AVR 8ﾋﾞｯﾄ ﾏｲｸﾛ ｺﾝﾄﾛｰﾗ

AVR42799 : XMEGA A1U Xplained ProでのGFXﾗｲﾌﾞﾗﾘ使用法

応用記述

序説

この応用記述は色彩豊または単色のLCD表示器に対して、釦、摺動子などのような入力を通して電子装置との相互作用を使用者に許す、文字に基づく使用者ｲﾝﾀｰﾌｪｰｽのGFXﾗｲﾌﾞﾗﾘを記述します。これはﾒﾆｭｰ ｼｽﾃﾑ、単色/色彩の図画ﾗｲﾌﾞﾗﾘ、回転ﾀﾞｲﾔﾙ/回転制御ｳｨｼﾞｪｯﾄ、単色/色彩ｼｽﾃﾑ ﾌｫﾝﾄを含むGUI対象物を提供します。

この応用記述はOLED1 Xplained Proを持つXMEGA® A1U Xplained ProでGFXﾗｲﾌﾞﾗﾘのGFX単色単位部を使用する方法を示すﾌｧｰﾑｳｪｱ例を提供します。この例はAtmel® | STARTからﾀﾞｳﾝﾛｰﾄﾞすることができます。

特徴

- GFXﾒﾆｭｰ例
- GFXｼｽﾃﾑ ﾌｫﾝﾄ例
- GFX図画ﾗｲﾌﾞﾗﾘ例
- 釦入力を持つGFX回転制御例

Rev. 42799A-11/2016, 42799AJ0-11/2016

目次

1. ハードウェア必要条件

GFXライブラリ例応用は以下のハードウェアが必要です。

