

# AVR065 : STK502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
- STK502 LCD表示器のインターフェース

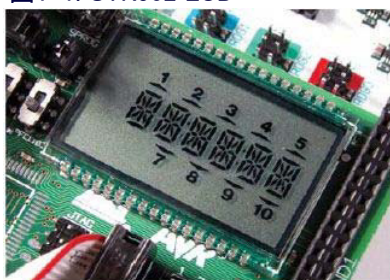
## 1. 序説

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用されます。LCDモジュールは図形と文字の表示に使用する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モジュールはインターフェースが容易ですが、明らかに高価です。これらは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、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。そしてMCUはLCD硝子を直接駆動することができ、LCDモジュールに統合された駆動部の必要を無くします。LCD硝子がLCDモジュールよりもっと安い価格なので、最大10倍程、表示器の費用を低減します。

ATMELのATmega169(P)フラッシュ マイクロ コントローラは最大100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対人インターフェースを必要とする電池応用に対してこれを理想的にします。

図1-1. STK502 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硝子関連で使用される用語に対する紹介を提供します。

加えてATmega169(P)表示器駆動部の要約と応用例で使用するLCD硝子の記述内容が検討されます。更に、本章はATmega169(P) LCD機能の要約と応用例で使用するLCD硝子の説明を含みます。



8ビット **AVR**<sup>®</sup>  
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. 2530E-07/08, 2530EJ2-12/13

## 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)を使用する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、換言すると、LCDセグメント上のより高い電圧がより良い濃淡と視野角になります。

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硝子に使用され、一方STN LCDは240背面程の数を持つ能力があります。この制限は背面がより増やされるための透過性減少のためです。TNとSTNについてのより多くの詳細は、<http://en.wikipedia.org/wiki/LCD> で得られます。

### 2.1.1. 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。

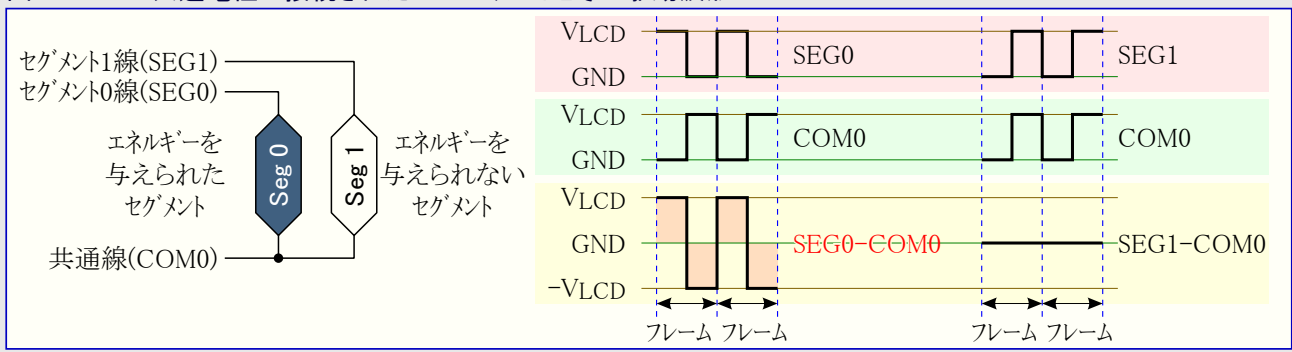
高いフレーム速度が使用される場合、残像が起き得ます。残像はLCDセグメントが正しくOFFされない時です。この効力は**デューティとバイス(以下で説明)**にも依存し、実際に使用されるデューティとバイスに対してフレーム速度を適合させる必要があるかもしれません。通常、残像は十分に低いフレーム速度を使用することによって避けることができます。100Hzのフレーム速度が残像を防ぎます。

### 2.1.2. 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群に対する共通を意味する用語としての記述です。

駆動波形は多数のセグメント間で共用される、1つの(共通)電極を通して個別セグメントを活性にするように符号化されなければなりません。1つの共通電極だけが使用されるなら、換言すると、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、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あるべきです。これはエネルギーを与えられるべきセグメント上の最大電位差と、エネルギーを与えられるべきでないセグメント上の小さな電位差または全くなしに帰着します。**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各々示します。

