

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 (ウォッチドッグ・タイマ2 (WDT2)) 暴走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、およびウォッチドッグ・タイマ2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、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ます。WDT2によるオーバフローが発生しない場合、SW1がオンの間、LED1が約55 ms周期で点灯と消灯を切り替えます。SW1をオフにすると、LED1は約120 ms周期で点灯と消灯を切り替えます。オーバフローが発生した場合、WDT2によるリセット信号が発生し、リセット解除後にLED2が点灯し、LED1は約120 msごとに点灯と消灯を切り替えます。

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 ウォッチドッグ・タイマ2 (WDT2) 動作 ... 5 | |
| 1.3 メイン処理 ... 5 | |
| 1.4 割り込み処理 ... 6 | |
| 1.5 ウォッチドッグ・タイマ2 (WDT2) オーバフロー発生 の例 ... 7 | |
| 第2章 回路図 ... 8 | |
| 2.1 回路図 ... 8 | |
| 2.2 周辺ハードウェア ... 9 | |
| 第3章 ソフトウェアについて ... 10 | |
| 3.1 ファイル構成 ... 10 | |
| 3.2 使用する内蔵周辺機能 ... 11 | |
| 3.3 初期設定と動作概要 ... 11 | |
| 3.4 フローチャート ... 13 | |
| 3.5 V850ES/JJ3とV850ES/JG3の違い ... 15 | |
| 3.6 セキュリティIDについて ... 15 | |
| 3.7 MINICUBE2使用時のサンプル・プログラム動作上 の注意... 15 | |
| 第4章 レジスタ設定について ... 16 | |
| 4.1 ウォッチドッグ・タイマ2 (WDT2) の設定 ... 17 | |
| 4.2 WDT2リセット検出の確認 ... 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 U20020JJ1V0AN00

発行年月 September 2009 NS

© NEC Electronics Corporation 2009

- ・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。

注2. 本事項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、製造製品をいう。

(M8E0909J)

第1章 概 要

このサンプル・プログラムでは、ウォッチドッグ・タイマ2 (WDT2) の使用例を示しています。

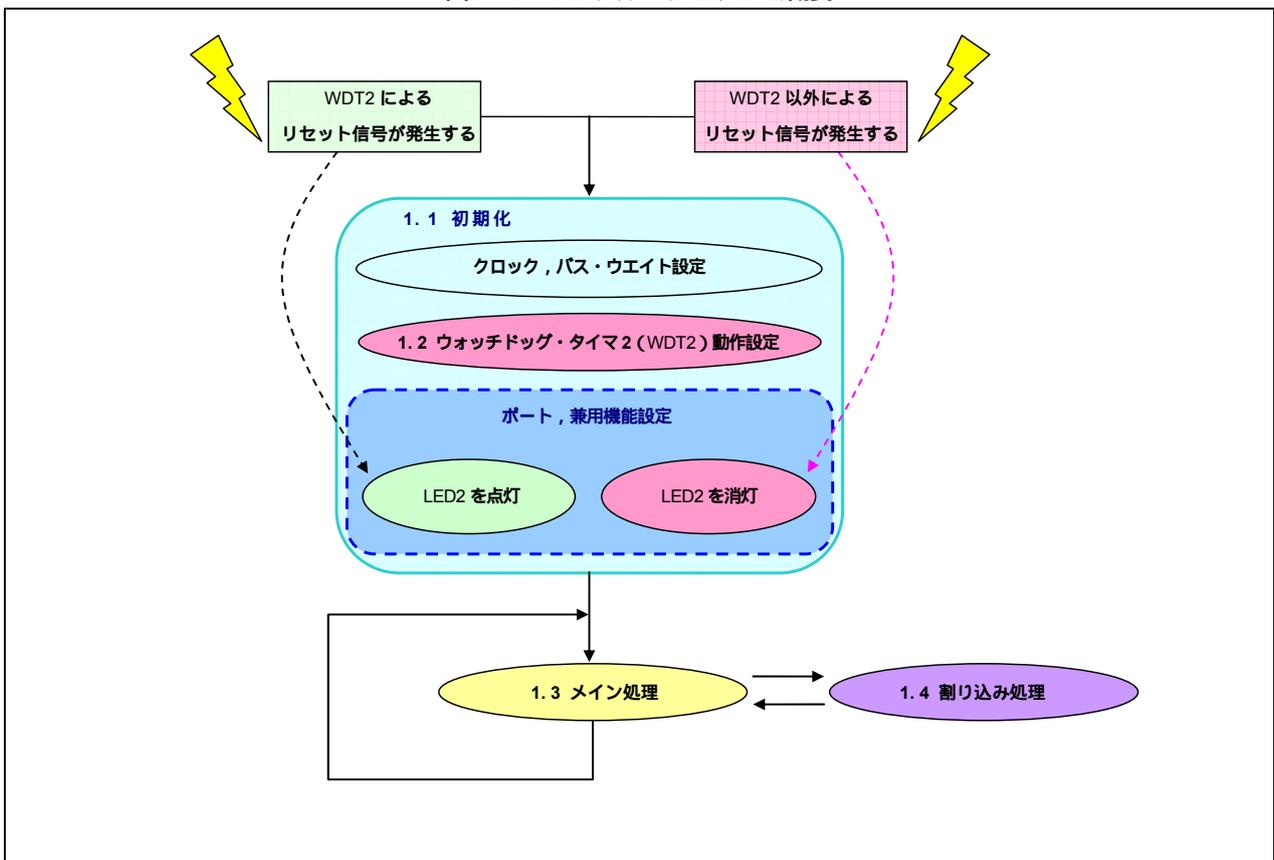
WDT2のオーバーフロー時間は、サブクロックで125 msに設定し、オーバーフロー発生で内部リセット (WDT2リセット) 信号を発生します。

初期設定完了後は、スイッチ入力がないときは120 ms周期でLED1の点灯と消灯を切り替え、同時にWDT2のカウンタをクリアしてオーバーフロー発生を防ぎます。またスイッチ入力があるとき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、スイッチ押下中は55 ms周期でLED1の点灯と消灯を切り替えてWDT2のカウンタをクリアします。WDT2によるリセットが発生した場合には、初期設定にてLED2を点灯させます。

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。

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・アクセスのウエイト<ウエイト：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をリセット・モードに設定し、サブクロック使用設定
- ・内部システム・クロックの設定
- ・PLLモードに設定し $4 \text{ MHz} \times 8 \text{ 通倍} = 32 \text{ MHz}$ 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

< 16ビット・インターバル・タイマM（TMM0）設定 >

- ・カウント・クロック（ $f_{xx}/512$ ）選択
- ・TMM0カウント設定（120 ms）
- ・タイマMの割り込みをマスク
- ・TMM0動作許可

