

AVR482 : DB101ソフトウェア使用者の手引き

要点

- 7つのドライバ
- 12個のライブラリ
- 9つの応用

1. 序説

DB101ファームウェアは使用者に対して応用一式を実装するために、多数のドライバとライブラリを使用する複合した1つのソフトウェアです。この資料は全てのドライバ、ライブラリ、応用に対して簡単な紹介を与えます。詳細な資料についてはDoxygen自動資料化ツールによって生成されたソースコードの資料を参照してください。

ドライバとライブラリの多くがDB101ハードウェア用に特に設計されているとは言え、それらの殆どは他の応用へも直接的に使用、または容易に移すことができます。ソースコードの資料は何れかの特定ハードウェアに関連する実装方法についてのより多くの情報を提供します。また、AVR481応用記述はDB101ハードウェアを詳細に記述します。

この資料はドライバ、ライブラリ、DB101実演応用を別々に網羅する3つの章に分けられます。

図1-1. DB101始動画面



2. ドライバ

ドライバは内蔵UARTと外部LCD制御器のように、内部と外部の或るハードウェアに対してインターフェースを提供するソフトウェア部です。DB101ソフトウェア一式は以下のドライバを含みます。

表2-1. ドライバ ファイル名

ドライバ名	ソース ファイル	ヘッダ ファイル
背面照明ドライバ	backlight_driver.c	backlight_driver.h
ジョイスティックドライバ	joystick_driver.c	joystick_driver.h
LCD制御器ドライバ	s6b1713_driver.c	s6b1713_driver.h
電力管理ドライバ	power_driver.c	power_driver.h
実時間時計ドライバ	rtc_driver.c	rtc_driver.h
音ドライバ	sound_driver.c	sound_driver.h
UARTドライバ	uart_driver.c	uart_driver.h

2.1. 背面照明ドライバ

背面照明ドライバはDB101上のLCD単位部に対してRGBバックライトを制御するためのインターフェースを提供します。このドライバは背面照明LEDに接続されたPWM出力の制御を隠して、背面照明の全体輝度を制御する共通輝度値に加えて、赤、緑、青の水準を個別に設定する関数を提供します。輝度0設定は個別色に対する設定に拘らず背面照明をOFFに切り替えます。

2.2. ジョイスティックドライバ

ジョイスティックドライバはDB101上の5方向ジョイスティックに対するインターフェースを提供します。このドライバは基礎を成すポーリング動作とジョイスティックの“鉤”のチャタリング除去を隠して、現在のジョイスティックの状態を読む、ダブルクリック感知と押下保持条件用の遅延設定変更、登録事象処理部呼び戻し関数に対する関数を提供します。

事象処理部がジョイスティック処理部と組み合わせられると、ジョイスティックの状態が変化(クリック、解放、ダブルクリック、押下保持状態)する時に必ず直ぐに通知されることが可能です。事象処理部の使用で、使用者応用に関してジョイスティックの状態を手動で継続的にポーリングする必要はありません。



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. 8094A-09/07, 8094AJ2-12/13

全ての応用がジョイスティックに対して5つの割り込み入力を割り当てる余裕があるとは限らないため、ジョイスティックドライバは割り込み駆動ではありません。従って、ジョイスティック入力が一時間隔で読まれるポーリングの仕組みをジョイスティックドライバは実装します。そしてジョイスティックの状態はそれらの一定間隔から更新されます。可能な限り柔軟なドライバを作成するために、使用者にはJOYSTICK_PollingHandlerと名付けられたポーリングルーチンと呼ぶ責任があります。殆どの場合でこれは計時器を用いて実装されるでしょう。DB101ファームウェアに関して、これは実時間ドライバ(2.5.をご覧ください)によって順番に駆動されるタイミングライブラリ(3.11.をご覧ください)を用いて実装されます。