図2-1. 1つの共通電極に接続された2つのセグメントとその駆動波形



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

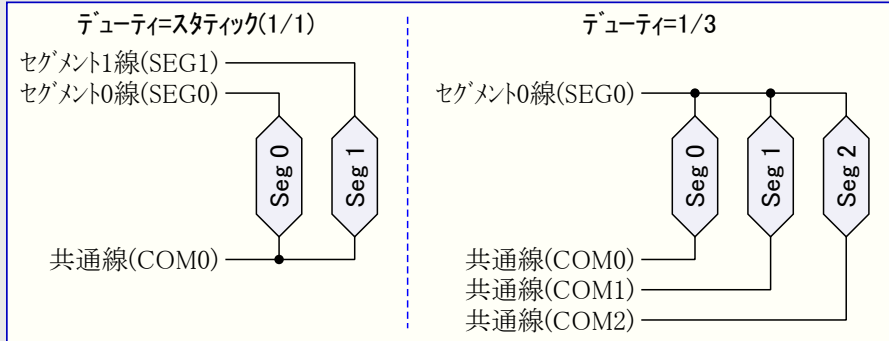
上図の波形の左側は活性なセグメントの駆動波形を、右側は不活性なセグメントの駆動波形を示します。

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をもっと複雑です。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。

2.1.3. 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用される数値です。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を制御する時に使用されるデューティ比間の関連を図解します。

図2-3. 1つ及び3つの共通電極を使用することによって制御されるLCDセグメント



2.1.4. 駆動バイアス

駆動バイアス(または単にバイアス)はLCDを駆動する時に使用さ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。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。デューティ比の定義により、使用さ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。

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つの共通線が使用される場合、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できるために、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用して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. 2つの異なるLCDセグメントの駆動波形

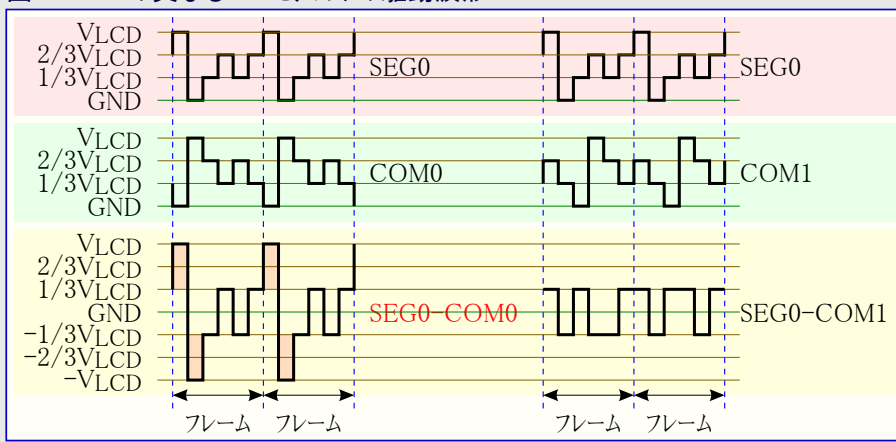


図2-4.は2つの異なる共通電極に接続された2つの異なるLCDセグメントの駆動波形を示します。セグメント線は共用されます。図の左側は活性なセグメントを図解し、図の右側は不活性なセグメントを図解します。図の右側で表されるセグメントはLCD活性電圧閾値を越えないので不活性です。

### 2.1.5. 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。飽和電圧は液晶が完全に偏光される電圧レベルです。

## 2.2. ATmega169(P)のLCD機能

ATmega169(P)は4つの背面(共通)線と25のセグメント線で最大100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13~25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駆動に使用されない線は標準入出力として使用することができます。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用される間だけ、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ATmega169(P)は設定可能なフレーム速度を使用し、“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、169(P)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用することができます。従って更新されたLCDデータレジスタの一部がLCD線にラッチされるのを避けることが可能です。

