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ものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、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。

H8/300L シリーズ
EEPROM とのインタフェース

要旨

SCL(Serial Clock)と SDA(Serial Data)の 2 線により接続された EEPROM(HN58X2408)との 1：1 のインタフェースを行ないます。LSB ファースト方式による転送を行ないます。

動作確認デバイス

H8/38024

目次

1. 仕様

(1) 図 1 に示すように，SCL(Serial Clock)と SDA(Serial Data)の 2 線により接続された EEPROM (HN58X2408) との 1：1 のインタフェースを行ないます。
(2) R0M 上のデータを EEPOROM にライトし, EEPROM にライトされたデータを再び RAM にリードします。
(3) 転送データは，ポート 7_3 に接続された LED を点灯させるプログラムです。
(4) データの LSB(最下位ビット)よりデータを送信/受信する LSB ファースト方式による転送を行ないます。

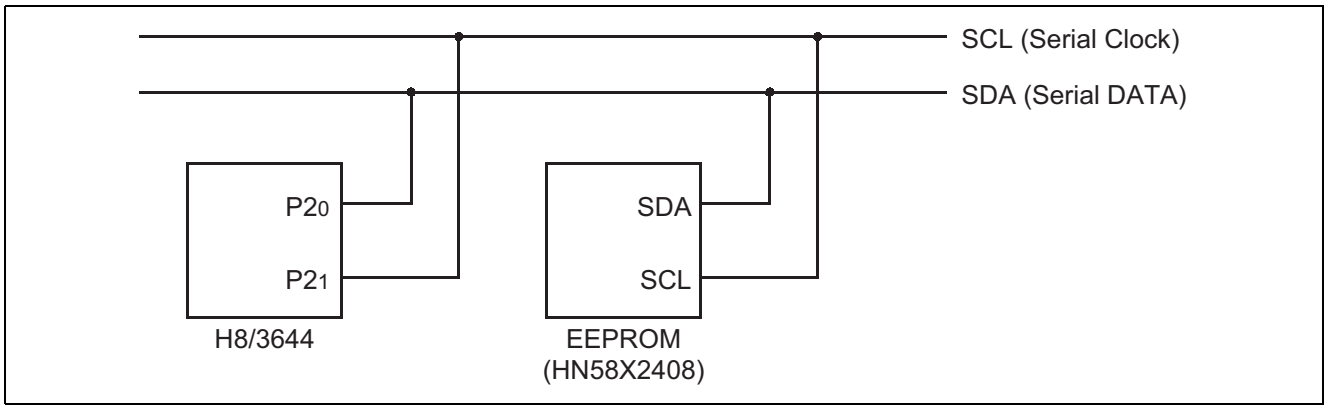


図 1　EEPROM との接続例

2. 考え方

(1) 図 2 に本タスク例で使用する EEPROM(HN58X2408)のバス・タイミングを示します。

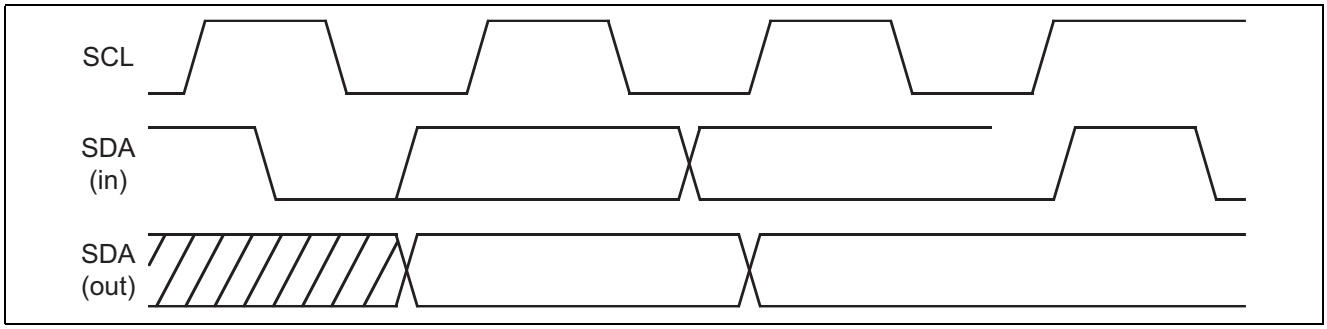


図 2　EEPROM(HN58X2408)バス・タイミング

(2) 使用する EEPROM のバス・タイミングが図 2 に示すようなバス・タイミングのため，本タスク例では，ソフトウェアにより SCL に出力するクロックを $P2_1$ 端子レベルを"High"，"Low"に設定することによりシリアル・クロックを生成しています。ソフトウェアにより生成したシリアル・クロックに同期して $P2_0$ 端子より，シリアル・データを出力/入力することにより EEPROM(HN58X2408)とのインタフェースを実現しています。図 3 に $P2_1$ 端子，および $P2_0$ 端子レベルのタイミング波形を示します。

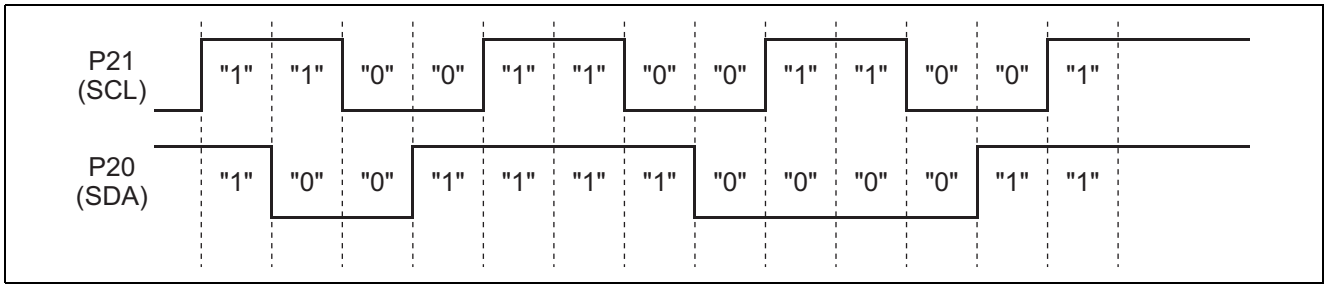


図 3　$P2_1$ 端子，および $P2_0$ 端子レベルのタイミング波形

3. 使用機能説明

(1) 本タスク例では，図 4 に示すように H8/3644 と EEPROM(HN58X2408)を接続し，インタフェースを行ないます。また，表 1 に EEPROM(HN58X2408)のピン説明を示します。

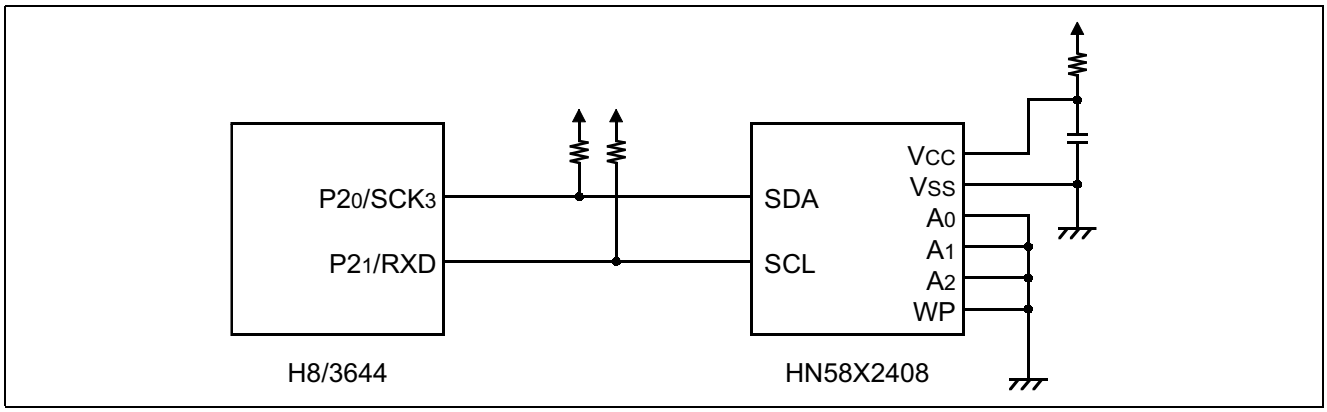


図 4　H8/3644 と HN58X2408 の接続図

表 1　HN58X2408 ピン説明

Pin 名称	機能
A0～A2	デバイス・アドレス
SCL	シリアル・クロック入力
SDA	シリアル・データ入力/出力
WP	ライト・プロテクト
V_{CC}	V_{CC}
V_{SS}	GND

(2) 図 5 に示すようなブロック図により，H8/3644 と EEPROM(HN58X2408)のインタフェースを行ないます。以下に H8/3644 の機能について説明します。

(a) ポート 2 は，3 ビットの入出力ポートで，$P2_0$/SCK_3，$P2_1$/RXD，$P2_2$/TXD の 3 つの端子により構成されています。

(b) $P2_0$/SCK_3 端子を HN58X2408 の SDA 端子に接続し，シリアル･データの入出力端子として使用します。

(c) $P2_1$/RXD 端子を HN58X2408 の SCL 端子に接続し，シリアル･クロックの出力端子として使用します。

(d) ポート 7 は，5 ビットの入出力ポートで，$P7_3$，$P7_4$/TMRIV，$P7_5$/TCIV，$P7_6$/TMOV，$P7_7$ の 5 つの端子により構成されています。

(e) $P7_3$ 端子を LED への出力端子として使用します。

(f) $P2_0$/SCK_3 端子は，シリアルコントロールレジスタ 3(SCR3)のクロックイネーブル 1(CKE1)を"0"に，クロックイネーブル 0(CKE0)を"0"に，シリアルモードレジスタ(SMR)のコミュニケーションモード(COM)を"0"に設定することにより $P2_0$ 入出力端子として機能します。また，ポートコントロールレジスタ 2(PCR2)のポートコントロールレジスタ 2_0($PCR2_0$)を"0"に設定すると $P2_0$ 入力端子として，$PCR2_0$ を"1"に設定すると $P2_0$ 出力端子として機能します。

(g) $P2_1$/RXD 端子は，SCR3 のレシーブイネーブル(RE)を"0"に設定することにより $P2_1$ 入出力端子として機能します。また，PCR2 のポートコントロールレジスタ 21($PCR2_1$)を"0"に設定すると $P2_1$ 入力端子として，$PCR2_1$ を"1"に設定すると $P2_1$ 出力端子として機能します。

(h) $P7_3$ 端子は，ポートコントロールレジスタ 7(PCR7)のポートコントロールレジスタ 7_3($PCR7_3$)を"1"に設定することにより $P7_3$ 出力端子として機能します。

