

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。

# アプリケーション・ノート

## V850ES/Jx3

### サンプル・プログラム（低電圧検出回路（LVI））

### 低電圧検出時リセット発生編

この資料は、サンプル・プログラムの動作概要や使用方法、および低電圧検出回路（LVI）の設定方法や活用方法を説明したものです。サンプル・プログラムでは、 $V_{DD} < V_{LVI}$  ( $V_{LVI} = 2.95 \text{ V (TYP.)} \pm 0.10 \text{ V}$ ) を検出して、内部リセット（LVIリセット）信号を発生します。LVIリセットでは、LVIリセット直前のLED点灯パターンが保持され、LVIリセット解除後に、LVIリセット直前のLED点灯パターンが復元されます。

#### 対象デバイス

V850ES/JG3 マイクロコントローラ

V850ES/JJ3 マイクロコントローラ

#### 目次

第1章 概要	3
1.1 初期化	4
1.2 低電圧検出回路（LVI）動作	5
1.3 メイン処理	5
1.4 割り込み処理	6
第2章 回路図	7
2.1 回路図	7
2.2 周辺ハードウェア	7
第3章 ソフトウェアについて	8
3.1 ファイル構成	8
3.2 使用する内蔵周辺機能	9
3.3 初期設定と動作概要	9
3.4 フロー・チャート	11
3.5 V850ES/JJ3版とV850ES/JG3版の違い	14
3.6 ROM化について	14
3.7 セキュリティIDについて	16
3.8 MINICUBE2使用時のサンプル・プログラム動作上の注意	16
第4章 レジスタ設定について	17
4.1 低電圧検出回路（LVI）の設定	18
4.2 LVIリセット検出の確認	22
第5章 関連資料	24
付録A プログラム・リスト	25

資料番号 U20019JJ1V0AN00

発行年月 September 2009 NS

- 本資料に記載されている内容は2009年09月現在のもので、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品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、一切その責を負いません。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。
- 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、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。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。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害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を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「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。

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

特別水準：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

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、生命維持のための装置またはシステム等

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。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNEC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う。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

(M8E0909J)

# 第1章 概 要

このサンプル・プログラムでは、低電圧検出回路（LVI）の使用例を示しています。

低電圧検出回路を使用し、 $V_{DD} < V_{LVI}$  ( $V_{LVI} = 2.95 \text{ V (TYP.)} \pm 0.10 \text{ V}$ )を検出して、内部リセット（LVIリセット）信号を発生するように設定します。

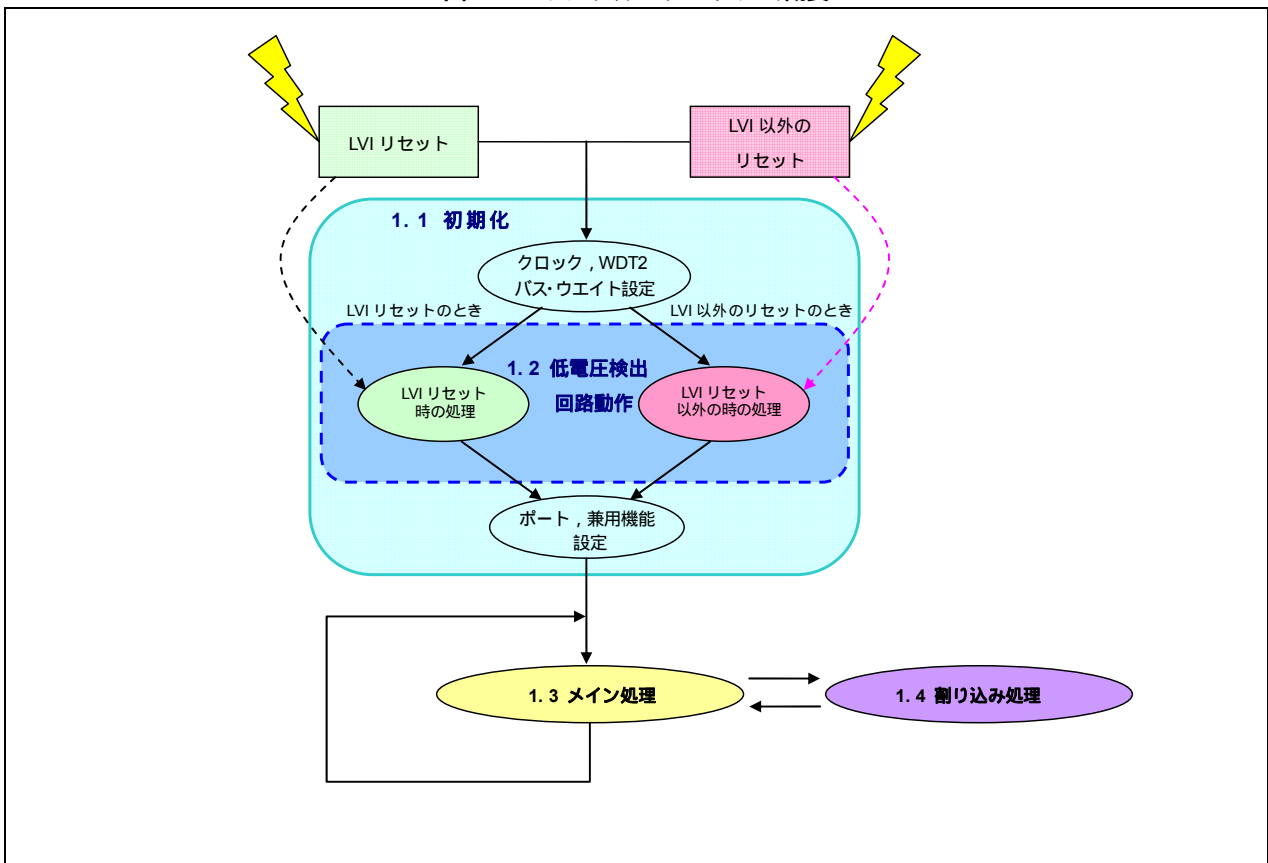
初期設定完了後は、スイッチ入力の立ち下がりエッジを検出して割り込み処理を行い、スイッチ入力回数に応じたLED点灯パターンを表示します（この処理は、V850ES/Jx3 サンプル・プログラム（割り込み）スイッチ入力による外部割り込み編 アプリケーション・ノートと同様です）。

LVI以外によるリセットが発生した場合には、内蔵RAMに保持しているLED点灯パターン・データを初期化します。LVIリセットが発生した場合には、内蔵RAMはLVIリセット直前のデータを保持し、LVIリセット解除後に、保持していたLED点灯パターンを復元して表示するプログラムです。

また低電圧検出機能を使用してRAMに保持されたデータを使用する場合、厳密な処理を行うにはRAMのデータが正しく保持されているかを確認する必要があります。このサンプル・プログラムではRAM保持電圧検出機能を用いて内蔵RAMが値を保持しているかを確認しています。

ソフトウェアの主な内容は、次のとおりです。

図1 - 1 サンプル・プログラム概要



## 1.1 初期化

### <内蔵周辺設定>

- ・内蔵周辺I/Oレジスタへのバス・アクセスのウェイト<ウェイト：2>設定
- ・オンチップ・デバッグ・モード・レジスタを通常動作モードに設定
- ・内蔵発振器を停止，ウォッチドッグ・タイマ2停止
- ・CPUクロックを分周なしに設定
- ・PLLモードに設定し 4 MHz x 8通倍 = 32 MHz 動作に設定
- ・VLVI（低電圧検出電圧）を 2.95 V（TYP.）±0.10 Vに設定，低電圧検出時動作<リセット>の設定<sup>注</sup>

注 LVIリセットにより，システムが再起動した場合には設定を行いません。

### <端子設定>

- ・未使用端子の設定
- ・外部割り込み端子の設定（エッジ指定，優先度指定，マスク解除）
- ・LED出力端子の設定（LVI以外のリセット時はLED1，LED2ともに消灯値，LVIリセット時はリセット直前の出力値を指定）

