

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、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。

M16C/62P グループ

電圧低下検出割り込みを使用したウェイトモードへの移行

1. 要約

この資料は、M16C/62P グループの電圧低下検出割り込みの応用例を掲載しています。

サンプルプログラムは電圧低下検出割り込みを使用して、電圧が低下するとウェイトモードへ移行し、電圧が上昇するとウェイトモードから復帰する処理を説明しています。

2. はじめに

この資料で説明する応用例は、次のマイコン、条件での利用に適用されます。

マイコン	M16C/62P グループ (注 1)M16C/62PT は、電圧検出回路の機能を使用しないでください。 (注 2)チップバージョン B 以降
評価 / 動作環境	Starter-Kit for M16C/62P (M3A-0664) 室温 (25)
VCC1	VCC1=VCC2=5 V
メインクロック	6MHz
CPU クロック	PLL クロック 24MHz (メインクロックの 4 通倍)

M16C/62P グループと同様の SFR(周辺機能制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

3. 機能概要

本サンプルプログラムで使用している機能について概要を説明します。

各機能の詳細については M16C/62P グループハードウェアマニュアルを参照ください。

機能	詳細についての参照先
3.1.電圧検出回路	「リセット」、「電圧検出回路」
3.2.ウェイトモード	「パワーコントロール」
3.3.コールドスタート/ウォームスタート	「コールドスタート/ウォームスタート」

本サンプルプログラムで設定している関連レジスタを「Appendix A」に添付します。詳細は最新のハードウェアマニュアルを参照ください。

3.1. 電圧検出回路

電圧検出回路は、VCC1 端子に入力される電圧を監視する回路です。
電圧検出回路には、次の 2 つがあります。

- ・ 電圧低下検出回路
- ・ リセット領域検出回路

以下にそれぞれの回路について説明します。

3.1.1 電圧低下検出回路

電圧低下検出回路は、VCC1 に入力される電圧を Vdet4 で監視する回路で、Vdet4 以上か、Vdet4 未満かを検出できます。

V27 ビット (電圧低下監視ビット) を "1" にすると電圧低下検出回路が有効になります。

(Vdet4 : 電圧低下検出電圧については、M16C/62P グループハードウェアマニュアルの「電気的特性」を参照してください)

3.1.2 電圧低下検出割り込み

電圧低下検出割り込みは、電圧低下検出回路を有効にし、D40 ビット (電圧低下検出割り込み許可ビット) を "1" にすると許可される割り込みです。

VCC1 端子に入力する電圧が上昇または降下して Vdet4 を通過したことを検出すると、D42 ビット (電圧変化検出フラグ) が "0" から "1" になり、電圧低下検出割り込み要求が発生します。D42 ビットはプログラムで "0" にしてください。

電圧低下検出割り込みの発生要因 ($VCC1 < Vdet4$ または $VCC1 > Vdet4$) の判定は、VC13 ビット (電圧低下モニタフラグ) で判定できます。

VC13 ビットの値が変化してからサンプリング時間経過した後、割り込み要求が発生します。

表 3.1 にサンプリング時間を、図 3.1 に電圧低下検出割り込み発生回路の動作例を示します。

電圧低下検出割り込みは、ウェイトモードからの復帰に使用できます。

電圧低下検出割り込みの割り込みベクタは、ウォッチドッグタイマ割り込み、および発振停止、再発振検出割り込みベクタと共用しています。

表 3.1 サンプリング時間

CPU クロック (MHz)	サンプリング時間 (μs)			
	DF1 ~ DF0=00b (CPU クロックの 8 分周)	DF1 ~ DF0=01b (CPU クロックの 16 分周)	DF1 ~ DF0=10b (CPU クロックの 32 分周)	DF1 ~ DF0=11b (CPU クロックの 64 分周)
16	3.0	6.0	12.0	24.0
PLL 24MHz(換算)	2.0	4.0	8.0	16.0

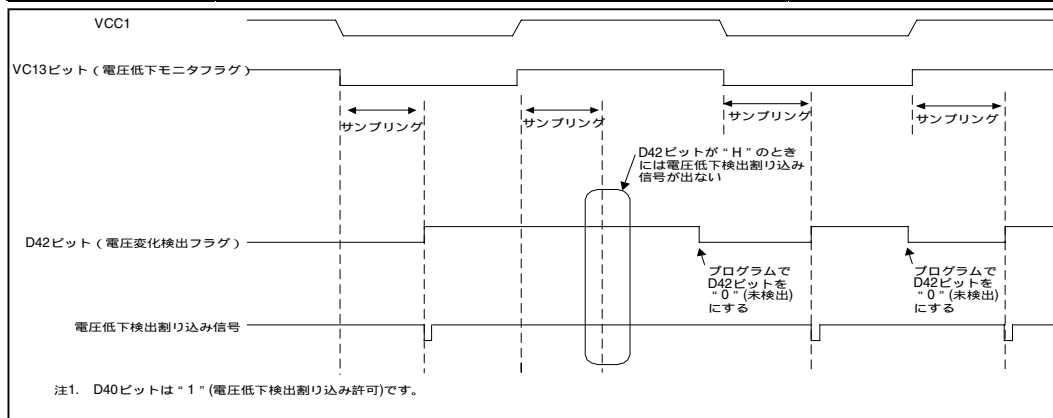


図 3.1 電圧低下検出割り込み発生回路の動作例

3.1.3 リセット領域検出回路

リセット領域検出回路は、電圧低下検出リセット（ハードウェアリセット2）で使用します。

3.1.4 電圧低下検出リセット（ハードウェアリセット2）

電圧低下検出リセットは、マイクロコンピュータに内蔵している電圧検出回路によるリセットです。VC26ビット（リセット領域監視ビット）を"1"にするとリセット領域検出回路が有効になります。

リセット領域検出回路を有効にすると、

- (1)VCC1 端子に入力される電圧が V_{det3} 以下になると端子が初期化されます。
- (2)次に VCC1 端子に入力される電圧が V_{det3r} 以上になると CPU、SFR が初期化され、リセットベクタで示される番地からプログラムを実行します。

以下に電圧低下検出リセットの動作例を示します。

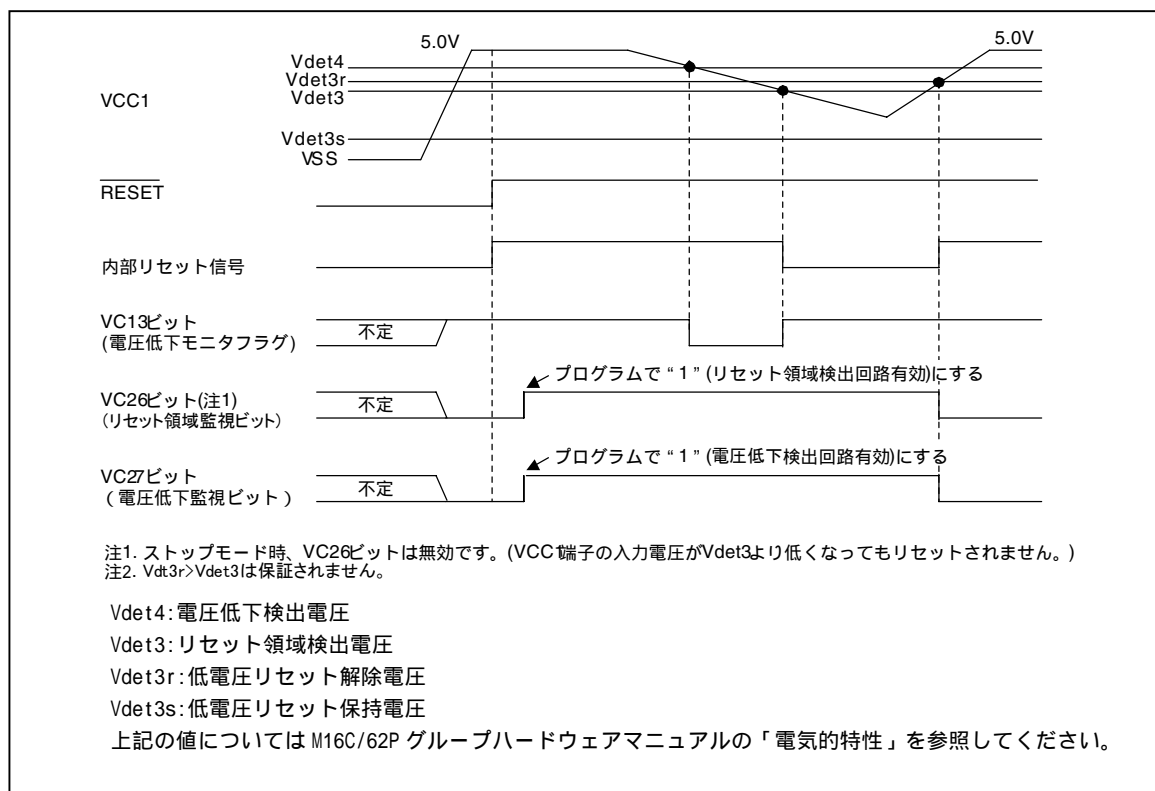


図 3.2 電圧低下検出リセット（ハードウェアリセット2）動作例

3.2. ウェイトモード

CPU クロックが停止するモードです。

ウェイトモードでは、メインクロック、サブクロック、オンチップオシレータクロックは停止しませんので、これらのクロックを使用する周辺機能は動作します。

周辺機能クロック停止機能 (CM02 ビット) により、

ウェイトモード時に周辺機能クロック (周辺機能クロックはメインクロックから生成されます) のうち $f_i(i=1,2,8,32)$, $f_{iSIO}(i=1,2,8,32)$, f_{AD} を停止させて消費電力を低減することができます。

3.2.1 ウェイトモードへの移行

WAIT 命令を実行するとウェイトモードになります。

CM11 ビットが"1" (CPU クロック源は PLL クロック) の場合は、CM11 ビットを"0" (CPU クロック源はメインクロック) にしてからウェイトモードに移行してください。

また、PLC07 ビットを"0" (PLL 停止) にすると、消費電力が低減できます。

3.2.2 ウェイトモードからの復帰

ハードウェアリセット、 \overline{NMI} 割り込み、または周辺機能割り込みにより、ウェイトモードから復帰します。表 3.2 にウェイトモードからの復帰に使用できる割り込みと使用条件を示します。

表 3.2 ウェイトモードからの復帰に使用できる割り込みと使用条件

割り込み	CM02=0 の場合	CM02=1 の場合
	(ウェイトモード時、周辺機能クロックを停止しない)	(ウェイトモード時、周辺機能クロックを停止する)
NMI割り込み	使用可	使用可
シリアル I/O 割り込み	内部クロック、外部クロックで使用可	外部クロックで使用可
キー入力割り込み	使用可	使用可
A/D 変換割り込み	単発モードまたは単掃引モードで使用可	使用しないでください
タイマ A 割り込み タイマ B 割り込み	全てのモードで使用可	イベントカウンタモードまたはカウントソースが f_{C32} のとき使用可
\overline{INT} 割り込み	使用可	使用可
電圧低下検出割り込み	使用可	使用可

- (1) ノンマスカブル割り込みで復帰する場合、周辺機能割り込みの ILVL2 ~ ILVL0 ビットを"000b" (割り込み禁止) にした後、WAIT 命令を実行してください。
- (2) ウェイトモードからの復帰に使用しない割り込みの割り込み優先レベルは 0 にしてください。

3.3. コールドスタート/ウォームスタート

- ・ コールドスタート：電源が投入された時のリセット処理
- ・ ウォームスタート：動作中にリセット信号が入力されたときのリセット処理

コールドスタートかウォームスタートかの判断は WDC (ウォッチドッグタイマ制御レジスタ) の WDC5 フラグ (コールドスタート/ウォームスタート判定フラグ) によって判定することができます。

WDC5 フラグは電源投入時"0" (コールドスタート) で、WDC レジスタに書き込み動作を行うと"1"になります。ソフトウェアリセットやリセット信号の入力に対しても"0"になりません。

