

AVR063 : STK504[®]用LCDドライバ

要点

- 英数文字(アルファベット/数字)用ソフトウェアドライバ
- 液晶表示器(LCD)濃淡制御
- 割り込み制御更新
- LCDセグメント制御符号(SCC)へのASCII変換
- STK504 LCD表示器のインターフェース

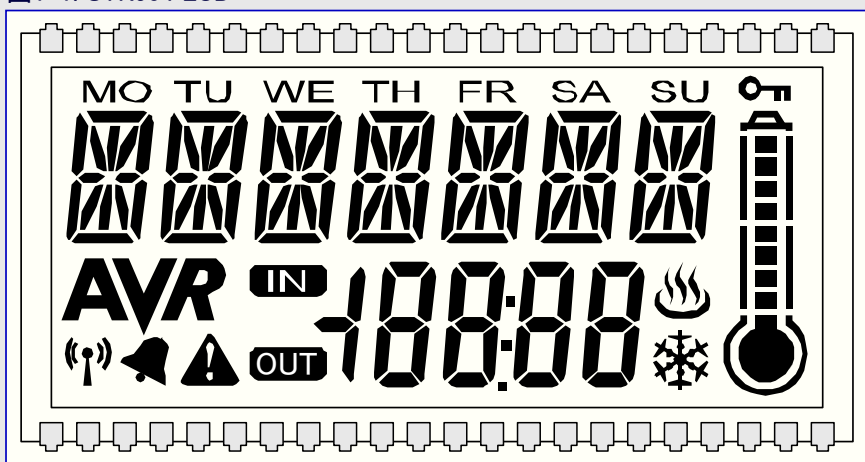
1. 序説

使用者との相互の影響を必要とする応用では、使用者へ情報を表示し得ることが度々有用です。簡単なインターフェースは情報LEDでできますが、もっと複雑な相互作用は文字、数字、語や文章さえも表示する能力のある表示器から得られます。液晶表示器(LCD)はメッセージ表示に度々使われます。LCD規格化部品は図形と文字の表示に使うことができるグラフィック型、または10～80桁を表示する能力の文字型のどちらかであります。標準的な文字型LCD規格化部品はインターフェースが容易ですが、LCD硝子での文字/図形の生成を処理する組み込み制御部/駆動部のため、明らかに高価です。

LCD硝子は液晶が含まれている面硝子です。表示器を必要とする応用の費用を低減するために、LCD硝子を直接駆動する組み込みLCD駆動部を持つMCUの使用を選択することができます。これはLCD規格化部品に統合された駆動部の必要を無くし、従ってLCD硝子がLCD規格化部品よりもっと安価なので、最大10倍程、費用を低減します。

AtmelのATmega3290フラッシュ マイクロ コントローラは最大160セグメントを制御する能力の統合されたLCD駆動部を持ちます。このデバイスの高効率コアと非常に低い電力消費は電池応用に対してこれを理想的にします。STK504は100ピンのAVRデバイス支援を追加する、STK500用ハードウェア拡張基板です。

図1-1. STK504 LCD



2. 動作の理屈

本章は一般的な特性の基本的な概要とLCD硝子関連で使われる用語に対する紹介を提供します。

加えてATmega3290表示器駆動部の要約と応用例で使うLCD硝子の記述内容が検討されます。更に、本章はATmega3290 LCD機能の要約と応用例で使うLCD硝子の説明を含みます。



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. 8009A-04/06, 8009AJ3-01/21

2.1. LCD硝子説明

液晶表示器は液体のように流れ、そして光を曲げる棒状分子(液晶)を使う表示技術に基きます。エネルギーが与えられないと、液晶は2つの偏光濾過部に真っ直ぐな光を通し、本来の背景色が見えることを許します。エネルギーが与えられると、偏光部の1つに於いて消されるように光の向きを変え、暗部出現を引き起こします。分子をもっと曲げることがより良い濃淡と視野角になります(注)。

