

# AVR132 : 強化型ウォッチドッグ タイマの使用法

## 要点

- ウォッチドッグ システム リセット元
- ウォッチドッグ システム リセットに先行する予備パラメータ保存
- 全休止形態からの起動タイマ
- 起動復帰とシステム リセット両用のウォッチドッグ 使用法
- ウォッチドッグ リセット フラグ(WDRF)の扱い
- ウォッチドッグ 設定変更
- ウォッチドッグ 操作用流れ図
- ソースコード例

## 1. 序説

“良く設計されたウォッチドッグ タイマは毎日発生し、素早くシステムを救い、人間の英雄への敬意を表すことなく生き永らえる。”・・・Jack Ganssle

非常に小さく減じられ、ソフトウェアの欠片もなければ、バグから開放されます。その応用は無制限繰り返しで止まって動かなくなり得ます。予期せぬ誤りコードは、正しく取り扱われない場合に深刻な問題を引き起こし得ます。電気的雑音や外部事象の異常な流れ順は設計者によって予測されていない状態にシステムを至らせ得ます。これらの全ての場合が潜在的にシステムの永久停止、または周辺への深刻な障害を引き起こし得ます。このような場合の自動的な取り扱いと回復がウォッチドッグ タイマの仕事です。

強化型ウォッチドッグ タイマ(WDT)は時間経過で必ずシステムリセットを発生し、システムの残り(WDT以外)とは独立して動作します。けれども、ソフトウェアが既知の健全な状態である限り、応用ソフトウェアはWDTを周期的にリセットすることによって、時間経過が決して起きないことを保証すべきです。システムが停止、またはプログラム実行が不正にされた場合、WDTは周期的なリセットを受け取らず、結局、時間超過でシステムリセットを引き起こすでしょう。

新しいAVRデバイスの全てがデバイスをリセットする代わりに割り込みを生成する能力を持ちます。WDTが自身の独立したクロックで走行するため、全ての休止形態からAVRを起動復帰するのに使用できます。これはシステムリセット元としての通常の動作と容易に組み合わせて、WDTを理想的な起動タイマにします。この割り込みは近づいているウォッチドッグ システムリセットの早期警告を得るのにも使用でき、故に重要なパラメータを不揮発性メモリに予備保存できます。

## 2. 動作の理屈

強化型ウォッチドッグ タイマ(WDT)周期が経過してしまうと、WDT時間経過が起きます。この時間経過周期は定係数によってWDT発振器クロックを分周する、設定可能な前置分周器を使用して調節されます。**WDR**(ウォッチドッグ リセット)命令は、このタイマ値をリセット(=0)します。WDTを使用する応用ソフトウェアは、システムが未だ正しく動作していると判断する限り必ず周期的に**WDR**命令を実行するように設計されなければなりません。このタイマ値はシステムリセットとWDTの禁止時、自動的にリセット(=0)されます。

強化型ウォッチドッグ タイマには3つの動作種別があります。WDTシステムリセット動作時、WDT時間経過はシステムリセットを引き起こします。WDT割り込み動作で全割り込み(ビット)が許可されているなら、WDT時間経過はシステムをリセットする代わりに、WDT割り込み要求フラグを設定(1)してWDT割り込み処理ルーチンを実行します。WDTシステムリセット動作とWDT割り込み動作の両方が許可されているなら、最初のWDT時間経過はWDT割り込み動作だけが許可された場合のように扱われます。そしてWDT割り込み動作が自動的に禁止され、WDTはWDTシステムリセット動作だけに戻ります。

次頁の図2-1はWDT時間経過発生時に何が起こるかを示します。破線枠項目はシステムによって実行される動きを記述します。実線枠項目は応用によって実行されるべき動きを記述します。

強化型ウォッチドッグ タイマ使用時、ウォッチドッグ常時ON(WDTON)ヒューズがプログラム(0)されている場合に、可能な動作種別がWDTシステムリセット動作だけなのを知ることが大事です。この安全機能はWDTシステムリセット機能を禁止し得る、予期せぬWDT割り込み動作許可からソフトウェアを守ります。WDTONヒューズが非プログラム(1)にされると、本資料で記述されるように、WDT割り込み動作が使用できます。

上記のようにWDTはシステムの残りから独立しています。これはWDT動作種別の1つが許可されている限り走行する、それ自身の内部128kHz発振器を持ちます。これは例えばCPU発振器が役に立たなくても、安全な動作を保証します。



8ビット **AVR**<sup>®</sup>  
マイクロコントローラ

## 応用記述

本書は一般の方々の便宜のため有志により作成されたもので、ATMEL社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 2551C-06/08, 2551CJ3-12/13

例えばソフトウェア設計者がWDT使用を決して意図していなくても、予期せず、例えば位置指示子逸脱や低電圧(ブラウンアウト)状態によって、それが許可され得ることに注意してください。WDTが意図せず許可された場合、WDTによって起されたシステムリセットはWDTを**禁止しない**ので、ファームウェアが禁止するまで(WDTは)許可のままでしょう。WDTによって起されないシステムリセットはヒューズ設定に従ってWDTを既定形態設定に再初期化します。WDTシステムリセットの場合でのこのWDT自動“再許可”は、信頼できるウォッチドッグ機能を保証するための安全機能です。従って、始動コードは例えば応用がWDTを使用していない場合、常にリセットフラグを調べ、WDTシステムリセットが起きていた場合に適切な動作を行うべきです。

