

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

H8/3069F

H8/3069F ユーザプログラムモード応用例

目次

1.	概要	2
1.1	概要	2
1.2	ユーザプログラムモード	4
2.	ユーザプログラムモード応用例	10
2.1	概要	10
2.2	機能	10
2.3	仕様	11
2.3.1	通信仕様	11
2.3.2	LED表示仕様	11
2.3.3	フラッシュ消去／書き込み仕様	13
2.4	ハードウェア	15
2.5	ソフトウェア	17
2.5.1	メイン処理	18
2.5.2	アプリケーション処理	19
2.5.3	フラッシュ消去／書き込み処理	22
2.6	コントロールソフト	27
2.7	評価ボードによる動作例	27
2.7.1	準備	28
2.7.2	動作確認	30
2.7.3	F/W書き換えの実行	32
2.7.4	F/W書き換え後の動作確認	33
3.	付録	34
3.1	H8/3069F評価ボード回路図	34
3.2	全ソースリスト	35

1. 概要

1.1 概要

H8/3069Fはオンボード（機器組み込み状態）でプログラムの書き込み／消去が可能なフラッシュメモリを内蔵しています。オンボードでプログラムの書き換えができるため、市場出荷後のソフトウェアバージョンアップ、メンテナンスなどを容易に行うことができます。

H8/3069Fにはオンボードによる書き込み可能なモードとして、3種類のオンボードプログラミングモード（ブートモード、ユーザブートモード、ユーザプログラムモード）があり、状況に応じたモードの選択が可能です。表1.1にオンボードプログラミングモードの比較表を示します。

(1) ブートモード

マイコンに内蔵されたブートプログラムにより、SCIによるホストとの制御コマンド、書き込みデータ通信を行いながら、消去／書き込みするモードです。ホスト上に制御コマンドを送信するためのツールを用意する必要がありますが、マイコン側のプログラムを用意する必要はなく、量産ラインにおけるマイコンの初期書き込み時に有効なモードです。

本モードを起動するためには、外部からのMD2、NMI、FWEなどの端子設定とリセット動作が必要です。（図1.1）

(2) ユーザブートモード

SCIを使用するブートモードとは異なるユーザ任意のI/Fを用いたブートモードを実現するモードです。あらかじめユーザブートマットと呼ばれるメモリマット上に手続きプログラムを書き込んでおく必要があります。

本モードを起動するためには、外部からのMD2、NMI、FWEなどの端子設定とリセット動作が必要です。（図1.2）

(3) ユーザプログラムモード

任意のI/Fを用いた、フラッシュメモリの消去／書き込みを行うモードです。通常のアプリケーション動作からフラッシュメモリの消去／書き込みが可能です。ユーザマットと呼ばれるメモリマット上に手続きプログラムを書き込んでおく必要があります。（図1.3）

本アプリケーションノートでは、このうちユーザプログラムモードにおけるハードウェア、ソフトウェアの応用例について説明します。

ブートモードおよびユーザブートモードの詳細については、「H8/3069F-ZTATハードウェアマニュアル」を参照してください。

表 1.1 オンボードプログラミングモードの比較

項目	モード		
	ブートモード	ユーザブートモード	ユーザプログラムモード
書き込み／消去可能マット	ユーザマット ユーザブートマット	ユーザマット	ユーザマット
消去単位	全ブロック自動消去	ブロック単位消去	ブロック単位消去
データ転送方法	SCI	任意の I/F	任意の I/F
モード遷移方法	MD2、NMI、FWE 設定 リセット	MD2、NMI、FWE 設定 リセット	FWE 設定

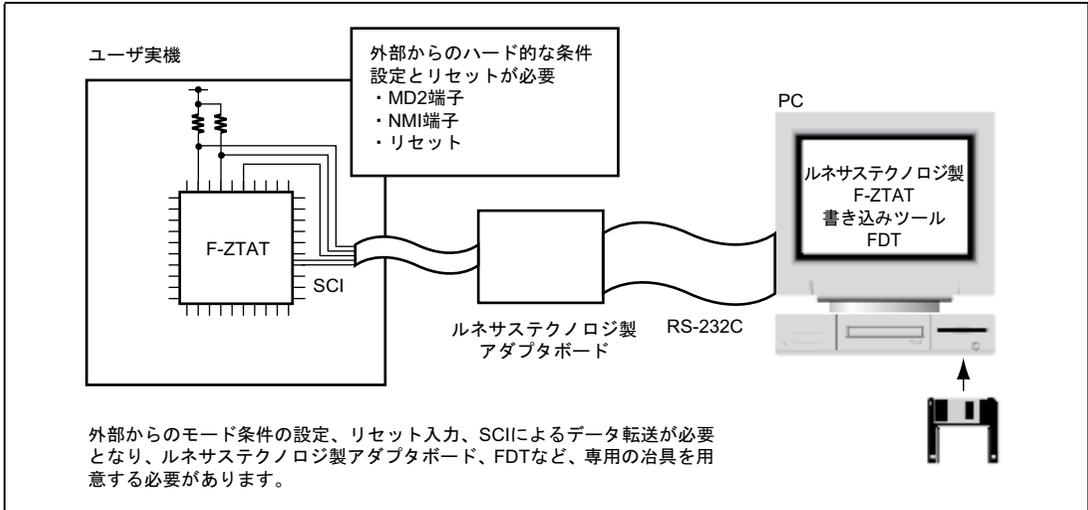


図 1.1 ブートモード構成例

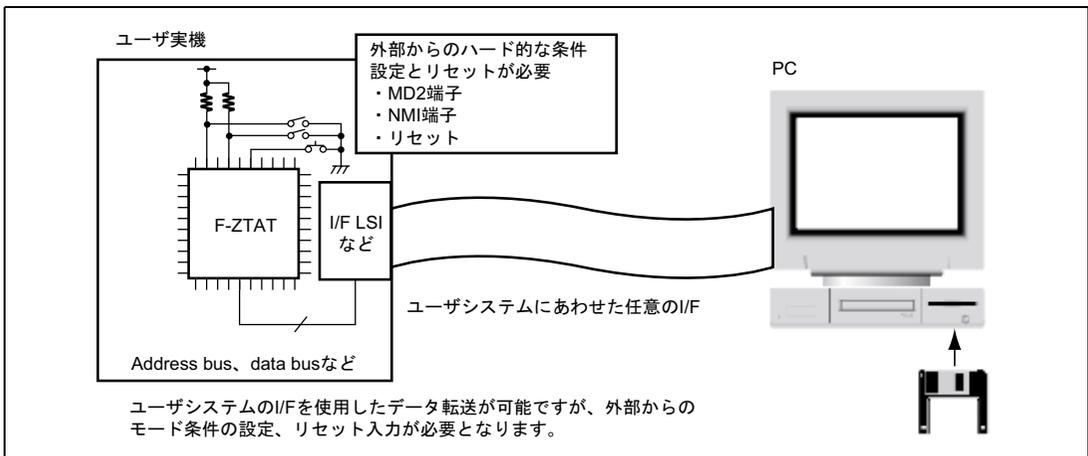


図 1.2 ユーザブートモード構成例

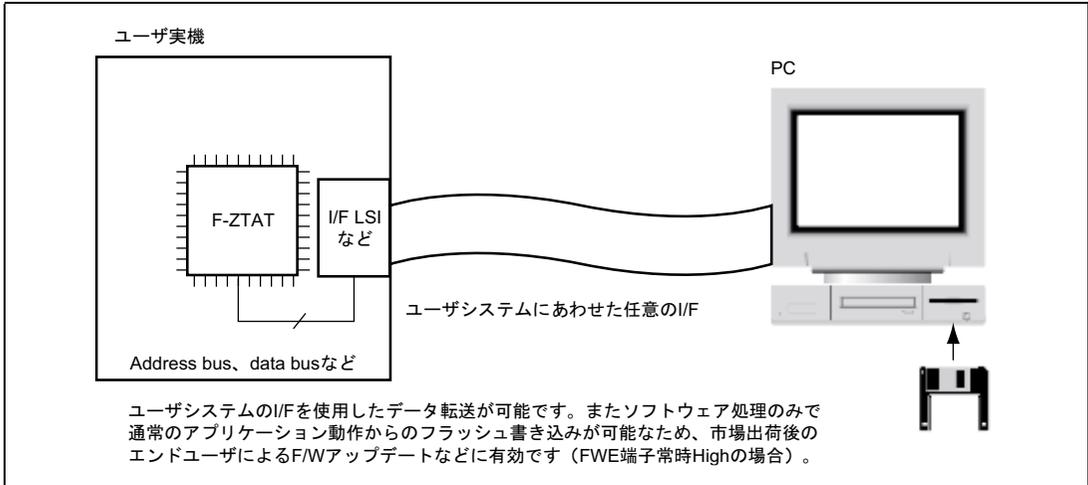


図 1.3 ユーザプログラムモード構成例

1.2 ユーザプログラムモード

H8/3069Fでは、消去／書き込みの制御プログラムがマイコンに内蔵されています。このため、処理の複雑な消去／書き込み制御プログラムを作成する必要はありません。これら内蔵プログラムのダウンロード（内蔵RAMへの転送）の要求、消去／書き込みの手順、結果の判定などを行う簡単な手続きプログラムを作成するだけで、容易にフラッシュメモリのオンボード書き換えを実現することができます。なお、手続きプログラムは、あらかじめブートモードまたはPROMモードでマイコンに書き込んでおく必要があります。ユーザプログラムモードにおける消去／書き込み手順を図1.4(1)～(4)に示します。

