WDT (Watchdog Timer)

KN0016-00／2024 年 12 月 07 日／こがねさん(著)
https://www.kumikomist.com/

■目次

■文書内の記号

	取り扱いにおける禁止事項（してはいけないこと）を示しています。
	取扱における指示事項（必ずしなければいけないこと）を示しています。

	取り扱いにおける注記事項を示しています。
	取り扱いにおけるポイントを示しています。

1. WDT とは

ウォッチドッグタイマー（WDT：WatchDog Timer）は暴走やフリーズ（ハングアップ）したシステムをリセットして、正常動作に戻すことを目的としたタイマー機能です。ウォッチドッグは番犬という意味で、システムがフリーズしていないかを常に監視しています。殆どのマイコンに組み込まれている基本的な機能の 1 つです。

2. WDT の仕組み

WDT の仕組みは単純です。起動と同時に内部のカウント値を繰り上げていき、カウント値がオーバーフロー（リミット超え）するとマイコンをリセットします。そうならないためにマイコンは、WDT のカウント値をクリアーし続けなければなりません。

実際の動作は次の通りです。マイコンのプログラムの中に、WDT のカウント値を定期的にクリアーする処理を組み込みます。プログラムが正常に動いている限り、カウント値は定期的にクリアーされて WDT が働くことはありません。しかしひとたびプログラムが暴走してカウント値がクリアーされなくなると、カウント値がオーバーフローしてプログラムは再起動されます（図 1 参照）。

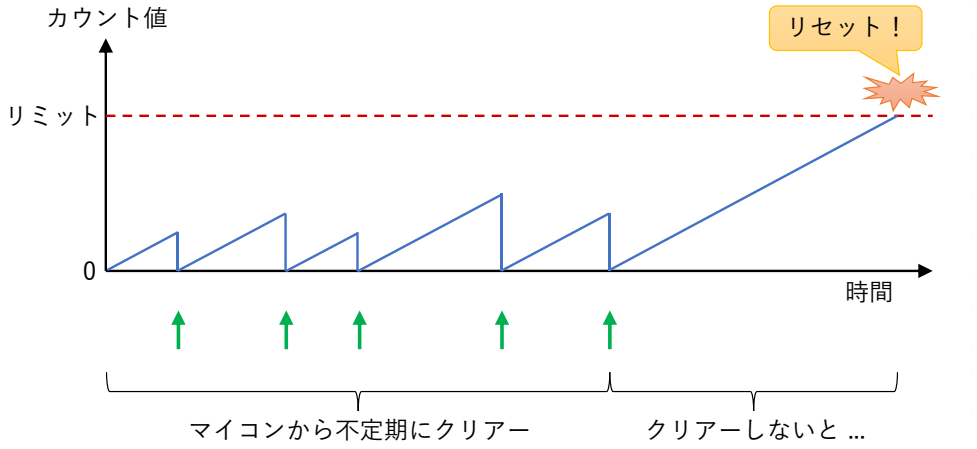

図 1　WDT の仕組み

⚠	● 暴走した中でも、カウント値のクリアー処理が走る可能性があります。WDT を過信しないでください。

3. WDT を使う際の注意点

最大処理時間を把握する

プログラムの中に定期的にカウント値をクリアーする処理を入れていたとしても、それが常に一定以下のタイミングで働き続けなければ意味がありません。通常動作中はまず問題ないでしょう。テストも万全です。しかしメニューや調整モードなどすべての処理が一定の時間内で完了する必要があります。特に気を付けなければならないのは異常系です。異常検知によるリトライ処理や異常系からの復帰処理、これらの最中にカウント値がオーバーフローすることがないよう、よく確認してください。

割り込み処理の中でカウント値をクリアーしない

割り込み処理の中でカウント値をクリアーすべきではありません。特にタイマー割り込みの中は絶対に避けるべきです。タイマー割り込みの中にクリアー処理を入れれば、ほぼ確実に一定時間毎にカウント値をクリアーできるでしょう。しかしプログラムが暴走した状態でもクリアー処理が走ってしまう恐れがあります。それでは本末転倒です。

WDT にまつわる小話

起動時は WDT が有効であるマイコンがあります。WDT を無効にしたければ、起動直後に自ら無効に設定しなければなりません。そして C/C++では起動時にグローバル変数を初期化します。
この 2 つが組み合わさった結果、予想だにしないことが起こりました。main 関数が走らないのです。この原因が、大量に宣言していたグローバル変数の配列でした。試しに配列をコメントアウトすると、main 関数が走り出したのです。この挙動を見て、グローバル変数の初期化中に WDT が働いたのだろうと結論付けました。対策として、大規模な配列を初期化させない設定を行い事なきを得たのです。