各種の目的に対してWDTを使用するために、様々な設定と機能が組み合わせられます。最も重要な設定は以降の項で記述されます。

システムリセット元としてだけ動作するようにWDTを設定するのは簡単です。WDTシステムリセット動作を許可し、合理的な時間経過遅延を設定してください。初期化ルーチンがWDT時間経過周期よりも長くかかる場合、実行中の適切な検査点でWDR命令を実行すべきです。これを行わない場合、WDTがシステムをリセットする前にコードは決して主ルーチンへ到達しないでしょう。

時間経過周期は応用の主繰り返しを通した最長可能性実行経路よりも長くなるように選ばなければなりません。これは更に予測される割り込み処理も含まれます。主繰り返しが非常に大きな場合、より短い時間経過周期を許すために多数の検査点が繰り返しの内側に挿入され得ます。

正しい時間経過周期の選択には主繰り返しのタイミング特性の詳細情報が必要です。多くの応用で、この最も体力を要する接近作業は数秒の時間経過周期を選択することができます。これはシステムが無限繰り返して動作を止めた場合に最低限、システムをリセットするでしょう。

殆どの組み込みシステムはいくつかの初期化コードと主繰り返しから成ります。この構造はウォッチドッグ使用に対して最も効果的な設定でもあります。このようなシステムでのWDT使用例は図2-2.で示されます。

時間経過周期が非常にきつく選択された場合に、普通でない割り込み数がWDTシステムリセットを引き起こし得ることに注意してください。これは時間経過周期選択時に考慮が払われなければなりません。

繰り返しの最後の'全てOK?'検査は応用が正しく動作しているか否かのどちらかを定める、繰り返しの一部です。1つの方法は、'良好'またはコードの重要部分が訪れたことを示すために主繰り返しの各部分で設定するフラグを使用することです。最終試験は全フラグを検査し、全てがOKならばWDTとフラグをリセットします。そうでなければ結果的に時間経過が起こるでしょう。

初期化コードはWDTリセットフラグ(WDRF)を検査し、適切な動作を行うべきです。これは「始動での考慮」項でより詳細に網羅されます。

図2-1. WDT時間経過発生時の事象の流れ

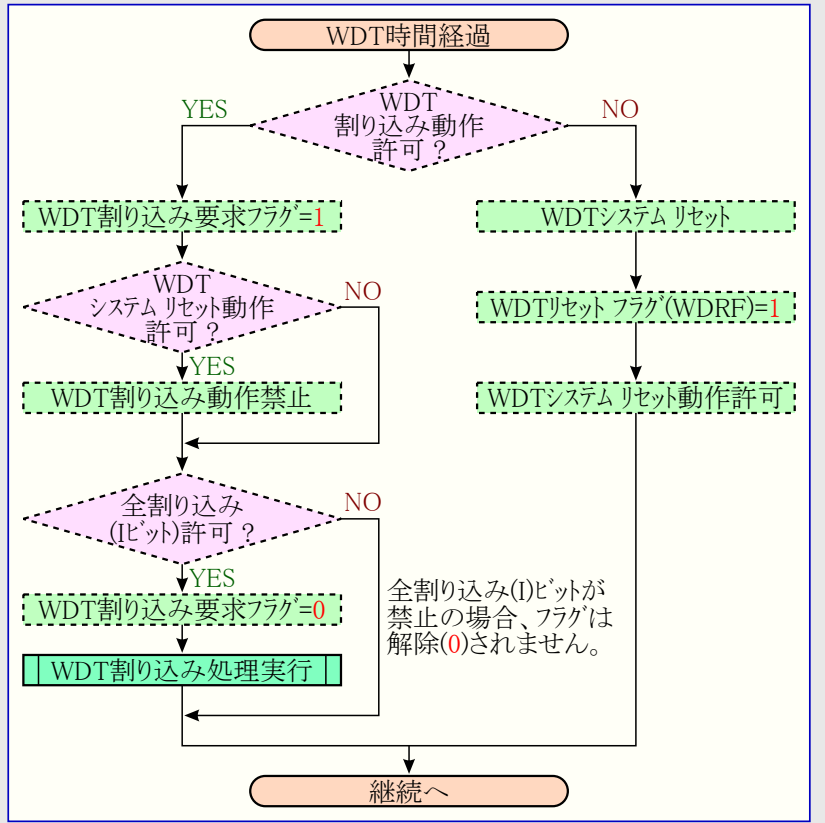
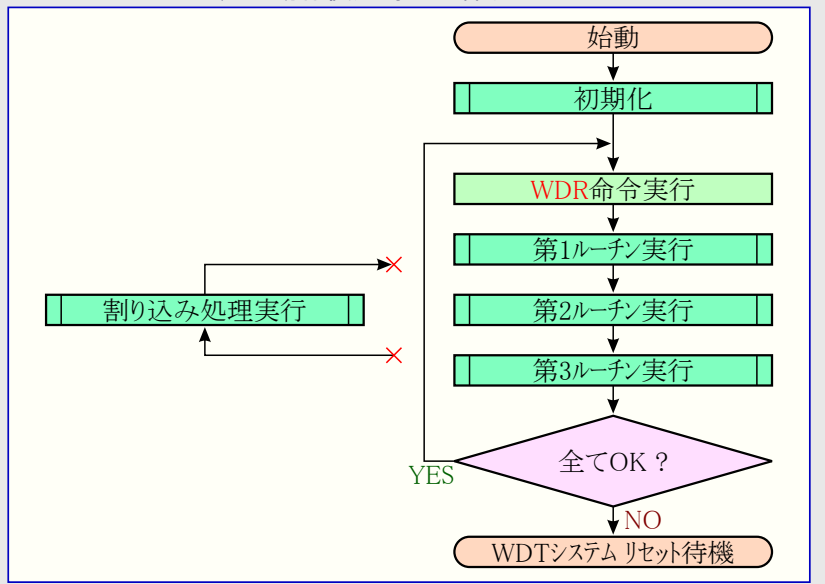


