AVR1618 : ATxmegaB ASCII文字割り当て

要点

- ATMEL® AVR® XMEGA® Bマイクロ コントローラで利用可能なASCII文字割り当て
- ATMEL XMEGA-B1 Xplain評価キット適合
- 基板上のLCD表示器
- 表示器での構造
  - 7セグメント数字
  - 14セグメント英数字

1. 序説

ATMEL AVR XMEGA Bデバイスの ASCII文字割り当て組み込みLCD制御器は対応するASCII符号書き込みによって簡単に7,14または16セグメント文字のピクセルの設定と解除を許します。

この応用記述はXMEGA-B1 Xplainキットを用いて7と14セグメント文字の操作を実演する例を提供します。

図1-1. XMEGA-B1 Xplainキット

2. 環境

実演はATMEL ATxmega128B1でのATMEL XMEGA-B1 Xplainキット(www.atmel.com/xplained)と専用のLED硝子の"C42048A"に基づきます。

ファームウェアは以下の例としてATMEL AVRソフトウェア枠組みとATMEL AVR Studio® 5で利用可能です。

- "LCD C42048A glass example - XMEGA-B1 Xplained - ATxmega128B1"
- "LCD controller example - XMEGA-B1 Xplained - ATxmega128B1"

表2-1. LCD定義と頭字語

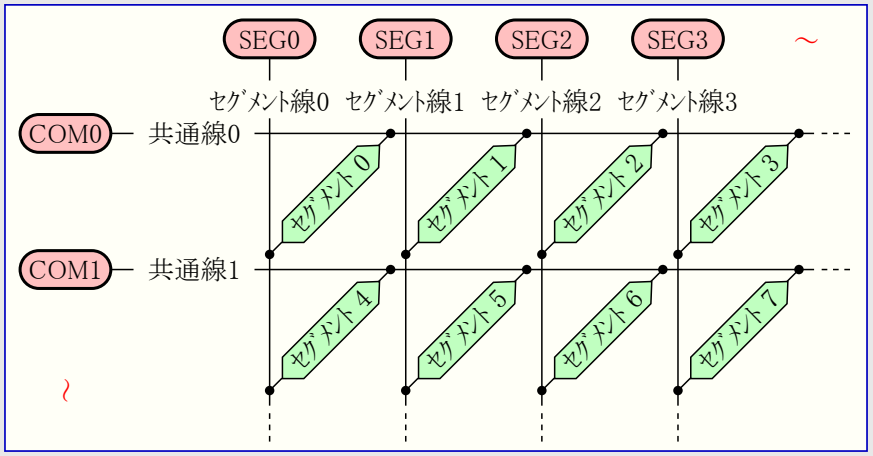
LCD	液晶表示器。直接セグメントに至る電極線を持つ受動表示パネル。
セグメントまたはピクセル	"ON"または"OFF"に変えることができるLCDパネルの活性領域。各セグメントは2つの線を持ちます。1つはSEG線に接続され、他方はCOM線に接続されます。全てのセグメントは配列で接続されます(図2-1.をご覧ください)。
COM	共通線または背面線
SEG	セグメント線
デューティ	1/(LCD表示器でのCOM線数)
バイアス	1/(LCD表示器を駆動するのに使用される電圧レベル数-1)
DP	小数点

