AVR054：内蔵RC発振器の走行時校正

要点

- UART経由での内蔵RC発振器の校正
- 単一同期ﾊﾞｲﾄでの目的周波数の±2%以内への同期/校正
- 代替としての2重同期ﾊﾞｲﾄでの目的周波数の±1%以内への走行時同期/校正
- 調整可能なRC発振器付きの全てのAVRに対する支援
- 正しい同期/校正に必要とされる50%ﾃﾞｭｰﾃｨ ｻｲｸﾙ
- 低費用ｸﾛｯｸ元での動作条件変化に強いUART通信が可能

8ﾋﾞｯﾄ AVR® マイクロ コントローラ

応用記述

1. 序説

この応用記述はUART経由で内蔵RC発振器を校正する方法を記述します。使われる方法では毎回のﾒｯｾｰｼﾞ ﾌﾚｰﾑの開始で従節点(ﾉｰﾄﾞ)が主節点に同期化されます。これは例え内蔵RC発振器のような安価なｸﾛｯｸ元で走行している時でも、指定された限度内のﾎﾞｰﾚｰﾄで他の節点と通信することを従節点に許します。

既存AVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗの大多数は内蔵RC発振器で走行する能力を提供します。内蔵RC発振器周波数は殆どのAVRでﾃﾞﾊﾞｲｽに対するﾃﾞｰﾀｼｰﾄで指定された周波数の±1%以内へ走行時に校正することができます。この特徴は同期目的に理想的で、外部発振器に比べて重要な費用節約を提供します。

この実装が内蔵RC発振器の周波数を変えるのに同期信号を使い、更にUART部のﾎﾞｰﾚｰﾄを変えることに注意してください。この場合に於ける用語の「同期」と「校正」は本質的に同じことを意味し、同義に使われます。語句の選択は単に目的に対する関連にすぎません。

2. 動作の理屈 – 内蔵RC発振器

製品に於ける内蔵RC発振器は5Vまたは3.3Vのどちらかで校正されます。校正中に使われる動作電圧についての情報に関しては個別ﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。工場校正の精度は±3%または±10%以内です(ﾃﾞｰﾀｼｰﾄを参照してください)。精度に対する設計の要求がAtmelによる工場での標準校正によって提供され得るものを越える場合、RC発振器の第2校正を実行することが可能です。これを行うことにより、5.0版の内蔵RC発振器に対して±1%以内の周波数精度を得ることが可能です。従って第2校正は発振器の精度改善または周波数の仕立て上げを実行することができます。

2.1. ｸﾛｯｸ選択

AVRのﾋｭｰｽﾞ設定は使われるｼｽﾃﾑ ｸﾛｯｸ元を制御します。内蔵RC発振器を使うには対応するﾋｭｰｽﾞ設定が選択されなければなりません。ﾋｭｰｽﾞの概要はﾃﾞｰﾀｼｰﾄで利用可能です。

2.2. 基準周波数

後続項はAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗで利用可能な内蔵RC発振器の概要を提供します。

いくつかのAVRは1つのRC発振器を持ち、一方他はそれらから選ぶための4つまでの異なるRC発振器を持ちます。周波数は1MHz～9.6MHzの範囲です。充分な精度の内蔵RC発振器にするためにAVRのI/O領域内に発振校正(OSCCAL)ﾚｼﾞｽﾀが存在します。OSCCALﾚｼﾞｽﾀは1ﾊﾞｲﾄ幅です。このﾚｼﾞｽﾀの目的は発振器周波数の調整を可能にすることです。この調整はRC発振器を校正する時に利用可能です。

Atmelがﾃﾞﾊﾞｲｽを校正するとき、校正ﾊﾞｲﾄがﾃﾞﾊﾞｲｽの識票列に格納されます。RC発振器周波数が工程に依存するため、この校正ﾊﾞｲﾄはﾃﾞﾊﾞｲｽ毎に変わり得ます。ﾃﾞﾊﾞｲｽが複数の発振器を持つ場合、RC発振器の各々に対する校正ﾊﾞｲﾄが識票列に格納されます。

既定のRC発振器校正ﾊﾞｲﾄは殆どのﾃﾞﾊﾞｲｽに於いて始動で自動的に識票列から取得されてOSCCALﾚｼﾞｽﾀ内に複写されます。例えば、ATmega8の既定ｸﾛｯｸ設定は1MHz内蔵RC発振器で、このﾃﾞﾊﾞｲｽについては1MHz RC発振器に対応する校正ﾊﾞｲﾄが始動で自動的に設定されます。ﾋｭｰｽﾞが変えられ、その結果として既定設定の代わりに4MHz発振器が使われる場合、校正ﾊﾞｲﾄは手動でOSCCALﾚｼﾞｽﾀに設定されなければなりません。識票列から4MHz校正ﾊﾞｲﾄを読み、従ってそれをﾌﾗｯｼｭ ﾒﾓﾘまたはEEPROMの位置へ格納するのに書き込みﾂｰﾙを使うことができます。主ﾌﾟﾛｸﾞﾗﾑは走行時にこの位置を読んでOSCCAL内に複写します。

(訳注) 本書での表記「UART」は非同期直列通信を指し、当然USART部にも適用されます。

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. 2563C-04/08, 2563CJ4-02/21

2.3. RC発振器概要

発振器の基準周波数は分周されていない発振器周波数として定義されます。

経歴を通してAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗで様々なRC発振器が利用可能です。RC発振器の概要とﾃﾞﾊﾞｲｽ例が**表1-1.**で見えます。このﾃﾞﾊﾞｲｽ一覧は発振器形式によって分類され、そしてこれは発売時期によるそれらの分類とも大体等価です。調整可能な発振器付きのﾃﾞﾊﾞｲｽだけがこの表で一覧にされます。支援されるﾃﾞﾊﾞｲｽの完全な一覧についてはｿｰｽ ｺｰﾄﾞ内の"device_specific.h"ﾍｯﾀﾞ ﾌｧｲﾙを参照してください。

表2-1. 発振器周波数とﾃﾞﾊﾞｲｽ例付き各種内蔵RC発振器の特徴（発振器版による分類）

発振器版番号	代表ﾃﾞﾊﾞｲｽ	RC発振器周波数 (MHz)	CKDIV8	CLKPR
1.1	ATtiny12	1.2	×	×
1.2	ATtiny15	1.6	×	×
2.0	ATmega163	1.0	×	×
3.0	ATmega8	1.0, 2.0, 4.0, 8.0	×	×
3.1	ATmega64	1.0, 2.0, 4.0, 8.0	×	XDIV (注1)
4.0	ATmega169 (注2)	8.0	○	○
4.1	ATtiny13	4.8, 9.6	○	○
4.2	ATtiny2313	4.0, 8.0	○	○
5.0	ATmega169P	8.0	○	○