### <ROM化>

- ・ROM化処理（初期値あり変数の初期化）

## 1.2 低電圧検出回路 (LVI) 動作

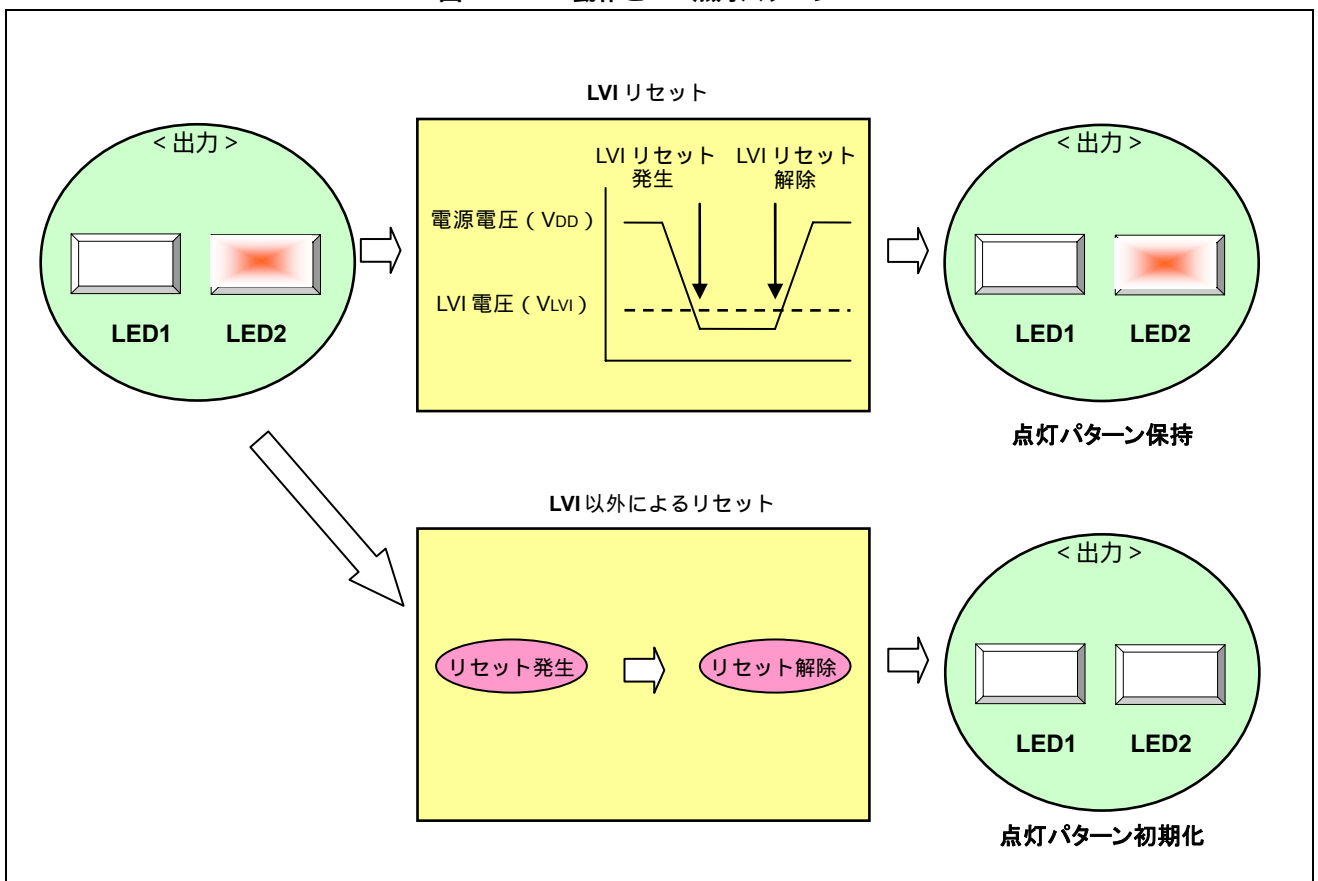
低電圧検出回路 (LVI) は、 $V_{DD} < V_{LVI}$  になった場合の動作として、内部リセット (LVIリセット) 発生と内部割り込み信号 (INTLVI) 発生のどちらかを設定できます (内部割り込み信号 (INTLVI) 選択時は  $V_{DD} < V_{LVI}$  になった後、 $V_{DD} > V_{LVI}$  となった際も割り込み信号が発生します)。

このサンプル・プログラムでは、低電圧検出時 ( $V_{DD} < V_{LVI}$ ) の動作として内部リセット (LVIリセット) を設定します。LVIリセット発生時、レジスタ値は初期化されますが、内蔵RAMはLVIリセット直前のデータを保持します<sup>注</sup>。したがって、LVIリセット時は、LVIリセット直前のLED点灯パターンを保持し、LVIリセット解除後にLED点灯パターンを復元することが可能となります。

なお、LVI以外のリセット時は、プログラムにてLED点灯パターンを初期化し、全消灯となります。

**注** 内蔵RAMの値が正常に保持されているかどうかを内蔵RAMデータ・ステータス・レジスタ (RAMS) で確認する必要があります。

図1-2 LVI動作とLED点灯パターン



## 1.3 メイン処理

- ・EI命令による割り込み許可
- ・何もしない無限ループを実行 (スイッチ入力による割り込み待ち)

## 1.4 割り込み処理

スイッチ入力によるINTP0端子の立ち下りエッジを検出し、割り込み処理を起動します。

割り込み処理では、INTP0端子の立ち下りエッジを検出してから約10 ms経過後に、スイッチがONであることを確認し、LEDの点灯パターンを変化させます。

なお、INTP0端子の立ち下りエッジを検出後、約10 ms経過後にスイッチがOFFである場合は、チャタリング・ノイズであると判定し、LEDの点灯パターンを変化させません。

図1-3 スイッチ入力とLED点灯パターン

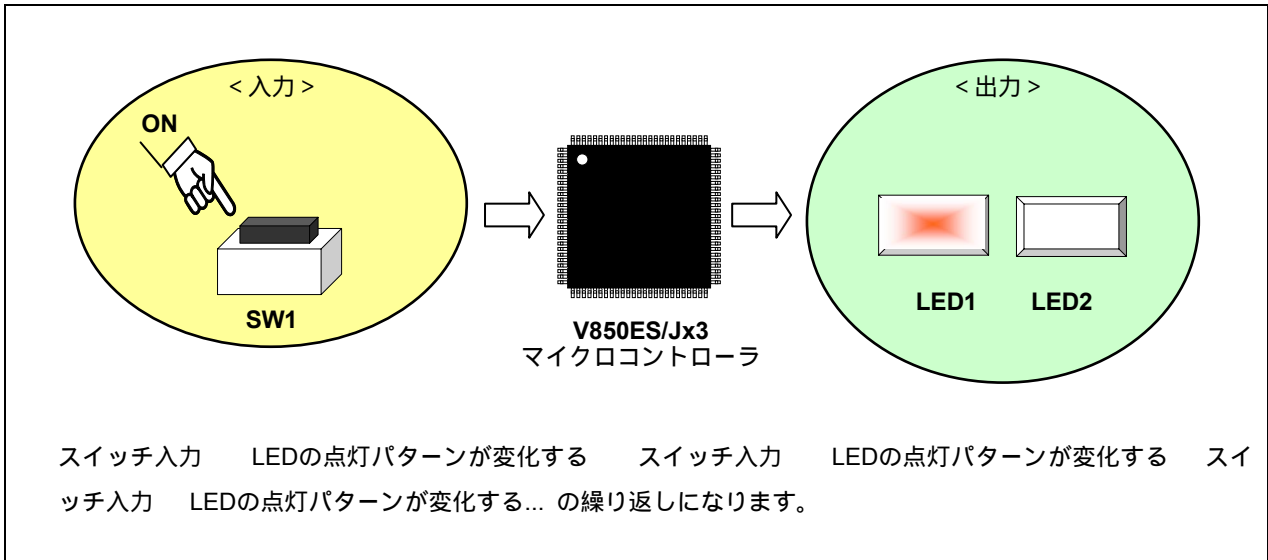


表1-1 LED点灯パターン

スイッチ (SW1) の入力回数 <sup>注</sup>	LED1	LED2
0回	OFF	OFF
1回	ON	OFF
2回	OFF	ON
3回	ON	ON

注 4回目以降の入力は0回からの繰り返しになります。

**備考** デバイス使用上の注意事項については、各製品のユーザーズ・マニュアル (V850ES/Jx3) を参照してください。



【コラム】 チャタリングとは

スイッチが切り替わった直後に、接点が機械的にばたつくことにより、電気信号がONとOFFを繰り返す現象のことです。



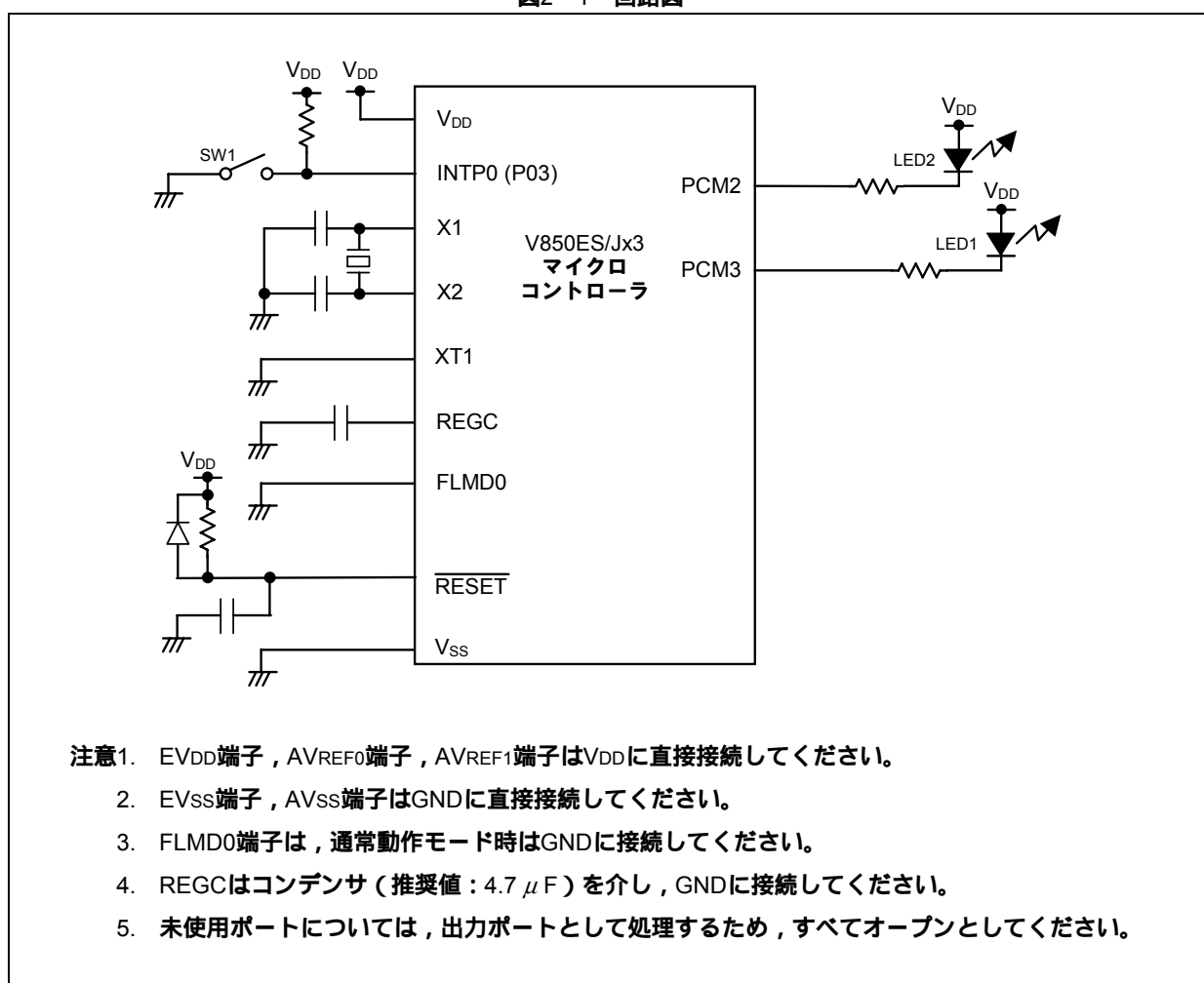
## 第2章 回路図

この章では、このサンプル・プログラムで使用する場合の回路図および周辺ハードウェアを説明します。

### 2.1 回路図

回路図を次に示します。

図2 - 1 回路図



### 2.2 周辺ハードウェア

使用する周辺ハードウェアを次に示します。

#### (1) スイッチ (SW1)

LED点灯制御用の割り込み入力として、スイッチを使用します。

#### (2) LED (LED1, LED2)



スイッチ入力に対応した出力として、LEDを使用します。

## 第3章 ソフトウェアについて


この章では、ダウンロードする圧縮ファイルのファイル構成、使用するマイコンの内蔵周辺機能、サンプル・プログラムの初期設定と動作概要、およびフロー・チャートを説明します。