## 2.3. STK502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STK502用に作られています。STK502のLCDの簡単な説明がここで提供され、STK500とSTK502の異なる詳細は各々それらの使用者の手引きで得られます。

STK502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(アルファハットと数字)象徴物と様々な図形象徴物、1~10の数字、鐘、電池低下記号、誘導矢印から成ります。

図2-5. STK502のLCD硝子配置

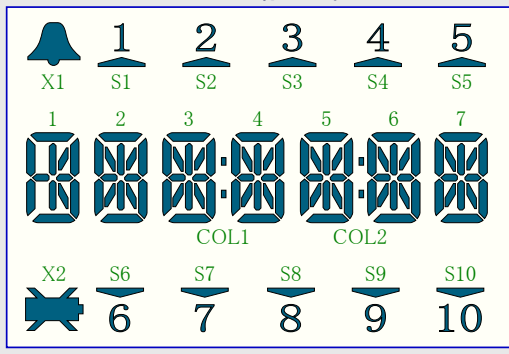
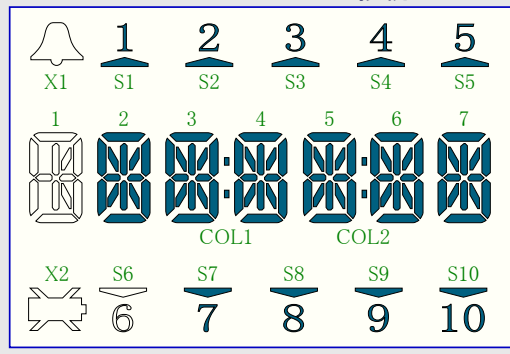


図2-6. STK502でのLCDセグメント接続

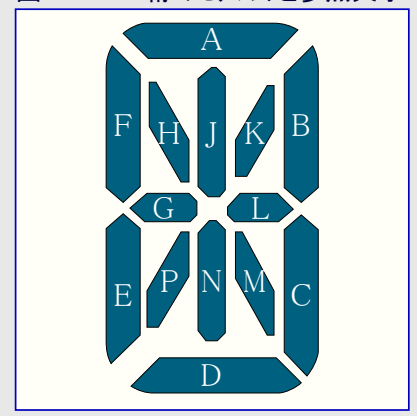


このLCD硝子は4つの背面(共通電極)と30セグメント線を通して制御される合計120のセグメントを持ちます。ATmega169(P)が100セグメントを駆動する能力なので、来た状態のSTK502の標準構成でLCD硝子のいくつかのセグメントは接続されていません(図2-6.をご覧ください)。この資料で記述されるソフトウェアドライバはSTK502のLCD硝子がSTK502の標準構成に従って接続されていると仮定します。

STK502のLCDは各セグメント群が英数文字を表示する能力を持つ、6つの同様のセグメント群を持ちます。このような英数文字を表示する能力を持つセグメント群の1つは以降でLCD桁として参照されます。これは14個の独立したセグメントから成ります。図2-7.はこのLCD桁とLCD桁内の各々のセグメントを参照するのに使用される文字を示します。

STK502使用者の手引きでのデータシートに従い、LCDは4/1デューティと1/3バイアスで動かされなければなりません。これは4つの背面全てを制御できるようにするためです。LCDには3Vを供給することが推奨されます。

図2-7. LCD桁のセグメントと参照文字



### 3. 実装

本章はこのドライバに含まれる関数についての情報とそれらの使用方法を含みます。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。コンパイラの情報と設定、デバイス設定、目的対象設定情報、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](#)をご覧ください。

#### 3.1. LCDとATmega169(P)間の接続

LCD表示器とATmega169(P)間の接続はSTK502使用者の手引きで記述されます。LCDへの電力がATmega169(P)から供給され、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。

##### 3.1.1. 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。20個全てのLCDデータレジスタが使用される訳ではなく、ATmega169(P)では各共通電極に対する最上位レジスタ内の上位7ビットが使用されません。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部署はLCDへの物理的な駆動波形の符号化を処理します。