ウォームスタート時は、内部 RAM は保持されます。

但し、内部 RAM に書き込み中に RESET 端子が"L"になると、内部 RAM は不定となります。

図 3.3 にコールドスタート/ウォームスタートの動作例を示します。

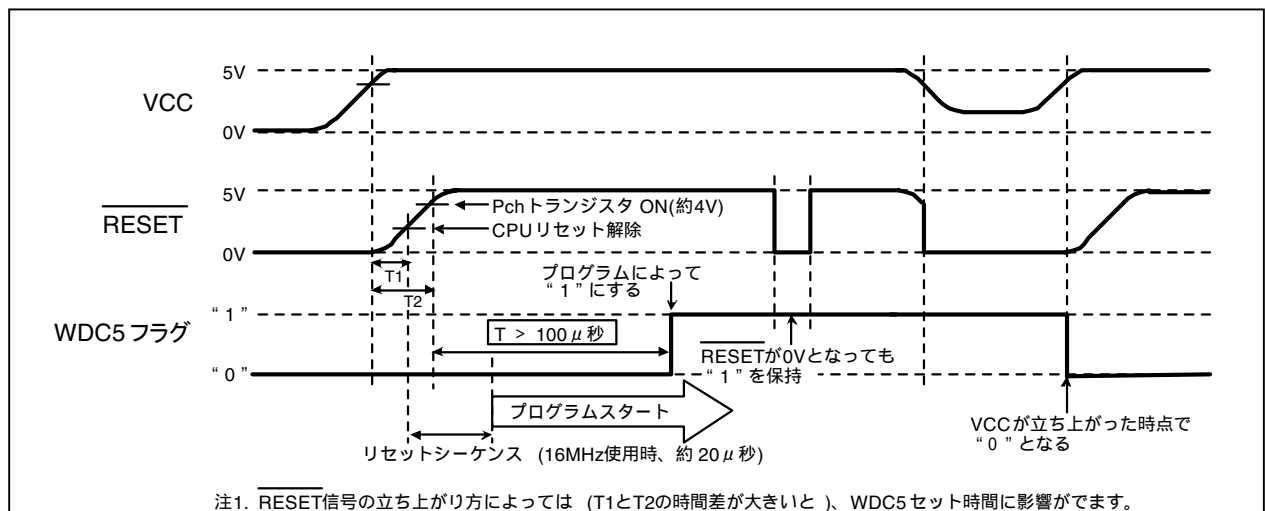


図 3.3 コールドスタート/ウォームスタートの動作例

4. サンプルプログラム仕様

Starter-Kit for M16C/62P (M3A-0664) を使用

メインクロック 6MHz

通常動作では、CPU クロック源を PLL 4 通倍 (24MHz) クロックでプログラムを動作させる。

メインプログラム実行前 (スタートアッププログラム) において WDC5 (コールドスタート/ウォームスタート判定フラグ) を参照して、RAM 初期化処理を行うかどうかを判定する。

電圧低下検出回路とリセット領域検出回路を有効にする。

電圧低下検出割り込みをウェイトモードへの移行とウェイトモードからの復帰に使用し、 $VCC1 < V_{det4}$ になったときウェイトモードに移行し、 $VCC1 \geq V_{det4}$ になったときウェイトモードから復帰する。

電圧低下検出割り込み処理内では以下の処理を行う。

- ・ $VCC1 < V_{det4}$ であれば、WAIT 命令を発行してウェイトモードへ移行する。
(この時、周辺機能クロックと PLL を停止させて消費電力を低減させる)
- ・ $VCC1 \geq V_{det4}$ であれば、ウェイトモードからの復帰であるので CPU クロック源を PLL 4 通倍クロックにして通常モードに戻す。

サンプルプログラム仕様

CPU クロック源を PLL 4 通倍 (24MHz) とする。1sec 周期で 7 セグメント LED に十進 2 桁のカウンタアップ表示 (ダイナミック点灯) を行う。

1sec 周期は、タイマ A0 で 1ms を生成し 1000 回カウントすることで作る。LED に使用するポートはスタータキットに準じて P0_7 ~ P0_0、P1_0, P1_1 を使用する。

以下に、Starter Kit for M16C/62P (M3A-0664) のポート接続を示します。

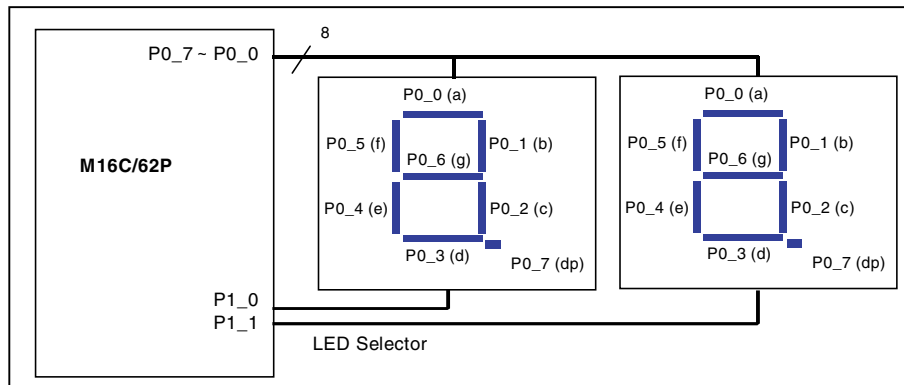


図 4.1 Starter Kit for M16C/62P (M3A-0664) のポート接続

サンプルプログラムの動作

- (1) 通常動作時は、1sec 周期でカウンタアップした十進 2 桁の値を 7 セグメント LED に表示します。
- (2) 電圧が低下して $VCC1 < V_{det4}$ になれば LED 表示が消灯してウェイトモードに移行します。
ウェイトモード時、周辺機能クロックを停止する設定にしているためタイマのカウント動作も停止します。
- (3) (2) の状態から電圧が上昇して $VCC1 \geq V_{det4}$ になれば、ウェイトモードから復帰し、カウンタアップ動作はウェイトモードに移行した時の状態から再開されます。
- (4) (2) の状態からさらに電圧が低下して $VCC1 < V_{det3}$ になると電圧低下検出リセット状態になります。
その後、リセット解除時にウォームスタートであれば RAM が保持されているので RAM 上の変数であるカウンタ値は残っています。カウンタ動作はリセット前のカウンタ値から再開されます。

以下にサンプルプログラムの状態遷移図と、 $VCC1$ 電圧変動とサンプルプログラムの状態遷移について示します。

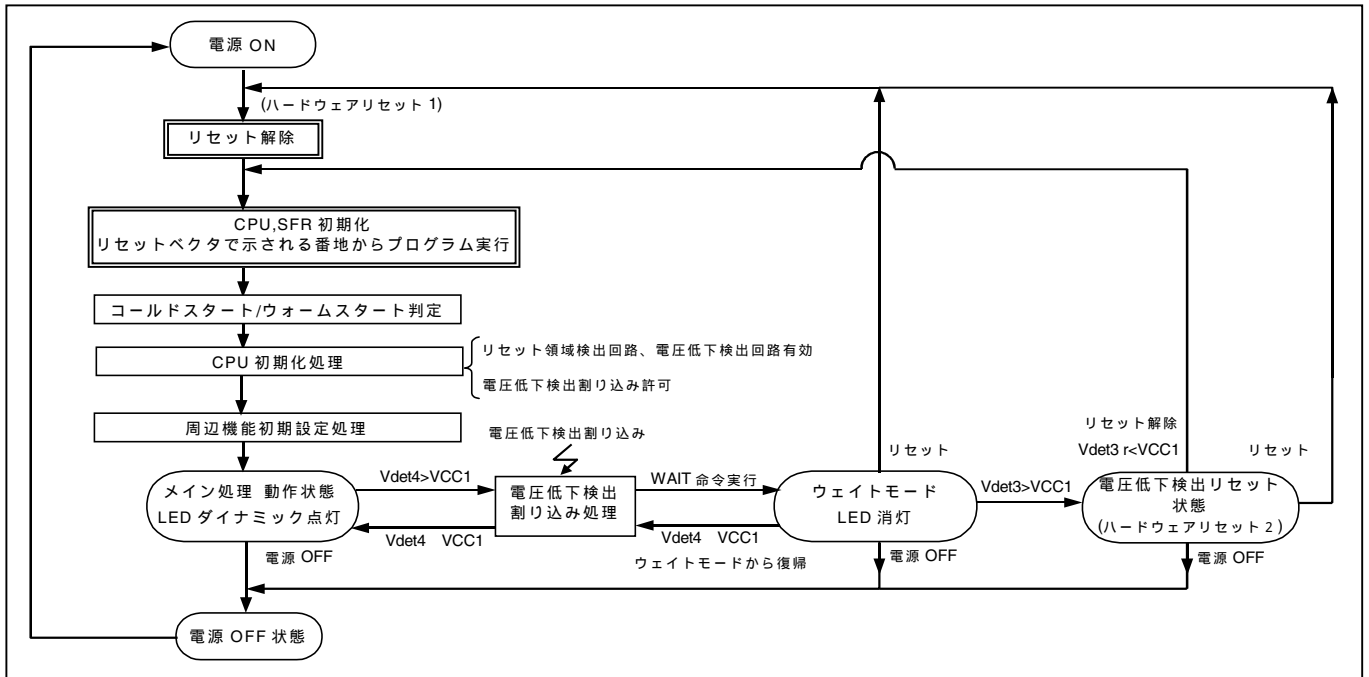


図 4.2 サンプルプログラムの状態遷移図

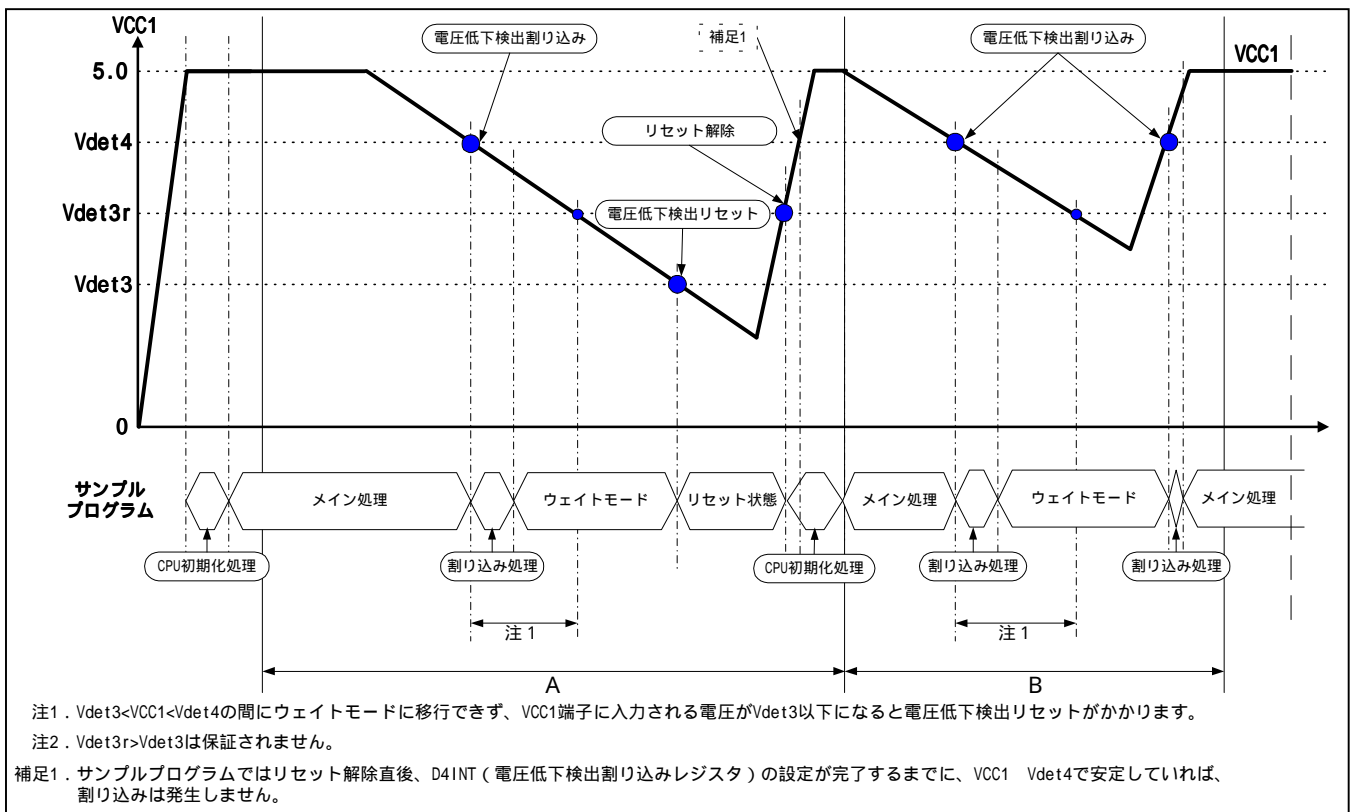


図 4.3 VCC1 の電圧変動とサンプルプログラムの状態遷移

【説明】

- A.
- (1) VCC1 端子に入力する電圧が V_{det4} 以下になると電圧低下検出割り込みが発生します。
 - (2) 割り込み処理内でウェイトモードに移行します。

- (3) ウェイトモードの状態では電圧が V_{det3} 以下になると電圧低下検出リセット(ハードウェアリセット 2)され、端子が初期化されます。
- (4) 電圧が上昇し、 V_{det3r} 以上になるとリセットが解除され、CPU、SFR が初期化され、リセットベクタで示される番地からプログラムを実行します。
- (5) リセット解除後、スタートアッププログラムでコールドスタート/ウォームスタートの判定を行います。
- (6) CPU 初期化処理では、
先頭でリセット領域検出回路と電圧低下検出回路を有効にします。
CPU 動作モードを設定した後、電圧低下検出割り込みを許可しています。
この時、 $VCC1 = V_{det4}$ であれば、CPU クロック源をメインクロック(6MHz)から PLL 4 逡倍に(24MHz)に変更します。
 $VCC1 < V_{det4}$ であれば、電圧低下検出割り込みが発生しウェイトモードに移行します。

B.

