

AVR204 : BCD演算

要点

- 16ビットと5桁間、8ビットと2桁間の変換
- 2桁の加算と減算
- 素晴らしい速度とコード密度
- 走行可能な例プログラム

(訳注) 本書は原書に対してプログラム例が追加されています。またプログラム内容も変更されています。これに伴ってソース内のシンボル名等も原書と異なります。

序説

この応用記述はBCD演算用ルーチンを一覧にします。核となる性能特性とで全実装の一覧が表1.で与えられます。

表1. 性能値要約

応用	コード量(語)	実行時間(周期数)
16ビット2進数⇒5桁詰込(パック)BCD変換	25	719~735 (注)
8ビット2進数⇒2桁BCD変換 (コード量最適化)	6	5~50
8ビット2進数⇒2桁詰込(パック)BCD変換 (コード量最適化)	7	6~51
8ビット2進数⇒2桁BCD変換 (速度最適化)	17	13~16
8ビット2進数⇒2桁詰込(パック)BCD変換 (速度最適化)	19	15~18
5桁詰込(パック)BCD⇒16ビット2進数変換	25	84
2桁BCD⇒8ビット2進数変換 (コード量最適化)	4	3~48
2桁詰込(パック)BCD⇒8ビット2進数変換 (コード量最適化)	5	4~49
2桁BCD⇒8ビット2進数変換 (速度最適化)	6	10
2桁詰込(パック)BCD⇒8ビット2進数変換 (速度最適化)	10	10
2桁詰込(パック)BCD加算	11	9~11
2桁詰込(パック)BCD減算	7	7

注: Zレジスタの事前減数付き間接アドレス指定が利用できないデバイスは44クロック追加されます。

8-bit 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. 0938A-01/03, 0938AJ4-03/21

B2D16: 16ビット2進数⇒5桁詰込(ハック化)BCD変換

このサブルーチンは16ビット2進値を5桁詰込(ハック化)BCDに変換します。この実装はコード量最適化です。この実装はZポインタの自動減数機能(事前減少付き間接アドレス指定)を使いますが、AT90Sxx0x系統に関してはこのように使うことができません。AT90Sxx0xで本ルーチンを使うには、ファイルのリストで示される所で“DEC ZL”を追加してください。

算法内容

“B2D16”は以下の算法を実行します。

1. 結果(BCD値)3バイトを空にします。
2. 桁(ビット)計数値に16を設定します。
3. 手順9.から(実際の処理を)開始します。
4. BCD第5桁(第3バイト下位)が次回10(今回5)以上なら、3(次回6)を加算します。
5. BCD第4桁(第2バイト上位)が次回10(今回5)以上なら、3(次回6)を加算します。
6. BCD第3桁(第2バイト下位)が次回10(今回5)以上なら、3(次回6)を加算します。
7. BCD第2桁(最下位バイト上位)が次回10(今回5)以上なら、3(次回6)を加算します。
8. BCD最下位桁(最下位バイト下位)が次回10(今回5)以上なら、3(次回6)を加算します。
9. 2進値を左に1桁(ビット)移動し、対応桁をキャリーフラグに設定します。
10. BCD値を左に1ビット移動し、最下位ビットにキャリーフラグを設定します。
11. 桁(ビット)計数値を減数(-1)します。
12. 桁(ビット)計数値が0でなければ手順4.からを繰り返します。

本実装では手順4.~8.が繰り返し実行され、そこで結果の全3バイトの連続的なアクセスにZポインタが使用されます。これは図1.で示される流れ図で示されます。

使用法

1. 変換すべき16ビット数値を16ビットレジスタ変数“B2D16”に格納してください (上位バイトがB2D16H)。
2. “B2D16”を呼び出してください。
3. 結果は“BCD2”の下位ニブルに最上位桁(MSD)となる3バイトレジスタ変数“BCD2:BCD1:BCD0”で得られます。