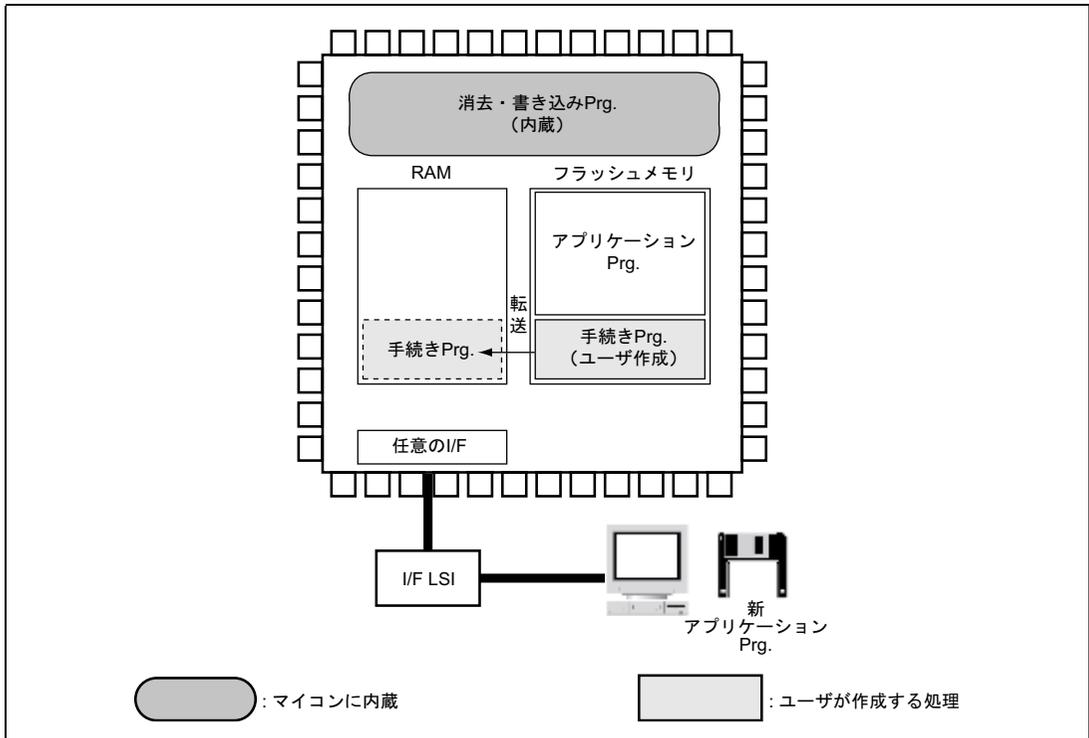


図 1.4(1) ユーザプログラムモードにおける消去／書き込み手順 1

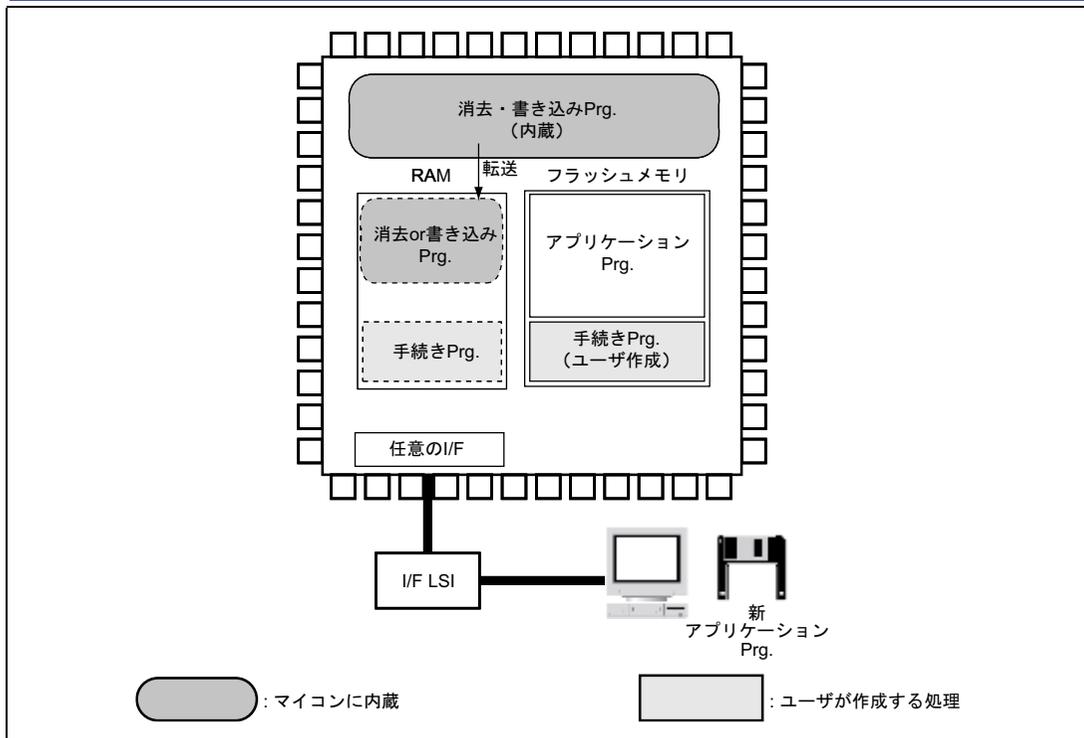


図 1.4(2) ユーザプログラムモードにおける消去／書き込み手順 2

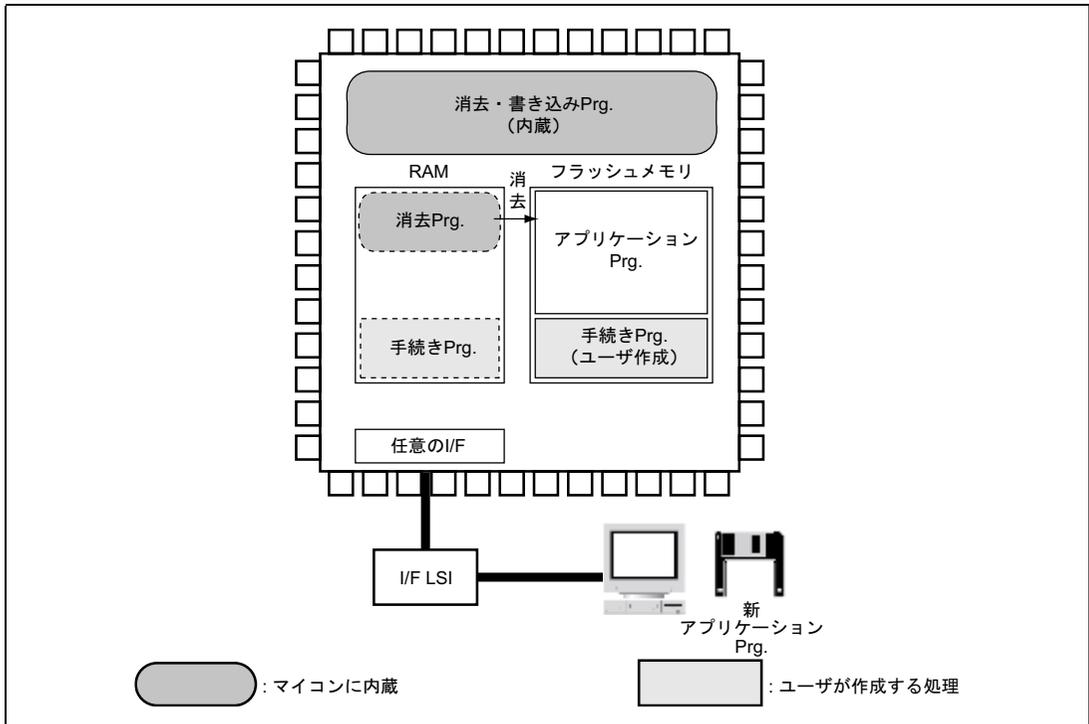


図 1.4(3) ユーザプログラムモードにおける消去／書き込み手順 3

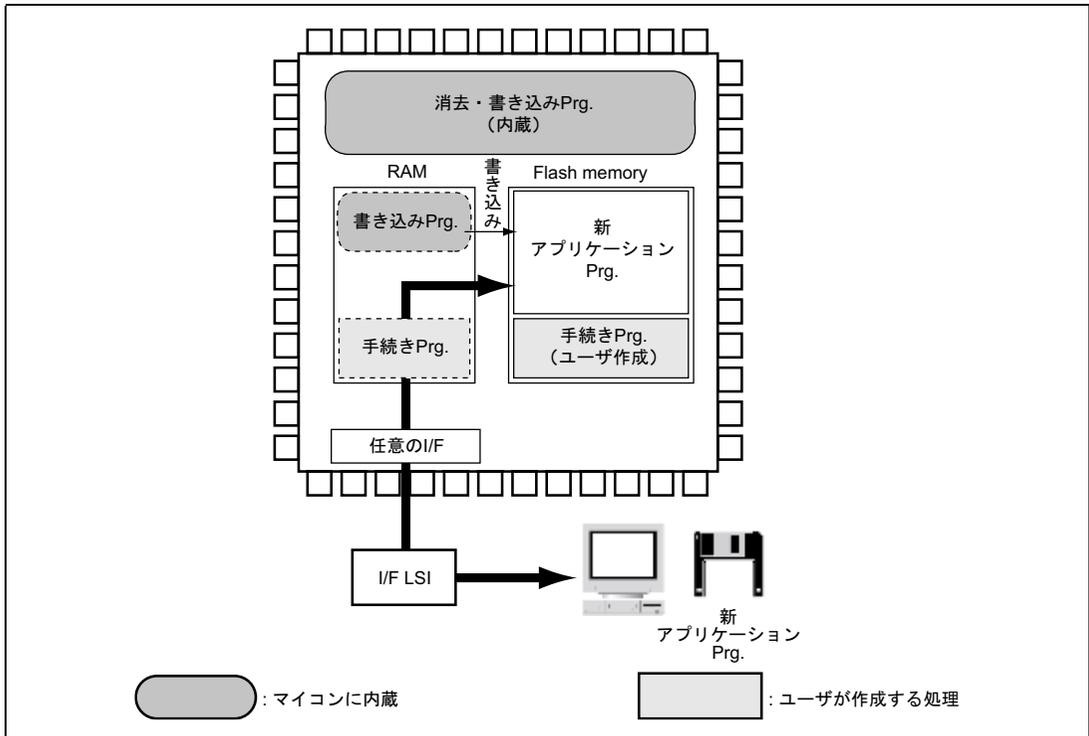


図 1.4(4) ユーザプログラムモードにおける消去／書き込み手順 4

(1) 手続きプログラムのRAM転送

ユーザプログラムモードによるフラッシュメモリの書き換えを行う前に、手続きプログラムを内蔵RAMへ転送し、RAM上で実行を開始します。手続きプログラムは、消去／書き込み対象となるフラッシュメモリ以外で実行する必要があります。

(2) 消去／書き込みプログラムの選択とRAM転送（ダウンロード）

手続きプログラムは、消去／書き込みどちらの処理を実行するかによって、消去プログラムか、書き込みプログラムかを選択し、転送先の内蔵RAMアドレスの選択を行い、転送（ダウンロード）を開始します。

(3) 消去処理の実行

内蔵プログラムをダウンロード後、初期化および消去プログラムを実行します。これらの実行はサブルーチンコール形式で行います。初期化では動作周波数、消去では消去対象となるブロック番号をパラメータに設定します（消去は消去ブロックと呼ばれるブロック単位で行います）。

(4) 書き込み処理の実行

内蔵プログラムをダウンロード後、初期化および書き込みプログラムを実行します。これらの実行はサブルーチンコール形式で行います。初期化では動作周波数、書き込みでは書き込みデータ／書き込み先アドレスをパラメータに設定します（書き込みは128バイト単位で行います）。

フラッシュメモリの書き込みには事前に書き込み対象となる領域が消去されている必要があります。

これら一連の消去／書き込み手順と手続きプログラムの概略フローを図1.5に示します。

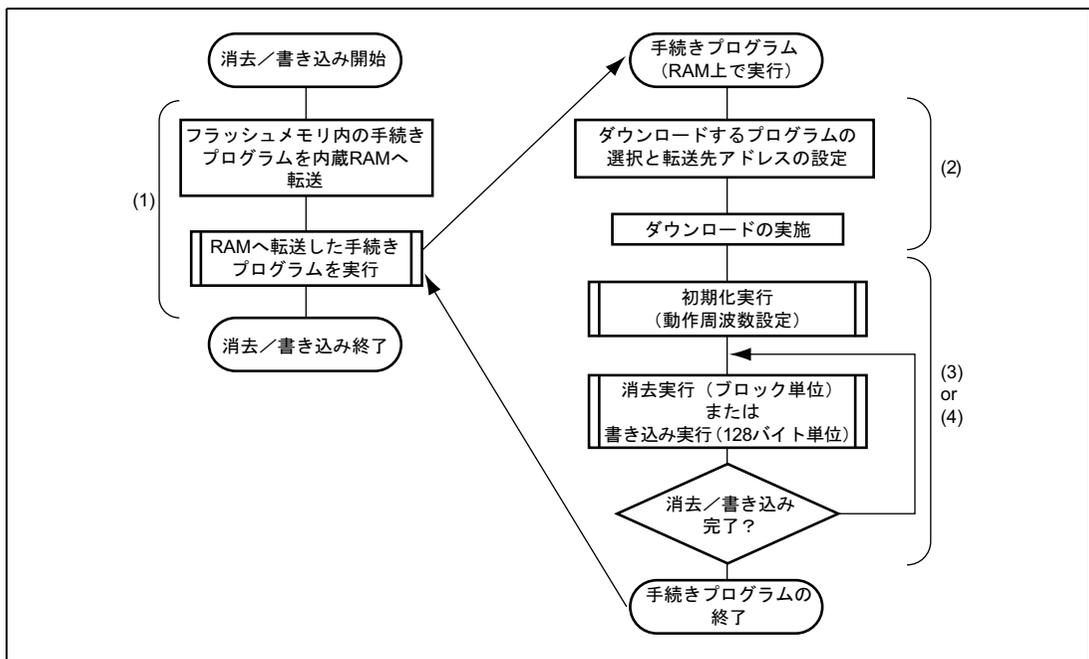


図 1.5 手続きプログラムの概略フロー

2. ユーザプログラムモード応用例

2.1 概要

H8/3069Fのユーザプログラムモードによるオンボード書き換え方法を、実際に動作するハードウェア、ソフトウェアを例にあげて説明します。

図2.1に本応用例におけるシステム構成を示します。

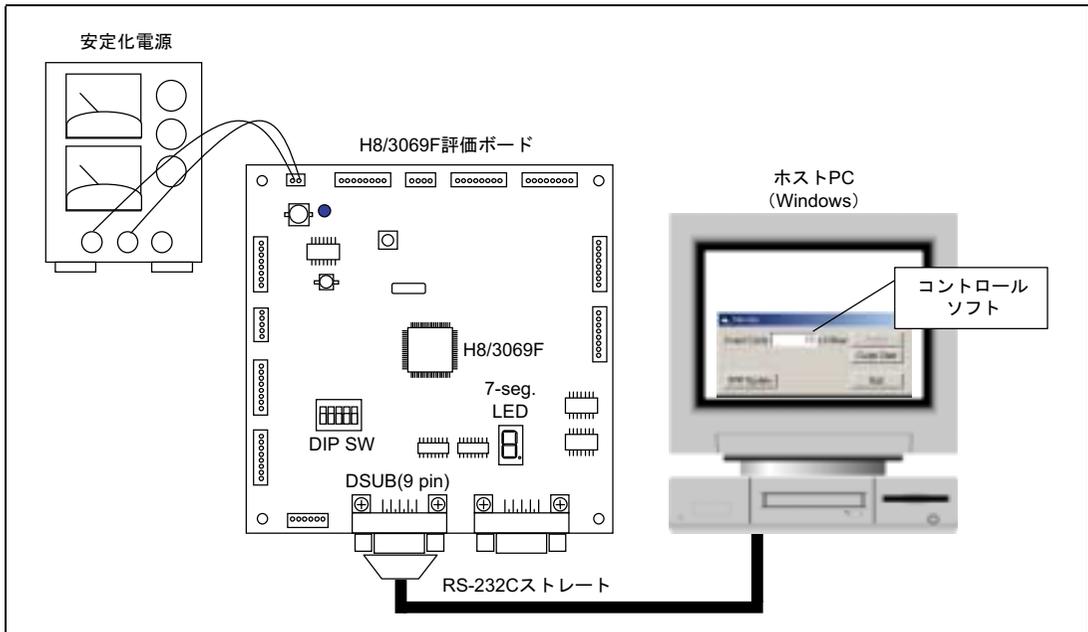


図 2.1 ユーザプログラムモード応用例システム構成

本システムは、H8/3069Fを搭載した評価ボードと、Windows が動作するホストPC、Windows上で動作するコントロールソフトで構成されます。評価ボードには7 Segment LED、モード設定用ディップスイッチが搭載されています。また、評価ボードとホストPCはRS232Cケーブルにより接続され、ホストPC上のコントロールソフトにより評価ボードとの調歩同期式シリアル通信が可能となっています。

2.2 機能

本システムは次に示す機能を有します。

1. 一定周期でH'0~H'Fまでのカウントアップを行い、7Segment LEDへ表示
2. カウントアップ周期の設定、カウントアップ動作の実行/停止をホストPCのコントロールソフトから制御
3. コントロールソフトによる、H8/3069F評価ボードのF/Wの書き換え

2.3 仕様

2.3.1 通信仕様

図2.2にコントロールソフトとH8/3069F評価ボード間の通信仕様を示します。

【仕様】
 通信モード : 調歩同期式モード
 ビットレート : 38400 bps
 データ長 : 8ビット
 ストップビット長 : 1ビット
 パリティ : なし

【コマンド/レスポンスフォーマット】
コマンド

コード (1バイト)	パラメータ (0~128バイト)
H'xx	}

ホストPCからは、1バイトのコマンドコードと0~128バイトのパラメータで構成されたフォーマットでコマンドが要求されます。

レスポンス

コード (1バイト)
H'xx

コマンドに対するレスポンスコードを1バイトで返します。

【コマンド】

No	コマンドコード	パラメータ	コマンド名	説明
1	H'42	なし	カウント開始	カウント動作を開始します。
2	H'53	なし	カウント停止	カウント動作を停止します。
3	H'43	時間 [100 ms] (2バイト固定)	カウント周期設定	カウント周期を100ms単位で設定します。
4	H'45	なし	F/W書き換え開始	F/W書き換え動作を開始するためのトリガです。
5	H'57	F/Wデータ (128バイト固定)	F/Wデータ転送	F/W書き換えのための新F/Wデータを転送します。
6	H'46	SUM値 (2バイト固定)	F/W書き換え終了	F/W書き換えのための新F/Wデータ転送が終了したことを通知します。

[Command]

No	レスポンスコード	レスポンス名	説明
1	H'06	ACK	コントロールソフトからのコマンドを受け付けました。
2N	H'15	AK	コントロールソフトからのコマンドに対しエラーが発生しました。