注1: 前置分周ﾚｼﾞｽﾀはこれらのﾃﾞﾊﾞｲｽでXDIVと名付けられています(訳補:分周係数もCLKPRと異なります)。

注2: ATmega169改訂A～E、ATmega169Pは5.0版発振器を使います。

2.3.1. 1.x版発振器

この版は校正することができるAVR用の最初の内蔵RC発振器です。これは1.2MHzから1.6MHzに渡る周波数で提供されます。校正ﾊﾞｲﾄは識票列に格納されていますが、始動で自動的に設定されません。OSCCALﾚｼﾞｽﾀの設定は走行時にﾌｧｰﾑｳｪｱによって処理されなければなりません。この版に於ける発振器周波数は動作電圧と温度に大きく依存します。

2.3.2. 2.x版発振器

この版は1.0MHzの周波数で提供されます。発振器周波数と動作電圧、温度間の依存性は1.x版と比べてかなり低減されています。

2.3.3. 3.x版発振器

発振器ｼｽﾃﾑは多数の発振器周波数を提供するために拡張されています。1,2,4,8MHzの周波数を持つ4つの異なるRC発振器がﾃﾞﾊﾞｲｽに存在します。この版は識票列から1MHzの校正ﾊﾞｲﾄを自動的に設定するのが特徴です。4つの異なるRC発振器が存在する事実のため、識票列に4つの異なる校正ﾊﾞｲﾄが格納されています。既定の1MHz以外の周波数が望まれる場合、OSCCALﾚｼﾞｽﾀは走行時に対応する校正ﾊﾞｲﾄを設定されるべきです。

2.3.4. 4.x版発振器

8MHzの単一周波数が4.0版で提供されます。後期の4.x版についてはATtiny2313に対して4MHzと8MHz、ATtiny13に対して4.8MHzと9.6MHzの2つの周波数が提供されます。OSCCALﾚｼﾞｽﾀは変更され、その結果として選択した発振器に対する周波数調整に7ﾋﾞｯﾄだけが使われます。MSBは使われません。既定校正値の自動設定が存在し、CLKPRはCKDIV8ﾋｭｰｽﾞに従って自動的に設定されます。

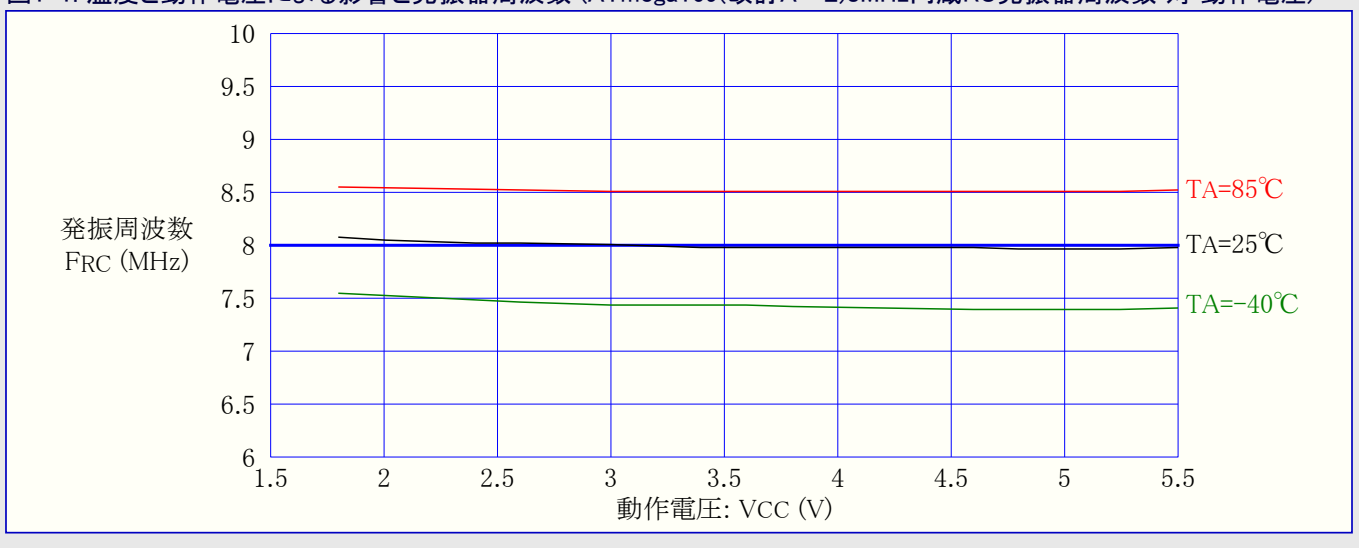
2.3.5. 5.x版発振器

8MHzの単一周波数が5.0版で提供されます。OSCCALﾚｼﾞｽﾀ内の全8ﾋﾞｯﾄが発振器周波数調整に使われます。既定校正値の自動設定とｼｽﾃﾑ ｸﾛｯｸ前置分周器が存在します。OSCCALﾚｼﾞｽﾀは2つの部分に分割されています。OSCCALのMSBは2つの重なっている周波数範囲の1つを選び、一方下位側7ﾋﾞｯﾄはこの範囲内で周波数を調整するのに使われます。

2.4. 発振器特性

内蔵RC発振器の周波数は温度と動作電圧に依存します。この依存性の例は図1-1.で見られ、そしてこれはATmega169(改訂A～E)の8MHz RC発振器の周波数を示します。この図から見られるように、周波数は温度上昇で増加し、動作電圧上昇で僅かに減少します。これらの特性はﾃﾞﾊﾞｲｽ毎に変わるでしょう。特定ﾃﾞﾊﾞｲｽの詳細についてはそのﾃﾞｰﾀｼｰﾄを参照してください。

図1-1. 温度と動作電圧による影響と発振器周波数 (ATmega169(改訂A～E)8MHz内蔵RC発振器周波数 対 動作電圧)



調整可能な発振器付きの全ﾃﾞﾊﾞｲｽは発振器周波数を調整するためのOSCCALﾚｼﾞｽﾀを持ちます。OSCCALの値増加は周波数に於いて"擬似単調"増加に帰着します。これを擬似単調と呼ぶ理由はOSCCAL値の或る単一増加に関して、周波数が増加しない、または僅かに減少するからです。けれども、次の単一増加は常に再び周波数を増加します。換言すると、OSCCALﾚｼﾞｽﾀを1増加することは周波数を増加しないかもしれませんが、OSCCAL値を2増加することは常に周波数を増加します。この情報は与えられた周波数に合致する最良の校正値を見つける時に大いに関係します。OSCCAL値と発振器周波数間に関連する擬似単調の例は図1-2.で見られ、そしてこれはATmega169の8MHz RC発振器です。ATmeg169でOSCCALﾚｼﾞｽﾀが発振器の調整に7ﾋﾞｯﾄだけを使うので(ATmeg169Pでは8ﾋﾞｯﾄ)、最高周波数はOSCCAL=127に対応します。図1-3.で示されるようにOSCCALのMSBが2つの周波数範囲の1つを選択するため、5.x版発振器はこの記述と少し違います。周波数範囲の各々の内(MSB一定)で、5.x版発振器は他の版の発振器と同じ擬似単調特性を示します。

