

# AVR205 : Atmel tinyAVRとAtmel megaAVRで 容易にさせる周波数測定

## 要点

- 10HzからTimer\_Clock\_frequency/2.5までの周波数測定
- 正確な測定: AVR®デバイス クロック元に依存して最大99%またはより良い
- 2つの測定方法: 多忙待ち(ホールド)と割り込み駆動

## 1. 序説

時間の単位に対して周期的に変化する信号での周期(周波数)測定は組み込み電子機器に於いて基本の活動です。この応用記述は可変デューティサイクルを持つどんな形式(正弦波、方形波など)の波形の周波数をも測定する方法を記述します。信号に対する基本的な必要条件是選択したAVRデバイスの入出力ピン閾値での振幅を持つことです。

動作用にここで提供されるコード例に関する2つ目の必要条件是選択したAVRデバイスが最低2つの計時器を持ち、その1つが外部クロック元で使えることです。それらの計時器の1つはソフトウェア遅延繰り返しによって作成することができます。この考えは或る計時器のクロック入力に測定されるべき信号を注入し、他の計時器を時間基準として使うことです。

図1-1. 1970年代の古典的な周波数計数器



## 2. コード走行の必要条件

この応用記述の一部として実装されるコードはAtmel®のtinyAVR®とAtmelのmegaAVR®に基づきます。AtmelのXMEGA®デバイスについては「AVR1617:Atmel XMEGA 事象システムの使い方」応用記述を参照してください。

上で簡単に言及したように、選択したデバイスで利用可能な2つの計時器が必要で、1つの計時器は外部供給元からクロック駆動できることが必要です。

図2-1. 同期回路として使われるAVR入力クロック回路図

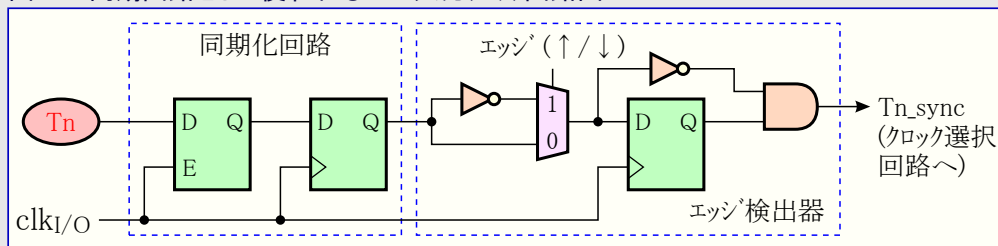


図2-1.は入力信号がどう採取されるかを示します。検知することができる外部クロックの最大周波数は採取周波数の半分(ナイキストの標本化定理)です。けれども、選択したクロック元(水晶発振子、セラミック振動子、コンデンサ)によって引き起こされるシステムクロック周波数とデューティサイクルでの変化のため、外部クロックの最大周波数はTimer\_Clock\_frequency/2.5より低いべきです。この応用記述では8MHzのRC発振器周波数がタイマ/カウンタクロックに供給されます。従って、この特定応用では測定されるべき推奨最大周波数は3.2MHz以下です。



8ビット AVR®  
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. 8365A-01/11, 8365AJ2-02/21

対称入力信号(50%デューティ サイクル)は先に記述したように最大3.2MHzを測定できるかもしれませんが、非対称入力信号については、入力のHighとLowの区間が1つの計時器クロックパルス幅よりも大きくなければならず、さもなければ先に示された同期回路が入力パルスを失って測定されつつある周波数での不正確を引き起こします。

