

AVR290 : Atmel tinyAVRでのSCLクロック伸長回避

要点

- Atmel tinyAVR®システムデバイス
- Atmel TWI従装置用コードドライバ
- PhilipsのI²C規約適合
- Atmel TWIハードウェア単位部を使用
- 割り込み駆動送信
- クロック伸長なし - 真の100kHz動作

1. 序説

Atmelの2線直列インターフェース(TWI)バスはPhilipsのI²C規約に適合します。このバスは電子装置内の集積回路間で簡単、強力且つ費用効率的な通信を許すように開発されました。

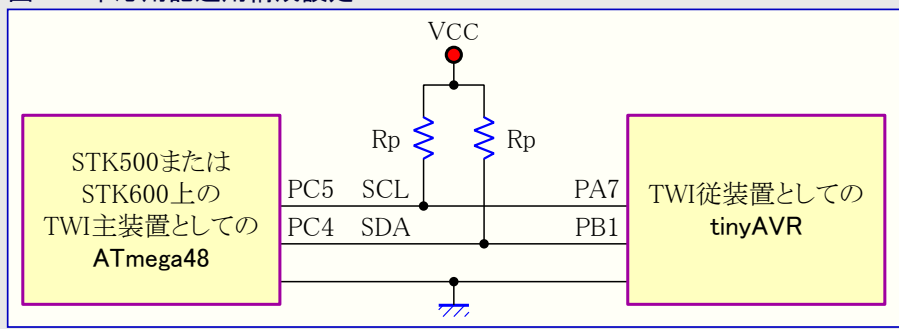
TWIバスの能力は同じTWIバスでの調停で同一バス上に複数の主装置を持つことができ、128装置までのアドレス指定能力を含みます。

AtmelのtinyAVR系統、特にAtmelのATtiny20とATtiny40デバイスはより速いACK応答を生成する改良されたTWI単位部を含み、それによって100kbpsほどの速さのデータ速度を許します。この単位部はI²CとSM-Busにも適合します。

この応用記述はATtiny20とATtiny40用のTWI従装置ドライバを記述します。従装置ソフトウェアドライバが含まれます。この応用記述はAVR®のクロック速度が直列クロック(SCL)を開放するのと直列データ(SDA)信号上で確認応答(ACK)するのに充分速ければ、TWIのSCLクロックは伸長されません。

実演目的のためにsoftwareフォルダ内にTWI主装置が含まれます。これはAtmel AVR315の変更版プログラムです。図1-1は構成設定の構成図を示します。

図1-1. 本応用記述用構成設定



2. 事前必要条件

この資料で検討されるAtmel TWIの従装置と主装置の実演は以下の技術と能力の基本的な熟知が必要です。より多くを学ぶには10章 参照を参照してください。

- ・ 組み込みシステムに関するCプログラミング言語
- ・ Atmel AVR用のIAR Embedded Workbench®でのCプロジェクトのコンパイル
- ・ <ftp://ftp.iar.se/WWWfiles/datasheet/New/DS-EWAVR-551.pdf> からのAtmel AVR用のIAR Embedded Workbench® 5.51またはそれ以降のIAR®始動版
- ・ 現時点でWinAVR GCC コンパイラは支援されていませんが、コードはWinAVRでのコンパイル用に変換することができます。
- ・ Atmel AVR Studio® 4.18またはより新版
- ・ Atmel ATtiny20用アダプタを持つ、Atmel STK®500またはSTK600
- ・ Atmel AVR JTAGICEmk II デバugg、コード変更が計画されている場合のデバugg用の任意選択
- ・ ATtiny20データシートで詳述されるTWIインターフェースの全般的な熟知



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. 8380A-03/11, 8380AJ3-04/21

3. 制限

- Atmel TWI従装置ドライバはそれらのTWI周辺機能だけでAtmelのATtiny20またはAtmelのATtiny40で動くことを目標とされます。他のデバイスの支援はTWI従装置ドライバのコードへの変更が必要かもしれません。
- この応用記述でのソフトウェア解決策はAtmel AVR用IAR Embedded Workbench 5.51またはそれ以降版で試験されています。それ以前の版のコンパイラはいくつかの変更が必要かもしれません。
- 応用ソフトウェアによってATtiny20 CPUが8MHzに設定される場合、100kbpsのTWI速度に制限されます。