##### 3.1.2. 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。

##### 3.1.3. LCDセグメント制御符号

STK502のLCDソフトウェアドライバで利用可能な6つのLCD桁の接続間の類似点はSTK502使用者の手引きのビット割り当て表から見ることができます。偶数桁は関連するLCDデータレジスタの下位ニブル使用で、奇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用され、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用される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、STK502使用者の手引き内のビット割り当て表が使用されます。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16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は次のように使用することができます。

$$SCC = \{\text{bit } 15:0\} = \{\text{Nibble } 3:0\} \sim \{\text{LCDDR}_x+15, \text{LCDDR}_x+10, \text{LCDDR}_x+5, \text{LCDDR}_x\};$$

SCCのニブルとLCDデータレジスタ間の関連はどのLCD桁がアクセスされるかによって条件付けされます。偶数LCD桁はLCDデータレジスタの下位ニブルを使用し、一方奇数LCD桁はLCDデータレジスタの上位ニブルを使用します。

桁で文字“A”が表示される場合の例を考察してください。文字“A”は図3-1に記載されるように見えるでしょう。

文字“A”に見えるにはLCDセグメント{A,B,C,E,F,G,L}の活性化が必要です。LCDのSCCでこれを記述するのにSTK502のビット割り当てが使用されます。表3-1.で示されるようなビット割り当て表は奇数と偶数の両方のLCD桁を表します。

表3-1. LCDデータレジスタ内のLCD桁セグメント割り当て

レジスタ	ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
LCDDR <sub>x</sub> +0	K	-	-	A	K	-	-	A
LCDDR <sub>x</sub> +5	J	F	H	B	J	F	H	B
LCDDR <sub>x</sub> +10	L	E	G	C	L	E	G	C
LCDDR <sub>x</sub> +15	M	P	N	D	M	P	N	D

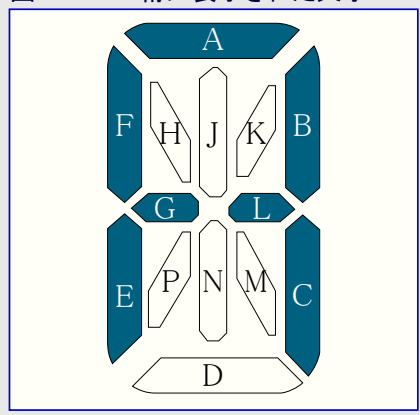
ASCII文字“A”に対するLCDのSCCを決めるには以下を考察してください。

ニブル	セグメント	文字“A”
SCC = {ニブル0}	{K, -, -, A}	{0, 0, 0, 1}
{ニブル1}	{J, F, H, B}	{0, 1, 0, 1}
{ニブル2}	{L, E, G, C}	{1, 1, 1, 1}
{ニブル3}	{M, P, N, D}	{0, 0, 0, 0}

従って文字“A”に対するSCCは次のように決めることができます。

$$SCC\_A = \{0x0, 0xF, 0x5, 0x1\} = 0x0F51$$

図3-1. LCD桁に表示された文字“A”



多くの標準ASCII文字と記号がLCDのSCC形式に変換されます。LCDのSCC符号はフラッシュメモリの表に格納され、ASCII文字をLCDのSCCに変換する時に使用されます。

正当なASCII記号のいくつかが表示不能のため、これらの多くはフラッシュメモリの表で使用されません。これらの表内の位置は、例えばLCDに対するお客様の記号の定義に使用することができます。

### 3.2. AVR Studio®のLCDプラグイン

STK502なしでこの応用記述で記述されるコードを評価/試験するために、AVR Studio 4にLCDプラグインをインストールすることができます。このプラグインはATMELのウェブ <http://www.atmel.com/products/avr/> でAVR Studio 4.12と共に見つかるAVR LCD Visualizerでダウンロードできます。これはAVR StudioでSTK502 LCD用形態設定ファイルを開くことによってSTK502のLCDのように見えるものの観測を可能にします。このファイルはMega169.disと名付けられ、標準インストールに於いてC:\Program Files\Atmel\AVR Tools\AvrLcd\Displays\Mega169\Folderで見つかります。

