

序説

IEC60730は家庭用機器に対する機能的な安全規格です。マイクロ コントローラ(MCU)は家庭用機器に於いて広範囲に使用される電気制御部品です。MCUシステムの機能的な安全の必要条件は家庭用機器の組み込みソフトウェアとハードウェアの正しく安全な動作を提供するための詳細な試験項目、診断、測定とで、規格の付属書Hの”電気制御用必要条件”で提示されます。この応用記述はIEC60730規格の簡素な序説を与え、各MCU部品に対する機能的な検査の手引きを提供します。この応用記述はAtmel[®] megaAVR[®]とtinyAVR[®]での各々のIEC60730等級Bファームウェア例も提供します。

特徴

- IEC60730等級B適合機能試験の指針
- IEC60730等級B適合周期的試験の指針
- ATtiny817とATmega328PB用の等級Bファームウェア例

目次

序説	1
特徴	1
1. 概要	3
2. AVR用単一チャネル機能試験の手引き	3
2.1. CPUレジスタ試験(汎用レジスタR0~R25,X,Y,Zレジスタ検査)	3
2.2. スタックポインタレジスタ試験	3
2.3. ステータスレジスタ試験	3
2.4. RAMメモリ試験	4
2.5. フラッシュメモリ検査	4
2.5.1. SPM命令の使用での注意	4
2.6. EEPROMメモリ試験/検査	4
2.7. ウォッチドッグ試験	4
2.8. 割り込み機能試験	4
2.9. I/Oレジスタ	4
2.10. クロック周波数	4
2.11. A/D変換器	5
3. AVRでの単一チャネル周期的自己検査の手引き	5
3.1. 割り込み周期的事象	5
3.2. 関数周期的事象	5
3.3. 異常事象検出	5
4. 付加的指針	5
4.1. 外部通信	5
4.2. ウォッチドッグ	5
5. IEC60730等級B7ファームウェア例	5
5.1. 試験ルーチン	6
5.1.1. <code>_low_level_init()</code>	6
5.1.2. <code>watchdog_test()</code>	6
5.1.3. <code>sram_test()</code>	7
5.1.4. <code>cpu_status_test()</code>	7
5.1.5. <code>timer_tca_test()</code>	7
5.1.6. <code>timer_tcb_test()</code>	7
5.1.7. <code>interrupt_test()</code>	7
5.2. メモリ使用量と走行周期数	8
6. 応用記述参考	8
7. 追補	8
7.1. 安全な設計方法	8
7.2. 行進B試験説明	9
8. 改訂履歴	9

1. 概要

国際電気標準会議(IEC:International Electrotechnical Commission)は家庭用機器開発を対象にするIEC60730を導入しました。IEC 60730は製品の設計と動作の両面の多くを扱う家庭機器用安全規格です。この規格は安全が重要な装置に対する他の規格、例えばIEC60335によっても参照されます。この規格でのシステム全体の遵守は動作が安全と認定されるべき機器に対して必要です。この応用記述は電子制御装置を考える規格の付属書Hに応じるための手引きです。IEC60730の付属書Hは制御ソフトウェアに対して以下の等級別を記述します。

- 等級A – 装置の安全性に対して信頼されるべきことを意図されない制御機能 (H.2.21.1)
- 等級B – 機器でソフトウェア誤り以外の誤りが起きた場合の危険を防ぐことを意図されたコードを含むソフトウェア (H.2.21.2)
- 等級C – 他の保護的な装置を使うことなく危険を防ぐことを意図されたコードを含むソフトウェア (H.2.21.3)

この応用記述は食器洗浄器、洗濯機、冷蔵庫、冷凍庫、調理器を含む大きな範囲の製品に適用する等級Bを扱います。IEC60730に従い、家庭用機器製造者は今やクラスB規則に従ってこれらの製品を設計しなければなりません。

上で挙げた製品の殆どは組み込まれたメモリと周辺機能を持つ単一チップのマイクロコントローラを使用し、単一チャネル応用として参照されます。

IEC60730に従い、(1つのマイクロコントローラだけを使用する)単一チャネル応用は下の2つの技術の1つに基づいて設計されたファームウェアを持たなければなりません。