ポーリング動作に加えて、ドライバは起動形態へ入ることもできます。これは応用がアイドルで消費電力低減を望む時に使用されます。ドライバは何れかの動きでどの休止形態からもAVRを起すように形態設定された1つの外部割り込みを用いて、ジョイスティックの入出力線を改めて形態設定します。起動時、ドライバは再び自動的にポーリング動作形態に移行します。

2.3. LCD制御器ドライバ

LCD制御器ドライバはLCD単位部のS6B1713制御チップに対するインターフェースを提供します。このドライバはAVRとLCD制御器間の通信インターフェースの初期化を処理します。ドライバはLCD制御器で支援される全命令を反映する関数も持ちます。例えばS6B1713チップが直列と並列の両インターフェースを支援するとは言え、このドライバに対する既定実装はAVRのデータ空間にS6B1713チップの命令レジスタとデータレジスタが割り当てられる、データ空間メモリ割り当てインターフェースを使用します。これは最高の通信速度を持つインターフェースです。メモリ割り当てインターフェースがどう実装されるかのより多くの詳細についてはソースコードの資料を参照してください。

このドライバがどの図形機能も提供しないことに注意してください。全てのS6B1713命令に対して1対1のインターフェースを提供するだけです。LCD図形ライブラリ(3.4.項)はピクセル制御、線や円のような高位図形機能を実装するのにこのドライバを使用します。

2.4. 電力管理ドライバ

電力管理ドライバは電力管理とAVRマイクロコントローラの休止形態に関するインターフェースを提供します。このドライバは休止形態に対する準備の詳細と移行を隠します。

より深い休止形態、例えばパワーセーブ動作形態が使用される時に、いくつかの割り込みは未だ動作中です。けれども、使用者は起動を望まず、実時間時計の計時毎に主応用を再開するかもしれません。時計ドライバは計時を処理して再びパワーセーブ休止動作形態へ移行すべきです。選択した条件でだけ主応用の再開を使用者に許すために、ドライバは休止動作形態再移行の代わりに主応用再開を電力管理部に告げる、POWER_WakeupCallbackと名付けられた呼び戻し関数を提供します。DB101ファームウェアは時計の毎計時後ではなく、ジョイスティックの動きで主応用を再開するのにこの機能を使用します。

ドライバが電力削減形態(PRR)内の全周辺機能と共にAVRマイクロコントローラを初期化することに注意してください。これは必要な時に使用者が個別周辺機能に対して電力削減形態を手動で禁止しなければならないことを意味します。代表的には、周辺機能に関するドライバがあなたのためにこれを処理するでしょう。

2.5. 実時間時計ドライバ

実時間時計ドライバは、外部の時計用32.768kHzクリスタルで走り続ける非同期計時器に関するインターフェースを提供します。このドライバは秒、分、時、日の経緯を保ちます。また、1秒は非同期計時器割り込みが呼ばれる頻度に対応する計時数に分けられます。既定設定はジョイスティックドライバ(2.2.項)のポーリング処理部に適合する計時速度の1/128秒毎に溢れるように計時器を設定することです。

只時間の経緯を保つ以外の他の目的にこの計時を使用するため、使用者は毎計時後に呼ばれる関数の計時処理部を登録することができます。DB101ファームウェアでは、タイミングライブラリ(3.11.項)の主処理ルーチンが実時間時計ドライバの計時処理部として登録されます。そしてタイミングライブラリはジョイスティックドライバのポーリングルーチンと呼び出して処理します。

ドライバで日処理部を登録することも可能です。日処理部は日計数器が増加される毎に呼ばれ、高級な暦機能実装に使用することができます。

2.6. 音ドライバ

音ドライバはDB101の基板上圧電ブザーに関するインターフェースを提供します。

2.7. UARTドライバ

UARTドライバはUART周辺機能の1つに関するインターフェースを提供します。DB101ファームウェアに関する既定実装はDB101基板の外部UART接続に接続されるU(S)ART1部署をインターフェースすることです。