図 2.2 コントロールソフト⇄H8/3069F 評価ボード間通信仕様

2.3.2 LED 表示仕様

図2.3にLED表示仕様を示します。

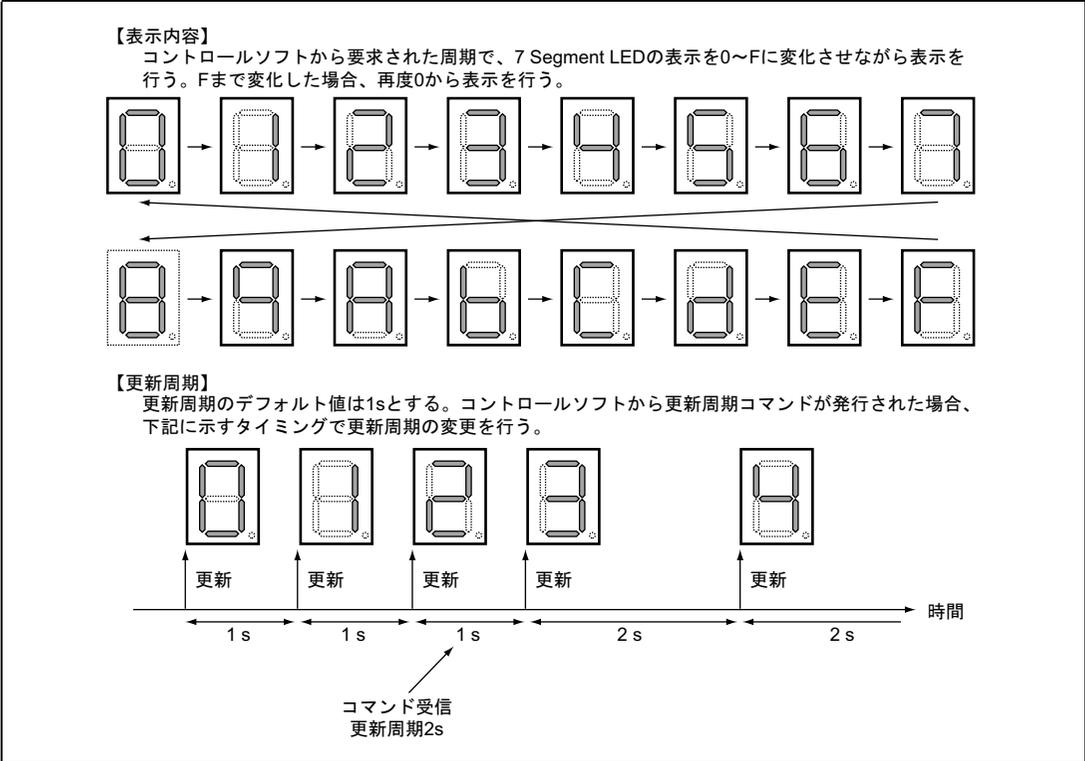


図 2.3 LED 表示仕様

2.3.3 フラッシュ消去／書き込み仕様

図2.4にフラッシュ消去／書き込み仕様を示します。

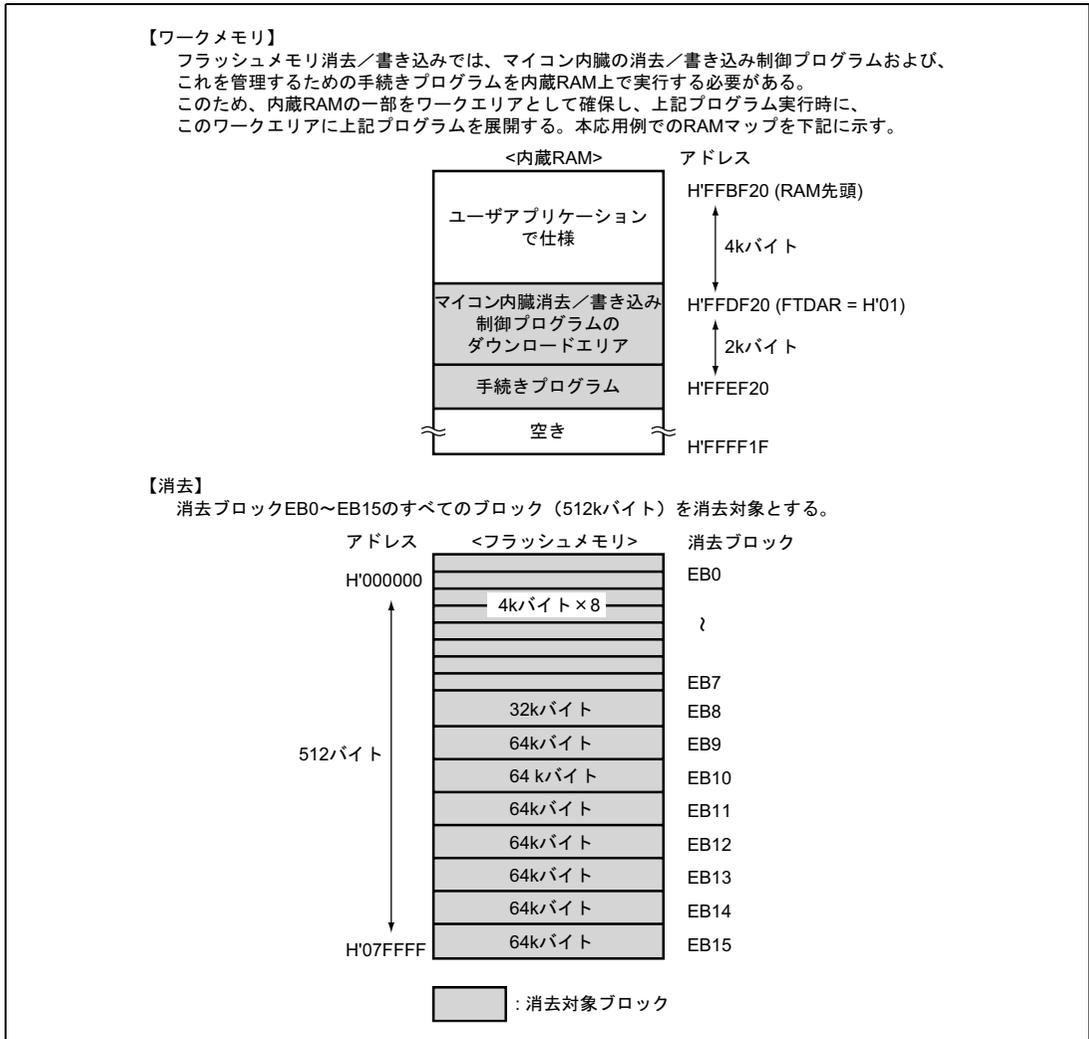


図 2.4(1) フラッシュ消去／書き込み仕様

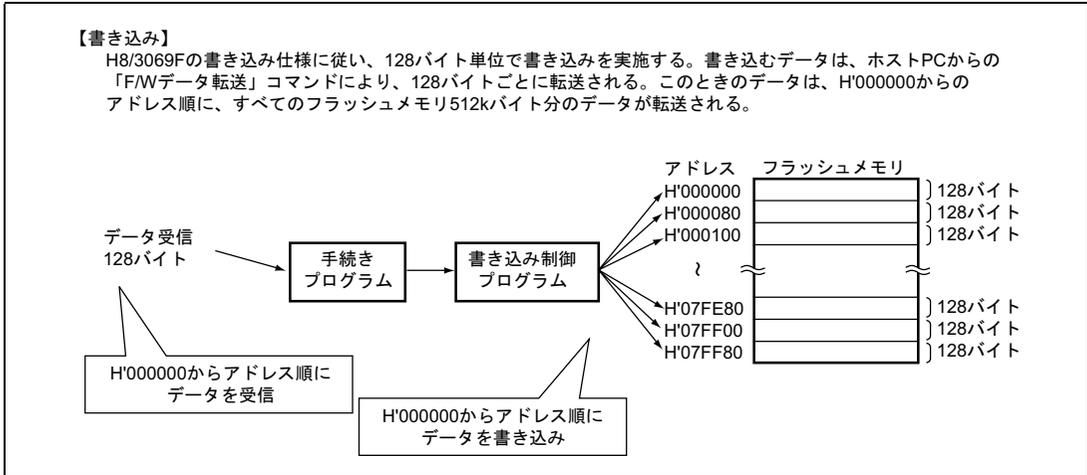


図 2.4(2) フラッシュ消去／書き込み仕様

2.4 ハードウェア

図2.5にH8/3069F評価ボードのハードウェア構成を示します。

本評価ボードはH8/3069Fのユーザプログラムモードにおける、オンボード消去／書き込み評価を目的としたマイコンボードです。マイコンの動作モードを切り替えるためのDip-SWやボード単体での簡単な動作確認が行えるよう7Segment LEDを搭載し、RS232Cドライバ/D-SUB 9pinコネクタを搭載することで、RS232CによるPCとの接続も容易となっています。

また、汎用マイコンボードとしてもご利用いただけるよう、すべてのポートに対し外部コネクタを用意しています。

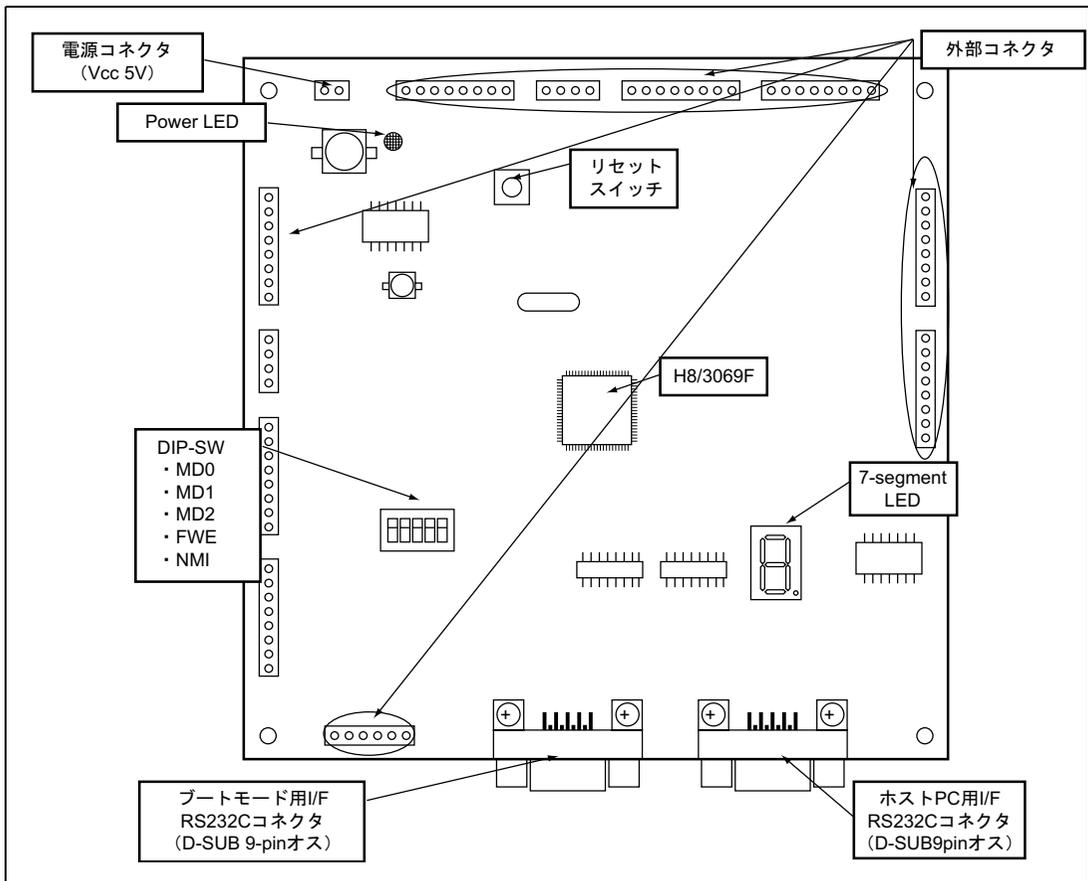


図 2.5 H8/3069F 評価ボード ハードウェア構成図

表2.1にH8/3069F評価ボードのハードウェア仕様を示します。

表 2.1 H8/3069F 評価ボードハードウェア仕様

No.	機能	接続先 ポート	説明																																																																																																																																																																										
1	マイコン	—	Vcc : 5V, 動作周波数 : 25MHz																																																																																																																																																																										
2	Dip-SW	MD0	MD0~2 の組み合わせで MCU 動作モードを設定します。 本応用例では、モード7 (シングルチップアドバンスモード MD0=H,MD1=H,MD2=H) で使用します。																																																																																																																																																																										
3		MD1																																																																																																																																																																											
4		MD2																																																																																																																																																																											
5		FWE		FWE	フラッシュメモリの消去/書き込み時のハードウェアプロテクト端子です。 Low レベル入力時には消去/書き込みプロテクト状態となります。 本応用例では、High レベル固定で使用します。																																																																																																																																																																								
6	NMI	NMI	マスク不可能な外部割り込み端子です。 ブートモード(H)/ユーザブートモード(L)のモード起動条件としても使用します。																																																																																																																																																																										
7	7 Segment LED	Port4	LED 表示と Port4 の各ビットの対応は下表のとおりです。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th rowspan="2">LED 表示</th> <th colspan="8">Port4 [Bit No.]</th> </tr> <tr> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr><td>クリア</td><td>H</td><td>H</td><td>H</td><td>H</td><td>H</td><td>H</td><td>H</td><td>H</td></tr> <tr><td>0</td><td>H</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td></tr> <tr><td>1</td><td>H</td><td>H</td><td>H</td><td>H</td><td>H</td><td>L</td><td>L</td><td>H</td></tr> <tr><td>2</td><td>H</td><td>L</td><td>H</td><td>L</td><td>L</td><td>H</td><td>L</td><td>L</td></tr> <tr><td>3</td><td>H</td><td>L</td><td>H</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td></tr> <tr><td>4</td><td>H</td><td>L</td><td>L</td><td>H</td><td>H</td><td>L</td><td>L</td><td>H</td></tr> <tr><td>5</td><td>H</td><td>L</td><td>L</td><td>H</td><td>L</td><td>L</td><td>H</td><td>L</td></tr> <tr><td>6</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>H</td><td>L</td></tr> <tr><td>7</td><td>H</td><td>H</td><td>H</td><td>H</td><td>H</td><td>L</td><td>L</td><td>L</td></tr> <tr><td>8</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td></tr> <tr><td>9</td><td>H</td><td>L</td><td>L</td><td>H</td><td>H</td><td>L</td><td>L</td><td>L</td></tr> <tr><td>A</td><td>H</td><td>L</td><td>L</td><td>L</td><td>H</td><td>L</td><td>L</td><td>L</td></tr> <tr><td>B</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>H</td><td>H</td></tr> <tr><td>C</td><td>H</td><td>H</td><td>L</td><td>L</td><td>L</td><td>H</td><td>H</td><td>L</td></tr> <tr><td>D</td><td>H</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td><td>H</td></tr> <tr><td>E</td><td>H</td><td>L</td><td>L</td><td>L</td><td>L</td><td>H</td><td>H</td><td>L</td></tr> <tr><td>F</td><td>H</td><td>L</td><td>L</td><td>L</td><td>H</td><td>H</td><td>H</td><td>L</td></tr> </tbody> </table>	LED 表示	Port4 [Bit No.]								7	6	5	4	3	2	1	0	クリア	H	H	H	H	H	H	H	H	0	H	H	L	L	L	L	L	L	1	H	H	H	H	H	L	L	H	2	H	L	H	L	L	H	L	L	3	H	L	H	H	L	L	L	L	4	H	L	L	H	H	L	L	H	5	H	L	L	H	L	L	H	L	6	H	L	L	L	L	L	H	L	7	H	H	H	H	H	L	L	L	8	H	L	L	L	L	L	L	L	9	H	L	L	H	H	L	L	L	A	H	L	L	L	H	L	L	L	B	H	L	L	L	L	L	H	H	C	H	H	L	L	L	H	H	L	D	H	H	L	L	L	L	L	H	E	H	L	L	L	L	H	H	L	F	H	L	L	L	H	H	H	L
LED 表示	Port4 [Bit No.]																																																																																																																																																																												
	7	6	5	4	3	2	1	0																																																																																																																																																																					
クリア	H	H	H	H	H	H	H	H																																																																																																																																																																					
0	H	H	L	L	L	L	L	L																																																																																																																																																																					
1	H	H	H	H	H	L	L	H																																																																																																																																																																					
2	H	L	H	L	L	H	L	L																																																																																																																																																																					
3	H	L	H	H	L	L	L	L																																																																																																																																																																					
4	H	L	L	H	H	L	L	H																																																																																																																																																																					
5	H	L	L	H	L	L	H	L																																																																																																																																																																					
6	H	L	L	L	L	L	H	L																																																																																																																																																																					
7	H	H	H	H	H	L	L	L																																																																																																																																																																					
8	H	L	L	L	L	L	L	L																																																																																																																																																																					
9	H	L	L	H	H	L	L	L																																																																																																																																																																					
A	H	L	L	L	H	L	L	L																																																																																																																																																																					
B	H	L	L	L	L	L	H	H																																																																																																																																																																					
C	H	H	L	L	L	H	H	L																																																																																																																																																																					
D	H	H	L	L	L	L	L	H																																																																																																																																																																					
E	H	L	L	L	L	H	H	L																																																																																																																																																																					
F	H	L	L	L	H	H	H	L																																																																																																																																																																					
8	ホスト PC 用 I/F	TxD0 RxD0	本応用例では、ホスト PC と接続しコントロールソフトによる制御用 I/F として使用します。 汎用 RS232C I/F としても使用可能です。																																																																																																																																																																										
9	ブートモード用 I/F	TxD1 RxD1	ルネサステクノロジ製 F-ZTAT 書き込みツール (FDT) によるブートモードでの書き換え用コネクタです。 汎用 RS232C I/F としても使用可能です。																																																																																																																																																																										

2.5 ソフトウェア

図2.6に本応用例でのソフトウェア構成を示します。本ソフトウェアは、メイン処理、ホストPCのコントロールソフトからの各種コマンド受信とLED表示制御を行うアプリケーション処理、F/Wの書き換えを行うフラッシュ消去／書き込み処理から構成されます。

