

AVR155 : AVR®の2線直列インターフェースを使う I²C LCD表示器のアクセス法

要点

- PhilipsのI²C規約に適合
- 容易な送受信機能用の2線直列インターフェース主装置ドライバ
- 2行16桁LCD表示器の初期化と使用
- C言語ソースコード
- 最も一般的なI²C LCDドライバに合致

1. 序説

消費者、電気通信、工業電子機器で使うのに効率的な費用のIC間バスの要求でPhilipsのI²Cバスの開発が行われました。今日、I²Cバスは多数の周辺装置やマイクロコントローラで実装され、低速応用に関してそれを良好な選択にします。

I²Cインターフェースを持つ大量のICに合わせるため、AVRのTWI系列はそれらの周辺装置に対する内蔵の2線直列インターフェース(TWI)を持ちます。TWIバスは、どのI²C適合装置とも従装置と主装置の両方として通信することができます。I²Cについてのより多くの情報とTWIについての詳細はPhilipsのI²C仕様とAVRのデータシートで得られます。

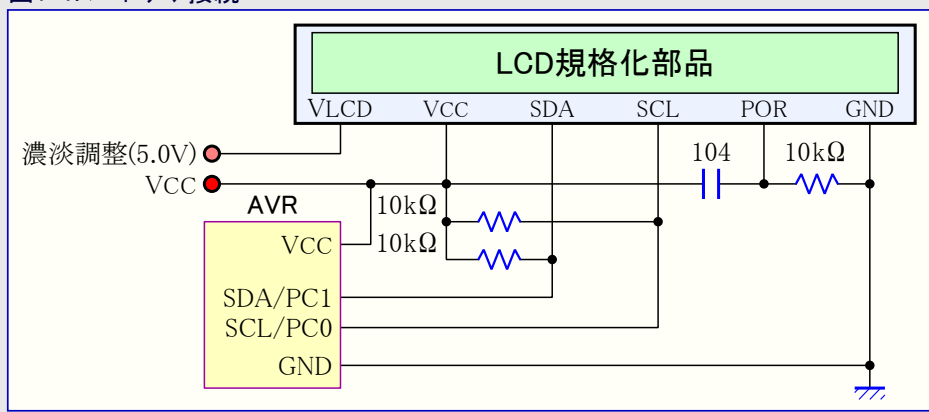
この応用記述はバス処理用のTWIドライバを含み、BatronのLCD表示器に於けるPhilipsのI²C LCD駆動器のアクセス方法を記述します。



8ビット **AVR**[®]
マイクロコントローラ

応用記述

図1-1. ハードウェア接続



2. 動作の理屈

BatronのLCDはI²C使用者インターフェースと実装済みASCII表を持つ2行16桁のアルファベットと数字の表示器です。このLCDはAVRとそのTWIハードウェア部からのI²C直列通信で構成設定及び制御が行われます。

AVRのTWIはI²Cに完全適合で、PhilipsのI²C仕様に従うどのI²C装置もアクセスすることができます。これはPhilips I²C LCD駆動部の構成設定と使用を容易にします。

AVRソフトウェアはCで書かれ、2つの主要な部分に分けられています。1つの部分はLCD制御で、LCDに対して必要な制御とデータバートを処理し、一方2つ目の部分はTWIドライバで、I²Cバスの処理を行います。

AVRのTWI部は基本的にTWINTフラグによって制御されます。このフラグはバスに対する全ての活動を開始すると、活動が行われた時を合図するのに使われます。TWINTフラグが設定(1)されると、TWSRは最後の活動とそれが成功したか否かに対応する状態値を含みます。TWIを通してI²Cバスへメッセージを送るとき、この符号は殆ど状態機構のように見え、状態機構での次の段階は常にTWSRから読まれる最後の段階の結果に基づきます。図2-1をご覧ください。

