

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、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-L

### 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

この資料は、サンプル・プログラムの動作概要や使用方法、および割り込み機能の設定方法や活用方法について説明したものです。サンプル・プログラムでは、スイッチ入力の立ち下がりエッジを検出して割り込みを発生させ、スイッチの入力回数に応じたLED点灯パターンを表示します。

#### 対象デバイス

V850ES/JF3-L マイクロコントローラ

V850ES/JG3-L マイクロコントローラ

#### 目次

第1章 概要 ... 3	
1.1 初期化 ... 3	
1.2 割り込み許可 ... 2	
1.3 メイン・ループ ... 3	
1.4 割り込み処理 ... 3	
第2章 回路図 ... 5	
2.1 回路図 ... 5	
2.2 周辺ハードウェア ... 5	
第3章 ソフトウェアについて ... 6	
3.1 ファイル構成 ... 6	
3.2 使用する内蔵周辺機能 ... 8	
3.3 初期設定と動作概要 ... 8	
3.4 フロー・チャート ... 9	
3.5 V850ES/JG3-LとV850ES/JF3-Lの違い ... 11	
3.6 ROM化について（C言語のみ） ... 11	
3.7 セキュリティIDについて ... 13	
第4章 レジスタ設定について ... 14	
4.1 割り込み端子設定とLED点灯用端子の設定 ... 15	
4.2 割り込み処理 ... 23	
第5章 関連資料 ... 24	
付録A プログラム・リスト ... 25	

資料番号 U19480JJ1V0AN00

発行年月 October 2008 NS

- 本資料に記載されている内容は2008年10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に、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)

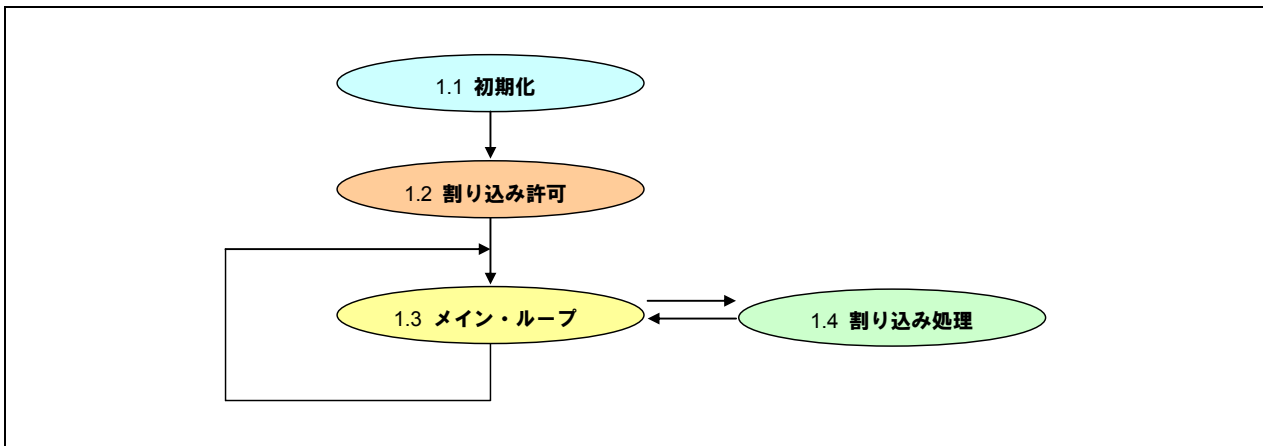
- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

M8E0710J

# 第1章 概 要

このサンプル・プログラムでは、割り込み機能の使用例を示しています。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。

ソフトウェアの主な内容は、次のとおりです。



## 1.1 初期化

### <オプション・バイトの参照>

- ・リセット解除後の発振安定時間を参照

### <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：1>設定
- ・オンチップ・デバッグ・モード<通常動作モード>の設定
- ・内蔵発振器を停止，ウォッチドッグ・タイマ停止
- ・CPUクロックを分周なしに設定
- ・PLLモードに設定し5 MHz×4通倍 = 20 MHz動作に設定

### <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（LED1, LED2共に消灯値，出力指定）

### <ROM化>

- ・ROM化処理（初期値あり変数の初期化）・・・C言語のみ

## 1.2 割り込み許可

- ・EI命令による 割り込み許可

### 1.3 メイン・ループ

- ・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端子の立ち下がりエッジを検出し，割り込み処理を行います。

割り込み処理では，INTP0端子の立ち下が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端子の立ち下がりエッジを検出後，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。

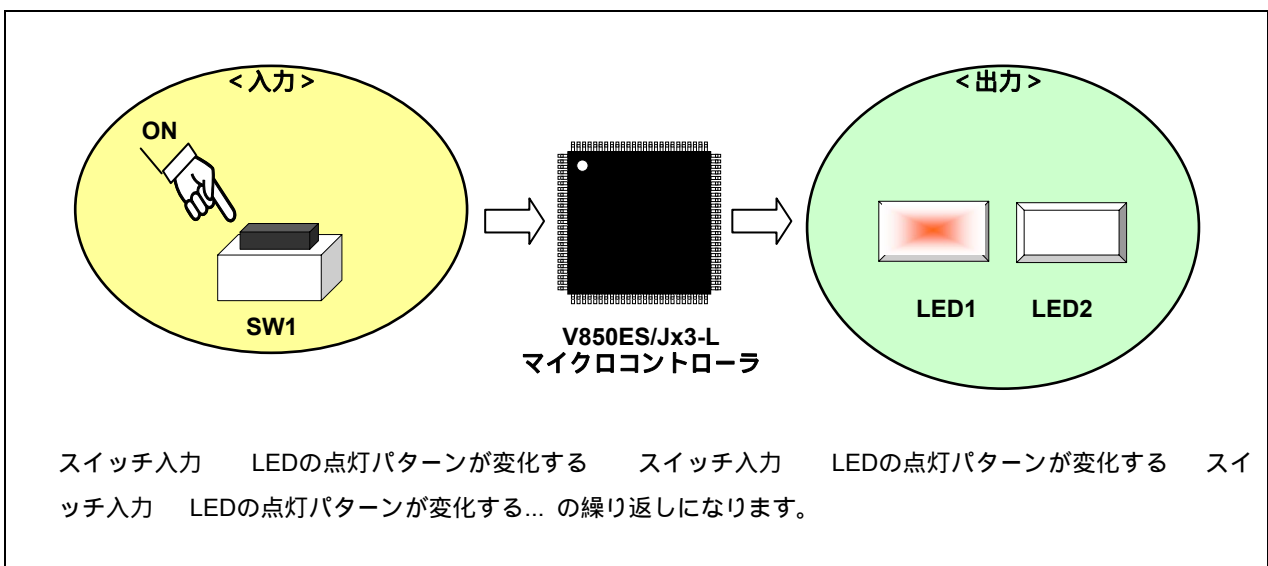


表1 - 1 LED点灯パターン

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回目以降の繰り返しになります。

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



【コラム】 チャタリングとは

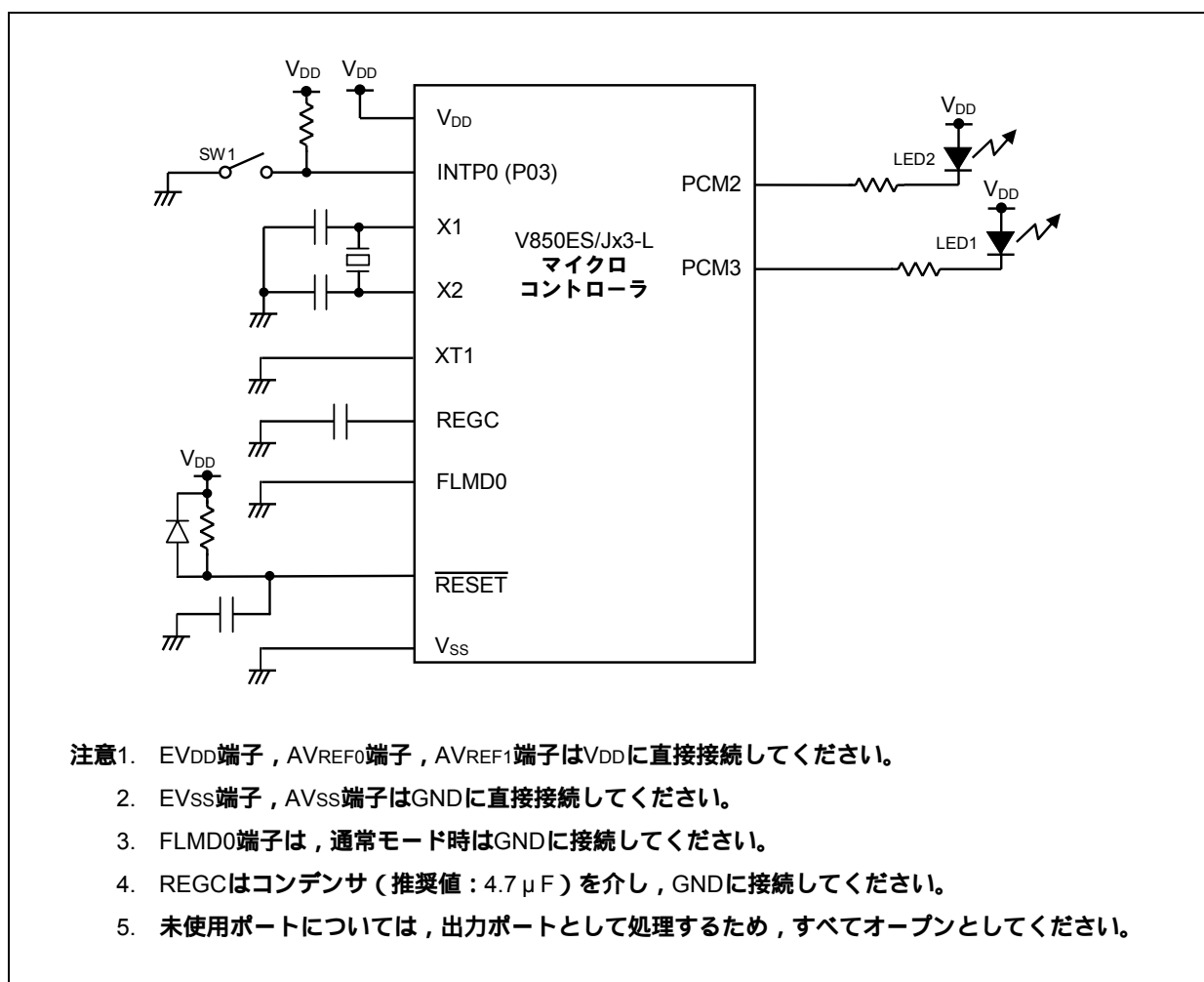
スイッチが切り替わった直後に，接点が機械的にばたつくことにより，電気信号がONとOFFを繰り返す現象のことです。