### 3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名 (ツリー構造)	説明	同封圧縮 (*.zip) ファイル	
			
<ul style="list-style-type: none"> <li>・ conf               <ul style="list-style-type: none"> <li>— crtE.s</li> <li>— AppNote_LVI.dir</li> <li>— AppNote_LVI.prj</li> <li>— AppNote_LVI.prw</li> </ul> </li> <li>・ src               <ul style="list-style-type: none"> <li>— main.c</li> <li>— minicube2.s</li> </ul> </li> </ul>	スタートアップ・ルーチン・ファイル <sup>注1</sup>	-	
	リンク・ディレクティブ・ファイル <sup>注2</sup>		
	統合開発環境 PM+用プロジェクト・ファイル	-	
	統合開発環境 PM+用ワーク・スペース・ファイル	-	
	マイコンのハードウェア初期化処理とメイン処理を記述したC言語ソース・ファイル		
	MINICUBE2用の領域予約を行うソース・ファイル		

- 注1. ワークスペース新規作成時の「スタートアップ・ファイルの指定」時に、「サンプルをコピーして使用する(C)」を選択した際にコピーされるスタートアップ・ファイル(デフォルト・インストール・パスであれば、C:\Program Files\NEC Electronics Tools\CA850\使用バージョン\lib850\r32\crtE.sのコピーとなります)。
2. ワークスペース新規作成時の「リンク・ディレクティブ・ファイルの指定」時に、「サンプルを作成して使用する(C)」を選択し、「メモリの使用方法：内蔵メモリのみ(C)」をチェックした際に、自動生成されるリンク・ディレクティブ・ファイルに、**MINICUBE2用のセグメントを追加**したもの(デフォルト・インストール・パスであれば、C:\Program Files\NEC Electronics Tools\PM+\使用バージョン\bin\w\_data\V850\_i.dat が基準となります)。

備考  : ソース・ファイルのみ同封

 : 統合開発環境 PM+で使用するファイルを同封

## 3.2 使用する内蔵周辺機能

このサンプル・プログラムでは、マイコンに内蔵する次の周辺機能を使用します。

- ・低電圧検出回路 ( $V_{DD} < V_{LVI}$ 検出用) : LVI
- ・外部割り込み入力 (スイッチ入力用) : INTP0 (SW1)
- ・出力ポート (LED点灯用) : PCM2 (LED2), PCM3 (LED1)

## 3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定にて、クロック周波数の選択や、ウォッチドッグ・タイマ2の停止設定、入出力ポートや外部割り込みの端子設定、割り込みの設定、LVIの設定などを行います。

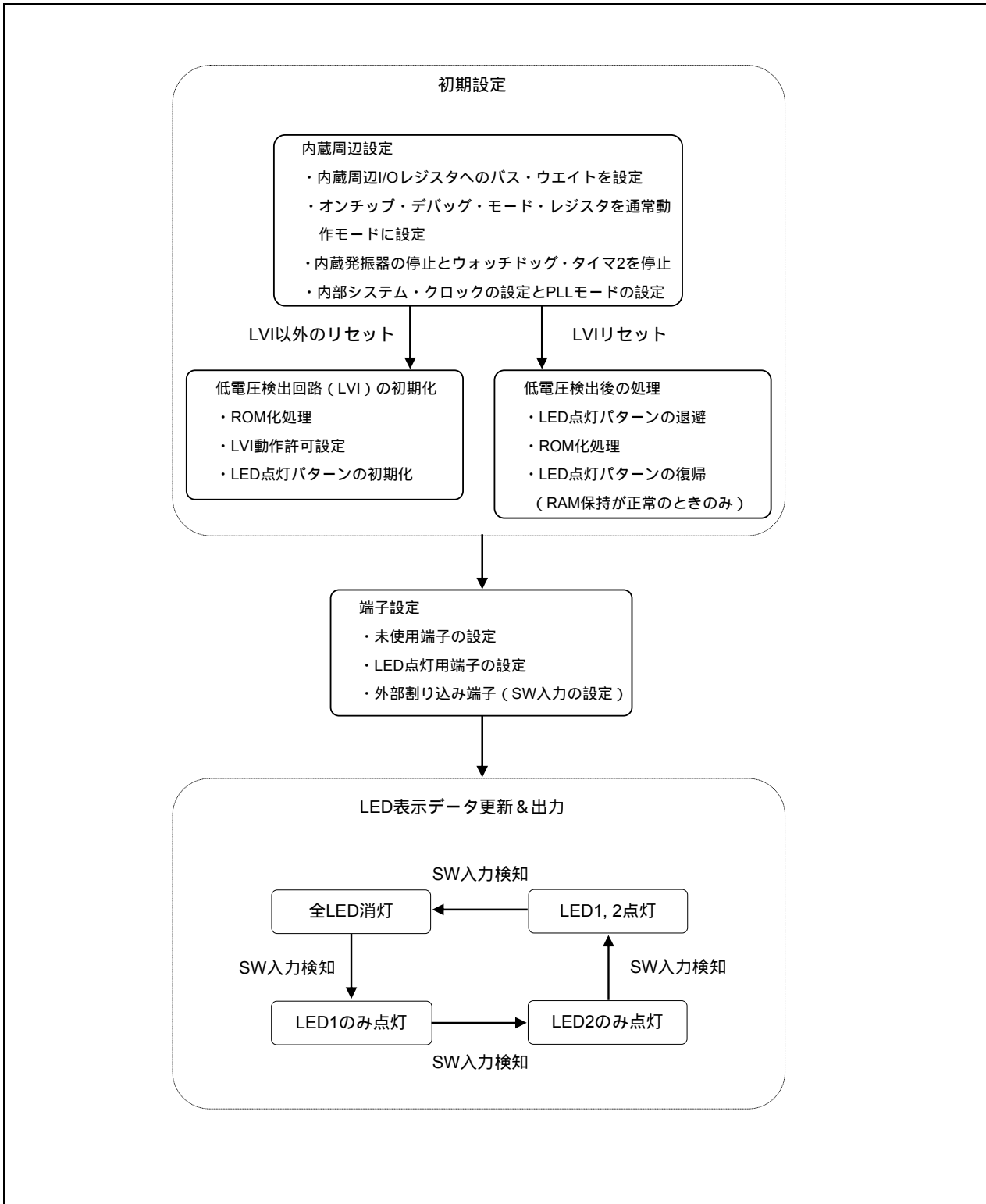
初期設定完了後は、スイッチ入力 (SW1) の立ち下りエッジを検出して割り込み処理を行い、スイッチの入力回数に応じて、2つのLED (LED1, LED2) の点灯パターンを制御します (この処理は、「V850ES/Jx3 サンプル・プログラム (割り込み) スwitch入力による外部割り込み編」アプリケーション・ノートと同様です)。

リセットがかかるとマイクロコンピュータの内蔵周辺機能は初期化されますが、内蔵RAMは初期化されずに値を保持します<sup>注</sup>。しかし、リセット解除後に起動するスタートアップ・ルーチンでは初期値なしの変数はクリアされてしまいます。LVIによるリセットの場合に、保持しているLEDの点灯パターンが初期化されないようにするためにはスタートアップ・ルーチンでクリアされない初期値あり変数を使用する必要があります。具体的には、初期値あり変数を使用したLEDの点灯パターンの値を初期値なし変数に一旦退避して、初期値あり変数の初期化 (ROM化) を行い、初期化完了後に元に戻します (ROM化の詳細に関しては3.6 ROM化についてを参照してください)。

なお、LVI以外によるリセットが発生した場合には、プログラムにてLED点灯パターンを初期化 (全消灯) します。次の状態遷移図に初期設定と動作の概要について示します。

**注** 内蔵RAMの値が正常に保持されているかどうかは内蔵RAMデータ・ステータス・レジスタ (RAMS) で確認する必要があります。

図3 - 1 状態遷移図



### 3.4 フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。

図3-2 フロー・チャート (1/2)