- (1) $VCC1$ 端子に入力する電圧が V_{det4} 以下になると電圧低下検出割り込みが発生します。
- (2) 割り込み処理内でウェイトモードに移行します。
- (3) 電圧が V_{det4} 以上になると電圧低下検出割り込みが発生します。この割り込みによってウェイトモードから復帰します。
割り込み処理内では CPU クロック源をメインクロックから PLL クロック(24MHz)に変更します。

4.1. レジスタの設定

サンプルプログラムで使用している機能に関連したレジスタの設定について説明します。

(1) コールドスタート/ウォームスタートの判定

WDC5 ビットを参照してコールドスタート/ウォームスタートの判定を行います。

サンプルプログラムでは、スタートアッププログラム中でコールドスタート/ウォームスタートの判定をし、以下の処理を行います。

- ・コールドスタート時：内部 RAM の初期化を行います。WDC5 ビットを"1"にします。
- ・ウォームスタート時：内部 RAM の初期化をスキップします。

ウォッチドッグタイマ制御レジスタ	シンボル WDC	アドレス 000Fh 番地																
<table border="1"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td></td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>	b7	b6	b5	b4	b3	b2	b1	b0		0								
b7	b6	b5	b4	b3	b2	b1	b0											
	0																	
<p>WDC5 コールドスタート/ウォームスタート判定フラグ (注 1,2)</p> <p>0:コールドスタート 1:ウォームスタート</p>																		
<p>(注 1) WDC レジスタに書くと、WDC5 ビットは、"1" (ウォームスタート) になります。 VCC1 端子に入力されている電圧が 4.0V より低い場合、CPU クロックを 2MHz 以下にするか、2 度書き込みを行ってください。</p> <p>(注 2) WDC5 ビットは、電源投入後"0" (コールドスタートです)。プログラムでのみ"1"にできます。</p>																		

(2) 電圧検出回路の設定

VCR2 (電圧検出レジスタ 2) を設定して、リセット領域検出回路と電圧低下検出回路を有効にします。

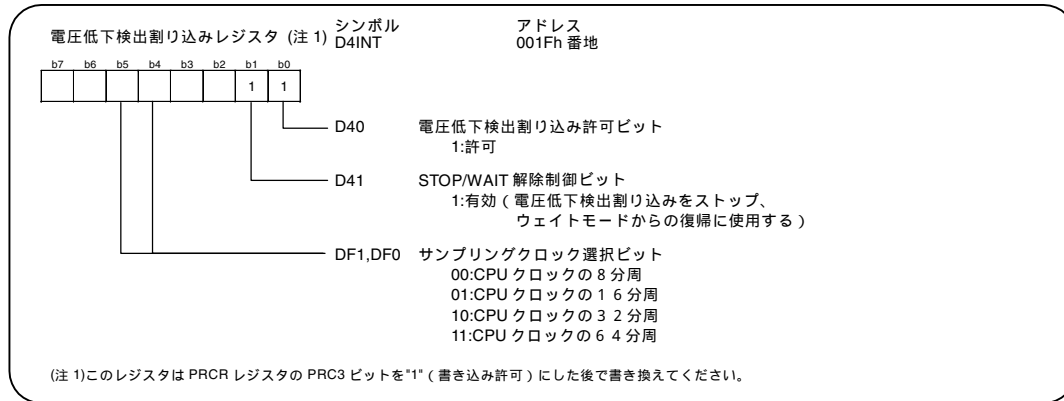
- ・VC26 ビットを"1"にして、リセット領域検出回路を"有効"にします。
- ・VC27 ビットを"1"にして、電圧低下検出回路を"有効"にします。

電圧検出レジスタ 2 (注 1)	シンボル VCR2	アドレス 001Ah 番地																
<table border="1"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>	b7	b6	b5	b4	b3	b2	b1	b0	1	1	0	0	0	0	0	0		
b7	b6	b5	b4	b3	b2	b1	b0											
1	1	0	0	0	0	0	0											
<p>b5-b0 予約ビット "0"にしてください</p>																		
<p>VC26 リセット領域監視ビット (注 2) 1:リセット領域検出回路有効</p>																		
<p>VC27 電圧低下監視ビット (注 2) 1:電圧低下検出回路有効</p>																		
<p>(注 1)このレジスタは PRCR レジスタの PRC3 ビットを"1" (書き込み許可) にした後で書き換えてください。</p> <p>(注 2)VC26 ビットまたは VC27 ビットを"1"にした後、td(E-A)経過してから検出回路が動作します。</p>																		

(3) 電圧低下検出割り込みの設定

D4INT (電圧低下検出割り込みレジスタ) を以下の設定にします。

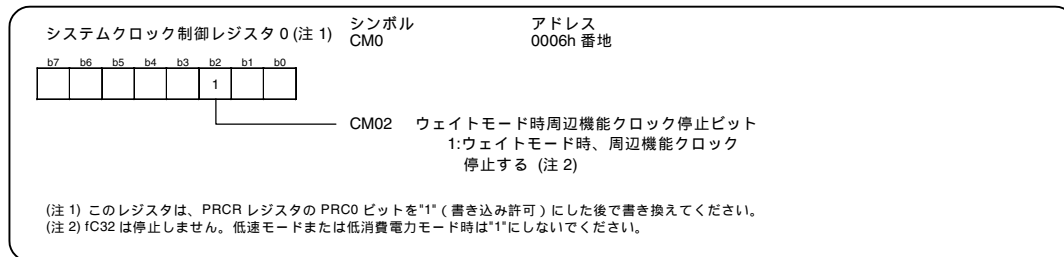
- D40 ビットを"1"にして、電圧低下検出割り込みを"許可"します。
- D41 ビットを"1"にして、"電圧低下検出割り込みをストップモード/ウェイトモードからの復帰に使用する"を選択します。
- サンプリングクロック選択ビットでサンプリングクロックを選択します。



(4) システムクロック制御レジスタ 0 の設定

CM02 ビットを"1"にして"ウェイトモード時、周辺機能クロック停止"を選択します。

これによりウェイトモード時に $f_i(i=1,2,8,32)$, $f_{iSIO}(i=1,2,8,32)$, f_{AD} が停止しますので、消費電力が低減できます。



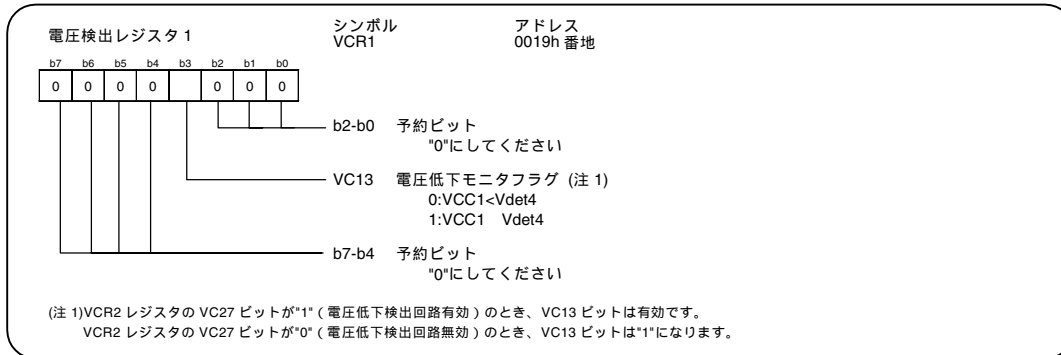
(5) 固定ベクタテーブルの設定

固定ベクタテーブルのベクタ番地 0FFFF0h ~ 0FFFF3h に、電圧低下検出割り込み処理のアドレスを設定します。固定ベクタテーブルについては、M16C/62P グループハードウェアマニュアルの「割り込み」を、具体的な設定方法については「6.2 sect30.inc (セクション定義ファイル)」とサンプルプログラムを参照ください。

(6) 電圧低下検出時の判定

電圧低下検出割り込みは、VCC1 端子に入力する電圧が上昇または降下して Vdet4 を通過すると発生します。

電圧低下検出割り込みの発生要因 (VCC1 > Vdet4 または VCC1 < Vdet4) の判定は、VC13 ビット (電圧低下モニタフラグ) で判定します。割り込み発生直後は VCC1 端子に入力する電圧は Vdet4 付近になります。この時 VCC1 端子に入力する電圧が振れると VC13 ビットの値が変化します。サンプルプログラムでは VC13 ビットを数回チェックして安定していることを確認して発生要因を判定しています。



5. サンプルプログラムフローチャート

5.1. 概要フロー

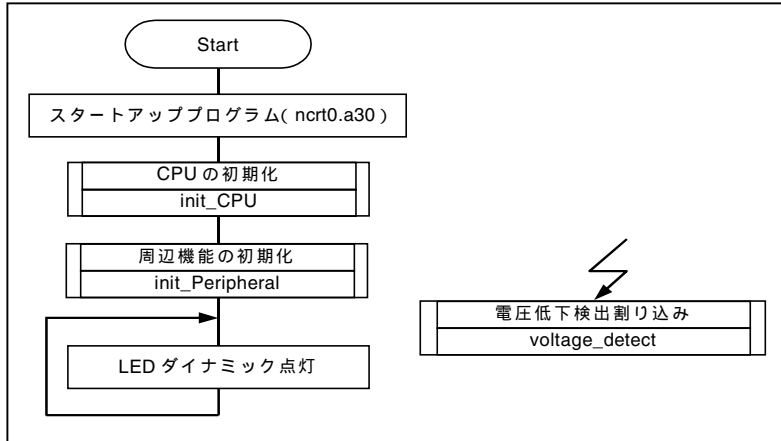


図 5.1 サンプルプログラム概要フロー図

5.2. スタートアッププログラム(ncrt0.a30)

スタートアッププログラムでは、WDC5 ビットを参照してコールドスタート/ウォームスタート判定を行っています。コールドスタート時は内部 RAM セクションの初期化を行います。ウォームスタート時は、RAM の初期化を行いません。