LCDは交流(AC)によって駆動されなければなりません。DC(直流)は液晶に於いて電気泳動効果を引き起こし、表示器を劣化させるでしょう。静的(スタティック)と動的(多重化,ダイナミック)の2つのAC駆動方法があります。

静的駆動法ではLCDが2つの方形波で駆動されます。静的駆動法は良好な表示品質を得ることができることで、最も基本的な方法です。けれども、セグメント毎に1つの液晶駆動回路を必要とするため、多数のセグメントを持つ液晶表示器に適しません。

動的駆動法では駆動部の製造に依存して、**駆動波形とバイスレベル(以下で説明)**の広範囲の変形があります。多重化レベルは(共通線または共通電極とも呼ばれる)背面数に依存します。例えば、3背面表示器(1/3多重化)は3つの背面(共通電極)を持ちます。

LCDの最も一般的な2つの形式はTN(Twisted Nematic)とSTN(Super Twisted Nematic)です。TNは16背面よりも少ないLCD硝子に使われ、一方STN LCDは240背面程の数を持つ能力があります。背面数の制限は背面数増加のための透過性減少によって引き起こされます。

注: LCDセグメント上のより高い電圧のためにLCDの液晶はもっと偏光される時にもっと曲げられます。

2.2. LCDフレーム速度

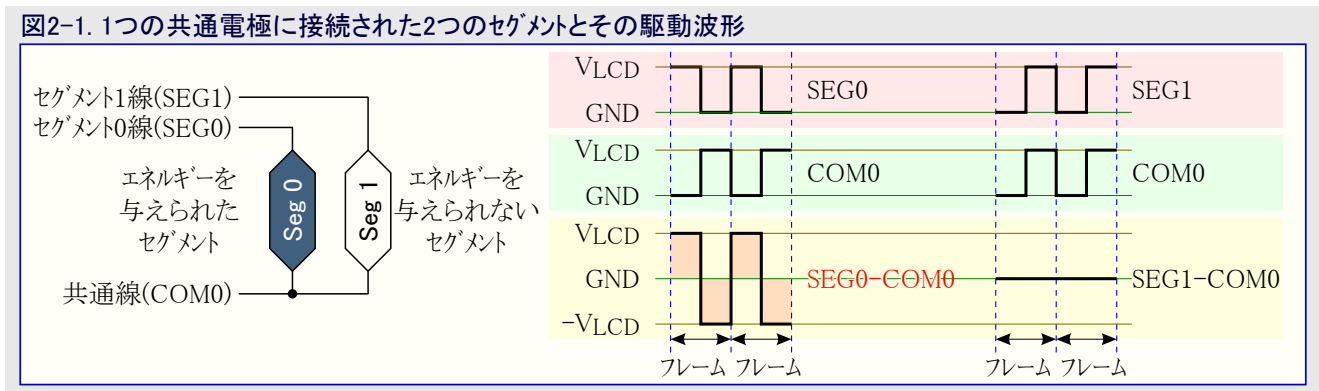
LCDセグメントが秒当たりでエネルギーを与えられる回数がLCDフレーム速度と呼ばれます。フレーム速度は人間の目がちらつきとしてセグメントを認知するのを避けるため、30Hz以上を保つべきです。

高いフレーム速度が使われる場合、残像が起き得ます。残像はLCDセグメントが正しくOFFされない時です。残像は**デューティとバイス(以下で説明)**にも依存し、実際に使われるデューティとバイスに対してフレーム速度を適合させる必要があるかもしれません。通常残像は十分に低いフレーム速度を使うことによって避けることができます。100Hzまたはそれ以下のフレーム速度が残像を防ぎます。