ドライバは割り込み制御された受信、FIFO待ち行列ライブラリ(3.2.項)からのFIFO待ち行列を通した送信、それと勿論UARTホーレトやパリティ形態などに対する基本的な機能を提供します。

ドライバのソースコードは1つのUARTだけを支援するように設計され、これは走行時にUART部署を変更できないことを意味します。この制限はドライバを非常に速度効率的にしますが、柔軟性を無くします。DB101基板に関しては、他のUART部署が基板上のDataflash®メモリデバイスに対してSPI主装置動作で使用されるため、これは問題ではありません。

3. ライブラリ

ライブラリはデータ待ち行列や図形のような特別な目的に対する関数を提供するソフトウェア部署です。ライブラリは代表的にドライバや他のライブラリを使用します。DB101ソフトウェア一式は右のライブラリを含みます。

表3-1. ライブラリ ファイル名

ライブラリ名	ソース ファイル	ヘッダ ファイル
ダイアログ ボックス ライブラリ	dialog_lib.c	dialog_lib.h
FIFO待ち行列ライブラリ	fifo_lib.c	fifo_lib.h
表現形式ライブラリ	forms_lib.c	forms_lib.h
LCD図形ライブラリ	lcd_lib.c gfx_lib.c	lcd_lib.h gfx_lib.h
メモリ塊ライブラリ	memblock_lib.c	memblock_lib.h
図画ライブラリ	picture_lib.c	picture_lib.h
ポップアップ ボックス ライブラリ	popup_lib.c	popup_lib.h
楽曲ライブラリ	song_lib.c	song_lib.h
端末偽装ライブラリ	terminal_lib.c	terminal_lib.h
端末フォント ライブラリ	termfont_lib.c	termfont_lib.h
タイミングと計画のライブラリ	timing_lib.c	timing_lib.h
ウィジェット ライブラリ	widgets_lib.c widgets_lib_*.c	widgets_lib.h

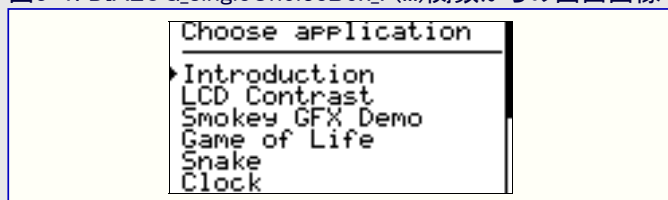
3.1. ダイアログ ボックス ライブラリ

ダイアログ ボックス ライブラリは一般的にメッセージ ボックスや検査一覧表ボックスのようなダイアログ ボックスを使用する表示と操作に対する関数を提供します。殆どの機能はダイアログに関する表題、ダイアログ ボックス内容用の文字列の一覧、表示のための釦選択(OK, CANCELなど)、予め割り当てられたメモリへの位置指示子を受け入れます。

ダイアログ ボックス ライブラリはそれ自身にどんなメモリも割り当てず、故に使用者はライブラリ関数に対する作業領域としてライブラリに空きメモリを提供しなければなりません。

ライブラリはダイアログ ボックスを表示するためにウィジェット ライブラリ(3.12.項)と表現形式ライブラリ(3.3.項)を、使用者入力を得るためにFIFO待ち行列ライブラリ(3.2.項)を通してジョイスティック ドライバ(2.2.項)を使用します。図3-1はDIALOG_SingleChoiceBox_F(...)関数によって生成されたダイアログ ボックスからの画面画像を示します。

図3-1. DIALOG_singleChoiceBox_F(...)関数からの画面画像



3.2. FIFO待ち行列ライブラリ

FIFO待ち行列ライブラリはバイト志向の先入り先出し待ち行列を管理するための関数を提供します。FIFO待ち行列ライブラリはそれ自身にメモリを割り当てず、故に使用者は新しいFIFO待ち行列が初期化される時に空きのメモリ緩衝部を提供しなければなりません。そしてFIFO待ち行列は何れかの量の標準と独自のデータ形式に対してPutとGetのルーチンと共に循環緩衝部として使用されます。