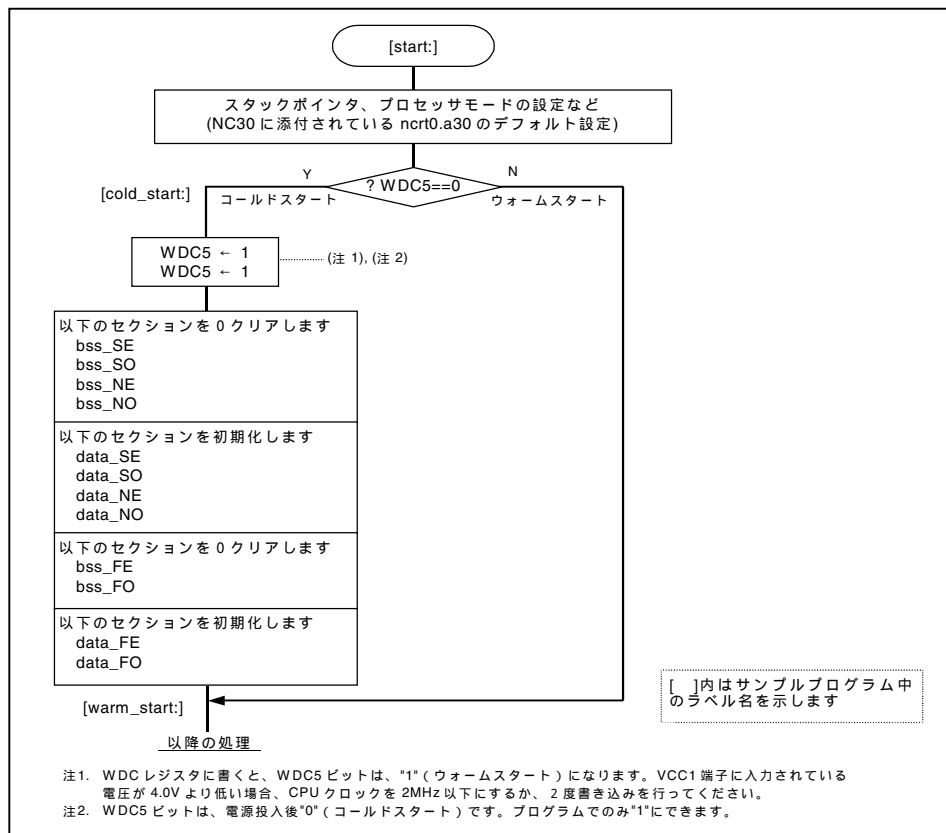


図 5.2 スタートアッププログラムのコールドスタート/ウォームスタート処理

5.3. CPU の初期化フロー (init_CPU)

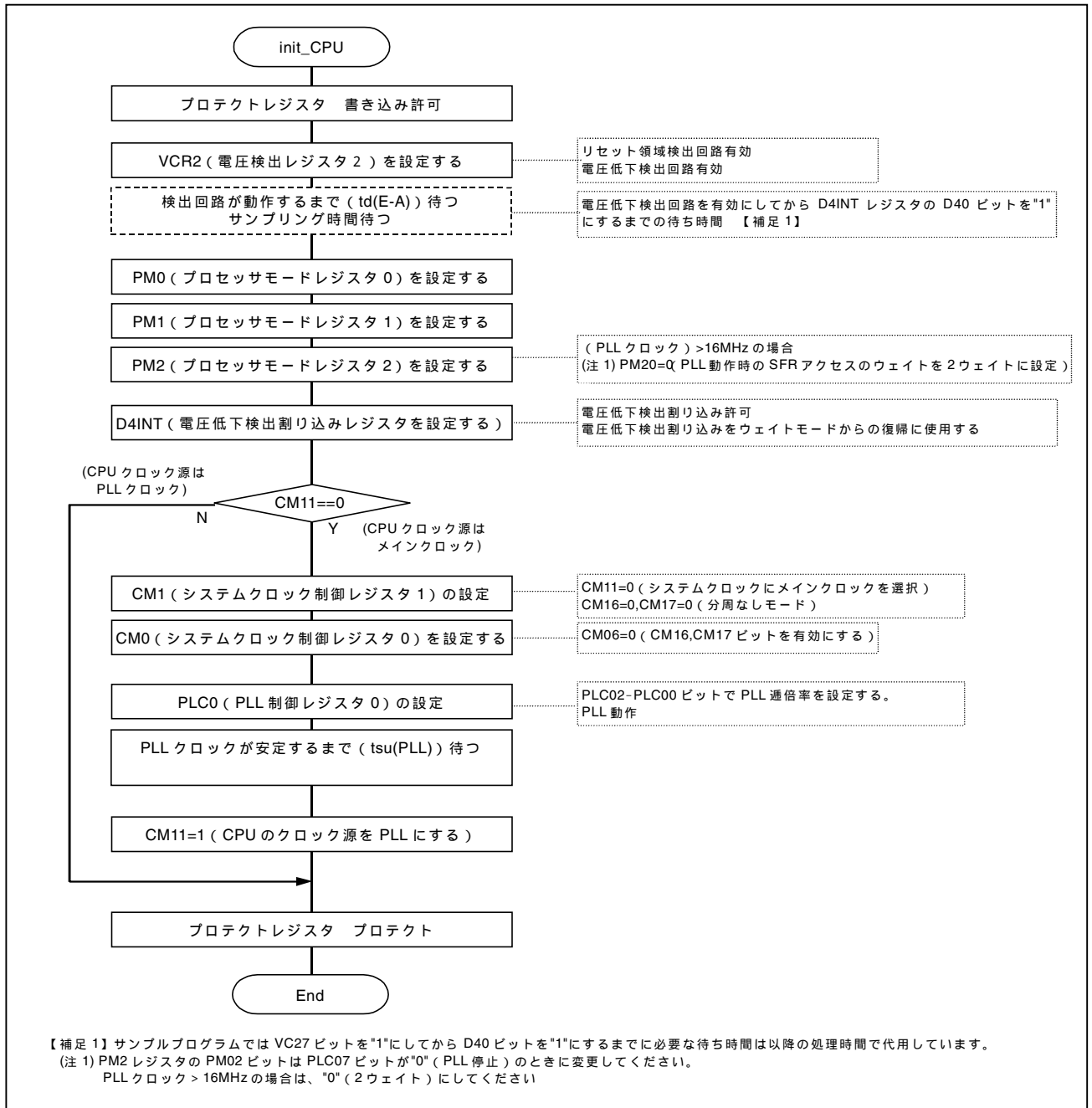


図 5.3 プロセッサの初期化フロー図

5.4. 周辺機能の初期化フロー (init_Peripheral)

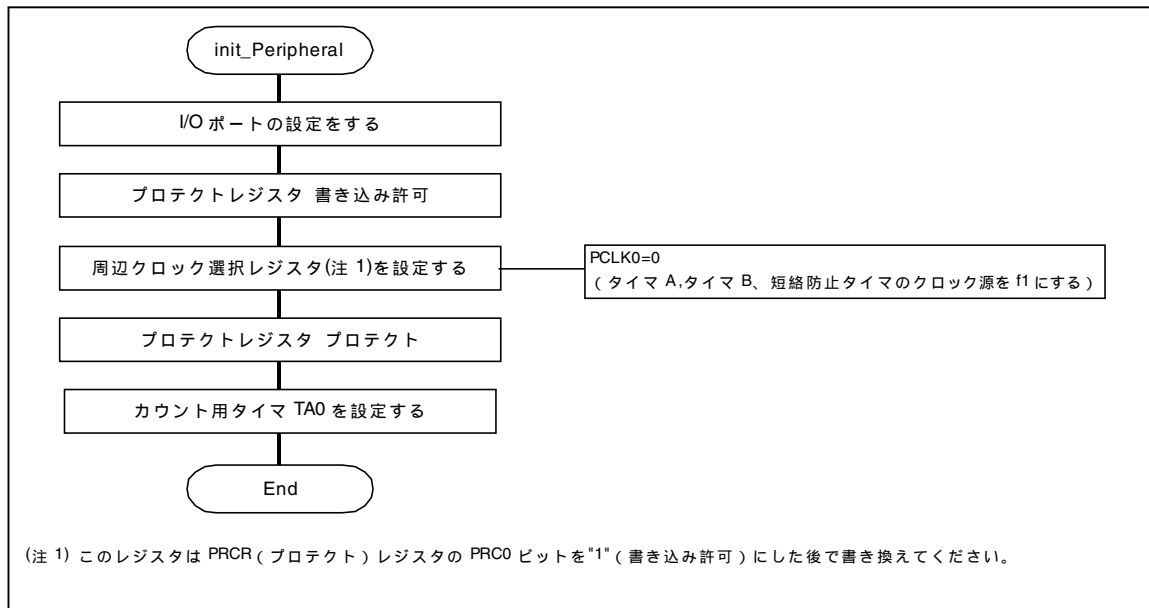


図 5.4 周辺機能の初期化フロー図

5.5. ウェイトモードへの移行と復帰

サンプルプログラムは、電圧低下検出割り込みにより、ウェイトモードへ移行またはウェイトモードからの復帰を行います。処理内容と処理フローを以下に示します。

VCC1<Vdet4 の時 (電圧が降下したとき→ウェイトモードへ移行する)

- (1) CPU クロック源を PLL クロックからメインクロックに変更する。
サンプルプログラムは CPU クロック源を PLL クロックに設定しているため、CPU クロック源をメインクロックに変更してからウェイトモードに移行する必要があります。
- (2) PLL を停止させて消費電力を低減させる。
- (3) LED 表示を消して LED のポートを入力に設定する。
- (4) WAIT 命令の実行によりウェイトモードへ移行する。
サンプルプログラムではウェイトモード時、周辺機能クロック停止を選択により消費電力を低減させています。

VCC1 > Vdet4 の時 (電圧が上昇したとき→ウェイトモードから復帰)

- (1) ウェイトモードからの復帰で電圧低下検出割り込みが発生しているため、CPU クロック源を PLL クロックに戻す。
- (2) LED のポートを出力に戻す。

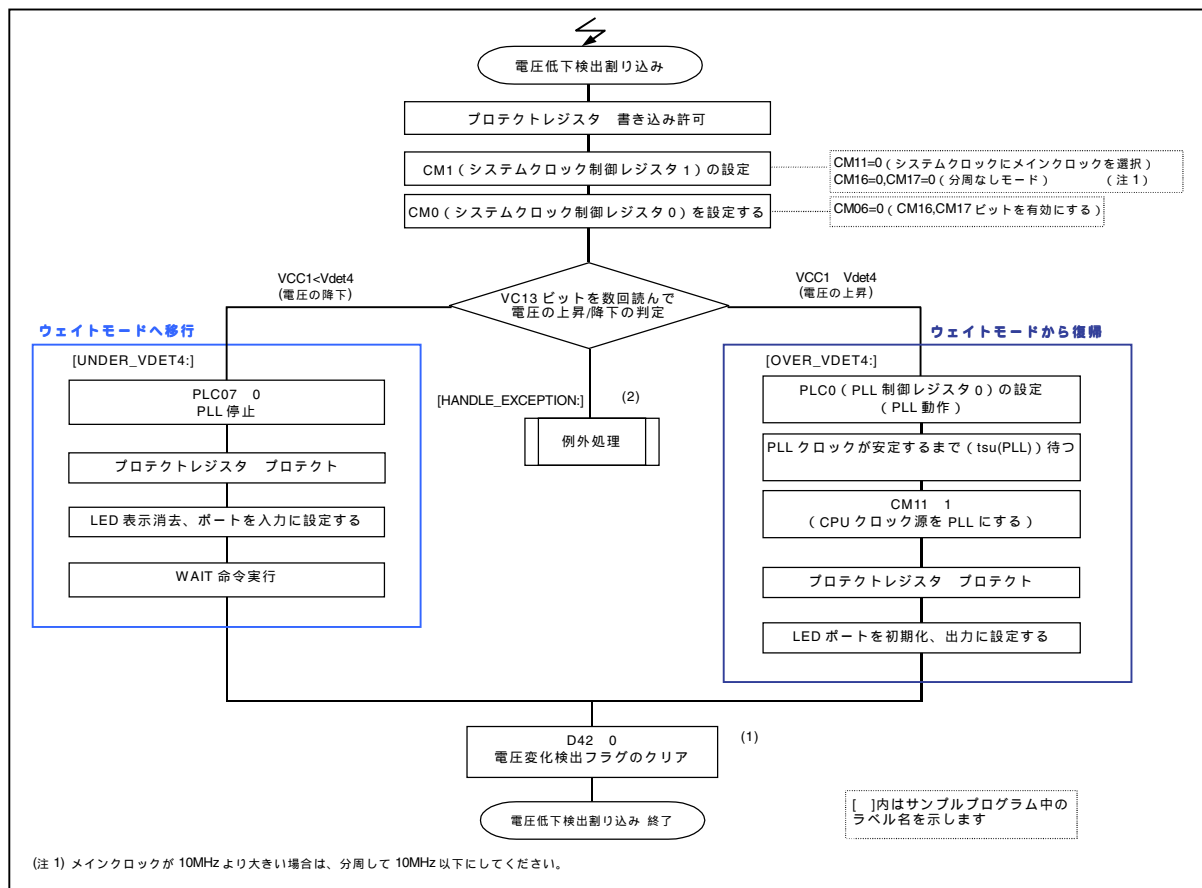


図 5.5 電圧低下検出割り込みフロー

【説明】

- (1) D42 ビットは VCC1 端子に入力する電圧が上昇または降下して Vdet4 を通過したことを検出したとき"1"になります。D42 ビットが 1 のとき、電圧低下検出割り込み信号はでません。D42 ビットはプログラムで"0"にしてください。
- (2) サンプルプログラムでは、前提条件の下で VC13 ビットの値が連続して同じ値に安定しなかった場合は、例外として扱います。サンプルプログラムの前提条件については次頁を参照ください。

5.5.1 サンプルプログラムの前提条件

サンプルプログラムではウェイトモードへ移行時とウェイトモードからの復帰時の VCC1 の上昇/降下について、次の条件を前提としています。

【サンプルプログラムの前提条件】

VCC1 端子に入力される電圧は、Vdet4 を通過後、40ms 以内に再度 Vdet4 を通過しないこと。

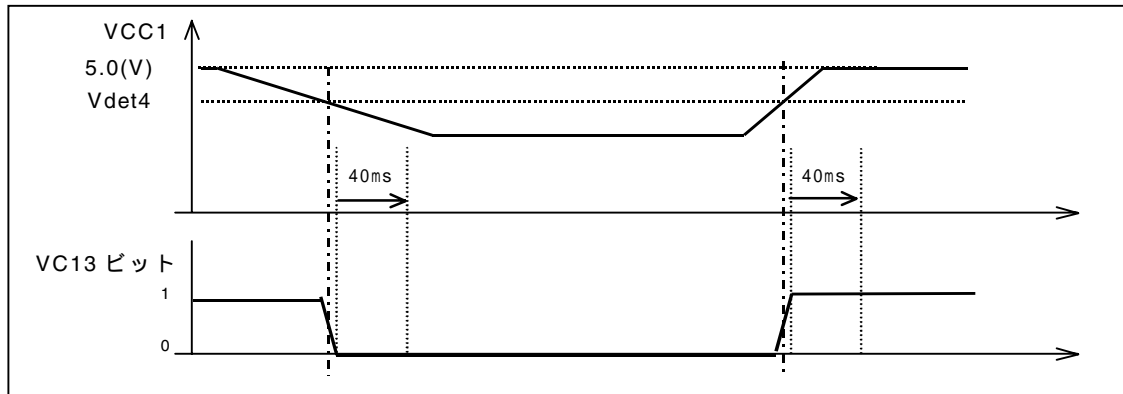


図 5.6 サンプルプログラムの VCC1 上昇/降下についての条件

5.5.2 VC13 ビット (電圧低下モニタフラグ) の判定例

「5.5.1 サンプルプログラムの前提条件」の下で、サンプルプログラムにおける VC13 ビットの判定例を示します。

- (1) VC13 ビットをチェック間隔(約 1ms)でチェックします。
(チェック間隔はループ処理で作っています)
- (2) VC13 ビットが連続 CNT_UNDER_VDET4 (=8)回"0"になれば VCC1 < Vdet4 と判定する。
- (3) VC13 ビットが連続 CNT_OVER_VDET4 (=8)回"1"になれば VCC1 > Vdet4 と判定する。
- (4) VC13 ビットのチェック回数が CNT_LOOP_MAX (=20)回を超えれば例外発生と判定する。