2.3. セグメント駆動部と共通電極

各LCDセグメントは2つの電極を持ちます。一方はセグメント駆動部に接続され、他方は共通電極に接続されます。セグメント駆動部と共通電極を横切る交流を印加することにより、液晶は偏光され(エネルギーを与えられ)見えるようになります。共通電極はLCDセグメントの1群に対する共通を記述する用語としての記述です。

多数のセグメント間で共用される1つの(共通)電極を通して1つのセグメントだけを活性化するため、セグメントが個別に活性化されるような方法で駆動波形が符号化されます。1つの共通電極だけが使われるなら、それは各セグメント駆動部が1つのセグメントだけを駆動する場合で、エネルギーを与えられるべきでないセグメントのセグメント駆動部はエネルギーを与えられるセグメントのセグメント駆動部の逆相です。これはエネルギーを与えられるべきセグメント上に最大電位差があり、一方エネルギーを与えられるべきでないセグメント上に小さな電位差があるか、または全くないことに帰着します。**図2-1**はLCD駆動部が各々1つのセグメントだけを駆動する(スタティック デューティ)場合のセグメントに関して、エネルギーを与えられる/られないセグメントとそれらの駆動波形を示します。



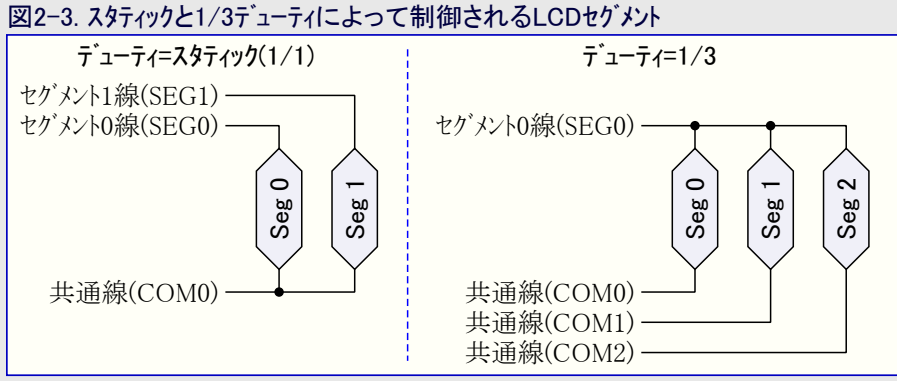
(訳注) 原書の**図2-1**と**図2-2**は**図2-1**として纏めました。

上図の波形の左側は活性なセグメントの駆動波形を、右側は不活性なセグメントの駆動波形を示します。

各LCD駆動部が複数のセグメントを駆動する場合のセグメントにエネルギーを与える方法はもっと複雑です。これは多数の背面が存在する時に起きます。この場合は駆動波形の所謂デューティサイクルが1/2またはそれ以下です。デューティサイクルは以下でもっと詳細に記述されます。

2.4. デューティ サイクルまたはデューティ比

デューティ サイクルまたはデューティ比は各フレーム中に各セグメントがどれ位長く活性にされるかを記述するのに使われる数値です。各セグメント駆動部が1つのセグメントだけを駆動する時のデューティ比は“スタティック”です。駆動部が複数のセグメントを駆動する場合、各デューティ比は1/(各LCD駆動部によって駆動されるセグメント数)として与えられます。各LCD駆動部によって駆動されるセグメント数は共通電極数に等しくなります。従ってデューティ比は与えられたLCD硝子に於ける共通電極数に依存します。図2-3は背面数とLCDを制御する時に使われるデューティ比間の関連を図解します。



駆動バイアス(または単にバイアス)はLCDを駆動する時に使われる電圧レベル数に関連します。バイアスは1/(電圧レベル数-1)として定義されます。各駆動部によって駆動される多数のセグメントはより多くの電圧レベル数が必要とされます(注)。デューティ比の定義により、使われるデューティ比と必要とするバイアス間には直接的な関連があります。

