスインターフェースの制御ビットと割り込み要求フラグで構成されています。
 - I²C バスモードレジスタ (ICMR)
 - 転送フォーマットと転送レートを設定します。ICCR の ICE ビットが 1 の時だけアクセスできます。
 - I²C バスステータスレジスタ (ICSR)
 - ステータスフラグで構成されます。
 - I²C バスデータレジスタ (ICDR)
 - 8 ビットのリード/ライト可能なレジスタで、送信時は送信用データレジスタとして、受信時は受信データレジスタとして機能します。

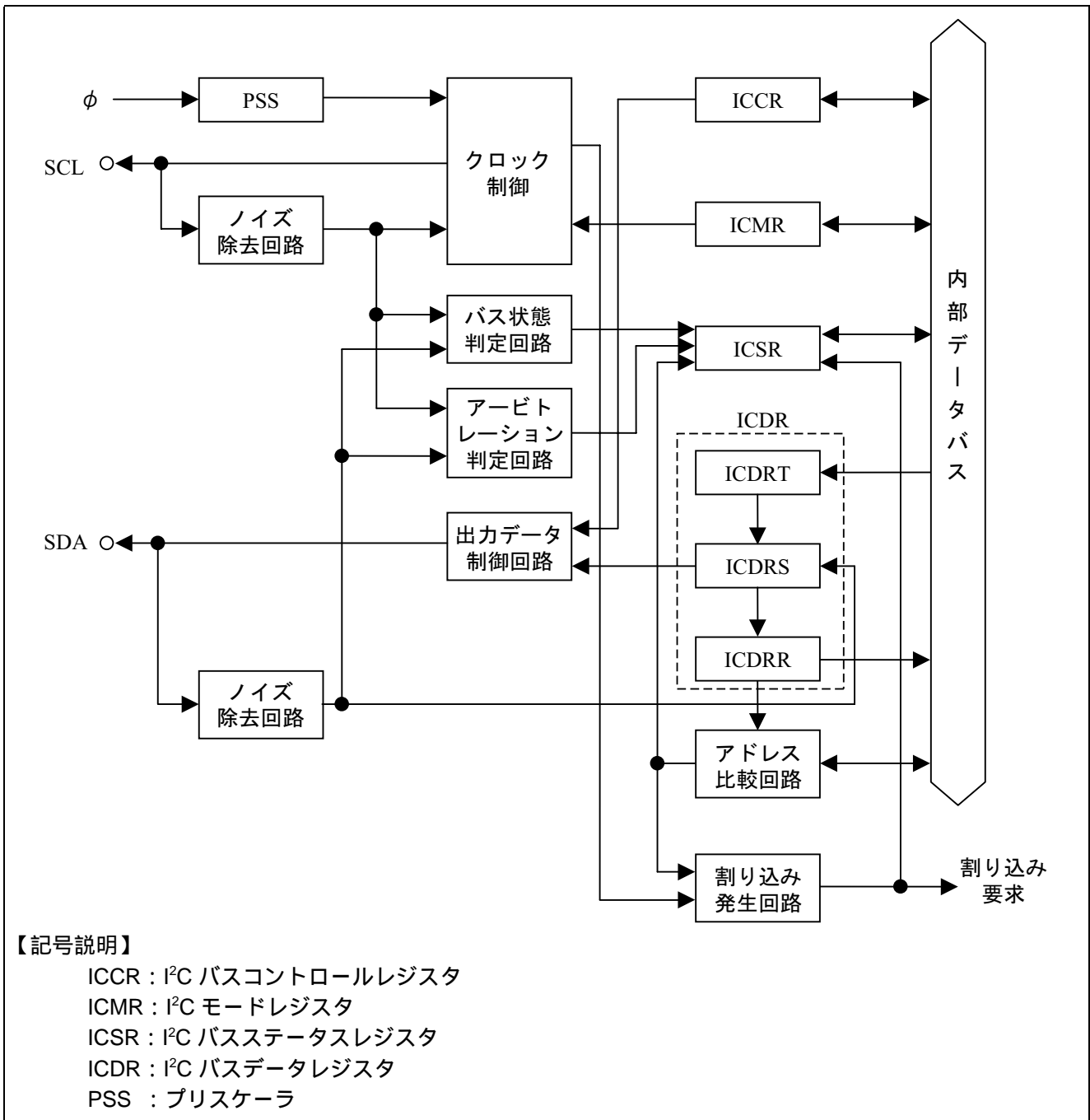


図 2 I²C バスインターフェースのブロック図

- I²C バスインターフェースで使用する端子を表 1 に示します。

表 1 I²C バスインターフェースで使用する端子

名称	記号	入出力	機能
シリアルクロック端子	SCL	入出力	I ² C シリアルクロック入出力端子
シリアルデータ端子	SDA	入出力	I ² C シリアルデータ入出力端子

2.2 アドレスブレイク

- アドレスブレイク割り込み処理ブロック図を図 3 に示します。以下に、アドレスブレイク割り込み処理ブロック図の説明をします。

- アドレスブレイクコントロールレジスタ (ABRKCR)
アドレスブレイクの条件設定を行います。
- アドレスブレイクステータスレジスタ (ABRKSR)
アドレスブレイクの割り込み要求フラグとそのイネーブルビットで構成されます。
- ブレイクアドレスレジスタ (BAR (BARL、BARH))
アドレスブレイク割り込み処理を発生させるためのアドレスを設定する 16 ビットのリード/ライト可能なレジスタです。BARH は (上位 8 ビット)、BARL は (下位 8 ビット) を表します。

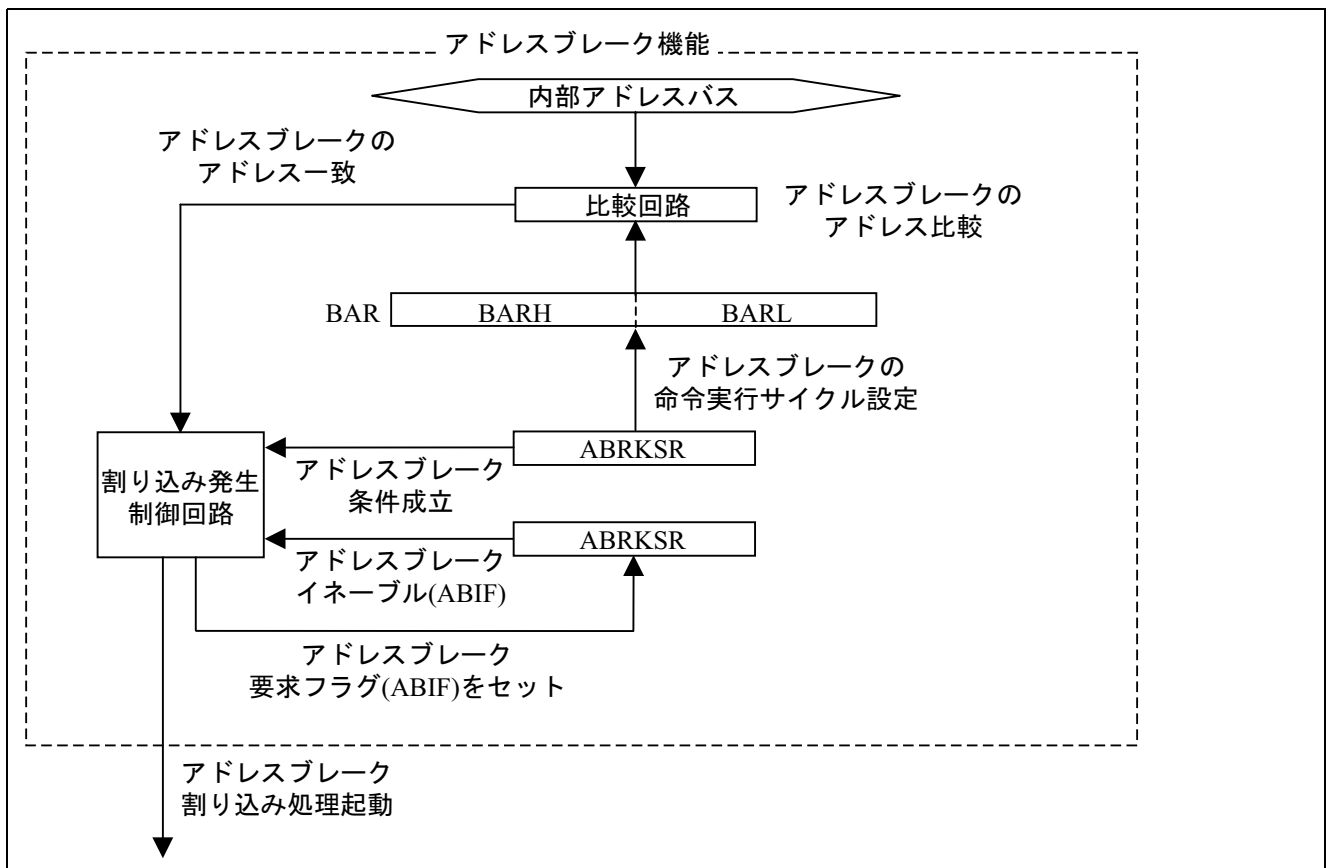


図 3 アドレスブレイク割り込み処理ブロック図

3. 動作説明

- 本タスク例における、ROM コレクションの動作概要を図 4 ~ 図 7に示します。以下に、ROM コレクションの動作概要を説明します。

まず、H8/3664 のプログラムを起動させます。

IRQ1 割り込み処理を発生させ、EEPROM のアドレス H'0000 から図 4に示すデータを読み込みます。

- ダウンロードしたブレークアドレス (H'011E) を、ブレークアドレスレジスタ (BAR) に設定します。
- 修正プログラムを内蔵 RAM の H'FC00 へ、修正プログラムサイズ (H'52) 分だけダウンロードします。