8ビット ATMEL
マイクロ コントローラ

応用記述

Rev. 8451A-11/11, 8451AJ1-03/14

図2-1. LCD配列



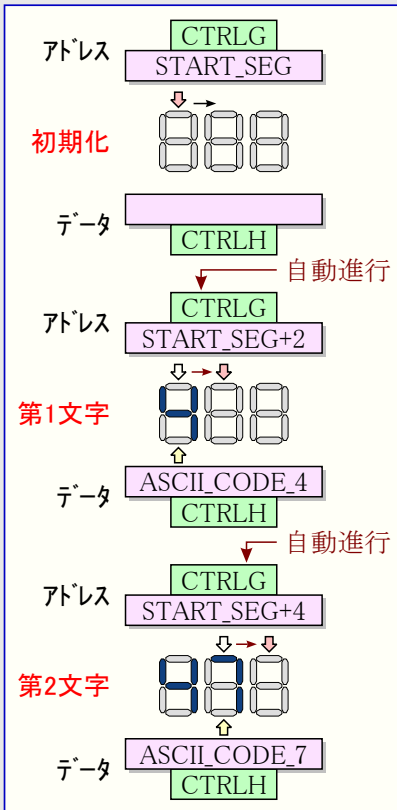
3. ASCII文字割り当て

3.1. 使い易さ

ATMEL ATxmega128B1 LCD制御器はASCII文字を自動的に扱うことができます。桁のセグメントを設定と解除する代わりに、使用者はASCII符号を入力し、桁復号部が表示メモリ内のセグメント値を更新します。

ASCII文字割り当てを始めるため、使用者はセグメント形式と、桁群の始まりのSEG線番号でのポインタ(LCD_CTRLG)を構成設定しなければなりません。その後これはFIFOとして働きます。使用者は制御レジスタ(LCD_CTRLH)内に最初の桁のASCII値を書きます。セグメントが更新されると、桁に使用されるSEG線数で自動的に増加(進行)(または減少(後退))されます。ポインタは今や新しいASCII書き込みに対する準備が整っています。

図3-1. ASCII割り当て



アセンブリ コード

```
LDI     temp, DIGIT_TYPE|START_SEG
STS     LCD_base+LCD_CTRLG_offset, temp
```

アセンブリ コード

```
LDI     temp, DIGIT_TYPE|START_SEG
STS     LCD_base+LCD_CTRLG_offset, temp

LDI     temp, ASCII_CODE_4
STS     LCD_base+LCD_CTRLH_offset, temp
```

アセンブリ コード

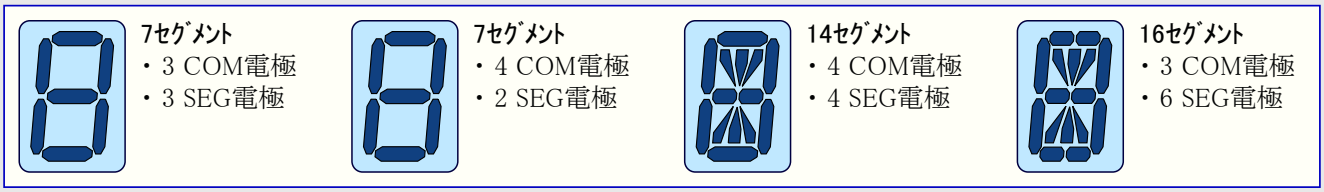
```
LDI     temp, DIGIT_TYPE|START_SEG
STS     LCD_base+LCD_CTRLG_offset, temp

LDI     temp, ASCII_CODE_7
STS     LCD_base+LCD_CTRLH_offset, temp

LDI     temp, ASCII_CODE_4
STS     LCD_base+LCD_CTRLH_offset, temp

～
```

図3-2. 支援される桁形式



3.2. LCD制御ﾚｼﾞｽﾀG

7	6	5	4	3	2	1	0	
TDG1,0		STSEG5~0						LCD_CTRLG

■ **TDG1,0 : 桁形式** (Type of Digit)

このﾋﾞｯﾄ領域は桁表示に使用されるｾｸﾞﾒﾝﾄ数と桁のCOM/SEG線数を指定します。

TDG1,0	桁形式
0 0	COM2~0/3つのSEGを持つ7ｾｸﾞﾒﾝﾄ
0 1	COM3~0/2つのSEGを持つ7ｾｸﾞﾒﾝﾄ
1 0	COM3~0/4つのSEGを持つ14ｾｸﾞﾒﾝﾄ
1 1	COM2~0/6つのSEGを持つ16ｾｸﾞﾒﾝﾄ

■ **STSEG5~0 : 開始ｾｸﾞﾒﾝﾄ** (Start Segment)