LCD Visualizerはあなた自身のLCD表示器を設計し、それをLCDのレジスタに関係させ、AVR Studioでその表示器を擬似動作/評価するのに使用することができます。

## 4. ファームウェア説明

本章はこのドライバの実装法と使用方法についての詳細な情報を含みます。

ここで得られる説明に加えて、LCDドライバと共にmain.cが提供されます。main.cファイルはLCDドライバ使用法の例で、ドライバを完全に理解するためにこれを学ぶことができます。

### 4.1.1. LCDドライバ関数

LCD\_driver.cファイル内で見られる関数は表4-1.で一覧にされ注釈されます。

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に使用される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。セグメント線と共通線が形態設定されます。
LCD_WriteDigit(global)	unsigned char c, unsigned char digit	void	引数cで渡されたASCII文字がLCDセグメント制御符号(SCC)に変換されます。SCCはLCDセグメント象徴物にASCIIを割り当てるデータです。SCCはLCD表示データ緩衝部に複写されます(実装の更新はLCD_SOF_interruptルーチンで処理されます)。引数digitはアクセスを望む桁番号です。本実装に於いて桁の番号は2~7に制限されます。
LCD_AllSegments(global)	unsigned char input (論理値として使用)	void	LCDの全セグメントを設定(1)または解除(0)します(LCD表示データ緩衝部だけを更新し、実際の更新はLCD_SOF_interruptルーチンで処理されます)。
LCD_SOF_interrupt (局所、割り込み処理ルーチン)	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。このラッチはLCD_timer変数とLCD_status.updateRequired変数に依存します。

### 4.1.2. マクロ

LCDソフトウェアドライバ使用時に関連するマクロが表4-2.で一覧にされます。

表4-2. LCDソフトウェアドライバで使用されるマクロ

マクロ名	引数	説明
LCD_CONTRAST_LEVEL	level	LCD濃淡を調整します。0~15の範囲で、15が最も高い濃淡レベルを生成します。
LCD_SET_COLON	active	LCD内のコロン(:)を表示または非表示にします。有効な入力は[TRUEまたはFALSE]です。

### 4.1.3. 全域変数

一覧にされる変数は主LCDドライバ関数とLCD割り込み関数間の通信を制御するのに必要な全域変数です。変数の機能詳細が提供されます。

表4-3. LCDソフトウェアドライバ使用時に必要な全域変数

変数名	説明
LCD_status.updateRequired (bitfield, bool)	TRUEなら、LCD_SOF_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されます。FALSE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を要求または防止するのに使用できます。LCD表示データ緩衝部更新中、LCD表示データ緩衝部が完全に更新されてしまうまでラッチが実行されないように、この変数はFALSEに設定されるべきです。
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_SOF_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表示データ緩衝部に書かれたデータがラッチされたかを調べるのに使用できます。LCD割り込みは固定間隔で起こるので、更新タイミングを処理するのに使用することができます。この変数はLCD_Write_Digit関数呼び出しの制御に使用することができます。
LCD_timer (Unsigned char)	この変数はLCD_SOF_interruptルーチンで減少されます。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。同時に既定計時数が再設定されます。この変数はLCD更新間隔を制御します。LCDの更新はLCD_status.updateRequired変数にも依存します。LCD_status.updateRequiredがFALSEのために更新が実行できない場合、LCDの更新は次のLCD_SOF割り込み中に試みられます。直ちにLCD更新を望む場合、主応用でこの変数を1に設定することができます。