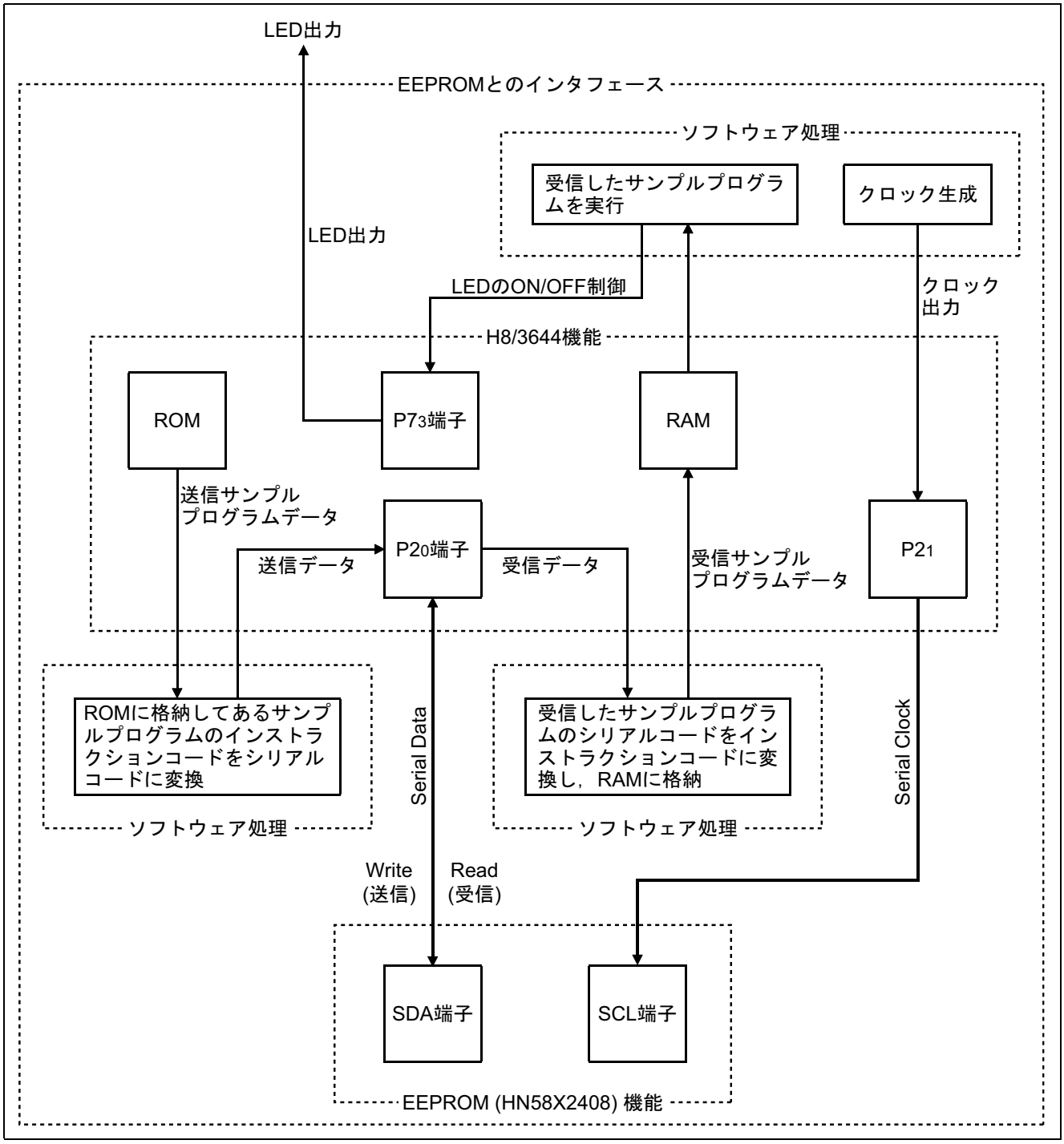


図 5　EEPROM **とのインタフェースのブロック図**

(2) 表 2 に本タスク例の機能割付けを示します。表 2 に示すように H8/3644 の機能を割付け，EEPROM とのインタフェースを行ないます。

表 2　機能割付け

H8/3644 機能	機能
$P2_0$ 端子	シリアル・データ入出力
$P2_0$	$P2_0$ 端子のデータを格納
$PCR2_0$	$P2_0$ 入出力端子機能の設定
$P2_1$ 端子	シリアル・クロック出力
$P2_1$	$P2_1$ 端子のデータを格納
$PCR2_1$	$P2_1$ 入出力端子機能の設定
$P7_3$ 端子	LED 出力
$P7_3$	$P7_3$ 端子のデータを格納
$PCR7_3$	$P7_3$ 入出力端子機能の設定

(4) 以下に本タスク例で使用している EEPROM(HN58X2408)の仕様について説明します。

(a) 本タスク例で使用する EEPROM(HN58X2408)は, 2 線式シリアルインタフェースの EEPROM(電気的に書き換え可能な ROM)で, 最新の NMOS メモリ技術, CMOS プロセスおよび低電圧回路技術を採用し, 低電源電圧動作・低消費電力・高速動作・高信頼性を実現した EEPROM です。32 バイトページ書き換え機能により，データ書き換えが高速化されています。

(b) 以下に本タスク例で使用している EEPROM(HN58X2408)の特徴を示します。

- 単一電源：1.8V～5.5V
- 2 線式シリアルインタフェース
- 動作周波数：400kHz
- 消費電流
  —スタンバイ時：3 μ A(max)
  —読み出し時：1mA(max)
  —書き換え時：3mA(max)
- ページ書き換え：ページサイズ 32 バイト
- 書き換え時間：10ms(2.7V 以上)/ 15ms(1.8V～2.7V)
- 書き換え回数：105 回(ページ書き換え時)

(c) Read，Write の動作を開始するには，SCL 入力が High の期間に，SDA 入力を High から Low にするスタート・コンディションにする必要があります。また，SDA 入力が High の期間に，SDA 入力を Low から High にすることで，ストップ・コンディションになります。Read の場合，ストップ・コンディションを入力すると Read が終了し，スタンバイ状態になります。Write の場合は，ストップ・コンディション入力で書き換えデータの入力終了となり，メモリへの書き込みを書き換え時間(t_{WC})の期間実施した後，スタンバイモードになります。図 6 にスタート・コンディション，ストップ・コンディションのタイミング波形を示します。

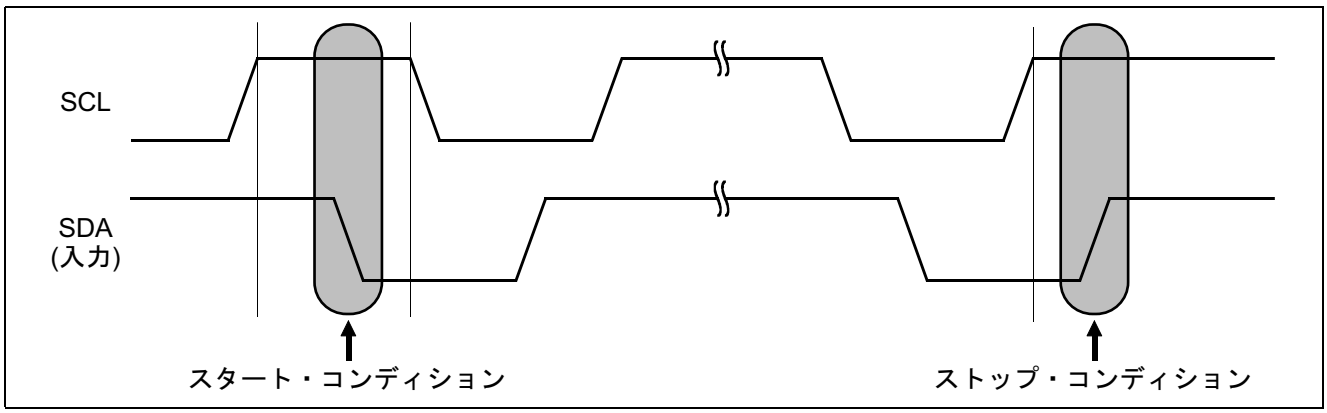


図 6　スタート・コンディション，およびストップ・コンディション信号出力タイミング波形

(d) アドレス情報，Read 情報等のシリアル・データは，8 ビット単位で送受信が行われます。この 8 ビットのデータが正常に送信または受信されたことを示すのが Acknowledge 信号で，SCL の 9 クロック目に受信側が"0"を出力します。送信側は，この 9 クロック目で Acknowledge 信号を受信するために，バスを開放します。EEPROM から見ると，Write の場合は全て受信となるため，8 ビットの受信が完了したら，9 クロック目に EEPROM から Acknowledge 信号の"0"を出力します。Read の場合は，スタート・コンディションの後の 8 ビット受信後に Acknowledge 信号の"0"を出力します。これに続いて, EEPROM は Read データを 8 ビット単位で出力しますが，出力後はバスを開放し，マスタ側から Acknowledge 信号の"0"が送られるのを待ちます。Acknowledge 信号の"0"を検出すると，EEPROM は次のアドレスの Read データを出力します。Acknowledge 信号の"0"が検出されずにストップ・コンディションを受信すると, Read 動作を終了しスタンバイ状態になります。なお, Acknowledge 信号の"0"が検出されず，かつストップ・コンディションも送られてこない場合は，データを出力せずにバス解放状態を持続します。図 7 に Acknowledge 信号のタイミング波形を示します。

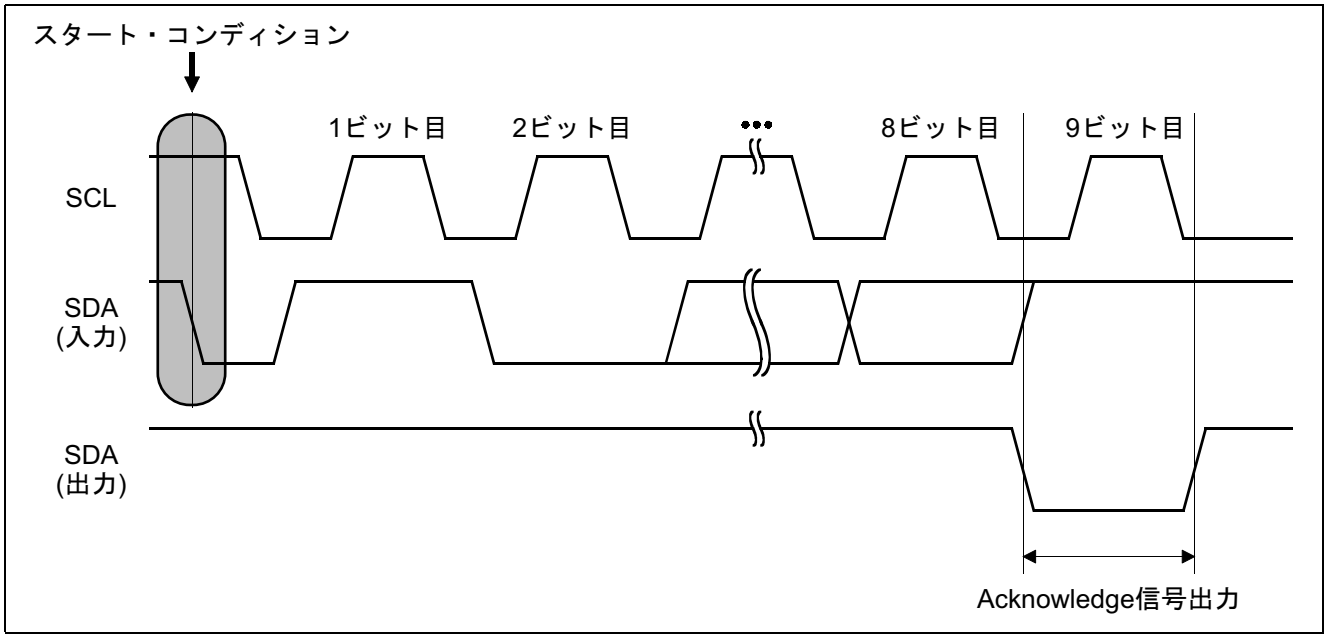


図 7　Acknowledge 信号出力タイミング波形

(e) スタート・コンディションに続いて 8 ビットのデバイス・アドレス・ワードを入力します。この入力でデバイスは Read，Write の動作を開始します。デバイス・アドレス・ワードはデバイス・コード 4 ビット，デバイス・アドレス・コード 3 ビット，Read/Write コード 1 ビットの 3 つのコードで構成されています。デバイス・アドレス・ワードの上位 4 ビットはデバイス・タイプを識別するデバイス・コードで，本タスク例で使用している EEPROM(HN58X2408)では"1010"の固定コードとなります。デバイス・コードに続けてデバイス・アドレス・コード 3 ビットを A2，A1，A0 の順に入力します。デバイス・アドレス・コードはバスに最大 8 ケ接続されたデバイスのうち，どれを選択するかを決定します。本タスク例で使用する EEPROM(HN58X2408)のデバイス・アドレス・コードは"000"に設定しています。デバイス・アドレス・ワードの 8 ビット目は R/W コードです。"0"入力の場合は Write 動作，"1"入力の場合は Read 動作になります。なお，デバイス・コードが"1010"でない場合，もしくはデバイス・アドレス・コードが一致しない場合は，Read/Write 動作に入らず，スタンバイモードになります。図 8 にデバイス・アドレス・ワードについて示します。