データ読み込み後、アドレスブレーク条件 (ブレークアドレスレジスタ以外) の設定を行います。

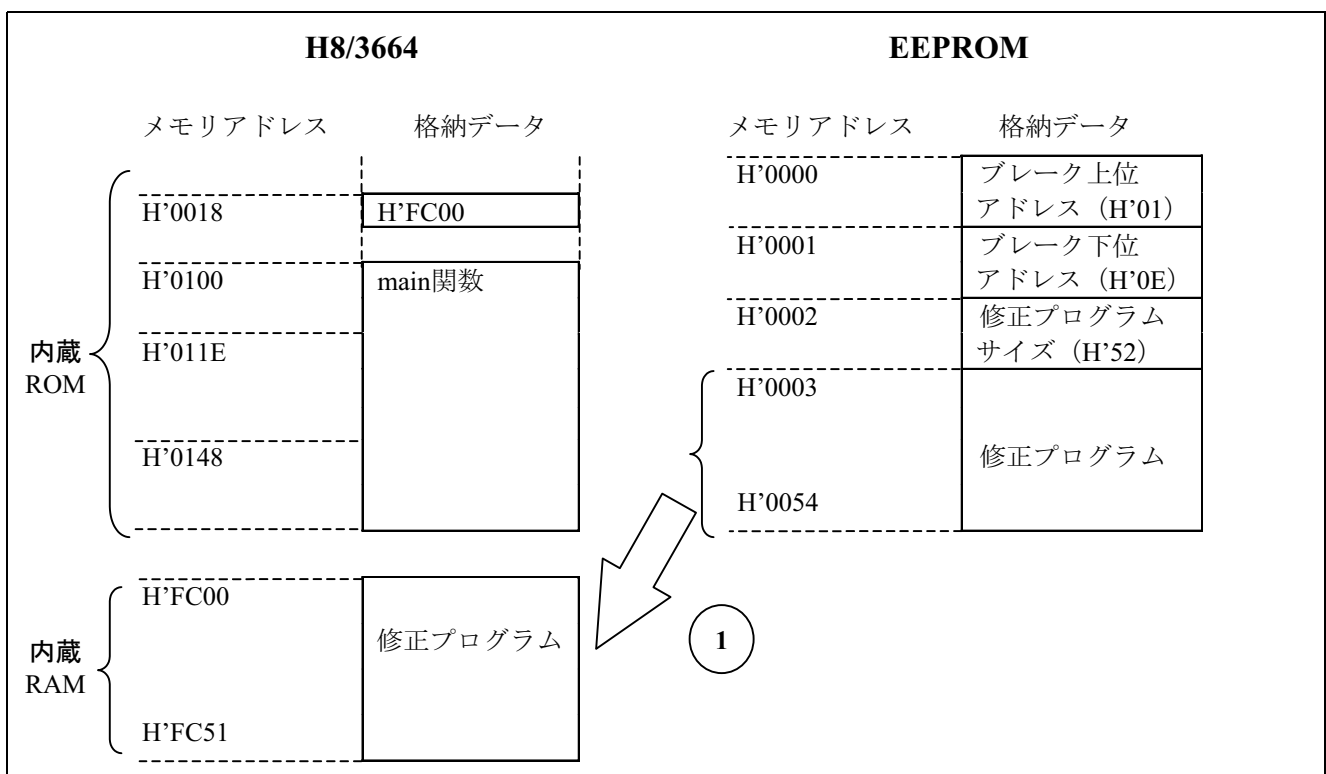


図 4 ROM コレクション動作概要 1

アドレスブレイク条件 (H'011E 番地の命令実行後) を満たすと、アドレスブレイク割り込みが発生し、ベクタアドレス H'0018 へ移動します。

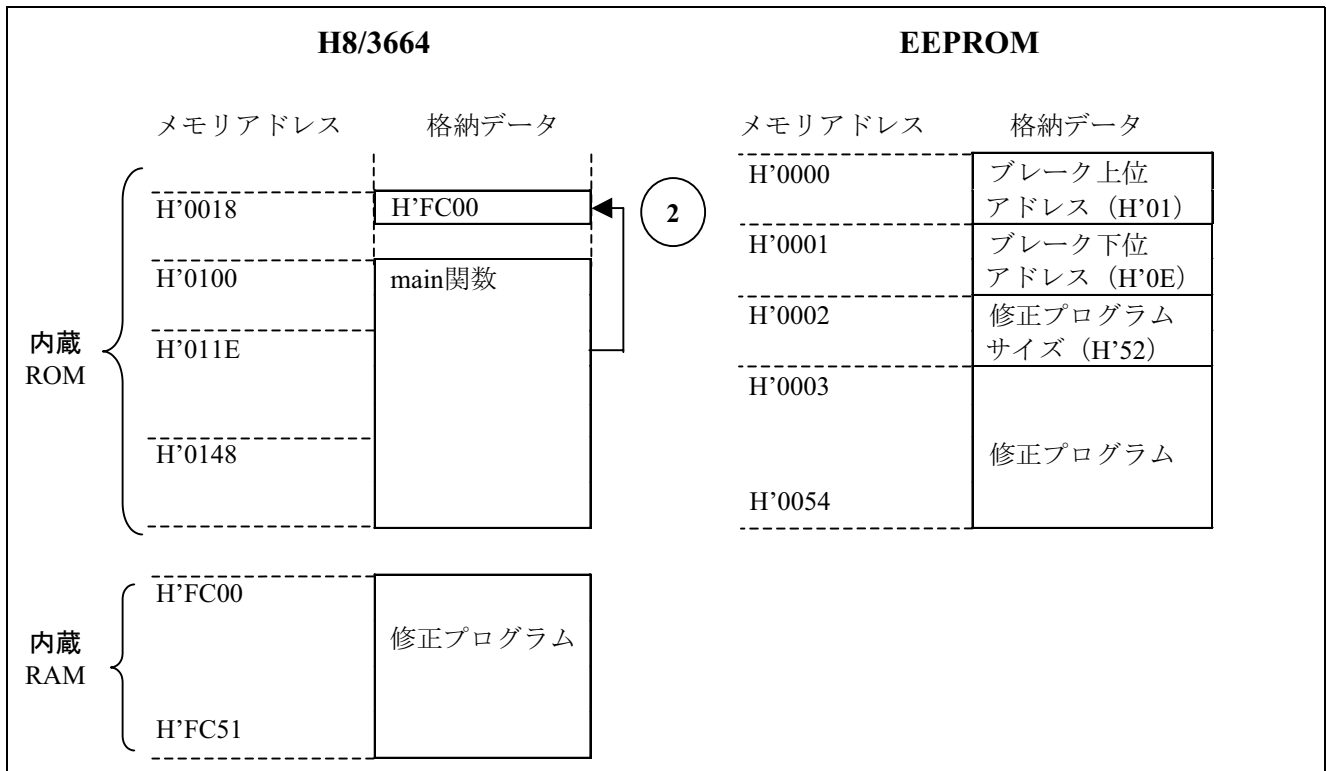


図 5 ROM コレクション動作概要 2

ベクタアドレス H'0018 に設定しているアドレス (H'FC00) へ移動し、修正プログラムを起動します。

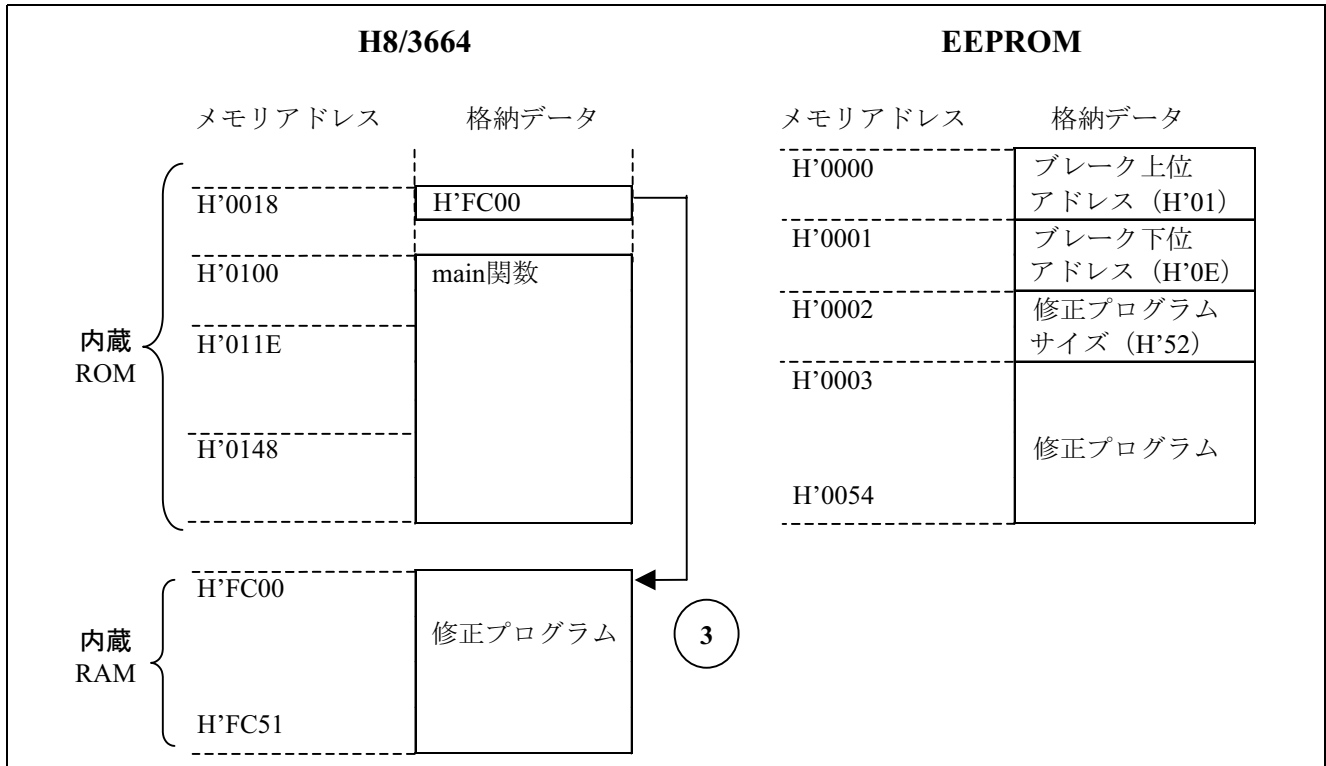


図 6 ROM コレクションの動作概要 3

修正プログラム実行後、修正プログラムで設定しているアドレス (H'0148) へ復帰します。

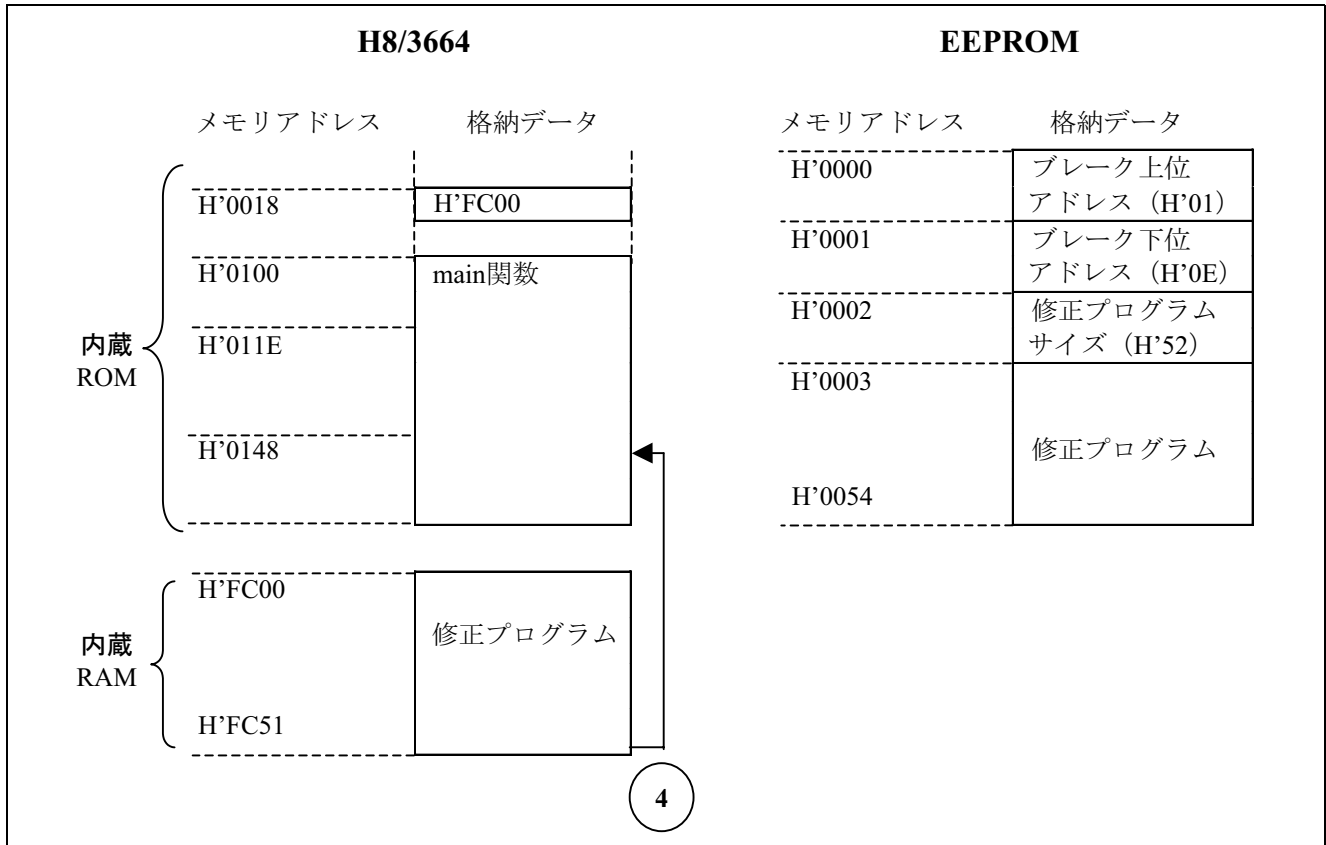


図 7 ROM コレクションの動作概要 4

4. ソフトウェア説明

4.1 モジュール説明

- 本タスク例のモジュールを表 2に示します。

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	IRQ1 割り込み設定、 P75 の High、Low 切り換え動作
IRQ1 割り込みルーチン	irq1int	EEPROM から Read_EEPROM 関数のコール、 アドレスブレイク条件の設定
ダウンロード設定ルーチン	Read_EEPROM	Set_Adrs、Recv_data 関数のコール
ダウンロードルーチン	Recv_data	EEPROM から修正プログラムをダウンロード
スレーブアドレス設定ルーチン	Set_Adrs	スレーブアドレスの設定

4.2 引数の説明

- 本タスク例で使用する引数について表 3に示します。

表 3 使用する引数

ラベル名	引数名	説明
Read_EEPROM	unsigned short adrs	EEPROM 読み込み開始アドレス
	unsigned char *rd_data	読み込みデータ
	unsigned short *br_ad	ブレイクアドレスポインタ
Recv_data	unsigned char *rd_data	読み込みデータ
	unsigned short *br_ad	ブレイクアドレスポインタ
Set_Adrs	unsigned short adrs	EEPROM 読み込み開始アドレス

4.3 戻り値の説明

- 本タスク例で使用する戻り値について表 4に示します。

表 4 各モジュールの戻り値

モジュール名	ラベル名	戻り値の説明
ダウンロード設定ルーチン	Read_EEPROM	0：正常終了 1：異常終了
ダウンロードルーチン	Recv_data	0：正常終了 1：異常終了
スレーブアドレス設定ルーチン	Set_Adrs	0：正常終了 1：異常終了

4.4 変数説明

- 本タスク例で使用する変数について表 5に示します。

表 5 使用する変数

ラベル名	変数名	説明
main	volatile unsigned long i	Wait のカウント
	unsigned char j	P75 の High, Low 切り替えカウント
irq1int	unsigned short adrs	EEPROM 読み込み開始アドレス
	unsigned short bar	ブ레이크アドレスを格納
	unsigned short *br_ad	ブ레이크アドレスポインタ
	unsigned char tmp	エラーチェック
	unsigned char *rd_data	読み込みデータ
Read_EEPROM	unsigned char tmp	エラーチェック
Recv_data	unsigned char recv	ダミーリード、修正プログラムサイズデータを格納
	unsigned char cnt	修正プログラムのリードカウント

4.5 定数説明

- 本タスク例で使用する定数について表 6に示します。

表 6 使用する定数

定数名	定数	説明
DEVICE_CODE	H'A0	デバイス・コード
SLAVE_ADRS	H'00	デバイス・アドレス・コード
IIC_DATA_W	H'00	WRITE コード
IIC_DATA_R	H'01	READ コード
EP_ADRS	H'0000	EEPROM 読み込み開始アドレス
RAM_AREA	H'FC00	EEPROM から読み込むデータの格納開始アドレス

4.6 使用 RAM 説明

- 本タスク例では、EEPROM からダウンロードする修正プログラムを H'FC00 から格納していきます。

4.7 使用内部レジスタ説明

- 本タスク例の使用内部レジスタを以下に示します。

— ICCR I²C バスコントロールレジスタ (ICCR) アドレス : H'FFC4

ビット	ビット名	設定値	R/W	機能
7	ICE	1	R/W	I ² C バスインターフェースイネーブル ICE=1 : I ² C バスインターフェースモジュールは送受信可能となり、SCL、SDA 端子と接続されてバスを駆動できる状態になる。また、ICMR、ICDR がアクセス可能となる。 ICE=0 : モジュールは停止状態となり SCL、SDA 端子から切り離される。
6	IEIC	0	R/W	I ² C バスインターフェース割り込みイネーブル このビットが 1 の時、IRIC による割り込み要求がイネーブルになる
5 4	MST TRS	1 1	R/W R/W	マスタ / スレーブ選択 送受信選択 00 : スレーブ受信モード 01 : スレーブ送信モード 10 : マスタ受信モード 11 : マスタ送信モード