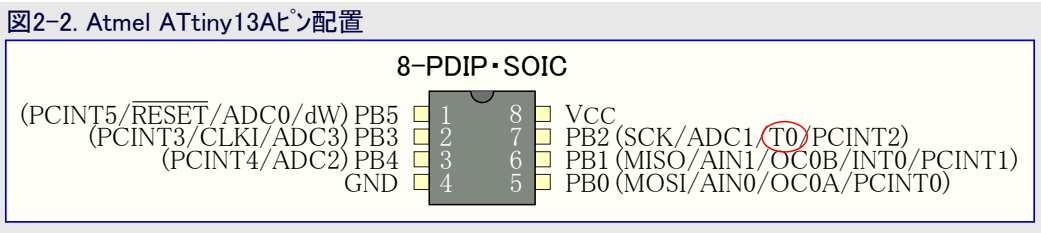
例えば、計時器を動かすのに8MHzクロックが使われる場合、クロックのHighとLowの区間は125nsです。従って、入力のHighとLowのパルス幅は125nsよりも大きくなければなりません。

この採取入力技術のより多くの詳細についてはtinyAVRとmegaAVRのデータシートをご覧ください。

## 2.1. タイマ/カウンタ クロック用外部入力を持つAtmel AVRデバイスの一覧

外部入力がタイマ/カウンタのクロックに使えるかを定める簡単な方法はAtmel AVRデバイスのピン配置図を観察してピン名称のT0を見つけることです。例については図2-2をご覧ください。

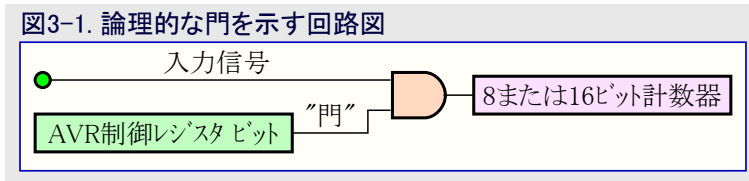
この機能を持つAtmel AVRデバイスの非包括的な一覧は、Atmel ATtiny10, ATtiny13, ATtiny20, ATtiny26, ATtiny28, ATtiny2313, ATmega48/88/168, ATmega8, ATmega16, ATmega32, ATmega8515/8535, ATmega163, ATmega164/324/644/1284, ATmega165/325/3250/645/6450を含みます。



## 3. 測定原理

この応用記述では“開門”区間が0.1s(100ms)に選ばれ、故に測定されるべき信号はこの時間区間に対して計数器をクロック駆動し、100msでの周波数計数を生成します。結果として生ずる周波数測定は実際の周波数の1/10です。

これはタイマ/カウンタの16ビット分解能の利用も選ばれ、この場合は測定する周波数が500Hzから5000Hzです。500kHzの周波数が測定されるべきなら、1msの開門時間が16ビット タイマ/カウンタの範囲の使用が良好な結果になるでしょう。



## 4. 2つの技法: 多忙待ちと割り込み駆動

### 4.1. 多忙待ち繰り返し

多忙待ち繰り返しは事象が起きるかを見るために繰り返し調べる繰り返しです。いくつかの応用は使われるべき割り込みの必要なしに、周回の繰り返しに関して0.1s(100ms)を待つことができます。