4. 資源必要条件

表4-1. 代表的な必要条件

周辺機能	ピン	構成設定の可能性
Atmel TWI従装置	SCL,SDA	なし
ポートA	PA6~0	実演専用
内部EEPROM	本デバイスで利用不可	

表4-2. メモリ必要条件

メモリ	代表的な量	最大容量
プログラムメモリ	514バイト	2048バイト
データメモリ	73バイト	128バイト
内部EEPROM	本デバイスではなし	

注: 正確なメモリ必要条件はコンパイラ版、最適化レベル、構成設定可能な機能の追加や削除のような要素の変化に依存します。

5. TWI従装置ドライバ：クロック伸長回避法

5.1. 最小実行時間用に最適化されたコード

ここで記述される従装置ドライバはクロック伸長なしを達成するように最適化されています。これは従装置ドライバ内のAVR命令数を減らすことによって達成され、従って最小の実行時間が必要です。これは100kbps動作に対して充分速く応答(ACK)してSCLクロックの開放を従装置ドライバに許し、SCLクロック伸長を無くします。従装置内の以下の命令がSCLクロックを開放し、故にSCLは伸長されません。

```
TWI_SLAVE_CTRLB=(uint8_t)((1<<TWI_SLAVE_CMD1)|(1<<TWI_SLAVE_CMD0));
```

重要: SCL伸長を防ぐには割り込み処理ルーチン(ISR)入口の1ビット時間内に上の命令が実行されなければなりません。**例:** 主装置が100kbpsで動いている場合、100kHzに等しい1ビット時間は10μsです。

図5-1. ACKとSCL伸長なしオシロスコープ像

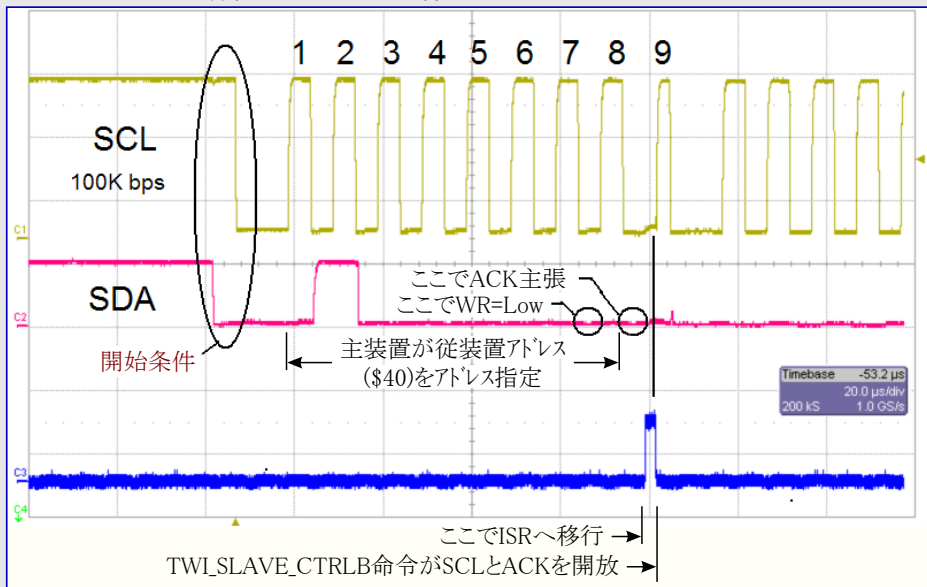


図5-1.は主装置が従装置をアドレス指定する流れを示します。

重要: 従装置アドレスが\$40=01000000であることを注意してください。けれども主装置と従装置内のコードでこのアドレスは\$20として参照されます。TWIソフトウェアはACK用の場所を許すためにこの\$20を1ビット左に移動します。

主装置への従装置の応答は次のとおりです。

- ISRへの移行はオシロスコープのチャンネル3(C3)で示される上昇端で示されます(これは説明目的だけにPB0を用いて行われます)。
- 従装置はSCLクロック信号をLowに保持するだけでなく、SDAデータ信号もLowに保持します。
- ここで10μsとして示された1ビット時間内に従装置のドライバソフトウェアはTWI_SLAVE_CTRLBレジスタに命令を実行します。これはSCLを開放し、それによってSCLクロックを伸長しません。