1. メイン処理
電源投入後の初期化や周辺機能の設定を行い、メインループにてアプリケーション処理、フラッシュ消去／書き込み処理の起動管理を行います。
2. アプリケーション処理
コントロールソフトからのカウント開始／停止コマンド、カウント周期設定コマンドの受信／実行を行います。また、F/Wの書き換えコマンド受信時には、フラッシュ消去／書き込み処理への起動要求を発行します。
3. フラッシュ消去／書き込み処理
H8/3069Fのユーザプログラムモードによる自F/Wの書き換え処理を行います。

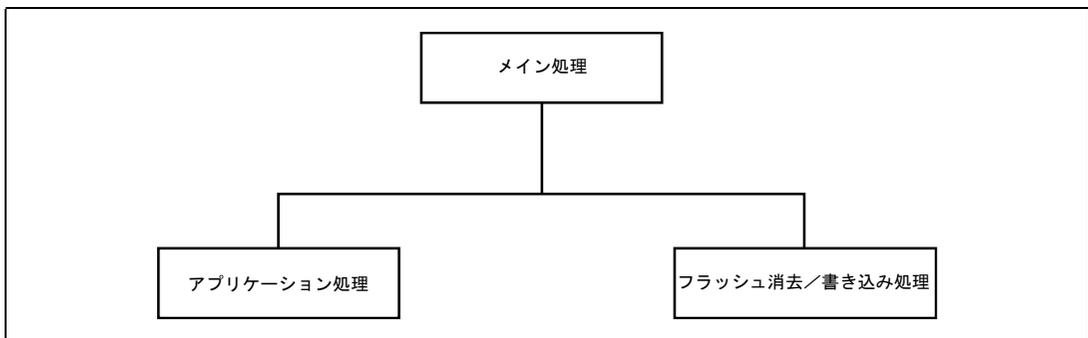


図 2.6 ソフトウェア構成

2.5.1 メイン処理

メイン処理では、電源投入後の初期化や周辺機能の設定を行い、アプリケーション処理、フラッシュ消去／書き込み処理の起動管理を行います。

メイン処理のフローチャートを図2.7に示します。

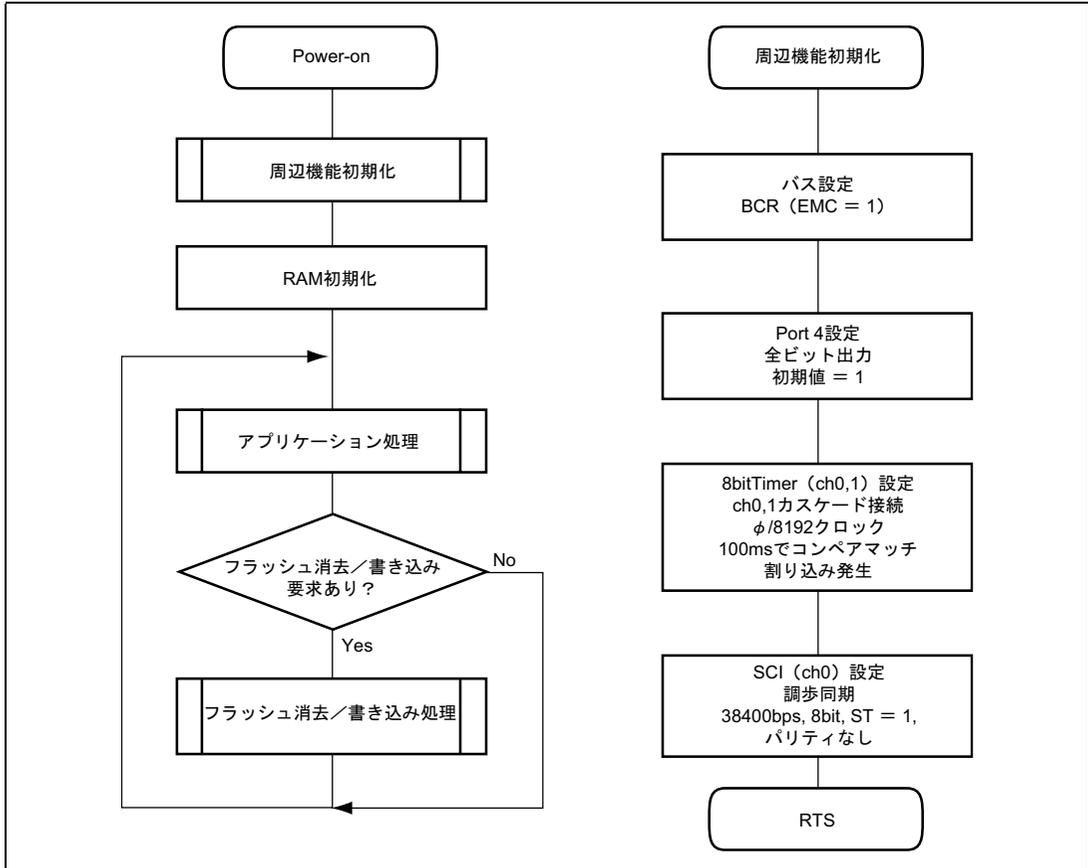


図 2.7 メイン処理フローチャート

2.5.2 アプリケーション処理

アプリケーション処理では、シリアル通信によるコントロールソフトからのコマンド受信、コマンド解析を行い、LED表示制御、フラッシュ消去／書き込み処理に対する起動要求を発行します。

カウント表示制御でのカウント周期は、8bit Timer ch0,1による100msの割り込みでチェックし、LED表示の更新を行います。

本処理内で処理するコマンドは、「カウント開始」、「カウント停止」、「カウント周期設定」、「F/W書き換え開始」コマンドです。「F/Wデータ転送」および「F/W書き換え終了」コマンドに関しては、フラッシュ消去／書き込み処理内で処理することとします。

図2.8にアプリケーション処理のフローチャートを示します。

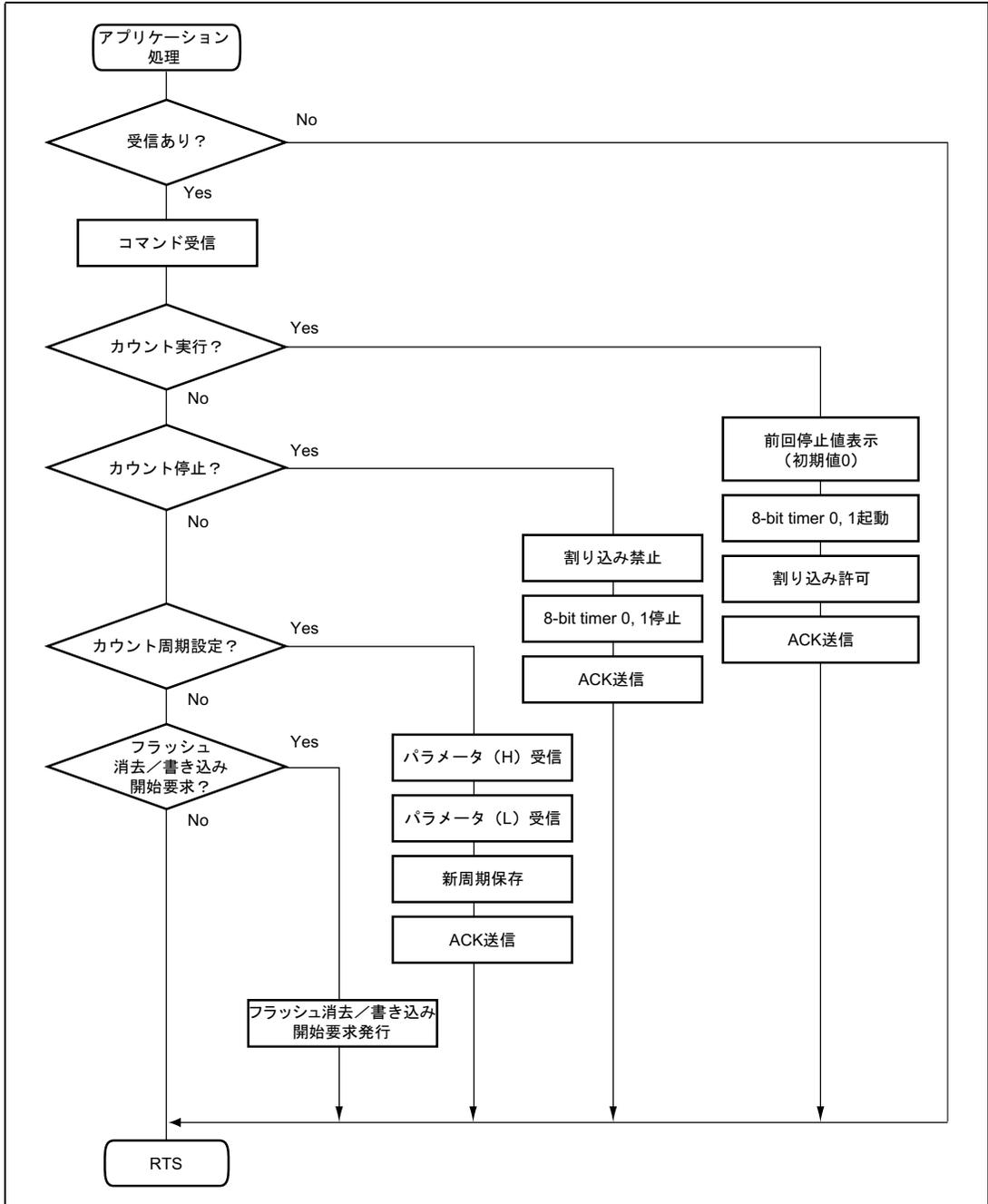


図 2.8(1) アプリケーション処理フローチャート

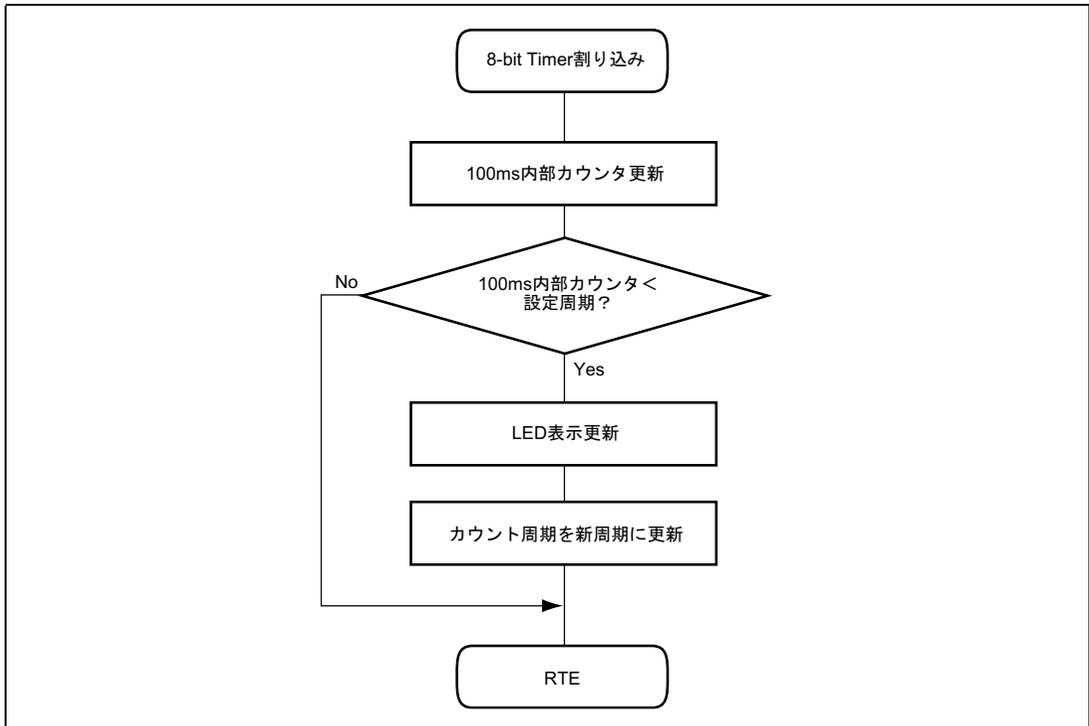
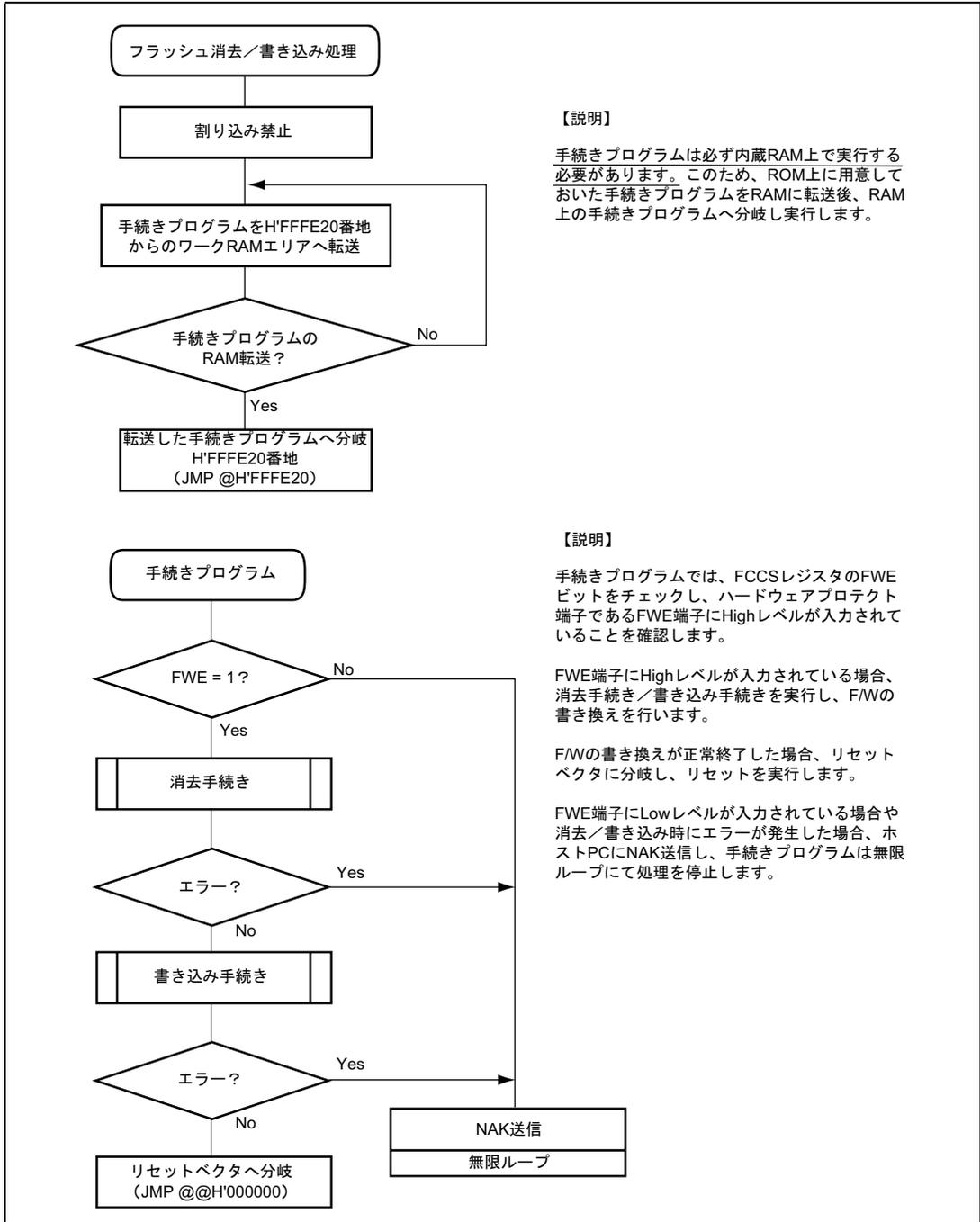