STSEGﾋﾞｯﾄ領域は桁を書くのに使用される最初のｾｸﾞﾒﾝﾄ線を定義します。このﾋﾞｯﾄ領域は桁で使用されるｾｸﾞﾒﾝﾄ線数により、(制御ﾚｼﾞｽﾀH(LCD.CTRLH)の減少(DEC)値に従って)自動的に増加(進行)または減少(後退)されます。

TDG1,0	桁形式	加減数
0 0	7ｾｸﾞﾒﾝﾄ - 3 COM/3 SEG	3
0 1	7ｾｸﾞﾒﾝﾄ - 4 COM/2 SEG	2
1 0	14ｾｸﾞﾒﾝﾄ - 4 COM/4 SEG	4
1 1	16ｾｸﾞﾒﾝﾄ - 3 COM/6 SEG	6

3.3. LCD制御ﾚｼﾞｽﾀH

7	6	5	4	3	2	1	0	
DEC		DCODE6~0						LCD_CTRLH

■ **DEC : 減少** (Decrement)

このﾋﾞｯﾄの1書き込みは桁で使用されるｾｸﾞﾒﾝﾄ線数によって制御ﾚｼﾞｽﾀG(LCD_CTRLG)の開始ｾｸﾞﾒﾝﾄ(STSEG)ﾋﾞｯﾄ領域を自動的に減少します。このﾋﾞｯﾄが0を書かれた場合、STSEGﾋﾞｯﾄ領域は桁で使用されるｾｸﾞﾒﾝﾄ線数によって増加されます。この活動は桁復号が一度終わると起こり、次の桁復号呼び出しの準備をします。

■ **DCODE6~0 : 表示符号** (Display Code)

DCODEﾋﾞｯﾄ領域は桁復号器によって計算され、表示符号に変換されて、その後に開始ｾｸﾞﾒﾝﾄ(STSEG)値に従って自動的に表示ﾒﾓﾘ内に書かれます。表の入口符号(DCODE6~0)は桁の7ﾋﾞｯﾄASCII符号です。

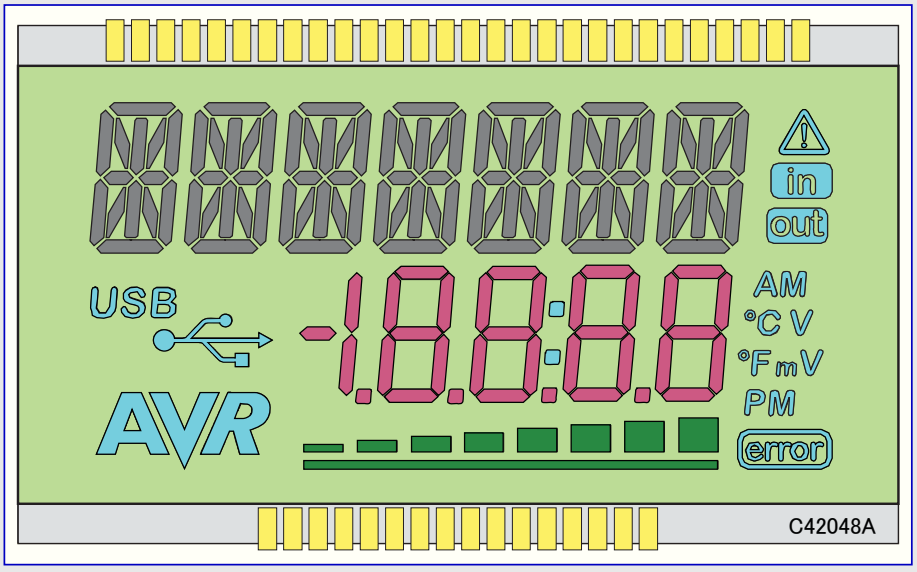
4. LCD C42048A硝子

LCD C42048AはATMEL XMEGA-B1 Xplainｷｯﾄ用に特に設計されました。それは3.0Vで動く4つのCOMと40のSEG線LED硝子で来て、以下のような4群のｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)を含みます。

- 7つの14ｾｸﾞﾒﾝﾄ英数字桁
- 負符号(-)と小数点(DP)を持つ5つの7ｾｸﾞﾒﾝﾄ数字桁
- 9ｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)の1つの棒ｸﾞﾗﾌ
- 1ｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)の13個のｱｲｺﾝ