図1-2. OSCCAL値の関数としてのATmega169校正付きRC発振器周波数 (8MHz内蔵RC発振器周波数 対 OSCCAL値)

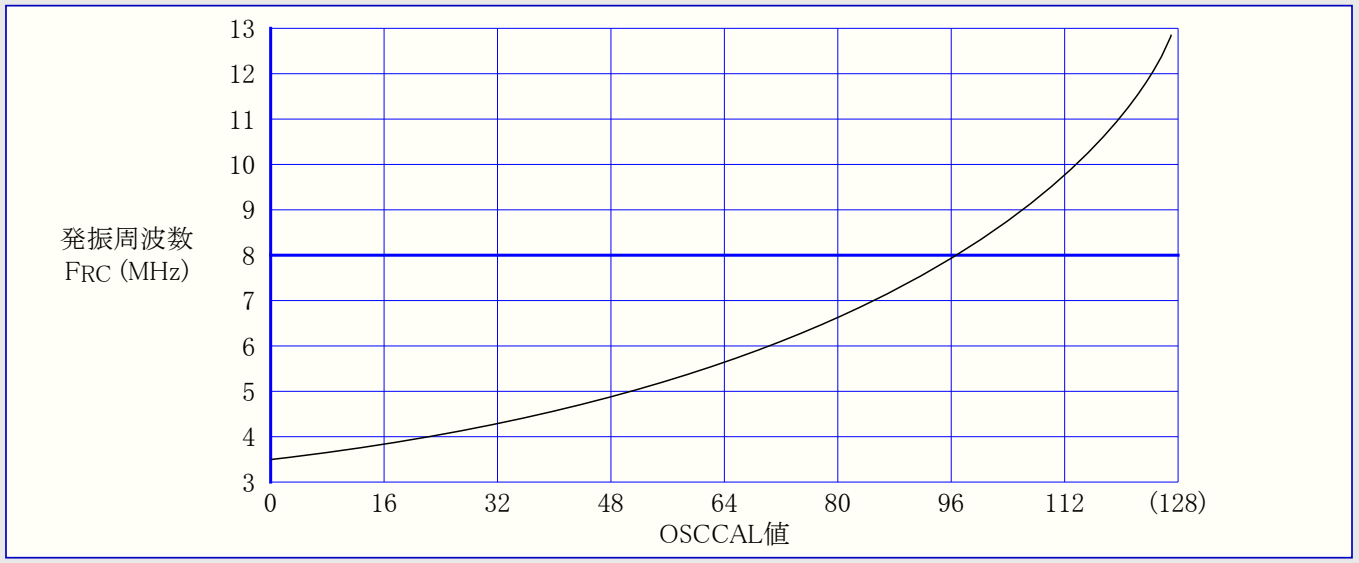
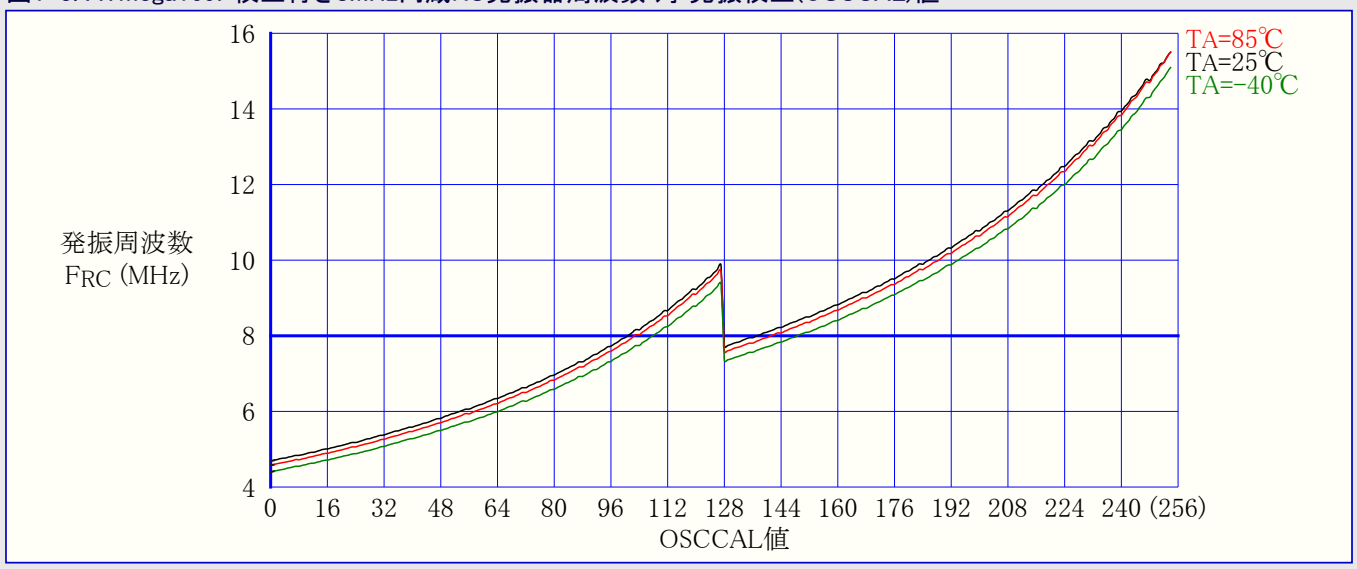


図1-3. ATmega169P校正付き8MHz内蔵RC発振器周波数 対 発振校正(OSCCAL)値



全ての調整可能な発振器について、ﾃﾞｰﾀｼｰﾄで指定された基準周波数を10%よりももっと外れる発振器の調整は推奨されないことに注意することが重要です。これの理由はﾃﾞﾊﾞｲｽの内部ﾀｲﾐﾝｸﾞがRC発振器周波数依存だからです。

RC発振器の基本特性を知れば、何れかの動作電圧と何れかの温度に於いて±1%の精度で基準周波数の10%以内で与えられた周波数にRC発振器を校正する効率的な校正ﾙｰﾁﾝを作ることが可能です。

2.5. 周波数安定時間

新しいOSCCAL値が設定されてしまうと、新しい周波数で安定させるために内蔵RC発振器に対して幾らかの時間がかかり得ます。この安定時間はRC発振器の各版で変わります。通常、発振器はOSCCALでの大きな変更よりも小さな変更に対してより早く安定します。この安定時間は5μsよりも長いことは決してありません。校正に関してどんな周波数測定をするのにも先立って、発振器を新しい周波数で安定させて置いてください。

3. UART同期法

信頼に足りる通信は内蔵RC発振器のような安価なｸﾛｯｸ元を使う時でも可能です。このようなｸﾛｯｸ元の固有の不正確さと環境依存特性のため、同期手段内に同期判定が含められなければなりません。