性能

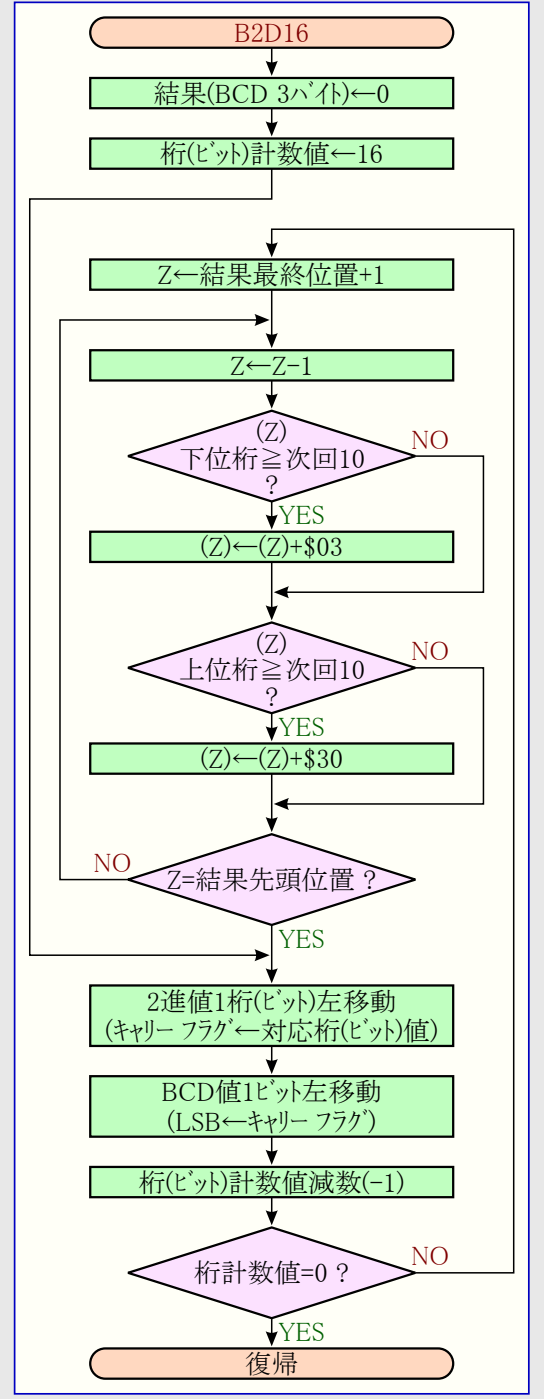
表2. “B2D16”使用レジスタ

レジスタ	入力	内部	出力
R13			BCD0:BCD最下位バイト
R14			BCD1:BCD第2バイト
R15			BCD2:BCD最上位バイト
R16	B2D16L:2進下位バイト		
R17	B2D16H:2進上位バイト		
R18		CNT:桁(ビット)計数値	
R19		TMP:一時変数	
R30		ZL:BCDポインタ下位	
R31		ZH:BCDポインタ上位	

表3. “B2D16”性能値

項目	値
コード容量(語)	25+RET
実行時間	719~735+RET (注)
使用レジスタ	下位レジスタ=3, 上位レジスタ=4, ポインタ=Z
使用割り込み	なし
使用周辺機能	なし

図1. “B2D16”流れ図



B2D8: 8ビット2進数⇒2桁(非パック化)BCD変換

このサブルーチンは8ビット2進値を2桁(非パック化)BCDに変換します。2桁のBCD数値が0～99だけを表現できるため、2進値は99を越えてはなりません。この実装は詰込(パック化)の結果を生成せず、換言すると2つの桁は2つの独立したバイトで表現されます。これを成し遂げるには以降の項で示されるように、いくつかの小さな修正が算法に対して行われなければなりません。

(訳注) 本書ではコード量最適化版と速度最適化版の2つがあります。

算法内容

“B2D16”は以下の算法を実行します。

1. 結果(BCD値)上位桁バイトを空にします。
2. 8ビット2進値から10を減算します。
3. 結果が負なら、8ビット2進値に10を加え戻して復帰します。
4. 結果(BCD値)上位桁バイトを進行(+1)し、2.からを繰り返します