5.2. CPUクロック速度が決定的に重要

従装置ドライバが相対的に少ない命令を実行し、CPUクロック速度が充分速いので、この応用記述は100kbpsでだけクロック伸長なしを達成します。この例では8MHzが選択されます。AtmelのATtiny20とATtiny40は12MHzで動作することができますが、外部発振器元經由だけです。

表5-1. 各種CPUクロック速度に対する最大kbps

CPUクロック速度 (MHz)	最大SCL (kbps)	応用記述での最大時間 (注)
12	150	666 μ s
8	100	1ms
4	50	2ms
2	25	4ms
1	12.5	8ms

注: 100kbpsのデータ速度に於いて、応用のコードからの新しいバイトは1バイト当たり1msの速度で主装置と送信または受信することができます。主装置がこの速度よりも速いデータの送信または受信を要求する時に応用コードで多すぎる時間が費やされないように注意されなければなりません。

5.3. TWI主装置と従装置の接続方法

この実演に関して、主装置は8MHzの内部RC発振器で走行するAtmelのATmega48を持つAtmelのSTK500またはSTK600から成ります。それを8MHzでの走行を許すためにATmega48の8分周ヒューズ(CKDIV8)を非チェック(禁止)にしてください。

主装置と従装置のソフトウェアの詳細は以下で説明されます。主装置を従装置に接続する時に図1-1を参照してください。

従装置はSTK600で走行するATtiny20から成ります。

- 図1-1.で示されるようにSCLとSDAに関してジャンパ線を接続してください。
- Rpに関して2つの2k Ω 抵抗が推奨されます。
- 各基板のGNDを接続するのに3つ目のジャンパが必要です。
- 主装置基板に於いて、8つのスイッチ(SW0~SW7)を10芯ケーブルで、ポートBとLEDポート間を2つ目の10芯ケーブルで接続してください。
- 従装置基板に於いて、ポートBとLEDポート間を10芯ケーブルで接続してください。LEDはcount1変数の値を表示し、それは3秒毎に1つ増加します。

6. ソフトウェアの構築と走行の方法

6.1. Atmel TWI ATtiny20従装置ソフトウェア

- www.atmel.comウェブサイトからAtmel AVR290応用記述と連携するzip圧縮されたソフトウェア ファイルを置いてください。ZIPファイル フォルダのtwi_slave_tinyで以下のファイルを調べてください。
twi-no_stretch-tiny20-demo.c
twi-no_stretch-tiny20-drvr.c
twi-no_stretch-tiny20-drvr.h
- ファイルを解凍してください。iarフォルダに位置付けてください。
- Atmel AVR用IAR Embedded Workbench コンパイラのインストール後、IAR.eww作業空間を開いてプロジェクトをコンパイルしてください。
- .hexファイルが生成されます。AtmelのSTK600または他のAVR書き込み器を用いてこの.hexファイルをAtmelのATtiny20に書き込んでください。
- ATtiny20の電源をOFF/ONしてください。7つのLEDは2進形式で3秒毎に1つ増加して緩やかに流れるでしょう。

6.2. 関数の呼び出しと使用の方法

呼び戻し関数はAtmelのTWIドライバと応用コード間でデータを交換します。応用コードで定義されるこれらの関数はtwi_data_from_master()とtwi_data_to_master()です。

unsigned char twi_data_to_master (void)

AtmelのTWI従装置ドライバ呼び戻し関数twi_data_to_masterは命令特有応答論理の例です。これはTWIドライバが主応用コードからどうデータを得るかを実演します。

入力: なし

出力: この関数は直前に主装置から従装置に送られた命令の符号とその値に基づいたcount1または~count1内の値を返します。

unsigned char twi_data_from_master (unsigned char)

このドライバ呼び戻し関数、命令特有応答論理の例は主応用コードへデータを送ることを従装置ドライバに許します。

入力: 従装置ドライバからのデータ。特に主装置が従装置へ送ったデータ、そしてこのデータは関数への入力です。応用コードではこのデータが命令として参照されます。

出力: 応用コードで利用可能なデータ。

void twi_slave_initialise (void)