FIFO待ち行列は内部的にバイト志向で、これは内部緩衝部位置指示子がメモリ緩衝部の最後から先頭へ巻き戻る場合に大きな塊のデータが分離しないことを保証しないことに注意してください。けれども、バイト志向の待ち行列であることで、使用者は待ち行列に於いて1つの形式だけに制限されません。大きな注意が払われるなら、待ち行列はどんなデータ形式の混合も含むことができます。

3.3. 表現形式ライブラリ

表現形式ライブラリは図形的な使用者インターフェース表現形式の構築と操作に対する関数を提供します。これはチェック ボックス、ラベル、釦のような要素から構築するための基本的な構造体を提供します。ウィジェット ライブラリ(3.12.項)のような他のライブラリは複雑な図形的ウィジェットを構築するためにこれらの基本構造体を使用します。

例えば表現形式ライブラリがそれ自身に完全なウィジェットを提供しないとは言え、大きな形式内に要素を積み重ね、代表的にジョイスティックドライバ(2.2.項)からやって来る誘導事象からその形式を操作するための完全なエンジンを提供します。このエンジンは形式内の要素を通して閲覧するための誘導事象の処理を行い、現在注目(選択)されている要素へ事象を供給し、注目(選択)を変更してLCD上の更新領域に対して再描画要求を発行します。LCD画面よりも後ろの形式に対して垂直スクロールを効率的に支援するために、LCD制御器のハードウェア スクロール機能も利用可能です。

3.4. LCD図形ライブラリ

LCD図形ライブラリは基本的なピクセル操作、線、箱、円などに対する関数を提供します。これはLCD単位部上での直接操作の代わりに画面外SRAMフレーム緩衝部上での操作に対する一式の関数も提供します。

LCD画面は標準5×7ピクセルフォントに適合する、8ピクセル高のページに分けられます。この特徴はLCD図形ライブラリによって利用され、そしてこれは端末フォントライブラリ(3.10.項)や表現形式ライブラリ(3.3.項)のようなページ志向応用に関して効率的な操作のためのページ志向関数一式を提供します。

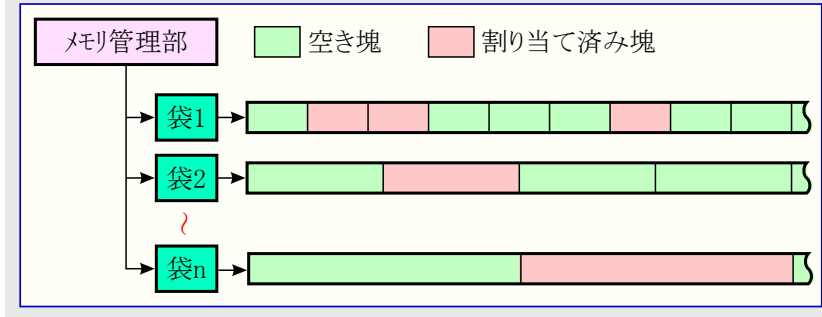
実際にライブラリは2つのソースファイルとそれに伴うヘッダファイルに分けられています。lcd_lib.cとlcd_lib.hのファイルは基本的なピクセル操作関数と、ページ志向関数と水平と垂直の線のような、LCD画面のページ志向の性質に対して最適化ができるその他の図形関数を提供します。gfx_lib.cとgfx_lib.hのファイルは円や任意の線のように、ピクセル操作関数を使用する高位図形関数を提供します。このような図形関数はLCD画面のページ志向の性質を利用することができません。

3.5. メモリ塊ライブラリ

メモリ塊ライブラリは断片化なしのメモリ管理部を用いる動的メモリ割り当てに対する関数を提供します。初期化に於いて、メモリ管理部は大きな塊のメモリを得て袋の数に分け、そしてそれは或る大きさに作られたメモリ塊に等しい数に順番に分けられます。図3-2.はメモリ管理部とそのメモリ塊の概要を示します。

