



不可能への挑戦

株式会社日昇テクノロジー

低価格、高品質が不可能？

日昇テクノロジーなら可能にする

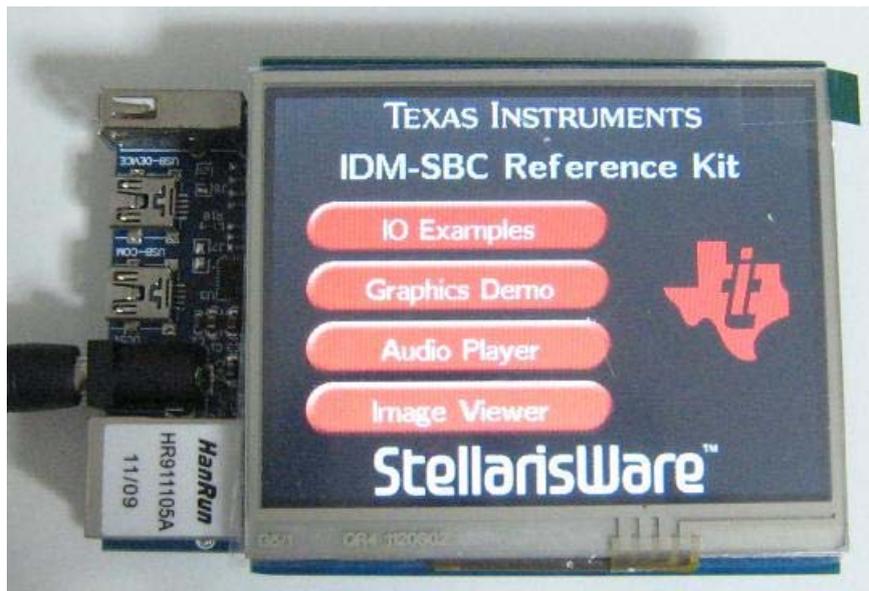
ARM Cortex-M3 LM3S9B92 開発キットマニュアル

株式会社日昇テクノロジー

<http://www.csun.co.jp>

info@csun.co.jp

2011/12/21



copyright©2011-2012

• 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2011/11/10
2	Ver1.1	MCU バージョンアップ (C1->C5)	2011/12/21

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。最新版は弊社ホームページからご参照ください。

「<http://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に禁じられています。



目次

第一章 LM3S9B92 開発キット概要.....	5
1.1 LM3S9B92 マイコンの主な特徴.....	5
1.2 基板ボード仕様.....	5
1.3 拡張ボード (LCD 搭載) 仕様.....	6
第二章 回路の説明.....	7
2.1 Power.....	7
2.2 USB Host.....	8
2.3 USB Device.....	9
2.4 USB COM.....	9
2.5 CAN.....	10
2.6 IIC EEPROM.....	10
2.7 SPI Flash.....	11
2.8 SD Card.....	11
2.9 LED.....	11
2.10 Key.....	12
2.11 JTAG.....	12
2.12 EXTEND I/O.....	13
第三章 開発ツール KEIL の応用.....	14
3.1 KEIL のインストール.....	14
3.2 環境の設定.....	16
3.2.1 ダウンロード関連.....	16
3.2.2 デバッグ関連.....	20
第四章 サンプルソースについて.....	22
4.1 Code¥StellarisWare¥boards¥Smart-9B92-C3¥.....	22
4.1.1 ¥blinky.....	24
4.1.2 ¥uart_echo.....	24
4.1.3 ¥softuart_echo.....	25
4.1.4 ¥timers.....	25
4.1.5 ¥watchdog.....	26
4.1.6 ¥sd_card.....	26
4.1.7 ¥usb_dev_serial.....	28
4.1.8 ¥usb_dev_msc.....	29