図4-1.はプログラムの主部に関する流れ図で、図4-2.はやって来る信号の周波数を実際に測定してそれを主(呼び出し)関数に返すC関数呼び出しに対する流れ図です。

図4-1. 主流れ図

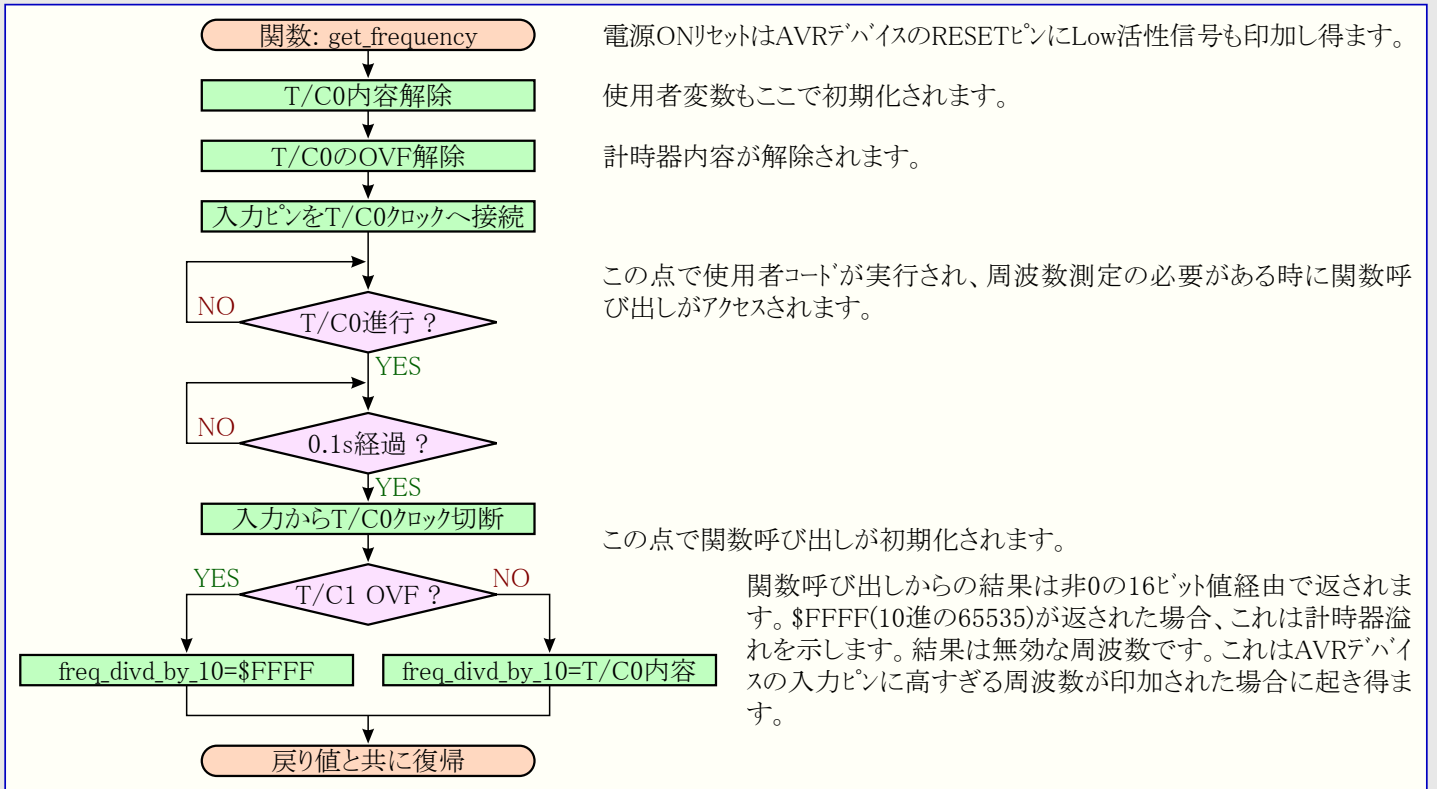


図4-2. 関数流れ図

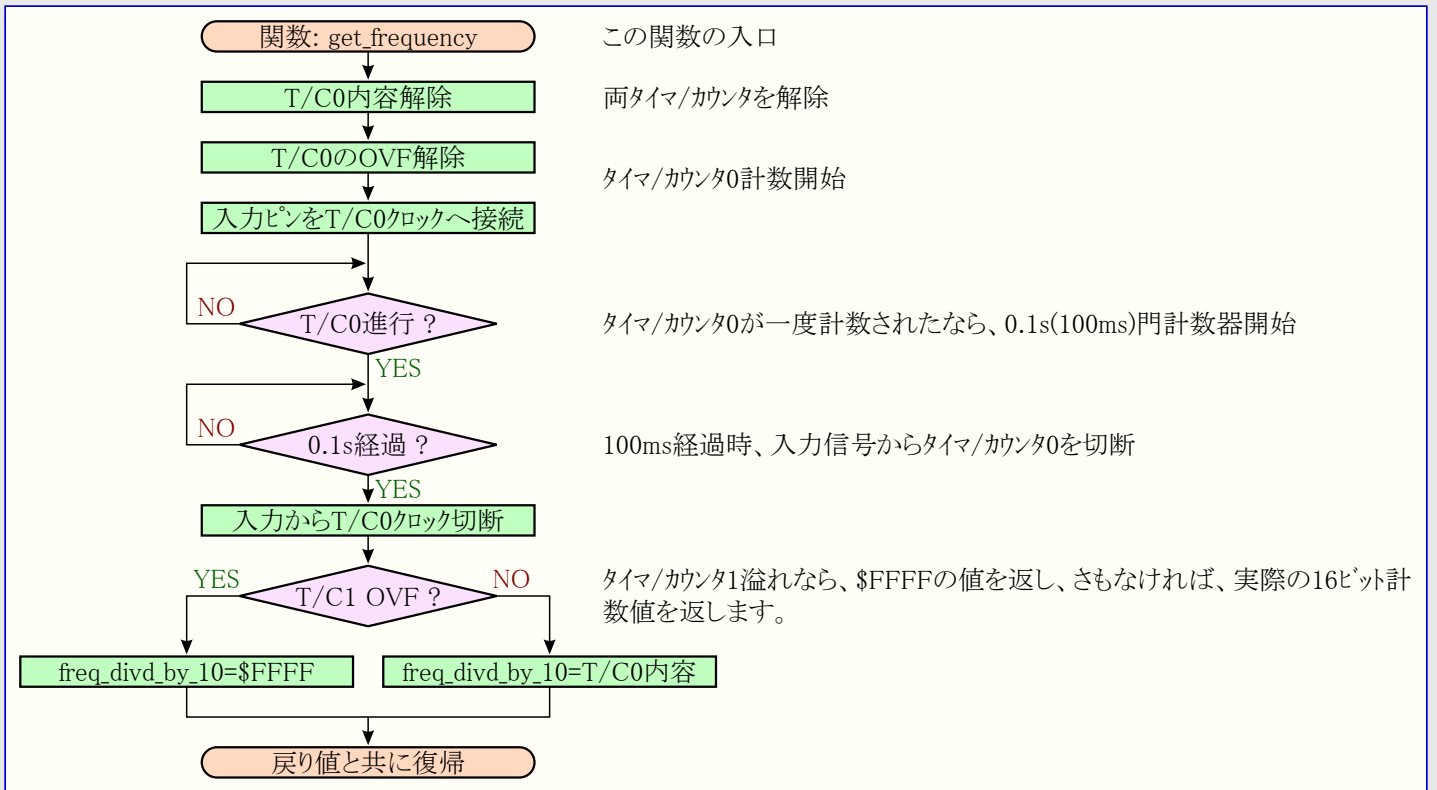
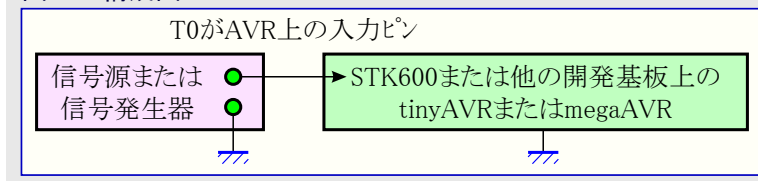


図4-3. 構成図



#### 4.1.1. 多忙待ちを用いるプロジェクト走行方法

このコードの開発にAtmelのAVR Studio 4.18が使われました。以降の版はコードを正しくコンパイルできるかも、またはできないかもしれません。WinAVR Cコンパイラ20100110版も使われました。以降の版はコードを正しくコンパイルできるかも、またはできないかもしれません。これは以下から入手可能です。