3	ACKE	1	R/W	アクノリッジビット判定選択 1 : 受信したアクノリッジビットが 1 の時、転送を中断する。 0 : 受信したアクノリッジビットを無視して連続的に転送を行う。受信したアクノリッジビットは ACKB ビットに反映されず、常時 0 となる。
2	BBSY	0	R/W	バスビジー マスタモードでは開始条件、停止条件の発行に使用する。 開始条件の発行 : BBSY に 1、SCP に 0 をライトすることで行う。開始条件の再送信も同様である。SCL=High レベルの状態では SDA が High レベルから Low レベルにすると発行認識。発行後、1 にセットされる。 停止条件の発行 : BBSY に 0、SCP に 0 をライトすることで行う。SCL=High レベルの状態では SDA が Low レベルから High レベルにすると発行認識。発行後、0 にクリアされる。 開始条件 / 停止条件の発行には、MOV 命令を用いる。開始条件の発行に先立って、I ² C バスインターフェースをマスタ送信モードに設定する必要がある。
1	IRIC	0	R/W	I ² C バスインターフェース割り込み要求フラグ [セット条件] (I ² C バスフォーマットでマスタモードの時) ・開始条件を発行後、バスラインの状態から開始条件を検出した時 ・WAIT=1 で、データとアクノリッジの間にウェイトを挿入した時 ・データ転送終了時 ・バス競合負けの後、スレーブアドレスを受信した時 ・ACKE ビットが 1 で、アクノリッジビットとして 1 を受信した時 (ACKB ビットが 1 にセットされた時) [クリア条件] ・1 の状態をリードした後、0 をライトした時
0	SCP	1	W	開始条件 / 停止条件発行禁止ビット SCP ビットはマスタモードでの開始条件 / 停止条件の発行を制御する。 開始条件を発行する場合、BBSY に 1、SCP に 0 をライトする。開始条件の再送信時も同様に行う。停止条件の発行は BBSY に 0、SCP に 0 をライトすることで行う。本ビットは、リードすると常に 1 が読み出される。1 をライトしてもデータは格納されない。

— ICSR I²C バスステータスレジスタ (ICSR) アドレス : H'FFC5

ビット	ビット名	設定値	R/W	機能
0	ACKB	0	R/W	<p>アクノリッジビット</p> <p>送信モードでは、受信デバイスから返されるアクノリッジデータをロードする。</p> <p>受信モードでは送信デバイスに対し、データを受信した後、あらかじめ本ビットに設定されたアクノリッジデータを送出する。</p> <p>本ビットをリードすると、送信時にはロードした値(受信デバイスから返ってきた値)が読み出され、受信時には設定した値が読み出される。</p>

— ICDR I²C バスデータレジスタ (ICDR) アドレス : H'FFC6

8 ビットのリード/ライト可能なレジスタで、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機能します。ICDR は内部でシフトレジスタ (ICDRS)、受信バッファ (ICDRR)、送信バッファ (ICDRT) に分かれています。

3 本のレジスタ間データ転送はバス状態の変化に関連付けられて自動的に行われ、TDRE や RDRF などの内部フラグの状態に影響を与えます。

- ・ TDRE は 1 の時、受信バッファが空の状態 CPU から次の送信データをライト可能であることを示します。

- ・ RDRF は 1 の時、受信バッファに有効な受信データが格納されていることを示します。

シフトレジスタで 1 フレームのデータ送受信後、

- ・ 送信モードでは、送信バッファに次の送信データがある場合 (TDRE フラグが 0)、自動的にシフトレジスタへ転送されます。
- ・ 受信モードでは、受信バッファに以前のデータがない場合 (RDRF フラグが 0)、自動的にシフトレジスタから受信バッファにデータが転送されます。

— ICMR I²C バスモードレジスタ (ICMR) アドレス : H'FFC7

ビット	ビット名	設定値	R/W	機能
7	MLS	0	R/W	MSB ファースト / LSB ファースト 選択 0 : MSB ファースト 1 : LSB ファースト I ² C バスフォーマットで使用する時は 0 に設定する
6	WAIT	0	R/W	ウェイト挿入ビット I ² C バスフォーマットでマスタモードの時のみ有効 WAIT=1 : データの最終ビットのクロックが立ち下がった後、ICCR の IRIC フラグが 1 にセットされ、ウェイト状態 (SCL=Low レベル) となる。 ICCR の IRIC フラグを 0 にクリアすることでウェイト状態を解除し、アクノリッジの転送を行う。 WAIT=0 : ウェイトは挿入されず、データとアクノリッジを連続して転送する。 ICCR の IRIC フラグは WAIT の設定に関わらず、アクノリッジの転送が完了した時点で 1 にセットされる。
5	CKS2	0	R/W	転送クロック選択 2~0 マスタモードの時のみ有効 TSCR レジスタの IICX との組み合わせで転送レートを選択する。
4	CKS1	0	R/W	
3	CKS0	1	R/W	
2	BC2	0	R/W	ビットカウンタ 2~0 次に転送するデータのビット数を指定する。I ² C バスフォーマットでは、データにアクノリッジ 1 ビットが加算されて転送される。設定は転送フレーム間で行う。また、000 以外の値を設定する場合は、SCL が Low 状態で行う。これらのビットはアクノリッジを含むデータ転送終了後、自動的に 000 に戻る。 I ² C バスフォーマット 000 : 9 ビット 001 : 2 ビット 010 : 3 ビット 011 : 4 ビット 100 : 5 ビット 101 : 6 ビット 110 : 7 ビット 111 : 8 ビット
1	BC1	0	R/W	
0	BC0	0	R/W	