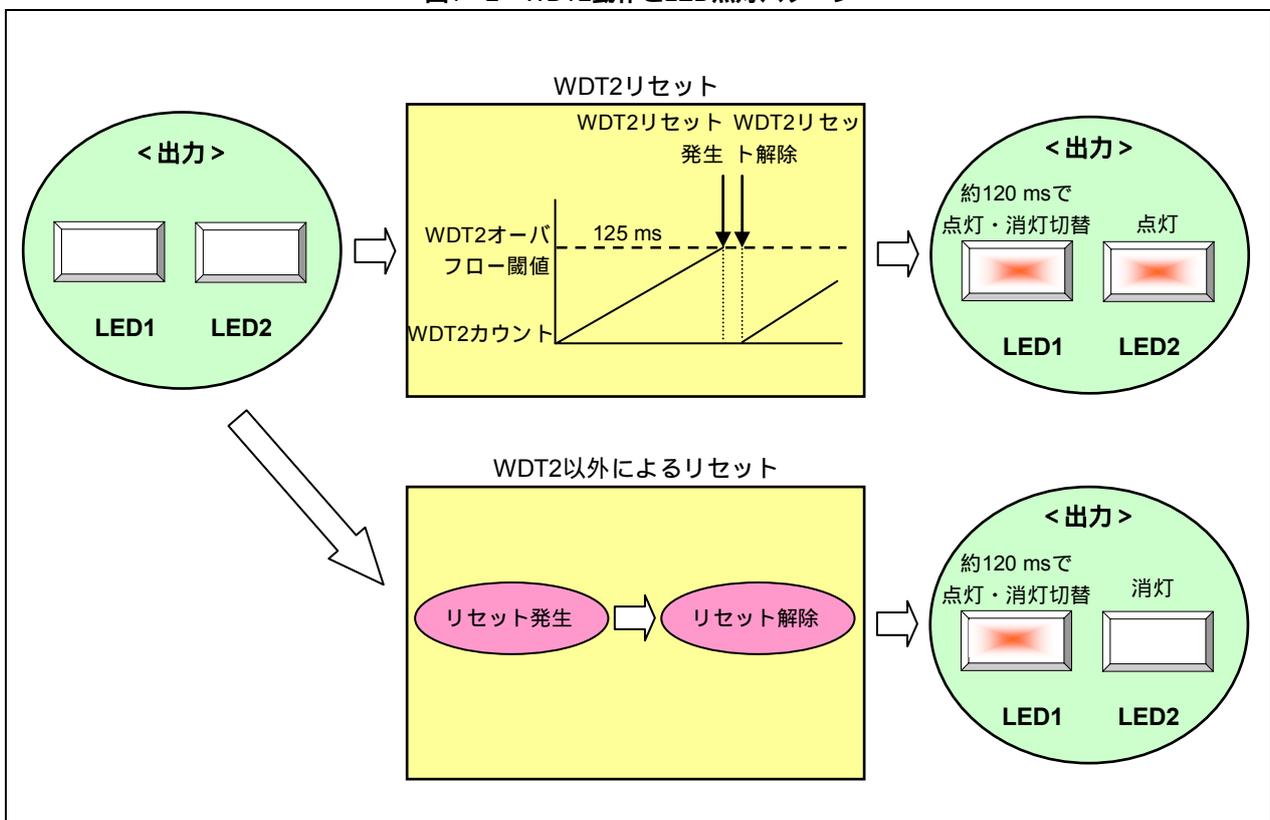
1.2 ウォッチドッグ・タイマ2 (WDT2) 動作

ウォッチドッグ・タイマ2 (WDT2) は設定された動作クロックによるカウントを行い、カウンタのオーバフロー時の動作として、内部リセット (WDT2リセット) と割り込みのどちらかを発生させます[※]。

このサンプル・プログラムではオーバフロー時の動作として内部リセットを利用しています。具体的な動作としては、WDT2以外のリセット時は、LED2は消灯状態となりLED1は約120 msごとに点灯と消灯が切り替わります。またWDT2リセット時は、LED2が点灯状態となりLED1は約120 msごとに点灯と消灯が切り替わります。

注 ウォッチドッグ・タイマ2 (WDT2) は、リセット解除後に自動的にリセット・モードでスタートします。ウォッチドッグ・タイマ2を使用しない場合は、この機能によるリセットが発生する前に停止するか、一度ウォッチドッグ・タイマ2をクリアし、次のインターバル時間内で停止してください。

図1-2 WDT2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
- ・無限ループを実行 (TMM0により約120 msごとにLED1の点灯 / 消灯の切り替えとWDT2カウント・クリアを行う。)

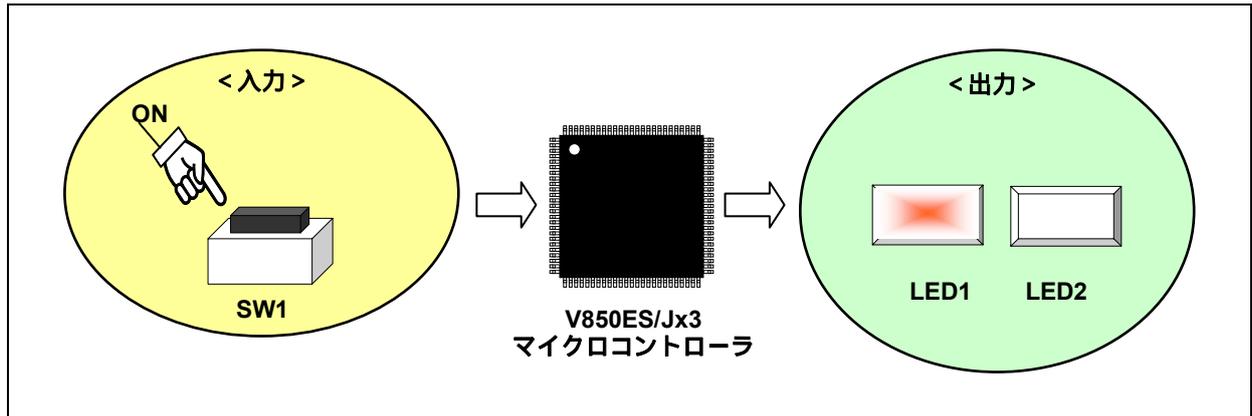
1.4 割り込み処理

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、LED1の点滅時間を変化させます。

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オフである場合は、チャタリング・ノイズであると判定し、LED1の点滅周期を変化させません。