図 2.8(2) アプリケーション処理フローチャート

2.5.3 フラッシュ消去／書き込み処理

フラッシュメモリの消去／書き込みを行うための手続きプログラムです。H8/3069Fには消去／書き込みの制御プログラムがマイコンに内蔵されています。手続きプログラムでは、これら内蔵プログラムのダウンロード（内蔵RAMへの転送）要求、消去／書き込みの手順、結果の判定などを行います。手続きプログラムの実行は内蔵RAMで行う必要があります、また書き換えデータを受信するための「F/W書き換えデータ転送」および「F/W書き換え終了」コマンドに関するコマンド受信、解析、応答処理についても、内蔵RAM上に転送し実行する必要があります。

図2.9にフラッシュメモリ消去／書き込み処理のフローチャートを示します。



【説明】

手続きプログラムは必ず内蔵RAM上で実行する必要があります。このため、ROM上に用意しておいた手続きプログラムをRAMに転送後、RAM上の手続きプログラムへ分岐し実行します。

【説明】

手続きプログラムでは、FCCSレジスタのFWEビットをチェックし、ハードウェアプロテクト端子であるFWE端子にHighレベルが入力されていることを確認します。

FWE端子にHighレベルが入力されている場合、消去手続き／書き込み手続きを実行し、F/Wの書き換えを行います。

F/Wの書き換えが正常終了した場合、リセットベクタに分岐し、リセットを実行します。

FWE端子にLowレベルが入力されている場合や消去／書き込み時にエラーが発生した場合は、ホストPCにNAK送信し、手続きプログラムは無限ループにて処理を停止します。

図 2.9(1) フラッシュ消去／書き込み処理フローチャート

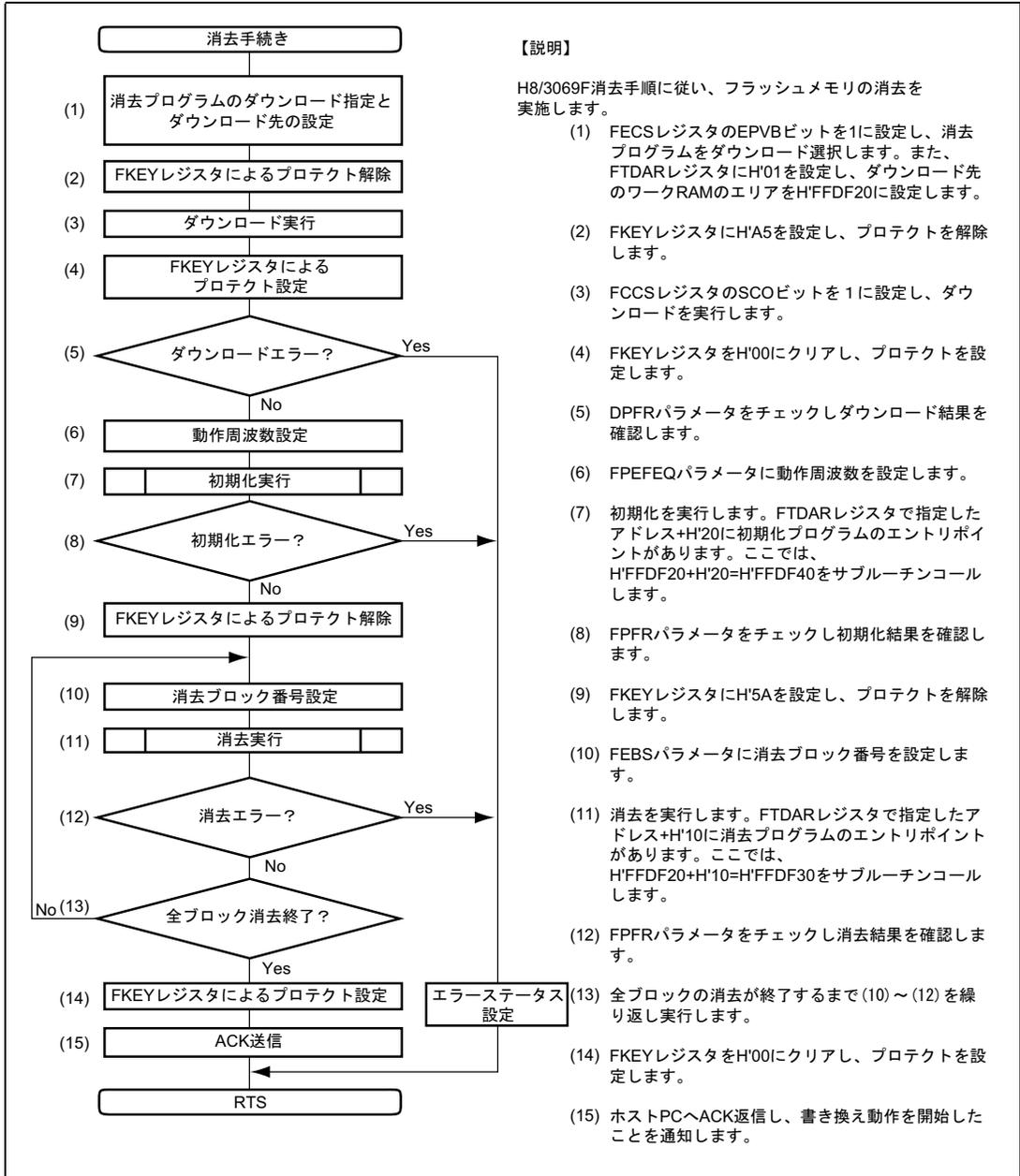


図 2.9(2) フラッシュ消去／書き込み処理フローチャート

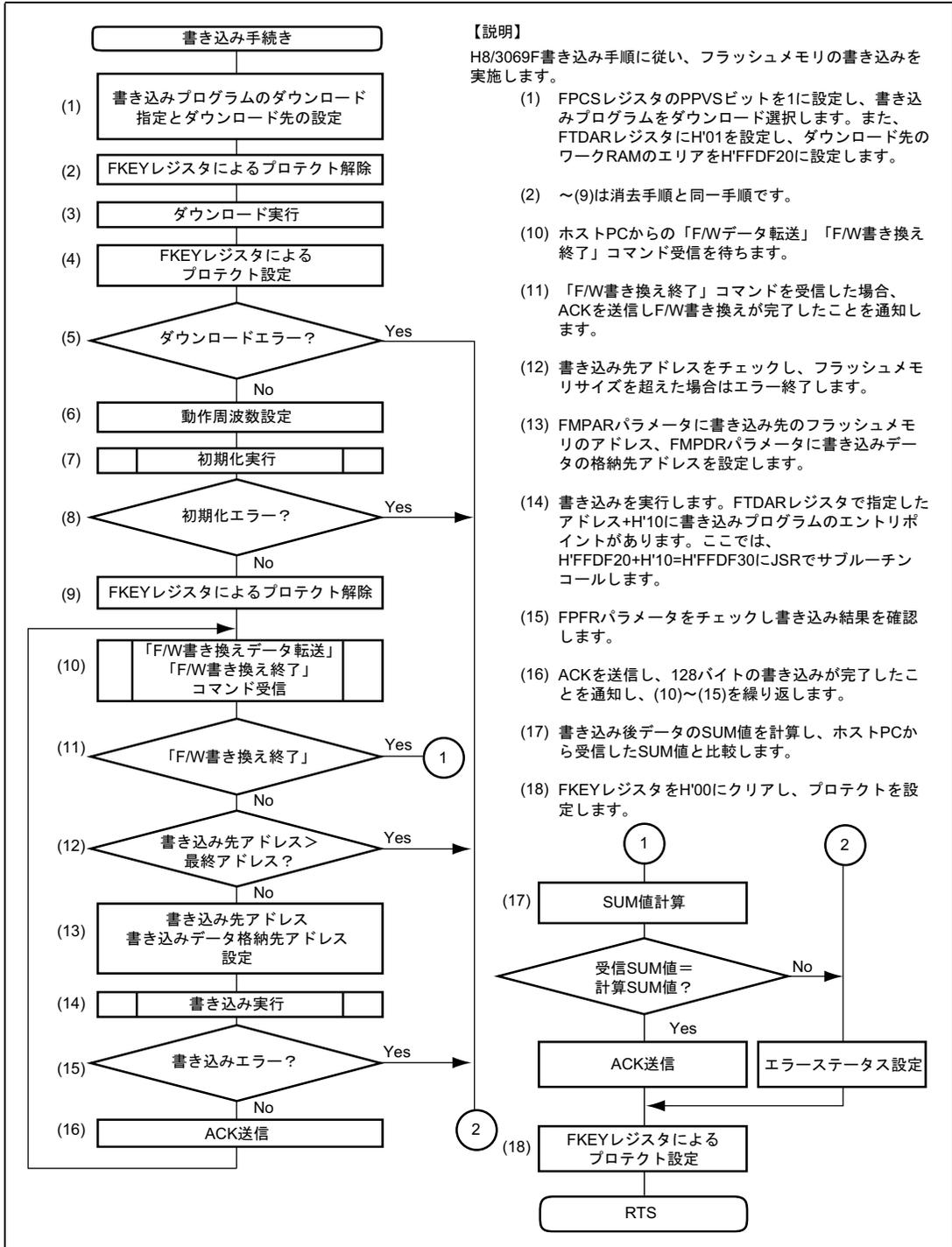


図 2.9(3) フラッシュ消去/書き込み処理フローチャート

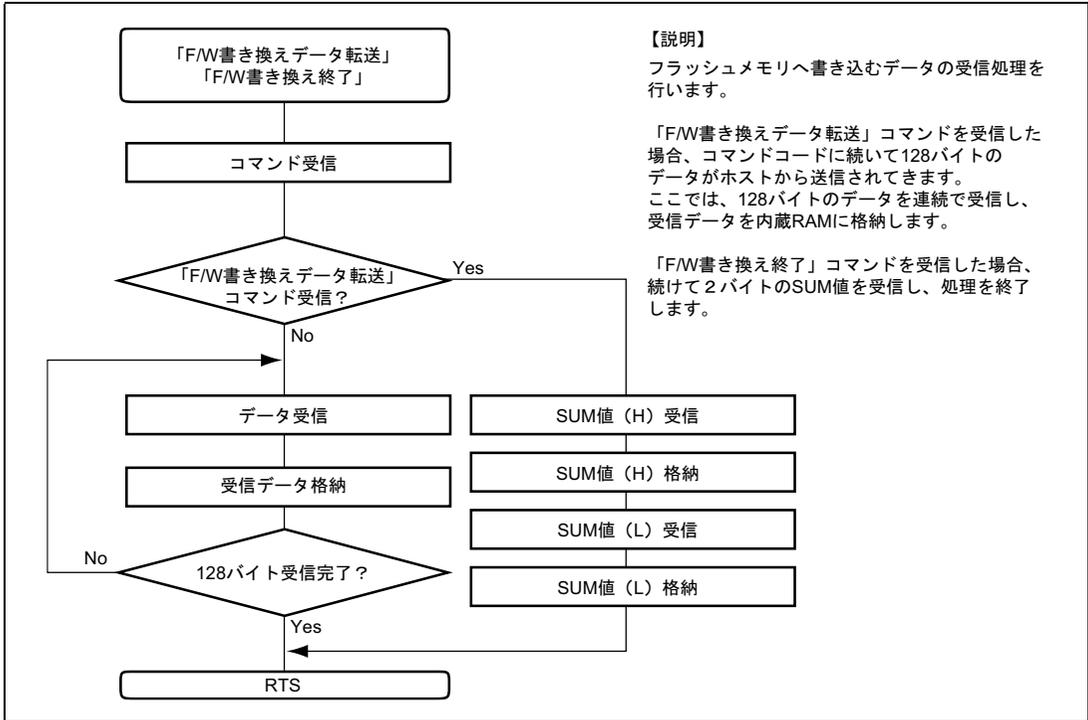


図 2.9(4) フラッシュ消去／書き込み処理フローチャート

2.6 コントロールソフト

H8/3069F評価ボードの通信相手となるコントロールソフトについて説明します。

コントロールソフトはWindows上で動作するH8/3069F評価ボード用制御アプリケーションです。RS232Cによる通信で各種コマンドを送信し、評価ボードを制御します。

図2.10にコントロールソフト画面を示します。

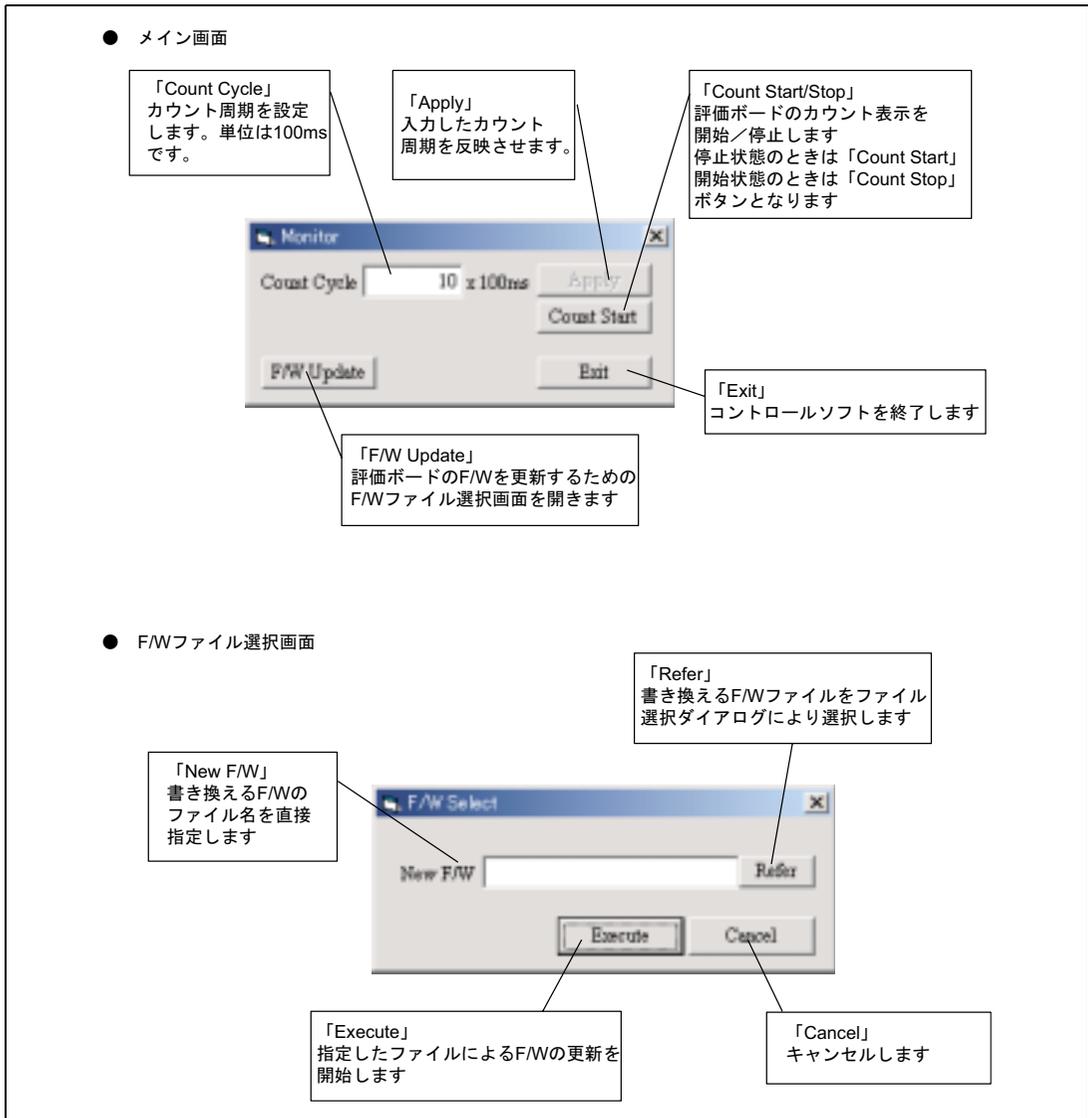


図 2.10 コントロールソフト

2.7 評価ボードによる動作例

これまで説明してきましたハードウェア、ソフトウェアを用いて、実際に本システムを動作させてみます。

2.7.1 準備

本システムを動作させるためには、あらかじめH8/3069FのフラッシュメモリにF/Wを書き込んでおく必要があります。H8/3069Fには、このようなフラッシュメモリの初期書き込みに対応するため、ブートモードと呼ぶオンボードプログラミングモードが存在します。