— TSCR タイマシリアルコントロールレジスタ アドレス : H'FFFC

ビット	ビット名	設定値	R/W	機能
7~2	-	全て 1	-	リザーブビットである。読み出すと常に 1 が読み出される
1	IICRST	0	R/W	IICRST は、I ² C のレジスタを除くコントロール部をリセットする。I ² C 動作中に通信不具合等によりハングアップした時、IICRST ビットを 1 にセットするとポートの設定、レジスタの初期化をせずに I ² C のコントロール部をリセットすることができる。
0	IICX	0	R/W	ICMR の CKS2 ~ CKS0 との組み合わせで、マスタモードでの転送レートを選択する。

表 7 本タスクの転送レート

TSCR	ICMR			クロック	転送レート
ビット 0	ビット 5	ビット 4	ビット 3		
IICX	CKS2	CKS1	CKS0		=16MHz
0	0	0	1	/40	400kHz

— ABRKCR アドレスブレイクコントロールレジスタ アドレス：H'FFC8

ビット	ビット名	設定値	R/W	機能
7	RTINTE	1	R/W	RTE 割り込みイネーブル RTINTE = 0 : RTE 命令実行直後の割り込みをマスクし、必ず 1 命令実行する RTINTE = 1 : 割り込みはマスクされない
6 5	CSEL1 CSEL0	0 0	R/W R/W	コンディションセレクト 1~0 アドレスブレイクの条件を設定する CSEL1 = 0, CSEL0 = 0 : 命令実行サイクルに設定する
4 3 2	ACMP2 ACMP1 ACMP0	0 0 0	R/W R/W R/W	アドレスコンペア 2~0 BAR と内部アドレスバスの比較条件を設定する ACMP2 = 0, ACMP1 = 0, ACMP0 = 0 : 16 ビット比較をする
1 0	DCMP1 DCMP0	0 0	R/W R/W	データコンペア 1~0 BDR と内部データバスの比較条件を設定する DCMP1 = 0, DCMP0 = 0 : データを比較しません

— ABRKSR アドレスブレイクステータスレジスタ アドレス：H'FFC9

ビット	ビット名	設定値	R/W	機能
7	ABIF	0	R/W	アドレスブレイク割り込み要求フラグ ABIF = 1 : ABRKCR で設定された条件が成立した時 ABIF = 0 : 初期値、1 の状態をリードした後、0 をライトした時
6	ABIE	1	R/W	アドレスブレイク割り込み要求イネーブル設定 ABIE = 0 : アドレスブレイク割り込み要求マスク ABIE = 1 : アドレスブレイク割り込み要求をイネーブルに設定
5~0	-	全て 1	-	リザーブビットである。リードすると常に 1 が読み出される。

— BAR ブレイクアドレスレジスタ アドレス：H'FFCA

(BARH ブレイクアドレスレジスタ H アドレス：H'FFCA)

(BARL ブレイクアドレスレジスタ L アドレス：H'FFCB)

機能：アドレスブレイク割り込み処理が発生するアドレスを 16 ビットで設定

— PMR1 ポートモードレジスタ 1 アドレス：H'FFE0

ビット	ビット名	設定値	R/W	機能
5	IRQ1	1	R/W	P15 / IRQ1 端子の機能を選択 0：汎用入出力ポート 1：IRQ1 入力端子

— IEGR1 割り込みエッジセレクトレジスタ 1 アドレス：H'FFF2

ビット	ビット名	設定値	R/W	機能
1	IEG1	1	R/W	IRQ1 エッジセレクト 0：IRQ1 端子入力の立ち下がりエッジを検出 1：IRQ1 端子入力の立ち上がりエッジを検出

— IENR1 割り込みイネーブルレジスタ アドレス：H'FFF4

ビット	ビット名	設定値	R/W	機能
1	IEN1	1	R/W	IRQ1 割り込み要求イネーブル このビットを1にセットするとIRQ1 端子割り込み要求がイネーブルになる

— IRR1 割り込みフラグレジスタ 1 アドレス：H'FFF6

ビット	ビット名	設定値	R/W	機能
1	IRRI1	0	R/W	IRQ1 割り込み要求フラグ [セット条件] IRQ1 端子が割り込み入力に設定され、指定したエッジを検出した時 [クリア条件] 0 をライトした時

— PCR7 ポートコントロールレジスタ 7 アドレス：H'FFEA

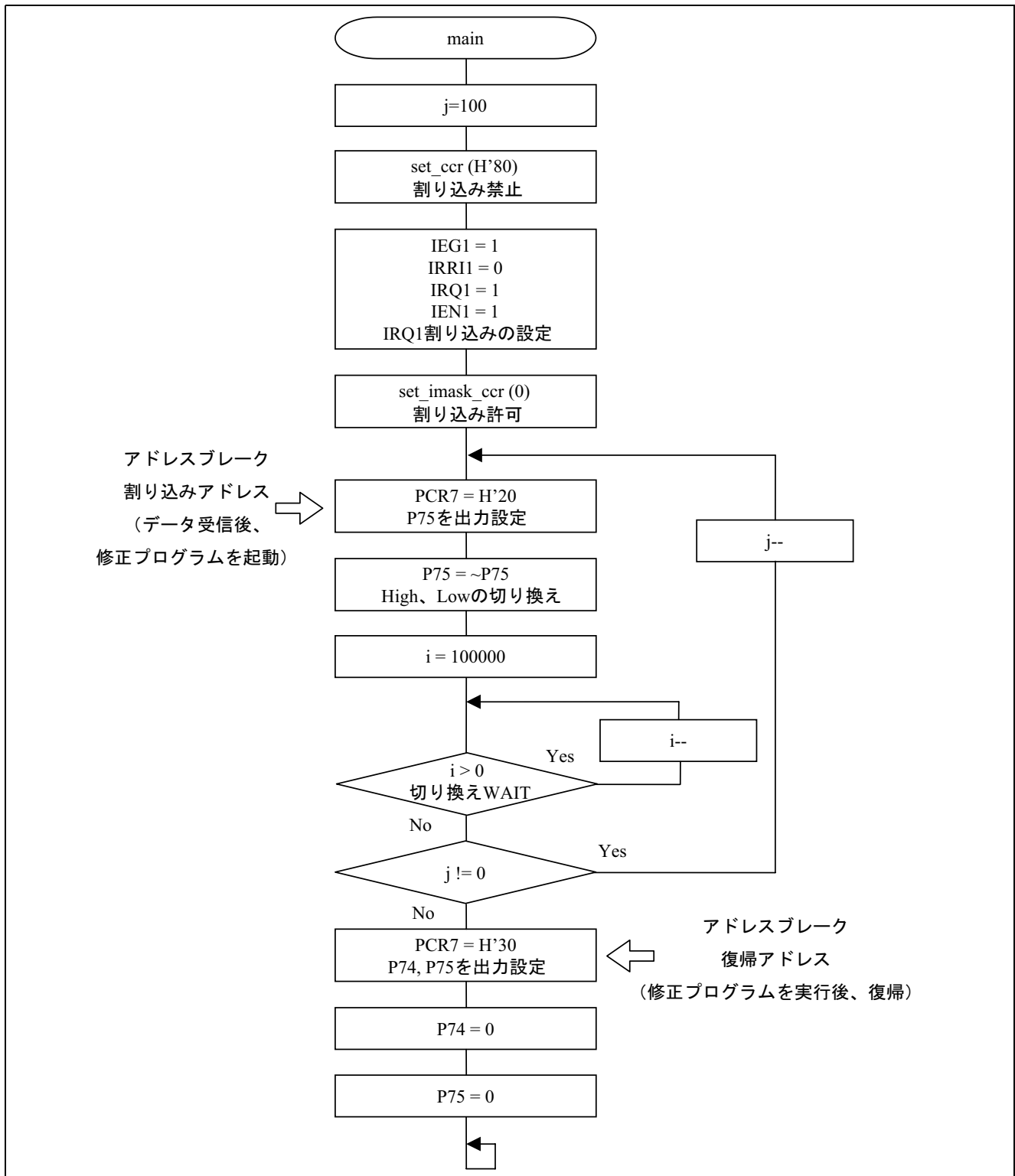
ビット	ビット名	設定値	R/W	機能
5	PCR75	0	W	このビットを1にセットすると対応する端子は出力ポートとなり、0にクリアすると入力ポートになる。
4	PCR74	0	W	

— PDR7 ポートデータレジスタ 7 アドレス：H'FFDA

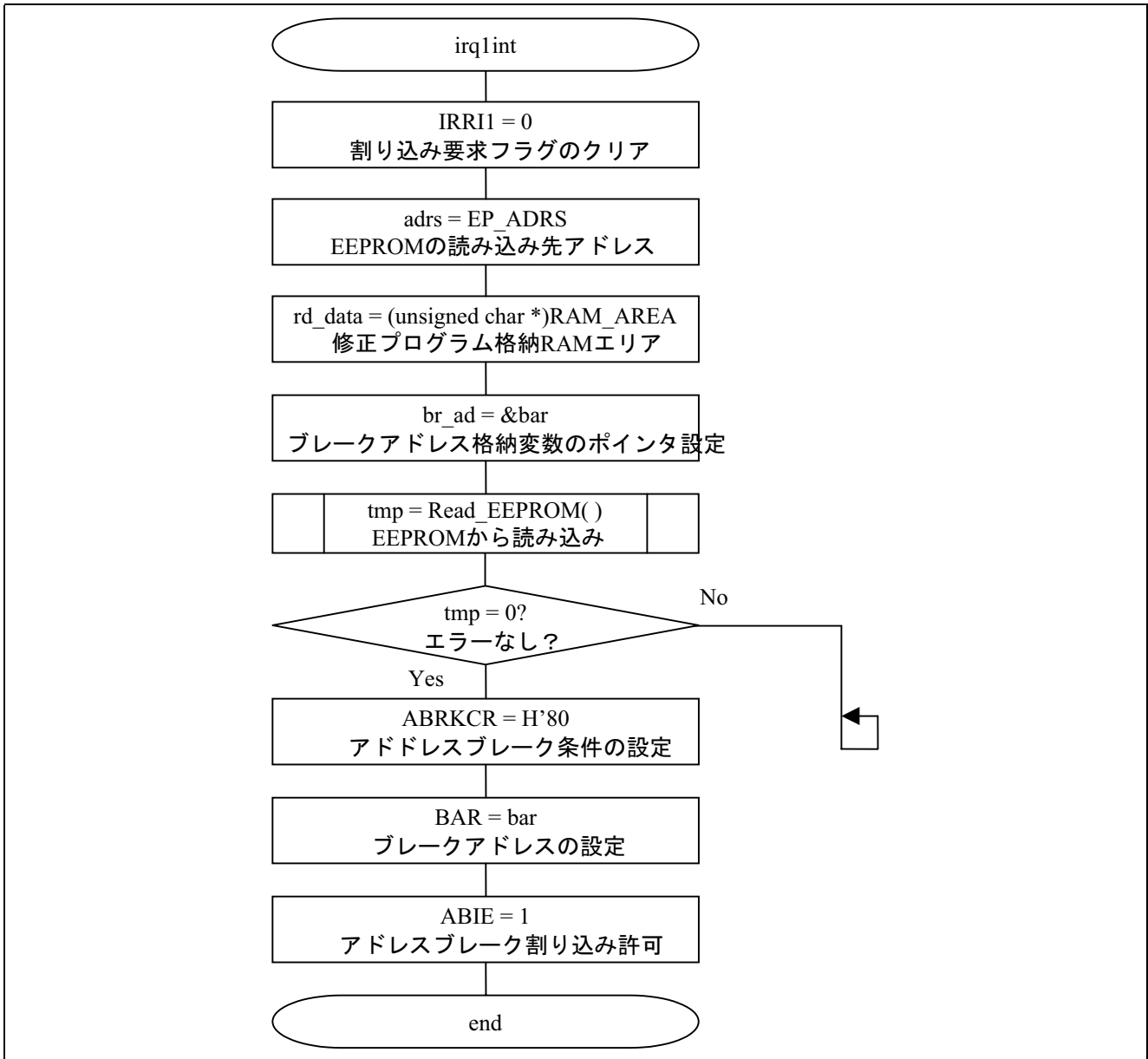
ビット	ビット名	設定値	R/W	機能
5	P75	0	R/W	汎用出力ポートの出力値を格納する。 このレジスタをリードすると、PCR7 がセットされているビットはこのレジスタの値が読み出される。PCR7 がクリアされているビットはこのレジスタの値に関わらず、端子の状態が読み出される。
4	P74	0	R/W	