4.1.9	¥usb_dev_mouse.....	29
4.1.10	¥usb_dev_keyboard.....	30
4.1.11	¥usb_dev_bulk.....	30
4.1.12	¥usb_dev_chidcdc.....	31
4.1.13	¥usb_host_mouse.....	32
4.1.14	¥usb_host_keyboard.....	33
4.1.15	¥usb_host_msc.....	34
4.1.16	¥enet_uip.....	35
4.1.17	¥enet_lwip.....	37
4.1.18	¥enet_ptpd.....	38
4.1.19	¥qs-blox.....	39
4.1.20	¥idm-checkout.....	41
4.1.21	¥hello_lcd.....	49
4.1.22	¥hello_widget.....	50
4.1.23	¥calibrate.....	50
4.1.24	¥scribble.....	52
4.1.25	¥lang_demo.....	53
4.1.26	¥showjpeg.....	53
4.1.27	¥i2s_demo.....	54
4.2	Code¥ek-1m3s9b92-C3¥RL¥TCPnet¥.....	54
4.2.1	¥Http_demo.....	55
4.2.2	¥LEDSwitch.....	58
4.2.3	¥DNS_demo.....	59
4.2.4	¥Telnet_demo.....	60
付録	: OpenLink で実行ファイルの書き込む.....	63
1	ドライバのインストール.....	63
2	J-FLASH ARM で実行ファイルを書き込む.....	63



第一章 LM3S9B92 開発キット概要

ARM コア新型プロセッサCortex-M3 を採用した TI 社の LM3S9B92 (100MHz 周波数、256KB Flash、96KB SRAM)。

標準外付け：JTAG/SWD、Ethernet、CAN 2.0、USB2.0 OTG/Host/Device、USBシリアル (TTL) 変換、SDIO、EEPROM、SPI、TFT LCDなど。

1.1 LM3S9B92 マイコンの主な特徴

- ARM コア新型プロセッサCortex-M3 を採用、最大周波数100MHz
- 256KB Flash、96KB SRAM 内蔵
- 最大 65 個の GPIO
- 10/100M Ethernet MAC 及び PHY
- CAN 2.0 A/B
- USB 2.0 OTG/Host/Device
- 2 UARTs、ISO7816 サポート
- 2 SPI
- 2 IIC
- 1 IIS
- 8 PWM出力
- 2 10-bit A/D
- 8/16/32ビット専用パラレルバス、SDRAM、SRAM、Flash、FPGAs、CPLDをサポート
- 関数ライブを内蔵

1.2 基板ボード仕様

- JTAG/SWD デバッグ用インタフェース (10pin、2.54mmピッチ)
- 10/100M Ethernetインタフェース (DP83848)
- USB2.0デバイスインタフェース (OTG)
- USB2.0ホストインタフェース
- USBシリアルTTL変換インタフェース (CP2102)
- CAN2.0インタフェース x 1
- SDカードインタフェース
- シリアルEEPROM (24C04)
- SPIシリアルFLASH (W25Q80)
- TTL LCDインタフェース
- 外部バスインタフェース
- ユーザーボタン x 1
- ユーザーLED x 2