図2-2. WDTシステムリセット動作使用時の主繰り返し



## 2.1. WDTシステム リセットに先行する予備パラメータ保存

直前項で記述された方法はWDTシステム リセットが来ることのどんな警告も与えません。応用はシステム リセット発生前にソフトウェアで時間経過を扱うことの意味を持ちません。けれども、WDT割り込み動作を使用することによって、応用は実際のリセットに先行して重要なパラメータを予備保存するのにWDT割り込み処理ルーチンを使用できます。

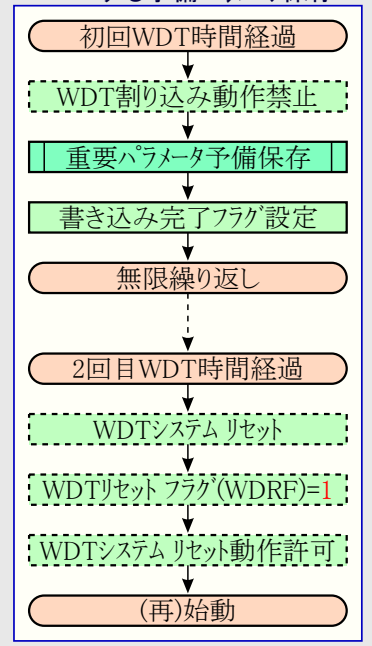
WDTシステム リセット動作とWDT割り込み動作の両方を許可することにより、最初の時間経過はWDT割り込み動作を禁止して割り込み処理ルーチンを走行します。そして2回目の時間経過がシステム リセットを起します。それでこの割り込み処理ルーチンはパラメータの(例えばEEPROMへの)予備保存に対して時間経過1周期(分の時間)を持ちます。事象の流れは図2-3.で示されます。破線枠項目はシステムによって実行される動きを記述します。実線枠項目は応用によって実行されるべき動きを記述します。

書き込み完了フラグはシステム リセット前に予備保存操作が完了されたかどうかを示すEEPROM内のビットで有り得ます。このフラグは始動コードでWDTリセット フラグ(WDRF)が設定(1)されている場合に、予備保存パラメータがシステム状態回復またはデバッグ目的に使用できるかを調べることができます。このフラグは他の形式のリセットが起きる場合に、パラメータを無効にするために初期化中に解除(0)されるべきです。

この割り込み処理ルーチンがWDTシステム リセットに先行して実行されるのを保証されないことに注意してください。長すぎる割り込み禁止の場合、割り込み処理ルーチンは2回目の時間経過前に決して実行されないでしょう。位置指示子逸脱や電氣的雑音も予期せぬWDT割り込み動作禁止を行い得ます。従ってこの書き込み完了フラグは格納されたパラメータが有効か無効かを知る手段です。

割り込み処理ルーチンの最後の無限繰り返しは潜在的に引き起こす、より多くの損傷から主コードを守ります。

図2-3. WDTシステムリセットに先行する予備パラメータ保存



## 2.2. WDT割り込み動作の使用法

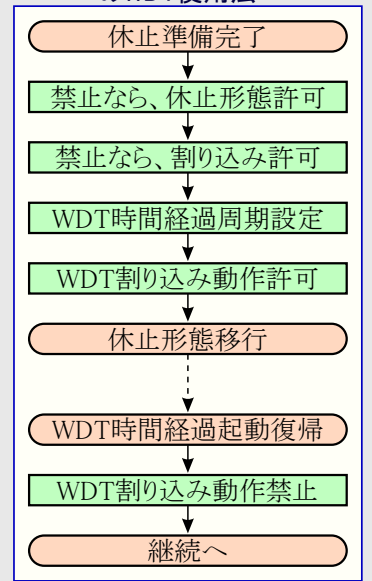
前で記述されたようにWDTはCPU主クロックから独立して走行する自身の内部発振器を持ちます。これは全休止形態からの起動復帰元としてWDT割り込み使用を可能にします。WDT割り込み動作だけを許可することによって、時間経過は割り込み要求を生成しますが、更なる時間経過でどんなシステム リセットも引き起こしません。

CPU主クロックの走行なしの起動復帰元を持つことは節電の優れた方法です。起動復帰元としてのWDTと共にパワーダウン休止形態を使用すると、供給電圧3V時に概ね3 $\mu$ A流れます。起動復帰元としてWDTを使用する方法の例は図2-4.で示されます。

周期的な起動復帰が好まれるなら、WDT割り込み動作禁止を取り去ることで行えます。これでWDTはCPUが休止形態の場合にCPUを起動復帰する、時間経過毎に割り込みを生成します。