- 単一チャネル機能試験
- 単一チャネル周期的自己検査

機能試験を使用する単一チャネルは今日使用される最も一般的な手法でそして実装が最も容易です。殆どの電気製品製造業者は今やそれらの新しい設計に周期的な自己検査を持つ単一チャネルを実装します。この資料はこれら2つの技術を扱い、Atmel AVR[®]マイクロコントローラ使用時にクラスBの必要条件に適合する方法の付加的な指針を与えます。

AVRマイクロコントローラの機能の全ては工場製造中に検査されます。いくつかの特徴は他より過酷な環境にもっと敏感です。例えば、CPUレジスタ、スタックポインタレジスタ、ステータスレジスタは動かないビットに要注意です。一方、RAM、フラッシュメモリ、EEPROMのようなデバイス上の大きな領域はもっと複雑な検証算法が必要かもしれません。この資料では機能の感受性と実装されるべき検査間の妥協案を与えます。お客様はそれらの必要条件に従って追加の検査を簡単にする、または追加することを決定することができます。

2. AVR用単一チャネル機能試験の手引き

試験が実行される順番は重要で、試験は以下で与えられるのと同じ順番で動かなければなりません。例えば、フラッシュ内容検査は走行するのにRAMを使用するため、フラッシュ内容の前にRAMを検査することが重要です。いくつかの検査はファームウェア侵入的でもあり(例えば割り込み検査)、最終応用ファームウェアに統合され得ます。機能試験での全ての異常は応用を停止して異常メッセージを生成するかもしれません。

周期的検査での異常は応用を再始動するためのウォッチドッグ時間超過に至るかもしれません。

検査実装は始動時間、使用される資源、そして応用の保護レベル間での妥協です。

誤り情報は誤り診断を助けるために応用によって保存することもできます。

2.1. CPUレジスタ試験(汎用レジスタR0~R25,X,Y,Zレジスタ検査)

この検査の目的: これらのレジスタのビットが動かないことが無いことを検査

CPUレジスタ検査はR0~R31の汎用作業レジスタとステータスレジスタの全てを検査します。この検査の目的はこれらのレジスタの或るビットが'1'または'0'で動かない場合を検出することです。この検査はそれらのレジスタ内への\$55と\$AAの値の連続的な書き込み、読み込み、調査によって行われます。この検査はスタックポインタ初期化直後にアセンブリ言語で行われ、最初にR29~R31が検査され、その後それらのレジスタがR0~R28の検査に使用されます。

2.2. スタックポインタレジスタ試験

この検査の目的: スタックポインタのビットが動かないことが無いことを検査

この検査はこれらのレジスタ内に\$55と\$AAの値を連続的に書き、読み、そして調査することによって行われます。この検査の前に**RET**と**RETI**の命令は全く実行されません。

2.3. ステータスレジスタ試験

この検査の目的: このレジスタのビットが動かないことが無いことを検査

この検査の目的はこのレジスタの或るビットが'1'または'0'で動かない場合を検出することです。この検査はこのレジスタ内への\$55と\$AAの値の連続的な書き込み、読み込み、調査によって行われます。

ステータスレジスタは前の検査によって使用され、故にこの検査に達するのに以下の3つの道筋を考察し得ます。

1. ステータスレジスタが完全に機能で、従って前の全ての検査は良しです。
2. ステータスレジスタが不正で、従って前の検査は検査それ自体が良好でなかったために失敗になり、この検査に達し得ません。
3. ステータスレジスタが不正で且つ前の検査を通過したけれども、ここで停止するでしょう。

全ての場合で、前の検査の1つまたはステータスレジスタ検査での何かの問題はプログラムの停止に至るでしょう。

2.4. RAMメモリ試験

この検査の目的：RAMメモリのビットが'1'または'0'で動かないことが無いことを検査

RAM検査は全てのRAMメモリ位置を検査します。この検査はRAMメモリ内への\$55と\$AAの値を連続的に読み、書き、調査することによって行われます。ファームウェアは"RAM検査状態レジスタ"内のビットを設定します。プログラムは検査の最後でこのビットの設定を確認します。