結果(BCD値)下位桁は入力値と同じレジスタで得られます。詰込(パック化)された結果が必要とされるなら、算法に以下の変更を行ってください。

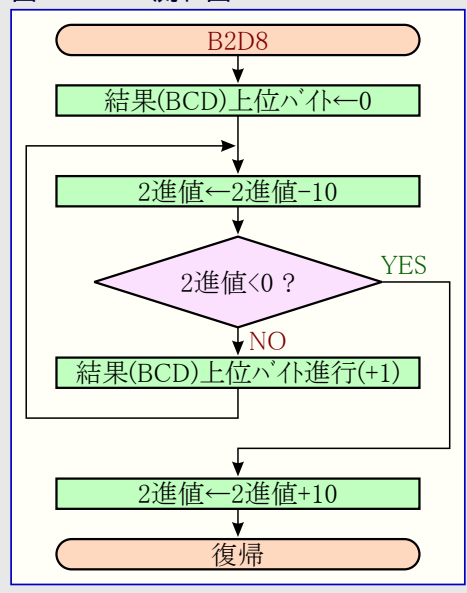
- ・手順4.でのBCD上位桁進行(+1)を上位桁へ\$10加算に代えてください。
 - ・手順3.での10加え戻し後に上位桁と下位桁を加算(または論理和)してください。
- どこでこれらの変更をするのかはプログラムリストで指示されます。

使用法

1. 変換すべき値をレジスタ変数“BIN”に格納してください。
2. “B2D8”を呼び出してください。
3. 上位桁と下位桁の結果は各々“BCDH”と“BCDL”で得られます。

(訳注) 使用法はコード量最適化版と速度最適化版の両方で同じです。詰込(パック化)BCDの場合は結果が“BCDL”で得られます。

図2. “B2D8”流れ図



性能

表4. “B2D8,B2D8S”使用レジスタ(両版共通)

レジスタ	入力	内部	出力
R16	BIN:2進数値		BCDL:BCD下位バイト/詰込(パック化)BCD
R17			BCDH:BCD上位バイト

表5. “B2D8”性能値(コード量最適化実装)

項目	値
コード容量(語)	6(+1)+RET
実行時間	5～50(+1)+RET
使用レジスタ	下位レジスタ=0, 上位レジスタ=2, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

表6. “B2D8S”性能値(速度最適化実装)

項目	値
コード容量(語)	17(+2)+RET
実行時間	13～16(+2)+RET
使用レジスタ	下位レジスタ=0, 上位レジスタ=2, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

(訳注) 表5.と表6.内の(+n)表記は詰込(パック化)BCD時に追加される語または周期数を示します。

D2B16: 5桁詰込(パック化)BCD⇒16ビット2進数変換

このサブルーチンは5桁詰込(パック化)BCD数値を16ビット2進値に変換します。

算法内容

a, b, c, d, e がBCD数値内の5桁を示すとしましょう(a =最上位桁、 e =最下位桁)。結果は次式を計算することによって生成されます。

$$10 \times (10 \times (10 \times (10 \times a + b) + c) + d) + e$$

4回の“10倍して加算”繰り返し操作がサブルーチンとして実装されます。このサブルーチンは入力パラメータとして16ビットレジスタ変数“BINH:BINL”とレジスタ変数“FIGD”を必要とします。このサブルーチンは2つの独立したアドレス“M10AH”と“M10AL”によって呼び出されます。違いは次のように要約されます。

- ・“M10AH”：“BINH:BINL”と“FIGD”の上位ニブルを乗算
- ・“M10AL”：“BINH:BINL”と“FIGD”の下位ニブルを乗算

このサブルーチンは以下の算法を実装します。

1. 加算値“FIGD”の上下桁(ニブル)を入れ替えます。(入口“M10AH”)
2. 加算値“FIGD”の上位ニブルを空にします。(入口“M10AL”)
3. 現在の2進値“BINH:BINL”の2倍値を得ます。
4. この2倍値を一時保存します。
5. 現在の2進値“BINH:BINL”の8倍値を得ます。
6. 現在の2進値“BINH:BINL”の10倍値(2値倍+8倍値)を得ます。
7. 加算値“FIGD”を2進値“BINH:BINL”の10倍値に加算します。

“M10AL”呼び出し時、手順1.が省略されます。

主ルーチンは、この算法に従います。

1. 結果(2進値)上位バイト“BINH”を空にします。
2. BCD第3バイト“BCD2”を2進値下位バイト“BINL”に複写します。
3. 加算値“FIGD”にBCD第2バイト“BCD1”(上位ニブル)を得ます。
4. BCD第5桁(第3バイト下位)×10+BCD第4桁(第2バイト上位)を行います(M10AH)。
5. 加算値“FIGD”にBCD第2バイト“BCD1”(下位ニブル)を得ます。
6. 現2進値×10+BCD第3桁(第2バイト下位)を行います(M10AL)。
7. 加算値“FIGD”にBCD第1バイト“BCD0”(上位ニブル)を得ます。
8. 現2進値×10+BCD第2桁(最下位バイト上位)を行います(M10AH)。
9. 加算値“FIGD”にBCD第1バイト“BCD0”(下位ニブル)を得ます。
10. 現2進値×10+BCD最下位桁(最下位バイト下位)を行います(M10AL)。