## 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。



### 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。


#### 【C言語版】

ファイル名 (ツリー構造)	説明	同封圧縮 (*.zip) ファイル	
			
<pre> c ├── conf │   ├── crtE.s │   ├── AppNote_INT.dir │   ├── AppNprj │   └── AppNote_INT.prw └── src     ├── main.c     ├── minicube2.s     └── opt_b.s                     </pre>	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		
	オプション・バイト設定を行うソース・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(I)」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



【アセンブラ版】

ファイル名 (ツリー構造)	説明	同封圧縮 (*.zip) ファイル	
			
<pre>asm ├── conf │   ├── crtE.s │   ├── AppNote_INT.dir │   ├── AppNote_INT.prj │   └── AppNote_INT.prw └── src     ├── main.s     ├── minicube2.s     └── opt_b.s</pre>	スタートアップ・ルーチン・ファイル <sup>※1</sup>	-	
	リンク・ディレクティブ・ファイル <sup>※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アセンブラソース・ファイル		
	MINICUBE2用の領域予約を行うソース・ファイル		
	オプション・バイト設定を行うソース・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(I)」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\850\_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。

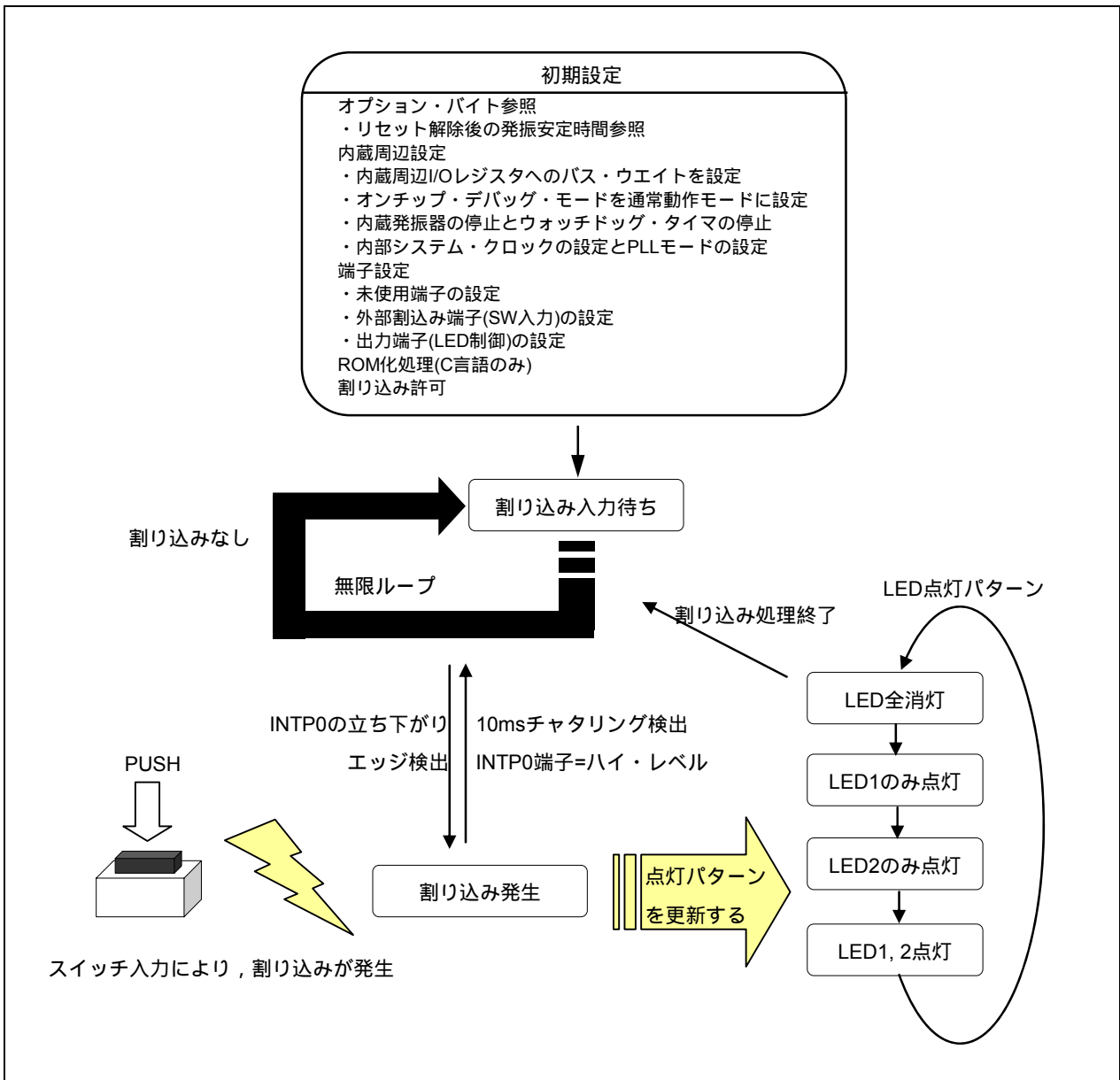
- ・外部割り込み入力（スイッチ入力用）：INTP0（SW1）
- ・出力ポート（LED点灯用）：PCM2（LED2）、PCM3（LED1）

### 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。

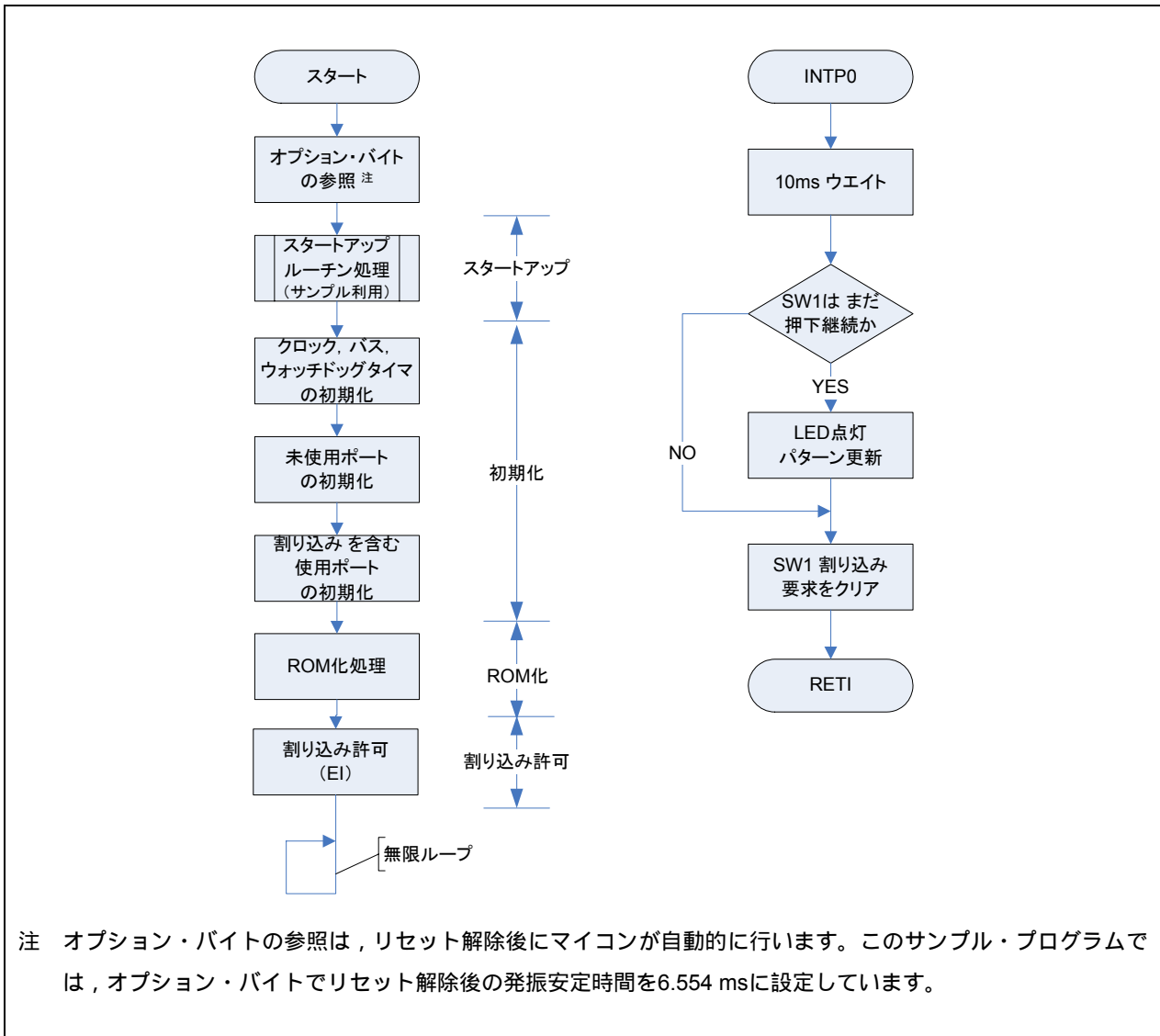
初期設定完了後は、スイッチ入力（SW1）の立ち下がりエッジを検出して割り込み処理を行い、スイッチの入力回数に応じて、2つのLED（LED1, LED2）の点灯パターンを制御します。