アドレス符号化部の検査を許す別の検査はRAM位置内に位置のアドレスの補数を書くことです。格納された値を検査するための読みと照合の手法もあります。検査後に取り戻されるべき積まれたデータを許すために、RAMの初期値は検査前に(周期的検査を行う時に)保存されてその後で復元されます。

全てのRAMを検査するために**行進B検査**が実装されます。この検査は独立して検査される2つの部分に分けられます。各部分検査の間、スタック内容は別の部分に保存され、スタックポインタは別の部分を指し示します。大きさと両部分の重複は形態設定可能で、物理メモリ構成に従って設定されなければなりません。

2.5. フラッシュメモリ検査

この検査の目的：フラッシュ内容を検査

フラッシュメモリ検査はどんなフラッシュの不正も防がなければなりません。それは簡単なフラッシュ内容のチェックサム、またはもっと複雑で時間を消費する巡回冗長検査(CRC)によって行うことができます。参照基準の結果はプログラミング時にフラッシュメモリの特別な場所に格納されます。計算された値は走行時にこの参照基準の結果と比較されます。

2.5.1. SPM命令の使用での注意

それが可能な時に、過酷な環境でSPM命令を使用しないことが推奨されます。

SPM(Store Program Memory)はフラッシュメモリ内への書き込みを許す命令です。SPM命令はブートローダ領域を含むフラッシュ全体をアクセスすることができます。例えば、関数がSPM命令を使用し、そして電力損失が起こる場合、フラッシュメモリは不正にされ得ます。

SPM命令の使用に対照してブートローダ領域に対する保護レベルはAVR製品の殆どで利用可能なブートローダ施錠ビットによって選択することができます。

2.6. EEPROMメモリ試験/検査

この検査の目的：EEPROM内容を検査

EEPROMメモリ検査はどんなEEPROMの不正も防がなければなりません。それは簡単なEEPROM内容のチェックサム、またはもっと複雑で時間を消費する巡回冗長検査(CRC)によって行うことができます。EEPROMの内容が応用の寿命中に変化し得るため、最適な解決策は製品寿命中にメモリ内容の動的比較を持つことができるように、EEPROMの各書き込みで参照基準の結果を更新すべきです。

2.7. ウォッチドッグ試験

この検査の目的：ウォッチドッグが機能していることを確認

この検査はウォッチドッグリセットの機能を調べます。リセットに於いて、この検査はリセットがウォッチドッグリセットから起きたかを調べます。そうでないなら、ウォッチドッグが開始され、検査はそれが起こるまで待ちます。

注: ウォッチドッグはヒューズが許可されなければなりません。

2.8. 割り込み機能試験

この検査の目的：割り込み制御器が正しく動くか検査

割り込み制御器の正しい動きを調べるために、応用によって使用されない全ての割り込みは割り込み検査関数によって活性化にされ得ます。割り込みはソフトウェアによって活性化にされ、対応する割り込みベクタが割り込み検査関数への合図を生成します。

2.9. I/Oレジスタ

この検査の目的：入出力ビットが'1'または'0'で動かないことが無いことを検査

I/O検査は全ての入出力を検査します。この検査はI/Oレジスタ内に\$55と\$AAの値を連続的に書き、読み、調査することによって行われます。この検査は応用依存で、ハードウェアがそれを許す場合にだけ走行することができます。

2.10. クロック周波数

この検査の目的：内部クロック周波数の検査

プロセッサの内部クロックを検査するには、利用可能な参照基準クロックを持つことが必要です。外部クリスタルを使用する応用については、クリスタルの存在を調べると、外部クリスタルの周波数発振を確認するのにも、内部RC発振器を使用することができます。