注: 単一セグメント線によって駆動されるセグメント数はLCD硝子の背面数に依存します。

表2-1. 表示器のデューティ、バイアスと電圧レベル数

共通電極数	1	2	3	4	6	7	11	12
デューティ	スタティック(1/1)	1/2	1/3	1/4	1/6	1/7	1/11	1/12
バイアス	1	1/2	1/3	1/3	1/3	1/4	1/4	1/5
電圧レベル数	2	3	4	4	4	5	5	6

LCDで3つの共通線が使われる場合、それは1/3デューティ比を持ちます(図2-3をご覧ください)。これは各セグメント駆動部が最大3つのセグメントを制御することを意味します。1つのセグメント駆動部から3つのセグメントを制御するために、バイアスは1/3が必要です。別の言葉では1つのセグメント駆動部だけを使って3つのセグメントを制御し得るには4つの異なる電圧レベルが必要です。単一セグメント線に接続されるLCDセグメントの各々は異なる共通電極を持ちます。

図2-4は2つの異なる背面(共通線)に接続され、且つ同じセグメント線によって駆動される2つのLCDセグメントの駆動波形を示します。この図は1/3のバイアスと4つの電圧レベルを持つ駆動波形を示し、そしてこれは6つまでの背面を駆動するのに充分です。けれども波形が1フレーム内の3周期を示すため、デューティが1/3で3つの背面を持つLCD硝子を示します。

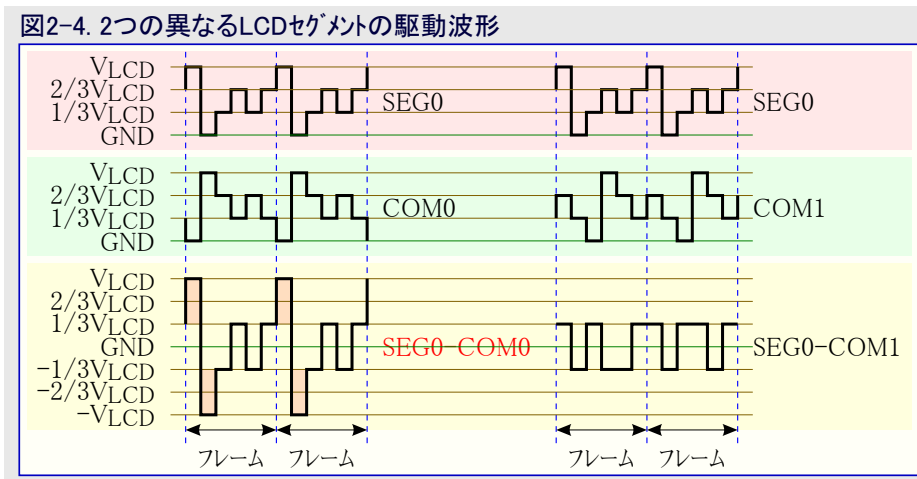


図2-4は2つの異なる共通電極に接続された2つの異なるLCDセグメントの駆動波形を示します。セグメント線は共用されます。図の左側は活性なセグメントを図解し、図の右側は不活性なセグメントを図解します。図の右側で表されるセグメントはLCD活性電圧閾値を越えないので不活性です。

2.5. LCD濃淡

LCDの濃淡はそのセグメント位置に於ける背面-セグメント線波形のRMS値の関数です。波形はLCD構造のどの点でも結果のRMS電圧が飽和電圧(注)以上、または可視閾値電圧以下のどちらかになるように生成することができます。

注: 飽和電圧は液晶が完全に偏光される電圧レベルです。

2.6. ATmega3290のLCD機能

ATmega3290は4つの背面(共通)線と40のセグメント線で最大160のLCDセグメントを駆動することができます。

高い柔軟性を提供するためにLCD駆動波形は選択可能です。デューティ比とバイアスは4つまでの背面(共通電極)と13~40セグメント線を持つLCD硝子のインターフェースを可能にするように設定可能です。LCD駆動に使われない線は標準入出力として使うことができます。LCDの濃淡は2.6~3.35V(VCCに依存)間で駆動部電圧レベルを変化することによって制御することができます。これらの電圧値はLCDインターフェースによって使われる間だけ、LCDピンに適用されます。