<http://sourceforge.net/projects/winavr/files/>

- 図4-1を参照してください。信号源をAVRのT0入力ピンに接続してください。
- AVR Studio 4を用い、プロジェクト(freq\_meter\_busy\_wait\_demo.apx)を開いてください。
- プロジェクトを構築してください。
- JTAGICEmk IIを用い、AVR Studioでデバッグ作業を開始し、C命令のdelay100msd(2);に中断点を設定してください。
- プロジェクトを走行し、中断点到達時、周波数計数結果はC変数のfreq\_div\_by\_10になります。
- AVRのタイマ/カウンタが100ms間計数するため、結果値は実際の周波数の1/10になります。周波数が算法能力を超える場合、返される結果は溢れを示す0xFFFFになります。

#### 4.1.2. 多忙待ちを用いるプロジェクト構築方法

事前必要条件：AVR Studio 4プロジェクトの構築、コンパイル、デバッグの方法の知識

- このプロジェクトを構築するには3つのファイルが必要です。この例でのそれらは以下です。
  - main()関数が置かれる、avr205\_frequency\_meter\_busy\_wait\_demo.c
  - 関数が置かれる、freq\_meter.c
  - この場合、8,000,000HzのCPUクロック速度を含む、ポートとレジスタの名前のようなものが指定される、freq\_meter.h
- ATtiny861やATmega48などのような、どのAVRかを指定してください。

#### 4.1.3. 多忙待ちを用い、これらの関数を呼び出して使う方法

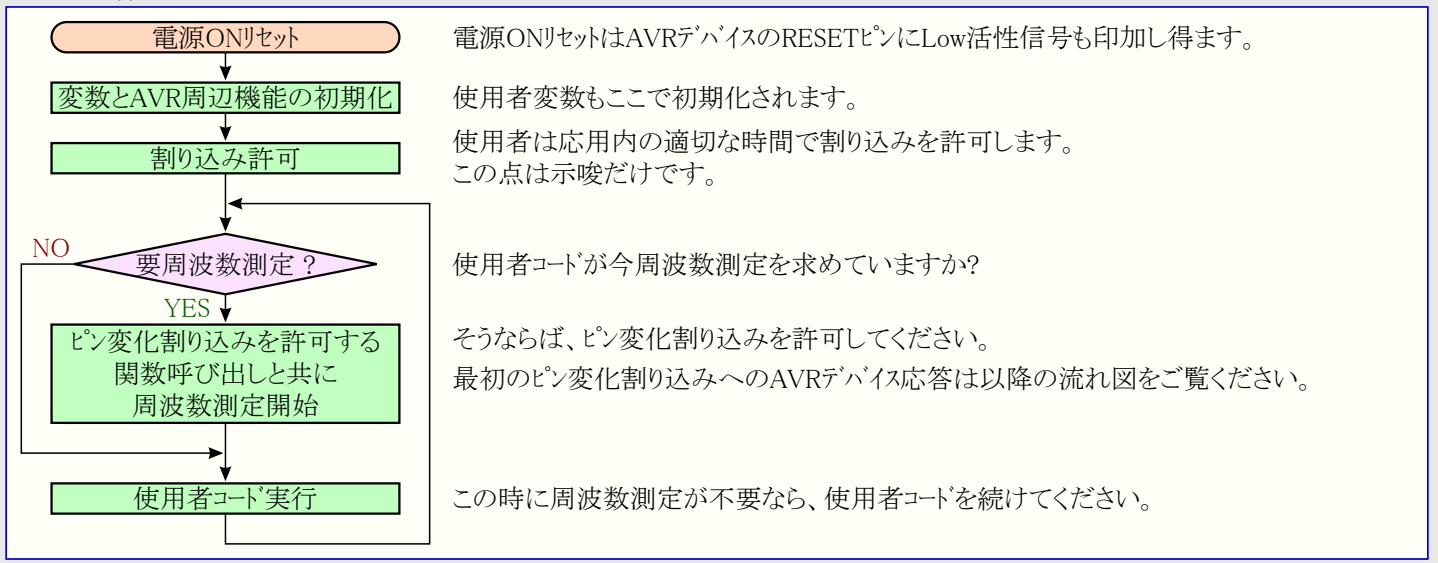
freq\_meter\_demo.cに於いて、

- freq\_meter\_init()関数はAVRの関連レジスタを初期化します。
- user\_init()関数は使用者変数を初期化します。
- この実演でAVR割り込みは使われません。
- freq\_cntr\_get\_frequency()の呼び出しはT0のピン変化で始める周波数測定を許可するためにT0でのピン変化割り込みを構成設定します。この関数の内側はdelay100ms(1)を呼び出します。
- 多忙待ち技法はdelay100ms(1)関数呼び出しによって実現されます。
- freq\_cntr\_get\_frequency()関数が再び呼ばれる前にdelay100ms(2)が使用者のコードを偽装します。