別の方法はSPIクロック持続時間を測定するためにSPIのような通信バスを使用することです。その結果は理論値と比較されます。

ファームウェアは"クロック検査状態レジスタ"内のビットを設定します。プログラムは検査の最後でこのビットが設定されていることを確認します。

2.11. A/D変換器

この検査の目的：A/D変換器(ADC)のアナログ機能の検査

ADCの正しい動きを管理するために空いているアナログ入力には既知の外部電圧に配線することができます。

この検査はADC入力として内部バンドギャップ基準電圧を使用して行うこともできます。例えば、ATmega16の内部バンドギャップ基準電圧はADCを検査するために一定不変で変換することができる1.22Vの電圧を配給します。

3. AVRでの単一チャネル周期的自己検査の手引き

周期的試験はファームウェアに組み込まれ、全ての物が通常のように機能していることを定期的に調べます。応用ファームウェアの実行に先立って機能検査が実行されます。周期的検査での全ての異常は応用を再開するためのウォッチドッグ時間超過に至るかもしれません。

3.1. 割り込み周期的事象

この検査の目的：定義した時の経過で割り込みが起こることを調査。同じように、計数器の使用により、この検査は割り込みが頻繁に起きすぎる場合を検出することができます。

各割り込みベクタアドレスに於いて、ファームウェアは使用者定義された“IT検査状態レジスタ”内のビットを設定します。プログラムはこのビットが設定されているのを周期的に確認し、その後に“IT検査状態レジスタ”を解除します。異常でウォッチドッグ事象が呼ばれます。

応用で使用されない割り込みは、割り込み制御器を定期的に調べるために、この検査によって使用することができます。

3.2. 関数周期的事象

この検査の目的：定義された時間の経過でいくつかの関数が定期的に呼ばれることを調査

割り込み検査と同じ機構を使用して、各関数は“関数事象状態レジスタ”内のビットを設定します。プログラムはそのビットが設定されていることを周期的に確認し、その後に“関数事象状態レジスタ”を解除します。異常でウォッチドッグ事象が呼ばれます。

3.3. 異常事象検出

この検査の目的：全ての関数が定常に基づく正しい値を与えることを調査

この検査は前のものと僅かに異なります。この検査は関数が有効な値を与えることを調べます。値が範囲外の場合、またはどれかの機能が故障を経験して正しい値(例えばADC変換結果)を返せない場合に、これは再始動を許します。周期的な関数は特定関数からの不正な値または多数の不正な値でウォッチドッグ事象を生成します。

4. 付加的指針

4.1. 外部通信

全ての通信は転送中の不正データを調べることを受信部に許すため、可能な時に余剰物を転送することによって確実にされなければなりません。失敗と無限繰り返しを防ぐために時間超過検出が実装できます。

4.2. ウォッチドッグ

強いウォッチドッグの使用はファームウェアからのコード誤りと異常値を防ぎます。

5. IEC60730等級Bファームウェア例

この応用記述は主応用に等級B試験を追加する方法を示すためにATmega328PBとATtiny817用のIEC60730等級Bファームウェア例を提供します。使用者はそれらの特定応用に基づいて安全性試験を更に高めることができます。この例はAtmel START(開始)からダウンロードすることができ、それらは他のmegaAVRやtinyAVRマイクロコントローラへ容易に移植することができます。