5. フローチャート

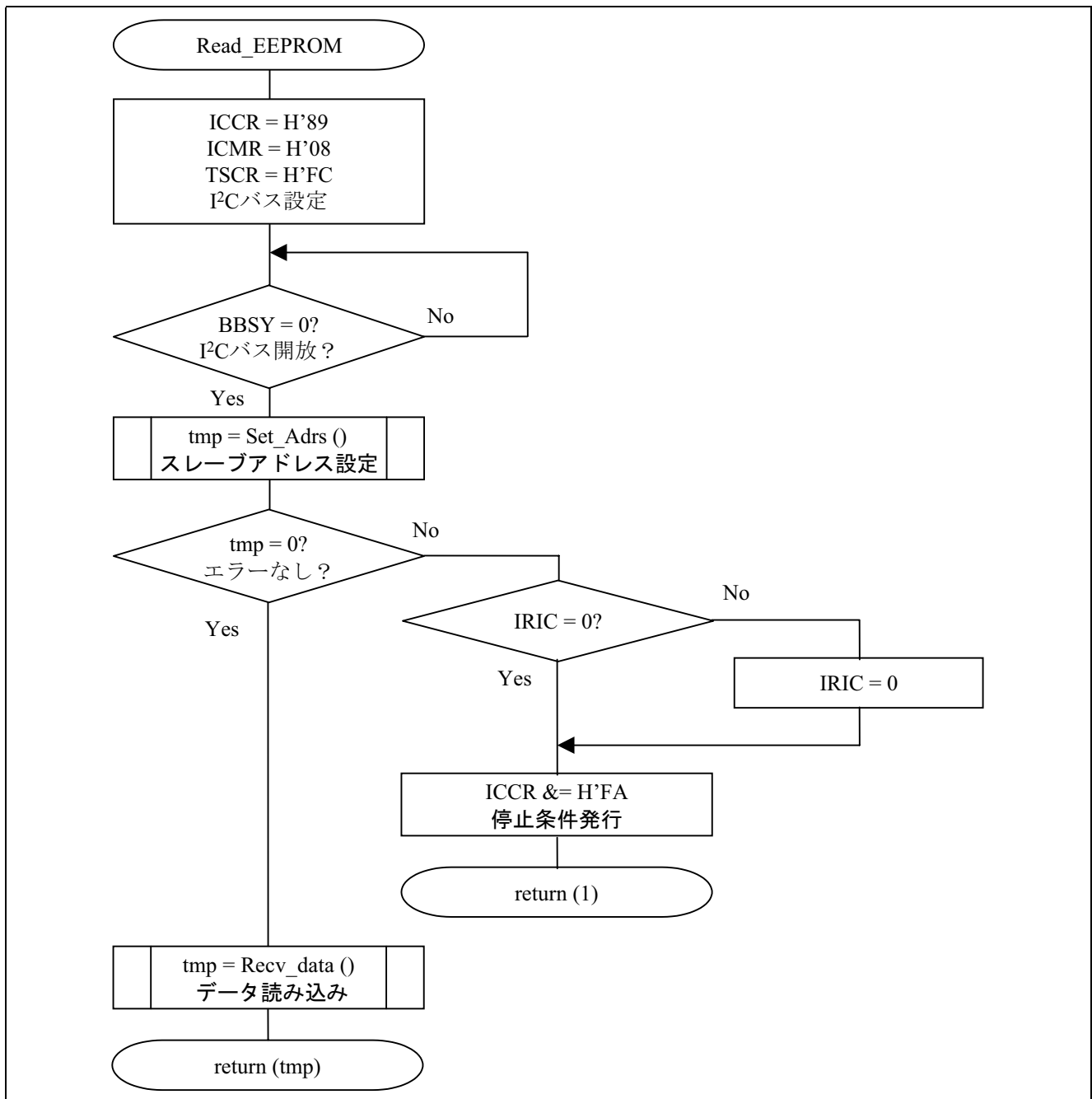
5.1 メインルーチン



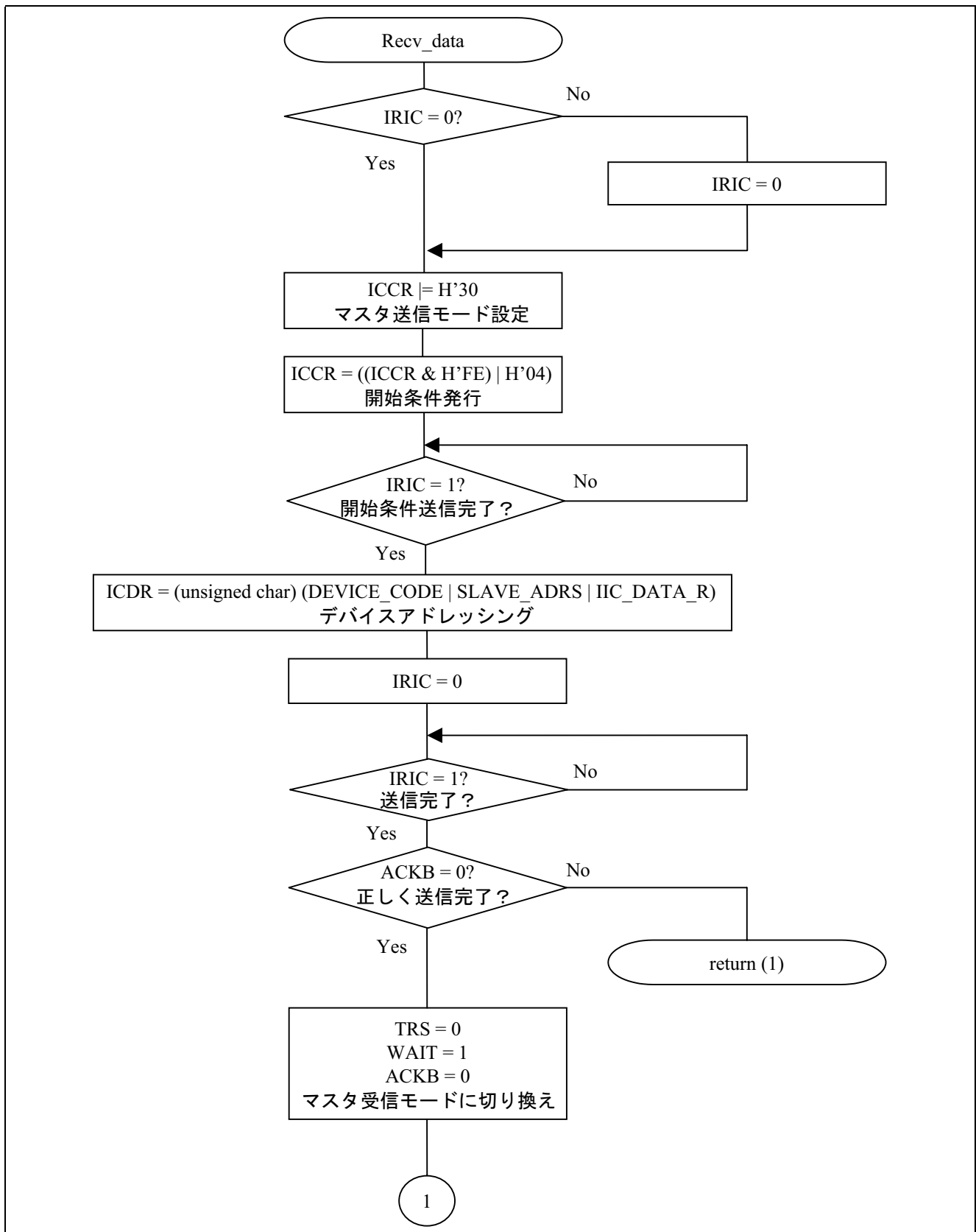
5.2 IRQ1 割り込みルーチン

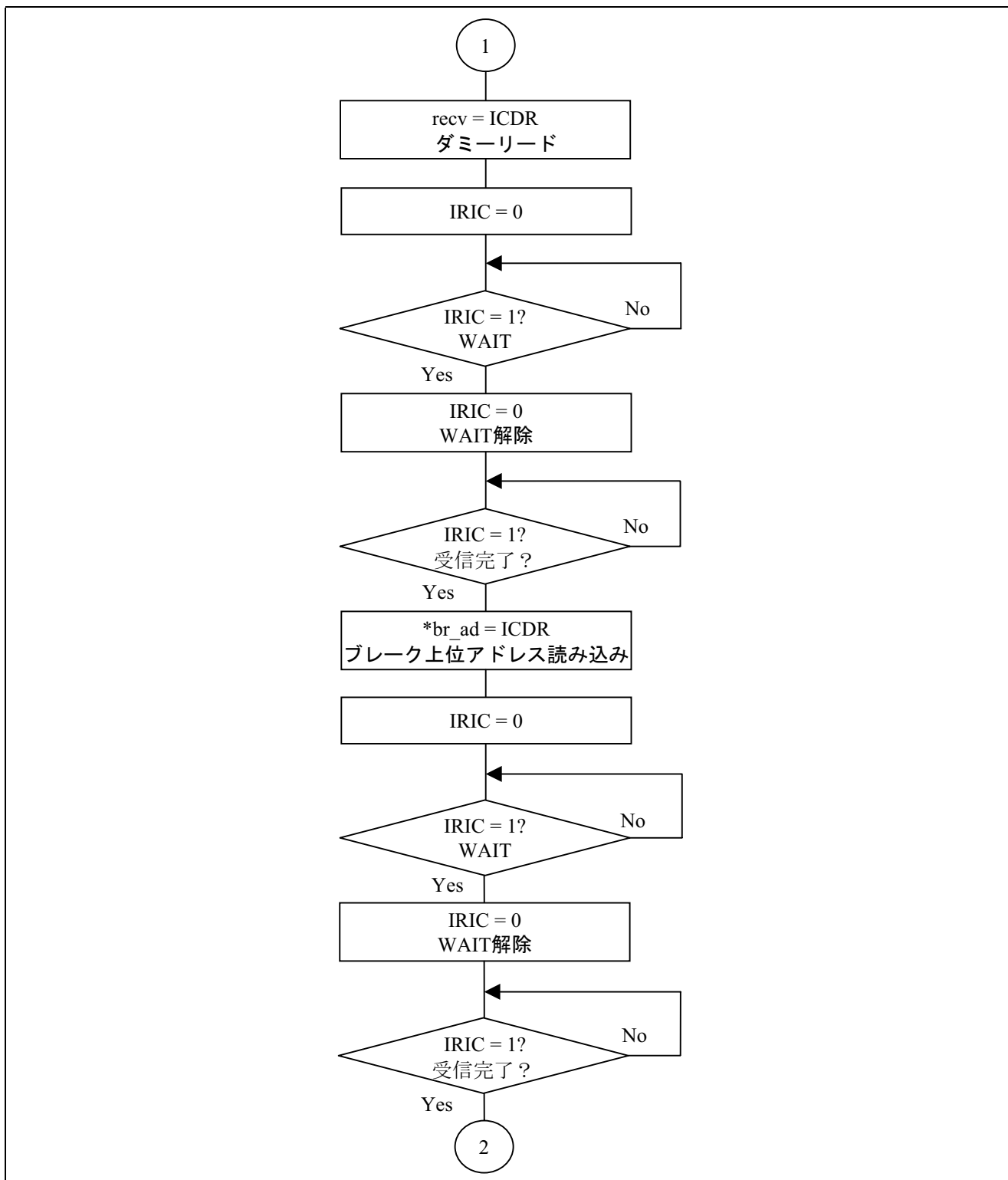