単に起動復帰計時器としてのWDT使用時、WDTシステム リセット動作が決して許可されてはいけないことに注意してください。許可された場合、次の時間経過でシステム リセットが起こるでしょう。起動復帰計時器とシステム リセット元の両方としてのWDT使用は後続項で記述されます。

図2-4. 起動復帰計時器としてのWDT使用法





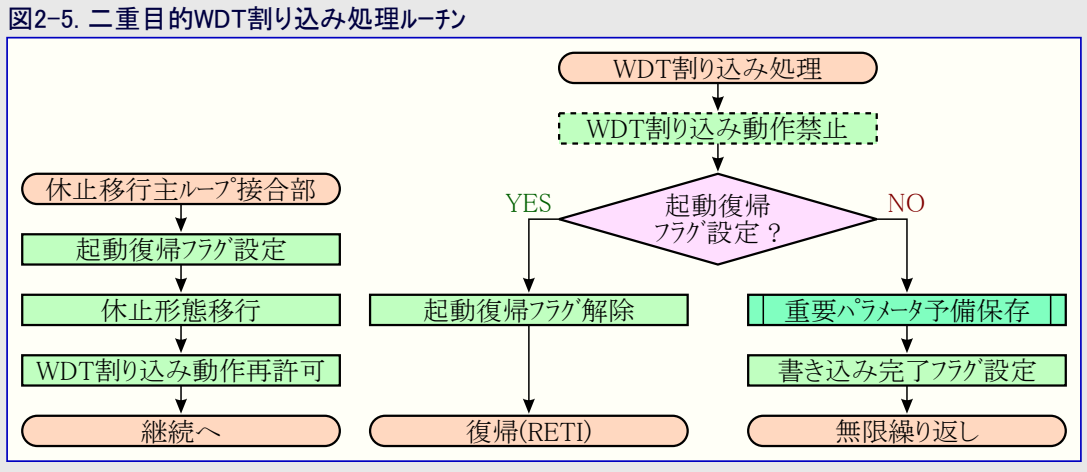
### 2.3. 組み合わせ動作でのWDT使用法

休止形態移行時に起動復帰計時器として働き、活動動作に戻った時にWDTシステムリセット動作へ切り替えるようにWDTを設定するのも可能です。この設定ではハードウェアによって自動的に禁止されるので、WDT割り込み動作を禁止する必要はありません。従って、周期的な起動復帰元としてWDTを使用するには、休止形態への移行に先立って、毎回WDT割り込み動作を許可しなければなりません。

割り込み処理ルーチン内でのWDT割り込み動作再許可は、コードのいくつかの部分が不正な場合にWDT割り込み動作内での動作停止をWDTに引き起こすので推奨されません。

CPUが活動動作に戻ると、始めの方で記述されるように主繰り返し内でWDTをリセットするのにWDR命令が使用されます。WDT割り込み動作禁止では、WDTが丁度起動復帰機能なしでのように機能します。

パラメータの予備保存等のためにシステムリセットに先行する時間経過警告が必要とされる場合、WDT割り込み処理ルーチンはいくつかの僅かな変更が必要です。割り込み処理ルーチンは起動復帰割り込み、または時間経過警告割り込みのどちらを扱うべきかを定めるためのフラグを使用しなければなりません。割り込み処理ルーチンの例は図2-5.で示されます。



起動復帰で正しい処理部が実行されるのを保証するために、起動復帰フラグが休止形態への移行に先って手動で設定されなければならないことに注意してください。WDT割り込み動作は起動復帰割り込みを扱った後で、割り込み処理ルーチンの外側で再許可されなければなりません。

流れ図の右側の分岐は「WDTシステムリセットに先行する予備パラメータ保存」項で記述されます。

### 2.4. 始動での考慮

強化型ウォッチドッグタイマを持つデバイスに対する設計時、始動コード内でのWDTリセットフラグ(WDRF)評価が重要です。これは例え応用が決してWDTの使用を意図していなくても適用されます。WDTシステムリセット動作が意図せずに許可されてシステムリセットを引き起こす場合、WDTリセットフラグが設定(1)され、WDTシステムリセット動作はシステムリセット後も許可を維持します。従って、始動コードはWDTリセットフラグを調べ、一度も使用されないのにそれが許可されている場合、WDTシステムリセット動作を禁止すべきです。これらの考慮はWDTONヒューズが非プログラム(1)にされている時にだけ適用します。WDTONヒューズがプログラム(0)されていると、WDTシステムリセット動作が常に許可されます。ヒューズ設定の変更方法はデバイスのデータシートで記述されます。

WDTが応用で意図的に使用されてシステムリセットが起こる場合、始動コードはWDTリセットフラグを取り扱う仕組みを持つべきです。最も安易な解決方法はこのフラグを全く無視して通常のように継続することです。この対策は時折現れるバグからシステムを救いますが、繰り返される、または持続的な異常を扱う方法を持ちません。

可能な拡張は不揮発性メモリ内にWDTシステムリセット計数器を保持することです。そして始動コードはこの計数器が予め定義された限度を越えた場合に安全にシステムを停止して操作者へ通知すべきです。或る種のシステムクロック計時(不揮発性メモリ内の予備保存)を使用し、始動コードは固定期間に渡って繰り返されるリセットの検知を試みることもできます。