性能と電流消費を最適化するためにATmega3290は設定可能なフレーム速度を使い、“低電力波形”の使用を許します。低電力波形はセグメント線と共通線の切り替えが最小周波数を保つことを保証します。更に、ATmega3290の節電形態(休止形態)はLCD硝子の駆動継続と同時に電流消費を最低に低減することをMCUに許します。

組み込みLCD割り込み元のため、ソフトウェアドライバは完全な割り込み駆動にすることができます。これはLCDの更新に関連するタイミングが正しいことを保証するために使うことができます。従って更新されたLCDデータレジスタの一部分がLCD線にラッチされるのを避けることが可能です。

2.7. STK504のLCD硝子

この資料で記述されるLCDソフトウェアドライバはSTK500開発基板用追加部のSTK504用に作られています。STK504のLCDの簡単な説明がここで提供され、STK500とSTK504の更なる詳細は各々それらの使用者の手引きで得られます。

STK504に装着されたLCD硝子が図2-5.で図解されます。これは7つの英数(アルファベットと数字)象徴物、4つの数字象徴物と様々な図形象徴物から成ります。



このLCD硝子は4つの背面(共通電極)と40セグメント線を通して制御される合計160のセグメントを持ちます。ATmega3290は160セグメントを駆動する能力があるので、デバイスは全てのセグメントを制御します。この資料で記述されるソフトウェアドライバはSTK504のLCD硝子がSTK504の標準構成設定に従って接続されていると仮定します。

セグメントは3つの性質が異なる群に分けることができます。

- ・ 英数(アルファベットと数字)の14セグメント象徴物
- ・ 数字の7セグメント象徴物
- ・ 図形象徴物

I/Oメモリ配置内のLCDセグメントの位置のため、これらの群は独立した関数によって処理されます。これは本応用記述の実装部で説明されます。

3. 実装

本章はATmega3290とSTK504のLCD表示器間の物理的な接続についての情報とLCDセグメントがどう制御されるかを得るための場所の説明を含みます。ASCII文字をLCD桁にどう割り当てるかも記述されます。

3.1. LCDとATmega3290間の接続

LCD表示器とATmega3290間の接続はAVR Studio®のオンライン ヘルプ システムで得られるSTK504使用者の手引きで記述されます。LCDへの電力がATmega3290から供給される、LCDが独立した供給線をもたないことに注意することが重要です。

3.2. LCDデータレジスタ

LCDセグメントは20個のLCDデータレジスタ(LCDDR19~0)のビットを通して個別に制御されます。各セグメントはLCDDR19~0内の対応するビットの設定(1)または解除(0)を通して個別に制御されます。AVRのLCD単位部はLCDへの物理的な駆動波形の符号化を処理します。

3.3. LCDデータレジスタとLCDセグメント間の関連

効率的なLCDソフトウェアドライバを作成するため、コード量に関してLCD桁とセグメント線/共通線間の物理的な接続は上手に構成されなければなりません。各LCD桁はLCDデータレジスタと同じまたは同様な方法で関連しなければなりません。これはASCII文字から以下で記述されるLCDセグメント制御符号(SCC:Segment Control Code)への変換を簡単にします。SCC符号はLCDDR19~0レジスタに書くことができます。従ってLCDDR19~0レジスタに書かれるデータはLCDデータレジスタ内のそれらの対応ビットの設定(1)または解除(0)を通して直接制御します。

3.4. LCDセグメント制御符号

STK504上のLCD表示器のセグメント割り当て分析は、7と14の両セグメント文字を網羅するための1つの標準関数の作成が効率的でないことを示します。セグメント割り当て表は(AVR Studioヘルプの)STK504使用者の手引きの追補Bで得られます。従って7または14のセグメント文字への書き込みは独立した関数によって網羅され、独立して説明されます。