図1-1. XMEGA A1U Xplained Pro評価キット

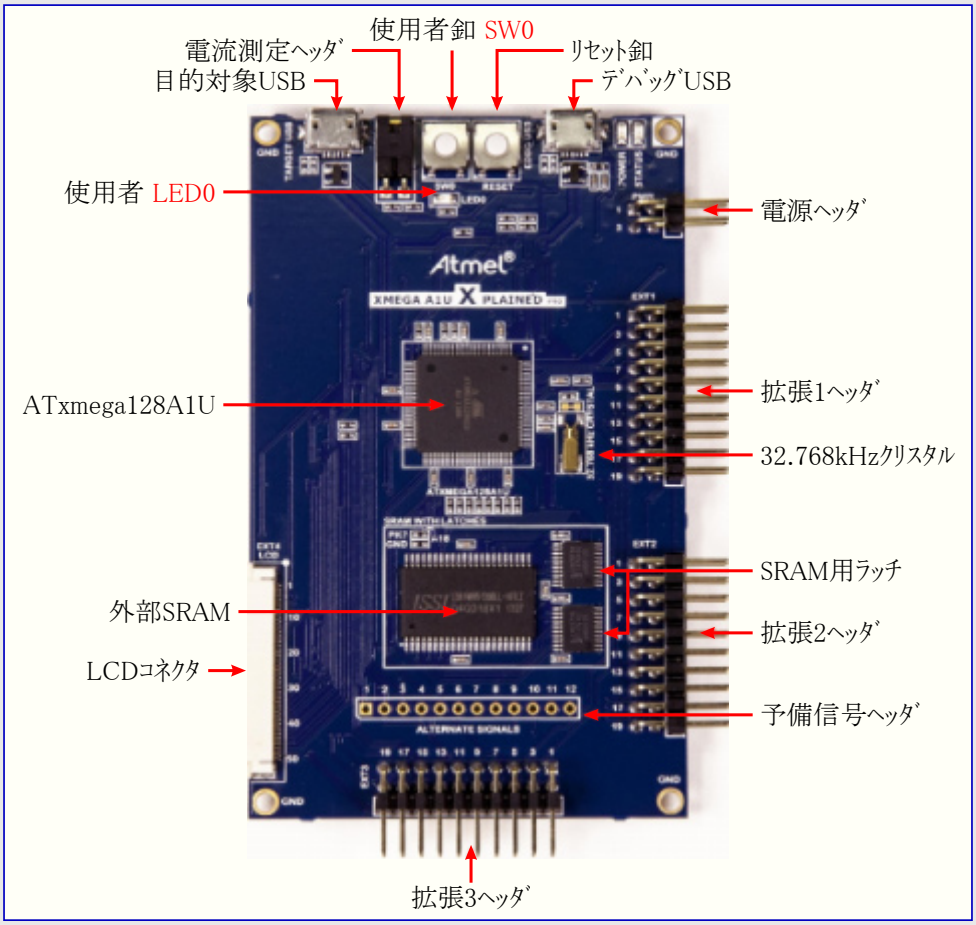
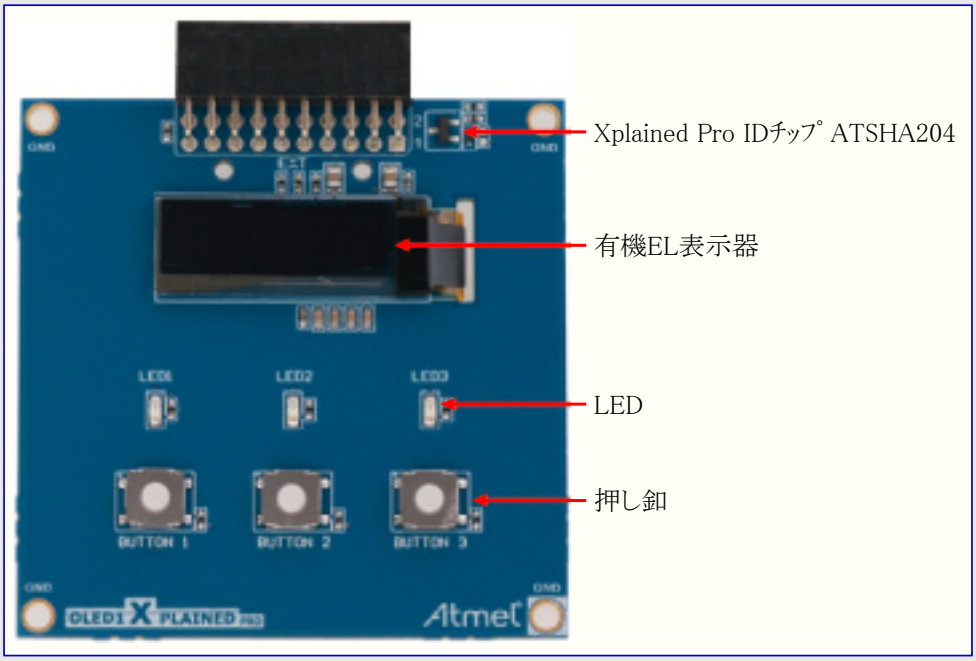


図1-2. OLED1 Xplained Pro



XMEGA A1U Xplained Pro評価ｷｯﾄは以下を含みます。

- Atmel ATxmega128A1Uﾏｲｸﾛ ｺﾝﾄﾛｰﾗ
- 1つの機械的なﾘｾｯﾄ釦
- 1つの機械的な使用者釦
- 1つの使用者黄色LED
- 32.768kHzｸﾘｽﾀﾙ
- 外部512Kﾊﾞｲﾄ SRAM
- 装置動作USBｲﾝﾀｰﾌｪｰｽ
- 3つのXplained Pro拡張ﾍｯﾀﾞ
- Xplained Pro LCD拡張ｺﾈｸﾀ
- 組み込みﾃﾞﾊﾞｯｶﾞ
- USB給電

OLED1 Xplained Proは以下を含みます。

- UG-2832HSWEG04単色有機EL表示器
  - 128×32ﾋﾟｸｾﾙ
  - 4線SPIｲﾝﾀｰﾌｪｰｽで最大100MHzによって制御
- 3つのLED
- 3つの機械的な押し釦
- Xplained Proﾊｰﾄﾞｳｪｱ識別ｼｽﾃﾑ