パラメータの予備保存が使用される場合、「WDTシステムリセットに先行する予備パラメータ保存」項で記述された書き込み完了フラグを調べてシステムを安全な状態に回復する試みをすべきで、または少なくとも操作者へいくつかのデバッグ情報を提供できます。

## 2.5. WDT形態設定変更

WDT設定への予期せぬ変更を防止するため、WDTシステムリセット動作禁止または時間経過周期変更に対して特別な時間制限手順が必要とされます。

WDTシステムリセット動作を禁止するには、禁止に先立つ4CPUクロック周期内にウォッチドッグ変更許可(WDCE)ビットが設定(1)されなければなりません。そうしなければWDTシステムリセット動作は許可に留まるでしょう。WDTONヒューズがプログラム(0)されている場合、WDTシステムリセット動作が常に許可されます。

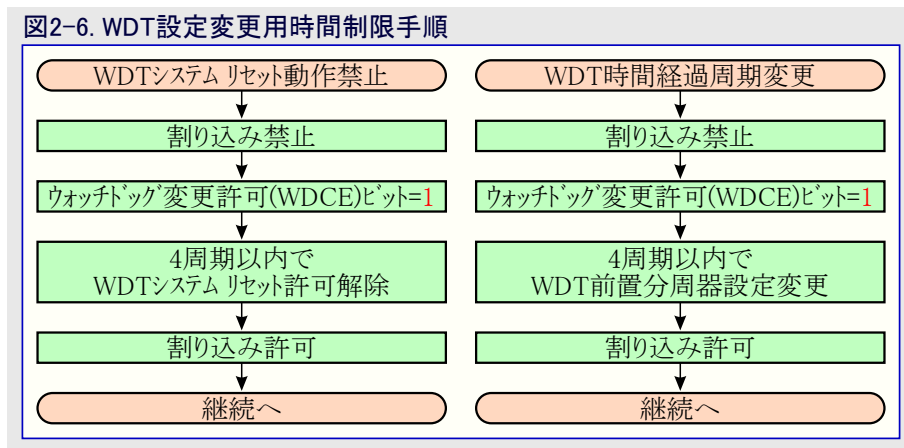
時間経過周期を変更するには、時間経過値変更先立つ4CPUクロック周期内にウォッチドッグ変更許可(WDCE)ビットが設定(1)されなければなりません。けれども通常動作中の時間経過周期変更は推奨されません。これは初期化コード内で一度行われるべきです。

ATtiny13とATtiny2313でWDTONヒューズが非プログラム(1)にされると、以降の時間制限手順なしでWDT時間経過周期の変更が可能です。

WDT割り込み動作設定の変更とWDTシステムリセット動作の許可は特別な考慮の必要がありません。

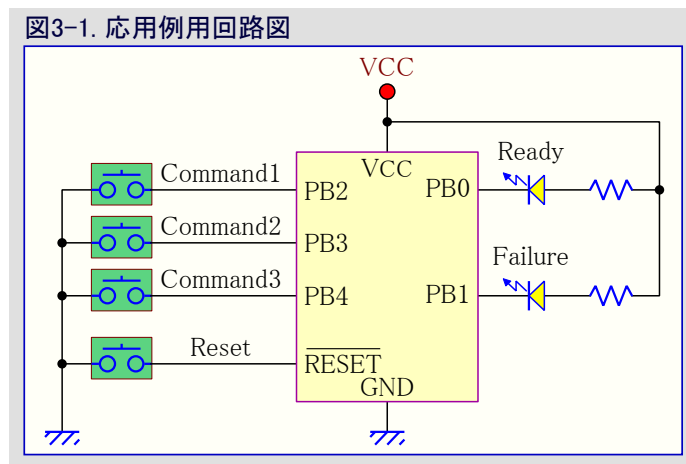
割り込みは設定変更時に禁止されるべきです。これは4周期制限経過を引き起こす割り込みが起きないことを保証します。

WDT設定変更に関する流れ図は図2-6.で示されます。



## 3. 実装

本応用記述はC言語で書かれた3つのコード例を提供します。これらは全てSTK<sup>®</sup>500開発基板または同等品上に配置されたATtiny13デバイス用に設計されています。ポートPB0とPB1は各々準備完了(Ready)LEDと失敗(Failure)LEDに接続され、PB2,PB3,PB4はSTK500の3つのスイッチに接続されます。Low出力駆動がLEDをON(点灯)し、釦押下が対応する入力をLow駆動することに注意してください。この設定は図3-1.で示されます。



本例は以下の概念を実演します。

- ・システムリセット元としてWDTを使用 (3.1項)
- ・起動復帰計時器としてWDTを使用 (3.2項)
- ・起動復帰計時器と予備パラメータ保存付きシステムリセット元の組み合わせとしてWDTを使用 (3.3項)