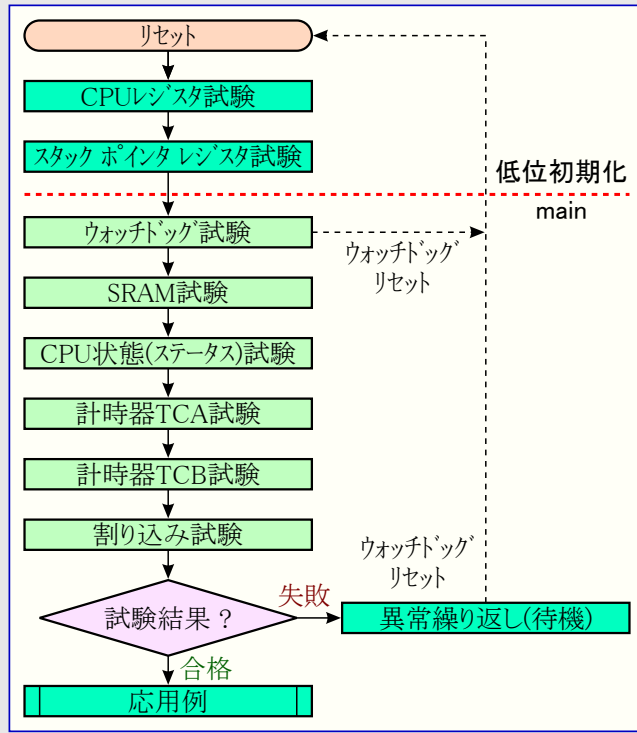
例の主なファイルは次の通りです。

- **main.c**ファイルは主応用です。
- **low_level_init.c**ファイルはC文脈の初期化の前に実行されなければならない等級B試験関数を組み込みます。
- **classB.c**ファイルは等級B試験の内容です。

注: この例はAtmel Studio 7.0.1006版(またはそれ以降版)またはIAR 6.80版(またはそれ以降版)が必要です。

試験全体の手順は次図で示されます。CPUレジスタとスタックポインタレジスタの試験は主関数へ入る前に低位初期化中に実行されます。他の試験は主関数で実行されます。

図5-1. 試験全体の手順



5.1. 試験ルーチン

一連の試験ルーチンは代表的な試験の範例となるように実装されます。試験項目とtinyAVR(ATtiny817)での対応するAPIは下表で示されます。

表5-1. tinyAVR例での試験ルーチン

試験項目	API	ソースファイル
CPUレジスタ試験	<code>_low_level_init()</code>	<code>low_level_init.c</code>
スタックポインタレジスタ試験	<code>_low_level_init()</code>	<code>low_level_init.c</code>
ウォッチドッグ試験	<code>watchdog_test()</code>	<code>classB.c</code>
SRAM試験	<code>sram_test()</code>	<code>classB.c</code>
CPU状態(ステータス)試験	<code>cpu_status_test()</code>	<code>classB.c</code>
計時器TCA試験	<code>timer_tca_test()</code>	<code>classB.c</code>
計時器TCB試験	<code>timer_tcb_test()</code>	<code>classB.c</code>
割り込み試験	<code>interrupt_test()</code>	<code>classB.c</code>

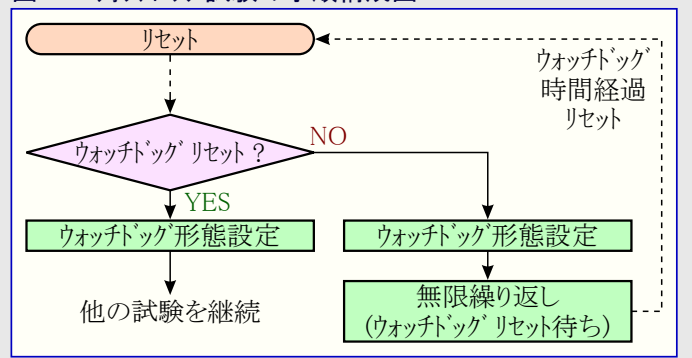
5.1.1. `_low_level_init()`

汎用レジスタとスタックポインタレジスタでの試験は`main()`関数に先立つ初期化中に実行される`_low_level_init()`で実装されます。この試験はR31/R30から始め、その後にR1/R0へ下って、そしてスタックポインタレジスタで終わります。