図1-3 スイッチ入力とLED点灯パターン



スイッチ入力なし LED1は約120 msごとに点灯と消灯を切り替える

スイッチ入力中 LED1は約55 msごとに点灯と消灯を切り替える

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



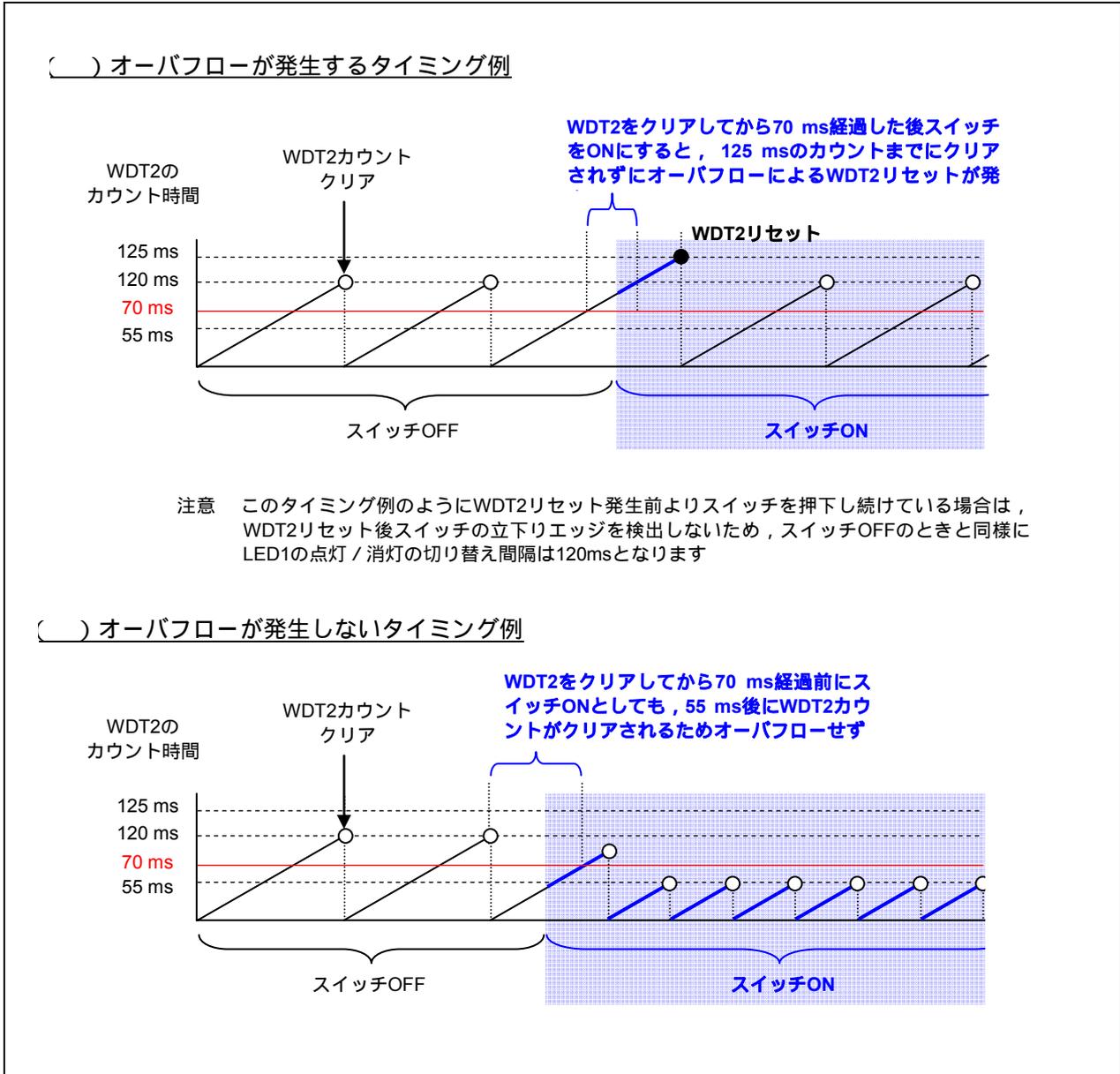
【コラム】チャタリングとは

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。

1.5 ウォッチドッグ・タイマ2 (WDT2) オーバフロー発生例

このサンプル・プログラムにおいて、ウォッチドッグ・タイマ2 (WDT2) がオーバフローし、内部リセットが発生するタイミング例について以下に示します。

図1-4 WDT2オーバフローの発生タイミング例



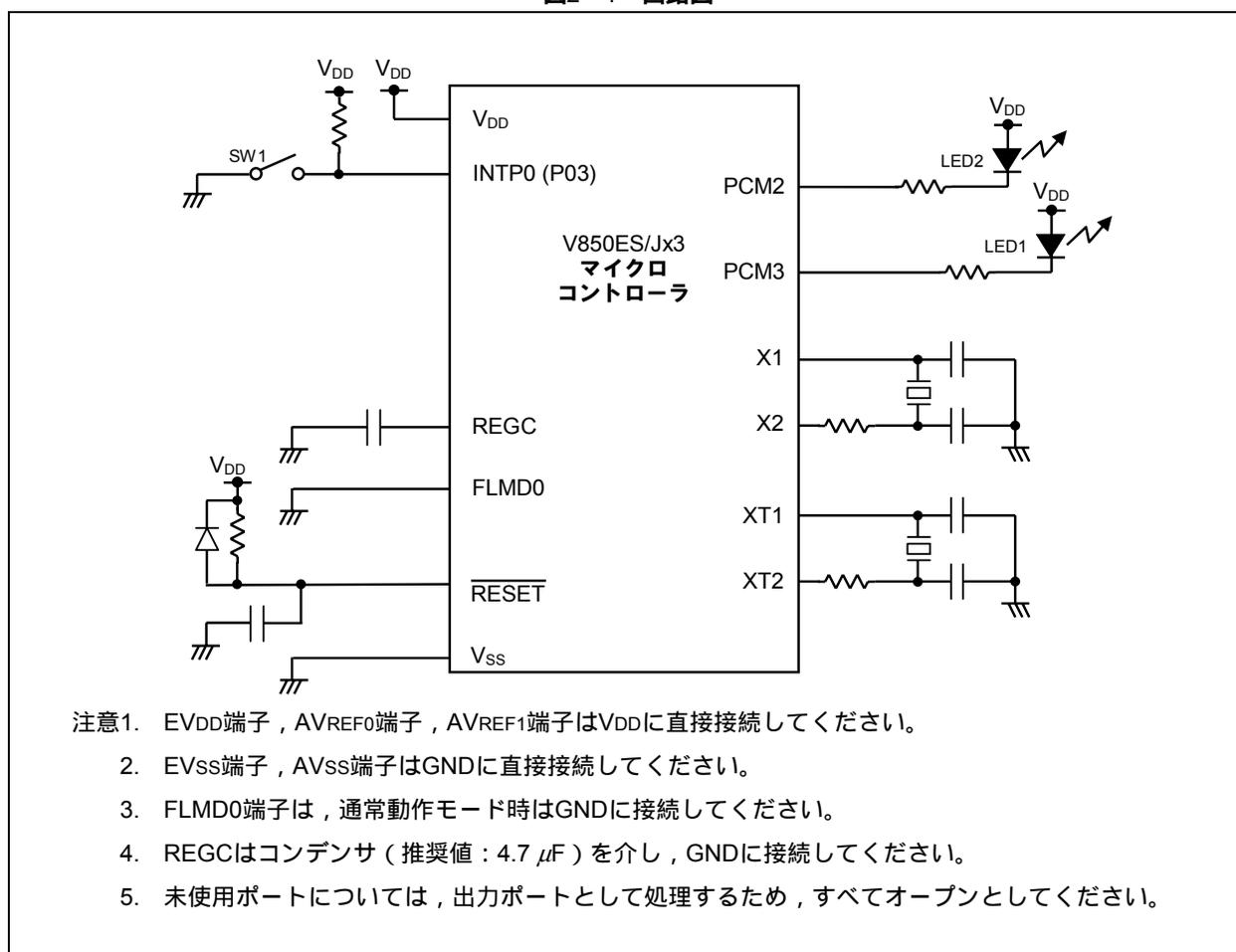
第2章 回路図

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)