### 4.2. 割り込み駆動

多忙待ち近似法が受け入れられない時に、割り込みを使う実装が用いられるかもしれません。区間計時器の完了に同期して待つ代わりに、単純に割り込んで開門計数器を停止することができます。詳細については以下の図解をご覧ください。

図4-4. 主繰り返し



#### 4.2.1. freq\_meter\_interrupt\_demo 走行方法

このコードの開発にAtmelのAVR Studio 4.18が使われました。以降の版はコードを正しくコンパイルできるかも、またはできないかもしれません。WinAVR Cコンパイラ20100110版も使われました。以降の版はコードを正しくコンパイルできるかも、またはできないかもしれません。これは以下から独立して入手可能です。

<http://sourceforge.net/projects/winavr/files/>

- ・ 図4-1を参照してください。信号源をAVRのT0入力ピンに接続してください。
- ・ AVR Studio 4を用い、プロジェクト(freq\_meter\_interrupt\_demo.apx)を開いてください。
- ・ プロジェクトを構築してください。
- ・ JTAGICEmk IIを用い、AVR Studioでデバッグ作業を開始し、C命令のfreq\_cntr\_clear\_result();に中断点を設定してください。
- ・ プロジェクトを走行し、中断点到達時、周波数計数結果はC変数のfreq\_cntr\_resultになります。
- ・ 門が100msなので、結果値は実際の周波数の1/10になります。周波数が算法能力を超える場合、返される結果は溢れを示す0xFF FFになります。

#### 4.2.2. 割り込みを用いるプロジェクト構築方法

事前必要条件：AVR Studio 4プロジェクトの構築、コンパイル、デバッグの方法の知識

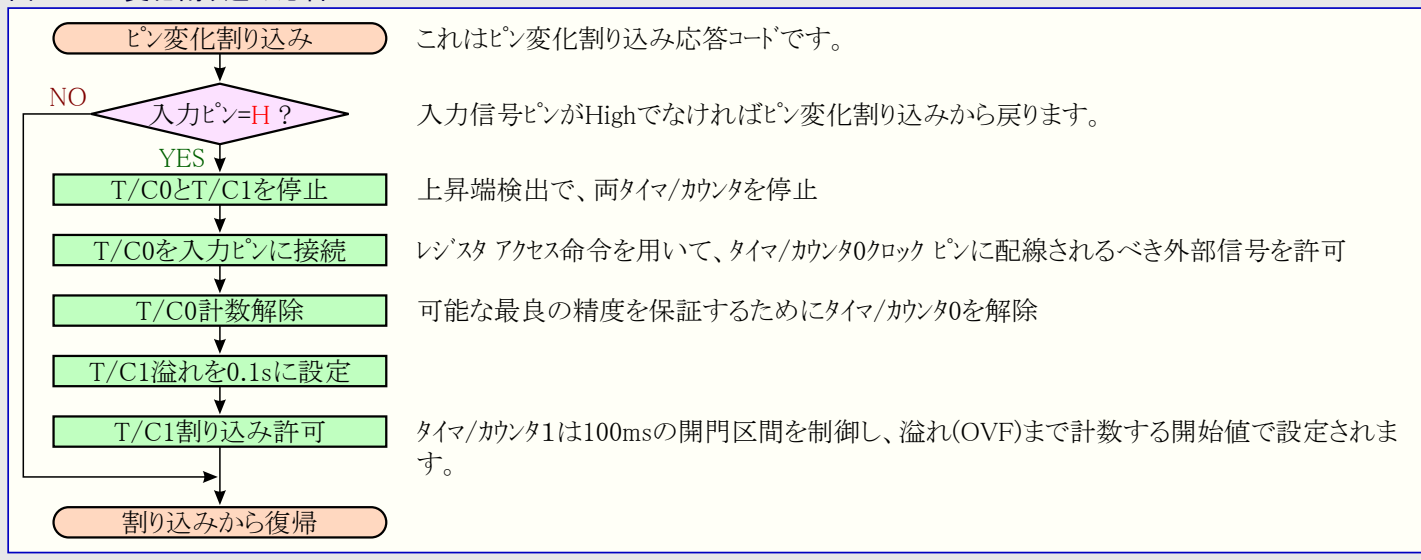
- ・ このプロジェクトを構築するには3つのファイルが必要です。この例でのそれらは以下です。
  - main()関数が置かれる、avr205\_frequency\_meter\_busy\_wait\_demo.c
  - 関数が置かれる、freq\_meter.c
  - この場合、8,000,000HzのCPUクロック速度を含む、ポートとレジスタの名前のようなものが指定される、freq\_meter.h
- ・ ATtiny861やATmega48などのような、どのAVRかを指定してください。