3.4.1. 14セグメント文字

奇数桁は関連するLCDデータレジスタの下位ニブル使用で、偶数桁は上位ニブル使用です。各LCD桁に対して4つの異なるLCDデータレジスタが使われ、それらは全て異なる背面(共通電極)に関連します。LCDデータレジスタと桁間の関連は使われるLCDデータレジスタ間のアドレス変位が固定化されるように構成されます。アドレス間の空間は全ての場合で5です。最後に、桁2と3、4と5、6と7は各々LCDDR0、LCDDR1、LCDDR2で始まる対です。

LCDデータレジスタ内のビットを設定(1)または解除(0)するために必要なLCDセグメント制御符号(SCC)にASCII文字を変換するのに、STK504使用者の手引き内のビット割り当て表が使われます。この表が全てのLCD桁に対して再使用できるので、1つのビット割り当て表だけが使用に必要です。

LCD桁の全セグメント線を制御するために14ビットが必要とされます。故にLCDのSCCは16ビット幅で、LCDのSCCが1つのLCDデータレジスタに関連するように整列されます。従ってSCCは次のように使うことができます。

$SCC = \{bit\ 15:0\} = \{Nibble3:0\} \sim \{LCDDR_x+15, LCDDR_x+10, LCDDR_x+5, LCDDR_x\};$

表3-1はLCD表示の先頭(左端)文字にセグメントがどう対するかを示します。

表3-1. 文字1用STK504 LCDセグメント割り当て

共通電極	ビット								レジスタ
	7	6	5	4	3	2	1	0	
COM3	2C	NB	2M	2D	1C	BEL	1M	1D	LCDDR15
COM2	2H	2N	2L	2E	1H	1N	1L	1E	LCDDR10
COM1	2B	2K	2J	2G	1B	1K	1J	1G	LCDDR5
COM0	TU	2A	2I	2F	MO	1A	1I	1F	LCDDR0

文字1に対するSCCは以下です。

$SCC = \{bit\ 15:0\} \sim \{LCDDR15[3:0], LCDDR10[3:0], LCDDR5[3:0], LCDDR0[3:0]\};$

$SCC = \{bit\ 15:0\} \sim \{1C, 0, 1M, 1D, 1H, 1N, 1L, 1E, 1B, 1K, 1J, 1G, 0, 1A, 1I, 1F\};$

例: 文字位置1に文字'A'を表示するには1A,1B,1C,1E,1F,1Gが設定(1)される必要があります。従って文字'A'に対する正しいSCCは以下です。

$SCC_A = \{bit\ 15:0\} = \{1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1\} = \{0x8995\};$

予め定義された全てのSCC符号の完全な一覧はstk504_lcd.cソースファイルで得られます。

関数は文字の部分でないビットを変えることなしに正しいビットを書く責任があります。

3.4.2. 7セグメント文字

7セグメント文字に対する書き込み関数は同様の方法で実装されますが、これらの文字が7ビットだけを必要とするため、これらは単一バイトに格納されます。従ってセグメント制御符号は8ビット幅です。14セグメント文字のように、LCDデータレジスタと桁間の関連は使われるLCDデータレジスタ間のアドレス変位が固定化され、アドレス間の空間は全ての場合で5です。

$SCC = \{bit\ 7:0\} = \{bit1:0\} \sim \{LCDDR_x+15, LCDDR_x+10, LCDDR_x+5, LCDDR_x\};$

表3-2. STK504 LCDセグメントの7セグメント文字割り当て

共通電極	ビット								レジスタ
	7	6	5	4	3	2	1	0	
COM3	9A	9B	10A	10B	11A	11B	12A	12B	LCDDR19
COM2	9F	9G	10F	10G	11F	11G	12F	12G	LCDDR14
COM1	9E	9C	10E	10C	11E	11C	12E	12C	LCDDR9
COM0	9D	OUT	10D	P	11D	COL	12D	ICE	LCDDR4