上のｷｯﾄのより多くの詳細については以下を参照してください。

XMEGA A1U Xplained Pro：http://www.atmel.com/tools/atxmegaa1u-xpro.aspxとXMEGA A1UXplained Pro使用者の手引き参照
OLED1 Xplained Pro評価ｷｯﾄ：http://www.atmel.com/tools/atoled1-xpro.aspxとOLED1Xplained Pro使用者の手引き参照

この例は他のﾃﾞﾊﾞｲｽへも容易に移植することができます。

2. 応用

ｿｰｽ ﾌﾟﾛｼﾞｪｸﾄは例を閲覧することによってAtmel | STARTからﾀﾞｳﾝﾛｰﾄﾞすることができます。Atmel STARTは組み込みｿﾌﾄｳｪｱ ﾌﾟﾛｼﾞｪｸﾄの直観的で図画的な形態設定用の革新的なｵﾝﾗｲﾝ ﾂｰﾙです。より多くの情報については始めの手引きを参照してください。

この例はGFXﾗｲﾌﾞﾗﾘの以下の4つの主な部分を示します。

- ｼｽﾃﾑ ﾌｫﾝﾄ
- 単色図画ﾗｲﾌﾞﾗﾘ
- ﾒﾆｭｰ ｼｽﾃﾑ
- 回転ﾀﾞｲﾔﾙ/回転制御ｳｨｼﾞｪｯﾄ

例ﾌﾟﾛｼﾞｪｸﾄはようこそﾒｯｾｰｼﾞを表示し、(長方形、円などの)図形を描き、ﾒﾆｭｰを示し。そして上の単位部の使い方を示すために、入力としてOLED Xplained Pro基板上の3つの釦で相互作用する回転ﾀﾞｲﾔﾙ集合を表示します。

GFX色彩(ｼｽﾃﾑ ﾌｫﾝﾄ/図画)単位部はOLED1 Xplained Proの単色有機EL上の例で実演されません。

GFXﾗｲﾌﾞﾗﾘを使うため、表示制御器だけでなくGFXﾗｲﾌﾞﾗﾘを初期化するのにgfx_mono_init()が呼ばれるべきです。

2.1. 表示器特定駆動部

表示器特定駆動部は図画的表示器へのｲﾝﾀｰﾌｪｰｽを提供します。それは表示器ﾊｰﾄﾞｳｪｱとの低位通信、表示器上にﾋﾟｸｾﾙを置いて線、円、矩形のような基本要素の描画を実行します。表示器駆動部実装に依存して、図画的基本要素の描画は表示器駆動部それ自身よりもむしろ標準的に図画描画基本要素によって処理されるかもしれません。

この例では、表示器特定駆動部がgfx_mono_ug_2832hsweg04.cとgfx_mono_ug_2832hsweg04.hで実装されます。

駆動部関数は以下で一覧にされます。

表2-1. 表示器特定駆動部関数

関数	説明
gfx_mono_ssd1306_draw_pixel()	画面にﾋﾟｸｾﾙを描画
gfx_mono_ssd1306_get_byte()	表示制御器のRAMからﾊﾞｲﾄ取得
gfx_mono_ssd1306_get_page()	LCD制御器からﾍﾟｰｼﾞ読み込み
gfx_mono_ssd1306_get_pixel()	x,yのﾋﾟｸｾﾙ取得
gfx_mono_ssd1306_init()	SSD1306とLCD表示器を初期化
gfx_mono_ssd1306_mask_byte()	表示制御器でﾊﾞｲﾄを読み/変更/書き
gfx_mono_ssd1306_put_byte()	表示制御器のRAMにﾊﾞｲﾄを置く
gfx_mono_ssd1306_put_framebuffer()	ﾌﾚｰﾑ緩衝部をLCD制御器へ置く
gfx_mono_ssd1306_put_page()	表示制御器へRAMからﾍﾟｰｼﾞを置く

gfx_mono_ug_2832hsweg04.hでは以下のようなﾏｸﾛによる抽象層定義があります。

```
#define gfx_mono_put_byte(page, column, data) ¥
    gfx_mono_ssd1306_put_byte(page, column, data, false)
#define gfx_mono_get_byte(page, column)    ¥
    gfx_mono_ssd1306_get_byte(page, column)
```