LED1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、LED1を使用します。

ウォッチドッグ・タイマ2によるリセット信号が発生した場合、リセット解除後LED2が点灯します。

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"> • conf <ul style="list-style-type: none"> — crtE.s — AppNote_WDT2.dir — AppNote_WDT2.prj — AppNote_WDT2.prw • src <ul style="list-style-type: none"> — main.c — minicube2.s | スタート・アップ・ルーチン・ファイル ^{注1} | - | |
| | リンク・ディレクティブ・ファイル ^{注2} | | |
| | 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します(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します(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。

- ・ウォッチドッグ・タイマ2 (オーバーフローが発生) : WDT2
- ・16ビット・インターバル・タイマM (LED点滅周期生成) : TMM0
- ・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)

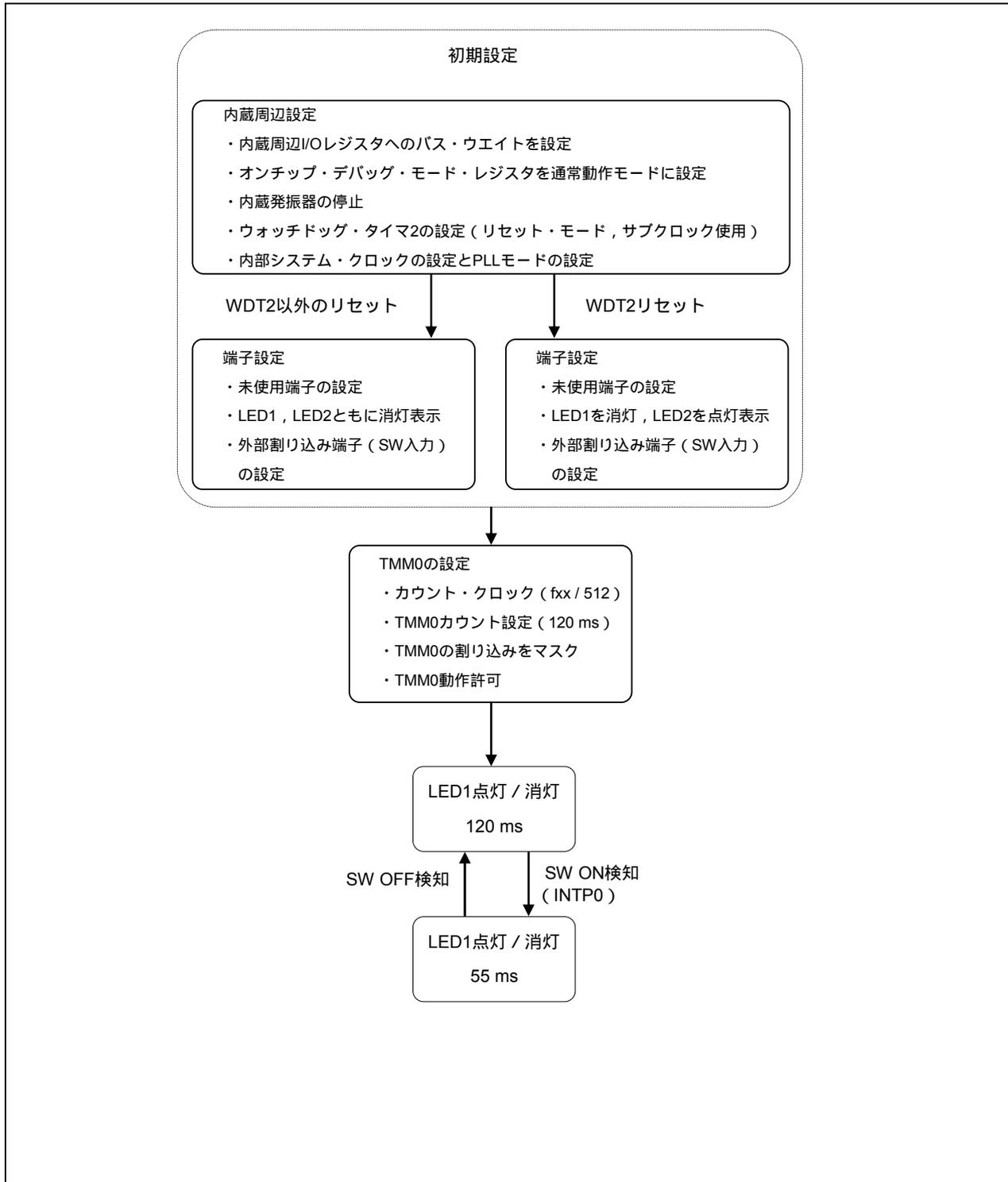
3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定として、クロック周波数の選択や、WDT2の設定、入出力ポートや外部割り込みの端子設定、TMMカウント・クロックの設定、割り込みの設定などを行います。

初期設定完了後は、LED1は約120 msごとに点灯と消灯を繰り返します。またスイッチ入力 (SW1) の立ち下りエッジを検出して割り込み処理を行い、スイッチがONの間はLED1は約55 msごとに点灯と消灯を繰り返します。スイッチをOFFにすると、LED1が約120 msごとに点灯と消灯を繰り返します。また割り込み処理中に、ウォッチドッグ・タイマ2によるオーバーフローが発生した場合は、ウォッチドッグ・タイマ2要因のリセット信号が発生します。リセット解除後、LED2が点灯し、LED1は約120 msごとに点灯と消灯を繰り返します。

次の状態遷移図に初期設定と動作概要について示します。

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)