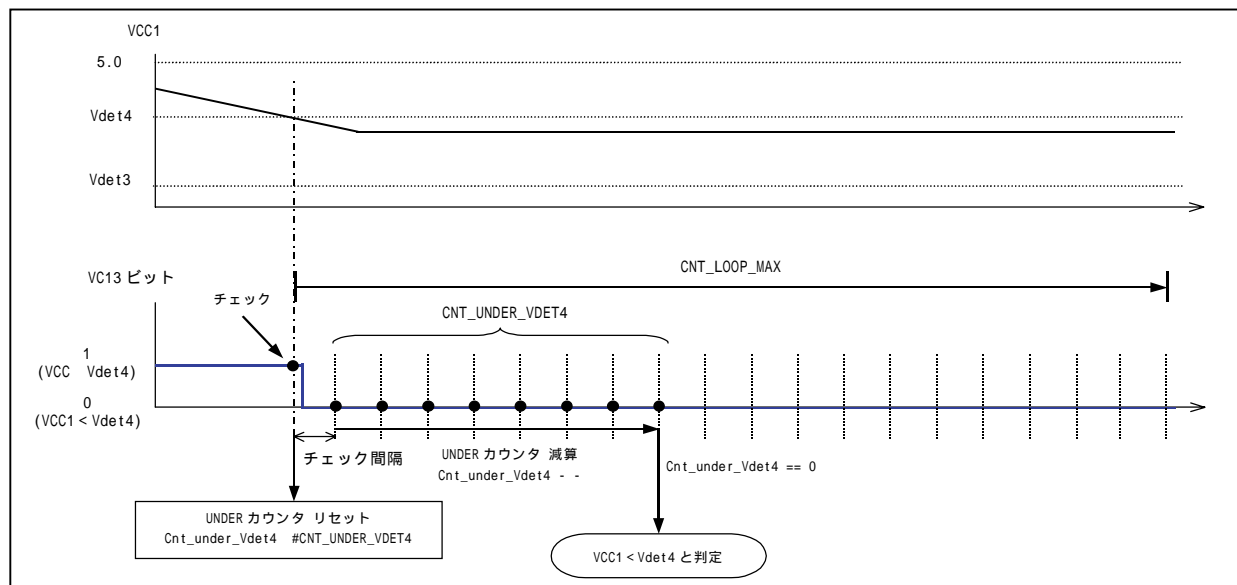


図 5.7 VC13 ビットの判定例 (ウェイトモードへの移行時)

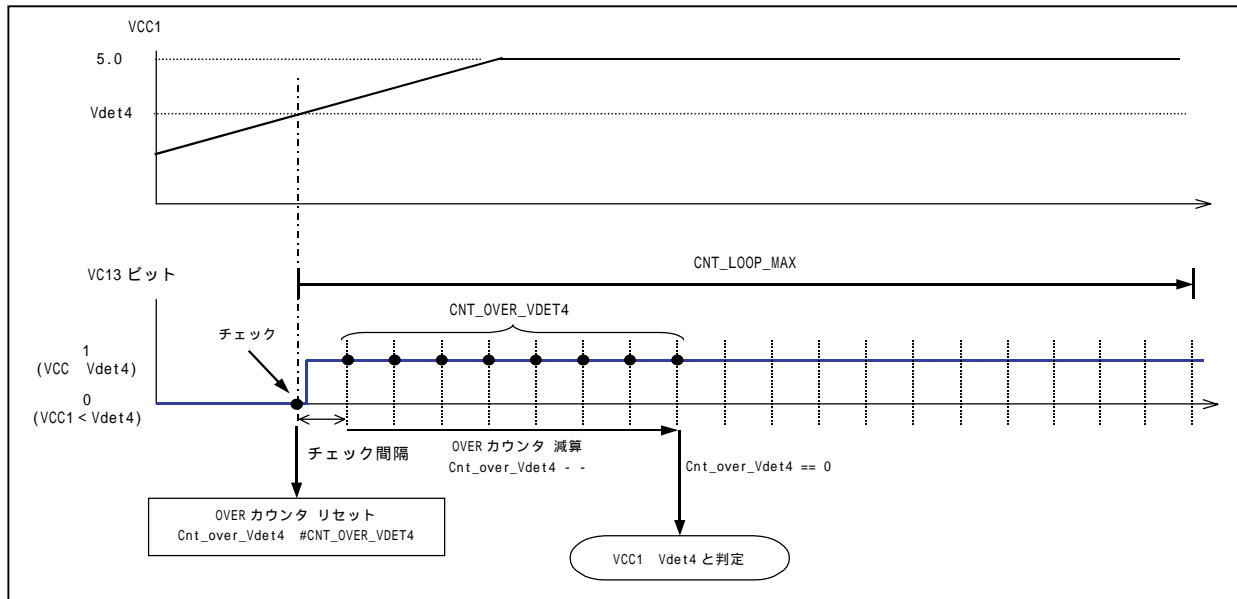


図 5.8 VC13 ビットの判定例 (ウェイトモードからの復帰時)

5.5.3 VC13 ビット (電圧低下モニタフラグ) のチェックフロー

サンプルプログラムの VC13 ビットの判定フローを以下に示します。

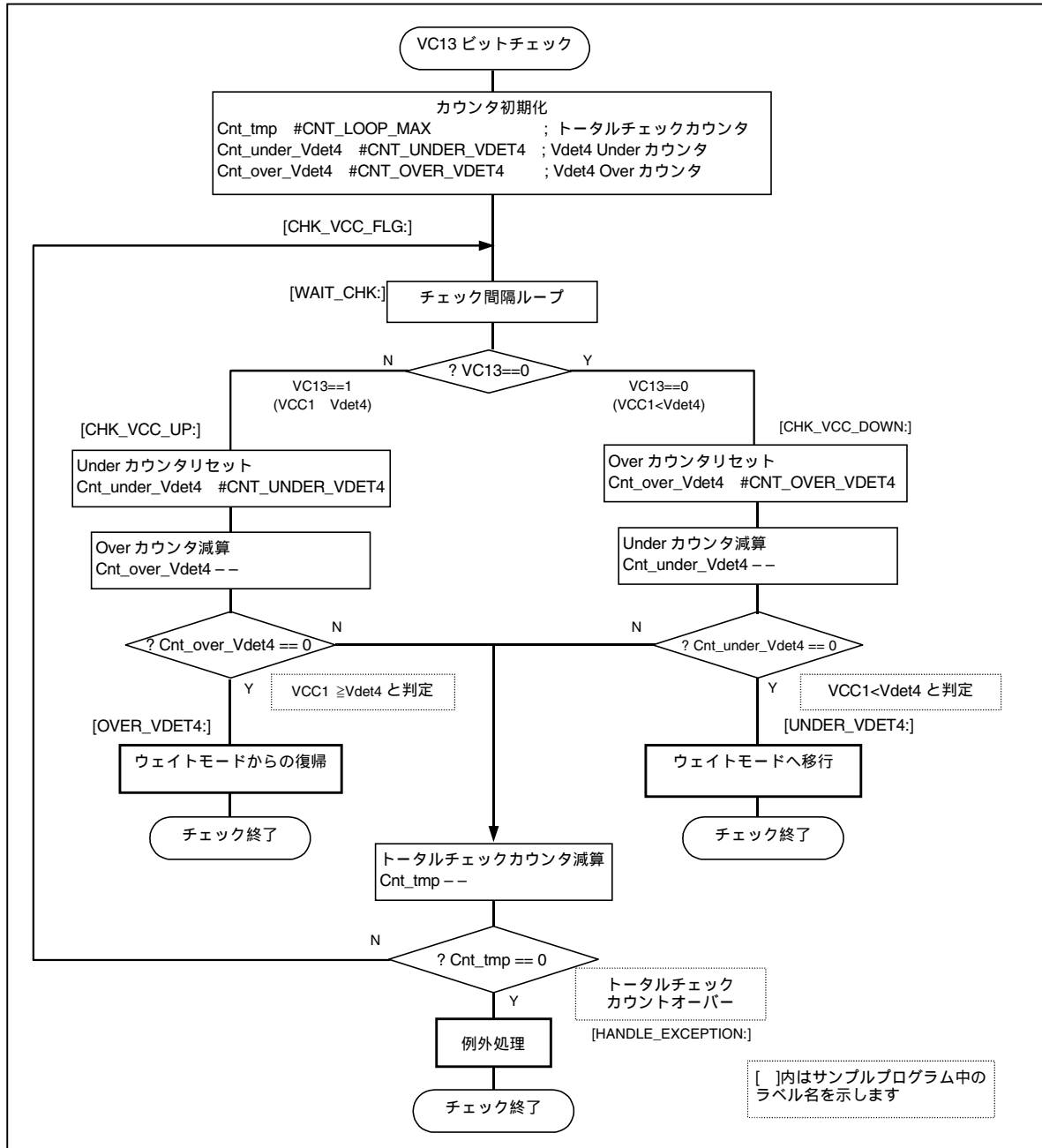


図 5.9 VC13 ビット (電圧低下モニタフラグ) のチェックフロー図

【説明】

サンプルプログラムでは、VC13 ビットを間隔をあけてチェックし、以下のように判定しています。

- ・連続して CNT_OVER_VDET4 (回)"1"で安定した場合は VCC1 >= Vdet4 と判定します。
- ・連続して CNT_UNDER_VDET4 (回)"0"で安定した場合は VCC1 < Vdet4 と判定します。

5.6. HANDLE_EXCEPTION (例外処理)

サンプルプログラムでは、電圧低下検出割り込み処理内で VC13 ビット (電圧低下モニタフラグ) が連続して安定せずチェック回数が CNT_LOOP_MAX を超えれば、例外として扱っています。

サンプルプログラムの例外処理は、電圧低下検出割り込みを禁止にした後、ウェイトモードに移行させてそれ以後のプログラム動作を停止させています。

電圧低下検出割り込みを禁止にしてウェイトモードへ移行しているので、例外発生後に電圧が VCC1 Vdet4 に上昇しても、ウェイトモードからは復帰しません。例外発生の情報として、LED に "EE" と表示します。

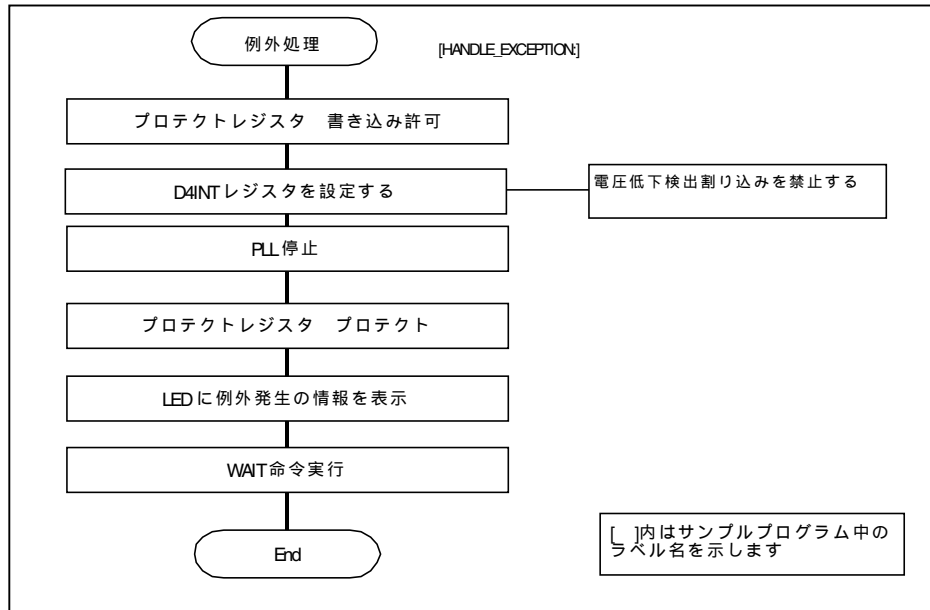


図 5.10 サンプルプログラムの例外処理

6. サンプルプログラム

本サンプルプログラムは「4. サンプルプログラム仕様」で示した条件でのサンプルプログラムであり、任意のシステム / アプリケーションにおいて動作を保証するものではありません。

本サンプルプログラムはC言語とアセンブリ言語で記述しています。

また、プログラムのROM化にはCコンパイラ NC30WA Ver.5.30 Release 1 に付属しているサンプルのスタートアッププログラムとセクション定義ファイルを流用しています。

使用コンパイラ

- NC30WA Ver.5.30 Release 1
- サンプルプログラムのコンパイルオプション (詳細は NC30 のユーザーズマニュアルを参照ください)