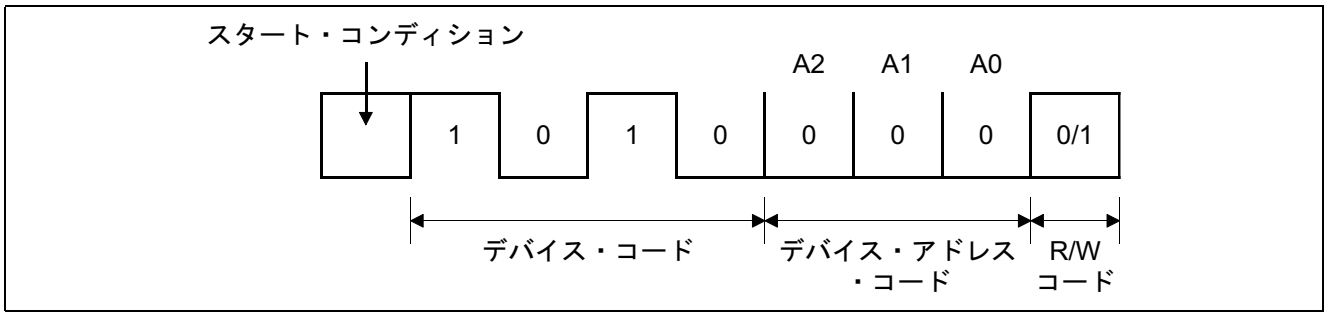


図 8　デバイス・アドレス・ワード

(f) 本タスク例では, 32 バイトまでの任意のバイト数を一度に書き換える Page Write 機能を使用して Write 動作を行ないます。スタート・コンディション→デバイス・アドレス・ワード→メモリ・アドレス(n)→Write データ(Dn)の順に，9 ビットごとの Acknowledge 信号の"0"出力を確認しながら入力します。Write データ(Dn+1)を入力すると，Page Write モード入ります。Write データ(Dn+1)を入力した時点で，ページ内アドレス(a0～a4)は自動的にインクリメントされ(n+1)番地になります。このように，Write データを次々と入力することができ, Write データ入力ごとにページ内アドレスがインクリメントされ, 最大 32 バイトの Write データを入力できます。ページ内アドレス(a0～a4)がページの最終番地に達した場合は，アドレスは"Roll Over"して，ページの先頭アドレスに戻ります。"Roll Over"した場合は，同一アドレスに Write データが 2 度(以上)入力されることになりますが，最後に入力した Write データが有効になります。ストップ・コンディションを入力すると，Write データの入力を終了し，書き換え動作に入ります。図 9 に Page Write 動作について示します。

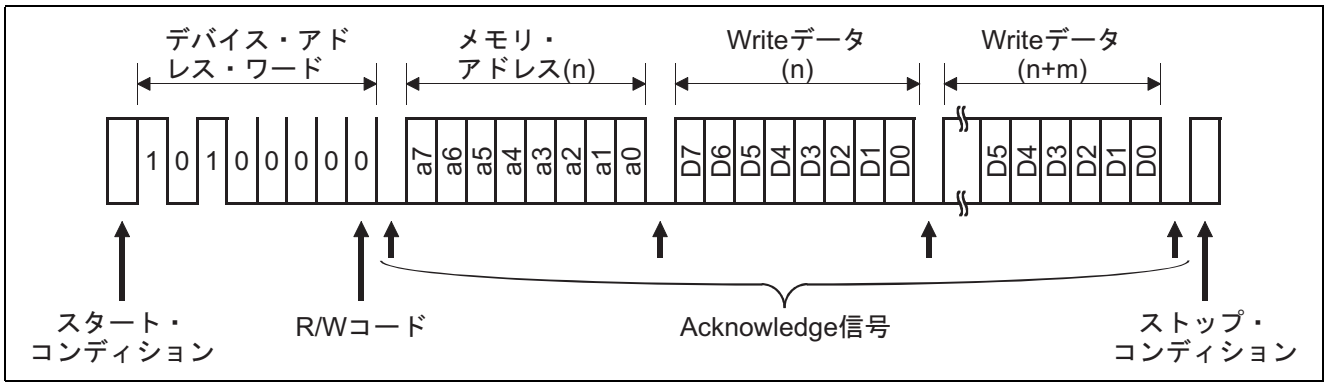


図 9　Page Write 動作

(g) EEPROM が書き換え中か否かを判定する機能として，Acknowledge Polling があります。書き換え期間中にスタート・コンディションに続いてデバイス・アドレス・ワード 8 ビットを入力します。Acknowledge Polling の場合，Read/Write コードは"1"/"0"のどちらでもかまいません。9 ビット目の Acknowledge 信号で書き換え中か否かを判定します。Acknowledge 信号が"1"のときは書き換え中，Acknowledge 信号が"0"のときは書き換え終了を示します。Acknowledge Polling は，Write データ入力後，ストップ・コンディションが入力された時点から機能します。

(h) 本タスク例では，データを連続して Read する Sequential Read モードを使用して Read 動作を行ないます。ダミーの Write モードで Read するデータの先頭アドレスを入力します。8 ビットのデータを出力した後，Acknowledge 信号の"0"を入力すると，アドレスがインクリメントされ，次の 8 ビットのデータが出力されます。データ出力後に Acknowledge 信号の"0"の入力を続けるとアドレスをインクリメントしながら次々とデータを出力します。アドレスが最終アドレスになった場合は，0 番地に"Roll Over"します。"Roll Over"後も Sequential Read が可能です。動作を終了するには，Acknowledge 信号の"1"(Acknowledge 信号の入力をせずに，バスを解放しても可)→ストップ・コンディションの順で入力します。図 10 に Sequential Read 動作について示します。

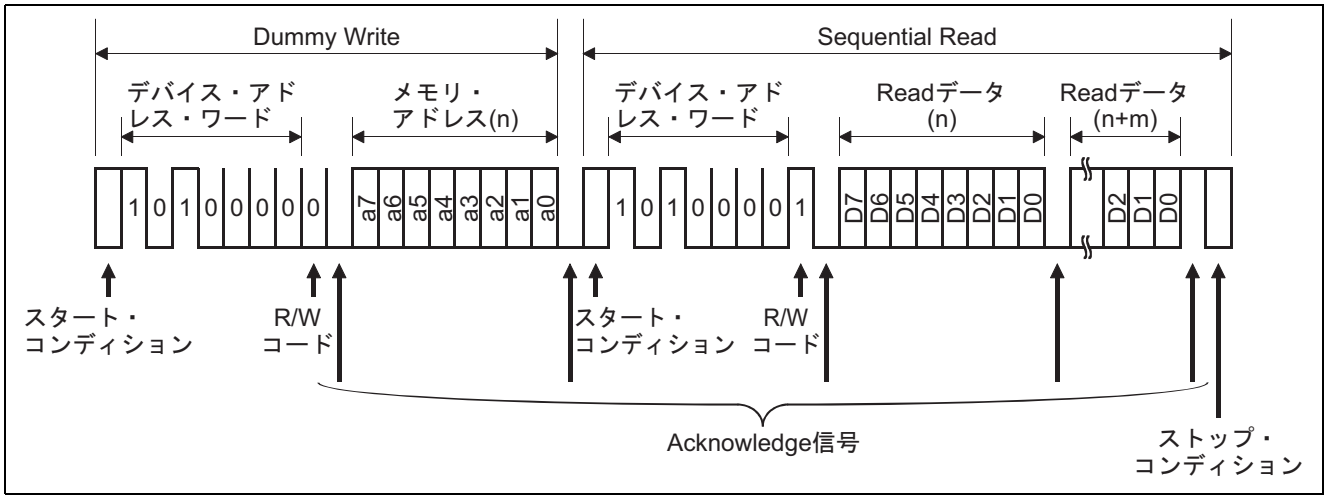


図 10　Sequential Read 動作

(i) EEPROM(HN58X2408)は，Write 動作が終了し，Read 動作に入るときには，書き換え時間(t_{WC})の期間，待機しなければなりません。書き換え時間(t_{WC})は，5V 動作時，10ms(max)です。書き換え時間のタイミング波形を図 11 に示します。

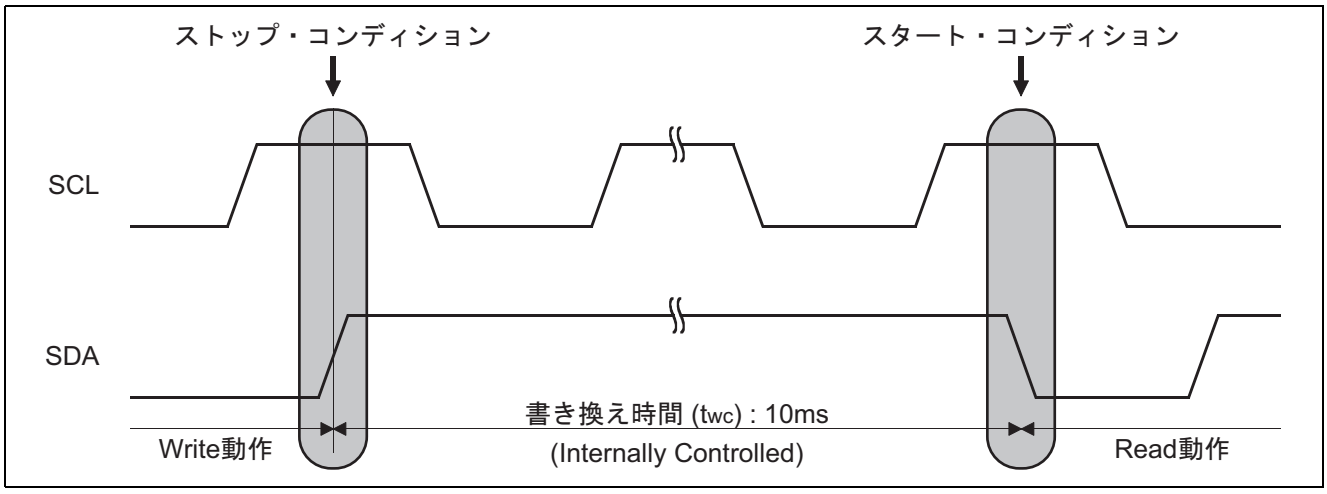


図 11　書き換え時間のタイミング波形

4. 動作原理

(1) 図 12 に Write(送信)時の動作原理を示します。図 12 に示すように H8/3644 のハードウェア処理，およびソフトウェア処理により EEPROM への Write(送信)を行ないます。