これらはGFXﾗｲﾌﾞﾗﾘに対するｲﾝﾀｰﾌｪｰｽです。

低位表示器駆動部はASF¥common¥components¥display¥ssd1306.h/cに置かれ、有機EL特定命令でのSPI通信はここで処理されます。先に言及された表示器特定駆動部は最終的に有機EL制御器をｱｸｾｽするのにこの低位駆動部を呼びます。

新規の表示装置でGFXﾗｲﾌﾞﾗﾘを使用するには、それによって低位駆動部が変更されるべきです。

2.2. GFXｼｽﾃﾑ ﾌｫﾝﾄ

GFXｼｽﾃﾑ ﾌｫﾝﾄは表示器で使用される低位の静的ｼｽﾃﾑ ﾌｫﾝﾄを提供します。これは限定された柔軟性を持つ軽量実装図形ﾌｫﾝﾄです。表示器へのｼｽﾃﾑ ﾒｯｾｰｼﾞ出力とﾃﾞﾊﾞｯｸﾞ出力に良好です。これは代表的にﾌｫﾝﾄの選択が重要でない、けれどもむしろﾌﾗｯｼｭﾒﾓﾘとRAMの量が少ない、地味な図画応用で使用されます。

ｼｽﾃﾑ ﾌｫﾝﾄ対象物は図画応用に対して全体的に利用可能です。ｼｽﾃﾑ ﾌｫﾝﾄは設計によって読み込み専用で、故に走行時にそれらを変更するどんな関数もありません。応用が追加のﾌｫﾝﾄを必要とする場合、追加ﾌｫﾝﾄ対象物を追加すべきです。

この例では以下を用いることによってﾒｯｾｰｼﾞを表示します。

```
gfx_mono_draw_string("My name is¥r¥nXMEGA-A1U Xplained!¥r¥nAnd I'm board...", 0, 0, &sysfont);
```

ﾌｫﾝﾄ文字はASF¥common¥components¥display¥ssd1306¥font.cで定義されます。

ｼｽﾃﾑ ﾌｫﾝﾄ関数は以下で一覧にされます。

表2-2. ｼｽﾃﾑ ﾌｫﾝﾄ関数

関数	説明
gfx_mono_draw_char()	RAMに置かれた文字を描画
gfx_mono_draw_string()	RAMに置かれた文字列を描画
gfx_mono_get_string_bounding_box()	RAMに置かれた文字枠を描画
gfx_mono_draw_progmem_string()	ﾌﾗｯｼｭ ﾒﾓﾘに置かれた文字列を描画
gfx_mono_get_progmem_string_bounding_box()	ﾌﾗｯｼｭ ﾒﾓﾘに置かれた文字枠を描画

2.3. GFX図画ﾗｲﾌﾞﾗﾘ

このﾗｲﾌﾞﾗﾘは単色図画表示器上に図画を描画するためのｲﾝﾀｰﾌｪｰｽを提供します。図画駆動部は以下から成ります。

- 表示器駆動部ｲﾝﾀｰﾌｪｰｽ (gfx_mono.h)
- 標準図画描画基本要素 (gfx_mono_generic.h)
- 表示器特定実装 (例:gfx_mono_ug_2832hsweg04.h)

標準描画基本要素は線、矩形、円のような図形基本要素を描くための関数のﾗｲﾌﾞﾗﾘです。これは基本要素を描くために表示器駆動部によって実装される他の関数を使用します。これらの関数の実装は表示器駆動部によって任意で使用することができますが、表示器のﾊｰﾄﾞｳｪｱがその基本要素のどれかのより高速な処理を許す場合、表示器駆動部はそれを直接実装することができます。