-c	リロケータブルファイル(拡張子.r30)を作成し、処理を終了します。
-dSL	C言語ソースリストをコメントとして出力したアセンブリ言語ソースファイル(拡張子".a30")を生成します。さらにアセンブリ言語リストファイル(拡張子".lst")を生成します。
-fER	レジスタ記憶クラスを有効にします。
-g	デバッグ情報をアセンブリ言語ソースファイル(拡張子".a30")に出力します。これにより、C言語レベルデバッグが可能になります。
-finfo	インスペクタ、"Stk Viewer"、"Map Viewer"、"utl30"、に必要な情報を出力します。エントリー版では、本オプションは指定できません。
-Wall	検出可能な警告をすべて表示します。
-WEF エラー出力ファイル名	エラーメッセージを指定したファイルに出力します。

以下にファイル構成について説明します。

ファイル構成

ファイル名	内 容	
sfr62p.h, sfr62p.inc	M16C/62P 用の SFR 定義ファイルです。最新のファイルを入手してご使用ください。	
ncrt0.a30 (スタートアッププログラム)	スタートアッププログラム (NC30WA Ver.5.30 Release 1 に添付されているファイルを流用しています。 コールドスタート/ウォームスタートを判定して内部 RAM の初期化処理で分岐しています)	
sect30.inc (セクション定義ファイル)	セクション定義ファイル (NC30WA Ver.5.30 Release 1 に添付されているファイルを流用しています) ・ 固定割り込みベクタに割り込み処理プログラムを登録しています ・ プログラムサイズが小さいのでプログラムの配置アドレスを変更しています	
board.h	サンプルプログラム用定義ファイル (Starter Kit for M16C/62P M3A-0664 用)	
C_main.c	メインプログラム	
	関数名	処理概要
	main	スタートアッププログラムから Call されるメイン関数です。 無限ループで LED のカウントアップ表示を行います。
	init_CPU	マイコンの初期設定を行います。 マイコンの動作モード、電圧検出レジスタ 2、 電圧低下検出割り込みレジスタなどの初期設定を行います。
	init_Peripheral	周辺機能の初期設定を行います。 LED と接続されているポートの設定、カウントアップに使用するタイマ A0 の設定を行います。
A_interrupt.a30	LED のダイナミック表示を行います。 1sec ごとに LED 表示カウンタ変数をカウントアップします。	
	電圧低下検出割り込み処理プログラム	
	関数名	処理概要
	voltage_detect	電圧低下検出割り込み処理です。 電圧低下検出割り込み要求が発生すると、この処理が実行されます。

6.1. ncr0.a30 (スタートアッププログラム)

本サンプルプログラムのスタートアッププログラムは、NC30WA Ver.5.30 Release 1 に添付されているサンプルのスタートアッププログラムを流用しています。コールドスタート/ウォームスタート判定フラグにより、初期化する/しない内部 RAM 領域は C コンパイラ NC30 が出力するセクションを対象としています。スタートアッププログラムの詳細については、NC30 のユーザーズマニュアルを参照ください。

```

;-----
; Cold Start/Warm Start Discrimination @100
;-----
    btst   wdc5      ;
    jc     warm_start ;
cold_start:
    bset   wdc5      ; Set Cold/Warm Start Discrimination Flag for Next Time
    bset   wdc5      ; (Note2)
                ; If the voltage applied to VCC1 is less than 4.0V, either write
                ; to this register when the CPU clock frequency is 2MHz or write twice.
;-----
; Cold Start (Initialize the RAM area) @100
;-----
;=====
; NEAR area initialize.
;-----
; bss zero clear
;-----
    N_BZERO bss_SE_top,bss_SE
    N_BZERO bss_SO_top,bss_SO
    N_BZERO bss_NE_top,bss_NE
    N_BZERO bss_NO_top,bss_NO
;-----
; initialize data section
;-----
    N_BCOPY data_SEI_top,data_SE_top,data_SE
    N_BCOPY data_SOI_top,data_SO_top,data_SO
    N_BCOPY data_NEI_top,data_NE_top,data_NE
    N_BCOPY data_NOI_top,data_NO_top,data_NO
;=====
; FAR area initialize.
;-----
; bss zero clear
;-----
    BZERO  bss_FE_top,bss_FE
    BZERO  bss_FO_top,bss_FO
;-----
; Copy edata_E(0) section from edata_EI(0I) section
;-----
    BCOPY  data_FEI_top,data_FE_top,data_FE
    BCOPY  data_FOI_top,data_FO_top,data_FO
;-----
; Warm Start (Does not initialize the RAM area) @100
;-----
warm_start:
    ldc #stack_top,sp
    .stk -40
;=====
; heap area initialize
;-----
(以下省略)
;
;

```

(1)コールドスタート/ウォームスタートの判定。

コールドスタート時、以下の RAM 領域の 0 クリア、初期化を行います

初期値を持たない near RAM 領域の 0 クリア。

初期値をもつ near RAM 領域の初期化。

初期値を持たない far RAM 領域の 0 クリア。

初期値をもつ far RAM 領域の初期化。

ウォームスタート時、上記の RAM 領域の 0 クリア、初期化を行わず、(1)からここにジャンプしてきます。

6.2. sect30.inc (セクション定義ファイル)

セクション定義ファイルでは、Flash ROM エリアの先頭アドレスの変更と、電圧低下検出割り込み処理 (voltage_detect) を固定ベクタテーブルに登録しています。

```

*****
;
; C Compiler for R8C/Tiny, M16C/60,30,20,10
; COPYRIGHT(C) 1999(2000-2002) RENESAS TECHNOLOGY CORPORATION
; AND RENESAS SOLUTIONS CORPORATION ALL RIGHTS RESERVED
;
;
; Written by T.Aoyama
;
; sect30.inc      : section definition
; This program is applicable when using the basic I/O library
;
; $Id: sect30.inc,v 1.22 2004/02/16 05:17:14 muranaka Exp $
;
; 2005.03.15 Ver.1.00
; Sample operation of Voltage Detection Circuit.
; This file was diverted from the sect30.inc attached to
; the NC30WA Ver.5.30 Release 1.
; The lines changed from original source are marked by @100
*****
;
.if __R8C__ != 1
;
; for M16C/60,30,20,10
;
;-----
; Arrangement of section
;-----
; Near RAM data area
;-----
; SBDATA area
; .section data_SE,DATA
; .org 400H
data_SE_top:

; .section bss_SE,DATA,ALIGN
bss_SE_top:

; .section data_S0,DATA
data_S0_top:

; .section bss_S0,DATA
bss_S0_top:

; near RAM area
; .section data_NE,DATA,ALIGN
data_NE_top:

; .section bss_NE,DATA,ALIGN
bss_NE_top:

; .section data_NO,DATA
data_NO_top:

; .section bss_NO,DATA
bss_NO_top:

;-----
; Stack area
;-----
; .section stack,DATA
; .blkb STACKSIZE
stack_top:

; .blkb ISTACKSIZE
istack_top:

```

```

;-----
; heap section
;-----
.section heap,DATA
heap_top:
.blkb HEAPSIZE

;-----
; Near ROM data area
;-----
.section rom_NE,ROMDATA,ALIGN
rom_NE_top:

.section rom_NO,ROMDATA
rom_NO_top:

;-----
; Far RAM data area
;-----
.section data_FE,DATA
.org 10000H
data_FE_top:

.section bss_FE,DATA,ALIGN
bss_FE_top:

.section data_F0,DATA
data_F0_top:

.section bss_F0,DATA
bss_F0_top:

;-----
; Far ROM data area
;-----
.section rom_FE,ROMDATA
.if 0
.org 0F0000H
.else
.org 0FF000H ; @100 Use Flash memory area from Block 0
.endif
rom_FE_top:

.section rom_F0,ROMDATA
rom_F0_top:

(中略)
:
:

```

プログラムサイズが小さいので Flash ROM エリアを Block0 から使用するように変更しています

```

;-----
; fixed vector section
;-----
.section fvector,ROMDATA
; .org 0ffffdch
;UDI:
; .lword dummy_int
;OVER_FLOW:
; .lword dummy_int
;BRKI:
; .lword dummy_int
;ADDRESS_MATCH:
; .lword dummy_int
;SINGLE_STEP:
; .lword dummy_int

```

電圧低下検出割り込み処理 (voltage_detect) を
固定ベクタテーブルに登録しています。

```

;-----
; @100
; Interrupt handling routine of the Voltage Down Detect Function
.glb voltage_detect
.org 0ffff0H ; Vector Table Address of
WDT: ; WDT, Oscillation Stop and, Re-oscillation, Voltage Down Detection
.lword voltage_detect ;
;-----
;DBC:
; .lword dummy_int
;NMI:
; .lword dummy_int
.org 0ffffch
RESET:
.lword start

.else ; __R8C__
(以下省略)
;
;

```

6.3. C_main.c

サンプルプログラムを動作させる場合、以下の値はコンパイルオプションやシステムに応じて設定値を変更してください。

定義	内容
CNT_WAIT_PLL	関数 init_CPU で使用しています。 PLL 発振安定待ち時間 tsu(PLL)を作るためのループ回数を定義しています。 なお、正確に tsu(PLL)以上の待ち時間を作るにはタイマを利用することを推奨します。

```

/*****
*
* FILE      : C_main.c
* CONTENTS  : Sample operation of Voltage Detection Circuit
*             Main routine
*
* CPU       : M16C/62P (Starter Kit for M16C/62P :M3A-0644)
* OS        : No used
* COMMENT   :
* HISTORY   : 2005.03.15 Ver.1.00
*
* NOTE      :
*
* Copyright(C)2004, Renesas Technology Corp.
* Copyright(C)2004, Renesas Solutions Corp.
* All rights reserved.
*****/

/*****
*
* Include files
*****/
#include "sfr62p.h"
#include "board.h"

/*****
*
* Defines
*****/
/* Set the values suitable for the system */
#define CNT_WAIT_PLL    (14000) /* Count value of
                               the PLL Frequency Synthesizer Stabilization Wait Time (@6MHz) */
#define TAO_INTERVAL   (24000-1) /* 1ms @24MHz PLL, f1 */
#define REFRESH_INTERVAL (4) /* (msec) Dynamically turn on LED interval */

/*****
*
* Global Variables
*****/
short Digit_1, Digit_10;

/*****
*
* Prototypes
*****/
void main( void );
void init_CPU( void );
void init_Peripheral( void );
void refresh_LED( void );

```

```

/*****
MODULE : main
FUNCTION: Main routine of sample program
PARAMETERS: None
RETURN : None
*****/
*/
void main( void )
{
    unsigned short cnt_wait = 0;

    init_CPU();
    init_Peripheral();

    ta0s = 1;
    while(1){
        if( ir_ta0ic == 1 ){
            ta0ic = 0;
            /* ? REFRESH_INTERVAL ms */
            if ( REFRESH_INTERVAL < cnt_wait++ ){
                cnt_wait = 0;
                refresh_LED();
            }
        }
    }
} /* while */
}

```

```

/*****
MODULE : init_CPU
FUNCTION: Initialize CPU
    Processor Mode: Single-Chip Mode
    CPU Clock: PLL Clock (Xin = 6MHz, Multiply by 4 --> 24MHz)
    Enable reset level detection circuit
    Enable voltage down detection circuit
    Use the voltage down detection interrupt to get out of stop/wait mode
PARAMETERS: None
RETURN : None
*****/
*/
void init_CPU( void )
{
    volatile short cnt_word, cnt_over_vdet4;

    prcr = 0x0B;    // Protect Register (write enabled)

    //=====
    // Setting Voltage Detection Register 2
    //=====
    vcr2 = 0xC0;
    // (b5-b0) Resrved Bit (Set to "0")           [000000]
    // (b6) Reset Level Monitor Bit               [1:Enable reset level detection circuit]
    // (b7) Voltage Down Monitor Bit             [1:Enable voltage down detection circuit]

    // Wait for td(E-A) until the detection circuit is actuated (See the following Note5 of D4INT)
    // Wait for the Sampling time (See the following Note5 of D4INT)
    // (In this sample, waiting time from setting 1 to VC27 to setting 1 to D40 is
    // substituted by the following processing time)

    //=====
    // Setting Processor Mode
    //=====
    // Processor Mode Register 0
    pm0 = 0x00;
    // (b1-b0) Processor Mode Bit                 [00:Single-Chip]
    // (b2) R/W Mode Select Bit                   [0:(Note2)]
    // (b3) Software Reset Bit                   [0]
    // (b5-b4) Multiplexed Bus Space Select Bit   [00:(Note2)]
    // (b6) Port P4_0 to P4_3 Function Select Bit [0:(Note2)]
    // (b7) BCLK Output Disable Bit              [0:(Note2)]
    // (Note2) Effective when memory expansion mode or microprocessor mode

    // Processor Mode Register 1
    pm1 = 0x08;
    // (b0) CS2 Area Switch Bit                   [0:Block A disable]
    // (b1) Port P3_7 to P3_4 Function Select Bit [0:(Note3)]
    // (b2) Watchdog Timer Function Select Bit    [0:Watchdog timer interrupt]
    // (b3) Internal Reserved Area Expansion Bit  [1:The entire area is usable]
    // (b5-b4) Memory Area Expansion Bit         [00:(Note3)]
    // (b6) Reserved bit (Set to "0")            [0]
    // (b7) Wait Bit                              [0:No wait state]
    // (Note3) Effective when memory expansion mode or microprocessor mode

    // Processor Mode Register 2 (Set the PM20 bit to "0" (2 wait states))
    pm2 = 0x00;
    // (b0) Specifying Wait when Accessing SFR at PLL Operation
    //                                           [0:2 waits (Note2)]
    // (b1) System Clock Protective Bit          [0:Clock is protected by PRCR register]
    // (b2) WDT Count Source Protective Bit      [0:CPU clock is used for the WDT count source]
    // (b4-b3) Reserved Bit (Set to "0")        [00]
    // (b7-b5) Nothing is assigned (When write, set to "0")
    //                                           [000]
    // (Note2) The PM20 bit become effective when PLC07 bit in the PLC0 register is
    // set to "1"(PLL On).
    // Change the PM20 bit when the PLC07 bit is set to "0"(PLL Off).
    // Set the PM20 bit to "0"(2 waits) when PLL clock > 16MHz.