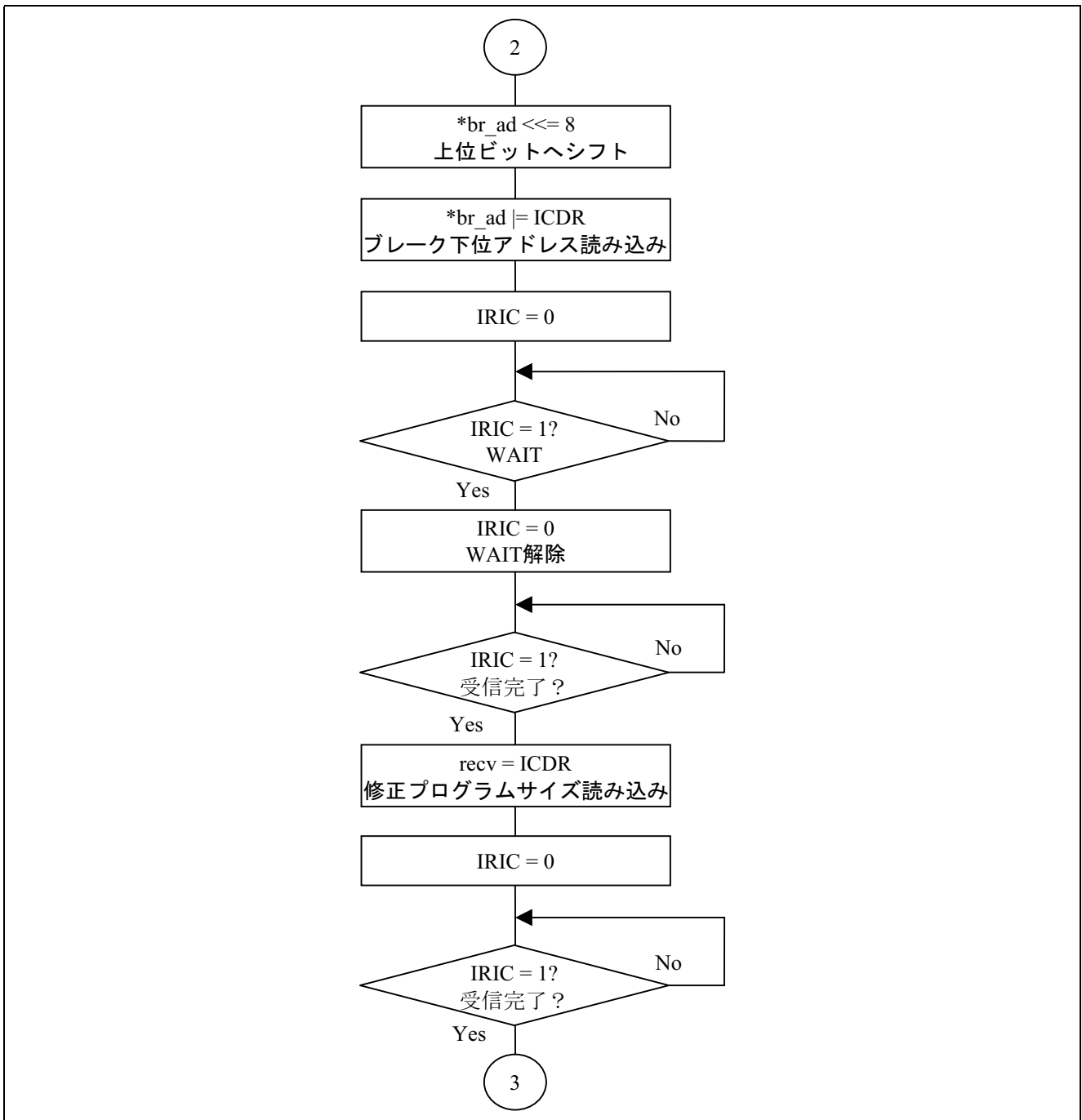
5.3 ダウンロード設定ルーチン

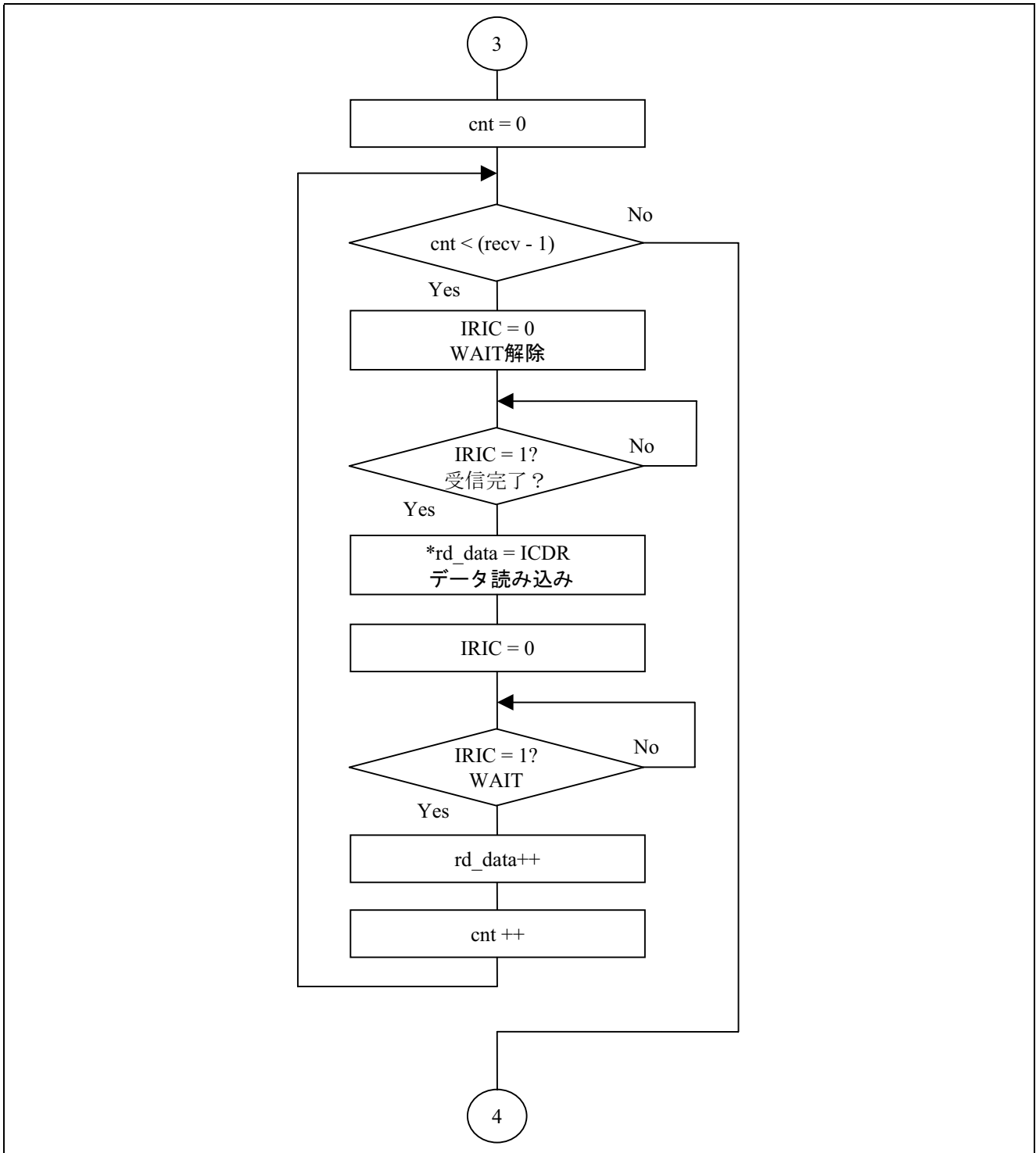


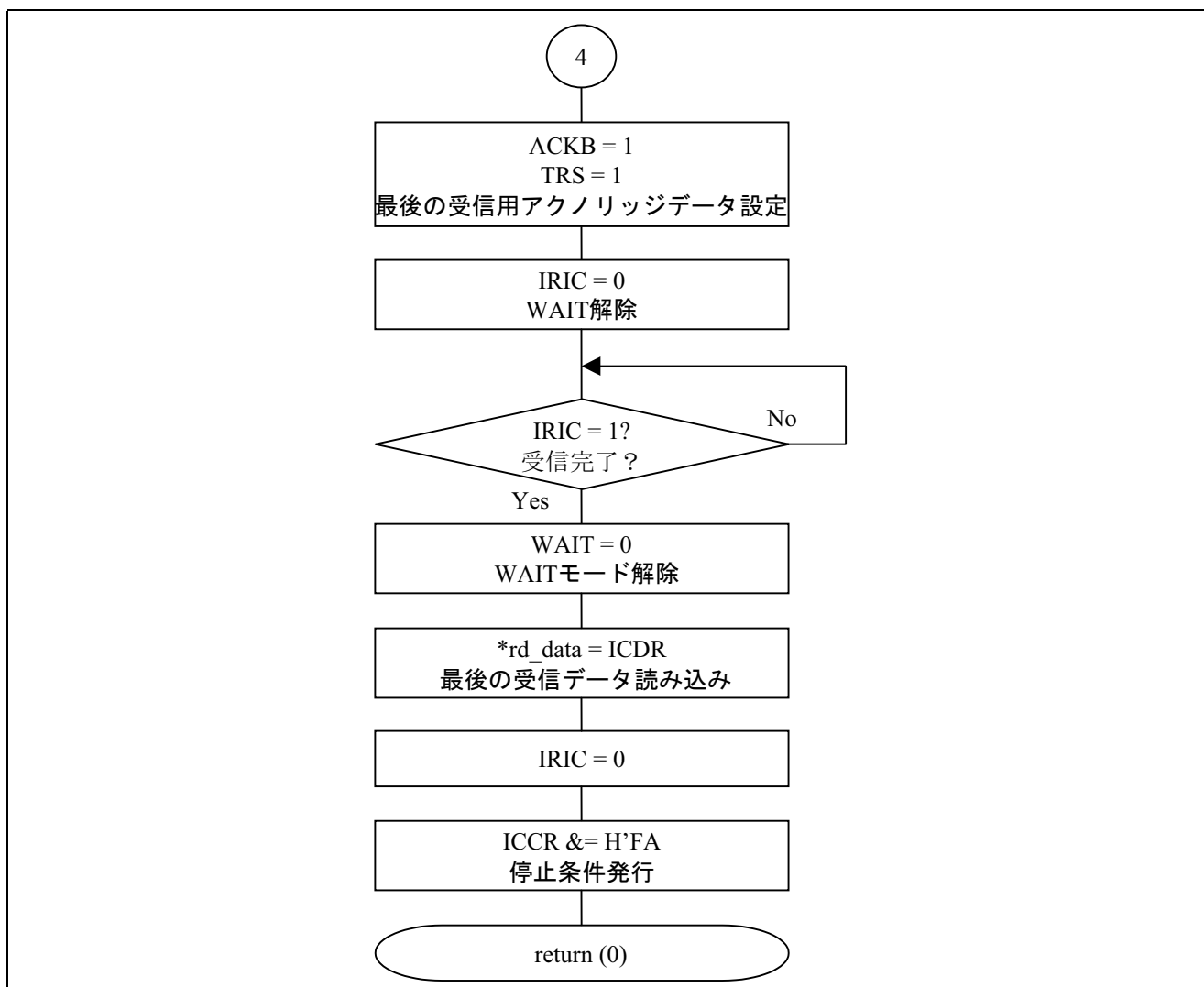
5.4 ダウンロードルーチン



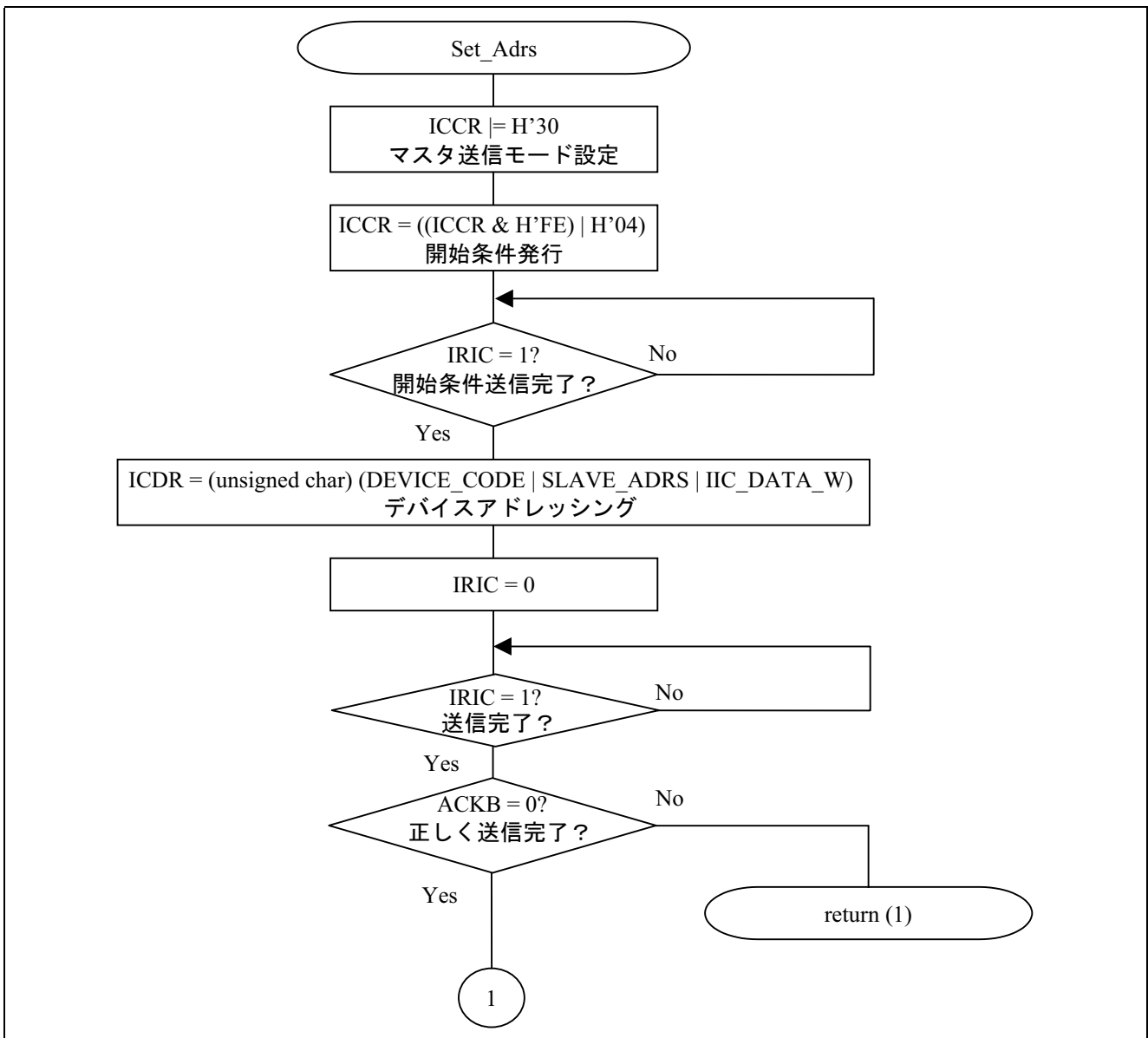