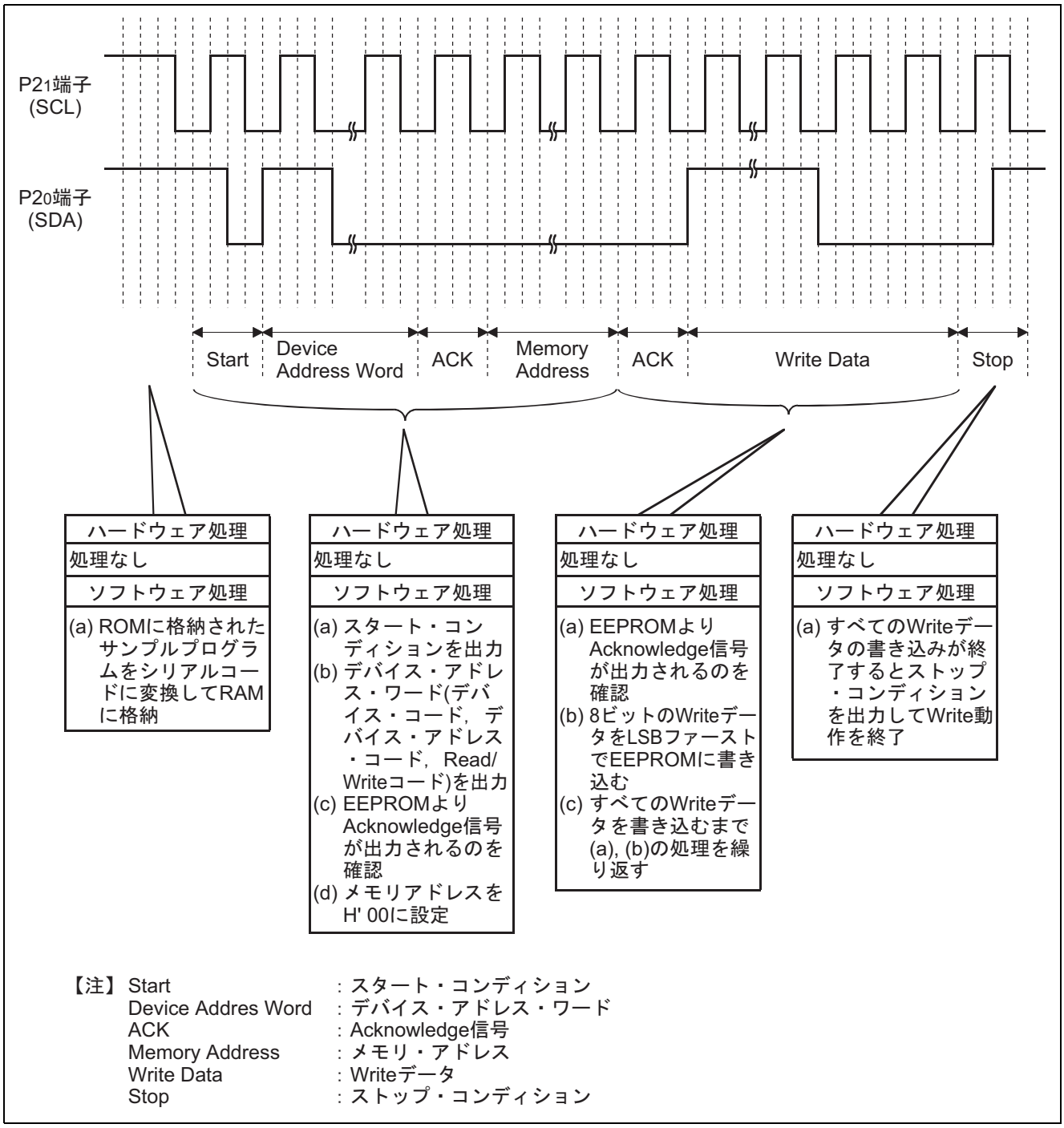
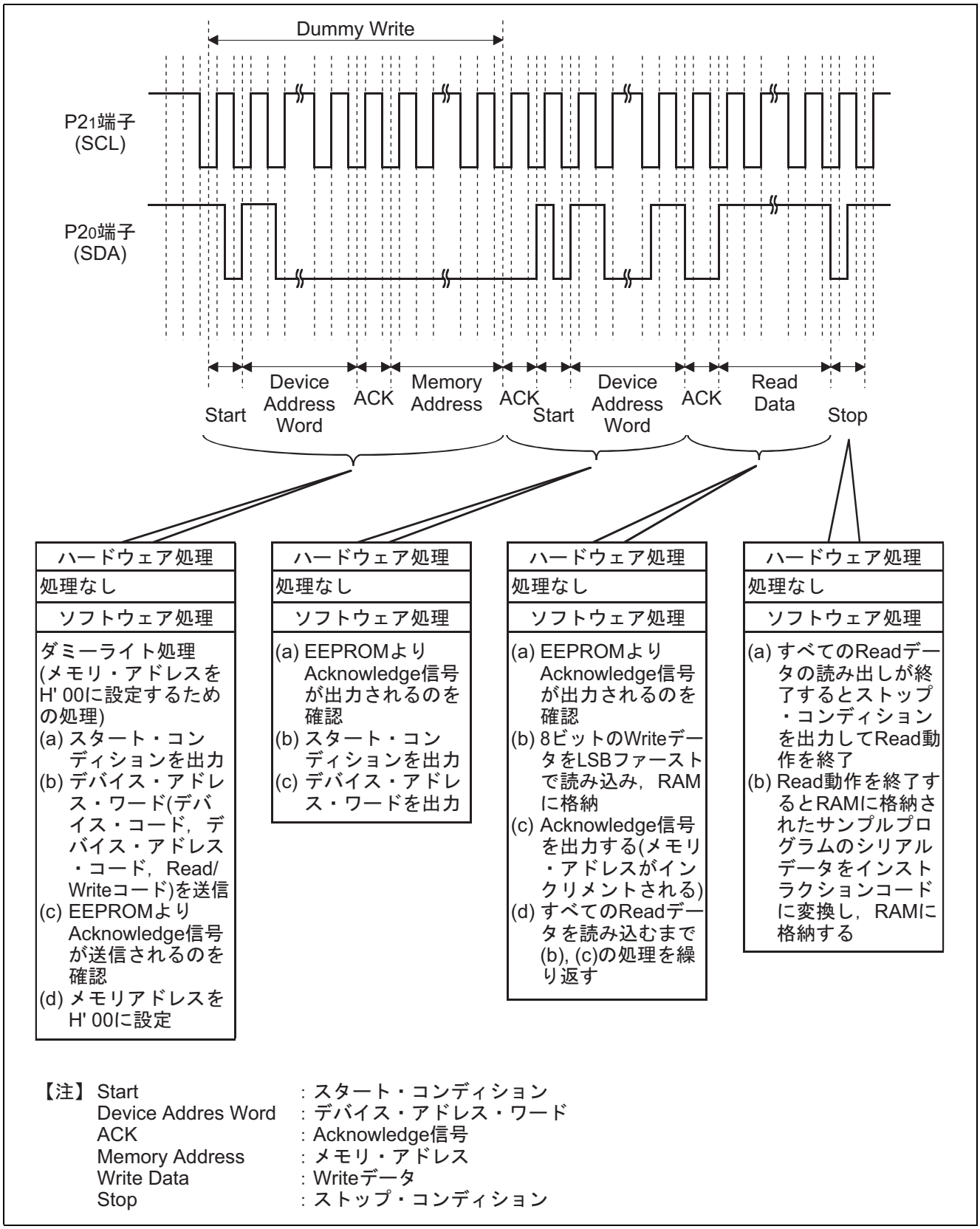


図 12　EEPROM への Write 時の動作原理

(2) 図 13 に Read(受信)時の動作原理を示します。図 13 に示すように H8/3644 のハードウェア処理，およびソフトウェア処理により EEPROM からの Read(受信)を行ないます。



【注】 Start : スタート・コンディション
Device Addres Word : デバイス・アドレス・ワード
ACK : Acknowledge信号
Memory Address : メモリ・アドレス
Write Data : Writeデータ
Stop : ストップ・コンディション

図 13　EEPROM からの Read 時の動作原理

(3) 本タスク例で使用しているポート 2 への出力，および入力の動作について表 3 に示します。表 3 に示すような設定により，シリアル・クロックの出力，およびシリアル・データの入出力を行ないます。

表 3　$P2_1$(SCL)および $P2_0$(SDA)入出力設定

端子設定 ＼ 出力値		$P2_1$(SCL)=1 $P2_0$(SDA)=1	$P2_1$(SCL)=1 $P2_0$(SDA)=0	$P2_1$(SCL)=0 $P2_0$(SDA)=1	$P2_1$(SCL)=0 $P2_0$(SDA)=0
PDR2	$P2_1$	0	0	0	0
	$P2_0$	0	0	0	0
PCR2	$PCR2_1$	0 (入力端子機能)	0 (入力端子機能)	1 (出力端子機能)	1 (出力端子機能)
	$PCR2_0$	0 (入力端子機能)	1 (出力端子機能)	0 (入力端子機能)	1 (出力端子機能)