```

```
//=====
// Setting Voltage Down Detection Interrupt Register
//=====
d4int = 0x03;
// (b0) Voltage Down Detection Interrupt Enable Bit [1:Enable] (Note5)
// (b1) STOP/WAIT Mode Deactivation Control Bit [1:Enable]
// (b2) Voltage Change Detection Flag [0:(Writing a "1" has no effect)]
// (b3) WDT Overflow Detect Flag [0:(Writing a "1" has no effect)]
// (b5-b4) Sampling Clock Select Bit [00:CPU clock divided by 8]
// (b7-b6) Nothing is assigned(When write, set to "0") [00]
// (Note5) The D40 bit is effective when the VC27 bit=1.
// To set the D40 bit to "1", set bits in the following order.
// (a) Set the VC27 bit to "1".
// (b) Wait for td(E-A) until the detection circuit is actuated.
// (c) Wait for the sampling time
// (d) Set the D40 bit to "1".

// Main clock for the CPU clock source
if( cm11 == 0 ){

//=====
// Using the PLL clock as the clock source for the CPU
//=====
// System Clock Control Register 1
cm1 = 0x20;
// (b0) All Clock Stop Control Bit [0:Clock on]
// (b1) System Clock Select Bit 1 [0:Main clock] (Note5)
// (b4-b2) Reserved Bit (Set to "0") [000]
// (b5) XIN-XOUT Drive Capacity Select Bit [1:HIGH]
// (b7-b6) Main Clock Division Select Bit [00:No division mode]
// (Note5) After setting the PLC07 bit in the PLC0 register to "1"(PLL operation),
// wait until Tsu(PLL) elapses before setting the CM11 bit to "1"(PLL Clock)

// System Clock Control Register 0
cm0 = 0x04;
// (b1-b0) Clock Output Function Select Bit [00: I/O port P5_7]
// (b2) WAIT Mode Peripheral Function Clock Stop Bit
// [1:Stop peripheral function clock in wait mode] (Note8)
// (b3) XCIN-XCOUT Drive Capacity Select Bit [0:Low]
// (b4) Port XC Select Bit [0:I/O port P8_6, P8_7]
// (b5) Main Clock Stop Bit [0:On]
// (b6) Main Clock Division Select Bit [0:CM16 and CM17 valid]
// (b7) System Clock Select Bit [0:Main clock, PLL clock, or on-chip oscillator clock]
// (Note8) The fC32 clock does not stop. During low speed or low power dissipation mode,
// do not set this bit to "1" (peripheral clock turned off when in wait mode).

// Set PLL Control Register 0
plc0 = 0x92;
// (b2-b0) PLL Multiplying Factor Select Bit [010:Multiply by 4]
// (b3) Nothing is assigned (When write, set to "0") [0]
// (b4) Reserved Bit (Set to "1") [1]
// (b6-b5) Reserved Bit (Set to "0") [00]
// (b7) Operation Enable Bit [1:PLL On]

// Wait until the PLL clock becomes stable (tsu(PLL))
cnt_word = CNT_WAIT_PLL; while( cnt_word -- );

// Set the CM11 bit to "1" (Use the PLL clock as the CPU clock)
cm11 = 1;
// (b1) System Clock Select Bit 1 [1:PLL clock]

}

prcr = 0x00; // Protect Register (write protected)
}
```

```

/*****
MODULE : init_Peripheral
FUNCTION: Initialize Peripheral
PARAMETERS: None
RETURN : None
*****/
*/
void init_Peripheral( void )
{
    //=====
    // Setting I/O port
    //=====
    NEGATE_LEDS; // Negate tens and first digit LED
    LED_PORT = LED_data_blank; // Initialize LED Port
    LED_PD = 0xff; // LED Port Direction
    LED_SELECTOR_PD |= 0x03; // LED Selector Port Direction

    //=====
    // Peripheral Clock Select Register
    //=====
    prcr = 0x01; // Protect Register (write enabled)
    // Peripheral Clock Select Register
    pclkr = 0x03;
    // (b0) Timers A, B Clock Select Bit [1:f1]
    // (b1) SI/O Clock Select Bit [1:f1SI0]
    // (b7-b2) Reserved bit (Set to "0") [000000]
    prcr = 0x00; // Protect Register (write protected)

    //=====
    // Timer A0 (Timer Mode, 1ms)
    //=====
    // Stop counting
    ta0s = 0;
    // Timer Mode Register
    ta0mr = 0x00;
    // (b1-b0) Operation Mode Select Bit [00:Timer mode]
    // (b2) Pulse Output Function Select Bit [0:Pulse is not output]
    // (b4-b3) Gate Function Select Bit [00:Gate function not available]
    // (b5) Set to "0" in timer mode [0]
    // (b7-b6) Count Source Select Bit [00:f1 or f2]
    // Interrupt Priority Level (Interrupt disable)
    ta0ic=0;
    // Timer Register
    ta0 = TA0_INTERVAL;
}

```



```

/*****
MODULE : refresh_LED
FUNCTION: Dynamically turn on LED and step up digit by 1sec
PARAMETERS: None
RETURN : None
*****/
*/
void refresh_LED( void )
{
    static short sec_cnt = 0;    // second counter

    // Dynamically turn on LED
    LED_PORT = LED_data_blank;
    if( FIRST_LED == ACTIVE_LED ){
        NEGATE_LEDS;
        LED_PORT = LED_data[ Digit_10 ];
        TENS_LED = ACTIVE_LED;
    }
    else{
        NEGATE_LEDS;
        LED_PORT = LED_data[ Digit_1 ];
        FIRST_LED = ACTIVE_LED;
    }

    // Digit is stepped up by 1sec
    if ( (1000/REFRESH_INTERVAL) < sec_cnt++){
        sec_cnt = 0;
        Digit_1++;
        if( 10 <= Digit_1 ){
            Digit_1 = 0;
            Digit_10++;
            if( 10 <= Digit_10 ) Digit_10 = 0;
        }
    }
}
}

```

6.4. A_interrupt.a30

サンプルプログラムの電圧低下検出割り込み処理を以下に示します。
サンプルプログラムを動作させる場合、以下の値は必要に応じて設定してください。

定義	内容
CNT_LOOP_MAX	トータルチェック回数 (VC13 ビットチェック回数の上限値)
CNT_OVER_VDET4	VC13 ビット(電圧低下モニタフラグ)カウント回数 VC13 ビットが連続 CNT_OVER_VDET4(回)"1"で安定した場合、VCC1 > Vdet4 と判定します。
CNT_UNDER_VDET4	VC13 ビットが連続 CNT_UNDER_VDET4(回)"0"で安定した場合、VCC1 < Vdet4 と判定します。
CNT_WAIT_CHK	チェック間隔 (チェック間隔は CNT_WAIT_CHK(回)ループするループ処理で作っています)
CNT_WAIT_PLL	ウェイトモードからの復帰時に CPU クロック源を PLL クロックに戻す際の PLL 発振安定待ち時間 tsu(PLL)を作るためのループ回数を定義しています。

```

;*****
;
;
; FILE      : A_interrupt.a30
; CONTENTS  : Sample operation of Voltage Detection Circuit
;             Interrupt handling routine
;
; CPU       : M16C/62P (Starter Kit for M16C/62P :M3A-0644)
; OS        : No used
; COMMENT   :
; HISTORY   : 2005.03.15 Ver.1.00
;
; NOTE      :
;
; Copyright(C)2004, Renesas Technology Corp.
; Copyright(C)2004, Renesas Solutions Corp.
; All rights reserved.
;*****
;
;*****
;
; Include files
;*****
;
; .list OFF
; .include sfr62p.inc
; .list ON
;
;*****
;
; Defines
;*****
;
; Set the values suitable for the system
CNT_LOOP_MAX .equ (20)           ; Number of total check
CNT_OVER_VDET4 .equ (8)         ; Number of times that check VCC1 above Vdet4
CNT_UNDER_VDET4 .equ (8)        ; Number of times that check VCC1 below Vdet4
CNT_WAIT_CHK .equ (700)         ; Check interval Approx. 1ms @6MHz
CNT_WAIT_PLL .equ (14000)       ; Number of the PLL Frequency Synthesizer Stabilization Wait Time
;
; Definitions for the Starter Kit for M16C/62P(M3A-0644)
LED_PORT .equ p0
LED_PD .equ pd0
LED_SELECTOR .equ p1
LED_SELECTOR_PD .equ pd1
LED_data_blank .equ ~(00h)
LED_data_E .equ ~(01h | 20h | 10h | 08h | 40h)
NEGATE_TENS .equ (01h)
NEGATE_FIRST .equ (02h)
ASSERT_TENS .equ ~(NEGATE_TENS)
ASSERT_FIRST .equ ~(NEGATE_FIRST)
ASSERT_LEDS .equ (ASSERT_TENS & ASSERT_FIRST)
NEGATE_LEDS .equ (NEGATE_TENS | NEGATE_FIRST)

```

```

*****
;
; Global symbols
;*****
.glb voltage_detect

;*****
; Variables
;*****
.section bss_NE, DATA
Cnt_under_Vdet4: .blkw 1
Cnt_over_Vdet4: .blkw 1
Cnt_wait: .blkw 1
Cnt_tmp: .blkw 1

.section program, align
;*****
; MODULE : voltage_detect
; FUNCTION: Interrupt handling routine of Voltage Detect Circuit.
;
; NOTE : This routine does not use any registers.
; Except for automatically saved registers, if there are any registers
; that are likely to be modified in the interrupt handling routine,
; save them to the stack in software.
;
; PARAMETERS: None
; RETURN : None
;*****
voltage_detect:
;=====
; Change CPU clock source from the PLL clock to the main clock
;=====
; If the main clock is beyond 10MHz, be sure below 10MHz by dividing.
; Protect Register (write enabled)
mov.b #01h, prcr
; System Clock Control Register 1
mov.b #020h, cm1
; (b0) All Clock Stop Control Bit [0:Clock on]
; (b1) System Clock Select Bit 1 [0:Main clock]
; (b4-b2) Reserved Bit (Set to "0") [000]
; (b5) XIN-XOUT Drive Capacity Select Bit [1:HIGH]
; (b7-b6) Main Clock Division Select Bit [00:No division mode]
; System Clock Control Register 0
mov.b #004h, cm0
; (b1-b0) Clock Output Function Select Bit [00: I/O port P5_7]
; (b2) WAIT Mode Peripheral Function Clock Stop Bit
; [1:Stop peripheral function clock in wait mode]
; (b3) XCIN-XCOUT Drive Capacity Select Bit [0:Low]
; (b4) Port XC Select Bit [0:I/O port P8_6, P8_7]
; (b5) Main Clock Stop Bit [0:On]
; (b6) Main Clock Division Select Bit [0:CM16 and CM17 valid]
; (b7) System Clock Select Bit [0:Main clock, PLL clock, or on-chip oscillator clock]

```

```

;=====
; Check Voltage Down Monitor Flag
;=====
mov.w #CNT_LOOP_MAX,      Cnt_tmp
mov.w #CNT_UNDER_VDET4,   Cnt_under_Vdet4
mov.w #CNT_OVER_VDET4,    Cnt_over_Vdet4

CHK_VCC_FLG:
  mov.w #CNT_WAIT_CHK,    Cnt_wait
WAIT_CHK:
  adjnz.w #-1, Cnt_wait,   WAIT_CHK

  ; Check VC13 (Voltage Down Monitor Flag)
  btst vc13
  jnz   CHK_VCC_UP

CHK_VCC_DOWN:
  mov.w #CNT_OVER_VDET4,   Cnt_over_Vdet4 ; Reset Over Vdet4 counter
  adjnz.w #-1, Cnt_over_Vdet4, CHK_LOOP_MAX
  jmp   UNDER_VDET4

CHK_VCC_UP:
  mov.w #CNT_UNDER_VDET4,  Cnt_under_Vdet4 ; Reset Under Vdet4 counter
  adjnz.w #-1, Cnt_under_Vdet4, CHK_LOOP_MAX
  jmp   OVER_VDET4

CHK_LOOP_MAX:
  adjnz.w #-1, Cnt_tmp,    CHK_VCC_FLG
  jmp   HANDLE_EXCEPTION

;=====
; Entering Wait Mode
;=====
; When the CPU clock source is the PLL clock, be sure CPU clock source is
; the main clock before going wait mode.
UNDER_VDET4:

  ; Stop PLL
  bclr plc07
  mov.b #00h, prcr ; Protect Register (write protected)

  ; Reduce Power Consumption
  mov.b #LED_data_blank, LED_PORT
  or.b #NEGATE_LEDS, LED_SELECTOR
  and.b #11111100b, LED_SELECTOR_PD ; LED Selection Port direction --> Input
  mov.b #00h, LED_PD ; LED Port direction --> Input

  jmp.b _LABEL_001 ; (Reference: TECHNICAL UPDATE) TN-16C-128A
_LABEL_001:
  ; Insert the jmp.b instruction between the writing instruction to
  ; the internal RAM area and wait instruction.

  wait ; Execute WAIT instruction
  nop
  nop
  nop
  nop

  jmp END_VOLTAGE_DETECT