#### 4.2.3. 割り込みを用い、これらの関数を呼び出して使う方法

freq\_meter\_demo.cに於いて、

- ・ freq\_meter\_init()関数はAVRの関連レジスタを初期化します。
- ・ user\_init()関数は使用者変数を初期化します。
- ・ AVR割り込みは使用者によって許可されます。
- ・ freq\_cntr\_start\_mesurement()の呼び出しはT0のピン変化で始める周波数測定を許可するためにT0でのピン変化割り込みを構成設定します。
- ・ freq\_cntr\_get\_result()の呼び出しは周波数測定が未だ準備不可の場合に0x00、または結果としての周波数として0xFFFF未満の16進数のどちらかを返します。異常は0xFFFFの結果を経由して報告されます。
- ・ freq\_cntr\_clear\_result()の呼び出しは次の周波数測定を準備するために上で返された結果を解除します。

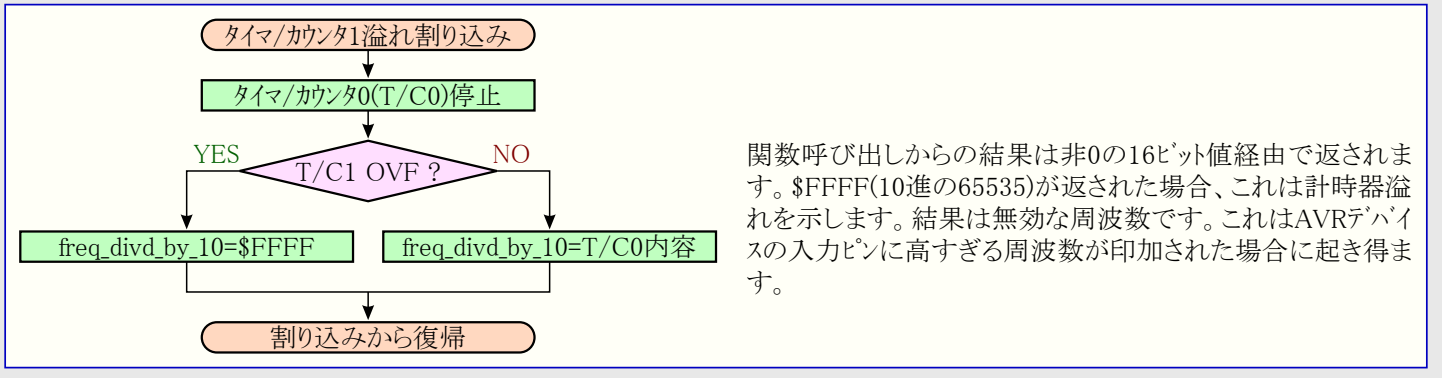
図4-5. ピン変化割り込み応答



これは図4-5内のコードに対して移行されて実行されることを許します。



図4-6. タイマ/カウンタ1溢れ割り込みコード



関数呼び出しからの結果は非0の16ビット値経由で返されます。\$FFFF(10進の65535)が返された場合、これは計時器溢れを示します。結果は無効な周波数です。これはAVRデバイスの入力ピンに高すぎる周波数が印加された場合に起き得ます。