4. 内部 WDT と外部 WDT

先に述べたとおり、WDT は殆どのマイコンに内蔵されています。しかしそれでもなお、多種多様な WDT が販売されています。それはひとえに、内蔵式 WDT の信頼性の低さにあります。プログラムのフリーズがマイコンの誤作動によるものの場合、WDT そのものも停止してしまう恐れがあります。このため長時間にしろ短時間にしろフリーズが許されないシステムでは、外付けの WDT が多用されます。とはいえ内蔵式にもメリットはあります。表 1 を参考に内部 WDT と外部 WDT のどちらを選択するかを検討してみてください。

表 1　内部 WDT と外部 WDT のメリット／デメリット

	メリット	デメリット
内部 WDT	● 追加コストはゼロ。 ● 使いやすい。レジスターに特定の値を設定するだけでクリアーできることが殆ど。 ● 途中でリミット時間を変更できる。	● マイコンの誤作動により、WDT が正常に動作しない恐れがある。
外部 WDT	● マイコンの誤作動により、WDT が無効化されることはない。 ● 信頼性の高さこそが最大のメリット。	● 追加コストが掛かる。 ● マイコンの出力ポートを 1 本専有する。 ● 温度の影響を受けて、オーバーフローまでの時間が変わりやすい。 ● マイコンがスリープなど省電力モードに入った状態でも、カウント値をクリアーし続ける必要がある。

● マイコン内蔵の WDT の中にも、マイコンのシステムクロックとは独立したクロックで動作するものもあります。STM マイコンの IWDG（Independent WatchDoG）がそれです。そういったものであれば、内蔵式であっても信頼性は高いでしょう。

5. WDT のタイプ

5.1. ノーマルタイプ

「2. WDT の仕組み」で紹介したとおり、カウント値がオーバーフローするとリセットする一般的なタイプです。

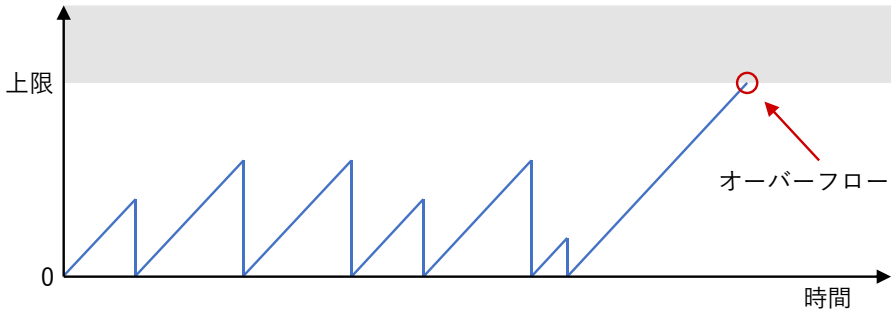

図 2　ノーマルタイプの挙動

5.2. ウィンドウタイプ

ノーマルタイプの一定時間経過するとリセットが掛かる機能に加えて、カウンター値のクリアー頻度が速すぎてもアンダーフローでリセットするのがウィンドウタイプです。このアンダーフローまでの、正規のクリアー信号を受け付けない期間をクローズドウィンドウ、正規のクリアー信号を受け付ける期間をオープンウィンドウと呼ぶこともあります。

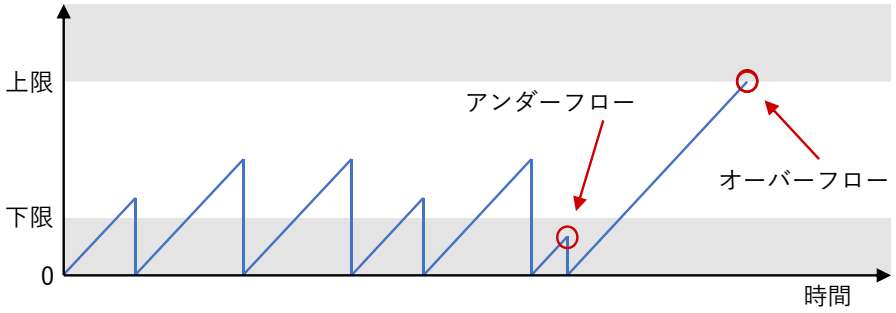

図 3　ウィンドウタイプの挙動