## 4.2. 関数の流れ図

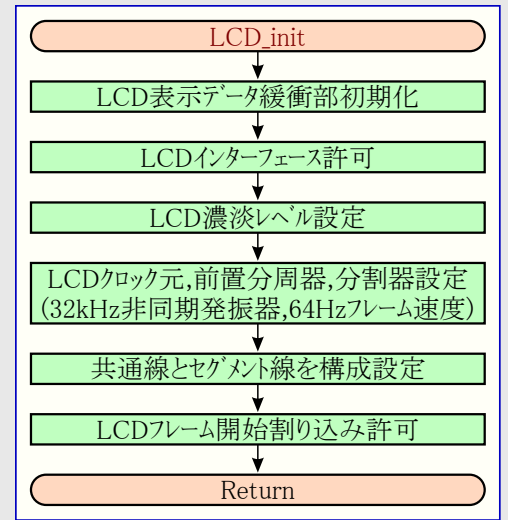
ドライバは以下の文章と流れ図で記述される関数から成ります。

### 4.2.1. LCD\_init

LCDソフトウェアドライバを初期化するために使用される関数はLCD\_initと呼ばれます。

LCD\_init関数は最初にLCD表示データ緩衝部を解除します。LCD\_displayData緩衝部が定義される時の初期化よりも、これがより効率的なコード量のため、この手法が選ばれています。AVRのLCD部署がSTK502に配されたLCDに合うように形態設定されて許可されます。濃淡レベルが設定され、クロック元が外部32kHz非同期発振器に設定され、LCDクロック前置分周器と分割器が64Hzのフレーム速度を生成するように設定されます。最後にLCDフレーム開始割り込みが許可されます。全体割り込みは許可されません。これは殆ど主関数で扱われます。