使用法

1. 変換すべき値(“BCD2”の下位ニブルが最上位桁)を3バイトレジスタ変数“BCD2:BCD1:BCD0”に格納してください。
2. “D2B16”を呼び出してください。
3. 16ビットの結果が“BINH:BINL”で得られます。

性能

表7. “D2B16”使用レジスタ

レジスタ	入力	内部	出力
R11		ZERO:0定数	
R12		TMPL:一時変数下位	
R13		TMPH:一時変数上位	
R14			BNYL:2進下位バイト
R15			BNYH:2進上位バイト
R16	BCD0:詰込(パック化)BCD最下位バイト		
R17	BCD1:詰込BCD第2バイト		
R18	BCD2:詰込BCD最上位バイト		
R19		FIGD:加算値	

図3. “D2B16”流れ図

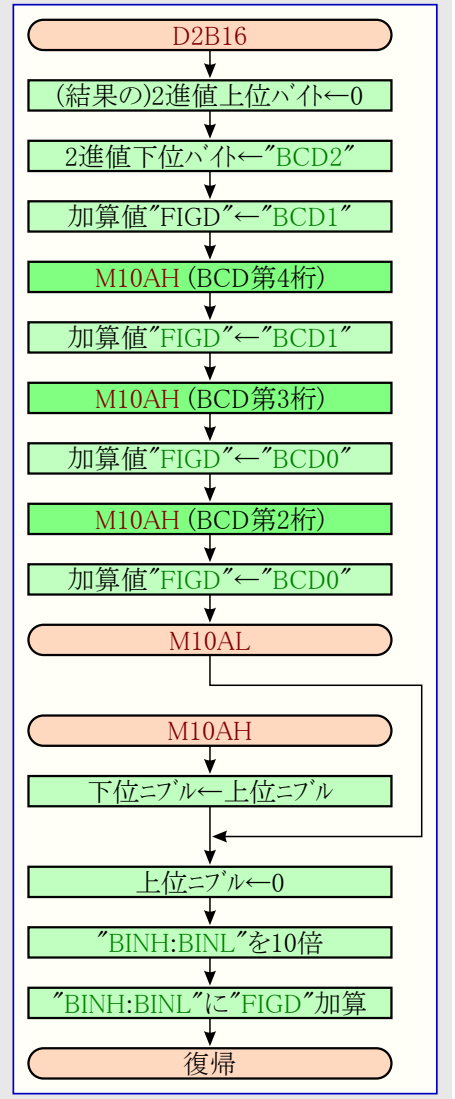


表8. “D2B16”性能値

項目	値
コード容量(語)	25+RET
実行時間	84+RET
使用レジスタ	下位レジスタ=5, 上位レジスタ=4, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

D2B8: 2桁(非パック化)BCD⇒8ビット2進数変換

このサブルーチンは2桁(非パック化)BCD数を8ビット2進値に変換します。この実装は詰込(パック化)BCD入力を受け容れず、換言すると2桁は2つの独立したバイトで表現されなければなりません。これを成し遂げるには以降の項で示されるように、いくつかの小さな修正が算法に対して行われなければなりません。

(訳注) 本書ではコード量最適化版と速度最適化版の2つがあります。

算法内容

“B2D16”は以下の算法を実行します。

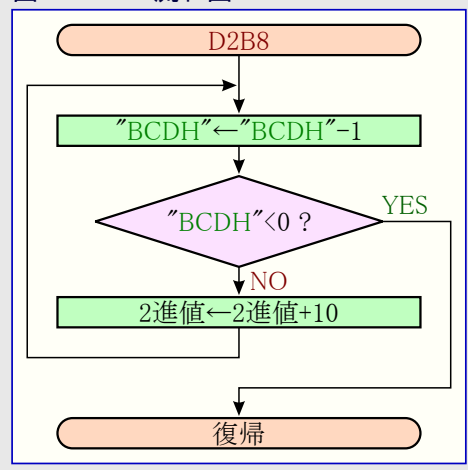
1. BCD上位桁バイトから1を減算します。
2. 結果が負なら、復帰します。
3. 8ビット2進値に10を加算します。
4. 手順1.からを繰り返します。