合計: 155ｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)

図4-1. LCD C42047Aのｾｸﾞﾒﾝﾄ群



4.1. 14セグメント英数字桁

英数字桁はATMEL ATxmega128B1デバイスの桁復号部によって支援され、LCD制御レジスタG(LCD_CTRLG)で定義される、3つ目の桁形式(TDG1,0=2、4つのCOM線を持つ14セグメント)に合うように配線されています。

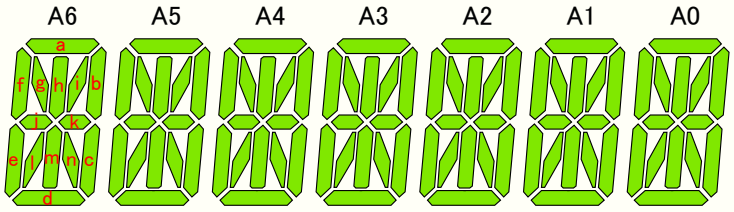
この割り当てはCOM0/SEG(n+1)とCOM3/SEG(n+2)の位置で桁復号部によって使用されない2つのピクセルを放置します。

英数字桁はA0からA6までと名付けられます。

- A6(左端の桁)はSEG12～15によって駆動されます。
- A5はSEG16～19によって駆動されます。
- ～
- A0(右端の桁)はSEG36～39によって駆動されます。

注: C文字列と同じ文字走査順(左⇒右)を保つため、開始セグメント(STSEG)は4で増加(DEC=0)されなければなりません。

図4-2. 英数字桁配線



COM	SEG																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
COM0													A6-h		A6-a	A6-g	A5-h		A5-a	A5-g
COM1													A6-i	A6-f	A6-b	A6-j	A5-i	A5-f	A5-b	A5-j
COM2													A6-k	A6-e	A6-c	A6-l	A5-k	A5-e	A5-c	A5-l
COM3													A6-n	A6-d		A6-m	A5-n	A5-d		A5-m

COM	SEG																			
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
COM0	A4-h		A4-a	A4-g	A3-h		A3-a	A3-g	A2-h		A2-a	A2-g	A1-h		A1-a	A1-g	A0-h		A0-a	A0-g
COM1	A4-i	A4-f	A4-b	A4-j	A3-i	A3-f	A3-b	A3-j	A2-i	A2-f	A2-b	A2-j	A1-i	A1-f	A1-b	A1-j	A0-i	A0-f	A0-b	A0-j
COM2	A4-k	A4-e	A4-c	A4-l	A3-k	A3-e	A3-c	A3-l	A2-k	A2-e	A2-c	A2-l	A1-k	A1-e	A1-c	A1-l	A0-k	A0-e	A0-c	A0-l
COM3	A4-n	A4-d		A4-m	A3-n	A3-d		A3-m	A2-n	A2-d		A2-m	A1-n	A1-d		A1-m	A0-n	A0-d		A0-m

4.2. 7セグメント数字桁

数字桁はATMEL ATxmega128B1デバイスの桁復号部によって支援され、LCD制御レジスタG(LCD_CTRLG)で定義される、2つ目の桁形式(TDG1,0=1、4つのCOM線を持つ7セグメント)に合うように配線されています。

この割り当てはCOM3/SEG(n+1)の位置で桁復号部によって使用されない1つのピクセルを放置します。

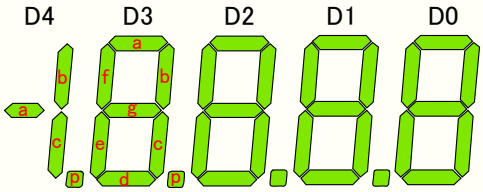
数字桁はD0からD3までと名付けられます。

- D3はSEG8,9によって駆動されます。
- D2はSEG6,7によって駆動されます。
- D1はSEG4,5によって駆動されます。
- D0はSEG2,3によって駆動されます。D0は小数点(DP)を持ちません。