袋の数と、各袋に対する塊容量と塊数は特定応用のメモリ必要条件に併せるように微調整することができます。微調整処理を手助けするため、メモリ管理部は最小と最大の塊容量要求と最高点と平均の利用で、各袋に対するメモリ使用での統計を保持します。

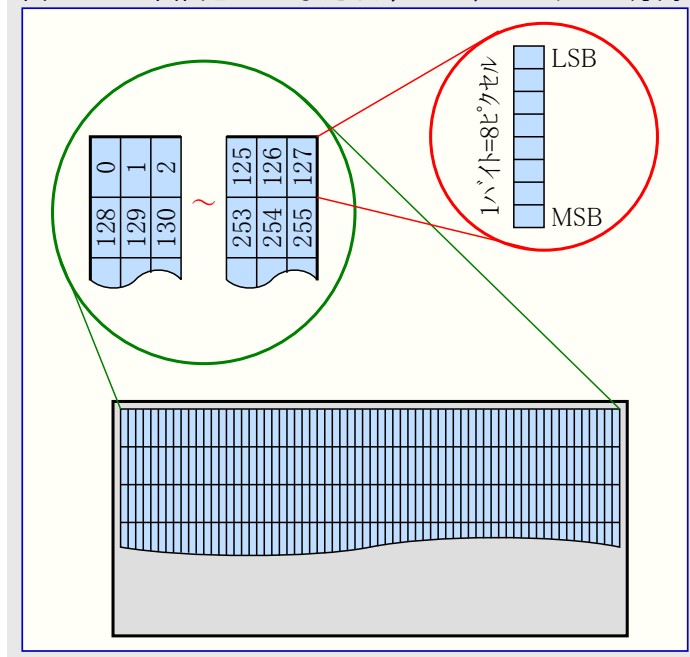
図3-2. メモリ管理部概要



3.6. 図画ライブラリ

図画ライブラリはLCD単位部とSRAMまたはフラッシュメモリ間で画像または前後の画像の区画を複写するための関数を提供します。画像は先頭バイトが左上列の8ピクセルで第2バイトがその右側の8ピクセルになる、LCD制御器自身によって使用されるのと同じページ志向の形式で格納されます。同じ理由のため、画像の位置指示時の垂直分解能は8ピクセルです。図3-3.はピクセルがメモリ内のバイトとビットにどう対応するかを示します。

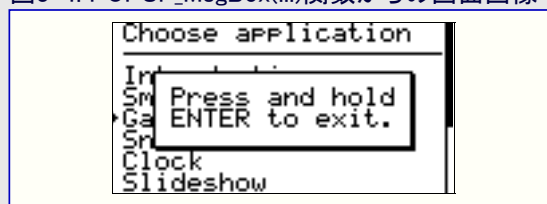
図3-3. LCD画面とSRAMまたはフラッシュメモリでのピクセルの方向



3.7. ホップアップ ボックス ライブラリ

ポップアップ ボックス ライブラリは他の何れかの図形上にポップアップ メッセージを表示し、ジョイスティックを動かしてその後には抜け出す使用者を持つための関数を提供します。任意で、ポップアップ ライブラリは直前の画面の形式を再描画することができます。ライブラリは画面上でボックスを中央にするように扱い、多数の行に渡って文を包み込みます。図3-4はPOPUP_MsgBox(...)関数によって生成されたダイアログ ボックスからの画面画像を示します。

図3-4. POPUP_MsgBox(...)関数からの画面画像



3.8. 楽曲ライブラリ

楽曲ライブラリは音ドライブ(2.6.項)を用いてフラッシュ メモリまたはSRAMから音楽を演奏する関数を提供します。

3.9. 端末偽装ライブラリ

端末偽装ライブラリはLCD上で全画面21×8文字端末に於いて文字を表示して、制御の符号と手順を解析するための関数を提供します。これはカーソル移動、選択領域消去、下線や反転色のような文字属性を含む、VT100端末の制御符号とエスケープ手順の広範囲に渡る部分一式を支援します。文字背景色制御も実装されますが、背面照明LEDがLCD全体を網羅するため、LCD全体に対して背面照明色が変わります。