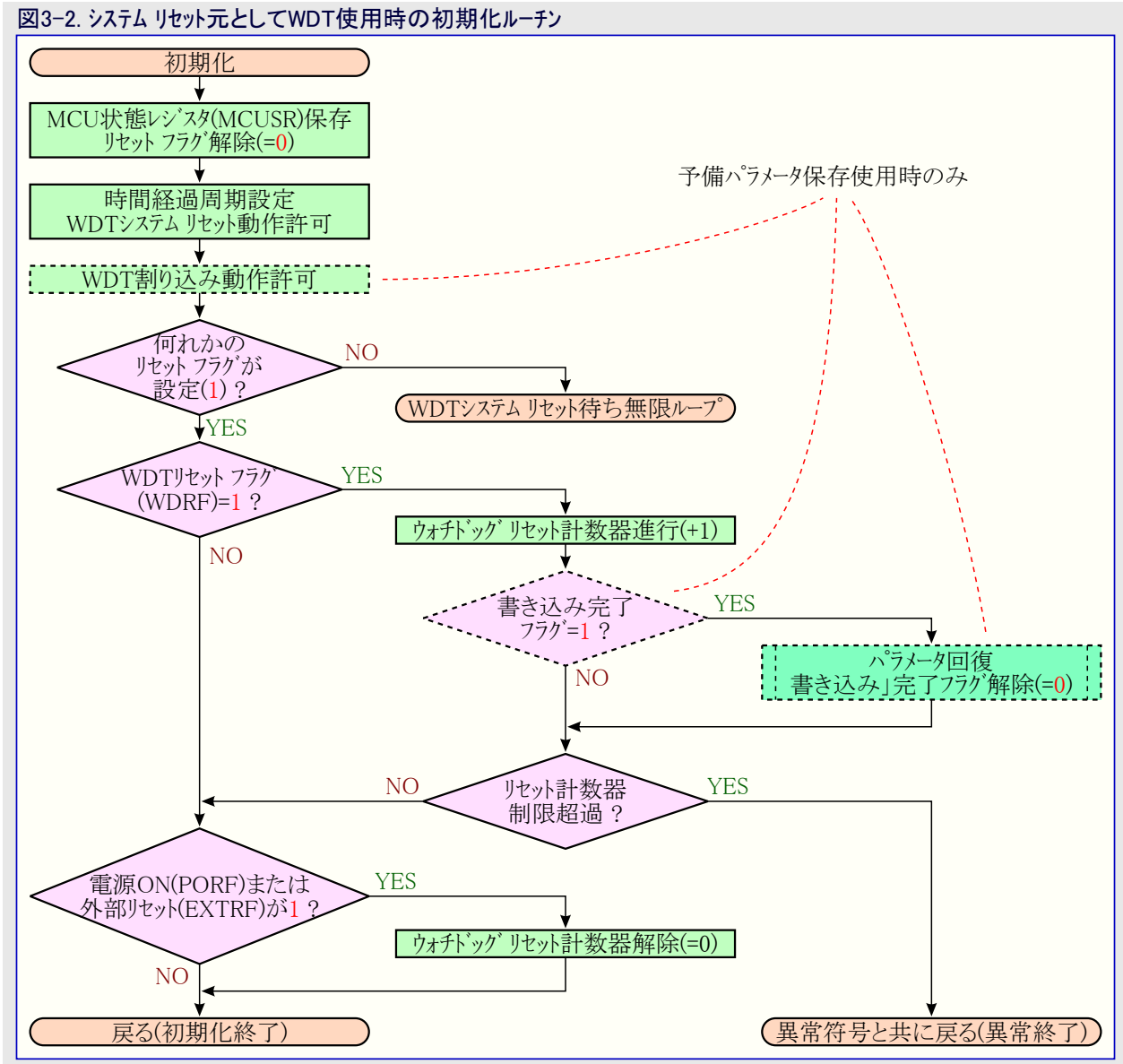
**注:** WDT割り込み動作を使用する例を走行するとき、WDTONヒューズは非プログラム(1)にされなければなりません。

### 3.1. システム リセット元としてのWDT使用法

本例は3つのルーチンと最後での健全検査を含む主繰り返しと初期化ルーチンの、[図2-2.「WDTシステム リセット動作時の主繰り返し」](#)で記述された構造を実装します。各ルーチンは全てOKを示す自身の健全フラグを持ちます。3つのルーチンは各々命令を得て、それを解析して実行します。

#### 3.1.1. 初期化

初期化ルーチンには、周辺機能の初期化、及びリセットフラグの操作と、2つの主な目的があります。この流れ図は[図3-2.](#)で示されます。破線枠項目は組み合わせ動作例でだけ使用され、後で記述されます。



最初の条件分岐は始動でどのリセットフラグも設定(1)されていない場合を扱います。リセットフラグは初期化ルーチンで常に解除(全0)されるので、これは暴走コードがアドレス\$0000へ戻るのを含み、リセットなしで再び一度始動コードを走行する時に起きます。これは明らかにソフトウェアのバグまたは誤りで、WDTシステムリセットのように扱われます。初期化ルーチンは直ぐに無限繰り返しに移行し、デバイスの正しいリセットのためにWDT(システムリセット)を待ちます。

それからコードはWDTリセットフラグ(WDRF)を調べます。設定(1)されていれば、このルーチンはWDTリセット計数器を進行(+1)して予め定義された制限に対して検査します。この制限を越えていれば、応用は定常的な異常繰り返しと仮定し、失敗表示LEDによってこれを示して実行を停止します。内部にWDR命令を含む無限繰り返しへの移行により、実行は外部リセットが起こるまで事実上停止されます。