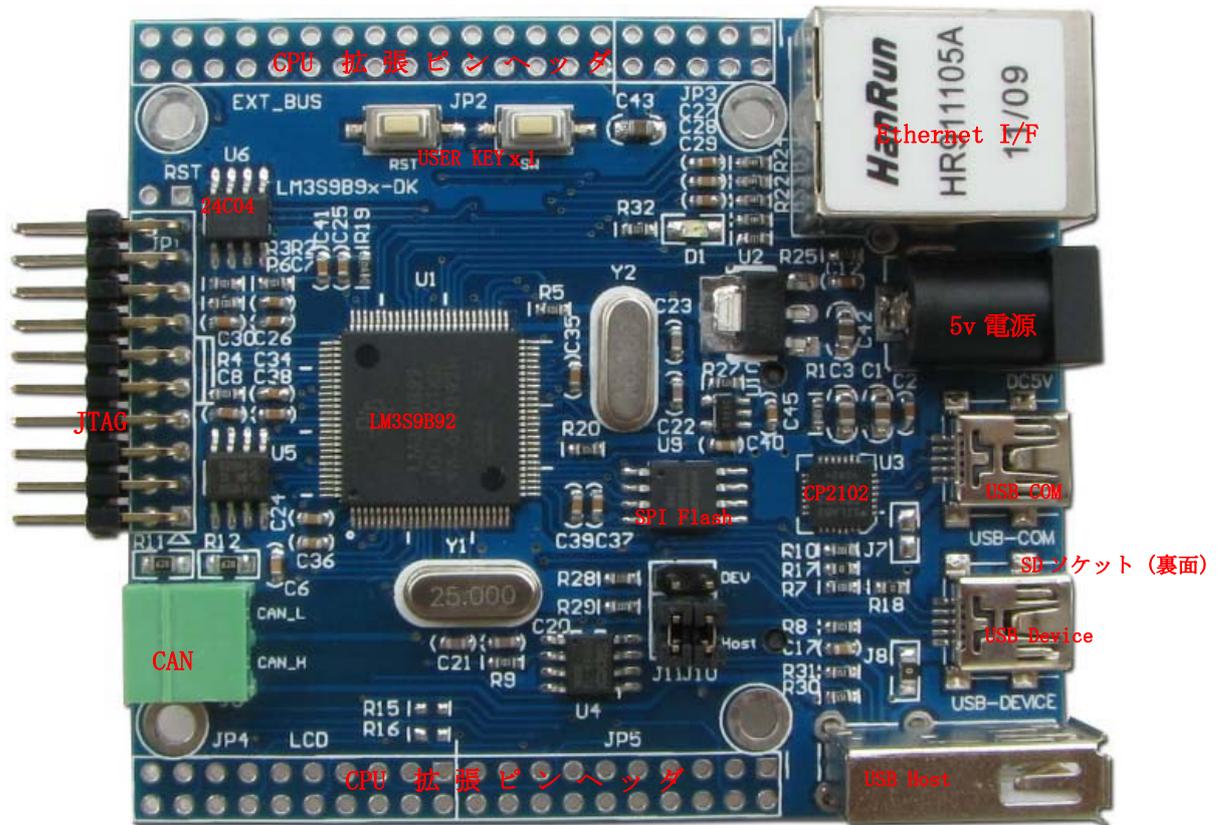


- 5V/1A DC外部電源インタフェース、極性：センタープラス
- 外形寸法：74×64(mm) ※突起物は除く

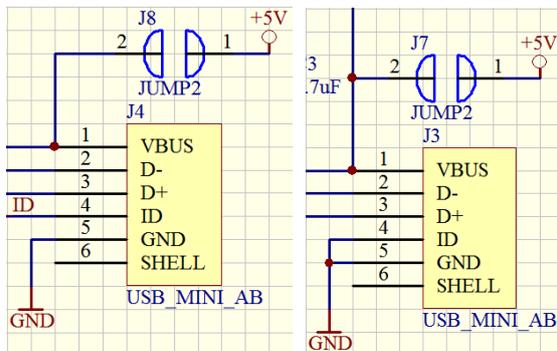
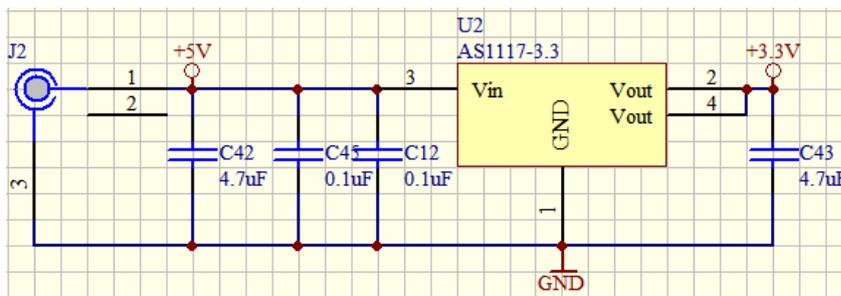
1.3 拡張ボード (LCD搭載) 仕様

- タッチパネル付きLCD搭載
- Audioプレイ機能搭載(WM8510)、イヤホン搭載
- 8MB SDRAM(MT48LC4M16A2P-7E)
- 外形寸法：77×69(mm) ※突起物は除く