各LCDDRレジスタ内に2つのビットだけが格納され、このため完全な文字を書くには4つのレジスタ書き込みが必要です。SCCは次のように構成されます。

$SCC = \{ A, B, F, G, E, C, D, 0 \};$

例: 7セグメント文字9(STK504使用者の手引き内のLCDセグメント割り当てをご覧ください)に文字'A'を表示するには、次のセグメント、9A,9B,9C,9E,9F,9Gが設定(1)される必要があります。従って正しいSCCは以下です。

$SCC_A = \{ 1, 1, 1, 1, 1, 1, 0, 0 \} = 0xFC;$

3.5. 象徴物

7セグメントと14セグメントの文字に加えて使うことができる幾つかの図形象徴物があります。I/O割り当て一面に散らされているため、これら进行处理する最も容易な方法は独立した関数によることです。これらのSCC符号の形式は7と14のセグメント関数と僅かに異なります。

表B-3. LCDセグメント割り当て

共通電極	ビット								レジスタ
	7	6	5	4	3	2	1	0	
COM3	9A	9B	10A	10B	11A	11B	12A	12B	LCDDR19
	AVR	S7	S8	S9	7C	KEY	7M	7D	LCDDR18
	6C	WRM	6M	6D	5C	8BC	5M	5D	LCDDR17

象徴物についてのSCC符号は次の形式を持ちます。

$SCC = \{ \text{bit}:b[7..0] \text{ n}[19..0] \}$

ここでの**b**はLCDDRレジスタ内のビット位置で、**n**はLCDDR0に対する相対変位です。

例: AVR象徴物を設定するためのSCC符号は以下です。

$SCC_{AVR} = \{ \text{bbb nnnnn} \} = \{ \text{bit}:7 \text{ offset}:18 \} = \{ 111 10010 \} = 0xF2;$

完全なSCC符号一覧はstk504_lcd.cソースファイルで得られます。

4. ファームウェア説明

本章はこのドライバに含まれる関数とそれらの使用法についての情報を含みます。ファームウェアはAtmelのウェブサイト <http://www.atmel.com/products/AVR/> からダウンロードすることができます。コンパイラ情報と設定、デバイス設定、目的対象設定の情報と包括的なソース資料に関してはソースと共に含まれるreadme.htmlファイルをご覧ください。

4.1. LCDドライバ関数

表4-1. LCDソフトウェアドライバ関数

関数名	引数	戻り値	説明
LCD_init(global)	void	void	LCDデータレジスタの緩衝部に使われるLCD表示データ緩衝部を初期化します。LCDフレーム速度と濃淡が選択されます。電力削減レジスタのPRLCDビットが解除(0)されます。
LCD_Update(global)	void	void	LCDデータ緩衝部の内容をLCDDR _x レジスタに転送します。レジスタを更新するためにLCD割り込みルーチンに代わって更新を強制します。
LCD_vect_interrupt(local, interrupt service routine)	void	void	LCD表示データ 緩衝部をLCDデータレジスタにラッチします。ラッチはLCD_timer変数とLCD_status.updateRequired変数に依存します。
LCD_PutChar(global)	unsigned char c, unsigned char digit	void	ASCII文字は引数cで渡されます。この引数は14セグメント文字に対するSCC符号に変換され、引数digitによって示される位置に表示されます。データが直接LCDDRレジスタではなく、LCD表示緩衝部に書かれることに注意してください。実際のデータ更新はLCD_vect_interrupt関数によって処理されます。
LCD_PutDigit(global)	unsigned char c, unsigned char digit	void	ASCII文字は引数cで渡されます。この引数は7セグメント文字に対するSCC符号に変換され、引数digitによって示される位置に表示されます。データが直接LCDDRレジスタではなく、LCD表示緩衝部に書かれることに注意してください。実際のデータ更新はLCD_vect_interrupt関数によって処理されます。
LCD_AllSegments(global)	unsigned char input	void	引数が0ならばLCD表示緩衝部内の全セグメントを解除(0)し、さもなければ全セグメントを設定(1)します。
LCD_SetSeg(global)	unsigned char symbol, unsigned char input	void	引数inputが0ならばLCD表示緩衝部に於いて引数symbolで示された象徴物を解除(0)し、さもなければその象徴物を設定(1)します。
LCD_BarSeg(global)	unsigned char value	void	温度/電池の象徴物内の8つのセグメントに書くための関数です。復号が多少複雑なので、これらの象徴物への書き込みを簡単にします。