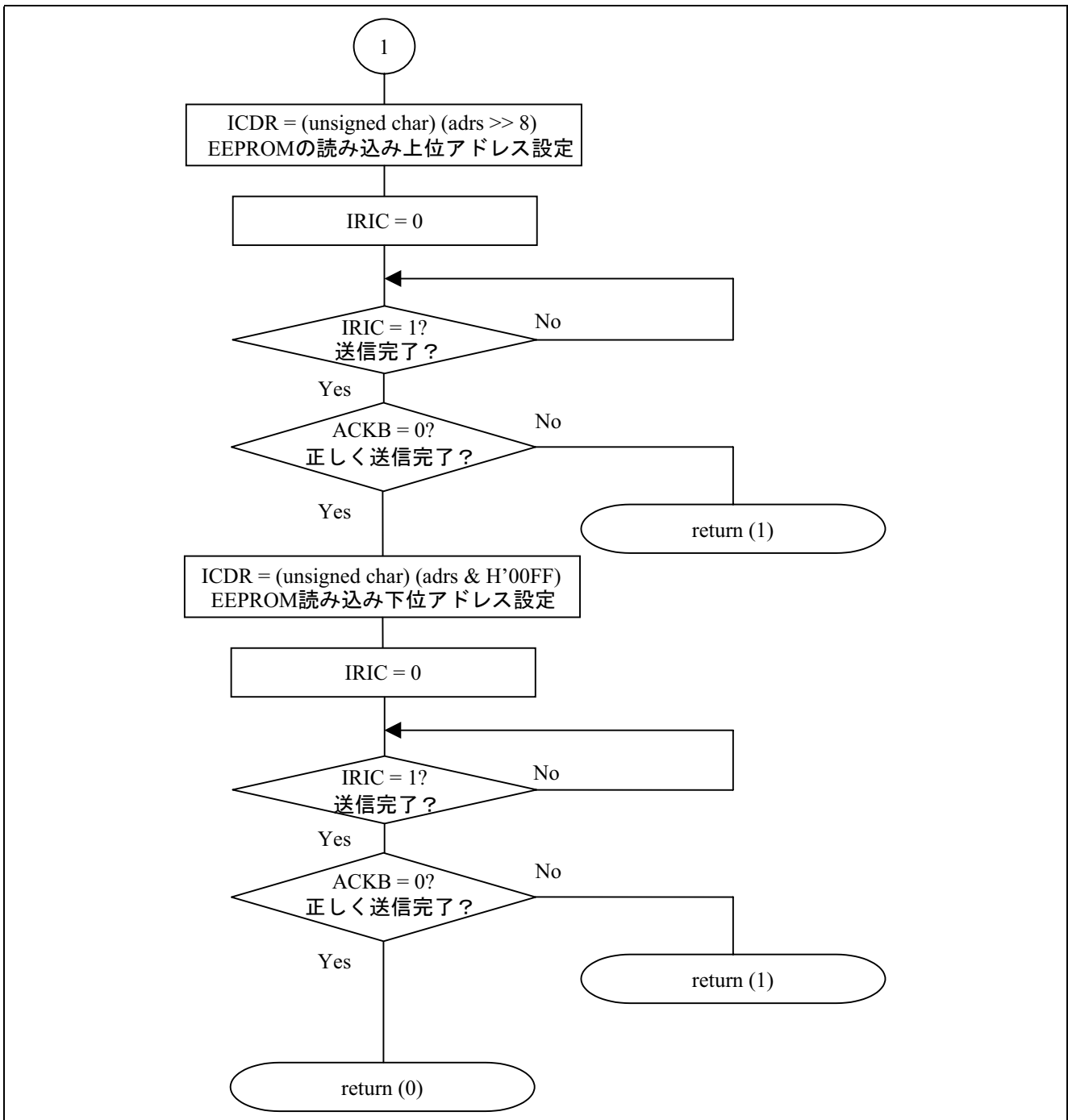






5.5 スレーブアドレス設定ルーチン





5.6 リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'0018
CV3	H'001E
P	H'0100