注: C文字列と同じ文字走査順(左⇒右)を保つため、開始セグメント(STSEG)は2で減少(DEC=1)されなければなりません。

D4桁は特殊で、-, -1, 0, 1の4つの値だけを表示します。これは手動で駆動されます。D4の小数点(DP)はCOM3/SEG10に配置されます。

図4-3. 数字桁配線



COM	SEG																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
COM0			D0-a	D0-f	D1-a	D1-f	D2-a	D2-f	D3-a	D3-f	D4-a									
COM1			D0-b	D0-g	D1-b	D1-g	D2-b	D2-g	D3-b	D3-g	D4-b									
COM2			D0-c	D0-e	D1-c	D1-e	D2-c	D2-e	D3-c	D3-e	D4-c									
COM3				D0-d	D1-p	D1-d	D2-p	D2-d	D3-p	D3-d	D4-p									

COM	SEG																			
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
COM0																				
COM1																				
COM2																				
COM3																				

5. ソフトウェア実装

5.1. ソフトウェア ドライバ

5.1.1. ドライバのコード

```
/**
 * ¥brief LCD硝子へのASCII文字の流れ送出
 *
 * この関数は桁復号部経由でのLCDセグメント(ピクセル)を許します。
 * 関数は引数として渡された最大バイト数を書き、'NULL'文字が見つかった場合に書き込みを止めます。
 *
 * ¥param  lcd_tdg    復号する桁形式
 * ¥param  first_seg  最初のデータが書かれる最初のSEG
 * ¥param  data       データ緩衝部
 * ¥param  width      最大データ数
 * ¥param  dir        方向 (==0: 左←右, !=0: 左←右)
 */
void lcd_write_packet(enum LCD_TDG_enum lcd_tdg, uint8_t first_seg,
                      const uint8_t *data, size_t width, uint8_t dir)
{
    LCD.CTRLG = lcd_tdg | ((first_seg<<LCD_STSEG_gp) & LCD_STSEG_gm);
    if (dir != 0) {
        dir = LCD_DEC_bm;
    }
    while (width--) {
        if (*data == '¥0') {
            break;                          // NULL文字で停止
        }
        LCD.CTRLH = dir | (*data++);
        }
    }
}
```

5.1.2. ﾄﾞﾗｲﾊﾞ引数

・"lcd_tdg"

この引数はﾃﾞﾊﾞｲｽ ﾍｯﾀﾞ(GCCの場合に"iox128b1.h"、IAR™の場合に"ATxmega128B1.h")に含まれるLCD_TDG_enmの1つの要素を選ぶのに使用されます。

```
typedef enum LCD_TDG_enum {
        LCD_TDG_7S_3C_gc  = (0x00 << 6), /* 3つのCOMを持つ7ｾｸﾞﾒﾝﾄ */
        LCD_TDG_7S_4C_gc  = (0x01 << 6), /* 4つのCOMを持つ7ｾｸﾞﾒﾝﾄ */
        LCD_TDG_14S_4C_gc = (0x02 << 6), /* 4つのCOMを持つ14ｾｸﾞﾒﾝﾄ */
        LCD_TDG_16S_3C_gc = (0x03 << 6), /* 3つのCOMを持つ16ｾｸﾞﾒﾝﾄ */
        } LCD_TDG_t
```

・"first_seg"

最初に書く桁の最初のSEG線を定義します。
桁内側でSEG線の走査は増加されますが、桁外側でこれは桁の組を考慮することを意味し、走査順は"dir"引数によって定義されます。

・"*data"

表示するASCII文字列でのﾎﾟｲﾝﾀ。

・"width"

駆動する行の桁数または最大桁数。
"size_t"がANSI形式で"unsigned char"に等しいことに注意してください。
ﾄﾞﾗｲﾊﾞに於いて、null文字(ASCII文字列終了印)が見つかった場合、この値は中断(無効化)され得ます。

・"dir"

桁走査の方向を定義します。
この走査が"*data++"と同じなら、この引数は0と等しくなければなりません。
そうでなければ、この引数は0と異ならなければなりません。