結果(2進値)はBCD下位桁バイト入力値と同じレジスタで得られます。詰込(パック化)BCD入力を受け容れる算法にするには、この算法を用いてください。

1. 詰込(パック化)BCD値を結果(2進値)“BIN”に複写します。
2. 結果(2進値)“BIN”の上位ニブルを空にします(=\$0x)。
3. 詰込(パック化)BCD値から\$10(上位桁に対する1)を減算します。
4. 結果が負なら、復帰します。
5. 8ビット2進値に10を加算します。
6. 手順3.からを繰り返します

プログラムリストはどこをどう変更するのかを示します。図4.は非詰込(非パック化)入力実装に適用する流れ図を示します。

図4. “D2B8”流れ図



使用法

1. 非詰込(非パック化)BCD値の上位桁バイトと下位桁バイトを各々レジスタ変数“BCDH”と“BCDL”に格納してください。
2. “D2B8”を呼び出してください。
3. 8ビットの結果が“BIN”で得られます。

(訳注) 使用法はコード量最適化版と速度最適化版の両方で同じです。詰込(パック化)BCDの場合は“BCDL”に格納してください。

性能

表9. “D2B8,D2B8S”使用レジスタ(両版共通)

レジスタ	入力	内部	出力
R16	BCDL:BCD下位バイト/詰込(パック化)BCD		BIN:2進数値
R17	BCDH:BCD上位バイト		

表10. “D2B8”性能値(コード量最適化実装)

項目	値
コード容量(語)	4(+1)+RET
実行時間	3~48(+1)+RET
使用レジスタ	下位レジスタ=0, 上位レジスタ=2, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

表11. “D2B8S”性能値(速度最適化実装)

項目	値
コード容量(語)	6(+4)+RET
実行時間	6(+4)+RET
使用レジスタ	下位レジスタ=0, 上位レジスタ=2, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

(訳注) 表10.と表11.内の(+n)表記は詰込(パック化)BCD時に追加される語または周期数を示します。

BCDADD: 2桁詰込(パック化)BCD加算

このサブルーチンは2つの2桁詰込(パック化)BCD数を加算します。出力は2つの入力数値、または2桁詰込(パック化)BCDとしての合計と溢れキャリー状態です。

算法内容

“BCDADD”は以下の算法を実行します。

1. 2進として値を加算します。
2. 2進加算のキャリー フラグを保存します(=BCDキャリー フラグ)。
3. ハーフ キャリーありなら、下位桁を0～9範囲に補正します。
4. ハーフキャリーなし且つ下位桁10以上なら、下位桁を0～9範囲に補正し、上位桁を+1します。
5. 上位桁10以上なら、上位桁を0～9範囲に補正し、BCDキャリー フラグを設定(1)します。

使用法

1. 加算すべき値をレジスタ変数“BCDX”と“BCDY”に格納してください。
2. “BCDADD”を呼び出してください。
3. BCDの総和は“BCDX”、“FLG(BCDY)”に溢れキャリー状態を得られます。