## 5. (この応用記述で使われない)捕獲入力についての話

多くのAVRデバイスは8ビットのラッチの組を用いる“捕獲入力”を提供するタイマ/カウンタを含みます。クロック駆動時、これらのラッチはタイマ/カウンタの瞬間的な内容を保存します。捕獲入力はAtmelのATtiny13のような、より単純なAVRデバイスで利用できない高度な機能です。捕獲レジスタは入力ピンによってクロック駆動されます。しかし、周波数測定に関して、この動きは出力ピンによって起動されることが必要で、故にこの技法は2つの入出力ピンの使用が必要で、より少ないピン数のデバイスで度々利用できません。Atmel XMEGAの周波数計数器の実装は捕獲起動に事象システムを使うことができます。

## 6. 結び

殆どのAtmel tinyAVRとAtmel megaAVRのデバイスは外部クロックによってタイマ/カウンタをクロック駆動する能力を持ちます。この機能は2.5MHzの速さまたはより速い(AVRデバイスの主クロック周波数の機能)クロック駆動を内部計数器に許します。計数区間として100msの開門タイミングが使われる場合、計数器は入力信号周波数の1/10の値を持ちます。この技法は割り込み有りまたはなしで行うことができます。これら両方の技法がここで実演されます。

## 7. 目次

要点 .....	1
1. 序説 .....	1
2. コード走行の必要条件 .....	1
2.1. タイマ/カウンタ クロック用外部入力を持つAtmel AVRデバイスの一覧 .....	2
3. 測定原理 .....	2
4. 2つの技法: 多忙待ちと割り込み駆動 .....	2
4.1. 多忙待ち繰り返し .....	2
4.1.1. 多忙待ちを用いるプロジェクト外走行方法 .....	4
4.1.2. 多忙待ちを用いるプロジェクト外構築方法 .....	4
4.1.3. 多忙待ちを用い、これらの関数を呼び出して使う方法 .....	4
4.2. 割り込み駆動 .....	4
4.2.1. freq_meter_interrupt_demo走行方法 .....	5
4.2.2. 割り込みを用いるプロジェクト外構築方法 .....	5
4.2.3. 割り込みを用い、これらの関数を呼び出して使う方法 .....	5
5. (この応用記述で使われない)捕獲入力についての話 .....	6
6. 結び .....	6
7. 目次 .....	6



#### *Atmel Corporation*

2325 Orchard Parkway  
San Jose, CA 95131  
USA  
TEL (+1)(408) 441-0311  
FAX (+1)(408) 487-2600  
[www.atmel.com](http://www.atmel.com)

#### *Atmel Asia Limited*

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG  
TEL (+852) 2245-6100  
FAX (+852) 2722-1369

#### *Atmel Munich GmbH*

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY  
TEL (+49) 89-31970-0  
FAX (+49) 89-3194621

#### *Atmel Japan*

141-0032 東京都品川区  
大崎1-6-4  
新大崎勸業ビル 16F  
アトメル ジャパン合同会社  
TEL (+81)(3)-6417-0300  
FAX (+81)(3)-6417-0370

© 2011 Atmel Corporation. 不許複製 / 改訂CORP072610

Atmel<sup>®</sup>、ロコ<sup>®</sup>とそれらの組み合わせ、それとAVR<sup>®</sup>、AVR Studio<sup>®</sup>、XMEGA<sup>®</sup>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。

**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。

© HERO 2021.

本応用記述はAtmelのAVR205応用記述(doc8365.pdf Rev.8365A-01/11)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。