ライブラリは画面上で1度に1つだけが活性である限り、同時に多数の操作上の端末を支援します。やって来るデータはデータ供給元としてどれかの通信インターフェースを許可する、TERMINAL_ProcessChar関数を用いて端末へ渡されます。応用それ自体は使用者へ情報を提供するために、まさに便利な出力装置として端末を使用できます。

3.10. 端末フォント ライブラリ

端末フォント ライブラリはSRAMとフラッシュ メモリからLCDまたは画面外フレーム緩衝部へ個別文字と文字列を出力するための関数を提供します。ライブラリはページ志向で広範囲に受け入れた5×7ピクセル端末フォントを使用します。このライブラリは平文を表示するために他の全てのライブラリによって使用される基本ライブラリです。

3.11. タイミングと計画のライブラリ

タイミングと計画のライブラリは周期的または単発の時間事象のために他の型式が予約することができる、タイミングと下部構造を計画するための関数を提供します。時間粒度は計時(tick)として参照され、代表的に実時間時計ドライブ(2.5.項)からの計時(tick)周期に対応します。DB10 1ファームウェアはこのライブラリのTIMING_TickHandler関数を実時間時計ドライブ用の計時処理部として登録します。

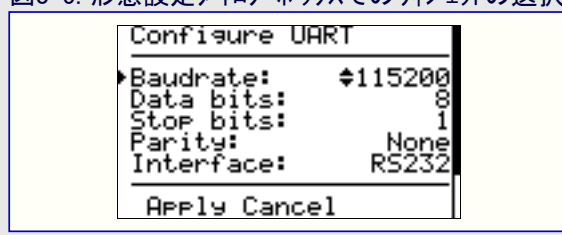
タイミング事象は呼び戻しまたは計数器更新のどちらかとして実装することができます。呼び戻し事象では、タイミング事象発生時に関数が呼ばれます。計数器更新については、タイミング事象が起こる度毎に1回増加される計数器変数へ使用者が位置指示子を登録します。

3.12. ウィジェット ライブラリ

ウィジェット ライブラリは表現形式ライブラリ(3.3.項)から基本表現形式要素構造体を用いて、ウィジェットと呼ばれる複雑な図形使用者インターフェース コントロールを構築するための関数を提供します。ウィジェットはあなたの応用に対して複雑な相互作用と訴える表現形式を構築するのに順番に使用することができます。右の図3-5はダイアログ ボックスに対して表題行を作るために使用される静止文字(StaticText)ウィジェットと分離子(Separator)ウィジェットを示します。ボックスはパラメータのために多数の回転制御(SpinCtrl)ウィジェットと何れかの変更の適用または取り消しを使用者に許すための釦で釦列(ButtonRow)ウィジェットを含みます。

このライブラリのwidgets_lib.cソース ファイルは個別のウィジェットを順番に実装する、第2のwidgets_lib.*.cソース ファイルをインクルードするための#include擬似命令を使用する置き換え場所だけです。この方法ではソース コードが多くの管理可能なコードの部分に分解されます。

図3-5. 形態設定ダイアログ ボックスでのウィジェットの選択



4. 応用

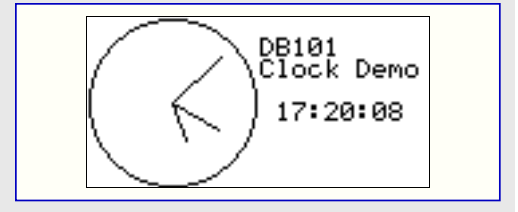
DB101応用ファームウェアは主始動メニューから利用可能な以下の応用を含みます。