詳細については、次の状態遷移図に示します。



### 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。



**【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・ルーチンとは、V850をリセットしたあと、メイン関数を実行する前に、実行するルーチンを言います。基本的にはシステムをリセットしたあとの初期化を行います。

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/JG3-LとV850ES/JF3-Lの違い

V850ES/JG3-Lは、V850ES/JF3-L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P7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・リストを参照してください。

### 3.6 ROM化について（C言語のみ）

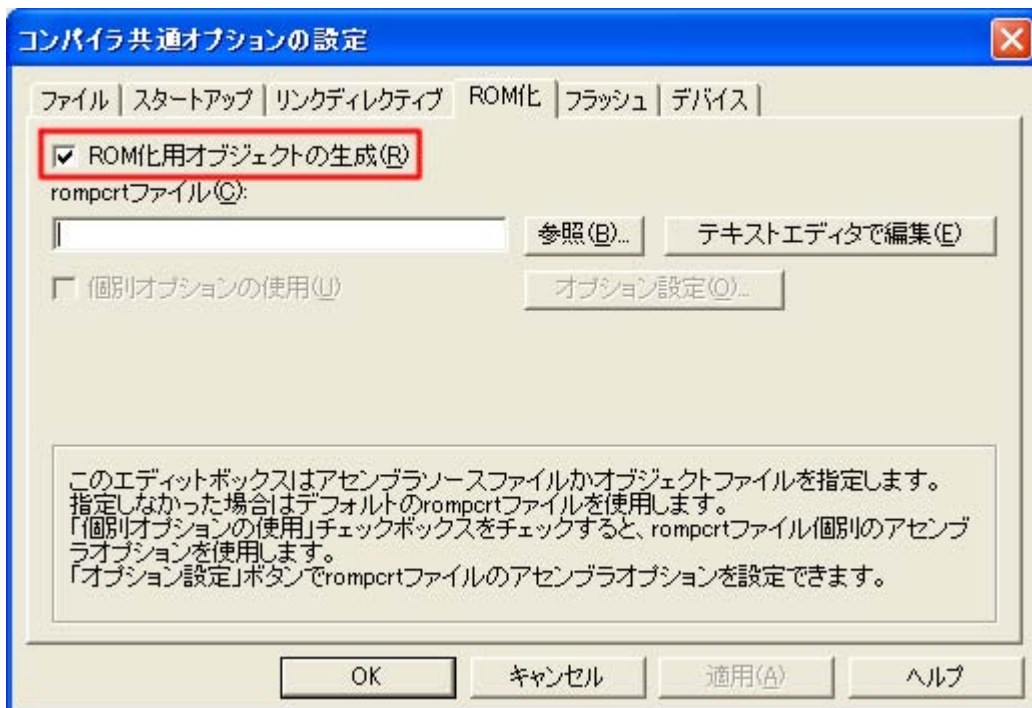
このサンプル・プログラム（C言語）では、内蔵周辺の初期化後に「ROM化」情報のコピー処理を行っています。「ROM化」情報とは、初期値のある変数（初期値のある変数が配置されるセクション）の初期値情報のことで、「ROM化」情報をRAMにコピーすることにより、初期値のある変数（初期値のある変数が配置されるセクション）は、初めて初期値を持つことになります<sup>※</sup>。