(3) 図 14 に本タスク例で使用している H8/3644 のメモリマップを示します。

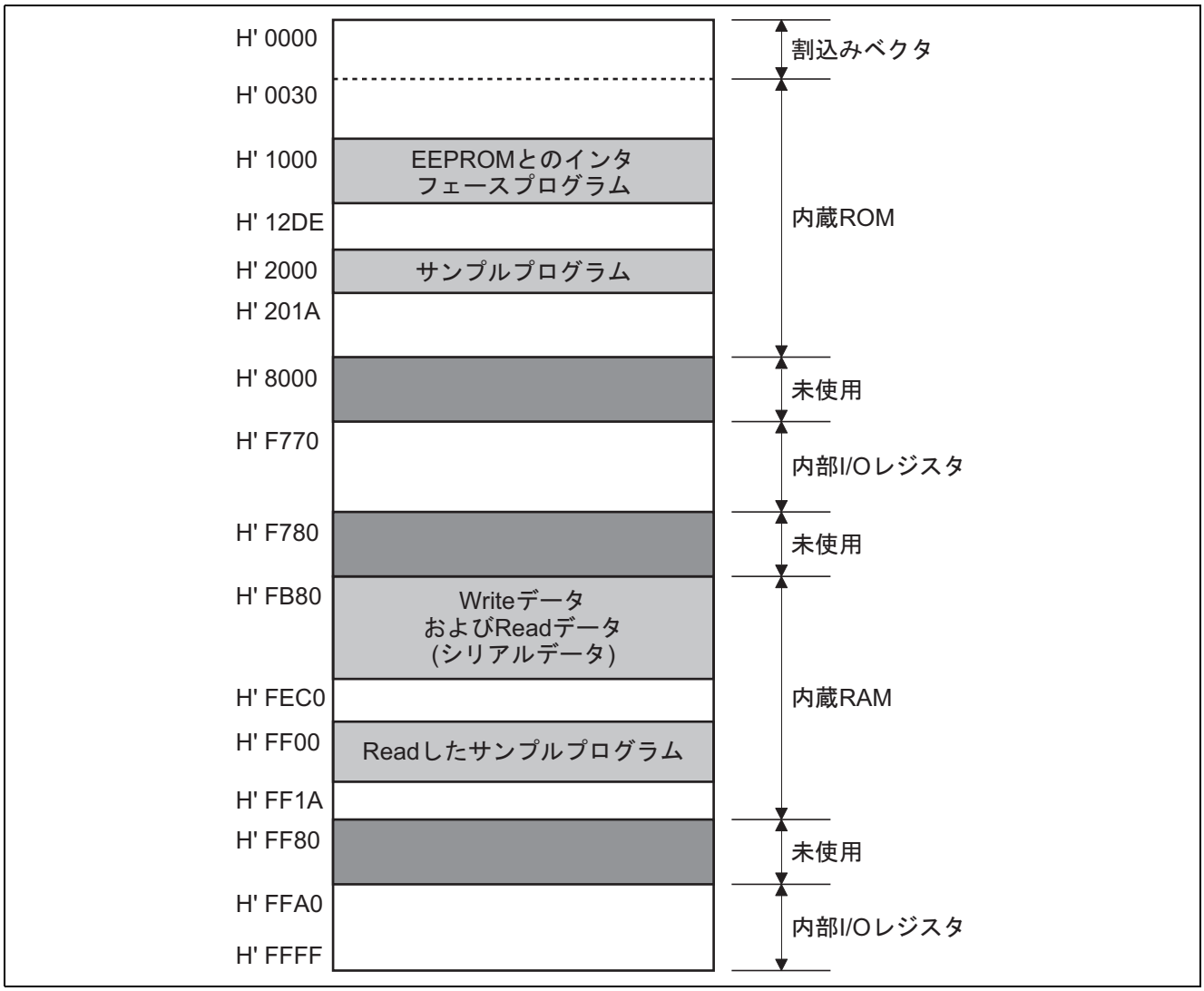


図 14　本タスク例で使用する H8/3644 のメモリマップ

5. ソフトウェア説明

(1) モジュール説明

表 4 に本タスク例におけるモジュール説明を示します。

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	スタックポインタのイニシャライズ，割込みの禁止，使用 RAM のイニシャライズ，書き換え時間の待機，EEPROM の書き込み，読み出しの制御を行なう
シリアルコード変換	SERCODE	R0M のサンプルプログラムのデータをシリアル・データに変換し，RAM に格納する
書き込み	WRITE	RAM に格納されているサンプルプログラムのシリアル・データを EEPROM に LSB ファースト方式により送信
読み出し	READ	EEPROM からサンプルプログラムのシリアル・データを LSB ファースト方式で受信し，RAM に格納
インストラクションコード変換	PARCODE	RAM に格納されたサンプルプログラムのシリアル・データをインストラクションコードに変換
受信サンプルプログラム	SPLPGM	EEPROM から受信したサンプルプログラムで，P73 端子に接続された LED を 262ms ごとに点灯，または消灯を繰り返す
エラールーチン	ERROR	エラー処理を行なう

(2) 引数の説明

本タスク例では，引数は使用していません。

(3) 使用内部レジスタ説明

表 5 に本タスク例における H8/3644 の使用内部レジスタ説明を示します。

表 5　H8/3644 の使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
PWCR	P21	ポートデータレジスタ 2(ポートデータレジスタ 21) ：$P2_1$ が"0"のとき，$P2_1$ 端子のデータは"0" ：$P2_1$ が"1"のとき，$P2_1$ 端子のデータは"1"	H'FFD5 ビット 1	0/1
	P20	ポートデータレジスタ 2(ポートデータレジスタ 20) ：$P2_0$ が"0"のとき，$P2_0$ 端子のデータは"0" ：$P2_0$ が"1"のとき，$P2_0$ 端子のデータは"1"	H'FFD5 ビット 0	0/1
PCR2	PCR21	ポートコントロールレジスタ 2(ポートコントロールレジスタ 21) ：$PCR2_1$ が"0"のとき，$P2_1$ 端子は入力端子として機能 ：$PCR2_1$ が"1"のとき，$P2_1$ 端子は出力端子として機能	H'FFE5 ビット 1	0/1
	PCR20	ポートコントロールレジスタ 2(ポートコントロールレジスタ 20) ：$PCR2_0$ が"0"のとき，$P2_0$ 端子は入力端子として機能 ：$PCR2_0$ が"1"のとき，$P2_0$ 端子は出力端子として機能	H'FFE5 ビット 0	0/1
PDR7	P73	ポートデータレジスタ 7(ポートデータレジスタ 73) ：$P7_3$ が"0"のとき，$P7_3$ 端子のデータは"0" ：$P7_3$ が"1"のとき，P73 端子のデータは"1"	H'FFDA ビット 3	0/1
PCR7	PCR73	ポートコントロールレジスタ 7(ポートコントロールレジスタ 73) ：$PCR7_3$ が"0"のとき，$P7_3$ 端子は入力端子として機能 ：$PCR7_3$ が"1"のとき，$P7_3$ 端子は出力端子として機能	H'FFEA ビット 3	1