5.1.2. `watchdog_test()`

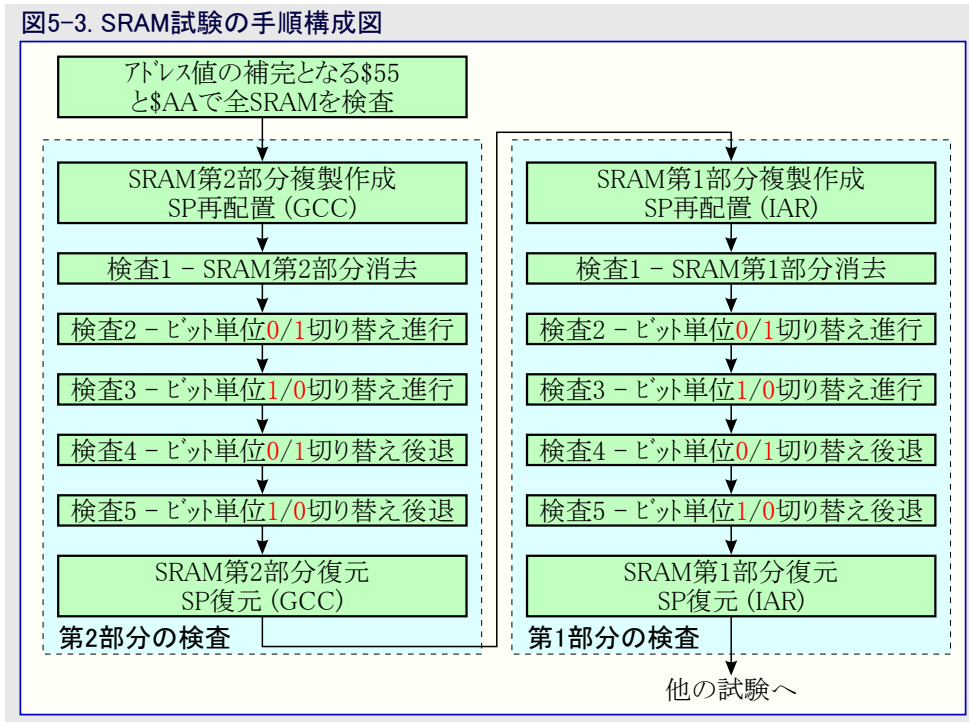
ウォッチドッグ試験は右図で示される手順で`watchdog_test()`に於いて実演されます。

図5-2. ウォッチドッグ試験の手順構成図



5.1.3. sram_test()

SRAM試験は下図で示される手順でsram_test()に於いて例示されます。



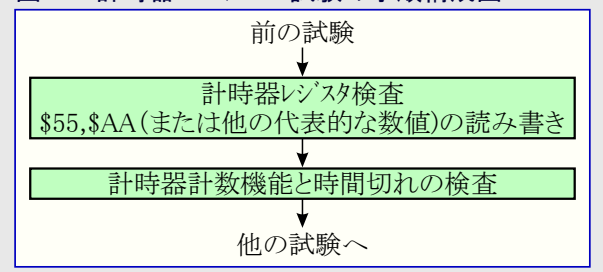
5.1.4. cpu_status_test()

CPUステータスレジスタ(SREG)はcpu_status_test()で検査されます。SREGは検査データが書かれる前に複製保存され、検査後に復元されるべきです。

5.1.5. timer_tca_test()

計時器TCAは右図で示される手順でtimer_tca_test()に於いて例示されます。

図5-4. 計時器TCA/TCB試験の手順構成図



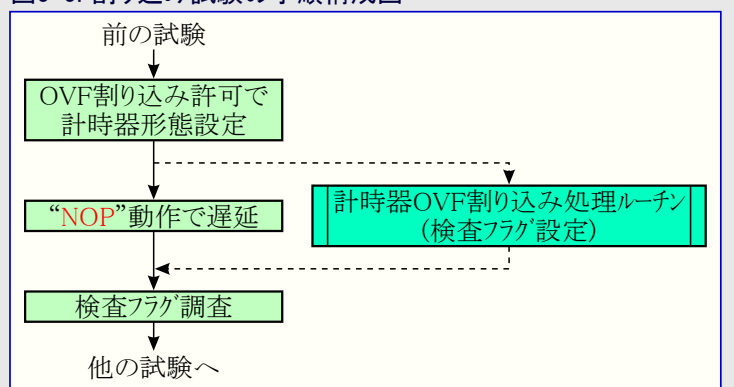
5.1.6. timer_tcb_test()