図4-1. LCD\_init関数の流れ図

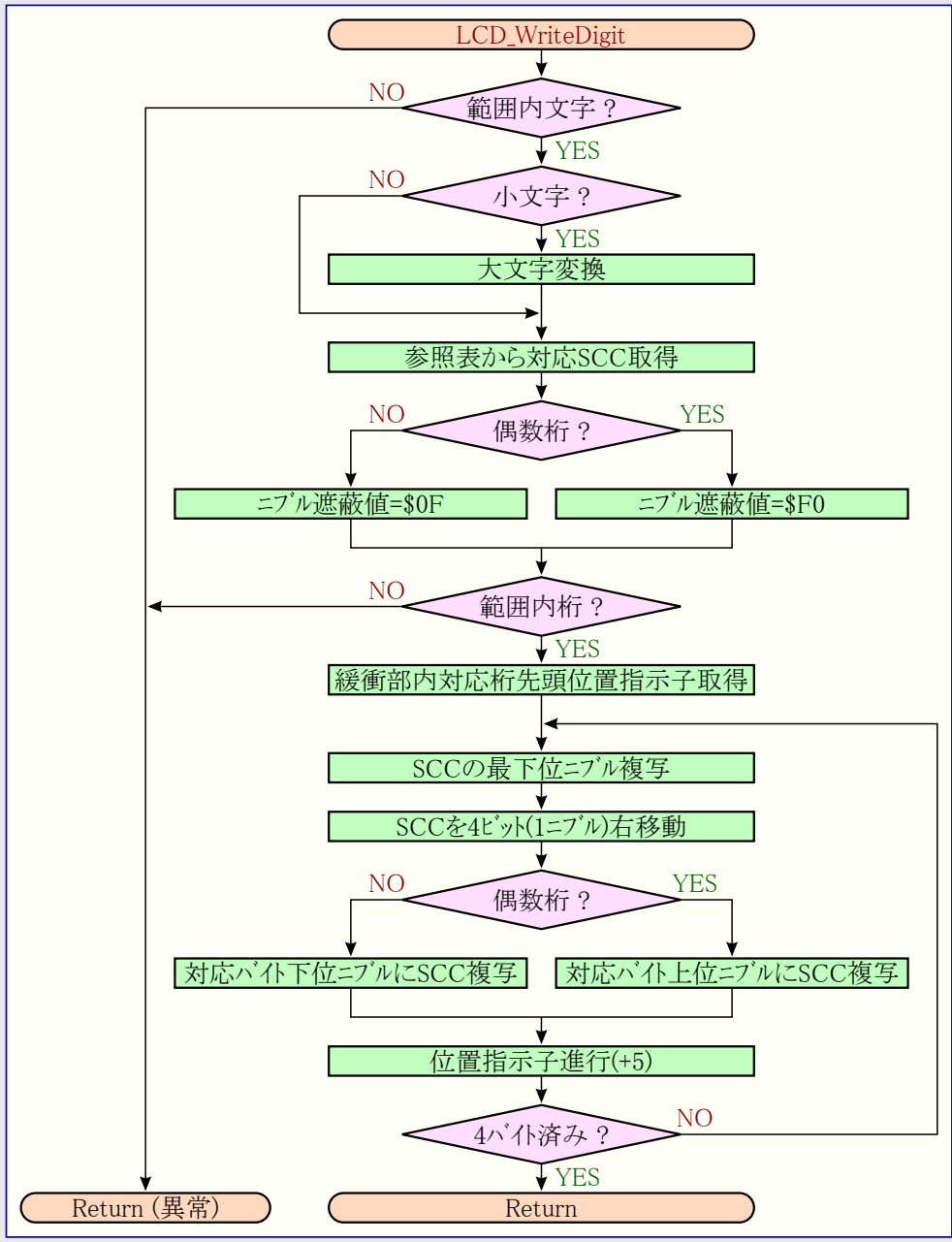


#### 4.2.2. LCD\_WriteDigit

LCD\_WriteDigit関数の目的はASCII文字を変換してLCD\_displayData緩衝部に複写することです。LCDデータレジスタへの緩衝部のラッチがLCDのSOF割り込みで扱われることに注意してください。

関数の引数として渡された文字はASCII表の範囲内の文字であることを確かめるために調べられます。その文字が小文字の場合、大文字に変換されます。文字-(NULL(\$00)からの差)がフラッシュメモリの表の参照として使用されます。表から取得したデータは2バイト値のLCDセグメント制御符号(SCC)です。SCC内の4つの各ニブルは更新されつつある指定LCD桁に関連するLCD表示データ緩衝部の対応する要素に結合(合成)されます。LCD表示データ緩衝部を読み戻した情報がLCDデータレジスタに直接ラッチされます。従ってSCCの4つのニブルはこの緩衝部を通して対応するLCDデータレジスタに結合(合成)されます。

図4-2. LCD\_WriteDigit関数の流れ図

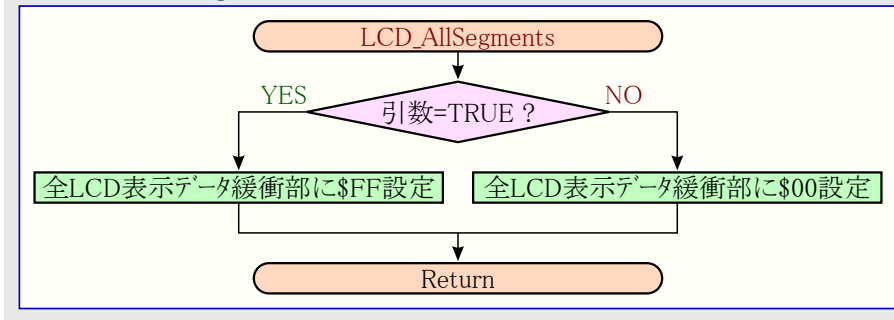




#### 4.2.3. LCD\_AllSegments

呼び出し引数に依存してLCD\_AllSegmentsはLCD\_displayData緩衝部全体を設定または解除します。これは結局、LCDセグメントの設定(表示)または解除(非表示)になります。