TWIはそれを非常に強力な単位部にするI²C信号とデータの流れにすごく近く、それはまた使うのにかなり複雑でもあります。TWIの使用を簡単にするため、この応用記述はLCD制御とTWI主装置ドライバの2つ領域に分けられ、両方共にCで書かれています。

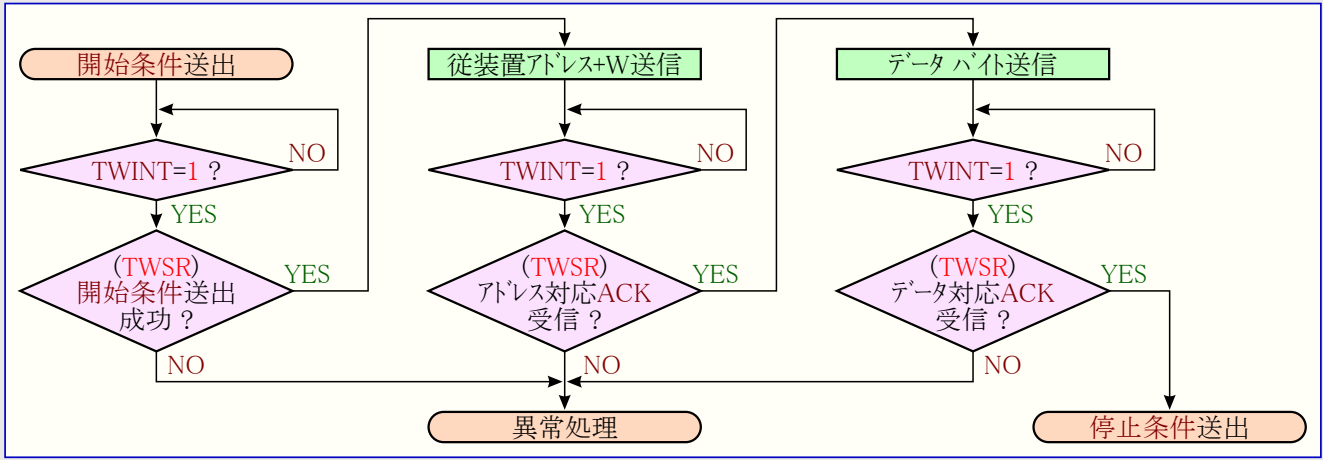
本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 1981B-09/05, 1981BJ4-04/21

TWI主装置はTWI部を通したI²Cバスに対する全ての信号とデータの送信を処理する通信ドライバです。これは標準的なドライバで、どの主装置I²Cアクセスにも使うことができ、呼び出す関数からの最小情報を必要とするだけです。このドライバはそれ自身のCファイルで実現され、TWIドライバを必要とする、どのCプログラムにもインクルードすることができます。

LCD制御はLCD仕様に注意して構成設定し、データと制御のバイトをTWI主装置ドライバを通してLCDへ送ります。LCDを2行表示に構成設定し、スクロール文字として文字列"AVR Atmega163 with TWI interface to I²C"を表示します。

図2-1. 基本TWI処理



2.1. TWI主装置ドライバ

TWI主装置ドライバはI²Cバスとの信号とデータの流れる行いうTWI単位部を取り扱います。これは呼び出し関数からの最小の入力だけを必要とし、活動結果に対応する状態を返します。ドライバはTWINTフラグを監視するポーリング関数として公開されますが、容易に割り込み駆動に変更することができます(3頁の「実装」をご覧ください)。

ドライバは従装置アドレスと送受すべきバイトについての情報が必要です。これは主装置自身のアドレスで終わる構造体の配列としてこの関数へ渡されます。表2-1をご覧ください。

表2-1. 1つの従装置への送出用の構造体配列

配列	0	1
構造体	SLA+W	自身のアドレス
	送信バイト数	(無効)
	送信バイトへのポインタ	(無効)