ブートモードによるフラッシュメモリの書き込みには、ある決められたインターフェース仕様で制御コマンドや書き込みデータを送信するツールを用意する必要があります。

ルネサステクノロジではブートモードを用いたオンボードFLASH書き込みツール「F-ZTATフラッシュ開発ツールキット(FDT)」をご用意しております。今回はこのFDTを用いてフラッシュメモリの初期書き込みを実施します。

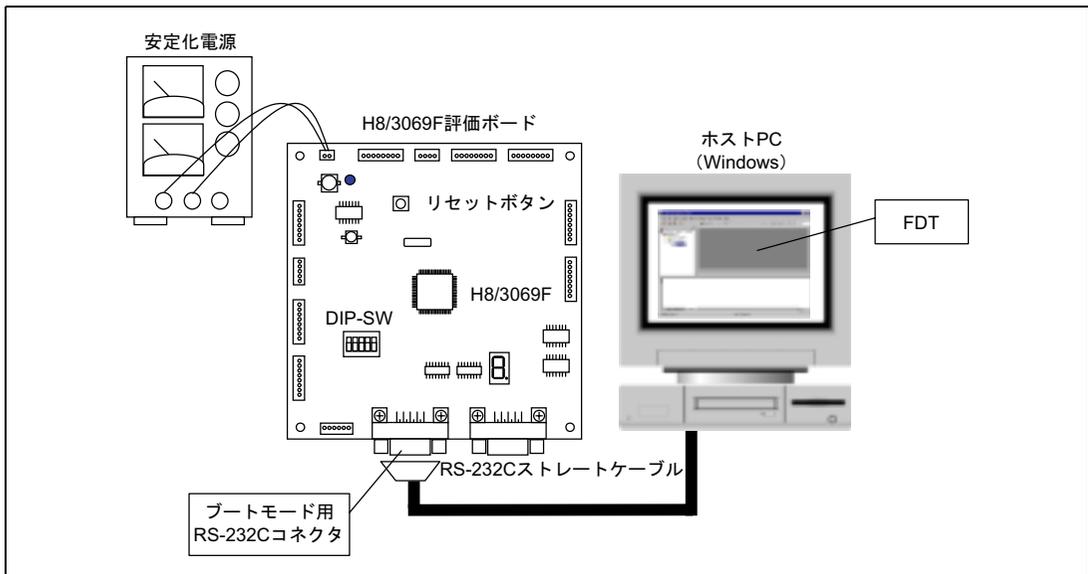


図 2.11 ブートモードによる初期書き込み

(1) 接続

H8/3069F評価ボードを安定化電源に接続し+5Vを供給します。また、評価ボードのブートモード用RS232CコネクタとホストPCのCOMポートをRS232Cストレートケーブルで接続します。

ホストPCではFDTを起動し新規ワークスペースを作成しておきます。FDTの操作方法に関しては、FDTのマニュアルを参照してください。

(2) ブートモードの起動

FDTによるフラッシュメモリの書き込みを行うためには、ターゲットデバイスであるH8/3069Fをブートモードで起動する必要があります。H8/3069F評価ボードでブートモードを起動するには、Dip-SWを表2.2に示す設定にし、リセットボタンを押下してください。

表 2.2 H8/3069F 評価ボード ブートモード時の Dip-SW 設定

Dip-SW 名	MD0	MD1	MD2	FWE	NMI
設定	High	High	Low	High	High

(3) F/W の書き込み

FDTによる書き込みを実施してください。FDTの操作方法に関しては、FDTのマニュアルを参照してください。

2.7.2 動作確認

アプリケーションとしての動作を確認します。

(1) 接続

RS232CケーブルをPCのCOM1ポートとH8/3069F評価ボードのHOST PC I/F用コネクタに接続してください。また、HOST PCではコントロールソフトを起動してください。

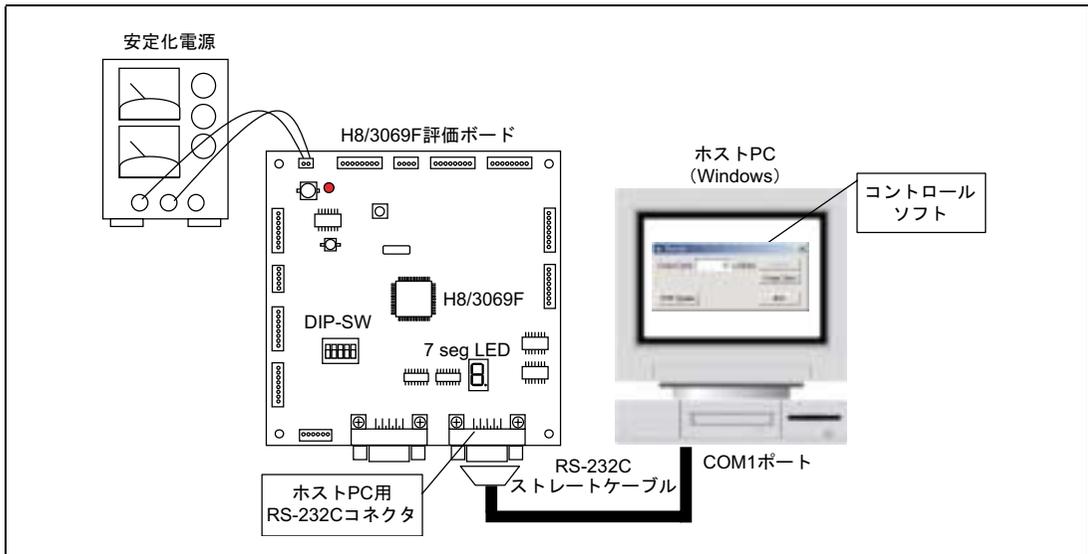


図 2.12 アプリケーション動作時の接続

(2) 再起動

Dip-SWを表2.3に示す設定に変更後、再度リセットスイッチを押下してください。本設定によりH8/3069Fはモード7で動作を開始します。本応用例では、ハードウェアプロテクト端子であるFWE端子をHighレベル（プロテクト解除状態）のまま使用します。

表 2.3 H8/3069F 評価ボード モード7のDip-SW 設定

Dip-SW 名	MD0	MD1	MD2	FWE	NMI
設定	High	High	High	High	High

(3) コントロールソフトの操作

コントロールソフトで評価ボードの動作を制御します。

コントロールソフトの画面から「Count Start」ボタンをクリックすると、評価ボード上のLEDが”0”～”F”まで1s周期でカウントアップを開始します。（図2.13）

「Count Cycle」に新しいカウントアップ周期(100ms単位)を入力し、「Apply」ボタンをクリックするとLEDのカウントアップ周期が変更されます。（図2.14）

コントロールソフトの画面から「Count Stop」ボタンをクリックすると、LEDのカウントアップが停止します。（図2.15）

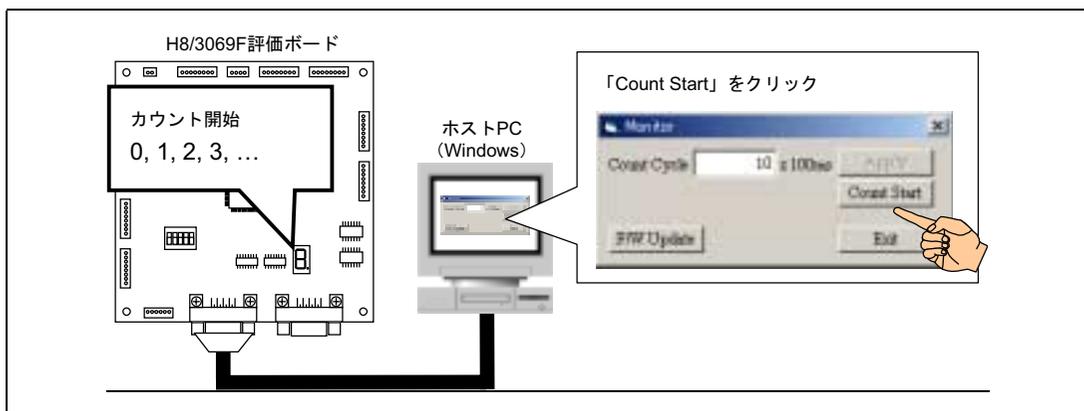


図 2.13 カウント開始

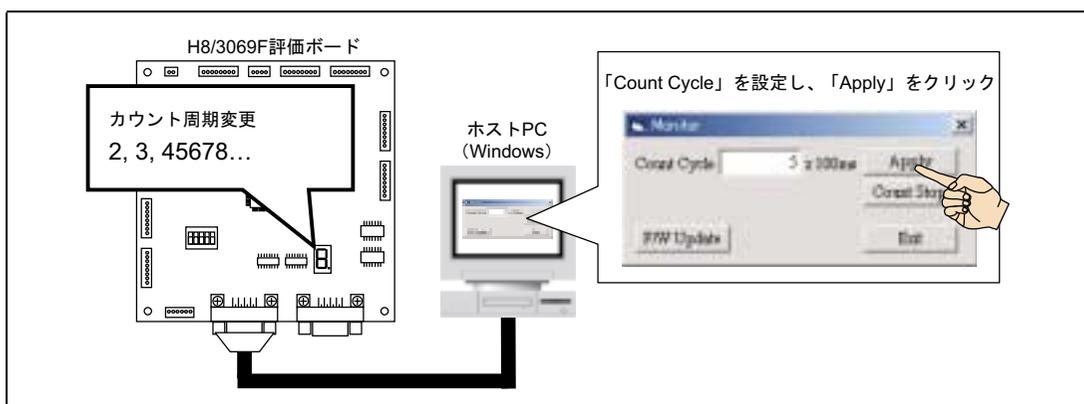


図 2.14 カウント周期変更

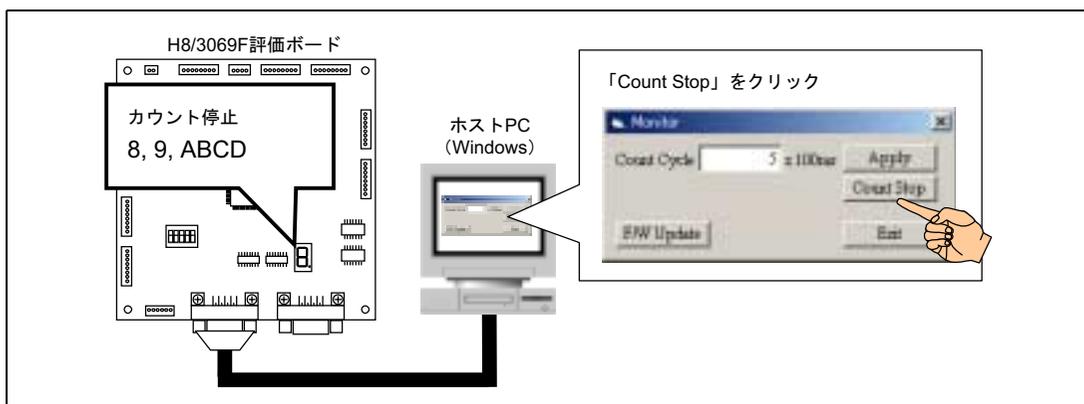


図 2.15 カウント停止

2.7.3 F/W 書き換えの実行

本応用例では、F/W書き換え後の動作の違いをわかりやすくするため、LED表示をカウントダウン表示に変更したF/Wをあらかじめ用意しておきます。カウントダウン表示への変更方法は、添付のソースファイル中に記述してあります。

(1) 新 F/W の選択

コントロールソフトの画面から「F/W Update」ボタンをクリックします。「F/W Select」ダイアログが開きますので、カウントダウン表示版F/Wのファイル指定を直接行うか、「Refer」ボタンをクリックしファイル選択ダイアログによる選択を行ってください（図2.16）。

(2) F/W 書き換え実行

「Execute」ボタンをクリックすると、F/Wの書き換えを開始します（図2.17）。

(3) F/W 書き換え終了

F/Wの書き換えが正常終了すると、「Complete」メッセージボックスが表示され、評価ボードが自動的にリセットされます（図2.18）。

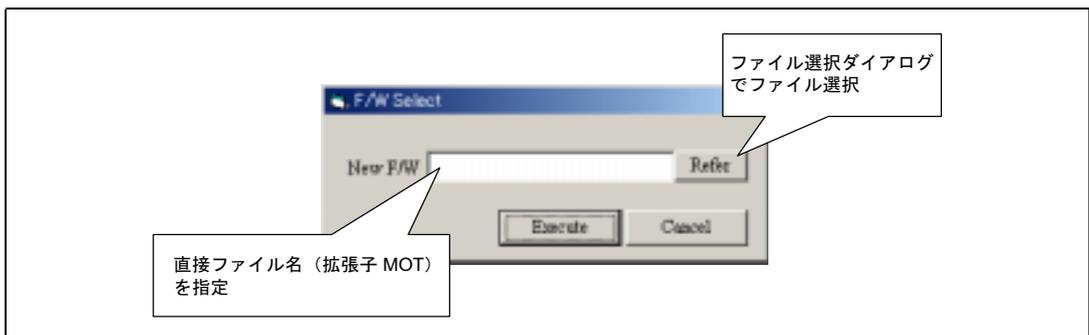


図 2.16 新 F/W の選択



図 2.17 F/W 書き換え実行

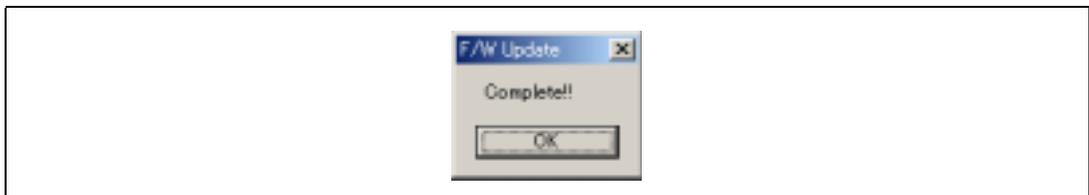


図 2.18 F/W 書き換え終了

2.7.4 F/W 書き換え後の動作確認

F/W書き換え後のアプリケーション動作を確認します。

「2.7.2 動作確認」の「(3)コントロールソフトの操作」と同様の手順でカウントを開始してください。F/W書き換え前、評価ボード上のLEDは“0・1・2・3…”とカウントアップ表示を行っていましたが、F/W書き換え後は“F・E・D・C…”とカウントダウン表示となっていることが確認できます。