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。

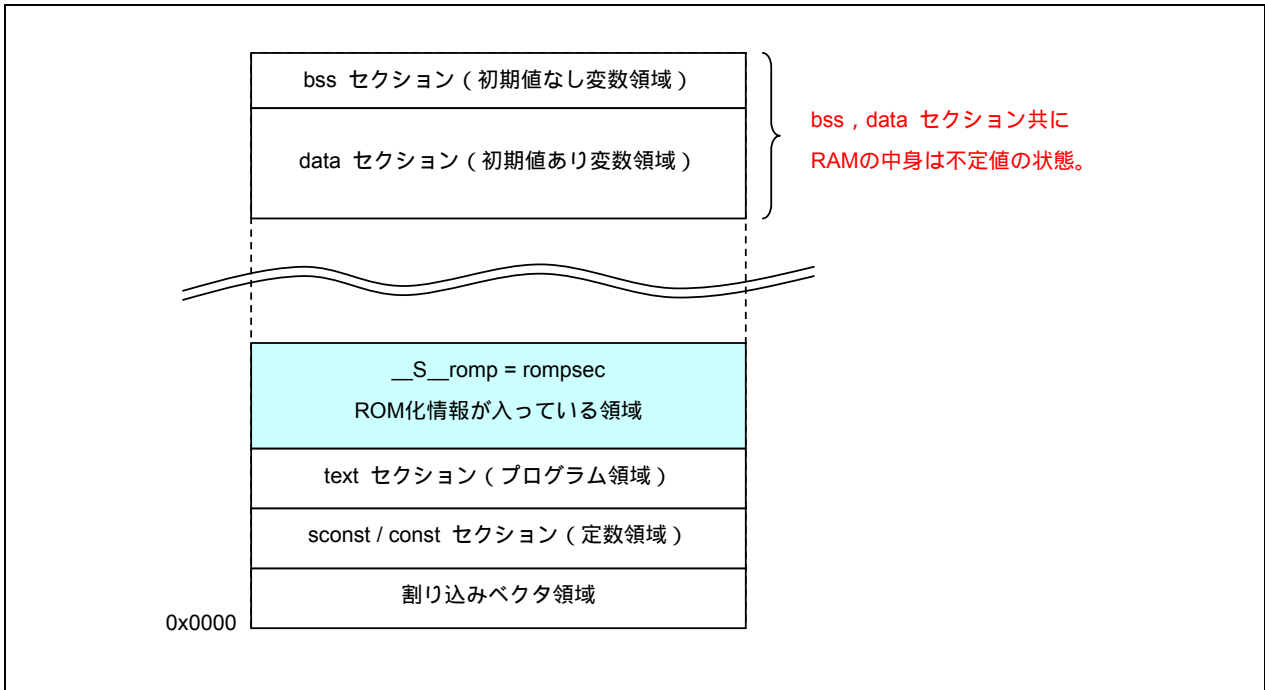
次に、ROM化を行うための手順を説明します。

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。



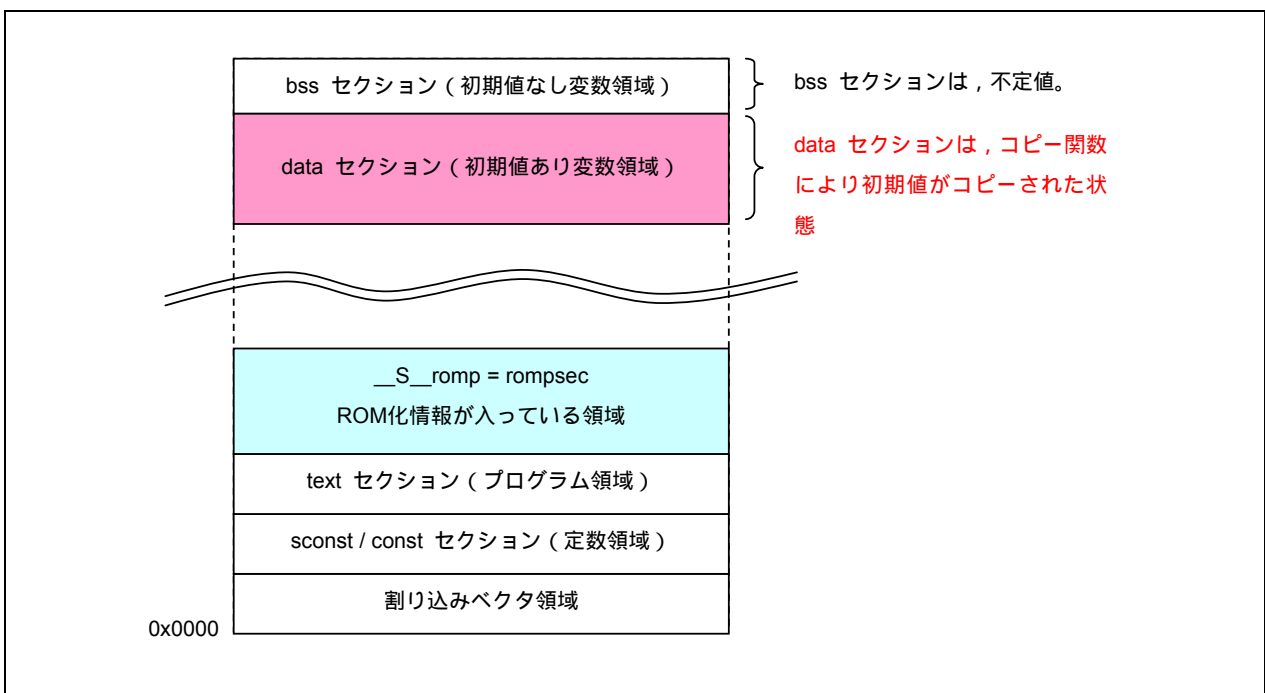
ROM化情報を格納するセクション (rompsec) は、自動でプログラム領域 (.text) セクションの直後に追加されますが、このチェック・ボックス操作によって、デフォルトの「rompct.o」で定義されたラベル“\_S\_romp”に、“rompsec”と同じアドレスを指すコードが生成され、コピー関数が格納されているライブラリ“lib.a”が自動的にリンクされます。

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。



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。



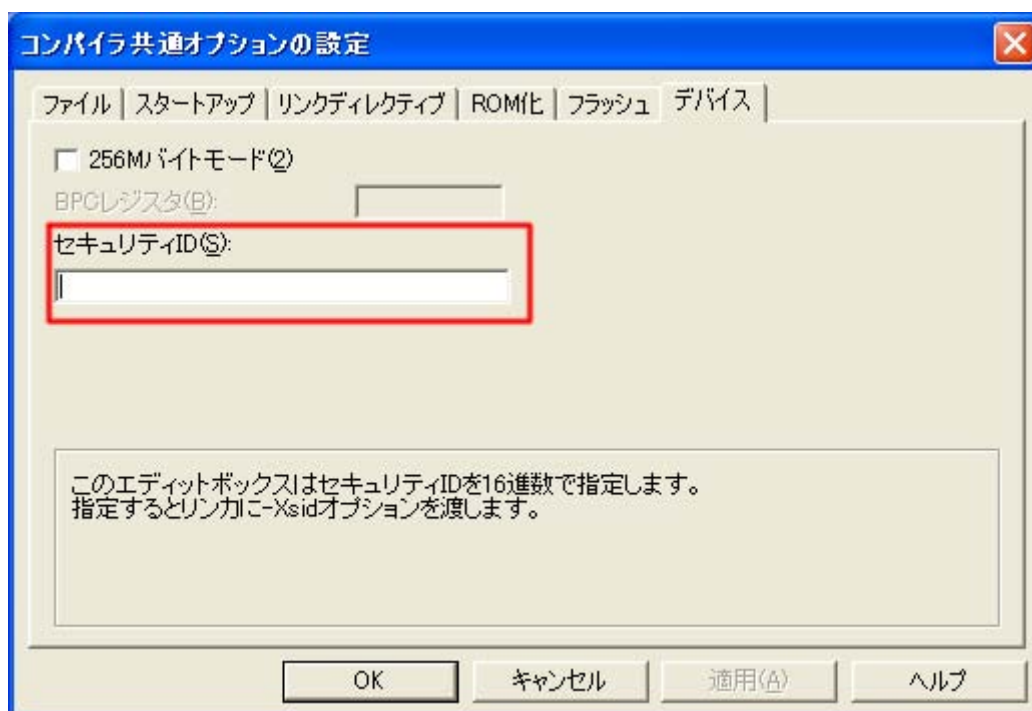
### 3.7 セキュリティIDについて

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。

IDコードは、あらかじめ内蔵フラッシュ・メモリ領域の0x0000070-0x0000079の10バイト分に設定し、デバッガがID認証を行います。

このID照合が一致していれば、セキュリティが解除され、フラッシュ・メモリ読み出し許可、オンチップ・デバッグ・エミュレータ使用許可となります。

このサンプル・プログラム（環境一式版）では、セキュリティIDの設定を行っていないため、デフォルトのセキュリティID値 “0xFFFFFFFFFFFFFFFF” が適用されるようになっています。

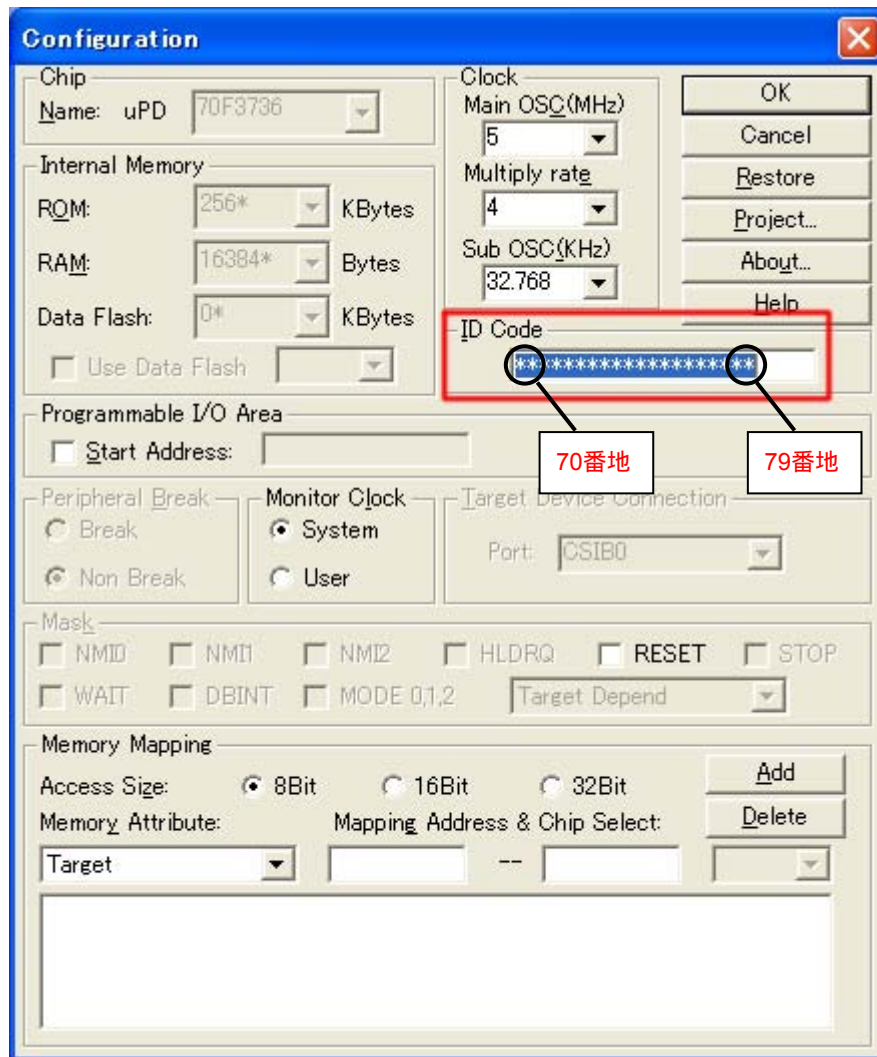


備考 コンパイラ共通オプションの設定「セキュリティID」では、フラッシュ・メモリ搭載デバイスの“セキュリティID”を設定します。

IDは、0xで始まる10バイト以内の16進数で指定します。

本オプションの指定、またはアセンブラ記述（.section "SECURITY\_ID" 使用）によるセキュリティIDの指定が省略された場合、“0xFFFFFFFFFFFFFFFF” が指定されたものとして扱います。

このサンプル・プログラム（環境一式版）を使用して、プログラムをダウンロードして動作させた場合、マイコンのセキュリティID領域には、“0xFF”が設定されるため、次回デバッガ接続時のID Code入力エリアに“FFFFFFFFFFFFFFFF”を設定（設定していない場合のデフォルト値）しないと、オンチップ・デバッグ・エミュレータが使用できないため、注意が必要です。