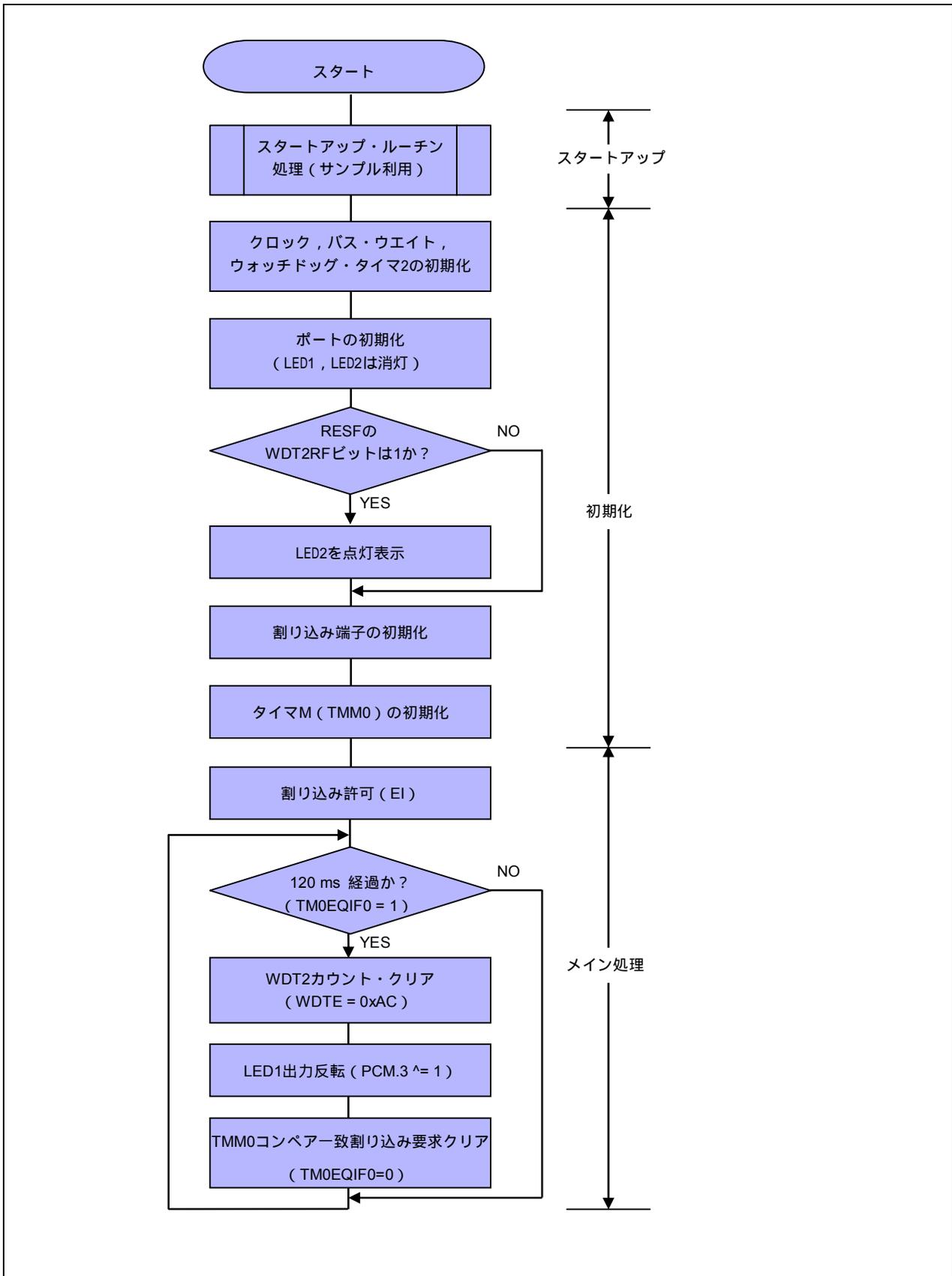
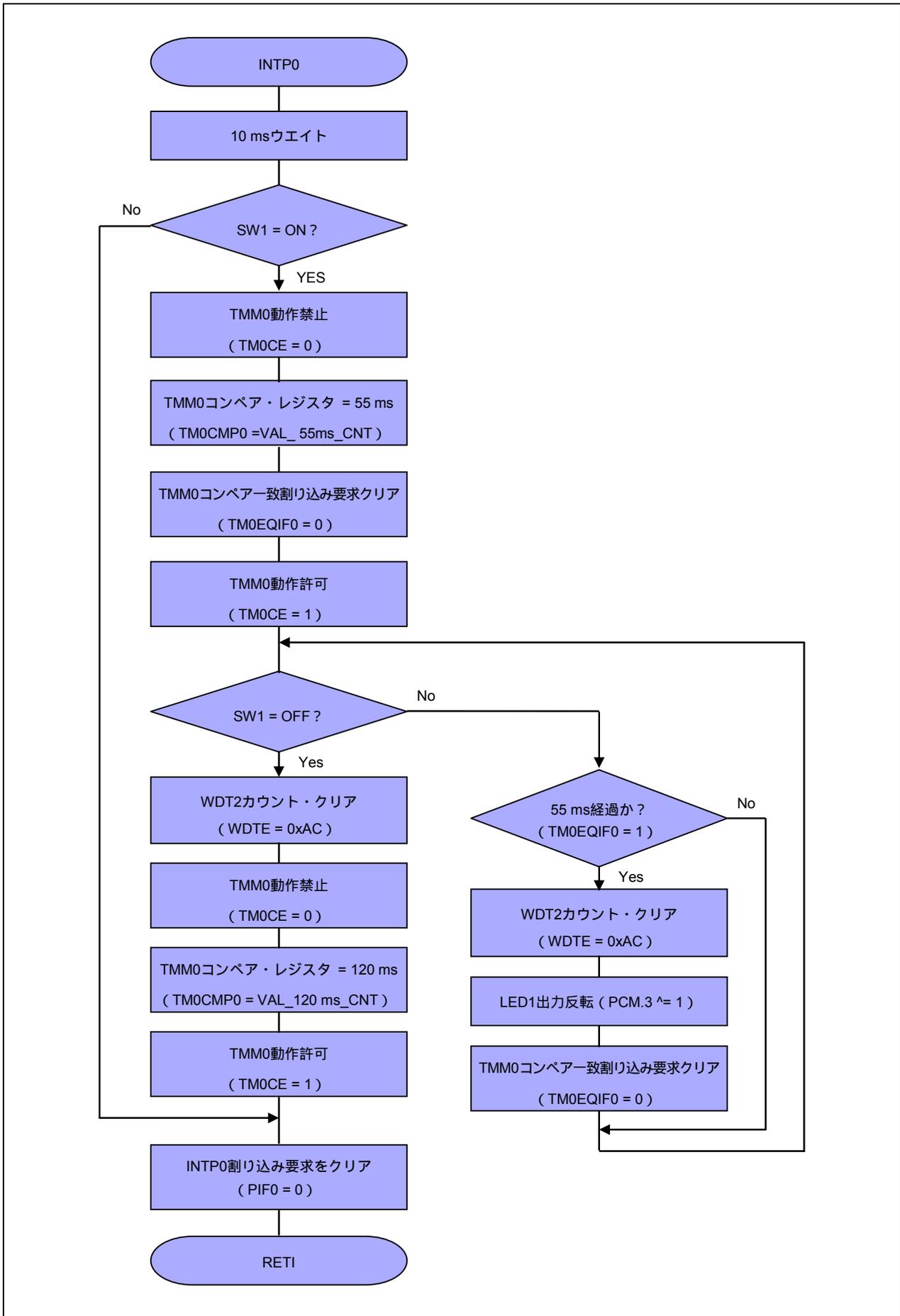


図3-2 フロー・チャート (2/2)



**【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、次のことを行います。

- ・ 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
- ・ エレメント・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関数の引き数に設定
- ・ main関数へ分岐する

3.5 V850ES/JJ3とV850ES/JG3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**を参照してください。

3.6 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 アプリケーション・ノート 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」を参照してください。

3.7 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、本サンプル・プログラムを動作させた場合、ウォッチドッグ・タイマ2 (WDT2) は動作しません。これはMINICUBE2を接続してデバッグを起動したときにデバッグ用モニター・プログラム内にてWDT2を動作停止とする処理を行っているためです。従って、MINICUBE2を使用したオンチップ・デバッグによる本サンプル・プログラムの動作確認は行えません。