```

```

;=====
;   Exiting Wait Mode
;=====
OVER_VDET4:
; Set PLL Control Register 0
mov.b #092h, plc0
; (b2-b0) PLL Multipling Factor Select Bit      [010:Multiply by 4]
; (b3) Nothing is assigned (When write, set to "0") [0]
; (b4) Reserved Bit (Set to "1")                [1]
; (b6-b5) Reserved Bit (Set to "0")            [00]
; (b7) Operation Enable Bit                    [1:PLL On]

; Wait until the PLL clock becomes stable
mov.w #CNT_WAIT_PLL, Cnt_tmp
WAIT_PLL_STABLE1:
adjnz.w #-1, Cnt_tmp, WAIT_PLL_STABLE1

; Use the PLL clock as the CPU clock
bset cm11 ; (b1) System Clock Select Bit 1 [1:PLL clock]
mov.b #00h, prcr ; Protect Register (write protected)

; Restore LED Port
mov.b #LED_data_blank, LED_PORT
or.b #NEGATE_LEDS, LED_SELECTOR
mov.b #0FFh, LED_PD
or.b #00000011b, LED_SELECTOR_PD

END_VOLTAGE_DETECT:
; Clear Voltage Change Detection Flag
bset prc3
bclr d42
bclr prc3
reit

;=====
;   Handle Exception
;=====
; At this sample, If an exception occurred,
; Disable Voltage Down Detection Interrupt and stop processing by entering WAIT mode.
HANDLE_EXCEPTION:
; Disable Voltage Down Detection Interrupt
mov.b #09h, prcr ; Protect Register (write enabled)
bclr d40 ; (b0) Voltage Down Detection Interrupt Enable Bit [0:Disabled]
bclr plc07 ; (b7) Operation Enable Bit [0:Stop PLL]
mov.b #00h, prcr ; Protect Register (write protected)

; Display to LED the exception
or.b #NEGATE_LEDS, LED_SELECTOR
mov.b #LED_data_E, LED_PORT
mov.b #0FFh, LED_PD
or.b #00000011b, LED_SELECTOR_PD
and.b #ASSERT_LEDS, LED_SELECTOR

; Stop processing by entering WAIT mode
jmp.b _LABEL_002 ; (Reference: TECHNICAL UPDATE) TN-16C-128A
_LABEL_002:
; Insert the jmp.b instruction between the writing instruction to
; the internal RAM area and wait instruction.

wait ; Execute WAIT instruction
nop
nop
nop
nop

LATCH_LOOP:
jmp.b LATCH_LOOP

.END

```

7. 参考ドキュメント

ハードウェアマニュアル
M16C/62P グループハードウェアマニュアル Rev.2.30
(最新版をルネサス テクノロジホームページから入手してください。)

テクニカルアップデート/テクニカルニュース
(最新の情報をルネサス テクノロジホームページから入手してください。)

8. ホームページとサポート窓口

ルネサス テクノロジホームページ
<http://www.renesas.com/jpn/>

M16C ファミリ MCU 技術サポート窓口
E-mail: csc@renesas.com

Appendix A. サンプルプログラム関連レジスタ

本サンプルプログラムは、以下のハードウェアマニュアルを参照しています。

・参照ハードウェアマニュアル：M16C/62P グループハードウェアマニュアル Rev.2.30

各レジスタの仕様や機能については、必ず最新のハードウェアマニュアルを参照、確認してください。

A.1. 電圧検出レジスタ1

電圧検出レジスタ1

		シンボル VCR1	アドレス 0019h番地	リセット後の値(注2) 00001000b
ビット シンボル	ビット名	機能		RW
- (b2-b0)	予約ビット	"0" にしてください。		RW
VC13	電圧低下モニタフラグ(注1)	0 : VCC1 < Vdet4 1 : VCC1 >= Vdet4		RO
- (b7-b4)	予約ビット	"0" にしてください。		RW

注1. VCR2レジスタのVC27ビットが"1" (電圧低下検出回路有効) のとき、VC13ビットは有効です。
VCR2レジスタのVC27ビットが"0" (電圧低下検出回路無効) のとき、VC13ビットは"1" (VCC1 >= Vdet4) になります。

注2. ソフトウェアリセット、ウォッチドッグタイマリセット、発振停止検出リセット時は変化しません。

A.2. 電圧検出レジスタ2

電圧検出レジスタ2(注1)

		シンボル VCR2	アドレス 001Ah番地	リセット後の値(注5) 00h
ビット シンボル	ビット名	機能		RW
- (b5-b0)	予約ビット	"0" にしてください。		RW
VC26	リセット領域監視ビット (注2、3、6)	0 : リセット領域検出回路無効 1 : リセット領域検出回路有効		RW
VC27	電圧低下監視ビット(注4、6)	0 : 電圧低下検出回路無効 1 : 電圧低下検出回路有効		RW

注1. このレジスタはPRCRレジスタのPRC3ビットを"1" (書き込み許可) にした後で書き換えてください。

注2. 電圧低下検出リセット(ハードウェアリセット2)を使用する場合、VC26ビットを"1" (リセット領域検出回路有効) にしてください。

注3. ストップモード時、VC26ビットは無効です(VCC1端子の入力電圧がVdet3より低くなってもリセットされません)。

注4. VCR1レジスタのVC13ビット、D4INTレジスタのD42ビットを使用する場合、またはD40ビットを"1" (電圧低下検出割り込み許可) にする場合、VC27ビットを"1" (電圧低下検出回路有効) にしてください。

注5. ソフトウェアリセット、ウォッチドッグタイマリセット、発振停止検出リセット時は変化しません。

注6. VC26ビットまたはVC27ビットを"1" にした後、td(E-A)経過してから検出回路が動作します。

A.3. 電圧検出割り込みレジスタ

電圧低下検出割り込みレジスタ(注1)

シンボル	アドレス	リセット後の値	
D4INT	001Fh番地	00h	
ビットシンボル	ビット名	機能	RW
D40	電圧低下検出割り込み許可ビット(注5)	0 : 禁止 1 : 許可	RW
D41	STOP解除制御ビット(注4)	0 : 無効(電圧低下検出割り込みをストップモードからの復帰に使用しない) 1 : 有効(電圧低下検出割り込みをストップモードからの復帰に使用する)	RW
D42	電圧変化検出フラグ(注2)	0 : 未検出 1 : Vdet4通過検出	RW (注3)
D43	WDTオーバフロー検出フラグ	0 : 未検出 1 : 検出	RW (注3)
DF0	サンプリングクロック選択ビット	b5 b4 0 0 : CPUクロックの8分周 0 1 : CPUクロックの16分周 1 0 : CPUクロックの32分周 1 1 : CPUクロックの64分周	RW
DF1			RW
- (b7-b6)	何も配置されていない。書く場合、“0”を書いてください。読んだ場合、その値は“0”。		-

注1 . このレジスタはPRCRレジスタのPRC3ビットを“1”(書き込み許可)にした後で書き換えてください。

注2 . VCR2レジスタのVC27ビットが“1”(電圧低下検出回路有効)のとき有効。VC27ビットを“0”(電圧低下検出回路無効)にすると、D42ビットは“0”(未検出)になります。

注3 . プログラムで“0”を書くと、“0”になります(“1”を書いても変化しません)。

注4 . 電圧低下検出割り込みをストップモードからの復帰に使用した後、再度、復帰に使用する場合、D41ビットに“0”を書き込み後、“1”を書き込んでください。

注5 . D40ビットは、VCR2レジスタのVC27ビットが“1”のとき有効です。

D40ビットを“1”にする場合は次の手順で設定してください。

- (1) VC27ビットを“1”にする
- (2) 検出回路が動作するまでtd(E-A)待つ
- (3) サンプリング時間(「表6.2 サンプリング時間」参照)待つ
- (4) D40ビットを“1”にする

A.4. システムクロック制御レジスタ 0

システムクロック制御レジスタ0(注1)

シンボル	アドレス	リセット後の値	
CM0	0006h番地	01001000b	
ビットシンボル	ビット名	機能	RW
CM00	クロック出力機能選択ビット (シングルチップモード時のみ有効)	b1 b0 0 0 : 入出力ポートP5_7 0 1 : fCを出力 1 0 : f8を出力 1 1 : f32を出力	RW
CM01			RW
CM02	ウェイトモード時周辺機能クロック停止ビット(注10)	0 : ウェイトモード時、周辺機能クロック停止しない 1 : ウェイトモード時、周辺機能クロック停止する(注8)	RW
CM03	XCIN-XCOUT駆動能力選択ビット(注2)	0 : Low 1 : High	RW
CM04	ポートXC切り替えビット(注2)	0 : 入出力ポートP8_6、P8_7 1 : XCIN-XCOUT発振機能(注9)	RW
CM05	メインクロック停止ビット(注3、10、12、13)	0 : 発振 1 : 停止(注4、5)	RW
CM06	メインクロック分周比選択ビット0(注7、13、14)	0 : CM16、CM17ビット有効 1 : 8分周モード	RW
CM07	システムクロック選択ビット(注6、10、11、12)	0 : メインクロック、PLLクロック またはオンチップオシレータクロック 1 : サブクロック	RW

- 注1. このレジスタは、PRCRレジスタのPRC0ビットを“1”(書き込み許可)にした後で書き換えてください。
- 注2. CM04ビットを“0”(入出力ポート)の間、またはストップモードへ移行したとき、CM03ビットは“1”(HIGH)になります。
- 注3. このビットは低消費電力モード、またはリオンチップオシレータ低消費電力モードにするときに、メインクロックを停止させるためのビットです。メインクロックが停止したかどうかの検出には使用できません。メインクロックを停止させる場合、次のようにしてください。
- (1) サブクロックが安定して発振している状態で、CM07ビットを“1”(サブクロック選択)にする、またはCM2レジスタのCM21ビットを“1”(オンチップオシレータ選択)にする
 - (2) CM2レジスタのCM20ビットを“0”(発振停止、再発振検出機能無効)にする
 - (3) CM05ビットを“1”(停止)にする
- 注4. 外部クロック入力時は、0”(発振)にしてください。
- 注5. CM05ビットが“1”の場合、XOUT端子は“H”になります。また、内蔵している帰還抵抗は接続したままですの、端子は帰還抵抗を介して、XOUT(“H”)にプルアップされた状態となります。
- 注6. CM04ビットを“1”(XCIN-XCOUT発振機能)にし、サブクロックの発振が安定した後に、CM07ビットを“0”から“1”(サブクロック)にしてください。
- 注7. 高速モード、中速モード、オンチップオシレータモード、またはオンチップオシレータ低消費電力モードからストップモードへの移行時、CM06ビットは“1”(8分周モード)になります。
- 注8. fC32は停止しません。低速モードまたは低消費電力モード時は“1”(ウェイトモード時、周辺機能クロック停止する)にしないでください。
- 注9. サブクロックを使用する場合、このビットを“1”にしてください。また、ポートP8_6、P8_7は入力ポートで、プルアップなしにしてください。
- 注10. PM2レジスタのPM21ビットが“1”(クロック変更禁止)の場合、CM02、CM05、CM07ビットに書いても変化しません。
- 注11. PM21ビットを“1”にする場合、CM07ビットを“0”(メインクロック)にした後で、PM21ビットを“1”にしてください。
- 注12. CPUクロックのクロック源をメインクロックにする場合、次のようにしてください。
- (1) CM05ビットを“0”(発振)にする。
 - (2) td(M-L)またはメインクロック発振安定時間のうち長い方を待つ
 - (3) CM11ビットを“0”、CM21ビットを“0”、CM07ビットを“0”にする。
- 注13. CM21ビットが“0”(オンチップオシレータ停止)、CM05ビットが“1”(メインクロックを停止)のとき、CM06ビットが“1”(8分周モード)、CM15ビットが“1”(駆動能力HIGH)に固定されます。
- 注14. オンチップオシレータモードから高速、中速モードに戻すときは、CM06ビットを“1”、CM15ビットを“1”にしてください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.03.15	-	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。