この関数は従装置の書き込みまたは読み込みをTWI主装置に許すようにAtmel AVR TWIハードウェア単位部を初期化します。変数は全く影響されません。

void twi_bus_error_check (void)

この関数は複数のTWI装置がTWIバスを同時使用を試みるための異常、即ち衝突を調べます。この関数は参照基準専用でここに含められ、今回はコードによって使われません。

6.3. 変更したTWI主装置ソフトウェアの構築と走行の方法

AtmelのAVR315で記述されるTWI主装置実装の変更版はこの実演に対してTWI主装置として働きますが、使用者選択の別の主装置を代わりに使うこともできます。この変更は\$20への従装置アドレス変更と、Atmel ATtiny20従装置へ特定の命令を送るためのSW0とSW1のスイッチの再定義を含みます。

1. Atmel AVR290ソフトウェア フォルダに位置付けて、6.1項で記述されるように前もって解凍してください。
2. twi_master_demo副フォルダに位置付けてください。
3. インストールしたAtmel AVR用IAR Embedded Workbench Cコンパイラを用いて、IARの.eww作業空間を開いてプロジェクトをコンパイルしてください。
4. .hexファイルが生成されるでしょう。AtmelのSTK500または他のAVR書き込み器を用いて、コンパイルされてリンクされた.hexファイルを、先に記述されたようにSDA,SCL,GNDと2つの2kΩプルアップ抵抗を用いて従装置に配線されたAtmelのATmega48に書き込んでください。
5. 書き込んだコードを走らせてください。一旦従装置プログラムが動くと、STK500上のSW0スイッチの押下と開放はATtiny20従装置をアクセスして主装置のLEDに表示されるcount1の値の要求をTWI主装置に始めさせます。
6. 同様に、STK500上のSW1スイッチの押下と開放はATtiny20従装置をアクセスして主装置のLEDに表示される反転した(2の補数)count1の値の要求をTWI主装置に始めさせます。
7. SCLクロック周波数は以下のように主装置で変更することができます。

$$\text{SCL周波数} = \frac{\text{CPUクロック周波数}}{16+2(\text{TWBR}) \times \text{前置分周値}}$$

この周波数は次のように設定されます。

- (1) TWI_master.hファイルに位置付けてください。
- (2) 例として、応用ソフトウェアを100kbpsに設定するにはTWI_TWBR=0x22に設定してください。

7. bpsデータ速度とTWI伸長なしの観測方法

1. 次のように、チャンネル1にSCL、チャンネル2にSDA、オシロスコープを実演に接続してください。
2. SCLクロック端間の周期を測定してください。
 - a. 100kbpsは100kHzまたはクロック端間の10μsと等価です。
3. 秒当たりのビット速度(bps)の変更方法の詳細に関しては6.3項をご覧ください。