- ・ 10バイトのIDコードのうち、0x0000079のビット7は、オンチップ・デバッグ・エミュレータ使用許可フラグです（0：使用禁止，1：使用許可）
- ・ オンチップ・デバッグ・エミュレータを起動すると、デバッガがID入力を要求します。デバッガ上で入力したIDコードと、0x0000070-0x0000079に埋め込んだIDコードが一致すればデバッガが起動します。
- ・ IDコードが一致しても、オンチップ・デバッグ・エミュレータ使用許可フラグが“0”である場合は、デバッグを行うことはできません。



## 第4章 レジスタ設定について

この章では、割り込み端子設定とLED点灯用端子の設定、および割り込み処理について説明します。

その他の初期設定については、V850ES/Jx3-L **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。

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/JG3-L 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/JF3-L 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、アセンブラ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
- CA850 Cコンパイラ・パッケージ アセンブリ言語編 ユーザーズ・マニュアル

## 4.1 割り込み端子設定とLED点灯用端子の設定

このサンプル・プログラムでは、P03端子をSW1入力の外部割り込み端子として使用し、割り込みは立ち下りエッジを検出する設定にします。

また、PCM2-PCM3端子をLED点灯用として使用するため、出力ポートとして設定します。

### 4.1.1 ポート0モード・レジスタ (PM0) の設定

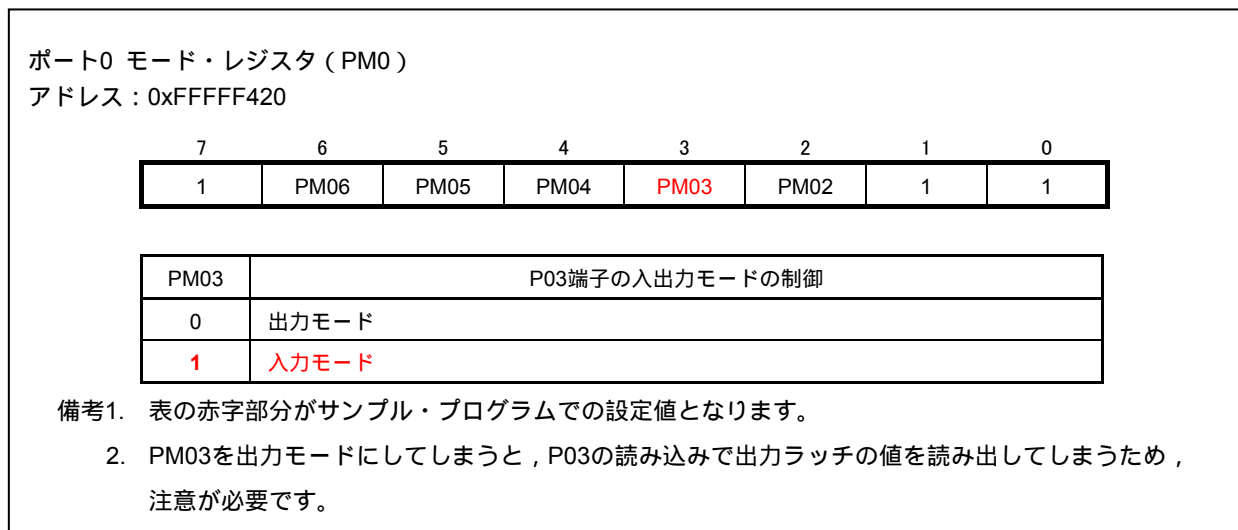
PM0レジスタは、P02-P06端子の入出力モードの制御ができます。

このサンプル・プログラムでは、チャタリングの確認で端子状態を読み込みますので、P03端子の入出力モードを「入力モード」に設定します。

8/1ビット単位でリード/ライト可能です。

リセットにより、FFHになります。

図4 - 1 PM0レジスタのフォーマット



PM0の設定値は「8BH」となります。

#### ・C言語の場合

```
PM0 = 0x8B; /* P03を入力ラッチに接続 */
```

#### ・アセンブリ言語の場合

```
SET_REG8 0x8B, PM0 -- P03を入力ラッチに接続
```

備考 SET\_REG8 マクロは、サンプル・プログラムで作成したマクロです。

8ビットの周辺I/Oレジスタへのアクセスに関して、プログラム記述を簡単にするためのレジスタ・アクセス・マクロです。

```
-- 【8bit周辺I/Oレジスタ・アクセス・マクロ】
.macro SET_REG8 val, reg
    mov    val, r6          -- テンポラリr6に値を入れておき
    st.b   r6, (reg)[zero] -- 周辺I/Oにr6の値を設定する
.endm
```

### 4.1.2 ポート0 モード・コントロール・レジスタ (PMC0) の設定

PMC0レジスタは、P02-P06端子の動作モードの指定ができます。

このサンプル・プログラムでは、P03端子の動作モードをINTP0（外部割り込み）入力に設定します。

8/1ビット単位でリード/ライト可能です。

リセットにより、00Hになります。

図4 - 2 PMC0レジスタのフォーマット

ポート0モード・コントロール・レジスタ (PMC0)  
 アドレス : 0xFFFFF440

	7	6	5	4	3	2	1	0
	0	PMC06	PMC05 <sup>注</sup>	PMC04	PMC03	PMC02	0	0

PMC03	P03端子の動作モードの指定
0	入出力ポート
1	INTP0入力 / ADTRG入力

注 P05/INTP2/DRST $\bar$ 端子は、OCDM.OCDM0ビット = 1 のときは、PMC05ビットの値に関係なく $\bar$ DRST端子となります。

備考1. 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。

2. P02-P06端子は、兼用機能の入力時にはヒステリシス特性を持ちますが、ポート・モード時にはヒステリシス特性を持ちません。

PMC0の設定値は「08H」となります。

・C言語の場合

```
PMC0 = 0x08; /* INTP0端子の入力設定 */
```

・アセンブリ言語の場合

```
SET_REG8 0x08, PMC0 -- INTP0端子の入力設定
```

### 4. 1. 3 外部割り込み立ち下がり，立ち上がりエッジ指定レジスタ0 (INTF0, INTR0) の設定

ビット2でNMI端子，ビット3-ビット6で外部割り込み端子 (INTP0-INTP3) の立ち下がり，立ち上がりエッジ検出を指定する8ビットのレジスタです。

このサンプル・プログラムでは，立ち下がりエッジを検出する設定にします。

8/1ビット単位でリード/ライト可能です。

リセットにより，00Hになります。

図4 - 3 INTF0レジスタのフォーマット

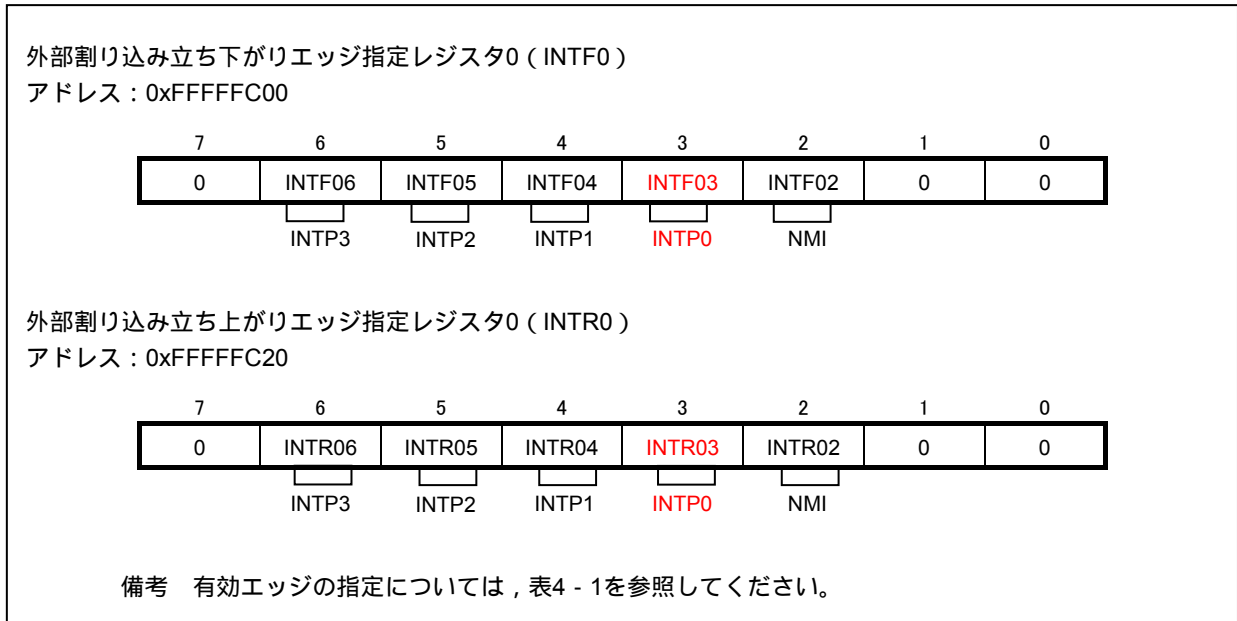


表4 - 1 有効エッジの指定

INTF03	INTR03	有効エッジ (立ち下り) 指定
0	0	エッジ検出なし
<b>0</b>	<b>1</b>	立ち上がりエッジ検出
<b>1</b>	<b>0</b>	立ち下がりエッジ検出
1	1	両エッジ