計時器TCBは図5-4.で示される手順でtimer_tcb_test()に於いて例示されます。

5.1.7. interrupt_test()

割り込み試験は右図で示される手順でinterrupt_test()に於いて例示されます。計時器溢れ(OVF)割り込みがこのルーチンで使用されます。お客様は特定応用に従って何の割り込みが使用されるべきかを決定すべきです。

図5-5. 割り込み試験の手順構成図



5.2. メモリ使用量と走行周期数

試験ファームウェアのメモリ使用量と走行周期数はどのコンパイルが使用されるだけでなく、その形態設定にも依存します。この項はどの位の量のフラッシュメモリとSRAMが必要とされ、全ての試験を走行するのにどの位の周期が費やされるのかの概要を提供します。データは最適化レベル”-Os”でのAtmel Studioで採取され、全てのデータは使用者への参考専用です。

表5-2. メモリ使用量と周期数

試験ルーチン	フラッシュメモリ量 (バイト)	SRAM量 (バイト)	周期数
watchdog_test()	36	0	27 + ウォッチドッグ時間切れ時間
sram_test()	934	0	1404727
cpu_status_test()	32	0	36
timer_tca_test()	916	2	684
timer_tcb_test()	382	2	386
interrupt_test()	152	1	93
_low_level_init() (CPU汎用レジスタとSPレジスタ)	812	0	344

6. 応用記述参考

- AVR040:電磁適合性(EMC)設計の考察 - <http://www.atmel.com/images/doc1619.pdf>
- AVR042:AVRハードウェア設計の考察
- http://www.atmel.com/images/atmel-2521-avr-hardware-design-considerations_applicationnote_avr042.pdf
- AVR132:強化したウォッチドッグタイマの使用法 - <http://www.atmel.com/images/doc2551.pdf>
- AVR180:外部低電圧(ブラウンアウト)保護 - <http://www.atmel.com/images/doc1051.pdf>
- AVR236:プログラムメモリのCRC検査 - <http://www.atmel.com/images/doc1143.pdf>
- IEC60730:民生と同様の使用に関する自動的な電氣的管理 - http://ulstandards.ul.com/standard/?id=60730-1_4
- 行進B:静的乱入出力メモリ(SRAM)検査についての様々な文献がインターフェースで利用可能です。この資料の付属書がクラスB試験を記述します。
- AVR3004:安全機能を持つQTouch® - <http://www.atmel.com/images/doc42041.pdf>
- AVR1610:XMEGA®でのIEC60730等級B適合への指針 - <http://www.atmel.com/images/doc42008.pdf>

7. 追補

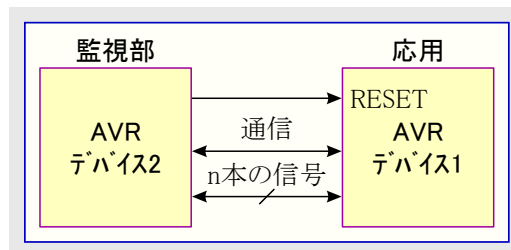
7.1. 安全な設計方法

設計を確実にする最良の方法は2つのデバイスでお互いを検査することで2重チャンネル形態設定を使用することです。以下の図はこのような実装を示します。勿論、2重チャンネル実装は単一のものよりもっと高価ですが、各デバイスが他のデバイスが何を行っているかを調べることができるために最良の保護適用範囲を持ち、そしてどの発振器の問題(CPUまたはウォッチドッグの発振器)も防ぐために外部ウォッチドッグとして使用されることも許します。

2重形態設定で安全性を増すことができます。下の例は応用を安全にするためにどの機能的な内部接続の種類を実装することができるかを示します。

- デバイス2は通信方法または周期的な外部信号時間超過によってデバイス1の機能を調べ、その後何らかの問題でデバイス1をリセットすることができます。
- デバイス1もどれかの外部信号時間超過によってデバイス2が走行していることを調べることができます。
- デバイス1が外部的な活動の実行を望む時に、活動が行われることを追認するためにデバイス2へ(SPI,TWI,UARTなどのような利用可能な何れかの手段によって)メッセージを送ることができます。その後デバイス2は活動の結果を調べ、活動が正しく行われたことをデバイス1に通知します。
- デバイス2はシステム障害の場合の分析のためにデバイス1の検査結果を保存することもできます。