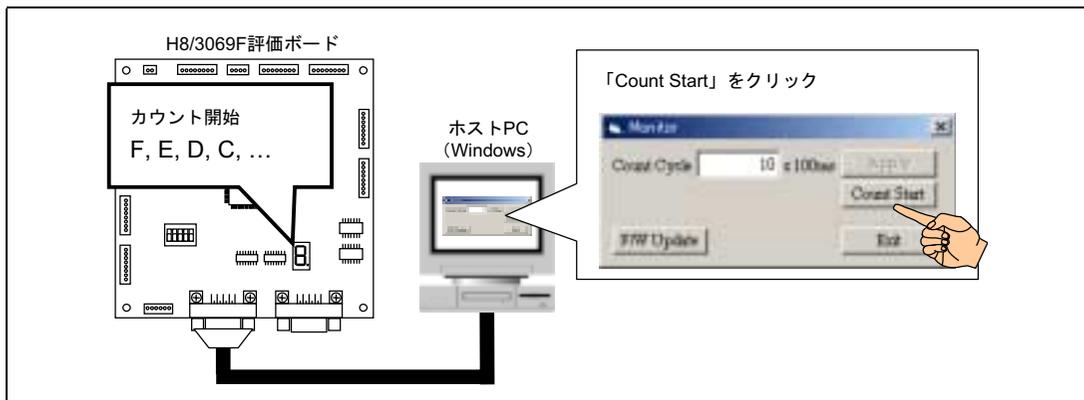
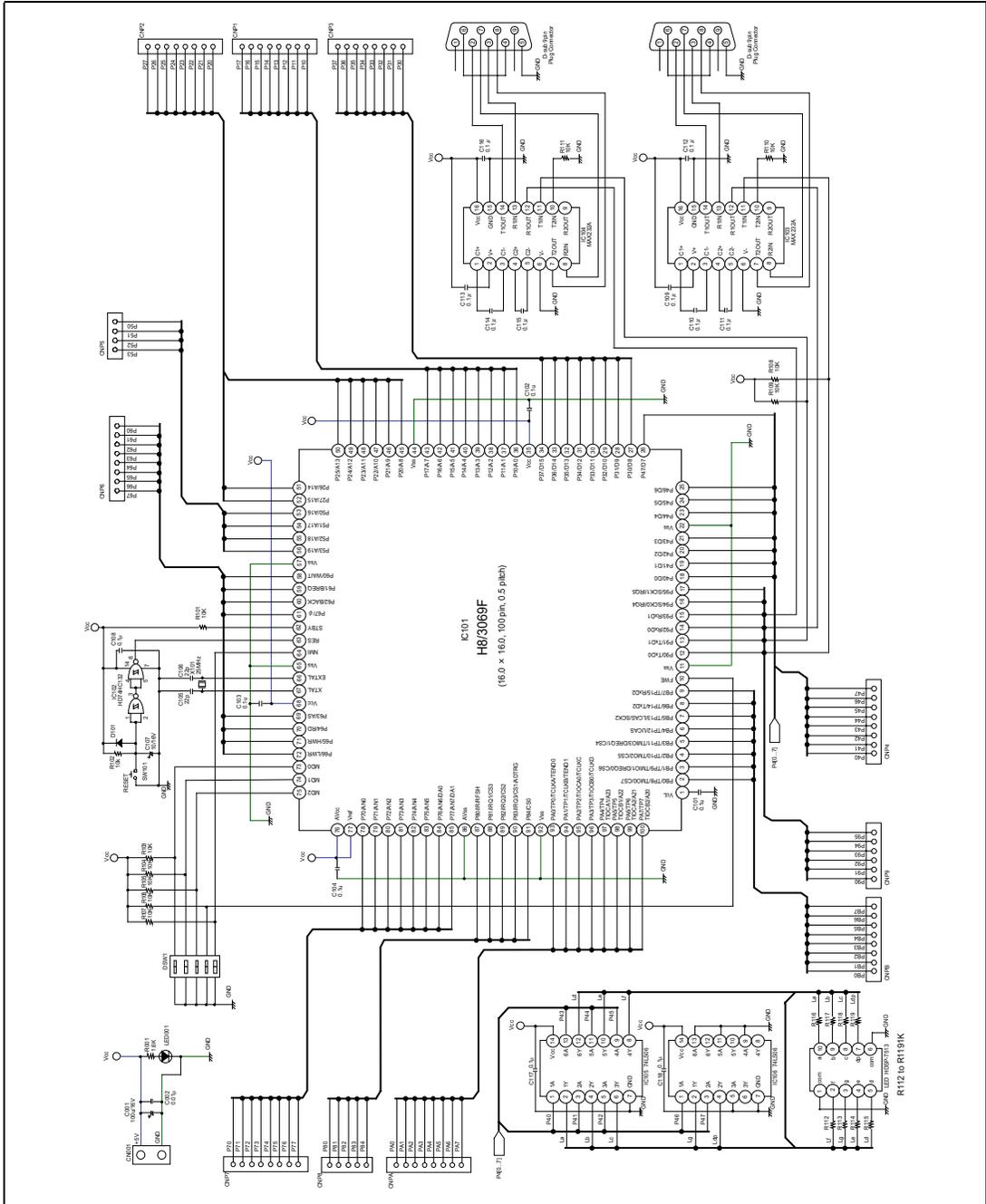


図 2.19 F/W 書き換え後の動作

3. 付録

3.1 H8/3069F 評価ボード回路図



3.2 全ソースリスト

```

;*****/
;/*
;/* H8/3069F User Program Mode Application Note */
;/* Sample software */
;/* Main routine */
;/* */
;*****/
.CPU 300HA:24

.INCLUDE "iodef.inc"
.INCLUDE "common.inc"

.IMPORT _FWUpdate
.IMPORT _SCIRecv
.IMPORT _SCISend

.SECTION Vector,DATA,ALIGN=2;
;=====
; Vector table
;=====
.DATA.L _Main ; No.0 Reset
.DATA.L _nouse ; No.1 System reserved
.DATA.L _nouse ; No.2 System reserved
.DATA.L _nouse ; No.3 System reserved
.DATA.L _nouse ; No.4 System reserved
.DATA.L _nouse ; No.5 System reserved
.DATA.L _nouse ; No.6 System reserved
.DATA.L _nouse ; No.7 NMI
.DATA.L _nouse ; No.8 TRAP0
.DATA.L _nouse ; No.9 TRAP1
.DATA.L _nouse ; No.10 TRAP2
.DATA.L _nouse ; No.11 TRAP3
.DATA.L _nouse ; No.12 IRQ0
.DATA.L _nouse ; No.13 IRQ1
.DATA.L _nouse ; No.14 IRQ2
.DATA.L _nouse ; No.15 IRQ3
.DATA.L _nouse ; No.16 IRQ4
.DATA.L _nouse ; No.17 IRQ5
.DATA.L _nouse ; No.18 System reserved
.DATA.L _nouse ; No.19 System reserved
.DATA.L _nouse ; No.20 WOVI
.DATA.L _nouse ; No.21 CMI
.DATA.L _nouse ; No.22 System reserved
.DATA.L _nouse ; No.23 ADI
.DATA.L _nouse ; No.24 IMIA0
.DATA.L _nouse ; No.25 IMIB0
.DATA.L _nouse ; No.26 OVIO
.DATA.L _nouse ; No.27 System reserved
.DATA.L _nouse ; No.28 IMIA1
.DATA.L _nouse ; No.29 IMIB1
.DATA.L _nouse ; No.30 OV11
.DATA.L _nouse ; No.31 System reserved

```

```

.DATA.L    _nouse                ; No.32 IMIA2
.DATA.L    _nouse                ; No.33 IMIB2
.DATA.L    _nouse                ; No.34 OVI2
.DATA.L    _nouse                ; No.35 System reserved
.DATA.L    _countCycleInt       ; No.36 CMIA0
.DATA.L    _nouse                ; No.37 CMIB0
.DATA.L    _nouse                ; No.38 CMIA1/CMIB1
.DATA.L    _nouse                ; No.39 TOVI0/TOVI1
.DATA.L    _nouse                ; No.40 CMIA2
.DATA.L    _nouse                ; No.41 CMIB2
.DATA.L    _nouse                ; No.42 CMIA3/CMIB3
.DATA.L    _nouse                ; No.43 TOVI2/TOVI3
.DATA.L    _nouse                ; No.44 DEND0A
.DATA.L    _nouse                ; No.45 DEND0B
.DATA.L    _nouse                ; No.46 DEND1A
.DATA.L    _nouse                ; No.47 DEND1B
.DATA.L    _nouse                ; No.48 System reserved
.DATA.L    _nouse                ; No.49 System reserved
.DATA.L    _nouse                ; No.50 System reserved
.DATA.L    _nouse                ; No.51 System reserved
.DATA.L    _nouse                ; No.52 ERI0
.DATA.L    _nouse                ; No.53 RXI0
.DATA.L    _nouse                ; No.54 TXI0
.DATA.L    _nouse                ; No.55 TEI0
.DATA.L    _nouse                ; No.56 ERI1
.DATA.L    _nouse                ; No.57 RXI1
.DATA.L    _nouse                ; No.58 TXI1
.DATA.L    _nouse                ; No.59 TEI1
.DATA.L    _nouse                ; No.60 ERI2
.DATA.L    _nouse                ; No.61 RXI2
.DATA.L    _nouse                ; No.62 TXI2
.DATA.L    _nouse                ; No.63 TEI2

.SECTION   C,DATA,ALIGN=2      ;
;=====
;   LED table
;=====
LEDTable:
.DATA.B    H'C0                ; "0"
.DATA.B    H'F9                ; "1"
.DATA.B    H'A4                ; "2"
.DATA.B    H'B0                ; "3"
.DATA.B    H'99                ; "4"
.DATA.B    H'92                ; "5"
.DATA.B    H'82                ; "6"
.DATA.B    H'F8                ; "7"
.DATA.B    H'80                ; "8"
.DATA.B    H'98                ; "9"
.DATA.B    H'88                ; "A"
.DATA.B    H'83                ; "B"
.DATA.B    H'C6                ; "C"
.DATA.B    H'A1                ; "D"
.DATA.B    H'86                ; "E"
.DATA.B    H'8E                ; "F"

```

```

        .SECTION    P, CODE, ALIGN=2            ;
;=====
;   Main routine
;=====
_Main:
    MOV.L    #STACKTOP, ER7                    ; Stack pointer set
;-----
;           Initialize
;-----
    BSR     _Initialize                        ; Initialize
;-----
;           Memroy clear
;-----
    MOV.W   #H'0000, R0                        ;
    MOV.L   #RAMTOP, ER6                      ;
CLRRAM:
    MOV.W   R0, @ER6                          ;
    INC.L   #2, ER6                            ;
    CMP.L   #RAMEND, ER6                      ;
    BLS     CLRRAM:8                          ;

    SUB.W   R0, R0                            ;
    MOV.B   R0L, @bytLEDCnt                   ;
    MOV.B   R0L, @flgFWUpdate                 ;
    MOV.W   R0, @wrdd100msCounter            ;
    MOV.W   #10, R0                           ;
    MOV.W   R0, @wrddTargetCycle             ;
    MOV.W   R0, @wrddNewTargetCycle         ;
;-----
;           Main loop
;-----
MAINLOOP:
    BSR     _Application                      ;
    MOV.B   @flgFWUpdate, R0L                 ; F/W Update request?
    BEQ     MAINLOOP:8                       ; No
    JSR     @_FWUpdate                       ; F/W Update
    MOV.B   #H'00, R0L                       ;
    MOV.B   R0L, @flgFWUpdate                 ; F/W Update request clear
    BRA     MAINLOOP:8                       ;
;=====
;   Initialize routine
;=====
_Initialize:
;-----
;           BUS setting
;-----
    MOV.B   @BCR, R0L                        ;
    BSET.B  #EMC, R0L                        ;
    BSET.B  #RDEA, R0L                      ;
    MOV.B   R0L, @BCR                        ;

```

```

;-----
;          PORT4 setting
;-----
MOV.B    #H'00,R0L          ;
MOV.B    R0L,@P4PCR        ; Internal Pull.up Off
MOV.B    #H'FF,R0L        ;
MOV.B    R0L,@P4DR        ; High Output
MOV.B    R0L,@P4DDR       ; Output direction
;-----
;          8bits Timer setting
;-----
MOV.B    #H'4C,R0L        ;
MOV.B    R0L,@T8TCR0      ; CMIA enable
MOV.B    #H'08,R0L        ;
MOV.B    R0L,@T8TCR1     ; 16bits count mode
MOV.W    #305,R0          ;
MOV.W    R0,@TCORA0       ; 100ms compare match
SUB.W    R0,R0            ;
MOV.W    R0,@T8TCNT0     ; Counter clear (Ch0,1)
;-----
;          SCI0 setting
;-----
MOV.B    #H'00,R0L        ;
MOV.B    R0L,@SCR0        ; TE,RE clear
MOV.B    R0L,@SMR0        ; ASYNC,8bits,NoParity,1STOP,CLK=fai
;; MOV.B    #80,R0L        ;
;; MOV.B    R0L,@BRR0      ; 9600bps
;; MOV.B    #40,R0L        ;
;; MOV.B    R0L,@BRR0      ; 19200bps
;; MOV.B    #19,R0L        ;
;; MOV.B    R0L,@BRR0      ; 38400bps
;; MOV.B    #12,R0L        ;
;; MOV.B    R0L,@BRR0      ; 57600bps
MOV.B    #6,R0L           ;
MOV.B    R0L,@BRR0        ; 115200bps
;; MOV.B    #5,R0L        ;
;; MOV.B    R0L,@BRR0      ; 128000bps
MOV.B    #H'30,R0L        ;
MOV.B    R0L,@SCR0        ; TE,RE set

RTS

```

```

;=====
; Application routine
;=====
_Application:
    BTST.B    #RDRF,@SSR0          ; Received ?
    BEQ      APPLIEXIT             ; No
;-----
;          Command check
;-----
JSR        @_SCISRecv             ; Command receive
CMP.B     #COM_CNTSTART,R0L      ; Count start command ?
BEQ      CSTART:8                ; Yes
CMP.B     #COM_CNTSTOP,R0L       ; Count stop command ?
BEQ      CSTOP:8                 ; Yes
CMP.B     #COM_CNTCYCLE,R0L      ; Count cycle change command ?
BEQ      CCYCLE:8                ; Yes
CMP.B     #COM_UPDSTART,R0L      ; F/W Update start command ?
BEQ      USTART:8                ; Yes
MOV.B     #RSP_NAK,R0L           ;
JSR        @_SCISend              ; NAK response send
BRA      APPLIEXIT:8             ;
;-----
;          Count start
;-----
CSTART:
    SUB.L    ER0,ER0              ;
    MOV.B   @bytleDCnt,R0L        ;
    MOV.B   @(LEDTable,ER0),R1L   ;
    MOV.B   R1L,@P4DR             ;
    MOV.B   #H'0B,R0L            ;
    MOV.B   R0L,@T8TCR1          ;
    MOV.B   #RSP_ACK,R0L         ;
    JSR     @_SCISend             ; ACK response send

    ANDC   #H'7F,CCR              ; Imask clear
    BRA   APPLIEXIT:8             ;
;-----
;          Count stop
;-----
CSTOP:
    ORC    #H'80,CCR              ; Imask set

    MOV.B   #H'08,R0L            ;
    MOV.B   R0L,@T8TCR1          ;
    MOV.B   #RSP_ACK,R0L         ;
    JSR     @_SCISend             ; ACK response send
    BRA   APPLIEXIT:8             ;