注: ﾗｲﾌﾞﾗﾘ内の関数は割り込み対応ではありません。

gfx_mono_generic.h/cで実装される標準図画ﾗｲﾌﾞﾗﾘ関数は以下で一覧にされます。

表2-3. GFX図画ﾗｲﾌﾞﾗﾘ関数

関数	説明
gfx_mono_generic_put_bitmap()	ﾌﾗｯｼｭ ﾒﾓﾘまたはRAMから表示器にﾋﾞｯﾄﾏｯﾌﾟを置く
gfx_mono_generic_draw_circle()	円または弧の外形を描画
gfx_mono_generic_draw_filled_circle()	塗り潰された円または扇形を描画
gfx_mono_generic_draw_filled_rect()	塗り潰された矩形を描画
gfx_mono_generic_draw_horizontal_line()	1ﾋﾟｸｾﾙ幅で水平線を描画
gfx_mono_generic_draw_line()	任意の2点間に線を描画
gfx_mono_generic_draw_rect()	矩形の外形を描画
gfx_mono_generic_draw_vertical_line()	1ﾋﾟｸｾﾙ幅で垂直線を描画

2.4. GFXメニュー システム

この単位部は単色図画表示器用の簡単なﾒﾆｭｰ ｼｽﾃﾑを提供します。

ﾒﾆｭｰ ｼｽﾃﾑを用いる応用の代表的な流れは次の通りです。

1. ﾒﾆｭｰ構造体を定義してください。
2. gfx_mono_menu_initを呼んでください。
3. 使用者入力を取得してください。
4. gfx_mono_menu_process_key関数を用いて使用者入力でﾒﾆｭｰを更新してください。
5. gfx_mono_menu_process_key戻り値を解釈してください。
6. 3.へ行ってください。

ﾒﾆｭｰは単色図画表示器構造体用のﾒﾆｭｰ ｼｽﾃﾑを使用して宣言されます。ﾒﾆｭｰ ｼｽﾃﾑを開始するには、gfx_mono_menu_init関数を呼んでください。この関数は表示器を一掃してﾒﾆｭｰを描きます。ﾒﾆｭｰが更新され得る前に、使用者からの入力が必要です。入力を得るための方法はﾒﾆｭｰ単位部の一部ではありません。入力が受け取られると直ぐに、gfx_mono_menu_process_key関数を使用してﾒﾆｭｰ ｼｽﾃﾑに知らせてください。この関数はその後に状態符号を返し、与えられたｷｰ符号に応じて働きます。

- MENU_KEYCODE_DOWN ：次のﾒﾆｭｰ項目(最後の場合は先頭)へ選択を変更。MENU_EVENT_IDLEを返します。
- MENU_KEYCODE_UP ：直前のﾒﾆｭｰ項目(先頭の場合は最後)へ選択を変更。MENU_EVENT_IDLEを返します。
- MENU_KEYCODE_ENTER ：ﾒﾆｭｰでの変更なし。選択された行を返します。
- MENU_KEYCODE_BACK ：ﾒﾆｭｰでの変更なし。MENU_EVENT_EXITを返します。

使用されるｷｰ符号の値はconf_menu.hで定義されます。これらの値は必要ならば変更することができます。ﾒﾆｭｰ選択を示すのに使用される図画指示子はconf_menu.hで定義されます。この指示子は必要ならば変更することができます。

gfx_mono_menu.h/cで実装されるﾒﾆｭｰ ｼｽﾃﾑ関数は以下で一覧にされます。

表2-4GFXﾒﾆｭｰ ｼｽﾃﾑ関数

関数	説明
gfx_mono_menu_init()	ﾒﾆｭｰ処理初期化。画面一掃とﾒﾆｭｰ描画。
gfx_mono_menu_process_key()	入力に応じてﾒﾆｭｰ更新

2.5. GFX回転ﾀﾞｲﾔﾙ/回転制御ｳｨｼﾞｪｯﾄ

この単位部は単色図画表示器用の回転ﾀﾞｲﾔﾙ ｳｨｼﾞｪｯﾄを提供します。画面上の1つの単一回転ﾀﾞｲﾔﾙまたは回転ﾀﾞｲﾔﾙの集合に対する支援があります。

回転ﾀﾞｲﾔﾙは上または下のどちらかの矢印上をｸﾘｯｸすることによってﾃｷｽﾄ ﾎﾞｯｸｽ内の値に(上矢印が押し続けられる場合に)増加を、または(下矢印が押し続けられる場合に)減少を引き起こして隣のﾃｷｽﾄ ﾎﾞｯｸｽ内の値を使用者が調整し得る図画制御要素です。