1つの主節点(ﾉｰﾄﾞ)と多数の従節点から成る網(ﾈｯﾄﾜｰｸ)では主節点にﾊﾞｽ上の全ての通信を制御する責任があります。ﾊﾞｽ上にﾒｯｾｰｼﾞ ﾌﾚｰﾑを送ることによって通信が発生します。ﾌﾚｰﾑ ﾍｯﾀﾞで始まる毎回のﾒｯｾｰｼﾞ ﾌﾚｰﾑは主節点によって開始されます。ﾍｯﾀﾞはBREAKとSYNCHの様式で始まり、ﾊﾞｽ上でのどの通信にも先立って主装置に同期することを従装置で可能にします。BREAK/SYNCH様式は以下から成ります。

- BREAK (中断) 信号：最低13ﾋﾞｯﾄ時間の優性(Low)値。図3-1.をご覧ください。
- BREAK DELIMITER (中断分離子)：最低1ﾋﾞｯﾄ時間の劣性(High)値。図3-1.をご覧ください。
- SYNCH (同期) ﾊﾞｲﾄ：$55が送信されます。開始と停止のﾋﾞｯﾄを含み、これは0101010101のﾋﾞｯﾄ様式送信の結果になります。図3-2.をご覧ください(送出ﾋﾞｯﾄ順がLSB先行であることに注意してください)。

図3-1. BREAK信号

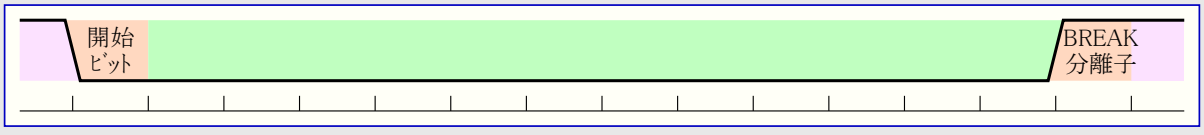
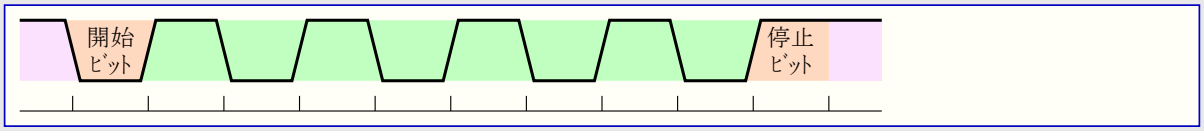


図3-2. SYNCHﾊﾞｲﾄ



SYNCHﾊﾞｲﾄ後、識別子が送信されます。識別子はﾊﾞｽ上のﾃﾞｰﾀ送信をどの従節点が支援されるのかと、何の情報が従節点に要求されるのかを唯一的に定義します。

以降ではSYNCH中の2つの下降端間の信号が同期周期と呼ばれます。この用語の使用で、SYNCHﾊﾞｲﾄは5つの同期周期から成ります。

4. 2分検索と近傍検索

3つの異なる校正/同期法がこの応用記述に記述されています。3つ全てに共通するのはOSCCAL値が計算される方法です。

4.1. 検索法

2分検索は指定された精度制限内で内蔵RC発振器に望む周波数を生成させるOSCCAL値を計算するのに使われます。

4.1.1. 2分検索

2分検索は以下の方法で動きます。

1. 検索範囲の中央のOSCCAL値で始めて下さい。
2. 段階量を検索範囲の1/4に設定してください。
3. 現在の周波数が高すぎるかまたは低すぎるかを判定してください。
4. 現在の周波数が高すぎるならばOSCCALから段階量を引き、低すぎるならばOSCCALに段階量を足してください。周波数が精度制限内なら、OSCCALを変更しないでください。
5. 段階量を2で割ってください。
6. 段階量が0なら、検査は完了です。検索を中止するか、または(後で記述される)近傍検索へ飛んでください。
7. 手順3.へ飛んでください。

この検索法は厳密に単調な関連を持つﾃﾞｰﾀを検索する時(最悪実行時間になる時)に最適です。既に知っているように、OSCCALと結果の発振器周波数間の関連は完全な単調ではありません。けれども、それは最適なOSCCAL値の近隣の値を探すための最も効率的な方法である2分検索に対して充分な近さです。この位置から最適値を探すのは容易です。

4.1.2. 近傍検索

これを行うには近接値が調べられなければなりません。ｸﾛｯｸ周波数に於ける増加は2以上の値によるOSCCAL増加時に保証されるため、段階量が1の時の最終繰り返しでだけ問題が予測されます。この場合では増加されたOSCCAL値がｸﾛｯｸ周波数での減少の結果、またその逆も同様になるかもしれません。同じ方向でのもう1つ(都合2)は望む方向での周波数変更を保証します。けれどもこの段階が計数器に対する充分大きな最終段階の影響を与えるかもしれません。従って、最適なOSCCAL値を得ることを確実にするため、もう1つの段階までも行われるべきです。そしてそれらの繰り返しの各々のﾀｲﾏ/ｶｳﾝﾀ0値が保存され、望むｸﾛｯｸ周期数と比較されます。望む周波数に最も近いｸﾛｯｸ周波数を生じるOSCCAL値が使われます。これは近傍検索として参照されます。

4.1.3. SYNCH信号での2分検索の使用

SYNCHﾊﾞｲﾄは5つの同期周期から成ります。完全な同期周期を測定して新しいOSCCAL値を計算し、そしてそれを再び試行する場合、2回だけの2分検索繰り返しが可能です。測定後に計算が実行されなければならず、新しいOSCCAL値は設定されなければならず、そして発振器は新しい周波数での安定を許されなければなりません。

その代わりとして、現在の発振器周波数を計算するのに、RXD線がLowの時のﾋﾞｯﾄ時間を測定することができます。そしてRXD線がHighの時の残された時間は新しいOSCCAL値を計算するのに使われ、そして発振器を新しい値で安定にさせます。主装置でのLowとHighのﾋﾞｯﾄ時間は50%のﾃﾞｭｰﾃｨ ｻｲｸﾙを要求する長さに等しいことが必要です。これは大部分のUARTに関する場合です。

実行され得る最大検索繰り返し数がﾊﾞｽに送信された同期周期数と等しいことに注意してください。

4.1.4. 範囲(検索対象数)

式4-1.は2分検索の間でのOSCCAL値の最大変化を示します。

式4-1. 2分検索の間でのOSCCALの最大変化

$$Cmax = \sum_{i=0}^{n-1} 2^i = 2^n - 1$$

ここでの$Cmax$は最大変化、nは繰り返し数です。そして1つの2分検索の総範囲は$r=2\times Cmax-1=2^{n+1}-1$値(個)に及びます。

近傍検索が使われる場合、同期の最後で2つの同期周期がそれに対して予約されなければなりません。それらは2分検索に利用可能な周期から減じられなければなりません。8ﾋﾞｯﾄのOSCCALﾚｼﾞｽﾀの範囲全体を網羅するためには、2分検索での7回の繰り返しが必要とされ、7つの同期周期を使います。近傍検索が含められるなら、もう2つの周期が必要とされ、合計9つにさせます。7ﾋﾞｯﾄのOSCCALﾚｼﾞｽﾀに対応する値は近傍検索なしで6、近傍検索付きで8です。