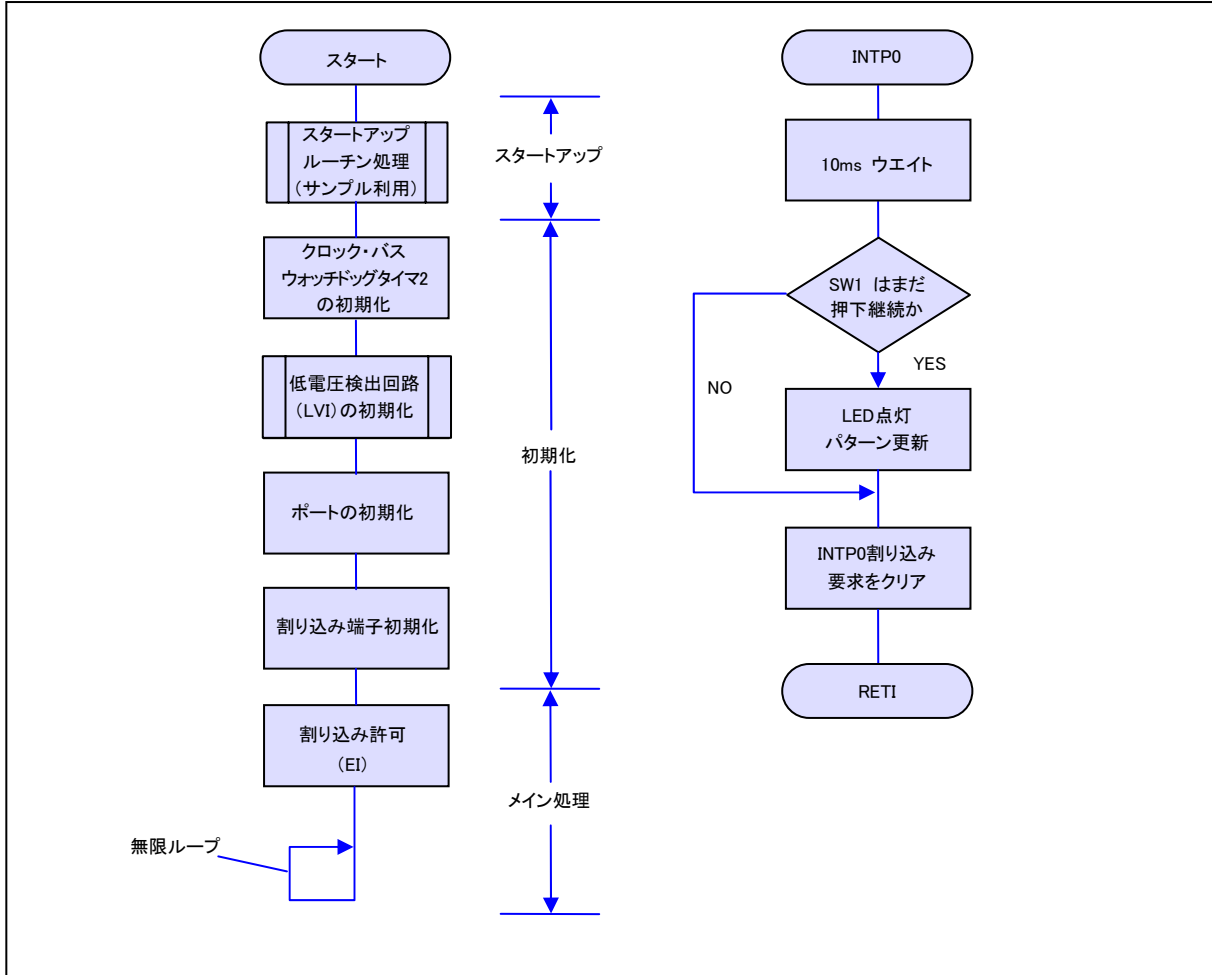
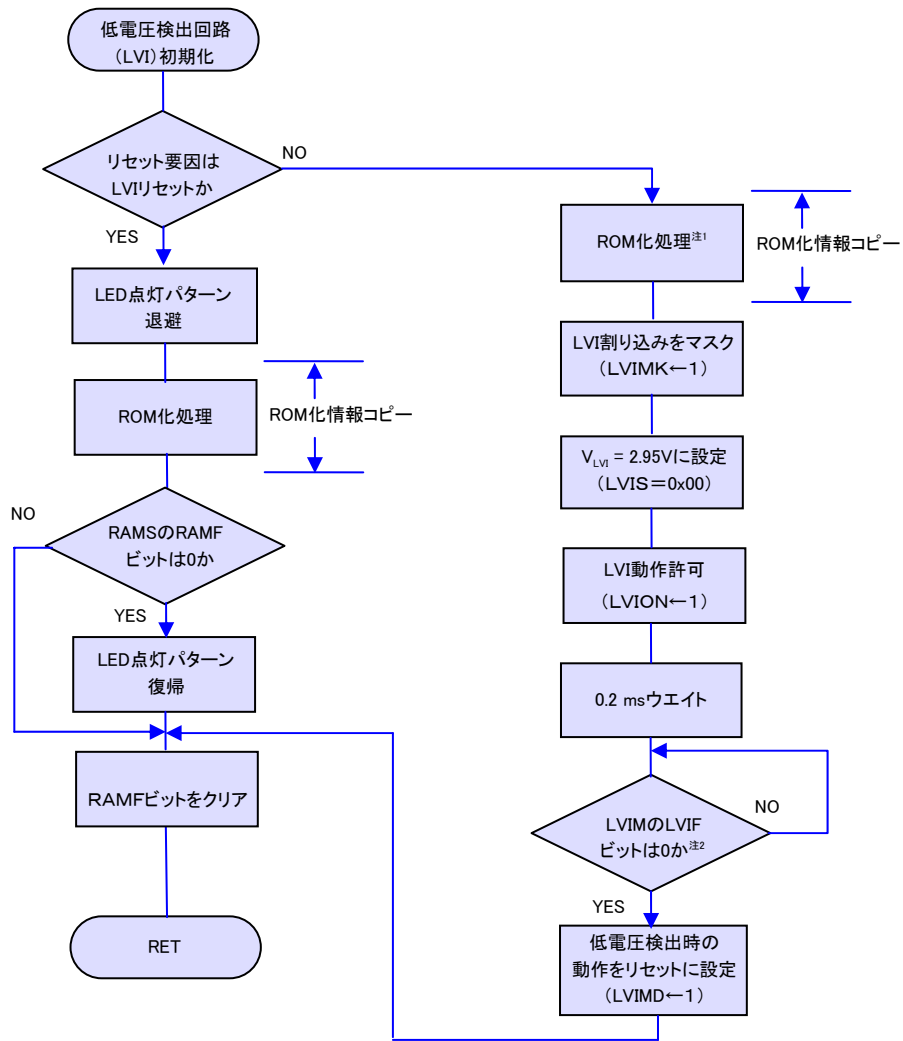


図3-3 フロー・チャート (2/2)



注1. リセット要因がLVIリセット以外の時は，ROM化処理（ROM化情報のコピー）を行うことにより，LED点灯パターンの指定が初期化されます。

2.  $V_{DD} > V_{LVI}$ になることにより，LVIMのLVIFビットがクリアされます。

**【コラム】スタートアップ・ルーチン処理の内容**

スタートアップ・ルーチンは、V850のリセットが解除されたあと、メイン関数を実行する前に、実行されるルーチンです。基本的にはC言語で記述されたプログラムが動作するための初期化処理を行います。具体的には、以下のことを行います。

- ・ main関数の引数領域の確保
- ・ スタック領域の確保
- ・ リセットが入った時のRESETハンドラの設定
- ・ テキスト・ポインタ (tp) の設定
- ・ グローバル・ポインタ (gp) の設定
- ・ スタック・ポインタ (sp) の設定
- ・ エlement・ポインタ (ep) の設定
- ・ マスク・レジスタ (r20, r21) へマスク値を設定
- ・ sbss領域, bss領域 のゼロ・クリア
- ・ 関数のプロローグ・エピローグ・ランタイム・ライブラリ用のCTBP値の設定
- ・ r6とr7をmain関数の引数に設定
- ・ main関数へ分岐する

### 3.5 V850ES/JJ3版とV850ES/JG3版の違い

V850ES/JJ3は、V850ES/JG3に対して、I/O、タイマ/カウンタ、シリアル・インターフェースなどの機能を拡張したものです。

このサンプル・プログラムにおいては、I/Oの初期化におけるポートの初期化範囲が異なります。

サンプル・プログラムの詳細については、付録A プログラム・リストをご覧ください。

### 3.6 ROM化について

このサンプル・プログラムでは、LVI初期化の中で「ROM化」情報のコピー処理を行っています。「ROM化」情報とは、初期値のある変数（初期値のある変数が配置されるセクション）の初期値情報のことで、「ROM化」情報をRAMにコピーすることにより、初期値のある変数（初期値のある変数が配置されるセクション）は、初めて初期値を持つこととなります<sup>注</sup>。

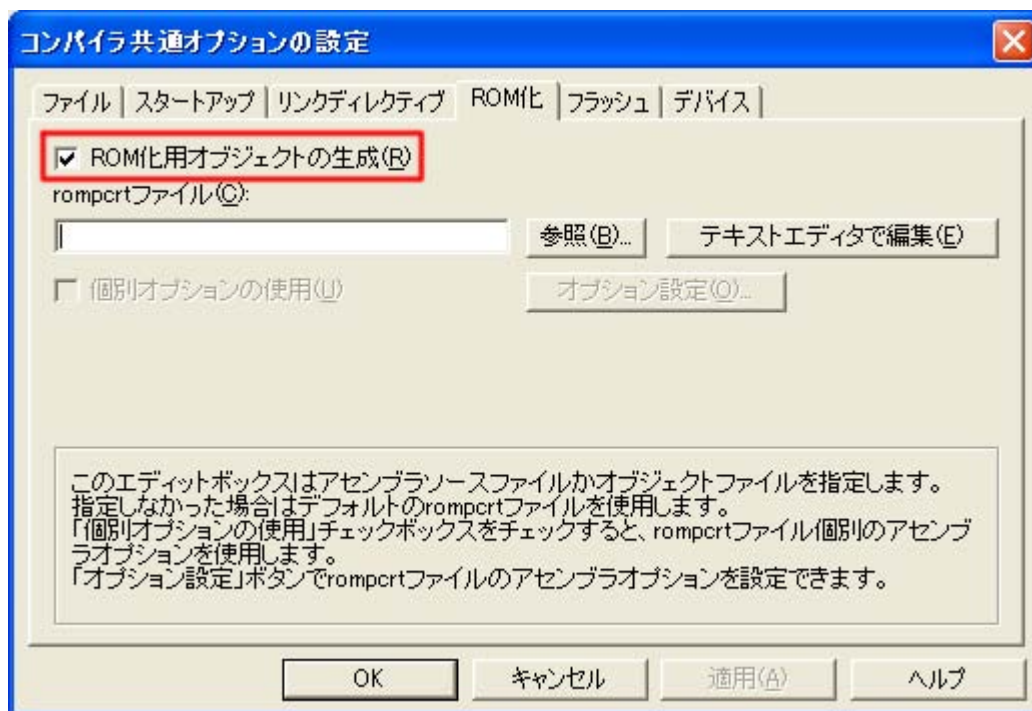
作成するプログラムで、初期値あり変数を使用している場合は、必ず「ROM化」情報の生成とコピー処理を行う必要があり、コピー処理の実施は、必ずその初期値あり変数を利用する前に行う必要があります。