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。

INTF0の設定値は「08H」，INTR0の設定値は「00H」となります。

・C言語の場合

```
INTF0 = 0x08;          /* INTP0端子の入力設定 */
INTR0 = 0x00;
```

・アセンブリ言語の場合

```
SET_REG8    0x08, INTF0    -- INTP0立ち下りエッジ指定
SET_REG8    0x00, INTR0    --↓
```

#### 4.1.4 割り込み制御レジスタ (PIC0) の設定

割り込み要求信号 (マスカブル割り込み) ごとに割り当てられ, 各割り込みに対する制御条件を設定します。

このサンプル・プログラムでは, INTP0を最低優先順位で使用可能にします。

8/1ビット単位でリード/ライト可能です。

リセットにより, 47Hになります。

**注意** xxlCn.xxIFnビットを読み出す場合は, 割り込み禁止 (DI) 状態または割り込みをマスクした状態で行ってください。割り込み許可 (EI) 状態または割り込みマスクを解除した状態でxxIFnビットを読み出すと, 割り込みの受け付けとビットの読み出しのタイミングが競合した場合に, 正常な値が読み出せないことがあります。

図4 - 4 PIC0レジスタのフォーマット

INTP0割り込み制御レジスタ (PIC0)

アドレス : 0xFFFFF112

7	6	5	4	3	2	1	0
PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00

PIF0	割り込み要求フラグ
0	割り込み要求信号なし
1	割り込み要求信号あり

PMK0	割り込みマスク・フラグ
0	割り込み処理を許可
1	割り込み処理を禁止 (保留)

PPR02	PPR01	PPR00	割り込み優先順位指定ビット
0	0	0	レベル0 (最高位) を指定
1	0	1	レベル1を指定
0	1	0	レベル2を指定
0	1	1	レベル3を指定
1	0	0	レベル4を指定
1	0	1	レベル5を指定
1	1	0	レベル6 を指定
1	1	1	レベル7 (最低位) を指定

**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。



【コラム】割り込み要求フラグPIF0について

PIC0レジスタのPIF0「割り込み要求フラグ」は, 割り込み要因発生時に「1」にセットされ, 割り込み要求信号が受け付けられると, ハードウェアにより自動的にリセットされます。

PIC0の設定値は「07H」となります。

• C言語の場合

```
PIC0 = 0x07; /* INTPO 優先度7, マスク解除 */
```

• アセンブリ言語の場合

```
SET_REG8 0x07, PIC0 -- 優先度7, マスク解除
```

#### 4.1.5 ポートCMレジスタ (PCM) , ポートCMモード・レジスタ (PMCM) の設定

ポートCMは1ビット単位で入出力を制御できる4ビットのポートです。

このサンプル・プログラムでは、LED1/LED2を点灯させない出力設定にしたあとに、PCM2, PCM3端子を出力ポートとして動作させる設定にします。

8/1ビット単位でリード/ライト可能です。

リセットにより、PCMレジスタは00Hに、PMCMレジスタはFFHになります。

図4 - 5 PCMレジスタのフォーマット

ポートCM レジスタ (PCM)  
アドレス : 0xFFFFF00C

7	6	5	4	3	2	1	0
0	0	0	0	PCM3	PCM2	PCM1	PCM0

PCM2	PCM3	出力データの制御 (出力モード時)
0	0	0を出力
1	1	1を出力

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。

図4 - 6 PMCMレジスタのフォーマット

ポートCM モード・レジスタ (PMCM)  
アドレス : 0xFFFFF02C

7	6	5	4	3	2	1	0
1	1	1	1	PMCM3	PMCM2	PMCM1	PMCM0

PMCM2	PMCM3	入出力モードの制御
0	0	出力モード
1	1	入力モード

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。



**【コラム】 出力ポートの制御について**

一般的に、出力ポートとして初期化を行う場合、ポート・モード (入出力) の設定よりも先に、出力ラッチの設定を行います (Pnの設定 PMnの設定の順番)。

先にポート・モードの設定を行ってしまうと、出力ラッチにその時に入っている値が端子から出力されてしまい、意図しない端子出力を一瞬行ってしまう可能性があるからです。

このサンプル・プログラムでは、PCM2-PCM3をLED点灯用の出力ポートとして使用するため、初期設定でPCM2-PCM3の出力ラッチのレベルをハイ・レベル出力にしておき、その後にPCM2-PCM3を出力ポートとして動作させます。

LEDは、PCM2-PCM3からの出力がロウ・レベルのときに点灯します（2.1 回路図を参照）。

PCMの設定値は「0CH」、PMCMの設定値は「F0H」となります。

・C言語の場合

```
PCM = 0x0C; /* LED1,2 共に消灯設定値 */
PMCM = 0xF0; /* PCM端子出力設定 */
```

・アセンブリ言語の場合

```
SET_REG8 0x0C, PCM -- LED1,2消灯値
SET_REG8 0xF0, PMCM -- PCM出力設定
```



## 4.2 割り込み処理

このサンプル・プログラムでは、割り込み処理内でチャタリング除去と、点灯パターンの更新、割り込み要求のクリアを行っています。

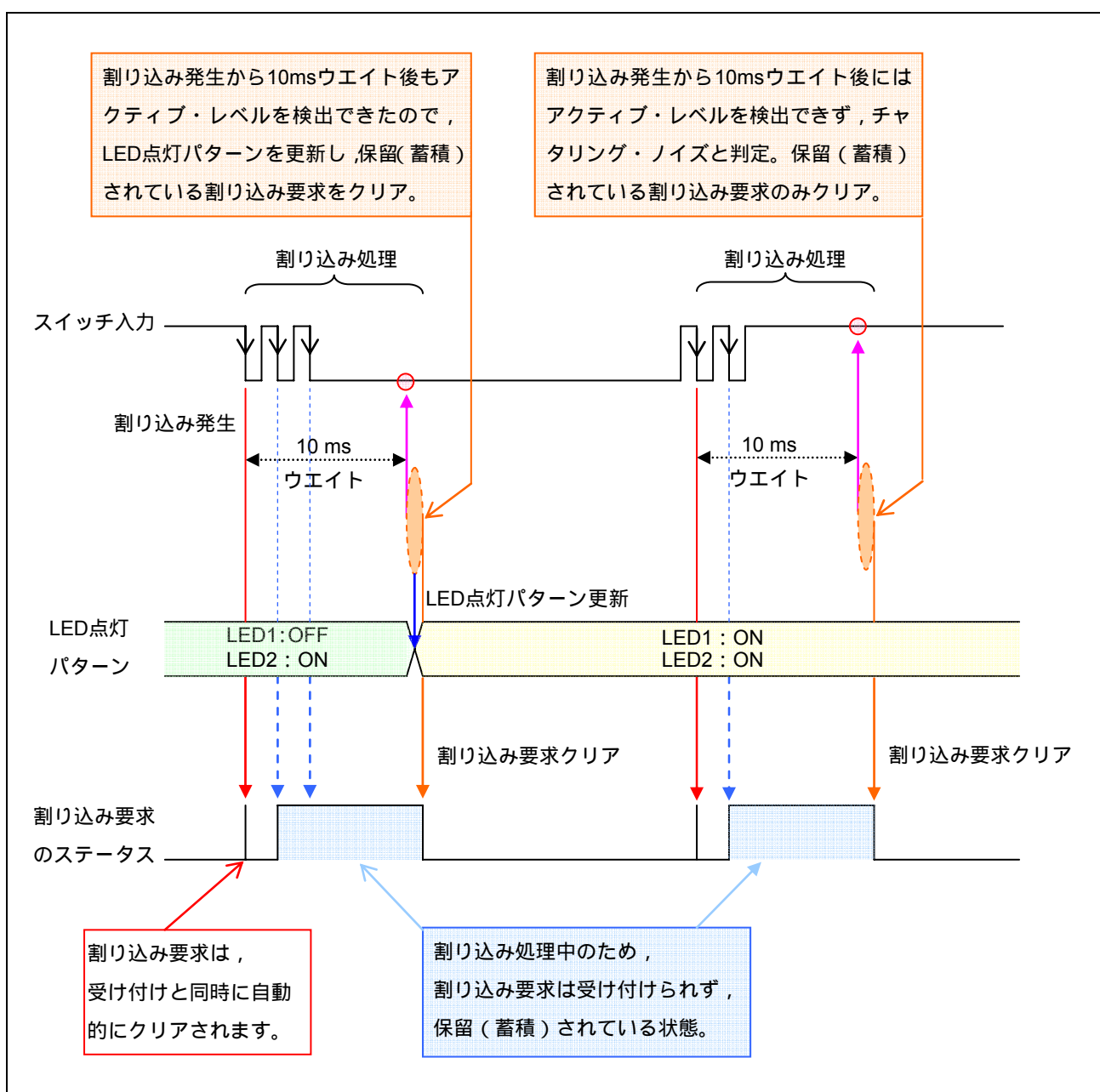
割り込みが発生すると、割り込み処理内で10 msウエイトし、ウエイト後にスイッチ入力状態を確認してロウ・レベル（アクティブ・レベル）であれば、スイッチ入力があったと見なし、ハイ・レベル（インアクティブ・レベル）であれば、チャタリング・ノイズと判定してスイッチ入力はなかったと見なします。