4.1.5. 精度

検索の精度は使われる検索方法に依存します。

2分検索だけが使われる場合、最適なOSCCAL値が常に見つかるとは限りません。けれども、見つけた周波数は目的周波数の±1%に校正できるﾃﾞﾊﾞｲｽに対して、望む周波数の未だ±2%以内であるべきです。2分検索が単独で使われるとき、検索は望む精度制限内のｼｽﾃﾑ ｸﾛｯｸ周波数が得られた時に中止されます。これはOSCCALと内蔵RC発振器周波数間の擬似単調関係による最後の段階での問題を避けるために行われるべきです(**訳補**:そうでなければOSCCAL値の微小振動で無限反復の可能性があります)。

2分検索に加えて近傍検索が使われるとき、同期の時間での最適なOSCCAL値はそれが検索範囲の内側にあれば常に見つかります。そして結果のｸﾛｯｸ周波数はこの精度内に校正できるﾃﾞﾊﾞｲｽに関して、望む周波数の±1%以内でしょう。これを成し遂げるために、完全な一致が得られた、換言すると、同期信号がLowの区間中に測定されたCPU周期数が望むCPUｸﾛｯｸ周期数と正確に一致する場合に検索が中止されるだけです。例え望むものにより近いｸﾛｯｸ周波数を生成するOSCCAL値があったとしても、その違いを測定することはできないでしょう。従って更なる検索は必要ありません。

4.2. 単一SYNCHﾊﾞｲﾄ同期法

同期ﾊﾞｲﾄは5つの同期周期から成り、そしてこれは2分検索の5回の繰り返しに匹敵します。そして同期中に検索できる最大OSCCAL範囲は $r=2^6-1=63$値です。結果としての検索範囲が非常に制限されるため、単一SYNCHﾊﾞｲﾄ同期法が使われる時に近傍検索を使うことは推奨されません。2分検索によって63個のOSCCAL値が網羅されますが、これは可能な値の意義深い分割(検索)で、それは望む周波数の±2%以内への校正が可能です。

OSCCALは毎回の同期に先立って既定値に設定されるべきです。

4.3. 2重SYNCHﾊﾞｲﾄ同期

近傍検索付きで8ﾋﾞｯﾄのOSCCALﾚｼﾞｽﾀの全範囲を検索するには、近傍検索に対する2同期周期に加えて、2分検索に対する7同期周期が必要とされます。これは9同期周期、殆ど(開始ﾋﾞｯﾄと停止ﾋﾞｯﾄを含む)2同期ﾊﾞｲﾄに匹敵し、これはOSCCALﾚｼﾞｽﾀ全体の検索を実行することに耐えられ、ﾃﾞｰﾀｼｰﾄで指定された温度と電圧の全範囲での動作をﾃﾞﾊﾞｲｽに許します。OSCCAL値の全範囲を検索し得るために、毎回の同期に先立ってOSCCALは最大OSCCAL値の1/2値を格納されるべきです。

この方法を使うには単一SYNCHﾊﾞｲﾄが2つの連続的な同期ﾊﾞｲﾄに置き換えられなければなりません。この2連SYNCHﾊﾞｲﾄ中の全てのﾋﾞｯﾄ時間は等しくなければなりません。

4.4. 再送ﾌﾚｰﾑ同期

この同期手順は単一ﾊﾞｲﾄ同期手順と非常に類似しています。違いは従節点(ﾉｰﾄﾞ)が1回の試行で主節点に同期できることを保証されないことです。最初の同期試行が不充分な場合、主節点は従節点からのﾃﾞｰﾀを正しく受信できないでしょう。この場合にはﾒｯｾｰｼﾞ ﾌﾚｰﾑ間に主装置がﾊﾞｽ上に新しいBREAK/SYNCHを発行します。この繰り返されたBREAK/SYNCH信号に続く識別子が割り込まれたﾒｯｾｰｼﾞ ﾌﾚｰﾑの識別子と等しければ、これは最後に同期を失敗した従装置への信号です。この場合は従装置が既定OSCCAL値を格納することなく新しい同期を開始し、そして再び送信を試みます。既定OSCCAL値は最後の送信が成功した時にだけ格納されるべきです。

再送ﾌﾚｰﾑ同期法は全OSCCAL範囲の検索を可能にしますが、既定から遠いOSCCAL値に対して検索する時に多数の試行を必要とし得ます。

4.5. 校正/同期法の選び方

同期を実行するのにかかる最悪時間が上手く定義されているため、単一SYNCHﾊﾞｲﾄ同期法は実時間応用に対して上手く適合されます。

増された精度が必要とされる場合、2重SYNCHﾊﾞｲﾄ同期法が推奨されます。この方法も単一SYNCHﾊﾞｲﾄ同期法と同じ実時間特性で処理します。

応用が温度と供給電圧での大きな変化を扱わなければならないけれども、殆どの時間は一定条件で動作する場合、再送ﾌﾚｰﾑ同期が良好な代替になり得ます。けれども、最悪同期時間が平均的な同期時間よりももっと長いので、実時間応用に対しては推奨されません。

近傍検索を実装するには追加ｺｰﾄﾞが必要とされるため、これは追加の精度が現実に必要とされる時にだけ使われるべきです。

5. 実装

本項はAVRに実装することができる走行時校正法を記述します。

5.1. ハードウェア

UART部は単一SYNCHバイト実装に対して良好な選択になります。この応用記述はこの実装にUARTを使いますが、どの汎用入出力ピンも使うことができます。

タイミングを楽にするために外部割り込み0(INT0)がエッジ検出に使われます。従って同期タイミングを容易にするために、UARTのRXDピンがINT0ピンに接続されなければなりません。

5.2. ソフトウェア

後続項に於いて走行時校正を行うのに必要なファームウェアが記述されます。

5.2.1. 初期化

AVRがリセットされると、以下のように構成設定されなければなりません。

- UARTを初期化してください。
- INT0ピンを構成設定してください。
- 必要なら、EEPROMまたはフラッシュ メモリから既定OSCCAL値を読んでください。

5.2.2. BREAK信号の検出

BREAK信号は2つの方法で検出することができます。

AVRが休止形態のとき、INT0はLowレベルでの起動に設定されなければなりません。BREAK信号はバス上にSYNCHバイトが発行される前にAVRを起すのに充分な長さです。UARTのRXDピンがINT0ピンに接続されているので、BREAKがINT0割り込み処理ルーチン(ISR)によって扱われるのを保証するために、UART受信部は休止形態へ移行する前に禁止されるべきです。