デバイス2は監視部だけの役割を持ち、それは非常に安価にできます。例えばATmega128を監視するのにATtiny13を使用することができます。



7.2. 行進B試験説明

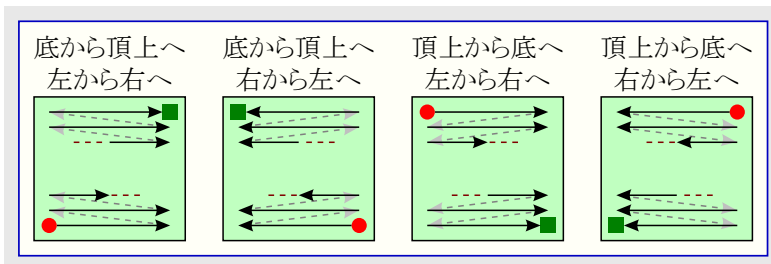
静的乱入出力メモリ(SRAM)検査は行進B算法を使用して実行されます。行進約定を使用するこの算法は次のように記述することができます。

{下(w0); 上(r0,w1,r1,w0,r0,w1); 上(r1,w0,w1); 下(r1,w0,w1,w0); 下(r0,w1,w0)}

意味:

1. 最後から最初のビットまでメモリ全体を'0'に初期化します。
2. 各ビットに対して、最初から最後まで、'0'読み、'1'書き、'1'読み、'0'書き、'0'読み、その後に次のビットへ行きます。
3. 各ビットに対して、最初から最後まで、'1'読み、'0'書き、'1'書き、その後に次のビットへ行きます。
4. 各ビットに対して、最後から最初まで、'1'読み、'0'書き、'1'書き、'0'書きその後に次のビットへ行きます。
5. 各ビットに対して、最後から最初まで、'0'読み、'1'書き、'0'書き、その後に次のビットへ行きます。

算法で指定された歩み順(下、上、上、下、下)は論理的なものではなく、物理的なアドレスを参照します。これは物理的に4つの実行に対応し得る最初の'下'に対する相対です。



故障検出率

この算法はSRAM領域内の以下の失敗処理を捉えることができます。

縮退故障(SAF:Stuck At Faults): ビットは状態が動かず、書くことができません。

遷移故障(TF:Transition Faults): その状態で一度、状態が動かない。例えば、ビットの値は'1'で、それは'0'に書くことができますが、一旦'0'になると、それ以降、'1'に設定することができません。

反転結合故障(CFin:Inversions Coupling Faults): ビットでの遷移が2つ目のビットの状態を反転します。

べき乗結合故障(CFid:Idempotent Coupling Faults): ビットでの遷移が2つ目のビットを或る状態に強制します。

結合故障短絡(BF:Coupling Faults Bridgling): 2つビットが短絡。これらのビットの結果の状態はこれらのビットの直前の状態間の論理積(AND)または論理和(OR)です。

状態結合故障(SCF:State Coupling Faults): 結合したセルは結合したセルが与えられた状態の場合にだけ或る値に強制されます。

注: 繋がった故障の或る非常に特定の配置に於いて、それらの故障のいくつかは未検出になり得ます。これらの配置は1ppm未満の悪影響と見做されます。

この検査中に失敗を得ることはどの種類の区別もなしに前の故障の1つが見つかったことを意味します。

8. 改訂履歴

資料改訂	日付	注釈
7715A	不明	初版資料公開
7715B	2008年4月	
7715C	2016年9月	ATtiny817用ファームウェア支援を追加

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, megaAVR®, QTouch®, tinyAVR®, XMEGA®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。他の用語と製品名は一般的に他の商標です。

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のウェブサイトに表示する販売の条件と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、または利益と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。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2016.

本応用記述はAtmelのAVR998応用記述(Rev.7715C-09/2016)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。