5.1.3. C42048Aの14ｾｸﾞﾒﾝﾄ桁用引数

・"lcd_tdg"

LCD_TDG_14S_4C_gc (4本のCOM線を持つ14ｾｸﾞﾒﾝﾄ)

・"first_seg"

12 (**図4-2. 英数字桁配線**を参照してください。) これは左端A6桁の最初のSEG線に対応します。

・"*data"

lcd_text (例で定義されて使用される名称)

・"width"

7 (**図4-2. 英数字桁配線**を参照してください。)

・"dir"

0 (**図4-2. 英数字桁配線**を参照してください。) A6が左端桁で、"*data++"によって指定される最初の文字、"lcd_text[0]"を受け取ります。

5.1.4. C42048Aの7ｾｸﾞﾒﾝﾄ桁用引数

D4が手動で管理されるため、D3がこのﾄﾞﾗｲﾊﾞに対する左端になります。

・"lcd_tdg"

LCD_TDG_7S_4C_gc (4本のCOM線を持つ7ｾｸﾞﾒﾝﾄ)

・"first_seg"

8 (**図4-3. 数字桁配線**を参照してください。) これは左端D3桁の最初のSEG線に対応します。

・"*data"

lcd_num (例で定義されて使用される名称)

・"width"

4 (**図4-3. 数字桁配線**を参照してください。)

・"dir"

1 (**図4-3. 数字桁配線**を参照してください。) D3が左端桁で、"*data++"によって指定される最初の文字、"lcd_numdata[0]"を受け取ります。

5.1.5. 限界の制御

桁復号部それ自体がｱﾄﾞﾚｽ限界を調査します。自動増加/減少は最後にｱﾄﾞﾚｽ指定されたLCDﾃﾞｰﾀ(LCD_DATAn)ﾚｼﾞｽﾀや隣接ﾚｼﾞｽﾀを上書きしません。

5.2. 特殊化したC42048A用ﾏｸﾛ関数

5.2.1. 14ｾｸﾞﾒﾝﾄ桁用ﾏｸﾛ

```
/**
 * ¥brief C42048A LCD硝子の英数字領域へ文字列書き込み
 *
 * この関数はLCD硝子の英数字領域に入力文字列を書きます。
 *
 * ¥param data ﾃﾞｰﾀ入力文字列へのﾎﾟｲﾝﾀ
 */
static inline void c42048a_write_alpha_packet(const uint8_t *data)
{
    lcd_write_packet(LCD_TDG_14S_4C_gc, FIRST_14SEG_4C, data, WIDTH_14SEG_4C, DIR_14SEG_4C);
}
```

ここで、

以下はﾃﾞﾊﾞｲｽ ﾍｯﾀﾞで定義

・LCD_TFG_14S_4C_gc (5.1.2.ﾄﾞﾗｲﾊﾞ引数をご覧ください。)

以下は硝子部品ﾍｯﾀﾞで定義

・#define FIRST_14SEG_4C 12
・#define WIDTH_14SEG_4C 7
・#define DIR_14SEG_4C 0

5.2.2. 7ｾｸﾞﾒﾝﾄ桁用ﾏｸﾛ

```
/**
 * ¥brief C42048A LCD硝子の数字領域へ文字列書き込み
 *
 * この関数はLCD硝子の数字領域に入力文字列を書きます。
 *
 * ¥param data ﾃﾞｰﾀ入力文字列へのﾎﾟｲﾝﾀ
 */
static inline void c42048a_write_num_packet(const uint8_t *data)
{
    lcd_write_packet(LCD_TDG_7S_4C_gc, FIRST_7SEG_4C, data, WIDTH_7SEG_4C, DIR_7SEG_4C);
}
```

ここで、

以下はﾃﾞﾊﾞｲｽ ﾍｯﾀﾞで定義

・LCD_TFG_7S_4C_gc (5.1.2.ﾄﾞﾗｲﾊﾞ引数をご覧ください。)