応用名	ソース ファイル
アナログ時計 (Analog Clock)	clock.c
コンウェイのライフゲーム実演 (Conway's "Game of Life" Demo)	gameoflife.c
LCD濃淡 (LCD Contrast)	lcdcontrast.c
スライドショー (Slideshow)	slideshow.c
"煙" 図画実演 ("Smokey" Graphics Demo)	smokeydemo.c
"蛇" ゲーム ("Snake" Game)	snake.c
音実演 (Sound Demo)	sounddemo.c
端末エミュレータ (Terminal Emulator)	terminal.c
UART形態設定 (UART Configurator)	configuart.c
"仮想大画面" 画像表示 ("Walkabout" Picture Viewer)	walkabout.c

4.1. アナログ時計応用

アナログ時計応用は画面の左側で3針アナログ時計を、そして画面の右側でデジタル版を用いて現在時刻を表示します。例え時計応用が動いていなくても、実時間時計ドライバ(2.5.項)が現在時刻を維持します。ジョイスティックの上下移動または中央(Enter)押下が現在時刻を補正します。右の図4-1.は時計応用の主画面を示します。

図4-1. アナログ時計応用の主画面

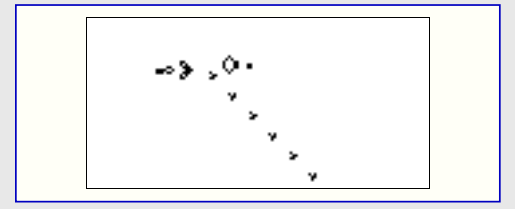


4.2. コンウェイの"ライフゲーム"実演応用

この実演応用は"ライフゲーム"セルラー自動進行法の変形の"ゴスパーのグライダー銃"の実装です。右の図4-2.は実演応用走行中の画面画像を示します。算法のより多くの詳細については以下のURLを調べ上げてください。

http://en.wikipedia.org/wiki/Conway's_Game_of_Life

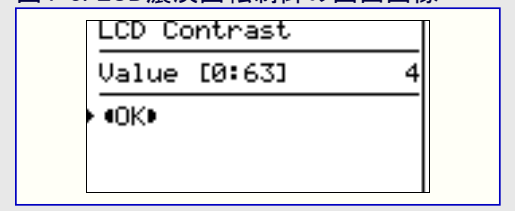
図4-2. ライフゲーム実演応用からの画面画像



4.3. LCD濃淡

このメニューはLCDの濃淡補正の可能性を与えます。これは0~63間で補正することができ、既定値は4です。

図4-3. LCD濃淡回転制御の画面画像



4.4. スライドショー応用

スライドショー応用はLCD画面に一連の画像を表示することによってタイミングと計画のライブラリ(3.11.項)と図画ライブラリ(3.6.項)を実演します。ジョイスティックの移動は直前または直後の画像へ移動します。けれども、触らないままなら、応用は一定の間隔で画像も変えます。右の図4-4.はDB101ファームウェアのようこそ画面像でも現れる画像の1つを示します。

図4-4. スライドショー応用での画像の1つ

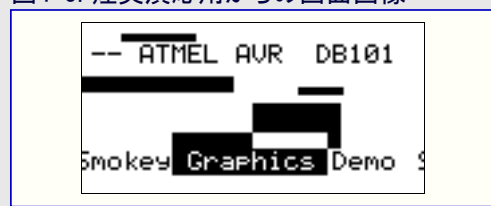


4.5. “煙” 図画実演応用

“煙”実演はLCD画面で素早い図画更新を実行するためにLCD図画ライブラリ(3.4.項)の能力を実演します。実演は壁で弾む多数の任意の大きさの矩形を表示し、絵的に相互作用し合い、加えて画面を横切るいくつかのスクロール文を表示します。