```

```

;-----
;          Count cycle change
;-----
CCYCLE:
STC.B     CCR,R6L           ; CCR store
ORC.B     #H'80,CCR        ; Imask set

JSR       @_SCIRecv       ;
MOV.B     R0L,R2H         ;
JSR       @_SCIRecv       ;
MOV.B     R0L,R2L         ;
MOV.W     R2,@wrdNewTargetCycle ;
MOV.B     #RSP_ACK,R0L    ;
JSR       @_SCISend      ; ACK response send

LDC.B     R6L,CCR         ; CCR set
BRA       APPLIEXIT:8    ;
;-----
;          F/W Update
;-----
USTART:
MOV.B     #H'01,R0L       ;
MOV.B     R0L,@flgFWUpdate ; F/W Update request set

APPLIEXIT:
RTS

;=====
;  Count cycle interrupt routine
;=====
_countCycleInt:
PUSH.L    ERO             ;

MOV.W     @wrd100msCounter,R0 ; 100ms counter inc
INC.W     #1,R0           ;
MOV.W     R0,@wrd100msCounter ;

MOV.W     @wrdTargetCycle,E0 ;
CMP.W     E0,R0           ; Target cycle over ?
BCS      CCIntExit:8     ; No

SUB.L     ERO,ERO        ;
MOV.B     @bytLEDCnt,R0L ;
;*****
INC.B     R0L             ; Remove the comment when you use Increment mode
;; DEC.B  R0L             ; Remove the comment when you use Decrement mode
;*****

AND.B     #H'0F,R0L      ;
MOV.B     R0L,@bytLEDCnt ;
MOV.B     @(LEDDTable,ERO),R0L ;
MOV.B     R0L,@P4DR      ;

```

```

MOV.W    @wrdNewTargetCycle,R0    ;
MOV.W    R0,@wrdTargetCycle      ;

SUB.W    R0,R0                    ;
MOV.W    R0,@wrd100msCounter     ;

CCIntExit:
BCLR.B   #T8CMFA,@T8TCSR0        ;
BCLR.B   #T8CMFA,@T8TCSR1        ;
POP.L    ERO                      ;
RTE

;=====
;  nouse interrupt routine
;=====
_nouse:
RTE

        .SECTION  STACK,DATA,ALIGN=2    ;
;=====
;  Flag define
;=====
                .RES.W    128    ;
STACKTOP:      .EQU      $      ;

        .SECTION  B,DATA,ALIGN=2    ;
;=====
;  Flag define
;=====
flgFWUpdate    .RES.B    1    ;
bytLEDCnt      .RES.B    1    ;
wrdNewTargetCycle .RES.W    1    ;
wrdTargetCycle .RES.W    1    ;
wrd100msCounter .RES.W    1    ;

        .END

```

```

; /*****
; /*
; /* H8/3069F User Program Mode Application Note
; /* Sample software
; /*
; /* Flash erase/program routine
; /*
; /*****
.CPU          300HA:24

.INCLUDE      "iodef.inc"
.INCLUDE      "common.inc"

.EXPORT _FWUpdate      ;
.EXPORT _SCIRcv        ;
.EXPORT _SCISend       ;

;-----
; User define
;-----
CLOCK      .EQU      2500          ; 25MHz
ERASEBLK   .EQU      H'FFFF        ; EraseBlock define MSB(EB15)<----->LSB(EB00)
WKAREA     .EQU      H'01          ; Flash control work RAM area
; H'00 -> H'FFEF20 - H'FFFF1F(4KB)
; H'01 -> H'FFDF20 - H'FFEF1F(4KB)
; H'02 -> H'FFCF20 - H'FFDF1F(4KB)
; H'03 -> H'FFBF20 - H'FFCF1F(4KB)

;-----
; Constants define
;-----
SRAMWORK   .EQU      RAMTOP+((H'03-WKAREA)*H'1000)
; Flash erase/program system work address
URAMWORK   .EQU      SRAMWORK+H'1000 ; Flash erase/program user work address

ERUNENTRY  .EQU      SRAMWORK+H'10 ; Flash erase entry address
PRUNENTRY  .EQU      SRAMWORK+H'10 ; Flash program entry address
INITENTRY  .EQU      SRAMWORK+H'20 ; Flash erase/program initialize entry
UPROGSUM   .EQU      URAMWORK+H'00 ; Flash program data sum(2bytes)
UPROGDATA  .EQU      URAMWORK+H'02 ; Flash program data buffer address(128bytes)
UPRCENTRY  .EQU      URAMWORK+H'82 ; Flash erase/program user procedure entry

COMPLETE   .EQU      H'00          ; COMPLETE(No error)
DLERROR    .EQU      H'01          ; Download error
INIERROR   .EQU      H'02          ; Initialize error
ERSERROR   .EQU      H'03          ; Erase error
PRGERROR   .EQU      H'04          ; Program error
    
```

```

;=====
; F/W Update routine
;=====
_FWUpdate:
    ORC    #H'80,CCR          ; All interrupt disable

    MOV.L  #STARTOF FEPProc,ER5 ; Source address
    MOV.L  #UPRCENTRY,ER6      ; Destination address
    MOV.W  #SIZEOF FEPProc,R4  ; copy size
    EEPMOV.W

    MOV.L  #UPRCENTRY,ER6      ;
    JMP    @ER6                ; Jump to program on RAM

    .SECTION    FEPProc,CODE,ALIGN=2;
;=====
; Flash Erase/Program procedure routine
;=====
_FEPProc:
;-----
; FWE check
;-----
    MOV.B  @FCCS,R6L          ; FWE@FCCS bit check
    BTST.B #7,R6L             ;
    BEQ    FEPErrors:8        ;
;-----
; Erase
;-----
    BSR    _EraseProc          ; Erase
    MOV.B  R0L,R0L            ; Erase error?
    BNE    FEPErrors:8        ; Yes
;-----
; Program
;-----
    BSR    _ProgramProc        ; Program
    MOV.B  R0L,R0L            ; Program error?
    BNE    FEPErrors:8        ; Yes
;-----
; Reset
;-----
WaitSendend:
    BTST.B #TEND,@SSR0        ;
    BEQ    WaitSendend:8      ;
    JMP    @@0                 ; Jump to reset vector
;-----
; Erase/Program error
;-----
FEPErrors:
    MOV.B  #RSP_NAK,R0L        ; NAK response send
    BSR    _SCISend            ;
ENDLESS:
    BRA    ENDLESS:8          ;
    
```

```

;=====
; Flash erase procedure routine
;=====
_EraseProc:
;-----
; Erase code download
;-----
MOV.B #H'01,R6L ; EPVB@FECS bit set
MOV.B R6L,@FECS ;
MOV.B #WKAREA,R6L ; FTDAR set
MOV.B R6L,@FTDAR ;
MOV.B #H'FF,R6L ; DPFR(@FTDAR) clear
MOV.B R6L,@SRAMWORK ;

MOV.B #H'A5,R6L ; FKEY <- H'A5 set(SCO enable)
MOV.B R6L,@FKEY ;

MOV.L #FCCS,ER6 ; SCO@FCCS bit set
BSET.B #0,@ER6
NOP ; Download wait
NOP ;
NOP ;
NOP ;

MOV.B #H'00,R6L ; FKEY <- H'00 set(SCO disable)
MOV.B R6L,@FKEY ;

MOV.B @SRAMWORK,R0L ; DPFR(@FTDAR) check
BNE EDLDERR ; Not H'00 -> Download error!
;-----
; Erase initialize
;-----
MOV.L #CLOCK,ER0 ; Clock setting
MOV.L #INITENTRY,ER6 ; Initialize entry address
JSR @ER6 ; Initialize!!
NOP ; wait
MOV.B R0L,R0L ; R0L(FPFR) check
BNE EINIERR ; Not H'00 -> Initialize error!
;-----
; Erase
;-----
MOV.B #H'5A,R6L ; FKEY <- H'5A set(erase enable)
MOV.B R6L,@FKEY ;

MOV.B #0,R5L ;
MOV.W #ERASEBLK,E5 ;
SUB.L ER0,ER0 ; FEBS all clear
EP00:
SHLR.W E5 ;
BCC EPNEXT:8 ; Next block
MOV.B R5L,R0L ; R0L(FEBS) set
MOV.L #ERUNENTRY,ER6 ; Run entry address
JSR @ER6 ; Erase!!
NOP ; wait

```

```

MOV.B  R0L,R0L          ; R0L(FPFR) check
BNE    ERASERR:8       ; Not H'00 -> Erase error!
EPNEXT:
MOV.W  E5,E5          ;
BEQ    EPEND:8        ; No erase block
INC.B  R5L            ;
BRA    EP00:8         ;
EPEND:
MOV.B  #H'00,R6L      ; FKEY <- H'00 set(erase disable)
MOV.B  R6L,@FKEY      ;
;-----
;      Erase procedure complete
;-----
MOV.B  #RSP_ACK,R0L   ; ACK response send
BSR    _SCISend       ;
;
MOV.B  #COMPLETE,R0L ; COMPLETE(No error)
RTS    ;
;-----
;      Erase procedure error
;-----
EDLDERR:
MOV.B  #DLDError,R0L  ; Download error
RTS    ;
EINIERR:
MOV.B  #INIERROR,R0L  ; Initialize error
RTS    ;
ERASER
MOV.B  #ERSERROR,R0L  ; Erase error
RTS    ;
    
```

```

;=====
; Flash program procedure routine
;=====
_ProgramProc:
;-----
; Program code download
;-----
MOV.B #H'01,R6L ; PPVS@FPCS bit set
MOV.B R6L,@FPCS ;
MOV.B #WKAREA,R6L ; FTDAR set
MOV.B R6L,@FTDAR ;
MOV.B #H'FF,R6L ; DPFR(@FTDAR) clear
MOV.B R6L,@SRAMWORK ;

MOV.B #H'A5,R6L ; FKEY <- H'A5 set(SCO enable)
MOV.B R6L,@FKEY ;

MOV.L #FCCS,ER6 ; SCO@FCCS bit set
BSET.B #0,@ER6 ;
NOP ; Download wait
NOP ;
NOP ;
NOP ;

MOV.B #H'00,R6L ; FKEY <- H'00 set(SCO disable)
MOV.B R6L,@FKEY ;

MOV.B @SRAMWORK,R0L ; DPFR(@FTDAR) check
BNE PDLDERR ; Not H'00 -> Download error!
;-----
; Program initialize
;-----
MOV.L #CLOCK,ER0 ; Clock setting
MOV.L #INITENTRY,ER6 ; Initialize entry address
JSR @ER6 ; Initialize!!
NOP ; wait
MOV.B R0L,R0L ; R0L(FPFR) check
BNE PINIERR ; Not H'00 -> Initialize error!
;-----
; Program
;-----
MOV.B #H'5A,R6L ; FKEY <- H'5A set(write enable)
MOV.B R6L,@FKEY ;

MOV.L #FLASHTOP,ER5 ;
WP00:
BSR FWDDataRecv:8 ; F/W data receive
CMP.B #COM_UPDDATA,R0L ; Update data command?
BNE WP01:8 ; No
CMP.L #FLASHEND,ER5 ;
BCC PROGRAMERR:8 ;
    
```

```

MOV.L  #UPROGDATA,ER0          ; ER0(FMPDR) set
MOV.L  ER5,ER1                 ; ER1(FMPAR) set
MOV.L  #PRUNENTRY,ER6         ; Run entry address
JSR    @ER6                    ; Write!!
NOP                                     ; wait
MOV.B  R0L,R0L                 ; R0L(FPFR) check
BNE    PROGRAMERR:8           ; Not H'00 -> Write error!

MOV.B  #RSP_ACK,R0L           ; ACK response send
BSR    _SCISend                ;

ADD.L  #128,ER5                ;
BRA    WP00:8                  ;
WP01:
CMP.B  #COM_UPDEND,R0L        ; Update end command?
BNE    PROGRAMERR:8           ; No
;-----
;      SUM calculation
;-----
SUB.W  R0,R0                    ; SUM clear
MOV.L  #FLASHTOP,ER5          ;
WPSUM:
MOV.B  @ER5,R1L                ; FLASH data read
ADD.B  R1L,R0L                 ; SUM calculate
ADDX.B #0,R0H                  ;
INC.L  #1,ER5                  ;
CMP.L  #FLASHEND,ER5          ; Calculation is end ?
BLS    WPSUM:8                 ; No

MOV.W  @UPROGSUM,R1            ;
CMP.W  R1,R0                    ; SUM compare
BNE    PROGRAMERR:8           ; Not match

WPEND:
MOV.B  #H'00,R6L               ; FKEY <- H'00 set(write disable)
MOV.B  R6L,@FKEY                ;
;-----
;      Program procedure complete
;-----
MOV.B  #RSP_ACK,R0L           ; ACK response send
BSR    _SCISend                ;

MOV.B  #COMPLETE,R0L          ; COMPLETE(No error)
RTS                                     ;
    
```

```

;-----
;      Program procedure error
;-----
PDLDERR:
    MOV.B  #DLDERROR,R0L          ; Download error
    RTS                                     ;
PINIERR:
    MOV.B  #INIERROR,R0L         ; Initialize error
    RTS                                     ;
PROGRAMERR:
    MOV.B  #PRGERROR,R0L        ; Program error
    RTS                                     ;

;=====
;  F/W data receive routine
;=====
FWDDataRecv:
    PUSH.L ER6                    ;

    BSR    _SCIREcv:8             ; lbyte receive
    CMP.B  #COM_UPDDATA,R0L      ; Update data command?
    BEQ    FWDtrcv_128:8         ; Yes
    CMP.B  #COM_UPDEND,R0L      ; Update end command?
    BEQ    FWDtrcv_sum:8         ; Yes
    BRA    FWDtrcv_exit:8        ; No

FWDtrcv_128:
    MOV.B  R0L,R1H               ;
    MOV.B  #128,R1L              ;
    MOV.L  #UPROGDATA,ER6        ;

FWDtrcv_loop:
    BSR    _SCIREcv:8             ; lbyte receive
    MOV.B  R0L,@ER6              ; data store
    INC.L  #1,ER6                ;
    DEC.B  R1L                   ;
    BNE    FWDtrcv_loop:8        ;
    MOV.B  R1H,R0L              ;
    BRA    FWDtrcv_exit:8        ;

;-----
;      SUM receive
;-----

FWDtrcv_sum:
    MOV.B  R0L,R1H               ;
    BSR    _SCIREcv:8             ; lbyte receive
    MOV.B  R0L,R6H               ; sum(High) store
    BSR    _SCIREcv:8             ; lbyte receive
    MOV.B  R0L,R6L               ; sum(Low) store
    MOV.W  R6,@UPROGSUM          ; sum store
    MOV.B  R1H,R0L               ;

FWDtrcv_exit:
    POP.L  ER6                   ;
    RTS                                     ;

```

```

;=====
; 1 Byte receive routine
;=====
_SCIRecv:
    MOV.B  @SSR0,R0L          ;
    MOV.B  R0L,R0H           ;
    AND.B  #H'38,R0L         ;
    BEQ    SCIRcvExec:8      ;
    AND.B  #H'C7,R0H         ;
    MOV.B  R0H,@SSR0         ;
SCIRcvExec:
    BTST.B #RDRF,@SSR0       ;
    BEQ    _SCIRecv:8        ;
    MOV.B  @RDR0,R0L         ;
    BCLR.B #RDRF,@SSR0       ;
    RTS

;=====
; 1 Byte send routine
;=====
_SCISEND:
    BTST.B #TDRE,@SSR0       ;
    BEQ    _SCISend:8        ;
    MOV.B  R0L,@TDR0         ;
    BCLR.B #TDRE,@SSR0       ;
    RTS

.END

```

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。