性能

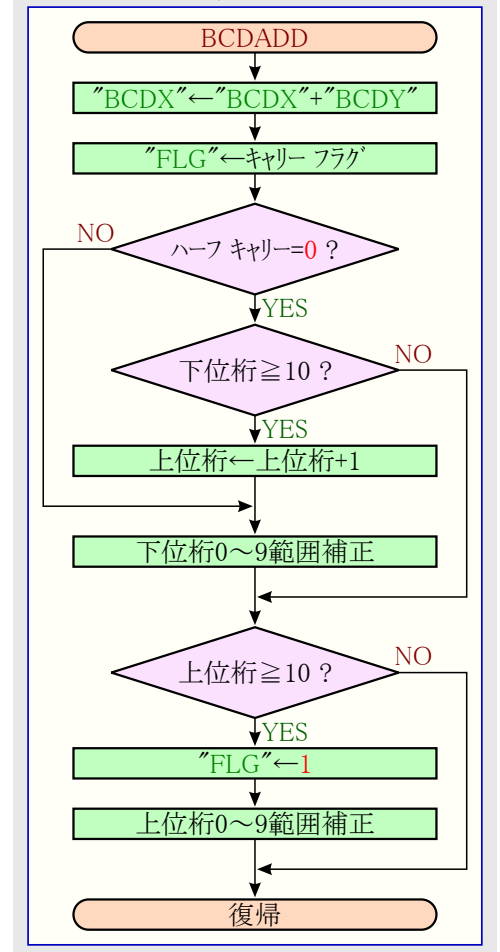
表12. “BCDADD”使用レジスタ

レジスタ	入力	内部	出力
R16	BCDX:詰込(パック化)BCD		BCDX:結果詰込(パック化)BCD
R17	BCDY:詰込(パック化)BCD		FLG:溢れキャリー フラグ

表13. “BCDADD”性能値

項目	値
コード容量(語)	11+RET
実行時間	9～11+RET
使用レジスタ	下位レジスタ=0, 上位レジスタ=2, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

図5. “BCDADD”流れ図



BCDSUB: 2桁詰込(パック化)BCD減算

このサブルーチンは2つの2桁詰込(パック化)BCD数を減算します。出力は2つの入力数値、または2桁詰込(パック化)BCDとしての差と漏れフラグ状態です。

算法内容

"BCDSUB"は以下の算法を実行します。

1. 2進として値を減算します。
2. 2進加算のフラグ(キャリー)フラグを保存します(=BCDフラグ)。
3. 下位桁フラグ(ハーフキャリー)ありなら、下位桁を0~9範囲に補正します。
4. 上位桁フラグ(2のキャリー)ありで、上位桁を0~9範囲に補正します。

使用法

1. 引かれるべき値をレジスタ変数"BCDX"、引く値を"BCDY"に格納してください。
2. "BCDSUB"を呼び出してください。
3. BCDの差は"BCDX"、"FLG(BCDY)"に漏れフラグ状態を得られます。

性能

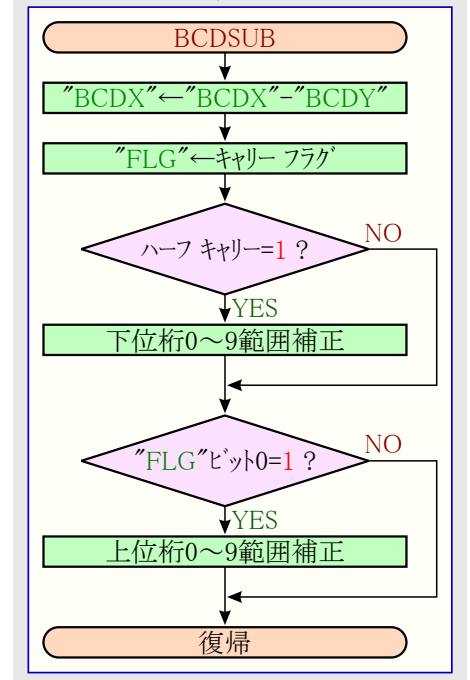
表14. "BCDSUB"使用レジスタ

レジスタ	入力	内部	出力
R16	BCDX: 詰込(パック化)BCD		BCDX: 結果詰込(パック化)BCD
R17	BCDY: 詰込(パック化)BCD		FLG: 漏れキャリーフラグ

表15. "BCDSUB"性能値

項目	値
コード容量(語)	7+RET
実行時間	7+RET
使用レジスタ	下位レジスタ=0, 上位レジスタ=2, ポインタ=0
使用割り込み	なし
使用周辺機能	なし

図6. "BCDSUB"流れ図





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
東熱新川ビル 9F
アトメル ジャパン株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製造拠点

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求

www.atmel.com/literature

© Atmel Corporation 2003.

Atmel製品は、ウェブサイト上にあるAtmelの定義、条件による標準保証で明示された内容以外の保証はありません。本製品は改良のため予告なく変更される場合があります。いかなる場合も、特許や知的技術のライセンスを与えるものではありません。Atmel製品は、生命維持装置の重要部品などのような使用を認めておりません。

本書中の®、™はAtmelの登録商標、商標です。
本書中の製品名などは、一般的に商標です。

© HERO 2021.

本応用記述はAtmelのAVR204応用記述(doc0938.pdf Rev.0938A-01/03)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。