AVRが走行しているとき、BREAK信号を検出するために"受信完了割り込み"が許可されなければなりません。UARTはバス上に送信されつつあるバイトとしてBREAK信号の解釈を試みるでしょうが、その時に停止ビットが検出されないため、BREAK信号の長さがフレーミング異常にさせるでしょう。これはU(S)ART制御/状態レジスタ(USRまたはUCSRA)内の"受信完了"(RXC)と"フレーミング異常"(FE)のフラグを設定(1)にさせます。"受信完了割り込み要求"が許可されているので、これはUART受信完了割り込み処理ルーチン(ISR)を走行させます。そしてFEフラグはBREAK信号が発行されたことを示します。

この構成設定はINIT0 ISRまたはUART RXC ISRのどちらでも、BREAK信号が検出される直ぐに走行することを保証します。図5-1.をご覧ください。"キャラクタ処理"部と"同期実行"部はこれらのISRに含められるべき機能を参照しますが、それらはBREAK検出に関係しません。"同期実行"部は実際に校正/同期を行う部分です。この部分は次項で網羅されます。"SYNCH用準備"部は図5-2.で示されます。これは両割り込み処理ルーチンに共通で、割り込み処理ルーチン内での関数呼び出しを避けるためにマクロとして実装されます。

図5-1. UART RXCとINT0の割り込み処理ルーチンでのBREAK検出

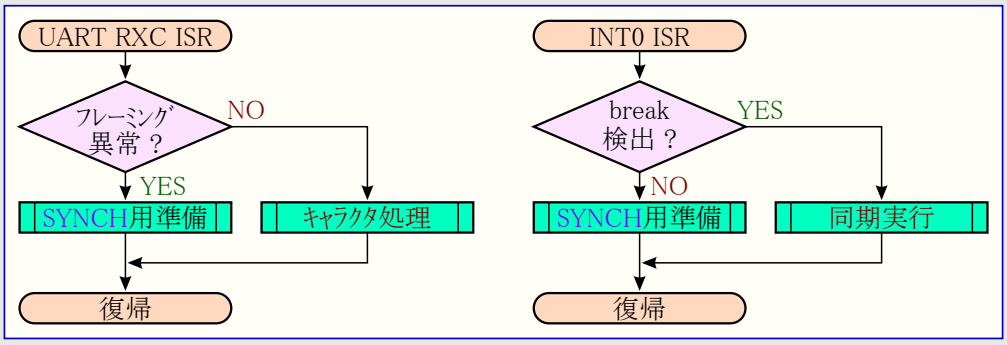
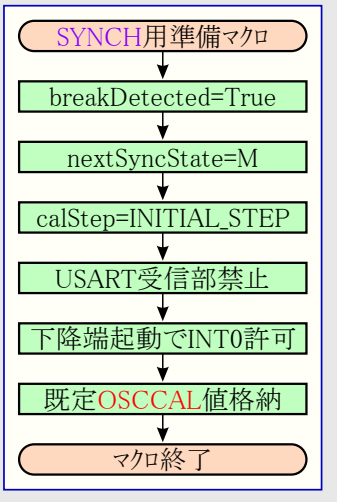


図5-2. "SYNCH用準備"マクロ



5.2.3. 同期

単一SYNCHﾊﾞｲﾄ同期法と2重SYNCHﾊﾞｲﾄ同期法に対する流れ図は図5-3.と図5-4.で示されます。再送ﾌﾚｰﾑ同期法はそれが単一SYNCHﾊﾞｲﾄ同期法への小さな変更を必要とするだけなので、ここで示されません。

この資料で記述される同期手法はSYNCH信号中にRXD線でﾎﾞｰﾚｰﾄを測定し、望むｸﾛｯｸ周波数と/またはﾎﾞｰﾚｰﾄを得るために内蔵RC発振器の周波数を変更します。UARTﾎﾞｰﾚｰﾄ ﾚｼﾞｽﾀは同期中に決して変更されません。

BREAK信号が検出された後、ﾃﾞﾊﾞｲｽはSYNCHﾊﾞｲﾄを処理するために準備すべきです。同期はINT0割り込み処理ﾙｰﾁﾝ(ISR)で実行されます。ﾀｲﾐﾝｸﾞが絶対的に正しいことが同期手法に関して重要です。従ってより高い優先権を持つ全ての割り込みが同期中に禁止され、割り込みが既定でAVRによって禁止されるISRの実行中を除き、"全割り込み許可"(I)ﾋﾞｯﾄは同期中何時も設定(1)されなければなりません。

ﾊﾞｽ上のﾋﾞｯﾄ時間毎のｸﾛｯｸ周期数を計数するのに8ﾋﾞｯﾄ ﾀｲﾏ/ｶｳﾝﾀ0が使われます。ﾎﾞｰﾚｰﾄに対する高いｸﾛｯｸ周波数の比率での走行時、8ﾋﾞｯﾄは1ﾋﾞｯﾄ時間中のｸﾛｯｸ周期を計数するのに充分ではないかもしれません。それなら第9ﾋﾞｯﾄとしてﾀｲﾏ/ｶｳﾝﾀ0溢れﾌﾗｸﾞを使うことができます。

INT0割り込み処理ﾙｰﾁﾝ(ISR)がBREAK検出と同期ﾀｲﾐﾝｸﾞの両方を実行するため、BREAKが検出された後で"breakDetected"全域ﾌﾗｸﾞ変数が(真に)設定されます。そしてこのﾌﾗｸﾞは同期処理が終了された時に解除されます。

第2の全域ﾌﾗｸﾞ変数"nextSynchState"は実際の同期を制御するために用いられます。このﾌﾗｸﾞの値はﾊﾞｽ上で次のｴｯｼﾞが検出された時にどの同期状態へ移行するかを決めます。測定(M)、2分検索(B)、近傍検索(N)の3つの状態が使われます。

"nextSynchState"はﾊﾞｽ上で上昇端を待つ時にMが設定されます。この状態でINT0割り込み処理ﾙｰﾁﾝ(ISR)へ移行すると、ﾀｲﾏ/ｶｳﾝﾀ0がﾘｾｯﾄされ、INT0は上昇端での起動に設定されます。そして"nextSynchState"ﾌﾗｸﾞはBかNのどちらかの状態に変更されます。

ﾊﾞｽ上で上昇端を待つ時に"nextSynchState"ﾌﾗｸﾞはBまたはNに変更されます。これらの状態でINT0割り込み処理ﾙｰﾁﾝ(ISR)へ移行すると、ﾀｲﾏ/ｶｳﾝﾀ0の値が読まれ、"nextSynchState"ﾌﾗｸﾞに対応する検索の1回の繰り返しが実行されます。これは検査のための新しいOSCCAL値に帰着します。これが検索の最終繰り返しでないなら、"nextSynchState"ﾌﾗｸﾞはMに設定され、INT0が下降端での起動に設定されます。