第二章 回路の説明



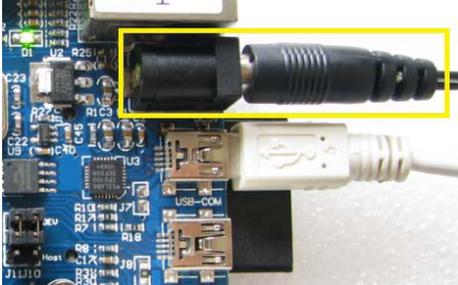
2.1 Power



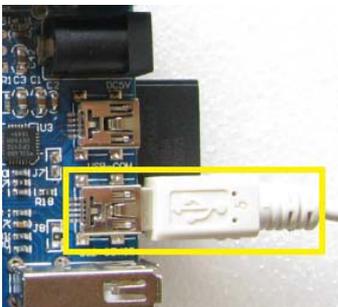
入力：5V DC

出力：3.3V DC

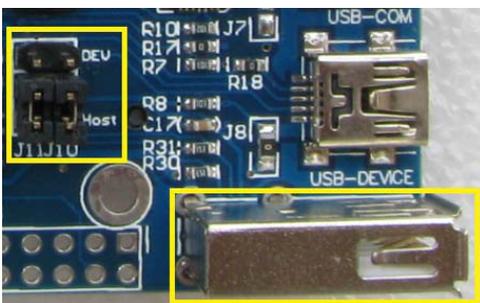
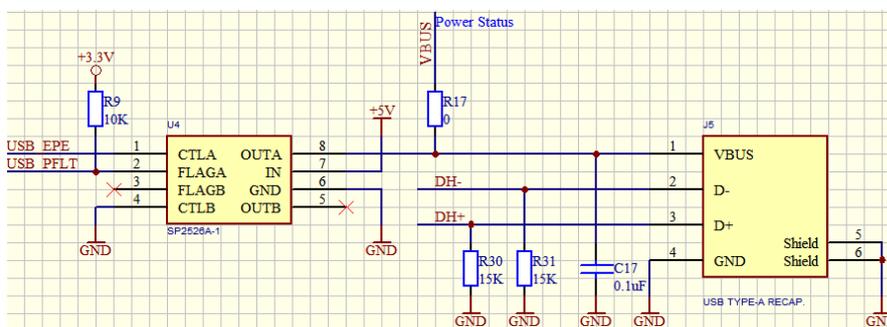
本ボードは J2 の DC2.1 で給電できます。



或いは J7、J8 の設定で Mini USB より給電できます。J8 はデフォルトでショートしております。USB ケーブルで “USB-DEVICE” の Mini USB からボードに給電できます。

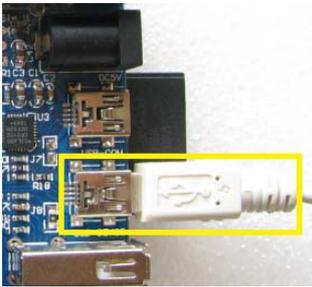
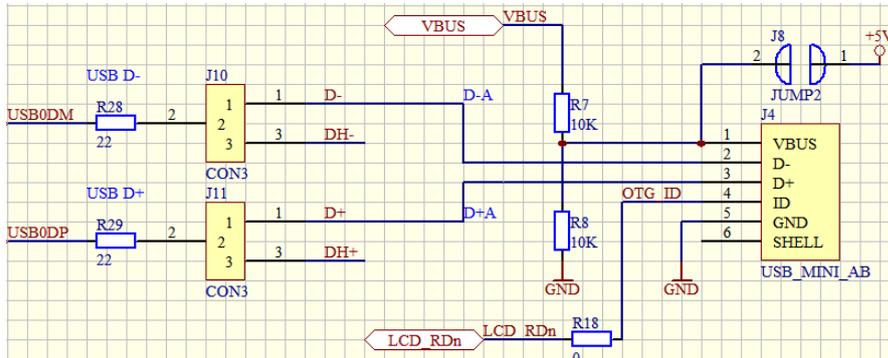


2.2 USB Host



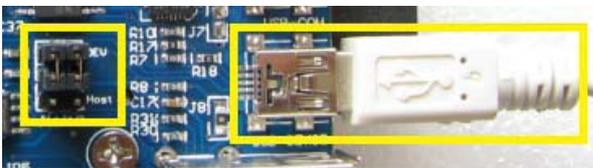
LM3S9B92 の USB Device と USB Host は共用していますので、J10 と J11 で選択します。Host として使用時は J10、J11 の 2-3 をショートします。