注 ROM化により、パッキングされる対象となるものは、デフォルトでは「書き込み可能な属性を持つセクションに割り当てられたデータ」です。その他のデータをパッキングすることも可能です。詳細はCA850のヘルプを参照してください。

次に、ROM化を行うための手順を説明します。

まず、PM+のコンパイラ共通オプションの〔ROM化〕のタブを選択し、〔ROM化用オブジェクトの生成(R)〕をチェックします。

図3-4 コンパイラ共通オプションの設定

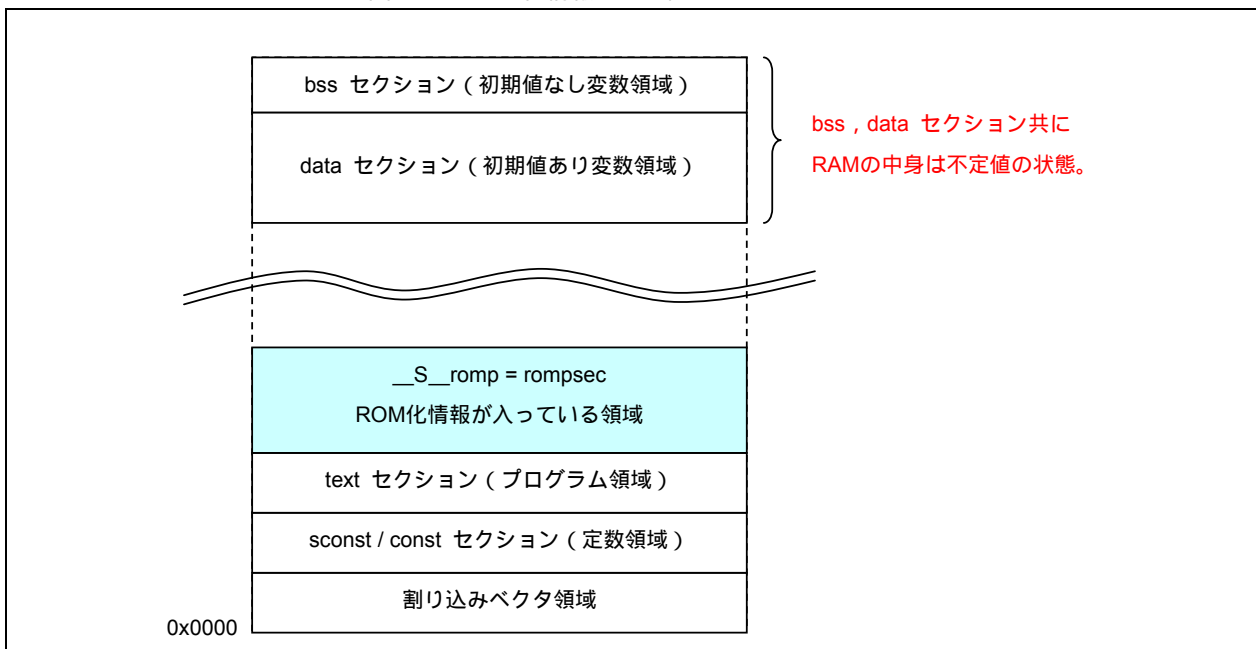




ROM化情報を格納するセクション (rompsec) は、自動でプログラム領域 (.text) セクションの直後に追加されますが、このチェック・ボックス操作によって、デフォルトの「rompct.o」で定義されたラベル“\_S\_romp”に、“rompsec”と同じアドレスを指すコードが生成され、コピー関数が格納されているライブラリ“lib.a”が自動的にリンクされます。

ここまでの手順で作成された、コピー前のメモリ・イメージは次のようになります。

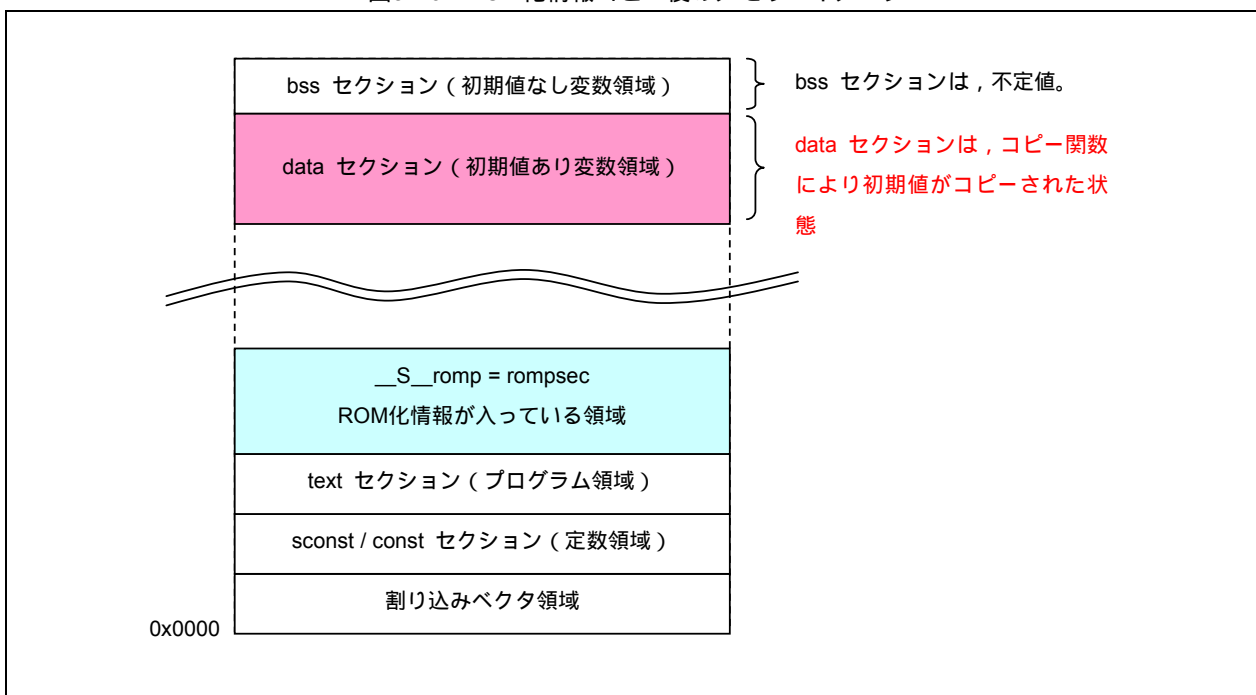
図3 - 5 ROM化情報コピー前のメモリ・イメージ



このままでは、初期値あり変数領域であるdataセクションの中身が不定のままなので、コピー処理を行う必要があります。

ROM化情報のコピーを行うために、\_rcopy()関数を呼び出すと、コピー後のメモリ・イメージは次のようになります。

図3 - 6 ROM化情報コピー後のメモリ・イメージ



### 3.7 セキュリティIDについて

オンチップ・デバッグ・エミュレータによるオンチップ・デバッグ時、フラッシュ・メモリの内容を第三者に読み出される事を防ぐために、10バイトのIDコードによる認証を行います。

セキュリティIDについての詳細説明は、V850ES/Jx3 **サンプル・プログラム（割り込み）スイッチ入力による外部割り込み編 アプリケーション・ノート**を参照してください。

### 3.8 MINICUBE2使用時のサンプル・プログラム動作上の注意

オンチップ・デバッグ・エミュレータにより、サンプル・プログラムを動作させた場合、LVIリセット発生時にブレークするため、注意が必要です。また、MINICUBE2の動作電圧が2.7 V未満の場合、MINICUBE2と対象デバイスの通信が正常に行われず、誤動作を起こす可能性があるため、注意が必要です。

MINICUBE2使用時の詳細説明は、QB-MINI2 **プログラミング機能付きオンチップ・デバッグ・エミュレータ ユーザーズ・マニュアル**、ID850QB Ver.3.40 **統合デバッガ 操作編 ユーザーズ・マニュアル**を参照してください。

## 第4章 レジスタ設定について

この章では、低電圧検出回路（LVI）機能の設定について説明します。

その他の初期設定については、V850ES/Jx3 サンプル・プログラム（初期設定） LED点灯のスイッチ制御編 アプリケーション・ノートを参照してください。また、割り込みについては、V850ES/Jx3 サンプル・プログラム（割り込み） スイッチ入力による外部割り込み編 アプリケーション・ノートを参照してください。

なお、リセット解除後に動作停止状態の周辺機能について、このサンプル・プログラムで使用しない周辺機能は、設定していません。

レジスタ設定方法の詳細については、各製品のユーザーズ・マニュアルを参照してください。

- ・ V850ES/JJ3 32ビット・シングルチップ・マイクロコントローラ ハードウェア編  
ユーザーズ・マニュアル
- ・ V850ES/JG3 32ビット・シングルチップ・マイクロコントローラ ハードウェア編  
ユーザーズ・マニュアル