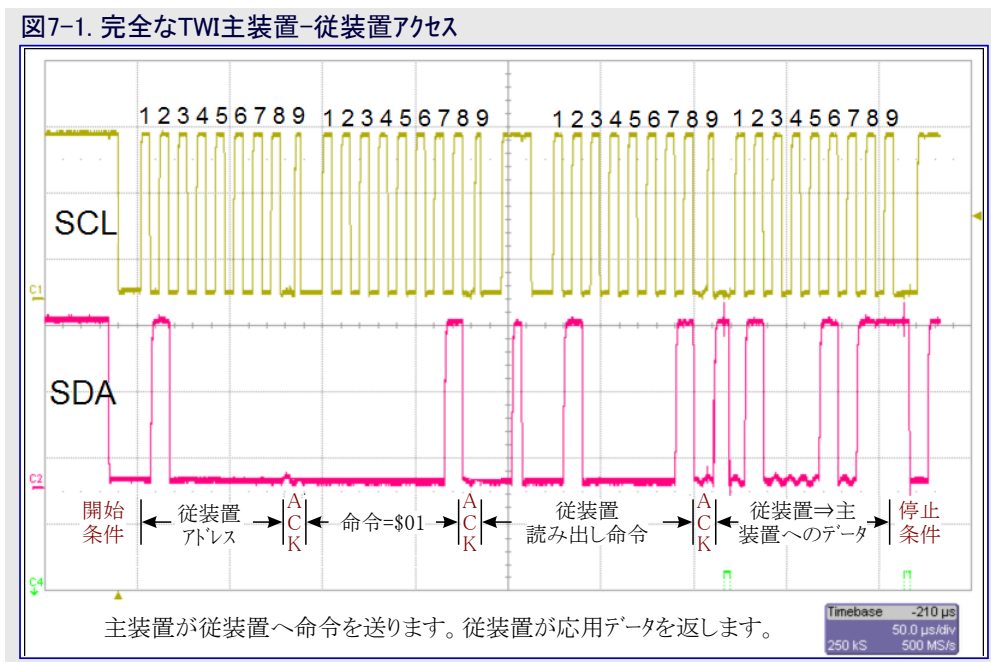


図7-1.は主装置から従装置への完全なデータ要求の実際のオシロスコープの波形です。データ速度は100kbpsです。以下を観測してください。

1. 開始指示後、TWI主装置は\$40(\$20左1ビット移動)の従装置アドレスを表明します。
2. これには伸長なしACKとSCLの第9クロック上昇端の開放が後続します。
3. 次に主装置は伸長なしACKとSCLの第9クロック上昇端の開放が後続する\$01(この実演については代替の\$02)の命令を送ります。
4. 従装置ソフトウェアはtwi_data_from_master()従装置呼び戻し関数経由で主装置命令に応答します。
5. 主装置はSCLクロックの第8上昇端でSDA信号をHighにすることによって示される“従装置読み込み”命令を発行します。これには伸長なしACKとSCLの第9クロック上昇端の開放が後続します。
6. 従装置は以下のように主装置の読み込み要求に応答します。
従装置ドライバソフトウェアはtwi_data_from_master()呼び戻し関数経由で応用をアクセスします。この関数は応用コード内に配置され、従装置ドライバへ応用変数を供給します。
7. 従装置TWI単位部はSDAバス上にデータを置きます。TWI主装置ソフトウェアはこのデータを受信し、使われるならばAtmelのSTK500上で主装置のポートBのLEDでそれを表示します。

8. TWIの理屈

本章はTWIインターフェースとAtmel tinyAVRを含むAtmel AVRデバイスのTWI単位部の短い記述を与えます。詳細な情報についてはAVRデータシートを参照してください。

8.1. 2線直列インターフェース

TWIは代表的なマイクロコントローラ応用に対して理想的に詭えられています。TWI規約は2つの双方向バス線、1つのクロック(SCL)と1つのデータ(SDA)だけを用い、最大128の個別にアドレス指定可能な装置の相互接続をシステム設計者に許します。

TWIバス線の各々に対して単一のプルアップ抵抗を実装するための外部ハードウェアだけが必要です。バスに接続された全てのTWI装置は個別のアドレスを持ちます。それらはバス衝突を解決する機構も持ちます。

TWIバスはバスの制御を取る能力を持つ1つまたはより多くの装置を接続することができる複数主装置バスです。主装置だけがSCLとSDAの両方の線を駆動することができ、一方従装置はSDA線にデータを送るだけが許されます。

データ転送は常に主装置によって始められます。SCLがHighの間のSDA線上のHighからLowへの遷移が**開始条件**または**再送開始条件**に定義されます。

開始条件は常に(個別の)7ビット従装置アドレスとその後の第8ビットのデータ方向ビットが後続します。アドレス指定された従装置は第9クロック周期間にSDAをLowに保持することによって主装置へ**確認応答(ACK)**を送ります。主装置がどの応答も受け取らない場合にその転送は終了されます。

データ方向ビットに依存して、主装置または従装置がSDA線上に8ビットのデータを送信します。そして、受信した装置はデータに応答します。主装置によって**再送開始条件**または**停止条件**が発行される前にその方向で複数バイトを転送することができます。