画像的な更新頻度が高いので、LCDパネル自体の持続(残像)性が重要な要素になります。これは図画がしみのように現れそしてしぼむようにし、故に“煙”の名称です。

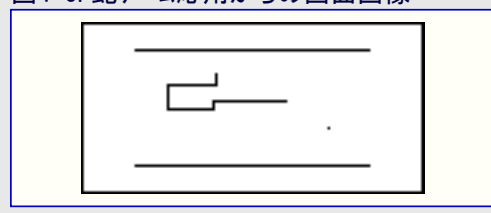
図4-5. 煙実演応用からの画面画像



4.6. “蛇”ゲーム応用

蛇ゲームは伝統的な蛇ゲームの簡単な実装です。使用者は食物を探して画面上の迷路を通して蛇を誘導します。食物の断片に達すると、蛇がそれを食べて数ピクセル成長して速度が増し、そして壁への衝突や食べ過ぎを避けるのをもっと難しくします。右の図4-6.はゲーム走行中からの画面画像を示します。

図4-6. 蛇ゲーム応用からの画面画像



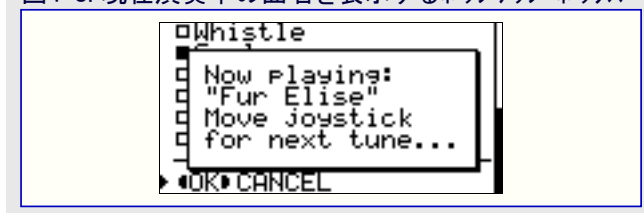
4.7. 音実演応用

音実演応用は使用者が演奏する曲を選べるジュークボックスのようなダイアログボックスを提供することにより、音ドライバ(2.6.項)と楽曲ライブラリ(3.8.項)を実演します。“OK”がクリックされると、応用はジョイスティックが移動された時に曲を変更して、かわるがわる曲を演奏します。下の図4-7.は曲選択全体を示し、図4-8.は曲を演奏している最中の応用画面を示します。

図4-7. 音実演応用での完全なメニュー画面



図4-8. 現在演奏中の曲名を表示するポップアップボックス



4.8. 端末エミュレータ応用

端末エミュレータ応用はVT100型端末でデータを表示するためのデータ供給元としてUARTを使用します。この応用は使用者へ素早い出力を提供する必要があるどの応用でも使用することができます。DB101基板をあなたのコンピュータのRS-232通信ポートへ接続して、RS-232ポートで出力を生成するどれかの応用を開始してLCD内容を見守ってください。右の図4-9.は接続したコンピュータからデータを受信する端末応用からの画面画像を示します。

図4-9. 端末エミュレータ応用からの画面画像

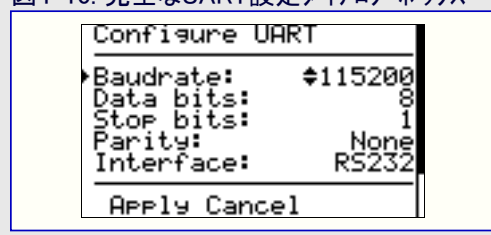


4.9. UART形態設定応用

UART形態設定応用はボーレートやパリティの形態のようなUART通信インターフェースの全ての項目の形態設定を使用者に許すダイアログボックスを表示します。この応用は表現形式、ウィジェット、ダイアログのライブラリの能力も実演します。右の図4-10.はダイアログボックスを示します。

端末エミュレータ応用(4.8.項)が始めて動く前にUARTが形態設定されなければならないことに注意してください。

図4-10. 完全なUART設定ダイアログボックス



4.10. “仮想大画面”画像表示応用

画像表示応用はもっと大きな画像(この場合はDB101基板自身の画像)で画面の大きさの領域を表示することによって図画ライブラリ(3.6.項)の能力を実演します。応用はどの方向へも画像をスクロールして動かすのに、ジョイスティックの使用を使用者に許します。右の図4-11.は応用走行中からの画面画像として示します。

図4-11. 仮想大画面画像表示応用からの画面画像





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

技術支援

avr@atmel.com

販売窓口

www.atmel.com/contacts

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 2007. 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AVR482応用記述(doc8094.pdf Rev.8094A-09/07)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。