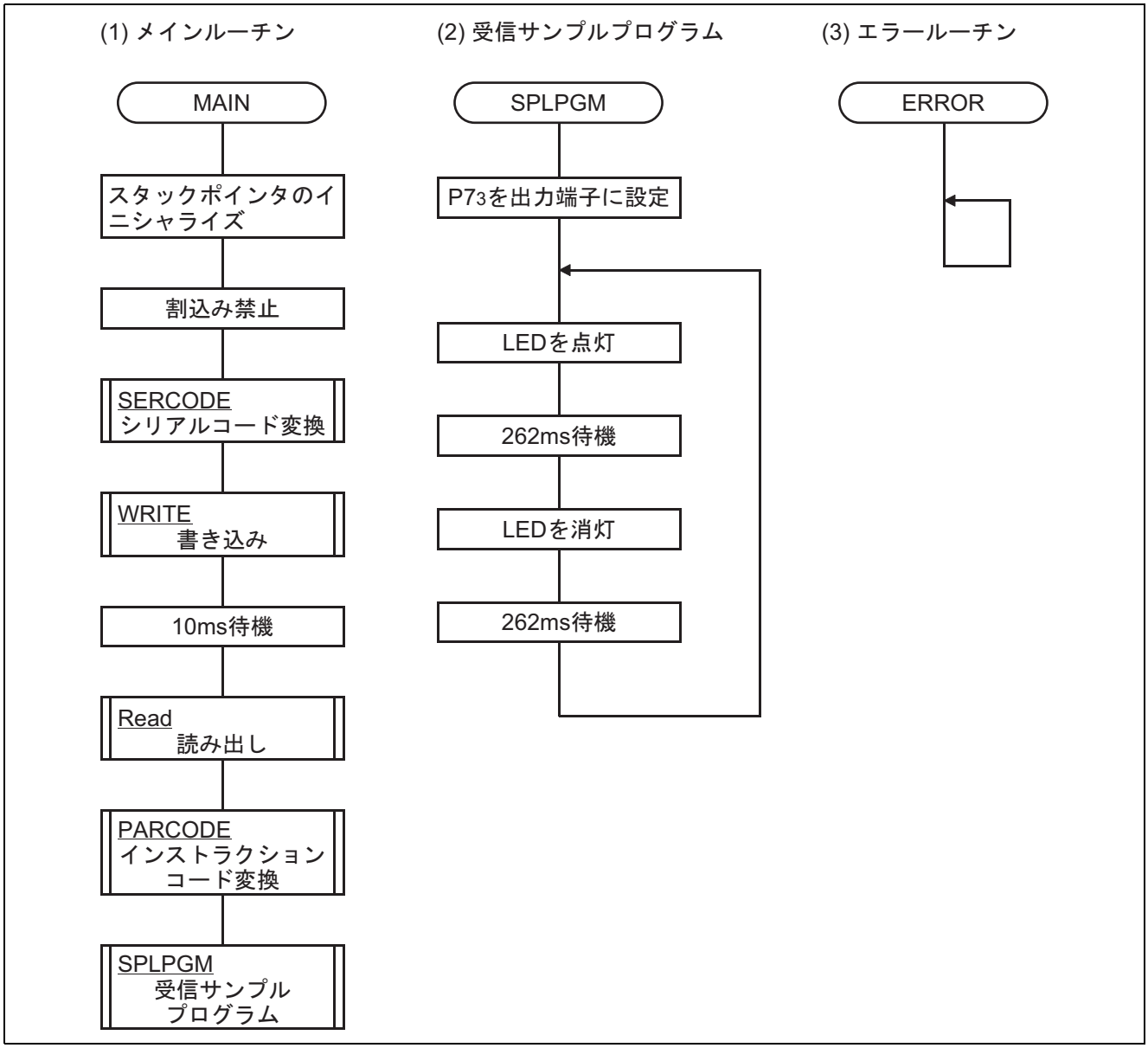
(4) 使用 RAM 説明

表 6 に本タスク例における使用 RAM 説明を示します。

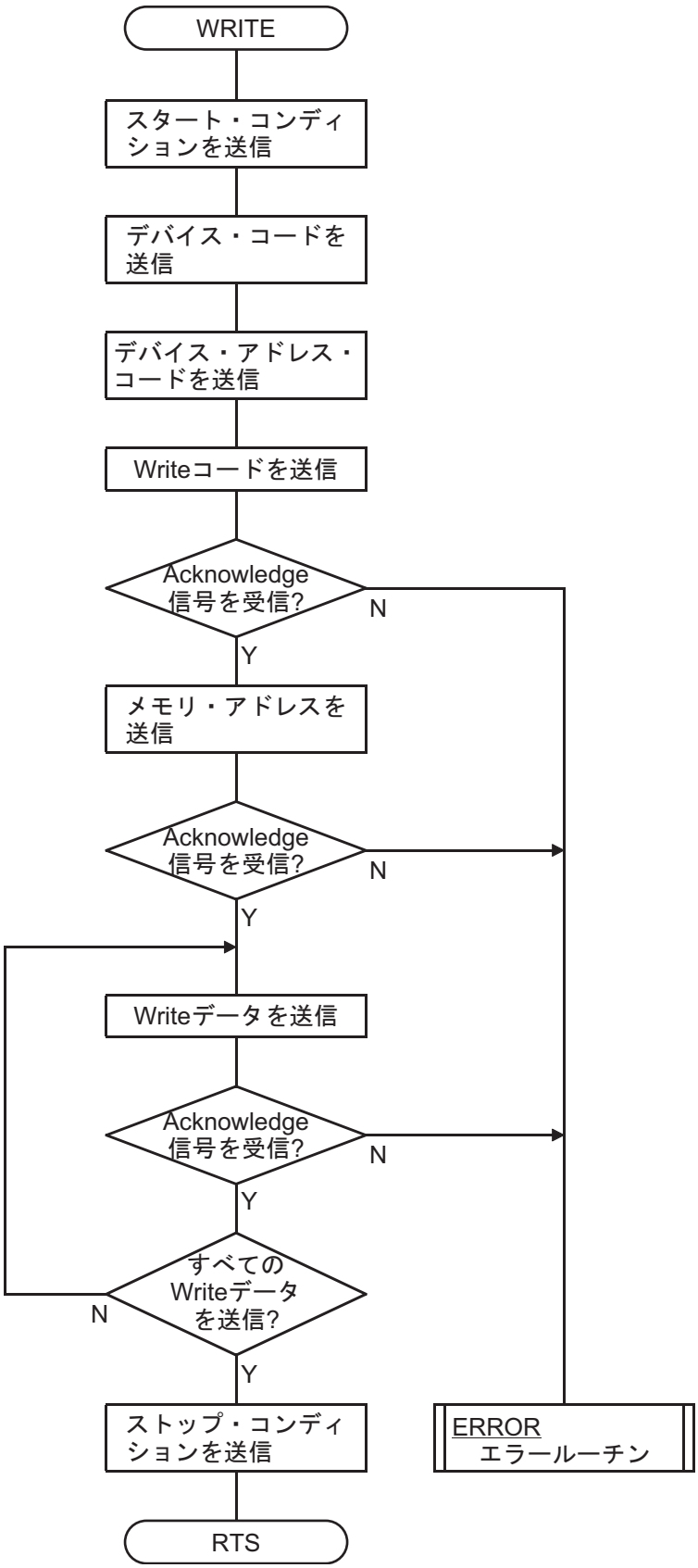
表 6　使用 RAM 説明

ラベル名	機能	アドレス	使用モジュール名
SERAREA	サンプルプログラムのシリアル・データを格納する RAM の先頭番地を格納	H'FB80	シリアルコード変換 書き込み 読み出し インストラクションコード変換
COUNTER	サンプルプログラムのシリアル・データを格納する RAM の最終番地を格納	H'FEC0	シリアルコード変換
SPLPGM	EEPROM から読み出したサンプルプログラムのインストラクションコードを格納する RAM の先頭番地を格納	H'FF00	シリアルコード変換 インストラクションコード変換

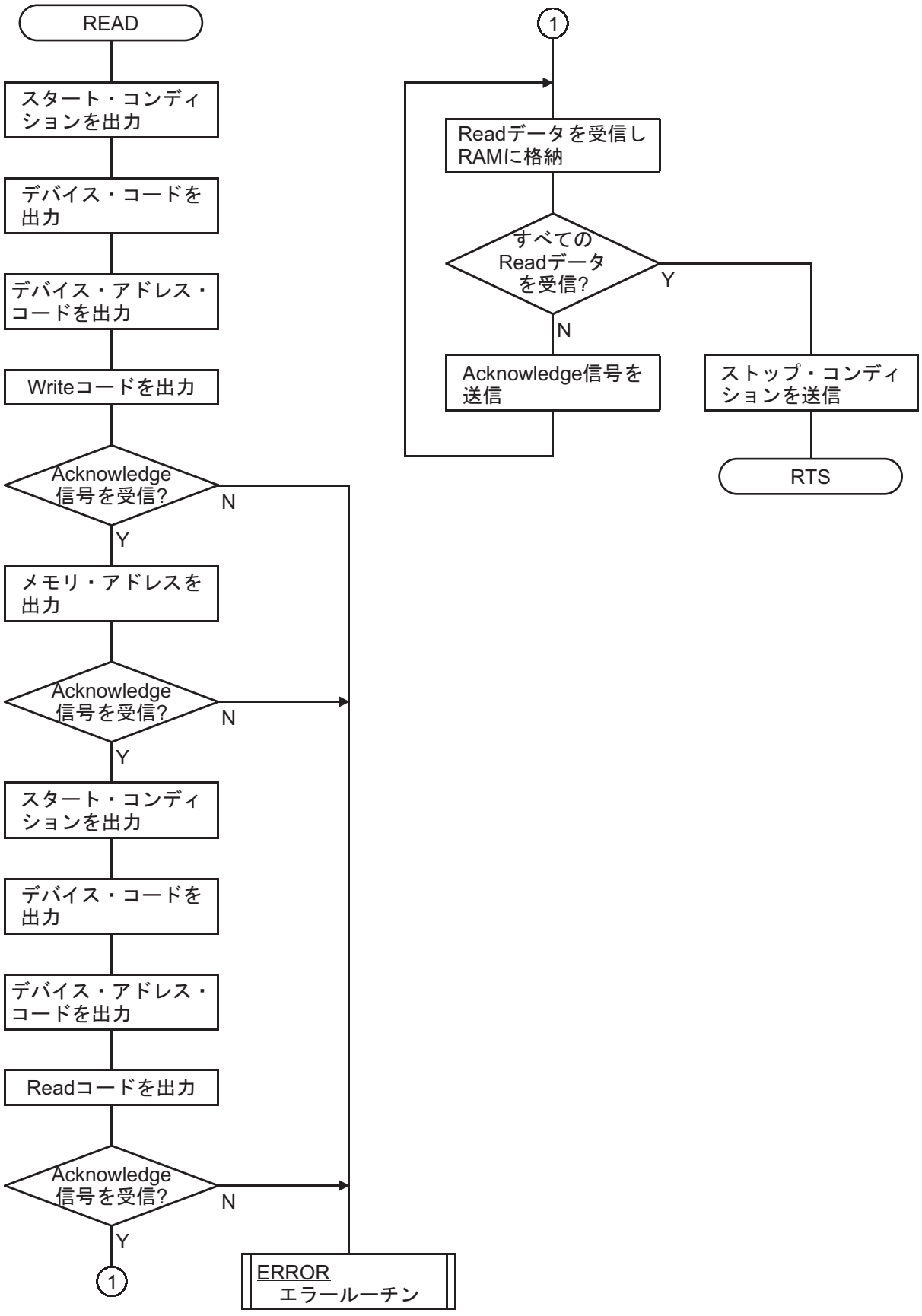
6. フローチャート



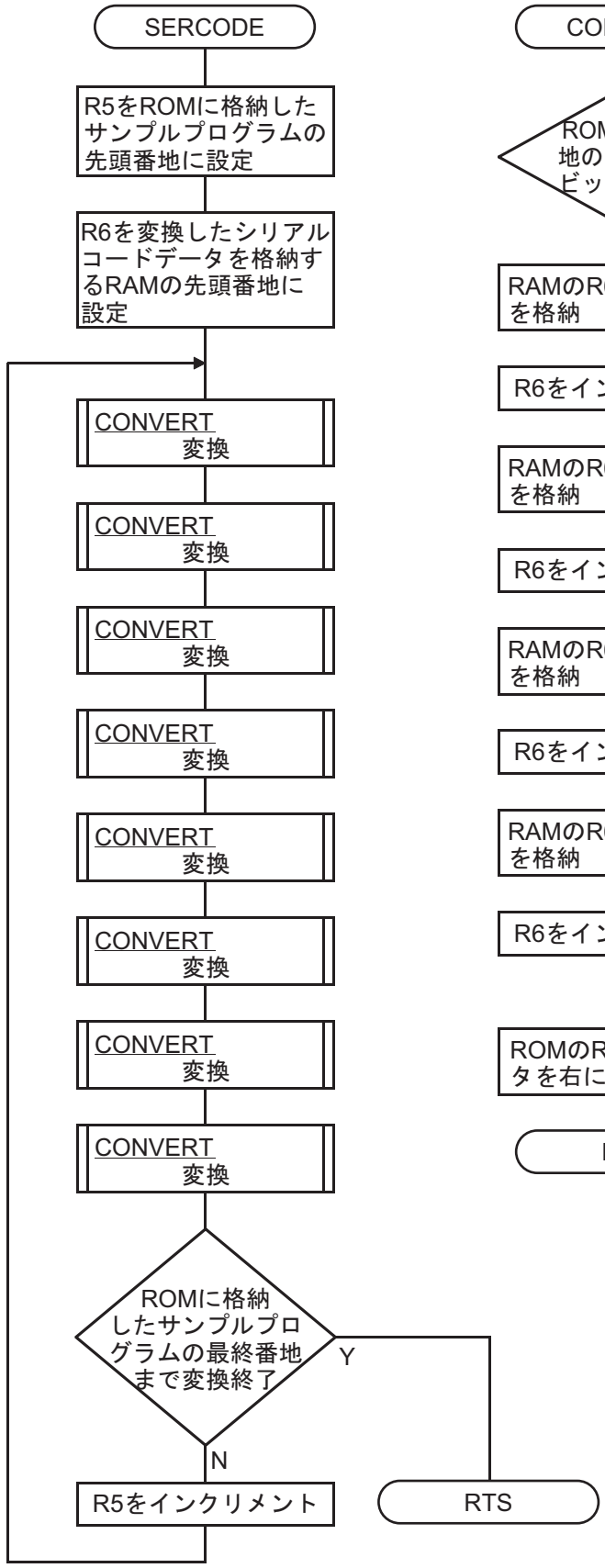
(4) 書き込み


WRITE
スタート・コンディションを送信
デバイス・コードを送信
デバイス・アドレス・コードを送信
Writeコードを送信
Acknowledge信号を受信?
N
Y
メモリ・アドレスを送信
Acknowledge信号を受信?
N
Y
Writeデータを送信
Acknowledge信号を受信?
N
Y
すべてのWriteデータを送信?
N
Y
ストップ・コンディションを送信
ERROR
エラールーチン
RTS

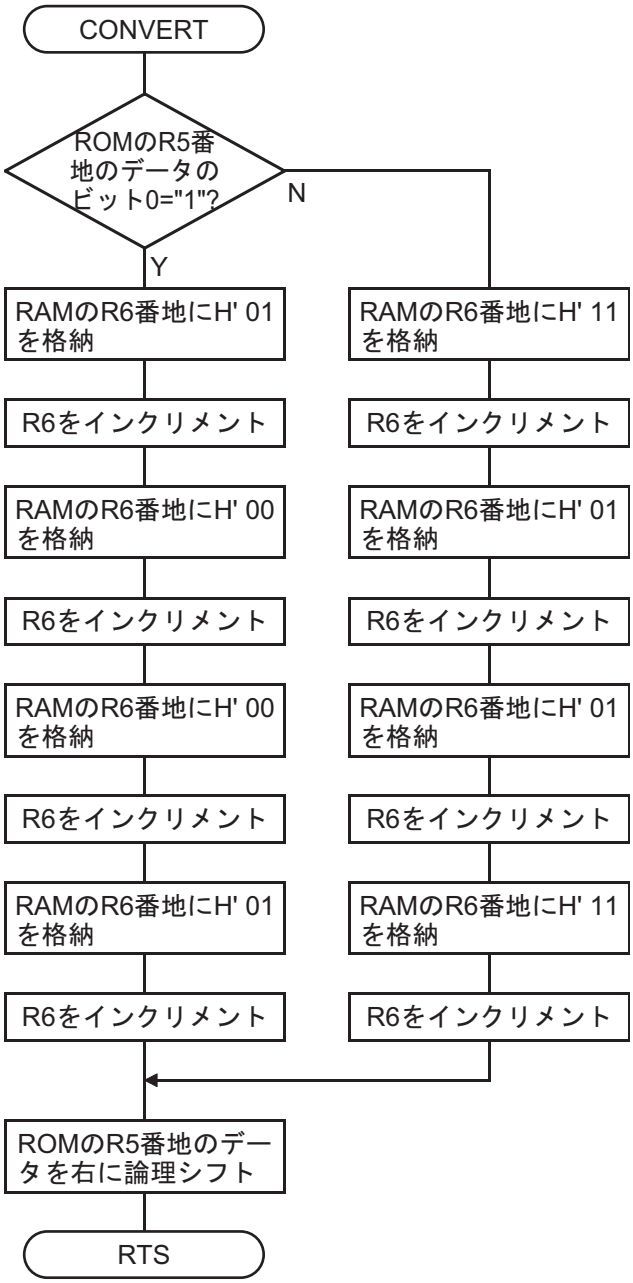
(5) 読み出し


READ
スタート・コンディションを出力
デバイス・コードを出力
デバイス・アドレス・コードを出力
Writeコードを出力
Acknowledge信号を受信?
N
Y
メモリ・アドレスを出力
Acknowledge信号を受信?
N
Y
スタート・コンディションを出力
デバイス・コードを出力
デバイス・アドレス・コードを出力
Readコードを出力
Acknowledge信号を受信?
N
Y
1
ERROR
エラールーチン
1
Readデータを受信しRAMに格納
すべてのReadデータを受信?
Y
N
Acknowledge信号を送信
ストップ・コンディションを送信
RTS