転送は主装置が**停止条件**を発行する時に終了されます。**停止条件**はSCLがHighの間のSDA線上でのLowからHighへの遷移によって定義されます。

TWIバスでの全てのデータ転送は8ビットのデータ(1データバイト)と応答(ACK)から成る9ビット長です。

データ転送中、主装置はクロック、(再送)開始条件と停止条件を生成します。TWI従装置は受信に応答する責任があります。**確認応答(ACK)**は従装置によって第9 SCLクロック周期間にSDA線をLowに引くことで生成されます。受信側がSDA線をHighのままにしたなら、**否定応答(NACK)**が送られます。

8.2. TWIクロック伸長

I²C規格に従って、バスに接続される全ての従装置はデータ処理中にクロック周波数全体を下げる、または待ち状態を挿入するためにクロックのLow区間を伸ばすことを許されます。クロック伸長が必要な従装置は線上のLowレベルを検出後、SCL線をLowへ強制的に保持することによってそれを行うことができます(これはこの線がオープントレインFETによってLow駆動され、そしてプルアップ抵抗を持つために可能です)。

8.3. Atmel tinyAVR従装置単位部 – 伸長なし動作

TWI主装置の応用がクロック伸長を支援しない場合、このAtmelのAVR290応用記述はAtmel ATtiny20またはAtmel ATtiny40での100kbps従装置動作を実演します。

TWI従装置はバイト志向で、各バイト後の割り込みを実演します。割り込み要求フラグはポーリング動作にも使うことができます。受信したACK/NACK、クロック保持、衝突、バス異常、読み/書き方向を示すための専用の状態フラグがあります。割り込み要求フラグが設定されると、SCL線がLowに強制されます。これは応答または何れかのデータを処理するための従装置時間を与え、殆どの場合、ソフトウェアの相互作用が必要でしょう。

9. 結び

AtmelのATtiny20とATtiny40は100kbpsのTWI従装置動作を支援します。100kbpsの速度は5.2項で記述されるように8MHzで走行するtinyAVRデバイスに基づく制限です。この応用記述で提供されるソフトウェア例は従装置動作でこのTWIインターフェースを処理する方法を説明します。この応用記述のコードは現時点に於いてIAR Cコンパイラでコンパイルし、GCCコンパイラでコンパイルしないでください。

9.1. Doxygen資料

全てのソースコードはDoxygenを用いる自動資料生成のための準備が整えられています。Doxygenは特別なキーワードを用いてソースコードを分析することによってソースコードから自動的に資料を生成するためのツールです。Doxygenについてのより多くの詳細については<http://www.doxygen.org>を訪ねてください。予めコンパイルされたDoxygen資料もこの応用記述と連携するソースコードと共に供給され、ソースコードフォルダ内の[readme.html](#)ファイルから利用可能です。

10. 参照

- Atmel AVR Studio
- Atmel AVR1320:TWI従装置の真の400kHz動作
- Atmel AVR315:I²C主装置としてのTWI部の使い方
- Atmel AVR JTAGICEmk II
- Atmel AVR用Embedded Workbench Cコンパイラ 5.51またはそれ以降版
<ftp://ftp.iar.se/WWWfiles/datasheet/New/DS-EWAVR-551.pdf>

11. 目次

要点	1
1. 序説	1
2. 事前必要条件	1
3. 制限	2
4. 資源必要条件	2
5. TWI従装置ドライバ: クロック伸長回避法	2
5.1. 最小実行時間用に最適化されたコード	2
5.2. CPUクロック速度が決定的に重要	3
5.3. TWI主装置と従装置の接続方法	3
6. ソフトウェアの構築と走行の方法	3
6.1. Atmel TWI ATtiny20従装置ソフトウェア	3
6.2. 関数の呼び出しと使用の方法	3
6.3. 変更したTWI主装置ソフトウェアの構築と走行の方法	4
7. bpsデータ速度とTWI伸長なしの観測方法	4
8. TWIの理屈	5
8.1. 2線直列インターフェース	5
8.2. TWIクロック伸長	5
8.3. Atmel tinyAVR従装置単位部 - 伸長なし動作	5
9. 結び	6
9.1. Doxygen資料	6
10. 参照	6
11. 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

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

Atmel[®]、ロコ[®]とそれらの組み合わせ、それとAVR[®]、AVR Studio[®]、tinyAVR[®]、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。

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AVR290応用記述(doc8380.pdf Rev.8380A-03/11)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。