回転集合ｼｽﾃﾑを用いる応用の代表的な流れは次の通りです。

1. 回転ﾀﾞｲﾔﾙを定義してください。
2. gfx_mono_spinctrl_initで回転ﾀﾞｲﾔﾙを初期化してください。
3. 回転集合構造体を定義してgfx_mono_spinctrl_spincollection_initでそれぞ初期化してください。
4. gfx_mono_spinctrl_spincollection_add_spinnerで回転ﾀﾞｲﾔﾙを回転集合に追加してください。
5. gfx_mono_spinctrl_spincollection_showで画面に回転集合を描いてください。
6. 回転ﾀﾞｲﾔﾙ選択を格納するために結果配列を定義してください。
7. gfx_mono_spinctrl_spincollection_process_key関数を用いて使用者入力で回転ﾀﾞｲﾔﾙと結果配列を更新してください。
8. gfx_mono_spinctrl_spincollection_process_key戻り値を解釈してください。
9. 7.へ行ってください。

回転ﾀﾞｲﾔﾙが更新され得る前に、使用者からの入力が必要です。入力を得るための方法は回転ﾀﾞｲﾔﾙ ｳｨｼﾞｪｯﾄの一部ではありません。

注: 回転ﾀﾞｲﾔﾙは回転集合に追加される時に共に繋げられ、従って同時に2つの回転集合で使用することはできません。

入力が受け取られると直ぐに、gfx_mono_spinctrl_spincollection_process_key関数またはgfx_mono_spinctrl_process_key関数を使用して回転集合または単一回転ﾀﾞｲﾔﾙに知らせてください。これら関数はその後に状態符号を返し、与えられたｷｰ符号に応じて働きます。

- GFX_MONO_SPINCTRL_KEYCODE_DOWN ：選択を次の回転ﾀﾞｲﾔﾙ値か次の回転ﾀﾞｲﾔﾙ、または回転集合のOK釦へ変更
- GFX_MONO_SPINCTRL_KEYCODE_UP ：選択を前の回転ﾀﾞｲﾔﾙ値か前の回転ﾀﾞｲﾔﾙ、または回転集合のOK釦へ変更
- GFX_MONO_SPINCTRL_KEYCODE_ENTER ：回転ﾀﾞｲﾔﾙ値か回転ﾀﾞｲﾔﾙ、または回転集合のOK釦を選択
- GFX_MONO_SPINCTRL_KEYCODE_BACK ：回転ﾀﾞｲﾔﾙ選択解除、または回転集合応用取り消し

使用されるｷｰ符号の値はconf_spinctrl.h.で定義されます。これらの値は必要ならば変更することができます。回転ﾀﾞｲﾔﾙ選択を示すのに使用される図画指示子はconf_spinctrl.h.で定義されます。この指示子は必要ならば変更することができます。

この例では入力方法として3つの釦が使用されます。釦1は上釦として使用され、釦2は下釦として使用され、釦3はOK釦として使用されます。これらの釦押下によって、各々GFX_MONO_SPINCTRL_KEYCODE_UP, GFX_MONO_SPINCTRL_KEYCODE_DOWN, GFX_MONO_SPINCTRL_KEYCODE_ENTERが返ります。

回転ﾀﾞｲﾔﾙに対して誘導矢印が指示する時のOK釦押下は回転ﾀﾞｲﾔﾙ値選択へ移行します。上または下の釦押下によって、値を変更することができます。この例では、1つの文字列回転ﾀﾞｲﾔﾙと2つの整数回転ﾀﾞｲﾔﾙが実演されます。全ての回転ﾀﾞｲﾔﾙ値が選択された後、矢印を”OK”へ移動してOK(釦3)を押し、その後に選択された値が表示されます。

3. 改訂履歴

資料改訂	日付	注釈
42799A	2016年11月	初版資料公開

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA TEL:(+1)(408) 441-0311 FAX: (+1)(408) 436-4200 | www.atmel.com

本応用記述はAtmelのAVR42799応用記述(Rev.42799A-11/2016)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。