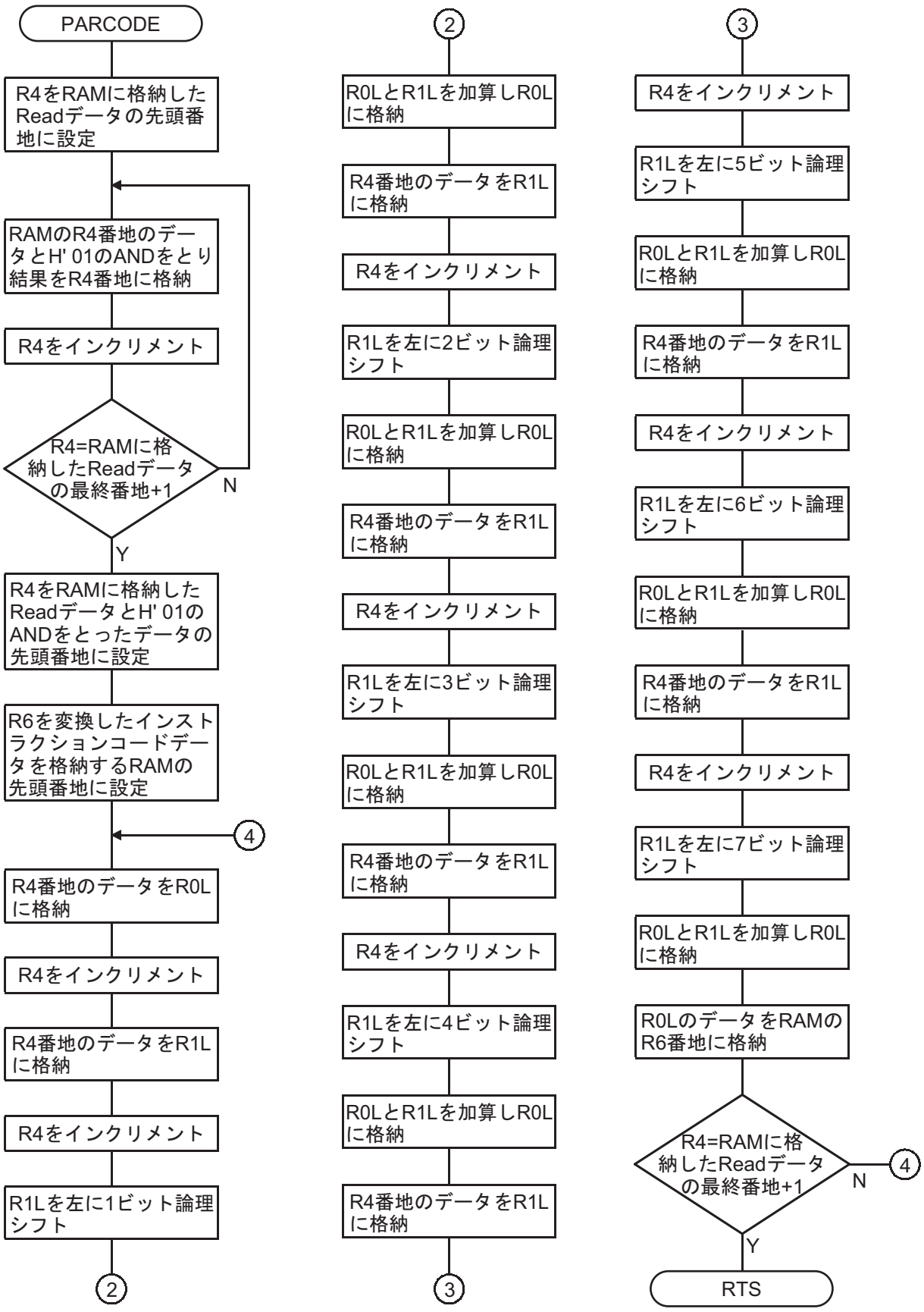
(6) シリアルコード変換



(7) 変換



(8) インストラクションコード変換



7. プログラムリスト

```
;*********************************************************
;
;       H8/3644 Application Note
;
;       'EEPROM Write & Read Control'
;
;       Function
;       : I/O Port Base
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;
;*********************************************************
;
;*********************************************************
        .cpu    300L
;*********************************************************
;
;*********************************************************
;       Symbol Defnition
;*********************************************************
;
PDR2    .equ    H'FFD5      ;Port Data Register 2
PCR2    .equ    H'FFE5      ;Port Control Register 2
;*********************************************************
;       Ram Allocation
;*********************************************************
;
SERAREA     .equ    H'FB80          ;Serial code area (H'FB80-H'FEBF: SERSIZE * 4 pulse)
COUNTER     .equ    H'FEC0          ;Counter (@H'FEC0 = H'1A: 26 bytes)
SPLPGM      .equ    H'FF00          ;Sample program area (H'FF00-H'FF19)
STACK       .equ        H'FF80              ;Stack Pointer
SPLSIZE     .equ    H'1A            ;Sample program size ( = 26 bytes)
SERSIZE     .equ    H'D0            ;Serial code size ( = SPLSIZE * 8-bit)
;*********************************************************
;       Vector Address
;*********************************************************
;
        .org    H'0000
        .data.w     MAIN            ;Reset Interrupt
        .org    H'0008
        .data.w     MAIN            ;IRQ0 Interrupt
        .data.w     MAIN            ;IRQ1 Interrupt
        .data.w     MAIN            ;IRQ2 Interrupt
        .data.w     MAIN            ;IRQ3 Interrupt
        .data.w     MAIN            ;INT0 - INT7 Interrupt
        .data.w     H'0014
        .data.w     MAIN            ;Timer A Interrupt
        .data.w     MAIN            ;Timer B1 Interrupt
        .data.w     H'0020
        .data.w     MAIN            ;Timer X Interrupt
        .data.w     MAIN            ;Timer V Interrupt
        .org    H'0026
        .data.w     MAIN            ;Sci1 Interrupt
        .org    H'002A
```

```
        .data.w     MAIN            ;Sci3 Interrupt
        .data.w     MAIN            ;A/D Converter Interrupt
        .data.w     MAIN            ;Sleep Interrupt
;*********************************************************
;       Main Program
;*********************************************************
;
        .org        H'1000
MAIN    MOV.W       #STACK,SP       ;Initialize Stack Pointer
        ORC     #H'80,CCR    ;Interrupt Disable
        BSR     SERCODE    ;Convert sample program data to serial code
        JSR         @WRITE          ;Write EEPROM
        MOV.W       #1,R4           ;10ms wait as tWC spec. of EEPROM
        MOV.W       #H'208C,R5
TWCWAIT     SUB.W       R4,R5
        BNE     TWCWAIT
        JSR     @READ       ;Read EEPROM
        JSR     @PARCODE      ;Convert Serial code to 8-bit sample program data
        MOV.W       #1,R0           ;Load software time data of LED on/off period
        MOV.W       #0,R1           ;as 262ms
        MOV.W       #1,R2
        JMP     @SPLPGM     ;Execute sample program at Internal RAM
;
ERROR       BRA     ERROR           ;ERROR area
;*********************************************************
;       Convert sample program to serial code
;*********************************************************
;
SERCODE     .equ        $
        MOV.B       #0,R0L          ;Load SCL = 1, SDA = 1 data
        MOV.B       #1,R1L          ;Load SCL = 1, SDA = 0 data
        MOV.B       #2,R2L          ;Load SCL = 0, SDA = 1 data
        MOV.B       #3,R3L          ;Load SCL = 0, SDA = 0 data
        MOV.W       #SAMPLE,R5      ;Load sample program address
        MOV.W       #SERAREA,R6     ;Load serial code address
        MOV.B       #SPLSIZE,R4L    ;Load sample program size
        MOV.B       R4L,@COUNTER
NEXTCON     MOV.B       @R5+,R4L    ;Load sample program data
        BSR     CONVERT     ;Convert sample program data to serial code bit0
        BSR     CONVERT     ;Convert sample program data to serial code bit1
        BSR     CONVERT     ;Convert sample program data to serial code bit2
        BSR     CONVERT     ;Convert sample program data to serial code bit3
        BSR     CONVERT     ;Convert sample program data to serial code bit4
        BSR     CONVERT     ;Convert sample program data to serial code bit5
        BSR     CONVERT     ;Convert sample program data to serial code bit6
        BSR     CONVERT     ;Convert sample program data to serial code bit7
        MOV.B       @COUNTER,R4L
        DEC     R4L
        MOV.B       R4L,@COUNTER
        CMP.B       #0,R4L          ;@COUNTER=0?
        BNE     NEXTCON       ;No.
        RTS
;
CONVERT     .equ    $
        BTST        #0,R4L          ;If sample program data bitn is "0",
        BEQ     BITEQU0     ;it branch to BITEQU0
        MOV.B       R2L,@R6         ;Store SCL & SDA data as follows.
```

```
        ADDS    #1,R6       ;SCL = 0110
        MOV.B       R0L,@R6          ;SDA = 1111
        ADDS    #1,R6
        MOV.B       R0L,@R6
        ADDS    #1,R6
        MOV.B       R2L,@R6
        ADDS    #1,R6
        BRA     BITn
BITEQU0     MOV.B       R3L,@R6      ;Store SCL & SDA data as follows.
        ADDS    #1,R6       ;SCL = 0110
        MOV.B       R1L,@R6          ;SDA = 0000
        ADDS    #1,R6
        MOV.B       R1L,@R6
        ADDS    #1,R6
        MOV.B       R3L,@R6
        ADDS    #1,R6
BITn        SHLR        R4L
        RTS
;*********************************************************
;       Write (Use Page Write Operation)
;*********************************************************
;
WRITE       .equ    $
        MOV.B       R0L,@PDR2        ;SCL = "0", SDA = "1"
        MOV.B       R2L,@PCR2
;
        JSR     @RW_start   ;Output "0" of Device address start bit
        JSR     @RW_H       ;Output "1" of Device address
        JSR     @RW_L       ;Output "0" of Device address
        JSR     @RW_H       ;Output "1" of Device address
        JSR     @RW_L       ;Output "0" of Device address
        JSR     @RW_L       ;Output "0" of Device address code
        JSR     @RW_L       ;Output "0" of Device address code
        JSR     @RW_L       ;Output "0" of Device address code
        JSR     @RW_L       ;Output "0" of Device address write bit
        JSR     @RW_ack     ;Input "0" of /ACK
;
        JSR     @RW_L       ;Output "0" of Memory address a7
        JSR     @RW_L       ;Output "0" of Memory address a6
        JSR     @RW_L       ;Output "0" of Memory address a5
        JSR     @RW_L       ;Output "0" of Memory address a4
        JSR     @RW_L       ;Output "0" of Memory address a3
        JSR     @RW_L       ;Output "0" of Memory address a2
        JSR     @RW_L       ;Output "0" of Memory address a1
        JSR     @RW_L       ;Output "0" of Memory address a0
        JSR     @RW_ack     ;Input "0" of /ACK
;
        MOV.W       #SERAREA,R4      ;Load serial code of sample program
        MOV.W       #SPLSIZE,R5
WRLOOP      JSR     @WR_data         ;Output Write data D0
        JSR     @WR_data     ;Output Write data D1
        JSR     @WR_data     ;Output Write data D2
        JSR     @WR_data     ;Output Write data D3
        JSR     @WR_data     ;Output Write data D4
        JSR     @WR_data     ;Output Write data D5
        JSR     @WR_data     ;Output Write data D6
        JSR     @WR_data     ;Output Write data D7
```

```
        JSR     @RW_ack       ;Input "0" of /ACK
        DEC     R5L         ;Counter=0?
        BEQ     WREND       ;No.
        BRA     WRLOOP
WREND       JSR     @RW_stop          ;Output stop bit
        RTS
;*********************************************************
;       Read (Random & Sequential Operation of EEPROM)
;*********************************************************
;
READ        .equ    $
        MOV.W       #SERAREA,R4       ;Load serial code address
        MOV.W       #SPLSIZE,R5       ;Load count data ( = 26 bytes)
        MOV.B       #0,R0L            ;Load SCL = 1, SDA = 1 data
        MOV.B       #1,R1L            ;Load SCL = 1, SDA = 0 data
        MOV.B       #2,R2L            ;Load SCL = 0, SDA = 1 data
        MOV.B       #3,R3L            ;Load SCL = 0, SDA = 0 data
        MOV.B       R0L,@PDR2
        MOV.B       R2L,@PCR2         ;SCL = "0", SDA = "1"
;