表2-1は従装置へデータを送るための配列を示します。SLA+Wは従装置アドレスと従装置受信動作への設定です。バイト数は関数が従装置へ何バイト送るのかを示し、ポインタはバイトを得る場所を示します。この関数は無限数の構造体を持つ配列を扱えますが、常に従装置アドレスとして自身のアドレスを見つけた場合に終了されます。この配列が1つよりも多くの組+自身のアドレスの組を含む場合、その組間で再送開始条件を使います。表2-2のような配列が使われた場合、開始条件、SLA+W、nバイトのデータ、再送開始条件、SLA+W、nバイトのデータ、停止条件を送出します。

表2-2. 1または2つの従装置への送出用の構造体配列

配列	0	1	2
構造体	SLA+W	SLA+W	自身のアドレス
	送信バイト数	送信バイト数	(無効)
	送信バイトへのポインタ	送信バイトへのポインタ	(無効)

この関数は主装置受信も扱えます。それは構造体内のポインタが受信バイト用の一時緩衝部へのポインタになります。表2-3をご覧ください。

表2-3. 1つの従装置からの受信用の構造体配列

配列	0	1
構造体	SLA+R	自身のアドレス
	受信バイト数	(無効)
	一時緩衝部へのポインタ	(無効)

このドライバはそれらの間での再送開始条件で混合した主装置受信と主装置送信を扱うこともできます。ドライバが通信を終えた後、呼び出した関数へ状態値を返します。この値は転送の結果で、これは成功した場合の\$FFまたは失敗した場合のTWSR値です。完全な一覧については表2-4をご覧ください。

表2-4. TWI主装置ドライバ関数からの戻り値

戻り値	説明
\$00	不法な開始条件または停止条件のためのバス異常
\$20	従装置アドレス+W送信でNACK受信
\$30	データバイト送信でNACK受信
\$38	調停敗退
\$48	従装置アドレス+R送信でNACK受信
\$58	データバイト受信でNACK受信
\$FF	送受信完了(正常終了)

2.2. LCD制御関数

このLCDのコードはLCD初期化、表示消去、スクロール文字作成の方法の例を与えます。表示器の全ての機能とより多くの詳細仕様についてはBatron LCDのデータシートとPhilips 2119 I²C LCD駆動器のデータシートをご覧ください。

3. 実装

このコードは可能な限り自身で説明するように書かれており、プログラムについてのより多くの詳細に関してはコード内の注釈を読んでください。

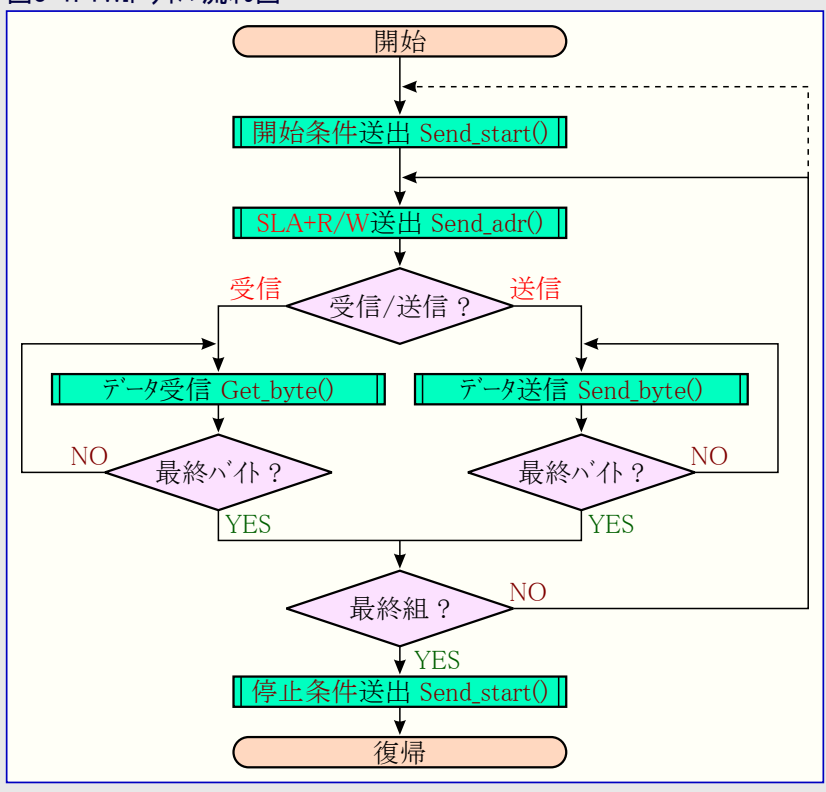
この関数はTWIドライバをより複雑でない方法で行うポータブル版のため、割り込み駆動ではありません。これを割り込み駆動版に変換するのはかなり簡単で、TWINT割り込みルーチン内へTWIドライバ関数全体を移動することによって行うことができます。状態機構の制御を維持するために状態機構であるべき関数を変更して全域変数を定義してください。送信すべきデータまたは受信用のバッファとしての全域配列も必要です。