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 統合デバッグ 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、ウォッチドッグ・タイマ2 (WDT2) 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 アプリケーション・ノート サンプル・プログラム (初期設定) LED点灯のスイッチ制御編」を、割り込みについては、「V850ES/Jx3 アプリケーション・ノート 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」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 ウォッチドッグ・タイマ2 (WDT2) の設定

ウォッチドッグ・タイマ2には、次の2種類の動作モードがあります。

- ・リセットとして使用 【例1】 参照
- ・ノンマスクابل割り込み^注として使用 【例2】 参照

注 ノンマスクابل割り込みとは、CPUが割り込み禁止状態でも無条件に受け付けられる割り込みを指します。また、割り込み優先順位の対象にならず、すべての割り込み要求信号に対して最優先されます。

ウォッチドッグ・タイマ2は、おもに次の2種類のレジスタで制御します。

- ・ウォッチドッグ・タイマ・モード・レジスタ2 (WDTM2)
- ・ウォッチドッグ・タイマ・イネーブル・レジスタ (WDTE)

4.1.1 ウォッチドッグ・タイマ2動作モードの設定

ウォッチドッグ・タイマ2の動作モード、オーバフロー時間および動作クロックを設定するレジスタです。

WDTM2は8ビット単位でリード/ライト可能です。ただし、リードは何回でもできますが、ライトはリセット解除後に1回のみできます。

リセットにより0x67になります。

注意 次に示す状態において、WDTM2 レジスタへのアクセスは禁止です。

- ・CPU がサブクロックで動作し、かつメイン・クロック発振を停止している場合
- ・CPU が内蔵発振クロックで動作している場合

図4 - 1 WDTM2レジスタのフォーマット

| | | | | | | | |
|---|----------------|----------------------------------|--------|--------|---|--------|--------|
| ウォッチドッグ・タイマ・モード・レジスタ2 (WDTM2) | | | | | | | |
| アドレス : 0xFFFFF6D0 | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | WDM21 | WDM20 | WDCS24 | WDCS23 | WDCS22 | WDCS21 | WDCS20 |
| WDM21 | WDM20 | ウォッチドッグ・タイマ2の動作モードの選択 | | | | | |
| 0 | 0 | 動作停止 | | | | | |
| 0 | 1 | ノンマスクابل割り込み要求モード (INTWDT2信号を発生) | | | | | |
| 1 | x ^注 | リセット・モード (WDT2RES信号を発生) | | | | | |
| WDCS24 | WDCS23 | WDCS22 | WDCS21 | WDCS20 | オーバフロー時間と動作クロックの選択 | | |
| 1 | x ^注 | 0 | 1 | 1 | $2^{12}/f_{XT} (125 \text{ ms})$, f_{XT} | | |
| 注 任意 | | | | | | | |
| 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 (WDTM2 = 0x53)。 | | | | | | | |
| ・ $f_{XT} = 32.768 \text{ kHz}$ | | | | | | | |


【コラム】WDT2動作クロックにサブクロックを選択する場合

通常、電源投入後のリセット解除後にWDT2は自動的に動作開始するため、WDTM2レジスタの設定はプログラムの初期の段階で行います。しかし、サブクロック動作を選択した場合には、発振安定までに時間がかかるため安定しない状態でWDT2の動作を開始してしまう恐れがあります。これを避けるためには、WDTM2レジスタでサブクロック動作を選択する前に、適切な発振安定時間だけウェイトを行ってください。また、このウェイトの間にもWDT2は動作しているため、オーバフロー前に適宜カウンタのクリアを行ってください。

4.1.2 ウォッチドッグ・タイマ・イネーブル・レジスタ (WDTE)

WDTEレジスタに“0xAC”を書き込むことにより、ウォッチドッグ・タイマ2のカウンタがクリアされ、再びカウントを開始します。

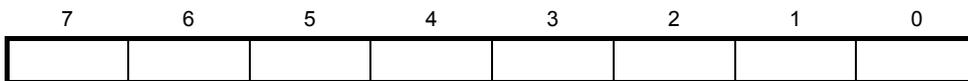
8ビット単位でリード/ライト可能です（1ビット・メモリ操作命令を実行した場合、強制的にオーバフロー信号を発生します）。

リセットにより0x9Aになります。

図4-2 WDTEレジスタのフォーマット

ウォッチドッグ・タイマ・イネーブル・レジスタ (WDTE)

アドレス: 0xFFFFF6D1



- 注意**
1. WDTEレジスタに“0xAC”以外の値を書き込んだ場合、強制的にオーバフロー信号を発生します。
 2. 意図的にオーバフロー信号を発生させたい場合は、WDTEレジスタに“0xAC”以外の値を1回書き込むか、WDTM2レジスタに2回書き込んでください。ただし、ウォッチドッグ・タイマ2を動作停止に設定した場合は、WDTEレジスタに“0xAC”以外の値を1回書き込む、またはWDTM2レジスタに2回書き込んでもオーバフロー信号は発生しません。
 3. WDTEレジスタのリード値は、“0x9A”（書き込んだ値（“0xAC”）とは異なる値）になります。

【例1】ウォッチドッグ・タイマ2オーバーフロー検出をリセットとして使用する場合

(サンプル・プログラムと同じ使用法)

ここではサブクロックを動作クロックとして、カウント開始から125 ms後にリセットが発生するように設定を行います。また125 ms経過前にカウンタのクリアを行い、リセットの発生を防止するように設定します。このように設定をしておくことで、CPU動作に何らかの異常が発生し、カウンタのクリアが行えなくなった際にリセットを発生するため、CPU動作を初期化することが可能です。

・設定手順

WDTM2 = 0x53 (リセット・モード, 動作クロック f_{XT} , オーバーフロー時間125 ms) を設定します。

オーバーフロー検出時間を越えるまでにWDTE = 0xAC (ウォッチドッグ・タイマ2のカウント・クリア) を設定します。

・プログラム例

```

    WDT2以外の初期化処理は省略しています
}

/* WDT2のオーバーフロー検出をリセットとして使用 */
WDTM2 = 0x53;                               /* ウォッチドッグ・タイマ2動作開始 */

if ( WDT2オーバーフロー前に必ず真となる条件 )
{
    WDTE = 0xAC;                             /* WDT2カウントをクリア */
}

```

【例2】ウォッチドッグ・タイマ2 オーバフロー検出を割り込みとして使用する場合

ここではサブクロックを動作クロックとして、カウント開始から125 ms後に割り込み (INTWDT2) が発生するように設定を行います。また125 ms経過前にカウンタのクリアを行い、割り込みの発生を防止するように設定します。