スイッチ入力があると、現在のLED点灯パターンから、次のLED点灯パターンに更新します。

LEDの更新パターンについては、表1-1 LED点灯パターンを参照してください。

LED点灯パターンの更新後、チャタリング・ノイズによって発生してしまった可能性のある、保留（蓄積）されている割り込み要求をクリアします。

スイッチ入力を基準とした、割り込み処理のタイミング・チャートを、次に示します。



## 第5章 関連資料

資料名	和文 / 英文
V850ES/JF3-L ユーザーズ・マニュアル	<a href="#">PDF</a>
V850ES/JG3-L ユーザーズ・マニュアル	<a href="#">PDF</a>
PM+ Ver.6.30 ユーザーズ・マニュアル	<a href="#">PDF</a>
CA850 Ver.3.20 Cコンパイラ・パッケージ 操作編	<a href="#">PDF</a>
CA850 Ver.3.20 Cコンパイラ・パッケージ C言語編	<a href="#">PDF</a>
CA850 Ver.3.20 Cコンパイラ・パッケージ アセンブリ言語編	<a href="#">PDF</a>
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	<a href="#">PDF</a>
V850ES アーキテクチャ編	<a href="#">PDF</a>

## 付録A プログラム・リスト

プログラム・リスト例として、V850ES/Jx3-L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。

- opt\_b.s (アセンブリ言語版・C言語版 共通)

```
#-----  
#  
# NEC Electronics      V850ES/Jx3-L シリーズ  
#  
#-----  
# V850ES/JG3-L JF3-L サンプル・プログラム  
#-----  
# 割り込み編  
#-----  
# 【履歴】  
# 2008.7.-- 新規作成  
#-----  
# 【概要】  
# 本サンプル・プログラムは、オプション・バイトの設定を行う  
#-----
```

```
.section "OPTION_BYTES"  
.byte 0b00000101 -- 0x7a (5MHz: 発振安定時間 6.554ms に設定)  
.byte 0b00000000 -- 0x7b      ↑  
.byte 0b00000000 -- 0x7c      ↑  
.byte 0b00000000 -- 0x7d 0x7b-0x7f番地には0x00を設定する必要がある  
.byte 0b00000000 -- 0x7e      ↓  
.byte 0b00000000 -- 0x7f      ↓
```

- minicube2.s (アセンブリ言語版・C言語版 共通)

```
#-----
#
# NEC Electronics      V850ES/Jx3-L シリーズ
#
#-----
# V850ES/JG3-L JF3-L サンプル・プログラム
#-----
# 割り込み編
#-----
# 【履歴】
# 2008.7.-- 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
```

--シリアル通信用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\_INT.dir (アセンブリ言語版・C言語版 共通)

```
# 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         : uPD70F3738 (C:¥Program Files¥NEC Electronics Tools¥DEV¥DF3738.800)
# Internal RAM   : 0x3ffb000 - 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 0x03F800{
    MonitorROM = $PROGBITS ?A MonitorROM;
};
```

内蔵ROMサイズが128 Kバイト  
製品の場合は 0x01F800

デフォルトのリンク・ディレクティブ  
ファイルとの差分 (追加コード)

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  
ファイルとの差分（追加コード）

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 (C言語版)

```
/*-----*/
/*
/*   NEC Electronics   V850ES/Jx3-L シリーズ
/*
/*-----*/
/*   V850ES/JG3-L サンプル・プログラム
/*-----*/
/*   割り込み編
/*-----*/
/* 【履歴】
/*   2008.7.--   新規作成
/*-----*/
/* 【概要】
/*   本サンプル・プログラムは、割り込み機能の使用例を示すものです。
/*   スイッチ入力の立ち下りエッジを検出して割り込みを発生させ、
/*   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。
/*   また、スイッチ入力に対して、10msのチャタリング除去時間を設けています。
/*
/*   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IOレジスタ名を記述可能にする
/*   ・チャタリング用10ms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・タイマの動作停止、クロック設定処理
/*   ・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化处理
/*   ・割り込み処理での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
/* 空きポート：P02,P04-P06, P10-P11, P30-P39, P50-P55, P70-P711, P90-P915,
/*             PCM0-PCM1, PCT0, PCT1, PCT4, PCT6, PDH0-PDH5, PDL0-PDL15
/*             空きの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 (0x6EE7) /* 10ms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・WDT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]  */

/*****
/*      メインモジュール      */
*****/
void main(void)
{
    extern unsigned int _S_romp;      /* ROM化シンボルの外部参照  */

    f_init();                          /* 初期化実施                */

    _rcopy( &_S_romp, -1 );          /* ROM化処理実施            */

    __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・ウエイト ,WDT2の停止 ,クロック設定処理 */

    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 = 0x01;                  /* 内蔵周辺へのバスウエイト設定 */

                                /* OCDMを「通常動作モード」に指定*/

    #pragma asm
    push r10
    mov 0x00, r10
    st.b r10, PRCMD
    st.b r10, OCDM
    pop r10
    #pragma endasm

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

                                /* 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

    PLLCTL = 0x03;              /* PLLモードに設定 */

    return;
}

```

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。

```

/*-----*/
/*   ポート/兼用機能の設定   */
/*-----*/
static void f_init_port_func( void )
{
    P0   = 0x00;          /* P02-P06を出力LOWに設定   */
    PM0  = 0x8B;
    PMC0 = 0x08;          /* P03をINTP0入力に設定   */

    P1   = 0x00;          /* P10,P11を出力LOWに設定   */
    PM1  = 0xFC;          /* V850ES/JF3-Lでの設定値は0xFE   */
                                /* V850ES/JF3-LではP10のみ   */
    P3   = 0x0000;        /* P30-P39を出力LOWに設定   */
    PM3  = 0xFC00;        /* V850ES/JF3-Lでの設定値は0xFCC0   */
                                /* V850ES/JF3-LではP30-P35,P38,P39   */
    PMC3 = 0x0000;

    #if(0) /* MINICUBE2使用時はP4をCSIB0として使用する為   */
        /* P4は空き端子として初期化しない(QB-V850ESJG3L-TB) */
        P4   = 0x00;          /* P40-P42を出力LOWに設定   */
        PM4  = 0xF8;
        PMC4 = 0x00;
    #endif

    P5   = 0x00;          /* P50-P55を出力LOWに設定   */
    PM5  = 0xC0;
    PMC5 = 0x00;

    P7H  = 0x00;          /* P70-P711を出力LOWに設定   */
    P7L  = 0x00;          /* V850ES/JF3-Lでは、レジスタが   */
                                /* 存在しないため、設定しない。   */
    PM7H = 0xF0;          /* V850ES/JF3-LではP70-P77   */
    PM7L = 0x00;

    P9   = 0x0000;        /* P90-P915を出力LOWに設定   */
    PM9  = 0x0000;        /* V850ES/JF3-Lでの設定値は0x1C3C   */
                                /* V850ES/JF3-LではP90,P91,P96-P99,P913-P915   */
    PMC9 = 0x0000;

    PCM  = 0x0C;          /* PCM0-PCM1を出力LOWに、PCM2-PCM3をLED消灯値に設定   */
    PMCM = 0xF0;
    PMCCM = 0x00;

    PCT  = 0x00;          /* PCT0,1,4,6を出力LOWに設定   */
    PMCT = 0xAC;
    PMCCT = 0x00;
}

```

```

PDH = 0x00; /* PDH0-PDH5を出力LOWに設定 */
PMDH = 0xC0;
PMCDH = 0x00;

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

/* 割込み機能設定 */
INTF0 = 0x08; /* INTP0立ち下りエッジ指定 */
INTR0 = 0x00; /* ↓ */
PIC0 = 0x07; /* INTP0 優先度7, マスク解除 */

return;
}

```

V850ES/JF3-Lでの設定値は0xFC

V850ES/JF3-LはPDH0,PDH1

```
/*
 * 割り込みモジュール
 */
__interrupt
void f_int_intp0( void )
{
    static unsigned int led_ptn_cnt = 0;
    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 ) /* チャタリング除去後もSW1認識 */
    {
        led_ptn_cnt++; /* 点灯パターンを変更(4種類) */
        if( led_ptn_cnt >= 4 )
        {
            led_ptn_cnt = 0;
        }

        PCM = LED_TBL[led_ptn_cnt]; /* 更新された点灯パターンを設定 */
    }

    PIC0 &= (unsigned char)~0x80; /* FailSafe: 複数要求をクリア */

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

## ● main.s (アセンブリ言語版)