C言語の拡張記述の詳細については、次のユーザーズ・マニュアルを参照してください。

CA850 Cコンパイラ・パッケージ C言語編 ユーザーズ・マニュアル

## 4.1 低電圧検出回路 (LVI) の設定

低電圧検出回路には、次の2種類の動作モードがあります (サンプル・プログラムではリセットとして使用)。

- ・リセットとして使用 [【例 1】](#)参照
- ・割り込みとして使用 [【例 2】](#)参照

低電圧検出回路は、主に次の3種類のレジスタで制御します。

- ・低電圧検出レジスタ (LVIM)
- ・低電圧検出レベル選択レジスタ (LVIS)
- ・内蔵RAMデータ・ステータス・レジスタ (RAMS)

### 4.1.1 低電圧検出動作の設定

低電圧検出レジスタ (LVIM) で、低電圧検出の動作モードの設定、および動作の制御を行います。

このサンプル・プログラムでは、低電圧検出時に内部リセット信号が発生するように設定します。

8/1ビット単位でリード/ライト可能です。ただし、LVIFビットはリードのみ可能です。

LVIMのリセット値は、低電圧検出によるリセットにより 0x82になり、その他の要因によるリセットでは、0x00になります。

図4 - 1 LVIMレジスタのフォーマット

低電圧検出レジスタ (LVIM)  
アドレス : 0xFFFFF890

	7	6	5	4	3	2	1	0
	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	低電圧検出動作の許可 / 禁止の制御
0	動作禁止
1	動作許可

LVIMD	低電圧検出の動作モード選択の制御
0	電源電圧値が検出電圧値を下回ったとき、または上回ったときに割り込み要求信号 INTLVIを発生
1	電源電圧 < 検出電圧時に内部リセット信号LVIRESを発生

LVIF	低電圧検出フラグの制御 (Read Only)
0	電源電圧 > 検出電圧、または動作禁止時
1	電源電圧 < 検出電圧

**備考1.** 表の赤字部分 (LVION, LVIMD) がサンプル・プログラムでの設定値となります。

**備考2.** 表の青字部分 (LVIF) がサンプル・プログラムでの確認するビットとなります。

### 4. 1. 2 低電圧検出レベルの設定

低電圧検出レベル選択レジスタ (LVIS) で、低電圧検出レベルの設定を行います。

V850ES/JJ3, V850ES/JG3の低電圧検出レベルは2.95 V (TYP.) ±0.10 Vで固定です。

このレジスタは8ビット単位でリード/ライト可能です。

また, LVIM.LVIONビット = 1かつLVIM.LVIMDビット = 1に設定したあと, 低電圧検出によるリセット以外のリセット要求が発生するまで, このレジスタへの書き込みは行えません。

図4 - 2 LVISレジスタのフォーマット

低電圧検出レベル選択レジスタ (LVIS)  
アドレス : 0xFFFFF891

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	LVIS0

LVIS0	低電圧検出レベルの制御
0	2.95 V (TYP.) ± 0.10 V
1	予約 (設定禁止)

**備考** 表の赤字部分がサンプル・プログラムでの設定値となります。

### 4. 1. 3 RAM保持電圧検出の制御

内蔵RAMデータ・ステータス・レジスタ (RAMS) は、内蔵RAMデータの有効/無効を示すフラグ・レジスタです。

8/1ビット単位でリード/ライト可能です。

RAMSレジスタは特定レジスタです。特定のシーケンスの組み合わせによってだけ書き込みができます。

次にRAMFビットのセット/クリア条件を示します。

- ・セット条件 : 検出レベル以下の電圧検出  
                  : 特定シーケンスによる1書き込み
- ・クリア条件 : 特定シーケンスによる0書き込み

図4 - 3 RAMSレジスタのフォーマット

内蔵RAMデータ・ステータス・レジスタ (RAMS)  
アドレス : 0xFFFFF892

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	RAMF

RAMF	内蔵RAM電圧検出
0	RAM保持電圧以下を非検出
1	RAM保持電圧以下を検出

**備考** 表の青字部分がサンプル・プログラムでの確認するビットとなります。

**【例 1】低電圧検出レベルを2.95 V (TYP.) ±0.10 Vに設定し、低電圧検出機能をリセットとして使用する場合  
(サンプル・プログラムと同内容)**

・設定手順

LVIの割り込みをマスクします。

LVIS.LVIS0ビットにて検出電圧を設定します。

LVIM.LVIONビット = 1に設定 (動作許可) します。

0.2 ms以上ソフトウェアにてウェイトを挿入します (プログラム例では、0.25 msのウェイトを設定)。

LVIM.LVIFビットで電源電圧 > 検出電圧であることを確認します。

LVIMDビット = 1 (内部リセット発生) を設定します。

**注意** LVIMDビット = 1に設定した場合、LVI以外のリセット要求が発生するまで、LVIM, LVISレジスタの変更はできません。

・プログラム例 (サンプル・プログラムと同内容)

```

LVIMK = 1;                /* LVIの割り込みをマスク          */ }
LVIS = 0x00;              /* 低電圧検出レベルを2.95±0.10 Vに設定*/ }
/* 低電圧検出動作許可 */
#pragma asm
    push r10
    mov 0x80, r10
    st.b r10, PRCMD
    set1 LVION
    pop r10
#pragma endasm
/* 250µs(0.25ms)のウェイト */
for( loop_wait = 0 ; loop_wait < LIMIT_250us_WAIT ; loop_wait++ )
{
    __nop();
}
while( LVIF == 1 );      /* 電源電圧が低電圧検出電圧よりも高くなるまで、*/ }
                        /* 無限ループとなる為、注意が必要です。      */ }
/* 低電圧検出時の動作：リセット動作に設定 */
#pragma asm
    push r10
    mov 0x82, r10
    st.b r10, PRCMD
    set1 LVIMD
    pop r10
#pragma endasm

```

【例 2】低電圧検出レベルを2.95 V (TYP.)  $\pm$  0.10 Vに設定し、低電圧検出機能を割り込みとして使用する場合

## ・設定手順

LVIの割り込みをマスクします。

LVIS.LVIS0ビットにて検出電圧を設定します。

LVIM.LVIONビット = 1に設定（動作許可）します。

0.2 ms以上ソフトウェアにてウェイトを挿入します(プログラム例では、0.25 msのウェイトを設定)。

LVIM.LVIFビットで電源電圧 > 検出電圧であることを確認します。

LVIの割り込み要求フラグをクリアします。

LVIの割り込みマスクを解除します。

## ・プログラム例

```

LVIMK = 1;                /* LVIの割り込みをマスク          */ }
LVIS = 0x00;              /* 低電圧検出レベルを2.95±0.10 Vに設定*/ }
/* 低電圧検出動作許可 */
#pragma asm
    push r10
    mov 0x80, r10
    st.b r10, PRCMD
    set1 LVION
    pop r10
#pragma endasm
/* 250µs(0.25ms)のウェイト */
for( loop_wait = 0 ; loop_wait < LIMIT_250us_WAIT ; loop_wait++ )
{
    __nop();
}
while( LVIF == 1 );      /* 電源電圧が低電圧検出電圧よりも高くなるまで、*/ }
                        /* 無限ループとなる為、注意が必要です。      */ }
LVIIIF = 0;             /* LVIの割り込み要求フラグをクリア  */ }
LVIMK = 0;              /* LVIの割り込みマスクを解除        */ }

```

## 4.2 LVIリセット検出の確認

LVIリセット時には、リセット要因フラグ・レジスタ (RESF) のLVIRFビットがセットされます。また、リセット端子入力によるリセットでは、LVIRFビットがクリアされます。そのため、リセット解除後にLVIRFビットを確認することにより、リセット要因がLVIリセットであったのか、リセット端子入力によるリセットであったのかを確認することができます。

### 4.2.1 リセット要因フラグ・レジスタ (RESF)

RESFレジスタは、どの要因から発生したリセット信号かを格納するレジスタです。

8/1ビット単位でリード/ライト可能です。

RESFレジスタは、特定のシーケンスの組み合わせによってだけ書き込みができます。

RESET端子入力により0x00になります。RESET端子以外のリセット要因である、ウォッチドッグ・タイマ2 (WDT2)、低電圧検出回路 (LVI)、クロック・モニタ (CLM) によるリセット時は、自身のリセット・フラグ (WDT2RF, CLMRF, LVIRFビット) をセットし、その他の要因フラグは保持します。

図4-4 RESFレジスタのフォーマット

リセット要因フラグ・レジスタ (RESF)  
アドレス : 0xFFFFF888

7	6	5	4	3	2	1	0
0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	WDT2からのリセット信号発生有無
0	発生なし
1	発生あり

CLMRF	CLMからのリセット信号発生有無
0	発生なし
1	発生あり

LVIRF	LVIからのリセット信号発生有無
<b>0</b>	発生なし
<b>1</b>	発生あり

**注意** 表の青字部分がサンプル・プログラムでの確認するビットとなります。

**備考1.** 各ビットへの書き込みは“0”ライトのみ可能で、“0”ライト書き込みとフラグ・セット (リセットの発生) が競合した場合、フラグ・セットが優先されます。

**2.** ウォッチドッグ・タイマ2 (WDT2)、低電圧検出回路 (LVI)、クロック・モニタ (CLM) を併用する場合は、リセット要因の確認後にリセット要因フラグのクリアが必要となるため、注意が必要です。



**【リセット要因フラグのクリアについて】**

上記の備考2のように、リセット要因の確認後にリセット要因フラグのクリアが必要となる場合がありますが、このサンプル・プログラムでは、低電圧検出回路（LVI）のみを使用しているため、リセット要因フラグのクリアは行っていません。

## 第5章 関連資料

資料名	資料番号
V850ES/JJ3 ユーザーズ・マニュアル ハードウェア編	U18376J
V850ES/JG3 ユーザーズ・マニュアル ハードウェア編	U18708J
V850ES アーキテクチャ編	U15943J
PM+ Ver.6.30 ユーザーズ・マニュアル	U18416J
CA850 Ver.3.20 Cコンパイラ・パッケージ 操作編	U18512J
CA850 Ver.3.20 Cコンパイラ・パッケージ C言語編	U18513J
CA850 Ver.3.20 Cコンパイラ・パッケージ リンク・ディレクティブ編	U18515J
QB-MINI2 プログラミング機能付きオンチップ・デバッグ・エミュレータ	U18371J
ID850QB Ver.3.40 統合デバッガ 操作編	U18604J