コードはTWI_driver.cとLCD_control.cの2つのファイルに分けられ、Cで書かれています。両ファイルは独立したヘッダファイルを持ち、IARの1.50C/WIN版Cコンパイラで書かれています。これらはANSI Cで書かれ、他のコンパイラ用にそれらを変更することが容易な筈です。I/Oとリンクファイルはこの応用記述に加えられているコード一式内に含まれています。

3.1. TWIドライバ関数

TWIドライバは呼び出した関数のためにI²Cバスに対する全ての通信を処理します。この関数についてのより多くの詳細に関してはコード内の注釈をご覧ください。

図3-1. TWIドライバ流れ図



3.1.1. char Init_TWI(void)

AVRのTWI単位部を初期設定します。AVR従装置アドレスを初期化し、通信用ビット速度を設定して単位部を許可します。

3.1.2. void Wait_TWI(void)

TWI割り込み要求フラグ(TWINT)が設定(1)になるまで繰り返し待機します。

3.1.3. unsigned char Send_start(void)

バス上に**開始条件**を送出してこの条件が終わるのを待ち、成功ならば\$FF、異常発生ならばTWSR値を返します。

3.1.4. void Send_stop(void)

バス上に**停止条件**を送出して直ちにに戻ります。

3.1.5. unsigned char Send_to_TWI(unsigned char *tx_frame, unsigned char *rx_frame)

全ての転送の動きを処理して**開始条件/停止条件**送出、データ送受信、アドレスバイト送受信のために他の関数を呼ぶ主ドライバ関数です。成功の場合に\$FF、異常の場合にTWSR値を返します。

3.1.6. unsigned char Send_byte(unsigned char data)

従装置へデータバイトを送信して**ACK/NACK**を待ちます。成功の場合に\$FF、異常の場合にTWSR値を返します。

3.1.7. unsigned char Send_adr(unsigned char adr)

バス上に従装置アドレスを送信して**ACK/NACK**を待ちます。成功の場合に\$FF、異常の場合にTWSR値を返します。

3.1.8. unsigned char Get_byte(unsigned char *rx_frame)

TWIフラグが設定(1)になるまで待ち、TWIデータレジスタを読み出します。読み込み値はSRAM内の'***rx_frame**'に保存されます。成功の場合に\$FF、異常の場合にTWSR値を返します。

3.1.9. void Reset_TWI(void) (訳注:実装されていません。)

TWI単位部をリセットしてバスを開放します。異常が起きた場合にバスを開放するのに使ってください。

3.2. LCD制御関数

LCD制御はLCD仕様に注意して構成設定し、データと制御のバイトをTWI主装置ドライバを通してLCDへ送ります。LCDを2行表示に構成設定し、先頭行でスクロール文字として文字列"AVR Atmega163 with TWI interface to I²C"を表示します。