```
#-----  
#  
# NEC Electronics      V850ES/Jx3-Lシリーズ  
#  
#-----  
# V850ES/JG3-L サンプル・プログラム  
#-----  
# 割り込み編  
#-----  
# 【履歴】  
# 2008.7.-- 新規作成  
#-----  
# 【概要】  
# 本サンプル・プログラムは、割り込み機能の使用例を示すものです。  
# スイッチ入力の立ち下りエッジを検出して割り込みを発生させ、  
# 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。  
# また、スイッチ入力に対して、10msのチャタリング除去時間を設けています。  
#  
# 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>  
# ・割り込みハンドラ処理の設定  
# ・チャタリング用10msウエイトのウエイト調整値を定義  
# ・レジスタ・アクセス・マクロ、モジュール・ジャンプ・マクロを定義  
#   (プログラムの可読性・保守性・移植性向上の為)  
# ・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・タイマの動作停止  
#   クロック設定処理  
# ・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  
# ・割り込み処理での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：P02-P06, P10-P11, P30-P39, P50-P55, P70-P711, P90-P915,
#             PCM0-PCM1, PCT0, PCT1, PCT4, PCT6, PDH0-PDH5, PDL0-PDL15
#             □未使用のポートは全て出力ポートに設定しておく
#
#-----#
#-----#
# 割り込みハンドラ処理の設定 #
#-----#
    .section    "INTP0", text
    jr        F_INT_UPDATE_LED_PATTERN

#-----#
# 10msウェイト調整用定数定義 #
#-----#
    .set      LIMIT_10ms_WAIT, 0x6EE7

```

```

#-----#
# 周辺I/Oレジスタ設定用マクロ #
#-----#

-- 【8bit周辺I/Oレジスタ・アクセス・マクロ】
    .macro    SET_REG8 val, reg
        mov    val, r6                -- テンポラリr6に値を入れておき
        st.b   r6, (reg)[zero]       -- 周辺I/Oにr6の値を設定する
    .endm

-- 【16bit周辺I/Oレジスタ・アクセス・マクロ】
    .macro    SET_REG16 val, reg
        mov    val, r6                -- テンポラリr6に値を入れておき
        st.h   r6, (reg)[zero]       -- 周辺I/Oにr6の値を設定する
    .endm

-- 【特定レジスタ・アクセス・マクロ】
    .macro    SET_SP_REG val, reg
        mov    val, r6
        st.b   r6, (PRCMD)[zero]     -- 事前にコマンドレジスタに設定
        st.b   r6, (reg)[zero]       -- 特定レジスタに同値を設定
    .endm

#-----#
#モジュール・ジャンプ・マクロ#
#-----#

-- 【モジュール・ジャンプ・マクロ】
    .macro    CALL label
        push   lp                    -- lpレジスタの値を退避
        jarl   label, lp             -- lpに戻り先PCを保存し, labelに遷移
        pop    lp                    -- 退避していたlpレジスタ値を復帰
    .endm

-- 【リターン・マクロ】
    .macro    RET
        jmp    [lp]                  -- 戻り先PCにジャンプ
    .endm

```



```

#-----#
# LEDパターン・テーブルの設定#
#-----#

        .const

T_LED_PATTERN:
        .byte    0x0C          -- LED表示パターン0 [OFF] [OFF]
        .byte    0x04          -- LED表示パターン1 [OFF] [ON]
        .byte    0x08          -- LED表示パターン2 [ON] [OFF]
        .byte    0x00          -- LED表示パターン3 [ON] [ON]

*****#
#       メイン・モジュール       #
*****#

        .text
        .align 4
        .globl  _main

_main:

        mov     0x00, r7      -- LED点灯パターン制御レジスタ値を初期化

        CALL    F_INIT       -- 初期化実施
                                -- 内蔵周辺I/Oへのバス・ウエイト, WDT停止, クロック設定処理
                                -- ポート/兼用機能の設定

        ei                 -- 割り込み許可

L_MAIN_LOOP:
        nop                -- ↑
                                -- 何もしない無限ループ(スイッチ入力による割り込み待ち)
        jr     L_MAIN_LOOP   -- ↓

```

```

#-----#
# 初期化モジュール #
#-----#
F_INIT:

    CALL    F_INIT_CLK_BUS_WDT2  -- 内蔵周辺I/Oへのバス・ウエイト, WDT停止, クロック設定処理

    CALL    F_INIT_PORT_FUNC     -- ポート/兼用機能の設定

    RET

#-----#
# クロック・バス・WDT2初期化 #
#-----#
F_INIT_CLK_BUS_WDT2:

    SET_REG8    0x01, VSWC        -- 内蔵周辺I/Oへのバスウエイト設定

    SET_SP_REG   0x00, OCDM       -- OCDMを通常動作モードに指定

    SET_REG8    0x01, RCM         -- 内蔵発振器停止
    SET_REG8    0x00, WDTM2      -- WDT2を停止

    SET_SP_REG   0x80, PCC        -- クロック分周をなしに設定

    SET_REG8    0x03, PLLCTL     -- PLLモードに設定

    RET

```

```
#-----#
# ポート/兼用機能の設定 #
#-----#
```

F\_INIT\_PORT\_FUNC:

```

SET_REG8 0x00, P0      -- P02-P06を出力LOWに設定
SET_REG8 0x8B, PM0     --
SET_REG8 0x08, PMC0    -- INTP0入力設定
SET_REG8 0x00, P1      -- P10,P11を出力LOWに設定
SET_REG8 0xFC, PM1     --
SET_REG16 0x0000, P3   -- ↑
SET_REG16 0xFC00, PM3  -- P30-P39を出力LOWに設定
SET_REG16 0x0000, PMC3 -- ↓
-- MINICUBE2使用時はP4をCSIB0として使用する為、P4は空き端子として初期化しない
-- (QB-V850ESJG3L-TB)
-- SET_REG8 0x00, P4      -- ↑
-- SET_REG8 0xF8, PM4     -- P40-P42を出力LOWに設定
-- SET_REG8 0x00, PMC4    -- ↓
SET_REG8 0x00, P5      -- ↑
SET_REG8 0xC0, PM5     -- P50-P55を出力LOWに設定
SET_REG8 0x00, PMC5    -- ↓
SET_REG8 0x00, P7H     -- ↑
SET_REG8 0x00, P7L     -- P70-P711を出力LOWに設定
SET_REG8 0xF0, PM7H    -- ↓
SET_REG8 0x00, PM7L    -- ↓
SET_REG16 0x0000, P9   -- ↑
SET_REG16 0x0000, PM9  -- P90-P915を出力LOWに設定
SET_REG16 0x0000, PMC9 -- ↓
SET_REG8 0x0C, PCM     -- PCM0-PCM1を出力LOWに、PCM2-PCM3をLED消灯値に設定
SET_REG8 0xF0, PMCM    -- PCM端子を出力ポートに設定
SET_REG8 0x00, PMCCM   -- ↓
SET_REG8 0x00, PCT     -- ↑
SET_REG8 0xAC, PMCT    -- PCT0,1,4,6を出力LOWに設定
SET_REG8 0x00, PMCCT   -- ↓

```

V850ES/JF3-LではP10のみ

V850ES/JF3-Lでの設定値は0xFE

V850ES/JF3-LではP30-P35,P38,P39

V850ES/JF3-Lでの設定値は0xFCC0

V850ES/JF3-Lでは、レジスタが存在しないため、設定しない。

V850ES/JF3-LではP70-P77

V850ES/JF3-LではP90, P91, P96-P99, P913-P915

V850ES/JF3-Lでの設定値は0x1C3C

SET_REG8	0x00, PDH	-- ↑	
SET_REG8	0xC0, PMDH	--	PDH0-PDH5を出力LOWに設定
SET_REG8	0x00, PMCDH	--	V850ES/JF3-Lでの設定値は0xFC
SET_REG16	0x0000, PDL	-- ↑	
SET_REG16	0x0000, PMDL	--	PDL0-PDL15を出力LOWに設定
SET_REG16	0x0000, PMCDL	-- ↓	

-- 割り込み機能設定

SET_REG8	0x08, INTF0	--	INTP0立ち下りエッジ指定
SET_REG8	0x00, INTRO	-- ↓	
SET_REG8	0x07, PIC0	--	優先度7, マスク解除

RET

```
#####  
#      割り込みモジュール      #  
#####  
F_INT_UPDATE_LED_PATTERN:  
  
    mov        LIMIT_10ms_WAIT, r6        -- チャタリング時間初期値を設定  
  
L_KEY_CHAT:  
    sub        1, r6                      -- チャタリングを除去する為,  
    nop                          -- 10ms経過する迄, 何もせずループ  
    jnz        L_KEY_CHAT                -- ↓  
  
    ld.b       (P0)[zero], r6            -- 10ms経過時にも, SW1入力が押下  
    and        0x08, r6                  -- されているか確認する  
    jnz        J_INT_P03_END              -- ↓  
  
    add        1, r7                      -- 4種類あるLED出力パターン番号を  
    and        3, r7                      -- 次のパターンに更新する  
  
    ld.b       (!T_LED_PATTERN)[r7], r6  
    st.b       r6, (PCM)[zero]           -- LED点灯パターンを設定する  
  
J_INT_P03_END:  
  
    clr1       7, (PIC0)[zero]           -- チャタリングで複数回受信した可能性のある, 割込要求をクリアしておく  
  
    reti
```

## 【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

---

お問い合わせ先

## 【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

## 【営業関係、技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : [info@necel.com](mailto:info@necel.com)

## 【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか、NECエレクトロニクスの販売特約店へお申し付けください。