・設定手順

INTWDT2の割り込みハンドラの指定と割り込み関数のプロトタイプ宣言をします。

WDTM2 = 0x33 (ノンマスカブル割り込み要求モード, 動作クロック f_{XT} , オーバフロー時間125 ms) を設定します。

オーバフロー検出時間を越えるまでにWDTE = 0xAC (ウォッチドッグ・タイマ2のカウント・クリア) を設定します。

INTWDT2の割り込み関数の定義を行います。

・プログラム例1

```
#pragma interrupt INTWDT2 f_int_intwtdt2 /* 割り込みハンドラ指定 */
static void f_int_intwtdt2( void ); /* INTWDT2割り込み関数 */

/* WDT2のオーバフロー検出を割り込みとして使用 */
WDTM2 = 0x33; /* ウォッチドッグ・タイマ2動作開始 */

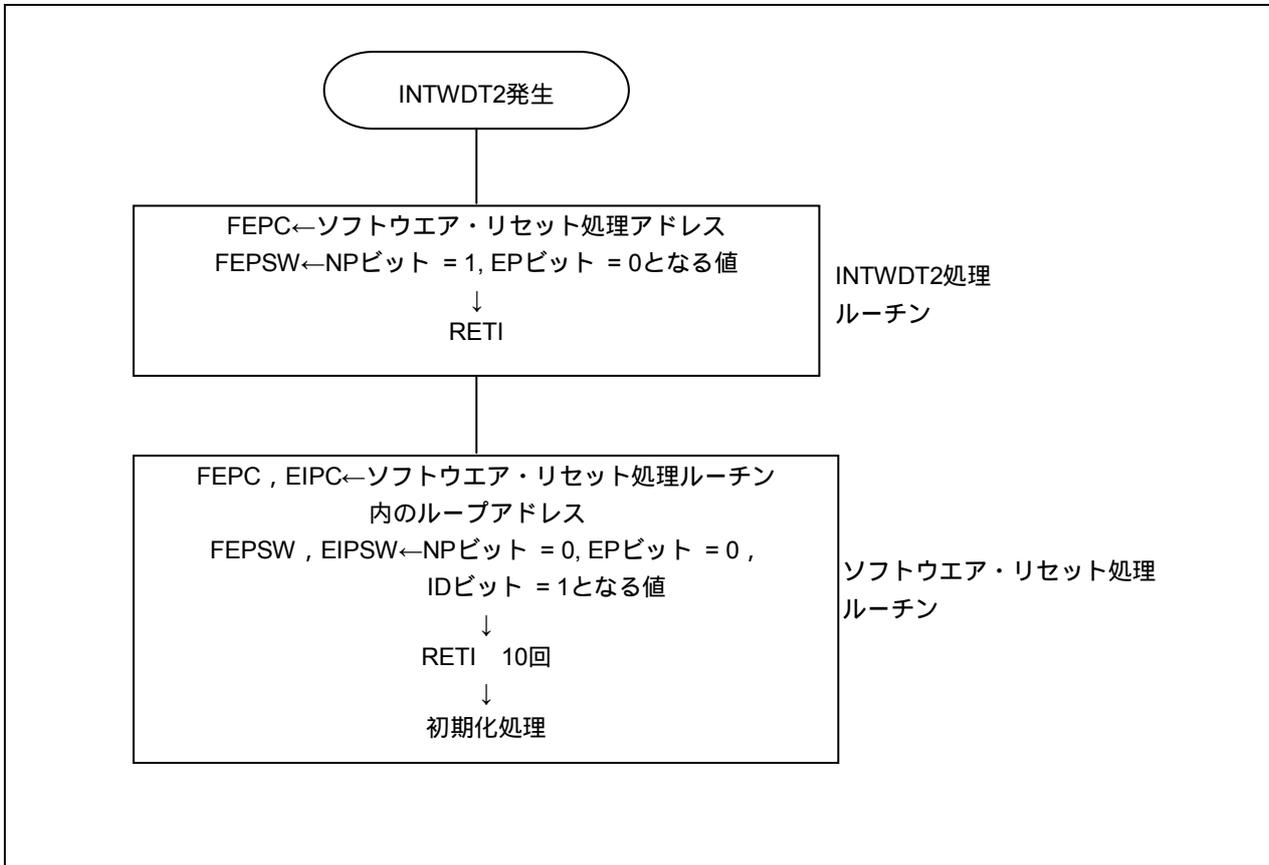
if ( WDT2オーバフロー前に必ず真となる条件 )
{
    WDTE = 0xAC; /* WDT2カウントをクリア */
}

__interrupt
void f_int_intwtdt2( void )
{
    割り込み処理
}
```

ただし、ウォッチドッグ・タイマ2のオーバフローが発生した場合、CPUは暴走状態になっている可能性が高いため、リセット・モード (WDM21ビット = 1) で、CPUがリセットされるように設定することを推奨します。ノンマスカブル割り込み要求モード (WDM21, WDM20ビット = 01) を選択する場合には、INTWDT2割り込み処理でシステムのエラー処理を行ったあと、システムを停止させてください。INTWDT2割り込み処理によるノンマスカブル割り込み処理実行後に、RETI命令による復帰はできません。

INTWDT2割り込み処理から復帰するには、次に示すソフトウェア・リセット処理を実行してください。
 ただし、ソフトウェア・リセット処理では、リセット解除後に一度しか設定できないレジスタ（WDTM2
 など）を再設定することはできません。これらのレジスタを初期状態に戻すためには、リセット端子入力
 などのハードウェア・リセットが必要になります。

図4 - 3 ソフトウェア・リセット処理



・プログラム例2

```
#####  
#          割り込み初期処理          #  
#####  
        .global  _initint  
  
_initint:  
        mov     initloop1,r6      -- 処理ルーチン・アドレスをセット  
        ldsr   r6,2              -- FEPCにセット  
        mov     0xa0,r6          -- NP=1, EP=0  
        ldsr   r6,3              -- FEPSWにセット  
        reti                   -- NMIから復帰してinitloop1へ  
  
initloop1:  
        mov     0x20,r6          -- 次のRETIでPSW.NPを0に  
        ldsr   r6,3              -- FEPSWにセット  
        ldsr   r6,1              -- EIPSWにもセット  
        mov     initloop2,r6     -- 処理ルーチン・アドレス2をセット  
        ldsr   r6,2              -- FEPCにセット  
        ldsr   r6,0              -- EIPCにセット  
        mov     0x0b,r6          -- カウント初期値(10回)  
        reti  
  
initloop2:  
        sub    1,r6              -- 回数カウンタ(-1)  
        bz     initend          -- 割り込み初期化完了  
        reti  
  
initend:
```

4.2 WDT2リセット検出の確認

ウォッチドッグ・タイマ2のオーバフローによるリセット時には、リセット要因フラグ・レジスタ (RESF) の WDT2RF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、WDT2RFビットがクリアされます。そのため、リセット解除後にWDT2RFビットを確認することにより、リセット要因がWDT2リセットであったのか、その他の要因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。

$\overline{\text{RESET}}$ 端子入力により0x00になります。 $\overline{\text{RESET}}$ 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 | | | | | |
| 0 | | 発生なし | | | | | |
| 1 | | 発生あり | | | | | |