以下は硝子部品ﾍｯﾀﾞで定義

・#define FIRST_7SEG_4C 8
・#define WIDTH_7SEG_4C 4
・#define DIR_7SEG_4C 1

6. 測定

ATMEL ATxmega128B1ﾃﾞﾊﾞｲｽの組み込み桁復号部の効率を測定するため、LCD.DATAnﾚｼﾞｽﾀに直接書き込む、桁ｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)の手動処理(設定または解除)のための特別なｿﾌﾄｳｪｱ ﾄﾞﾗｲﾊﾞを作成します。

この課題はLCD.CTRLGﾚｼﾞｽﾀで桁形式2(TDG1,0=2、LCD_TDG_14S_4C_gc)で"左⇒右"の方向だけに定義されるように配線された14ｾｸﾞﾒﾝﾄ桁に対してだけ行われます。

組み込み桁復号部が使用されないため、16ﾋﾞｯﾄ要素で自身のASCII表を定義します。最下位ﾆﾌﾞﾙはCOM0用に整えられたﾋﾞｯﾄの組で、同様に最上位ﾆﾌﾞﾙはCOM3用に整えられたﾋﾞｯﾄの組です。この表内で桁ｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)を得るのに使用される指標はASCII文字符号です。

6.1. 手動ﾄﾞﾗｲﾊﾞのｺｰﾄﾞ

```
// ATxmega128B1での40本のSEG線
#define   LCD_MAX_NBR_OF_SEG    40

// 14ｾｸﾞﾒﾝﾄ ASCII表
const uint16_t ascii_table[] = {
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x2E74, 0x0440, 0x23C4, 0x2544, 0x05E0, 0x25A4, 0x27A4, 0x0444,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x2365, 0x07E4, 0xA545, 0x2224, 0xA445, 0x22A4, 0x02A4, 0x2724,
      0x0000, 0x0000, 0x0000, 0x0000, 0x2220, 0x0678, 0x1668, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2660, 0x0000, 0x0000,
      0x1818, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0004, 0x07E4, 0xA545, 0x2224, 0xA445, 0x22A4, 0x02A4, 0x2724,
      0x0000, 0x0000, 0x0000, 0x0000, 0x2220, 0x0678, 0x1668, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2660, 0x0000, 0x0000,
      0x1818, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000 };

/**
 * ¥brief LCD硝子へのASCII文字の流れ送出
 *
 * この関数は14ｾｸﾞﾒﾝﾄASCII文字を表示するためにLCDｾｸﾞﾒﾝﾄ(ﾋﾟｸｾﾙ)を許可/禁止します。
 * 関数は引数として渡された最大ﾊﾞｲﾄ数を書き、'NULL'文字が見つかった場合に書き込みを止めます。
 *
 * ¥param  lcd_tdg     復号する桁形式
 * ¥param  data        ﾃﾞｰﾀ緩衝部
 * ¥param  width       最大ﾃﾞｰﾀ数
 */
void lcd_pix_write_packet(uint8_t first_seg, const uint8_t *data, uint8_t width)
{
register8_t  *pix_reg;
uint8_t      txt_i, com_i;
uint8_t      shift_todo,    temp_data;

    // ASCII文字の配列で繰り返し
    for (txt_i = 0; txt_i < width; txt_i++) {
        if (*data == '¥0') {
            break;                // NULL文字で停止
        }

        // どの位移動を行うか決定(0または4)
        shift_todo = ((first_seg + (4 * txt_i)) % 8);
        // 全COM線走査
        for (com_i = 0; com_i < 4; com_i++) {
            // 対応するﾃﾞｰﾀﾚｼﾞｽﾀのｱﾄﾞﾚｽ
            pix_reg = (register8_t*) ((uint16_t) &LCD.DATA0)
                    + (com_i * ((LCD_MAX_NBR_OF_SEG + 7) / 8))
                    + ((first_seg + (4 * txt_i))/ 8);
            // ﾃﾞｰﾀ取得
            temp_data = (uint8_t)
                        ((ascii_table[data[txt_i]] & (0x000F << (4 * com_i))) >> (4 * com_i));
            temp_data <<= shift_todo;
            // ﾃﾞｰﾀﾚｼﾞｽﾀ書き込み
            *pix_reg = (*pix_reg & ~(0x0F << shift_todo)) | temp_data;
        }
    }
}
```