        JSR     @RW_start     ;Output "0" of Device address start bit
        JSR     @RW_H       ;Output "1" of Device address
        JSR     @RW_L       ;Output "0" of Device address
        JSR     @RW_H       ;Output "1" of Device address
        JSR     @RW_L       ;Output "0" of Device address
        JSR     @RW_L       ;Output "0" of Device address code
        JSR     @RW_L       ;Output "0" of Device address code
        JSR     @RW_L       ;Output "0" of Device address code
        JSR     @RW_L       ;Output "0" of Device address write bit
        JSR     @RW_ack     ;Input "0" of /ACK
;
        BSR     RW_L        ;Output "0" of Memory address a7
        BSR     RW_L        ;Output "0" of Memory address a6
        BSR     RW_L        ;Output "0" of Memory address a5
        BSR     RW_L        ;Output "0" of Memory address a4
        BSR     RW_L        ;Output "0" of Memory address a3
        BSR     RW_L        ;Output "0" of Memory address a2
        BSR     RW_L        ;Output "0" of Memory address a1
        BSR     RW_L        ;Output "0" of Memory address a0
        BSR     RW_ack      ;Input "0" of /ACK
;
        BSR     RW_start      ;Output "0" of Device address start bit
        BSR     RW_H        ;Output "1" of Device address
        BSR     RW_L        ;Output "0" of Device address
        BSR     RW_H        ;Output "1" of Device address
        BSR     RW_L        ;Output "0" of Device address
        BSR     RW_L        ;Output "0" of Device address code
        BSR     RW_L        ;Output "0" of Device address code
        BSR     RW_L        ;Output "0" of Device address code
        BSR     RW_H        ;Output "1" of Device address read bit
        BSR     RW_ack      ;Input "0" of /ACK
;
RDLOOP      JSR     @RD_data          ;Input Read data D0
        JSR     @RD_data      ;Input Read data D1
        JSR     @RD_data      ;Input Read data D2
        JSR     @RD_data      ;Input Read data D3
        JSR     @RD_data      ;Input Read data D4
```

```
        JSR     @RD_data      ;Input Read data D5
        JSR     @RD_data      ;Input Read data D6
        JSR     @RD_data      ;Input Read data D7
        BSR     RW_L        ;Output "0" of /ACK bit
        DEC     R5L
        BEQ     SERDEND
        BRA     RDLOOP
SERDEND     .equ        $
        MOV.B       R0L,@PDR2
        MOV.B       R3L,@PCR2         ;SCL = "0", SDA = "0"
        BSR     RW_stop     ;Output stop bit
        RTS
;*********************************************************
;       Subroutine
;*********************************************************
;
RW_start    .equ        $
        MOV.B       R0L,@PDR2         ;Output "0" of Start bit
        MOV.B       R2L,@PCR2         ;SCL = "0", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2         ;SCL = "1", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R1L,@PCR2         ;SCL = "1", SDA = "0"
        MOV.B       R0L,@PDR2
        MOV.B       R3L,@PCR2         ;SCL = "0", SDA = "0"
        RTS
;
RW_H        .equ        $
        MOV.B       R0L,@PDR2         ;Output "1" of Device/Memory address
        MOV.B       R2L,@PCR2         ;SCL = "0", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2         ;SCL = "1", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2         ;SCL = "1", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R2L,@PCR2         ;SCL = "0", SDA = "1"
        RTS
;
RW_L        .equ        $
        MOV.B       R0L,@PDR2         ;Output "0" of Device/Memory address or /ACK
        MOV.B       R3L,@PCR2         ;SCL = "0", SDA = "0"
        MOV.B       R0L,@PDR2
        MOV.B       R1L,@PCR2         ;SCL = "1", SDA = "0"
        MOV.B       R0L,@PDR2
        MOV.B       R1L,@PCR2         ;SCL = "1", SDA = "0"
        MOV.B       R0L,@PDR2
        MOV.B       R3L,@PCR2         ;SCL = "0", SDA = "0"
        RTS
;
RW_ack      .equ    $
        MOV.B       R0L,@PDR2         ;Input "0" of /ACK bit
        MOV.B       R2L,@PCR2
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2
        BTST    #0,@PDR2      ;/ACK=0?
        BEQ     ACKOK       ;Yes.
        JMP     @ERROR
```

```
ACKOK       MOV.B       R0L,@PDR2
        MOV.B       R3L,@PCR2       ;SCL = "0", SDA = "0"
        RTS
;
RW_stop     .equ        $
        MOV.B       R0L,@PDR2       ;Output "1" of Stop bit
        MOV.B       R1L,@PCR2       ;SCL = "1", SDA = "0"
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2       ;SCL = "1", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R2L,@PCR2       ;SCL = "0", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R2L,@PCR2       ;SCL = "0", SDA = "1"
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2       ;SCL = "1", SDA = "1"
        RTS
;
WR_data     .equ        $
        MOV.B       @R4+,R1L        ;Output write data bitn
        MOV.B       R1L,@PCR2       ;SCL = "0", SDA = "Dn Output"
        MOV.B       @R4+,R1L
        MOV.B       R1L,@PCR2       ;SCL = "1", SDA = "Dn Output"
        MOV.B       @R4+,R1L
        MOV.B       R1L,@PCR2       ;SCL = "1", SDA = "Dn Output"
        MOV.B       @R4+,R1L
        MOV.B       R1L,@PCR2       ;SCL = "0", SDA = "Dn Output"
        RTS
;
RD_data     .equ        $
        MOV.B       R0L,@PDR2       ;Input read data bitn
        MOV.B       R2L,@PCR2       ;SCL = "0", SDA = "Dn input"
        MOV.B       R0L,@PDR2
        MOV.B       R0L,@PCR2       ;SCL = "1", SDA = "Dn input"
        MOV.B       @PDR2,R6L
        MOV.B       R6L,@R4         ;Store serial code at Internal RAM
        ADDS    #1,R4
        MOV.B       R0L,@PDR2
        MOV.B       R2L,@PCR2       ;SCL = "0", SDA = "Dn input"
        RTS
;
PARCODE     .equ        $
        MOV.W       #SERAREA,R4     ;Load serial code address
        MOV.B       #SERSIZE,R5L
ANDDATA     MOV.B       @R4,R0L     ;AND.B #1,@SERAREA(H'FB80-FEBF)
        AND     #H'01,R0L
        MOV.B       R0L,@R4
        ADDS    #1,R4
        DEC     R5L
        BNE     ANDDATA
        MOV.W       #SERAREA,R4     ;Load serial code address
        MOV.W       #SPLSIZE,R5
        MOV.W       #SPLPGM,R6      ;Load execution sample program address
SERPAR      MOV.B       @R4+,R0L    ;Load serial code bit0
        MOV.B       @R4+,R1L        ;Load serial code bit1
        SHLL    R1L         ;Convert bit1 code
        ADD.B       R1L,R0L         ;Calculate Bit1,0 code
        MOV.B       @R4+,R1L        ;Load serial code bit2
```

```
        SHLL    R1L         ;Convert bit2 code
        SHLL    R1L
        ADD.B       R1L,R0L         ;Calculate Bit2-0 code
        MOV.B       @R4+,R1L        ;Load serial code bit3
        SHLL    R1L         ;Convert bit3 code
        SHLL    R1L
        SHLL    R1L
        ADD.B       R1L,R0L         ;Calculate Bit3-0 code
        MOV.B       @R4+,R1L        ;Load serial code bit4
        SHLL    R1L         ;Convert bit4 code
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        ADD.B       R1L,R0L         ;Calculate Bit4-0 code
        MOV.B       @R4+,R1L        ;Load serial code bit5
        SHLL    R1L         ;Convert bit5 code
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        ADD.B       R1L,R0L         ;Calculate Bit5-0 code
        MOV.B       @R4+,R1L        ;Load serial code bit6
        SHLL    R1L         ;Convert bit6 code
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        ADD.B       R1L,R0L         ;Calculate Bit6-0 code
        MOV.B       @R4+,R1L        ;Load serial code bit7
        SHLL    R1L         ;Convert bit7 code
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        SHLL    R1L
        ADD.B       R1L,R0L         ;Calculate Bit7-0 code
        MOV.B       R0L,@R6
        ADDS    #1,R6
        DEC     R5L
        BNE     SERPAR
        RTS
;*********************************************************
;       H8/3644 Sample program of LED Control
;*********************************************************
;
        .org        H'2000
SAMPLE      BSET        #3,@H'FFEA    ;Initialize P73 Output Port
LEDCTL      BSET        #3,@H'FFDA    ;Turn on LED
        BSR     WAIT
        BCLR        #3,@H'FFDA        ;Turn off LED
        BSR     WAIT
        BRA     LEDCTL
WAIT    SUB.W       R0,R1
        MULXU       R0L,R2
        BNE     WAIT
```

```
        RTS
;
        .end
```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.19	—	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は負いません。
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、その他お気付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。