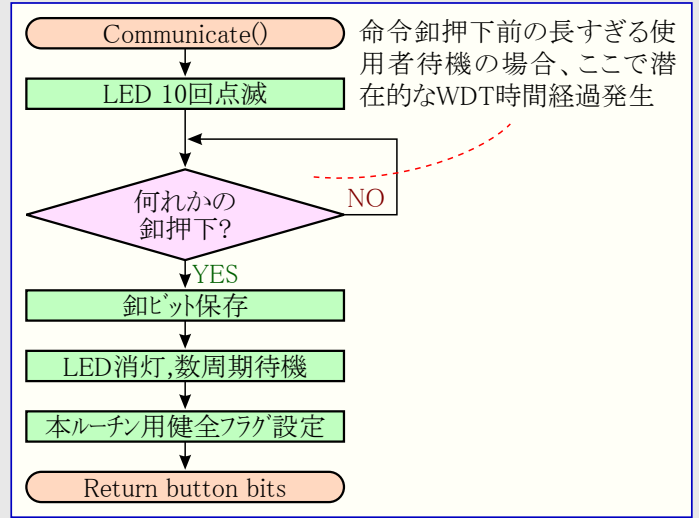
電源ONリセットと外部リセット事象は手動介入を考慮し、WDTリセット計数器が解除(=0)されます。これは多すぎるWDTシステムリセットによって停止されてしまった応用を手動リセットすることを人間の操作者に対して可能にします。勿論、操作者はリセットする前にWDTシステムリセットの原因を探すことを試みなければなりません。盲目的なリセットとそれら自体の修正を飛ばすのは推奨されない解決方法です。流れ図の残りは自己解釈すべきです。

### 3.1.2. 命令伝達

命令を得るルーチンは貧弱に設計された伝達ルーチンの例です。それは10回LEDを点滅して釦押下を待ちます。使用者が長すぎる待機を行う時に問題が起きます。強固な設計は或る種の時間超過検査を実装し、伝達時間超過の場合に異常符号と共に戻るでしょう。けれども、このルーチンはそれを行わず、WDT時間経過周期内に釦が押されない場合、WDTがデバイスをリセットします。伝達ルーチンの流れ図は図3-3.で示されます。

時間内に命令釦が押下される場合、このルーチンは健全フラグを設定して釦押下ビット様式を戻します。

図3-3. 或る釦を押下する使用者用のLED点滅と待機



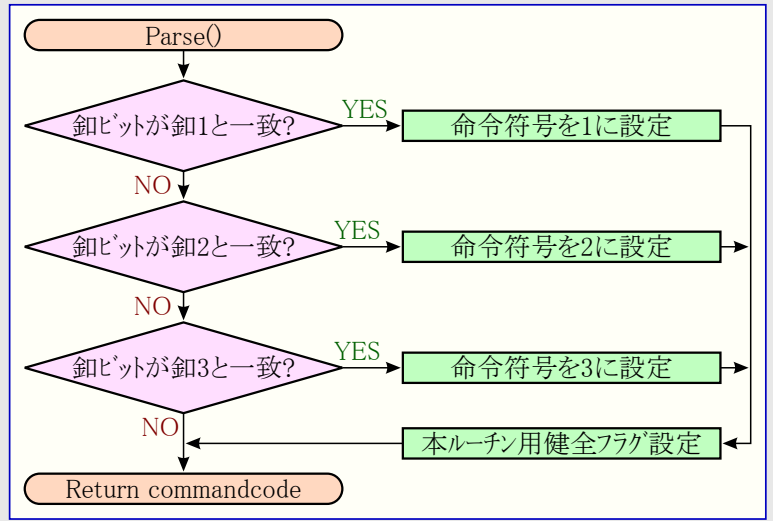
### 3.1.3. 命令解析

この命令解析部は釦押下ビット様式を命令符号に変換するためにC言語でswitchキーワードを使用します。この(ビット)様式は命令釦の各々についてビット遮蔽値に対して比較されます。一致が見つかり、対応する命令符号が設定され、このルーチン用の健全フラグが設定されます。

解析部の流れ図は図3-4.で示されます。

2つ以上の釦が同時に押された場合、解析部は決して一致を見つけず、これの健全フラグも決して設定されないことに注意してください。これで主繰り返しがWDTをリセットしないため、主繰り返しが成功裏に実行されず、十分に速い秒数ならばシステムリセットが起きるでしょう。これは(プログラム例で)、switch部でのdefault caseによって扱われない場合に予期せぬ入力が問題を起すことを示す例です。

図3-4. 釦押下様式を命令符号へ変換



### 3.1.4. 命令実行

このルーチンで、命令符号はどの動作を実行するかを決めます。

命令1は特別な動作を持ちませんが、有効命令となることによって主繰り返しの健全走行を維持します。他の命令は実動作応用で起こり得る色々なバグを実演します。

命令2はEEPROM操作可割り込みを許可します。この割り込みはEEPROM部が準備完了である限り継続的に実行され、初期化ルーチン後にEEPROMが決して使用されないため、この場合は常に実行されます。EEPROM操作可割り込みの繰り返し実行は主繰り返しをかなり遅くし、遂にWDTシステムリセットが起るでしょう。従って、命令2は多すぎるまたは手抜き設定の割り込みが主繰り返しを著しく遅くするかもしれないことを示す例です。

命令3は暴走コードの例を与えます。この例は当に未使用アドレスで関数を呼びます。プログラムカウンタ(PC)はプログラムメモリの最後へ走行し、丸められてアドレス\$0000に戻ります。リセットフラグは設定(1)されず、誤りが初期化ルーチン内で安全に受け取られます。