6.2. 自動対手動の処理

6.2.1. ｺｰﾄﾞの大きさ

ｺﾝﾊﾟｲﾗ：GCC
ｺﾝﾊﾟｲﾗ任意選択：-Os

自動処理 = 手動処理の25%

- 自動処理：42ﾊﾞｲﾄ
- 手動処理：180ﾊﾞｲﾄ+ASCII表に対する256ﾊﾞｲﾄ

6.2.2. 実行時間

ｺﾝﾊﾟｲﾗ：GCC
ｺﾝﾊﾟｲﾗ任意選択：-Os
ｼｽﾃﾑ周波数：2MHz
桁数("width")：7

自動処理 = 手動処理の30倍高速

- 自動処理：59.705μs
- 手動処理：1817.315μs

6.2.3. 実行時間の式

単位：C周期
関数構成設定：ST
桁当たりの実行または構成設定：DG
桁数：n
COM当たりの実行：COM
COM数：y

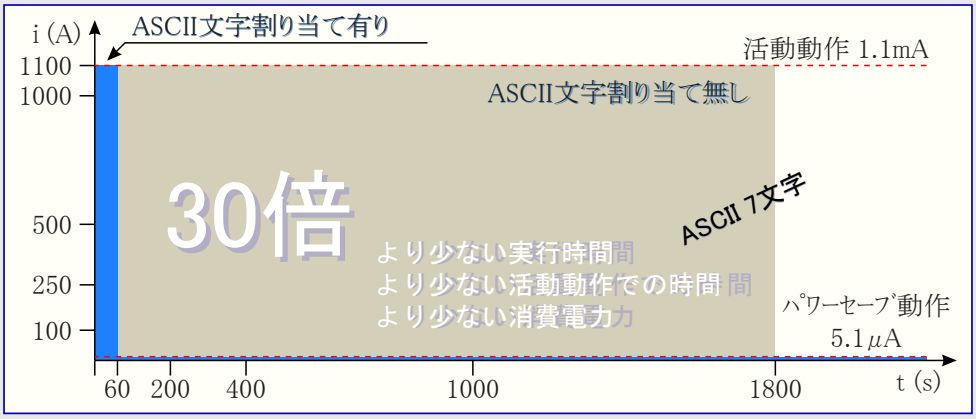
- 自動処理：ST=23, DG=14, COM=0
  ST + n×DG
- 手動処理：ST=89, DG=139, COM=92
  ST + n×DG + n×y×COM

6.2.4. 自動処理の恩恵

行うべき表示更新がなければ、ﾏｲｸﾛ ｺﾝﾄﾛｰﾗはﾊﾟﾜｰｾｰﾌﾞ動作へ移行することができます。活動動作では消費が代表的に1.1mAです。ﾊﾟﾜｰｾｰﾌﾞ動作では、LCDが走行している間で消費が接続したC42048A LCD硝子とで代表的に5.1μAです。

- より少ない実行時間
- より少ない活動動作での時間
- より少ない消費電力

図6-1. ASCII文字割り当ての恩恵



7. 推奨読み物

ATMEL AVR XMEGAについてと特にXMEGA Bﾃﾞﾊﾞｲｽについての全体的な考えを得るために以下の資料を読むことが推奨されます。

- XMEGA手引書とﾃﾞｰﾀｰｼｰﾄ
- AVR1000:XMEGAに対してCｺｰﾄﾞを書く前に
- AVR1005:XMEGAで始める前に
- AVR1010:XMEGAﾃﾞﾊﾞｲｽの電力消費最小化
- AVR1500:Xplain練習 – XMEGA 基礎
- AVR1912:ATMEL XMEGA-B1 Xplainﾊｰﾄﾞｳｪｱ使用者の手引き
- AVR1926:XMEGA-B1 Xplain開始の手引き

8. 目次

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
新大崎勧業ﾋﾞﾙ 16F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

本応用記述はATMELのAVR1618応用記述(doc8451.pdf Rev.8451A-11/11)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訳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。