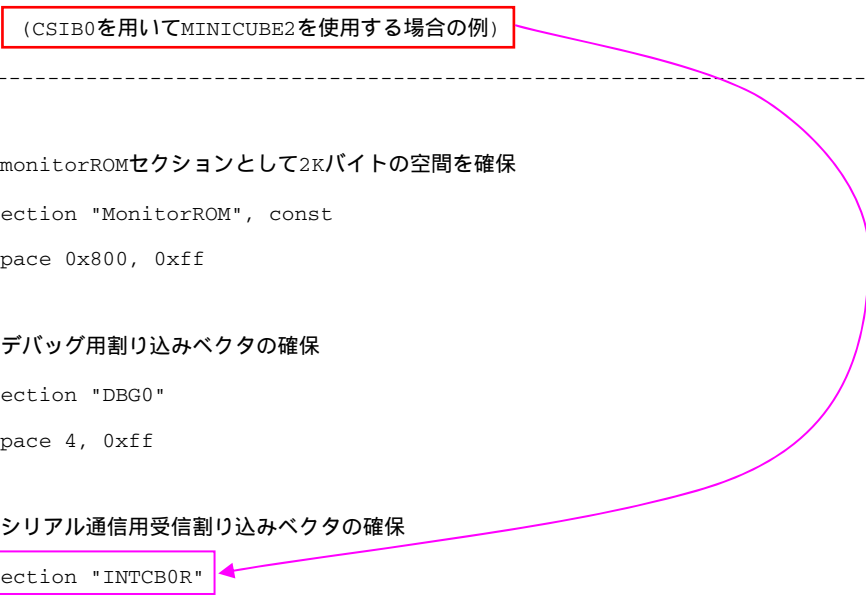
ドキュメント検索URL <http://www.necel.com/micro/ja/documentation.html>

# 付録A プログラム・リスト

プログラム・リスト例として、V850ES/JJ3マイクロコントローラのソース・プログラムを次に示します。

● minicube2.s

```
#-----  
#  
#   NEC Electronics   V850ES/Jx3 シリーズ  
#  
#-----  
#   V850ES/JJ3  JG3   サンプル・プログラム  
#-----  
#   低電圧検出時リセット発生編  
#-----  
# 【履歴】  
#   2009.9.--   新規作成  
#-----  
# 【概要】  
#   本サンプル・プログラムは、MINICUBE2使用時に必要なリソースの確保を行っている  
#   (CSIB0を用いてMINICUBE2を使用する場合の例)  
#-----  
  
--monitorROMセクションとして2Kバイトの空間を確保  
.section "MonitorROM", const  
.space 0x800, 0xff  
  
--デバッグ用割り込みベクタの確保  
.section "DBG0"  
.space 4, 0xff  
  
--シリアル通信受信割り込みベクタの確保  
.section "INTCB0R"  
.space 4, 0xff  
  
--MonitorRAMセクションとして16バイトの空間を確保  
.section "MonitorRAM", bss  
.lcomm monitorramsym, 16, 4
```



```

● AppNote_LVI.dir
# Sample link directive file (not use RTOS/use internal memory only)
#
# Copyright (C) NEC Electronics Corporation 2002
# All rights reserved by NEC Electronics Corporation.
#
# This is a sample file.
# NEC Electronics assumes no responsibility for any losses incurred by customers or
# third parties arising from the use of this file.
#
# Generated      : PM+ V6.31 [ 9 Jul 2007]
# Sample Version : E1.00b [12 Jun 2002]
# Device         : uPD70F3746 (c:¥program files¥nec electronics tools¥dev¥DF3746.800)
# Internal RAM   : 0x3ff0000 - 0x3ffefff
#
# NOTICE:
# Allocation of SCONST, CONST and TEXT depends on the user program.
#
# If interrupt handler(s) are specified in the user program then
# the interrupt handler(s) are allocated from address 0 and
# SCONST, CONST and TEXT are allocated after the interrupt handler(s).

SCONST : !LOAD ?R {
    .sconst      = $PROGBITS      ?A .sconst;
};

CONST   : !LOAD ?R {
    .const       = $PROGBITS      ?A .const;
};

TEXT    : !LOAD ?RX {
    .pro_epi_runtime = $PROGBITS    ?AX .pro_epi_runtime;
    .text          = $PROGBITS    ?AX .text;
};

### MINICUBE2用###
MROMSEG : !LOAD ?R 0x00000000
    MonitorROM = $PROGBITS ?A MonitorROM;
};

```

製品の内蔵ROMサイズによりアドレスが異なり  
ます  
(例は内蔵ROMが1024 Kバイトの場合)

デフォルトのリンク・ディレクティブ  
ファイルに追加するコード  
MINICUBE2用の予約領域を確保。

```

SIDATA : !LOAD ?RW V0x3ffb000 {
    .tidata.byte   = $PROGBITS    ?AW .tidata.byte;
    .tibss.byte    = $NOBITS      ?AW .tibss.byte;
    .tidata.word   = $PROGBITS    ?AW .tidata.word;
    .tibss.word    = $NOBITS      ?AW .tibss.word;
    .tidata        = $PROGBITS    ?AW .tidata;
    .tibss         = $NOBITS      ?AW .tibss;
    .sidata        = $PROGBITS    ?AW .sidata;
    .sibss         = $NOBITS      ?AW .sibss;
};

```

```

DATA : !LOAD ?RW V0x3ffb100 {
    .data          = $PROGBITS    ?AW .data;
    .sdata         = $PROGBITS    ?AWG .sdata;
    .sbss          = $NOBITS      ?AWG .sbss;
    .bss           = $NOBITS      ?AW .bss;
};

```

```

### MINICUBE2用###
MRAMSEG : !LOAD ?RW V0x03ffeff0{
    MonitorRAM = $NOBITS ?AW MonitorRAM;
};

```

デフォルトのリンク・ディレクティブ  
ファイルに追加するコード  
MINICUBE2用の予約領域を確保。

```

__tp_TEXT @ %TP_SYMBOL;
__gp_DATA @ %GP_SYMBOL &__tp_TEXT{DATA};
__ep_DATA @ %EP_SYMBOL;

```

```
● main.c

/*-----*/
/*
/*   NEC Electronics   V850ES/Jx3 シリーズ
/*
/*-----*/
/*   V850ES/JJ3 サンプル・プログラム
/*-----*/
/*   低電圧検出時リセット発生編
/*-----*/
/* 【履歴】
/*   2009.9.--   新規作成
/*-----*/
/* 【概要】
/*   本サンプル・プログラムは、低電圧検出回路(LVI)の使用例を示すものです。
/*   VDD < VLVI (VLVI = 2.95V(TYP.) ± 0.10 V)を検出して、内部リセット(LVIリセット)信号を
/*   発生するように設定します。
/*   初期設定完了後は、スイッチ入力の立ち下がりエッジを検出して割り込み処理を行い、
/*   スイッチ入力回数に応じたLED点灯パターンを表示します。
/*   LVI以外によるリセットが発生した場合には、内蔵RAMに保持しているLED点灯パターン
/*   データを初期化します。LVIリセットが発生した場合には、内蔵RAMはLVIリセット
/*   直前のデータを保持し、LVIリセット解除後に、LVIリセット発生直前のLED点灯パターンを
/*   復元して表示するプログラムになっています。
/*
/*
/*   リセット解除後に動作停止状態の周辺機能について、このサンプル・プログラムで
/*   使用しない周辺機能は、設定していません。
/*
/*
/*   <主な設定内容>
/*   ・pragma指令にて、割り込みハンドラの設定、周辺I/Oレジスタ名を記述可能にする
/*   ・チャタリング対策用10msウエイトのウエイト調整値を定義
/*   ・低電圧検出レジスタ(LVIM)設定用0.2msウエイトのウエイト調整値を定義
/*   ・プロトタイプ宣言を実施
/*   ・LED点灯パターンのテーブルを定義
/*   ・内蔵周辺I/Oへのバス・ウエイト、ウォッチドッグ・タイマ2の動作停止、クロック設定処理
/*   ・低電圧検出回路初期化を実施
/*   ・未使用ポートの初期化
/*   ・外部割り込みポート(立ち下りエッジ)、LED出力ポートの初期化
/*   ・ROM化処理
/*   <割り込み処理>
/*   ・LED点灯パターン更新
/*   (スイッチ入力時のチャタリング除去時間10ms)
```

```

/*
/* <スイッチ入力回数とLED点灯パターン>
/*
/* +-----+
/* | SW入力回数 | LED2 | LED1 |
/* | (P03/INTP0) | (PCM2) | (PCM3) |
/* |-----|
/* |    0回    | OFF | OFF |
/* |    1回    | OFF | ON  |
/* |    2回    | ON  | OFF |
/* |    3回    | ON  | ON  |
/* +-----+
/* # 4回目以降は0回からの繰り返し
/*
/*
/* 【ポート入出力の設定】
/*
/* 入力ポート：P03 (INTP0)
/* 出力ポート：PCM2, PCM3
/* 未使用ポート：P00-P02, P04-P06, P10-P11, P30-P39, P40-P42, P50-P55, P60-P615,
/*                P70-P715, P80-P81, P90-P915, PCD0-PCD3, PCM0-PCM1, PCM4-PCM5,
/*                PCS0-PCS7, PCT0-PCT7, PDH0-PDH7, PDL0-PDL15
/*                未使用ポートは全て出力ポート (Low出力) に設定しておく
/*-----*/

/*-----*/
/*          pragma指令          */
/*-----*/
#pragma ioreg                                /* 周辺IOレジスタを記述可にする */
#pragma interrupt INTP0 f_int_intp0         /* 割り込みハンドラ指定          */

/*-----*/
/*          定数定義          */
/*-----*/
#define LIMIT_10ms_WAIT (0x9D58)           /* 10ms ウエイトの調整用定数定義 */
#define LIMIT_250us_WAIT (0x3EF)          /* 250µs (0.25ms) ウエイトの調整用定数定義 */