このようにしてこの方法は同期が終了されるまで状態間を切り換えます。

5.2.4. ﾀｲﾐﾝｸﾞの精度

INT0割り込み処理ﾙｰﾁﾝ(ISR)が走行する度にﾀｲﾏ/ｶｳﾝﾀ0が読みとﾘｾｯﾄの両方を行われるのは、図5-3.と図5-4.の流れ図から知ることができます。毎回の測定繰り返しの始めでﾀｲﾏ/ｶｳﾝﾀをﾘｾｯﾄし、毎回の評価的繰り返しの始めでそれを読むことがもっと直感的かもしれないでしょう。けれどもこれはｺｰﾄﾞの煩雑さを増し、それをより大きくより遅くさせ、予測可能なﾀｲﾐﾝｸﾞを低めるでしょう。同期が正しいことを確実にするために、読み込み宣言とﾘｾｯﾄ宣言間の周期数が正確に得られることが極めて重要です。

ﾀｲﾏ/ｶｳﾝﾀ0が8ﾋﾞｯﾄ動作で使われるなら、INT0割り込み処理ﾙｰﾁﾝ(ISR)内でのﾀｲﾏ/ｶｳﾝﾀの停止と開始は現実的に必要ありません。この場合はﾀｲﾏ/ｶｳﾝﾀが時間初期化で開始されなければなりません。けれども、第9ﾋﾞｯﾄとして溢れﾌﾗｸﾞが使われる場合、非分断操作としての第9ﾋﾞｯﾄの読み込みを保証するために、ﾀｲﾏ/ｶｳﾝﾀ ﾚｼﾞｽﾀと溢れﾌﾗｸﾞがﾘｾｯﾄされる間、ﾀｲﾏ/ｶｳﾝﾀは停止されなければなりません。

図5-3. 単一SYNCHﾊﾞｲﾄ同期法-流れ図

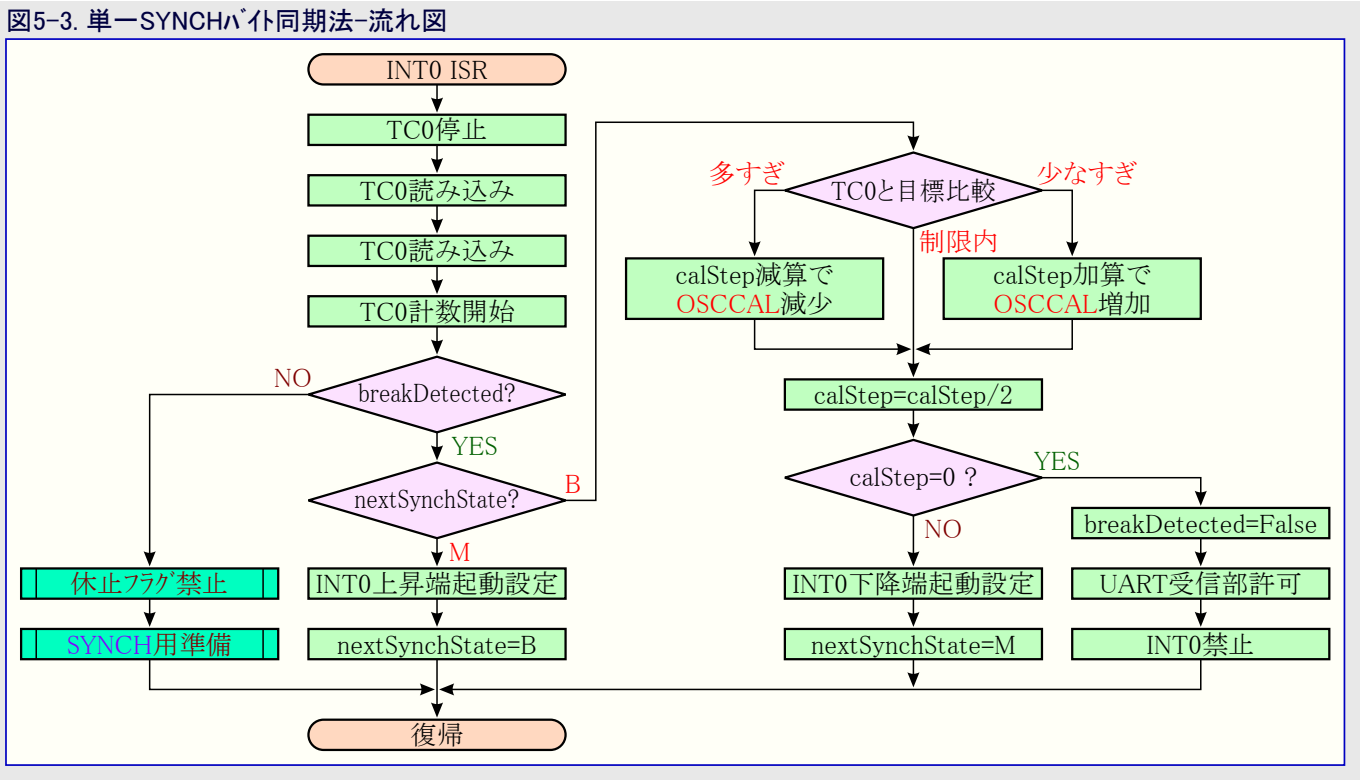
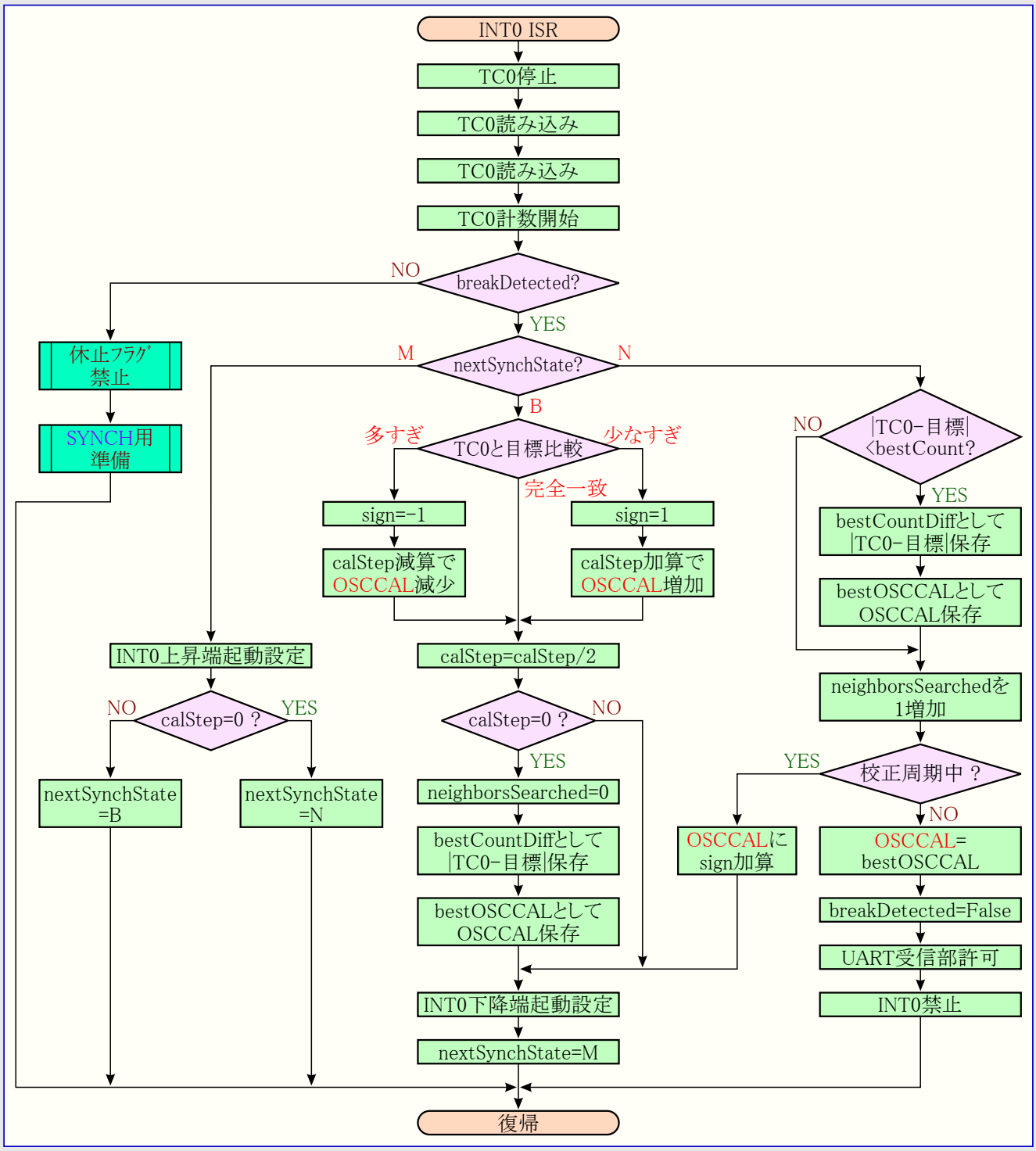