注意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。

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。

【リセット要因フラグのクリアについて】

上記注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、ウォッチドッグ・タイマ2 (WDT2) 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 マイクロコントローラ
#
#-----
#     V850ES/JJ3  JG3   サンプル・プログラム
#-----
#     暴走検出時リセット発生編
#-----
# 【履歴】
#     2009.09.--   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 マイクロコントローラ
/*
/*-----*/
/*      V850ES/JJ3 サンプル・プログラム
/*-----*/
/*      暴走検出時リセット発生編
/*-----*/
/* 【履歴】
/*      2009.09.--   新規作成
/*-----*/
/* 【概要】
/*      本サンプル・プログラムは、ウォッチドッグ・タイマ2 (WDT2) の使用例を示すものです。
/*      WDT2のオーバフロー時間を125msに設定し、オーバフロー発生で内部リセット
/*      (WDT2リセット) 信号を発生します。
/*      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、
/*      スイッチ押下中は55ms周期でLED1の点滅とWDT2カウント・クリアを行います。
/*      WDT2以外によるリセットが発生した場合には、初期設定にてLED2を点灯させます。
/*
/*
/*      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
/* ・チャタリング対策用10msウエイトのウエイト調整値を定義
/* ・LED1点滅時間(120ms, 55ms)のTMM0への設定値を定義
/* ・プロトタイプ宣言を実施
/* ・内蔵周辺I/Oへのバス・ウエイト, WDT2の動作開始, 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
/* ・TMM0コンペアー一致割り込み要求フラグを監視し、割り込み要求発生時、
/*   WDT2カウント・クリア, LED1出力反転, TMM0コンペアー一致割り込み要求フラグクリアを行う
/* <割り込み処理>
/* ・LED1点滅周期を55msに設定
/* ・TMM0コンペアー一致割り込み要求フラグを監視し、割り込み要求発生時、
/*   WDT2カウント・クリア, LED1出力反転, TMM0コンペアー一致割り込み要求フラグクリアを行う
/* ・スイッチOFF時, WDT2カウント・クリア, LED1点滅周期を120msに設定
/*   (スイッチ入力時のチャタリング除去時間10ms)
```

```

/*
/*
/* 【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      (40280)    /* 10msウエイトの調整用定数定義      */
#define LIMIT_250ms_WAIT    (11360)    /* 250msウエイトの調整用定数定義     */
#define VAL_55ms_CNT        (3437)     /* LED1点滅周期 (55ms)              */
#define VAL_120ms_CNT       (7499)     /* LED1点滅周期 (120ms)             */

/*-----*/
/*          プロトタイプ宣言          */
/*-----*/
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_port_func( void ); /* ポート/兼用機能初期化関数       */
static void f_init_int_tmm( void );  /* TMM0初期化関数                   */

```

```

/*****
/*      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) */
    {
        if ( TMOEQIF0 == 1 ) /* INTTMOEQ0割り込み要求信号あり? */
        {
            WDTE = 0xAC;    /* WDT2カウントをクリア          */
            PCM.3 ^= 1;     /* LED1出力反転                  */
            TMOEQIF0 = 0;   /* TMM0コンペア一致割り込み要求クリア*/
        }
    }

    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, クロック設定処理          */

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

    f_init_int_tmm();     /* TMM0タイマ設定                  */

    return;
}

```

```

/*-----*/
/* クロック , バス・ウエイト , WDT2初期化      */
/*-----*/
static void f_init_clk_bus_wdt2( void )
{
    unsigned int loop_wait;                /* forループ用カウンタ      */

    VSWC = 0x11;                          /* 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
    st.b    r0, PRCMD
    st.b    r0, OCDM
#pragma endasm

    RSTOP = 1;                            /* 内蔵発振器を停止        */

    /* サブクロック発振安定時間を250msとしたときのウエイト処理      */
    /* 注意：厳密な発振安定時間は発振評価に基づき設定してください    */
    for( loop_wait = 0 ; loop_wait < LIMIT_250ms_WAIT ; loop_wait++ )
    {
        WDTE = 0xAC;                      /* WDT2カウントをクリア    */
    }

    WDTM2 = 0x53;                         /* WDT2モード設定          */

    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         */

```

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。

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。

/* 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;

}

/*-----*/

/* 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;
PMC6 = 0x0000;
```

V850ES/JG3では設定不要

```
P7H = 0x00; /* P70-P715を出力LOWに設定 */
P7L = 0x00;
PM7H = 0x00;
PM7L = 0x00;
```

V850ES/JG3ではP70-P711のみ設定
(PM7H = 0xF0;)

```
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 = 0x0C; /* PCM0-PCM1を出力LOWに ,
PCM2-PCM3に消灯設定 */
```

```
if( RESF.4 == 1 )
```

```
{
    PCM = 0x08; /* リセット要因がWDT2の時はLED2を点灯設定 */
}
```

```
PMCM = 0xC0;
PMCCM = 0x00;
```

V850ES/JG3ではPCM0-PCM3のみ設定
(PMCM = 0xF0;)

```
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;
PMCCCT = 0x00;
```

V850ES/JG3ではPCT0, PCT1, PCT4, PCT6のみ設定
(PMCT = 0xAC;)

```

PDH   = 0x00;          /* PDH0-PDH7を出力LOWに設定 */
PMDH  = 0x00;          /* PDH0-PDH7を出力LOWに設定 */
PMCDH = 0x00;          /* PDH0-PDH7を出力LOWに設定 */

/* V850ES/JG3ではPDH0-PDH5のみ設定
   ( PMDH = 0xC0; ) */

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;         /* 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;
}

/*-----*/
/*   タイマMの設定   */
/*-----*/
static void f_init_int_tmm( void )
{
    TMOCTL0 = 0x04;     /* TMM0動作禁止 */
                        /* カウント・クロックはfxx/512 */
    TMOCMP0 = VAL_120ms_CNT; /* TMM0カウント設定 */
    TMOEQMK0 = 1;      /* タイマMの割り込みをマスク */
    TMOCE = 1;         /* TMM0動作許可 */
    return;
}

/*****
/*   割り込みモジュール   */
*****/
__interrupt
void f_int_intp0( void )
{
    unsigned int loop_wait; /* forループ用カウンタ */

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

```

```
if( ( P0 & 0x08 ) == 0x00 )          /* ウェイト後のSW1押下状態認識 */
{
    /* SW ON          */
    TMOCE = 0;                       /* カウント動作停止 */
    TMOCMP0 = VAL_55ms_CNT;          /* LED1点滅周期を55msに設定 */
    TMOEQIF0 = 0;                    /* TMM0コンペア一致割り込み要求クリア*/
    TMOCE = 1;                       /* カウント動作開始 */
    while ( ( P0 & 0x08 ) == 0x00 )
    {
        if ( TMOEQIF0 == 1 )        /* 割り込み要求信号あり? */
        {
            WDTE = 0xAC;            /* WDT2のカウントをクリア */
            PCM.3 ^= 1;             /* LED1出力反転 */
            TMOEQIF0 = 0;          /* TMM0コンペア一致割り込み要求クリア*/
        }
    }
    /* SW ON -> OFF */
    WDTE = 0xAC;                     /* WDT2のカウントをクリア */
    TMOCE = 0;                       /* カウント動作停止 */
    TMOCMP0 = VAL_120ms_CNT;         /* LED1点滅周期を120msに設定 */
    TMOCE = 1;                       /* カウント動作開始 */
}

PIF0 = 0;                            /* FailSafe：複数要求をクリア */

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