```

```
/*-----*/
/*   プロトタイプ宣言           */
/*-----*/

    void main( void );           /*   メイン関数           */
static void f_init( void );     /*   初期化関数           */
static void f_init_clk_bus_wdt2( void ); /*   クロック・バス・WDT2初期化関数*/
static void f_init_lvi( void ); /*   低電圧検出回路初期化関数   */
static void f_init_port_func( void ); /*   ポート/兼用機能初期化関数   */

/*-----*/
/* LEDパターンテーブルの設定 */
/*-----*/
const unsigned char LED_TBL[] ={ 0x0C, /* LED表示パターン0 [OFF][OFF] */
                                0x04, /* LED表示パターン1 [OFF][ON]  */
                                0x08, /* LED表示パターン2 [ON] [OFF]  */
                                0x00 }; /* LED表示パターン3 [ON] [ON]  */

/*-----*/
/*   グローバル変数の設定       */
/*-----*/
static unsigned char g_led_ptn_cnt = 0; /* LED表示パターン格納変数 */

/*****
/*   メインモジュール           */
/*****
void main(void)
{
    f_init(); /*   初期化実施           */

    __EI(); /*   割り込み許可           */

    while(1); /*   メイン・ループ(無限ループ)   */

    return;
}

```



```

/*-----*/
/*      初期化モジュール      */
/*-----*/
static void f_init( void )
{
    f_init_clk_bus_wdt2();          /* 内蔵周辺I/Oへのバス・ウェイト，WDT2の停止，クロック設定処理*/

    f_init_lvi();                  /* LVI設定 */

    f_init_port_func();           /* ポート/兼用機能の設定 */

    return;
}

```

```

/*-----*/
/* クロック・バス・WDT2初期化      */
/*-----*/
static void f_init_clk_bus_wdt2( void )
{
    VSWC = 0x11;                  /* 内蔵周辺へのバスウェイト設定 */
                                  /* OCDMを「通常動作モード」に指定*/

```

```

#pragma asm
    st.b r0, PRCMD
    st.b r0, OCDM
#pragma endasm

```

特定レジスタへのアクセスは、アセンブラで記述する必要がありますので、注意が必要です。

```

RCM = 0x01;                      /* 内蔵発振器を停止 */
WDTM2 = 0x00;                   /* ウォッチドッグ・タイマ2を停止 */

PLLON = 0;                      /* PLL動作停止 */
                                  /* PLLの通倍率を8通倍に設定 */

```

```

#pragma asm
    push r10
    mov 0x0B, r10
    st.b r10, PRCMD
    st.b r10, CKC
    pop r10
#pragma endasm

```

```

PLLON = 1;                      /* PLL動作許可 */
while( LOCK );                 /* PLL安定(ロック)待ち */

```

```
SELPLL = 1; /* PLLモードに設定 */
```

```
/* PCCレジスタ設定 */
```

```
/* クロック分周を「なし」に設定 */
```

```
#pragma asm
    push r10
    mov 0x80, r10
    st.b r10, PRCMD
    st.b r10, PCC
    pop r10
#pragma endasm
```

特定レジスタへのアクセスは、アセンブラで記述する必要がありますので、注意が必要です。

```
    return;
}

/*-----*/
/* 低電圧検出回路初期化 */
/*-----*/
static void f_init_lvi( void )
{
    extern unsigned int _S_romp; /* ROM化シンボルの外部参照 */
    static unsigned char save_led_ptn; /* LED点灯パターン退避変数 */
    unsigned int loop_wait;

    /* LVIリセット */
    if( RESF.0 == 1 )
    {
        save_led_ptn = g_led_ptn_cnt; /* LED点灯パターン退避 */
        _rcopy( &_S_romp, -1 ); /* ROM化処理実施 */

        if(RAMF == 0)
        {
            g_led_ptn_cnt = save_led_ptn; /* LED点灯パターン復帰 */
        }
    }
}
}
```

```

/* LVI以外のリセット */

else
{
    _rcopy( &_S_romp, -1 );          /* ROM化情報のコピー          */
                                    /* この処理でLED点灯パターンの指定 */
                                    /* (g_led_ptn_cnt)を初期化      */

    LVIMK = 1;                       /* LVIの割り込みをマスク      */
    LVIS = 0x00;                     /* 低電圧検出レベルを2.95±0.10 Vに設定 */
}

```

/\* 低電圧検出動作許可 \*/

```

#pragma asm
    push    r10
    mov     0x80, r10
    st.b   r10, PRCMD
    setl   LVION
    pop     r10
#pragma endasm

```

特定レジスタへのアクセスは、アセンブラで記述する必要がありますので、注意が必要です。

```

/* 250µs(0.25ms)のウェイト */

for( loop_wait = 0 ; loop_wait < LIMIT_250us_WAIT ; loop_wait++ )
{
    __nop();
}

```

```
while( LVIF == 1 );
```

/\* 電源電圧が低電圧検出電圧よりも高くなるまで、 \*/  
/\* 無限ループとなるため、注意が必要です。 \*/

/\* 低電圧検出時の動作：リセット動作に設定 \*/

```

#pragma asm
    push    r10
    mov     0x82, r10
    st.b   r10, PRCMD
    setl   LVIMD
    pop     r10
#pragma endasm

```

特定レジスタへのアクセスは、アセンブラで記述する必要がありますので、注意が必要です。

```
}
```

/\* 内蔵RAM保持電圧検出のフラグをクリア \*/

```
#pragma asm
    st.b  r0, PRCMD
    st.b  r0, RAMS
#pragma endasm
```

特定レジスタへのアクセスは、アセンブラで記述する必要がありますので、注意が必要です。

```
return;
}
```

/\*-----\*/

/\* ポート/兼用機能の設定 \*/

/\*-----\*/

static void f\_init\_port\_func( void )

{

```
P0 = 0x00; /* P00-P02, P04-P06を出力LOW, P03を兼用端子INTP0に設定*/
```

```
PM0 = 0x88; V850ES/JG3での設定値は0x8B
```

```
PFC0 = 0x00;
```

```
PMC0 = 0x08;
```

```
P1 = 0x00; /* P10, P11を出力LOWに設定 */
```

```
PM1 = 0xFC;
```

```
P3 = 0x0000; /* P30-P39を出力LOWに設定 */
```

```
PM3 = 0xFC00;
```

```
PMC3 = 0x0000;
```

```
P4 = 0x00; /* P40-P42を出力LOWに設定 */
```

```
PM4 = 0xF8;
```

#if(0) /\* MINICUBE2使用時はP4をCSIB0として使用する為、 \*/

/\* PMC4の設定を行いません (QB-V850ESJJ3-TB) \*/

```
PMC4 = 0x00;
```

#endif

```
P5 = 0x00; /* P50-P55を出力LOWに設定 */
```

```
PM5 = 0xC0;
```

```
PMC5 = 0x00;
```

```
P6 = 0x0000; /* P60-P615を出力LOWに設定 */
```

```
PM6 = 0x0000; V850ES/JG3では設定不要
```

```
PMC6 = 0x0000;
```

```
P7H = 0x00; /* P70-P715を出力LOWに設定 */
```

```
P7L = 0x00;
```

```
PM7H = 0x00; V850ES/JG3ではP70-P711のみ設定
```

```
PM7L = 0x00;
```

```

P8      = 0x00;          /* P80-81を出力LOWに設定      */
PM8     = 0xFC;
PMC8    = 0x00;
V850ES/JG3では設定不要

P9      = 0x0000;       /* P90-P915を出力LOWに設定    */
PM9     = 0x0000;
PMC9    = 0x0000;

PCD     = 0x00;          /* PCD0-PCD3を出力LOWに設定   */
PMCD    = 0xF0;
V850ES/JG3では設定不要

PCM     = LED_TBL[g_led_ptn_cnt]; /* PCM0-PCM1を出力LOWに、PCM2-PCM3に点灯パターンを設定*/
PMCM    = 0xC0;
PMCCM   = 0x00;
V850ES/JG3ではPCM0-PCM3のみ設定

PCS     = 0x00;          /* PCS0-PCS7を出力LOWに設定   */
PMCS    = 0x00;
V850ES/JG3では設定不要

PCT     = 0x00;          /* PCT0-PCT7を出力LOWに設定   */
PMCT    = 0x00;
PMCCT   = 0x00;
V850ES/JG3ではPCT0, PCT1, PCT4, PCT6のみ設定

PDH     = 0x00;          /* PDH0-PDH7を出力LOWに設定   */
PMDH    = 0x00;
PMCDH   = 0x00;
V850ES/JG3ではPDH0-PDH5のみ設定

PDL     = 0x0000;       /* PDL0-PDL15を出力LOWに設定  */
PMDL    = 0x0000;
PMCDL   = 0x0000;

/* 割り込み機能設定 */
INTF0   = 0x08;          /* INTF0立ち下りエッジ指定    */
INTRO   = 0x00;          /* ↓                            */
PIC0    = 0x07;          /* INTF0 優先度7, マスク解除  */

return;
}

/*****
/* 割り込みモジュール */
*****/
__interrupt
void f_int_intp0( void )

```

```
{
    unsigned int loop_wait;

    /* チャタリング用 10msウエイト */
    for( loop_wait = 0 ; loop_wait < LIMIT_10ms_WAIT ; loop_wait++ )
    {
        __nop();
    }

    if( ( P0 & 0x08 ) == 0x00 )          /* ウエイト後のSW1押下状態認識 */
    {
        g_led_ptn_cnt++;                /* 点灯パターンを変更(4種類) */
        g_led_ptn_cnt &= 3;
        PCM = LED_TBL[g_led_ptn_cnt];  /* 更新された点灯パターンを設定 */
    }

    PICO &= (unsigned char)~0x80;      /* FailSafe: 複数要求をクリア */

    return;                             /* __interrupt修飾子によりretiに */
}
```