図5-4. 2重SYNCHﾊﾞｲﾄ同期法-流れ図



6. 動作成功の条件

後続項は望む校正精度を得るために従うべきいくつかの重要な指針を一覧にします。

6.1. ﾀｲﾐﾝｸﾞ精度

ﾀｲﾏ/ｶｳﾝﾀ0は整数値を計数するだけなので、切り捨て誤差が誘引されます。この切り捨てによって誘引される不正確さはｸﾛｯｸ周波数に対するﾎﾞｰﾚｰﾄの比率によって制限されます。この比率の低減がもっと正確なﾀｲﾐﾝｸﾞに帰着します。1%の周波数精度を得るにはこの比率が0.01よりも小さくなければなりません。これは19200bpsのﾎﾞｰﾚｰﾄが最低2MHzのｸﾛｯｸ周波数を必要とすることを意味します。

6.2. OSCCAL溢れ

測定は校正中にOSCCALﾚｼﾞｽﾀが変更される時に溢れが起こらないことを保証するように作られていません。走行時に溢れ検査を行う必要がないので、これは故意に行われています。その代わりとしてﾌﾟﾛｸﾞﾗﾐﾝｸﾞ時に於いて既定OSCCAL値がEEPROMまたはﾌﾗｯｼｭ ﾒﾓﾘに格納される時にそれを調べることが推奨されます。検索範囲がOSCCAL溢れの危険に足る大きさなら、結果として溢れが起き得ないようにEEPROMに格納される既定OSCCAL値を変更してください。これは単一SYNCHﾊﾞｲﾄ同期法だけに適用します。

6.3. UARTﾎﾞｰﾚｰﾄ生成

通信に対するUART部の使用に於いて、全てのｸﾛｯｸ周波数で望むﾎﾞｰﾚｰﾄに一致することが常に可能な訳ではありません。この不正確さをなくすために、校正目標周波数に於いて望むﾎﾞｰﾚｰﾄが正確に生成できることを保証してください。例えば、正確な2MHzのｸﾛｯｸ周波数で走行するとき、誤差7%よりも低い19.2kbpsのUARTﾎﾞｰﾚｰﾄを得ることは不可能です。正確な2,150,400Hzのｸﾛｯｸ周波数で19.2kbpsのﾎﾞｰﾚｰﾄは正確に生成することができます。この場合、意図が19.2kbpsで通信することなら、2MHzに対する校正が無駄になるでしょう。

6.4. ﾀｲﾏ/ｶｳﾝﾀ分解能

選択したﾀｲﾏ/ｶｳﾝﾀの分解能が同期信号のLow区間中に全てのｸﾛｯｸ周期を計数するのに充分なことが極めて重要です。必要とされる分解能は最高ｸﾛｯｸ周波数/ﾎﾞｰﾚｰﾄの比率によって指示されます。同期中に例えｸﾛｯｸ周波数が最高値に到達してもﾀｲﾏ/ｶｳﾝﾀの分解能が充分なことを保証してください。

6.5. 5.0版発振器

5.0版発振器でのOSCCALﾚｼﾞｽﾀは2つの部分に分割されているため、2分検索法は範囲全体を検索するのに適していません。これに含まれるｿｰｽ ｺｰﾄﾞはOSCCALﾚｼﾞｽﾀの半分だけの検索を支援します。活かす半分はｺﾝﾊﾟｲﾙ時に指示されなければなりません。範囲全体が検索を必要とする場合、ﾚｼﾞｽﾀの半分に対して1つの検索が実行されなければならず、そしてその結果は最適なOSCCAL値を選ぶために比較されなければなりません。この方法は含まれるｿｰｽ ｺｰﾄﾞに実装されていませんが、含まれるｿｰｽ ｺｰﾄﾞへの小さな変更で行うことができるでしょう。

7. ｺｰﾄﾞ資料と始める前に

ｿｰｽ ｺｰﾄﾞに含まれるdoxygen資料は即時開始の指針、ｺﾝﾊﾟｲﾗとﾌﾟﾛｼﾞｪｸﾄの設定、支援ﾃﾞﾊﾞｲｽについての情報を含みます。ｿｰｽはwww.atmel.com/products/avr/ からﾀﾞｳﾝﾛｰﾄﾞすることができます。資料は'readme.html'ﾌｧｲﾙを開くことによってｱｸｾｽされます。

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
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ｳｪﾌﾞｻｲﾄ
www.atmel.com

技術支援
avr@atmel.com

販売窓口
www.atmel.com/contacts

文献請求
www.atmel.com/literature

本応用記述はAtmelのAVR054応用記述(doc2564.pdf Rev.2564C-04/08)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訳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。