図4-3. LCD\_AllSegments関数の流れ図



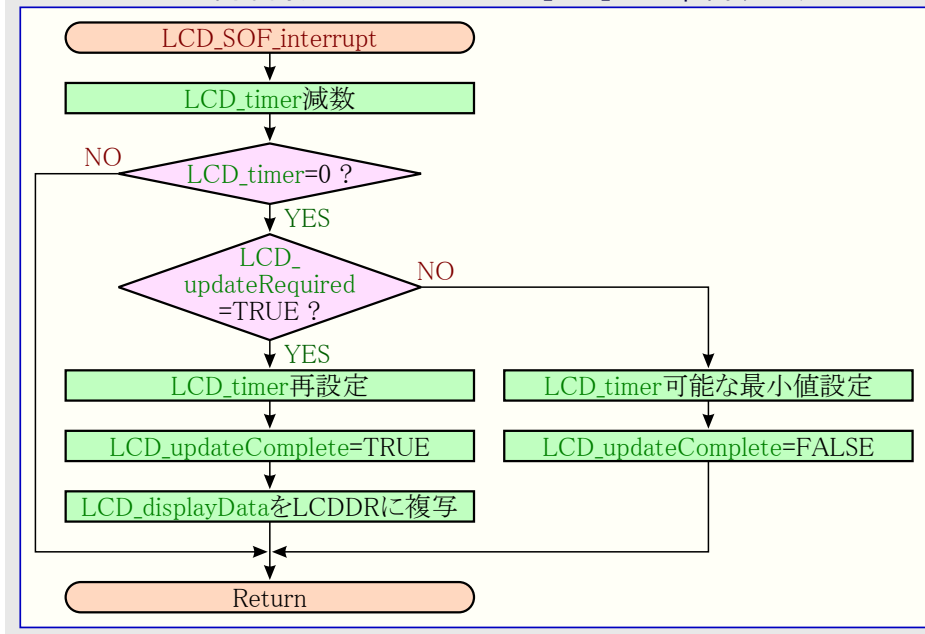
#### 4.2.4. LCD\_SOF\_interrupt

LCDの更新はLCD\_SOF\_interruptによって扱われ、割り込み発生時にLCD\_displayData緩衝部からのデータをLCDデータレジスタにラッチします。LCD\_status.updateRequiredフラグとLCD\_timer変数の2つの変数がLCD更新に影響を及ぼします。

LCD\_SOF\_interruptが実行される毎にLCD\_timerが減少されます。それが一度0に達すると、LCD\_status.updateRequiredフラグが調べられます。その検査がTRUEなら、LCD\_timerは既定計時値で再設定され、そしてLCD\_displayDataがLCDデータレジスタにラッチされます。LCDの更新が完了すると、LCD\_displayDataがラッチされたことを示すためにLCD\_status.updateCompleteフラグが設定(TRUE)されます。

けれどもLCD\_timerが0に減少されたけれど、FALSEであるLCD\_status.updateRequiredによってLCD更新が防がれている場合、LCD\_timerは可能な最小計時値で設定され、そしてLCD\_status.updateCompleteフラグがFALSEに設定されます。これは後続する最初のLCD SOF割り込みで新規のLCD更新が試みられことを保証します。LCD\_status.updateCompleteフラグが解除(FALSE)される理由はLCD\_displayData緩衝部が更新されたかを調べるために主ルーチンで使用されるべきだからです。LCD\_status.updateCompleteフラグが解除(FALSE)されている場合、主ルーチンはLCD\_displayData緩衝部の更なる更新を始めるべきではありません。尚、LCD\_status.updateCompleteフラグが解除(FALSE)されている間、LCD\_displayData緩衝部への進行中のアクセスは終了されるべきではありません。

図4-4. LCDフレーム開始割り込み処理ルーチン(LCD\_SOF\_interrupt関数)の流れ図



#### 4.3. 文献一覧

1. ATMELのAVRウェブ: <http://www.atmel.com/products/avr/>の応用記述下で得られる「STK502使用者の手引き」
2. ATMELのAVRウェブ: [http://www.atmel.com/dyn/products/datasheets.asp?family\\_id=607](http://www.atmel.com/dyn/products/datasheets.asp?family_id=607)で得られる「ATmega169(P)データシート」
3. STK502使用者の手引きに含まれるLCD, H4042-DL DE5156/Lのデータシート
4. LCD技術, <http://en.wikipedia.org/wiki/LCD>



## 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

## 製品窓口

### ウェブサイト

[www.atmel.com](http://www.atmel.com)

### 技術支援

[avr@atmel.com](mailto:avr@atmel.com)

### 販売窓口

[www.atmel.com/contacts](http://www.atmel.com/contacts)

### 文献請求

[www.atmel.com/literature](http://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のウェブサイトに位置する販売の条件と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 2008. 全権利予約済 ATMEL®、ロゴ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 2013.

本応用記述はATMELのAVR065応用記述(doc2530.pdf Rev.2530E-07/08)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。