6. プログラムリスト

```

/*****/
/*
/* H8/300HN Series -H8/3664-
/* Application Note
/*
/* 'ROM CORRECTION'
/*
/* External Clock : 16MHz
/* Internal Clock : 16MHz
/* Sub Clock      : 32.768kHz
/*
/*****/

#include <machine.h>

/*****/
/* Symbol Definition
/*****/
struct BIT {
    unsigned char  b7:1;    /* bit7 */
    unsigned char  b6:1;    /* bit6 */
    unsigned char  b5:1;    /* bit5 */
    unsigned char  b4:1;    /* bit4 */
    unsigned char  b3:1;    /* bit3 */
    unsigned char  b2:1;    /* bit2 */
    unsigned char  b1:1;    /* bit1 */
    unsigned char  b0:1;    /* bit0 */
};

#define ABRKCR      *(volatile unsigned char *)0xFFC8 /* Address Break Control
Register */
#define ABRKSR_BIT (*(struct BIT *)0xFFC9)           /* Address Break Status
Register */
#define ABIE        ABRKSR_BIT.b6 /* Address Break Interrupt Enable */
#define BAR         *(volatile unsigned short *)0xFFCA /* Break Address Register
H */

#define PCR7        *(volatile unsigned char *)0xFFEA
#define PCR7_BIT    (*(struct BIT *)0xFFEA)          /* Port Control Register
7 */
#define PDR7        *(volatile unsigned char *)0xFFDA
#define PDR7_BIT    (*(struct BIT *)0xFFDA)          /* Port Data Register 7
*/
#define P75         PDR7_BIT.b5 /* Port Data Register 7 bit5 */
#define P74         PDR7_BIT.b4 /* Port Data Register 7 bit4 */

#define PDR5        *(volatile unsigned char *)0xFFD8
#define PDR5_BIT    (*(struct BIT *)0xFFD8)          /* Port Data Register 7
*/
#define SCL         PDR5_BIT.b7 /* IIC serial clock input output */
#define SDA         PDR5_BIT.b6 /* IIC serial data input output */

#define ICCR        *(volatile unsigned char *)0xFFC4

```

```

#define ICCR_BIT (*(struct BIT *)0xFFC4) /* IIC bus control
register */
#define TRS ICCR_BIT.b4 /* Transmission, reception choice */
#define BBSY ICCR_BIT.b2 /* Bus busy */
#define IRIC ICCR_BIT.b1 /* IIC bus interface interrupt,
requirement flag */
#define SCP ICCR_BIT.b0 /* start/stop condition prohibition bit */
#define ICSR *(volatile unsigned char *)0xFFC5
#define ICSR_BIT (*(struct BIT *)0xFFC5) /* IIC bus status register
*/
#define ACKB ICCR_BIT.b0 /* Acknowledge bit */
#define ICDR *(volatile unsigned char *)0xFFC6
#define ICDR_BIT (*(struct BIT *)0xFFC6) /* IIC bus data register
*/
#define ICMR *(volatile unsigned char *)0xFFC7
#define ICMR_BIT (*(struct BIT *)0xFFC7) /* IIC bus mode register
*/
#define WAIT ICMR_BIT.b6 /* WAIT insertion bit */
#define TSCR *(volatile unsigned char *)0xFFFFC
#define TSCR_BIT (*(struct BIT *)0xFFFFC) /* Timer serial control
register */

#define PMR1 *(volatile unsigned char *)0xFFE0 /* Port Mode Register 1
*/
#define PMR1_BIT (*(struct BIT *)0xFFE0) /* Prot Mode Register 1
*/
#define IRQ1 PMR1_BIT.b5 /* P15/IRQ1 Select */

#define IEGR1_BIT (*(struct BIT *)0xFFFF2) /* Interrupt Edge Select
Register 1 */
#define IEGR1 IEGR1_BIT.b1 /* IEGR1 Edge Select */
#define IENR1_BIT (*(struct BIT *)0xFFFF4) /* Interrupt Enable
Register 1 */
#define IEN1 IENR1_BIT.b1 /* IEN1 Inetrrupt Enable */
#define IRR1_BIT (*(struct BIT *)0xFFFF6) /* Interrupt Request
Register 1 */
#define IRR1 IRR1_BIT.b1 /* IRR1 Interrupt Request Register */

#define DEVICE_CODE 0xA0 /* EEPROM DEVICE CODE:1010 */
#define SLAVE_ADRS 0x00 /* SLAVE ADRS:000 */
#define IIC_DATA_W 0x00 /* WRITE_DATA:0 */
#define IIC_DATA_R 0x01 /* READ_DATA:1 */

#define EP_ADRS 0x0000 /* The first address to read EEPROM */
#define RAM_AREA 0xFC00 /* Modication program storage address */

#pragma interrupt (irqlint)
/*****/
/* Function define */
/*****/
void main ( void );
unsigned char Read_EEPROM(unsigned short adrs, unsigned char *rd_data, unsigned
short *br_ad);
unsigned char Recv_data(unsigned char *rd_data, unsigned short *br_ad);
unsigned char Set_Adrs(unsigned short adrs);

```

```

void irqlint(void);

/*****/
/* Vector Address */
/*****/
#pragma section V1 /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = {
    main
};

#pragma section V2 /* VECTOR SECTION SET */
void (*const VEC_TBL2[])(void) = {
    (void *)RAM_AREA /* Address Break */
};

#pragma section V3 /* VECTOR SECTION SET */
void (*const VEC_TBL3[])(void) = {
    irqlint /* IRQ1 interrupt */
};

#pragma entry main(sp=0xFF80)
#pragma section /* P */
/*****/
/* Main Program */
/*****/
void main ( void )
{
    volatile unsigned long i;
    unsigned char j=100;

    set_ccr(0x80); /* Initialize CCR/Interrupt Disable */

    IEG1 = 1; /* Initialize IRQ1 Terminal Input Edge */
    IRR11 = 0; /* Initialize IRQ1 Interrupt Request Flag */
    IRQ1 = 1;
    IEN1 = 1; /* IRQ1 Interrupt Enable */

    set_imask_ccr(0);

    do{
        PCR7 = 0x20; /* P75 output set */
        P75 = ~P75; /* LED2 ON/OFF */
        for(i=100000; i>0; i--); /* wait */
    }while(j--);

    PCR7 = 0x30; /* P74, P75 output set */
    P74 = 0; /* LED2 ON */
    P75 = 0; /* LED3 ON */
    while(1);
}

/*****/
/* IRQ1 Interrupt */
/*****/
void irqlint( void )

```

```

{
    unsigned short adrs, bar, *br_ad;
    unsigned char tmp, *rd_data;

    IRR11 = 0;                /* Initialize IRQ1 Interrupt Request Flag */

    adrs = EP_ADRS;          /* initialize EEPROM read address */
    rd_data = (unsigned char *)RAM_AREA; /* modification program start
address */
    br_ad = &bar;           /* Break Address */
    tmp = Read_EEPROM(adrs, rd_data, br_ad); /* modification program read */
    if(tmp != 0)
        while(1);

    ABRKCR = 0x80;          /* A setup of Address Break condition */
    BAR = bar;              /* A setup of Address Break */
    ABIE = 1;               /* A setup of Address Break Enable */
}

```

```

unsigned char Read_EEPROM(unsigned short adrs, unsigned char *rd_data, unsigned
short *br_ad)

```

```

{
    unsigned char tmp;

    ICCR = 0x89;            /* ICE=1, ACKE=1, SCP=1 */
    ICMR = 0x08;           /* CKS0=1 */
    TSCR = 0xFC;           /* trase rate 400kHz */

    while(BBSY != 0);      /* Bus Busy? */

    tmp = Set_Adrs(adrs);   /* Address set */
    if(tmp != 0){
        if(IRIC == 1)
            IRIC = 0;
        ICCR &= 0xFA;      /* stop condition */
        return(1);
    }

    tmp = Recv_data(rd_data, br_ad); /* modification program data read */
    return(tmp);
}

```

```

unsigned char Recv_data(unsigned char *rd_data, unsigned short *br_ad)

```

```

{
    unsigned char recv, cnt=0;

    if(IRIC ==1)
        IRIC = 0;
    ICCR |= 0x30;          /* master trase mode (MST=1, TRS=1) */
    ICCR = ((ICCR & 0xFE) | 0x04); /* start condition */
    while(IRIC == 0);

    ICDR = (unsigned char)(DEVICE_CODE | SLAVE_ADRS | IIC_DATA_R);
    /* slave address set */
    IRIC = 0;
}

```



```

while(IRIC == 0);          /* trase OK?          */
if(ACKB != 0)             /* ACK?              */
    return(1);

TRS = 0;                  /* Master Receive    */
WAIT = 1;                 /* WAIT mode ON      */
ACKB = 0;                 /* set ACK = 0       */

recv = ICDR;              /* dummy read        */
IRIC = 0;
while(IRIC == 0);        /* dummy recv end?   */

IRIC = 0;                 /* WAIT OFF          */
while(IRIC == 0);        /* dummy recv OK?    */

*br_ad = ICDR;           /* Break high Address receive */
IRIC = 0;
while(IRIC == 0);        /* receive end?      */

IRIC = 0;                 /* WAIT OFF          */
while(IRIC == 0);        /* receive OK?       */

*br_ad <<= 8;            /* high bit shift    */
*br_ad |= ICDR;          /* Break low Address receive */
IRIC = 0;
while(IRIC == 0);        /* receive end?      */

IRIC = 0;                 /* WAIT OFF          */
while(IRIC == 0);        /* receive OK?       */

recv = ICDR;              /* modication program size receive */
IRIC = 0;
while(IRIC == 0);        /* receive end?      */

while(cnt < (recv - 1)){ /* (size - 1) LOOP   */
    IRIC = 0;             /* WAIT OFF          */
    while(IRIC == 0);    /* receive OK?       */

    *rd_data = ICDR;     /* modication program receive */
    IRIC = 0;
    while(IRIC == 0);    /* receive end?      */

    rd_data++;
    cnt++;
}

ACKB = 1;                 /* ACK=1            */
TRS = 1;                 /* trase mode set    */
IRIC = 0;                 /* WAIT OFF          */
while(IRIC == 0);        /* receive OK?       */

WAIT = 0;                 /* WAIT mode OFF     */
*rd_data = ICDR;         /* Last data receive */
IRIC = 0;

```

```

    ICCR &= 0xFA;                /* stop condition          */

    return(0);
}

unsigned char Set_Adrs(unsigned short adrs)
{
    ICCR |= 0x30;                /* master trase mode (MST=1, TRS=1) */
    ICCR = ((ICCR & 0xFE) | 0x04); /* strat condition          */
    while(IRIC == 0);

    ICDR = (unsigned char)(DEVICE_CODE | SLAVE_ADRS | IIC_DATA_W); /* slave address set          */
    IRIC = 0;
    while(IRIC == 0);           /* trase end?                */

    if(ACKB != 0){              /* ACK OK?                    */
        return(1);
    }

    ICDR = (unsigned char)(adrs >> 8); /* high address set          */
    IRIC = 0;
    while(IRIC == 0);           /* trase end?                */

    if(ACKB != 0){              /* ACK OK?                    */
        return(1);
    }

    ICDR = (unsigned char)(adrs & 0x00FF); /* low address set          */
    IRIC = 0;
    while(IRIC == 0);           /* trase end?                */

    if(ACKB != 0){              /* ACK OK?                    */
        return(1);
    }

    return(0);
}

```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.22	—	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留意ください。

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すは、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、慎重に制作したのですが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。