3.2.1. char setup(void)

AVR用の構成設定関数です。

3.2.2. unsigned char Init_DSP(void)

表示器用の構成設定関数です。2行で各々40桁に構成設定します。

3.2.3. unsigned char Clear_DSP(void)

表示器はメモリ領域全体に\$20を書く内部表示消去機能を持ちます。この駆動器版では\$20が空白文字ではなく、この関数が\$20の代わりに表示器全体へ空白文字を書きます。

3.2.4. unsigned char Write_String1(void)

1行目位置0で始まる表示器へ文字列"AVR Atmega163 with TWI interface to I²C"を書きます。

3.2.5. unsigned char Shift_left(void)

可視表示メモリ部分を左に1つ移動してスクロール効果を作成します。0.5秒毎に1回移動します。

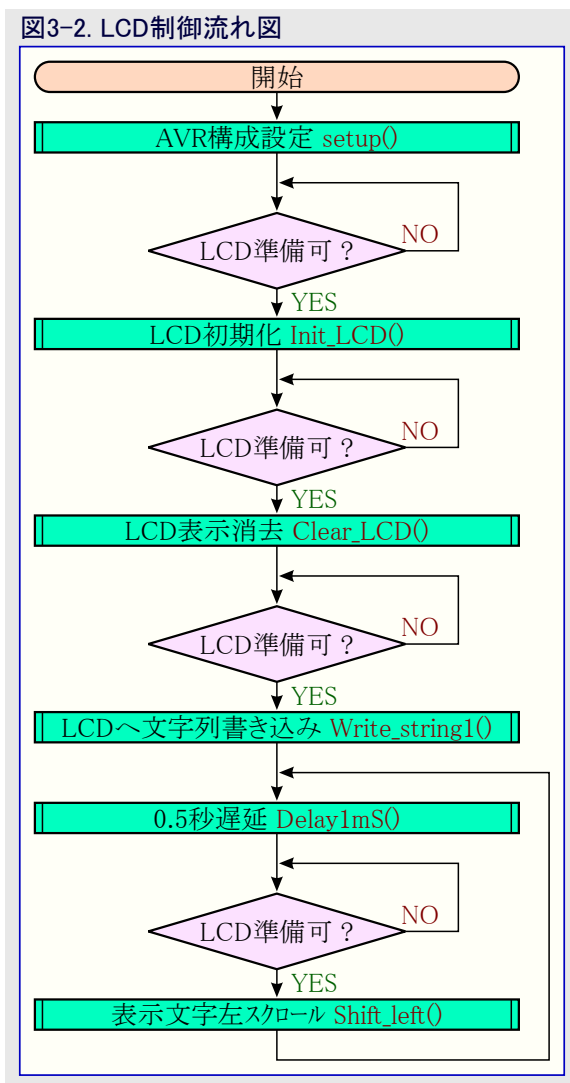
3.2.6. unsigned char Read_read_controlbyte(void)

多忙フラグと表示アドレス計数器を含む表示器読み込み制御バイトを読み出します。表示器が次の命令に対して多忙か準備可かを調べるのに使ってください。

3.2.7. void Delay1mS(char mS)

単純な遅延関数です。遅延はms遅延数を決める'mS'変数によって決められます。

図3-2. LCD制御流れ図



3.3. ハードウェア

LCDは図1-1.で示されるようにI²Cインターフェースを通してAVRへ接続されます。

4. 参照

- Philips 「I²Cバス仕様2.0版」 - (1998年12月)
- Philips 「PCF2119x LCD制御/駆動器」データシート - (1999年3月)
- Batron 「BT21605V-SRE-I2C-COG」 LCD規格化部品データシート
- Atmel 「ATmega163」データシート



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 2005. 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附属物です。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AVR155応用記述(doc1981.pdf改訂1981B-09/05)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。