表4-2. で一覧にされる変数は主LCDドライバ関数とLCD割り込みルーチン間の通信を制御するのに必要とされる全域変数です。変数の機能詳細は表4-2. で提供されます。

表4-2. LCDソフトウェアドライバ使用時に必要とする全域変数

関数名	説明
LCD_status.updateRequired(bitfield, bool)	真なら、LCD_vect_interruptルーチンはLCD表示データ緩衝部をLCDデータレジスタにラッチすることを許可されます。偽なら、割り込みルーチンはLCD表示データをラッチしません。従ってこの変数はLCD表示データ緩衝部のラッチの要求または防止に使うことができます。LCD表示データ緩衝部を更新する間、この変数はFALSEに設定されるべきで、このためにLCD表示データ緩衝部が完全に更新されるまでラッチは全く実行されません。
LCD_status.updateComplete(bitfield, bool)	LCD表示データ緩衝部がラッチされてしまうと、この変数がTRUEに設定されます。これはLCD_vect_interruptルーチンで行われます。従ってこの変数はLCD表示データ緩衝部のアクセス後にLCD表示緩衝部に書かれたデータがラッチされてしまったかを調べるのに使うことができます。LCD割り込みが固定間隔で起きるので、これは更新タイミングを処理するのに使うことができます。この変数はLCD_PutChar/LCD_PutDigit関数に対する呼び出し制御に使うことができます。
LCD_timer (Unsigned char)	この変数はLCD_vect_interruptルーチンで減少(-1)されます。値が0になると、LCD表示データ緩衝部の次のラッチが起きます。同時に既定計時値も再設定されます。変数はLCD更新間隔を制御します。このLCD更新はLCD_status.updateRequired変数にも依存します。LCD_status.updateRequiredがFALSEのために更新ができない場合、LCD更新は次のLCDフレーム開始割り込みの間に試みられます。LCDの更新が直ちに求められる場合、この変数を1に設定することができます。

5. 目次

要点	1
1. 序説	1
2. 動作の理屈	1
2.1. LCD硝子説明	2
2.2. LCDフレーム速度	2
2.3. セグメント駆動部と共通電極	2
2.4. デューティサイクルまたはデューティ比	3
2.5. LCD濃淡	4
2.6. ATmega3290のLCD機能	4
2.7. STK504のLCD硝子	4
3. 実装	4
3.1. LCDとATmega3290間の接続	4
3.2. LCDデータレジスタ	4
3.3. LCDデータレジスタとLCDセグメント間の関連	5
3.4. LCDセグメント制御符号	5
3.4.1. 14セグメント文字	5
3.4.2. 7セグメント文字	5
3.5. 象徴物	5
4. ファームウェア説明	6
4.1. LCDドライバ関数	6
5. 目次	7
お断り	8



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

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、または利益の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。

© Atmel Corporation 2006. 不許複製 Atmel®、ロコとそれらの組み合わせ、AVR®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。

© HERO 2021.

本応用記述はAtmelのAVR063応用記述(doc8009.pdf Rev.8009A-04/06)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。