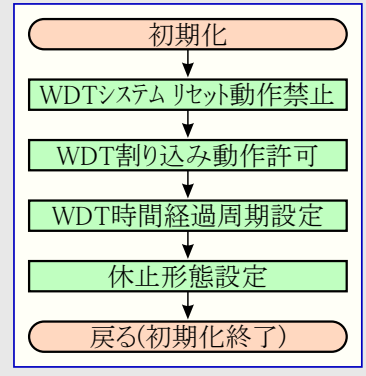
不正関数呼び出しのシミュレートのため、次のコード部分が使用されます。"`((void(*)())0x1FF)();`" integerの0x1FFは関数へのポインタに変換されて関数が呼ばれます。関数ポインタと型変換のより多くの詳細についてはANSI C規格を参照してください。

### 3.2. 起動復帰元としてのWDT使用法

本例はWDT割り込み動作だけを使用し、従って初期化ルーチンはかなり減少されています。先に記述されたように、例えばWDTシステムリセット動作が決して使用されなくても、始動でWDTシステムリセット動作を禁止することが重要です。この初期化ルーチンは図3-5.で示されます。

本例の主繰り返しは(CPUが)起きたことを示すためにPB0に接続されたLEDを10回点滅します。そしてWDTをリセットし、WDT割り込み動作を許可して休止形態へ移行します。WDT時間経過時にCPUを再び起動します。休止形態へ移行する前に主繰り返しが長く走行する場合に不必要な割り込みが幾度も生成されないように割り込み処理ルーチンはWDT割り込み動作を禁止します。

図3-5. 起動復帰計時器としてWDT使用時の初期化ルーチン



### 3.3. 組み合わせ動作

3つ目の例は予備パラメータ保存付きのシステムリセット元と起動復帰計時器の両方でのWDT使用法を示します。これは最初のコード例の拡張版で、休止形態移行に命令1(鉦)を使用します。

本例で、初期化ルーチンは流れ図内の破線枠で示された部分を含みます。これはWDT割り込み動作が許可され、始動で書き込み完了フラグが設定されている場合に予備保存されたパラメータが回復されることを意味します。

予備保存されるパラメータはタイマ/カウンタ1の値です。これは本応用で特別な機能(意味)を持ちませんが、リセットで解除され、回復されるべき必要があるパラメータとして扱います。

WDT割り込み処理ルーチンは図2-5.「二重目的WDT割り込み処理ルーチン」記載のように実装されます。起動復帰フラグとして休止許可(SE)ビットが使用されます。命令1が実行されると、応用はWDTをリセットし、休止許可(SE)ビットを設定(1)して休止形態へ移行します。割り込み処理ルーチンはWDT時間経過がCPUを起動復帰する時に実行され、起動復帰フラグは行う動作が何かを決めます。それが既に解除(0)なら、異常が起り、失敗LEDが点灯されます。割り込み処理ルーチン実装の残りは流れ図とで補完してください。

コードの残りは最初の例の記述と同じです。

## 4. 参考文献

- Michael Barr - Introduction to Watchdog Timers ..... <http://www.embedded.com/story/OEG20010920S0064>
- Niall Murphy - Watchdog Timers ..... <http://www.embedded.com/2000/0011/0011feat4.htm>
- Jack Ganssle - Born to Fail ..... [http://www.embedded.com/design\\_library/OEG20021211S0032](http://www.embedded.com/design_library/OEG20021211S0032)
- Kernighan & Ritchie - "The C Programming Language", 2nd edition.





## 本社

### *Atmel Corporation*

2325 Orchard Parkway  
San Jose, CA 95131  
USA  
TEL 1(408) 441-0311  
FAX 1(408) 487-2600

## 国外営業拠点

### *Atmel Asia*

Unit 1-5 & 16, 19/F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
Hong Kong  
TEL (852) 2245-6100  
FAX (852) 2722-1369

### *Atmel Europe*

Le Krebs  
8, Rue Jean-Pierre Timbaud  
BP 309  
78054 Saint-Quentin-en-  
Yvelines Cedex  
France  
TEL (33) 1-30-60-70-00  
FAX (33) 1-30-60-71-11

### *Atmel Japan*

104-0033 東京都中央区  
新川1-24-8  
東熱新川ビル 9F  
アトメル ジャパン株式会社  
TEL (81) 03-3523-3551  
FAX (81) 03-3523-7581

## 製品窓口

### ウェブサイト

[www.atmel.com](http://www.atmel.com)

### 技術支援

[avr@atmel.com](mailto:avr@atmel.com)

### 販売窓口

[www.atmel.com/contacts](http://www.atmel.com/contacts)

### 文献請求

[www.atmel.com/literature](http://www.atmel.com/literature)

お断り: 本資料内の情報はATMEL製品と関連して提供されています。本資料またはATMEL製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。ATMELのウェブサイトに位置する販売の条件とATMELの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、ATMELはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえATMELがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益の損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してATMELに責任がないでしょう。ATMELは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。ATMELはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、ATMEL製品は車載応用に対して適当ではなく、使用されるべきではありません。ATMEL製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© Atmel Corporation 2008. 全権利予約済 ATMEL®、ロゴとそれらの組み合わせ、AVR®、STK®とその他はATMEL Corporationの登録商標または商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

### © HERO 2013.

本応用記述はATMELのAVR132応用記述(doc2551.pdf Rev.2551C-06/08)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。