2.3 USB Device

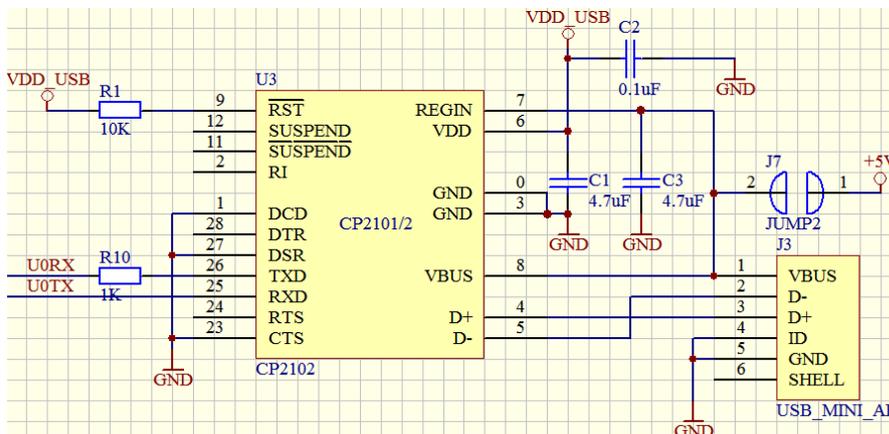


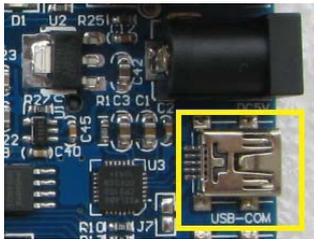
J8はデフォルトで0Ω の抵抗を設置しております。このインタフェースによりボードに給電できます。

LM3S9B92のUSB DeviceとUSB Hostは共用していますので、J10とJ11で選択します。Deviceとして使用時はJ10、J11の1-2をショートします。



2.4 USB COM

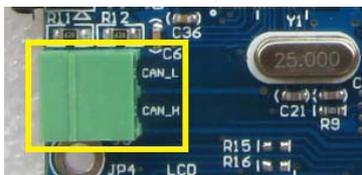
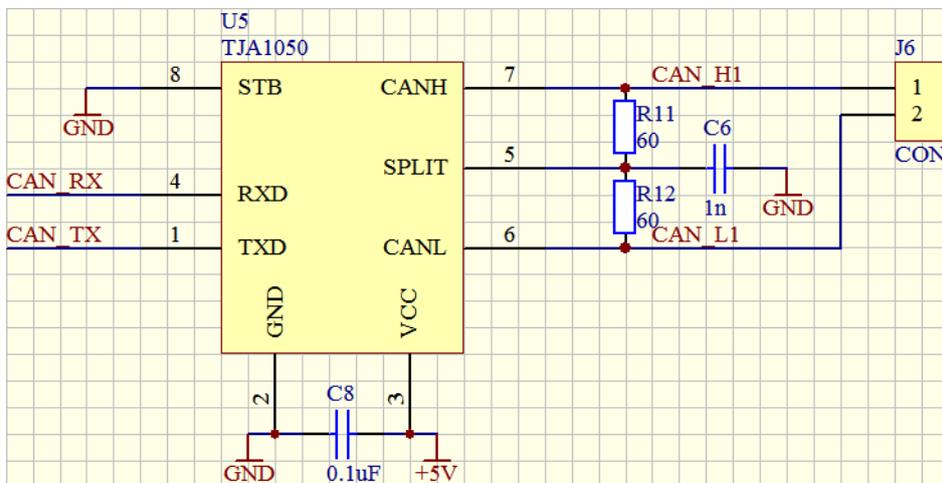




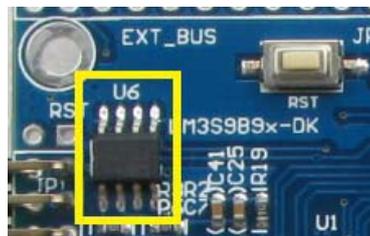
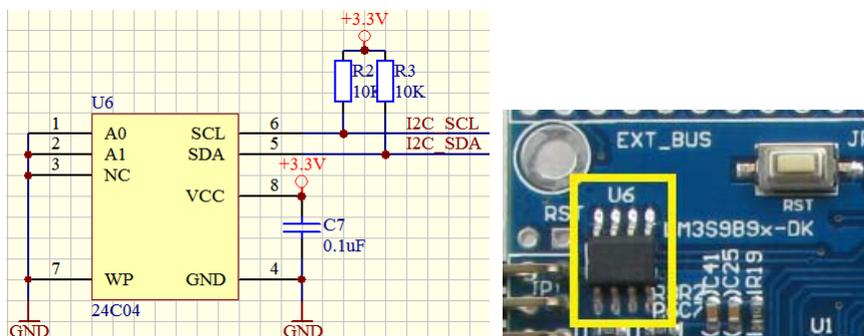
U3はUSBシリアル変換用でシリアルポートないPCなどでシリアルプログラムのデバッグができます。

J7はデフォルトでオープン。ボードに電源供給はできません。

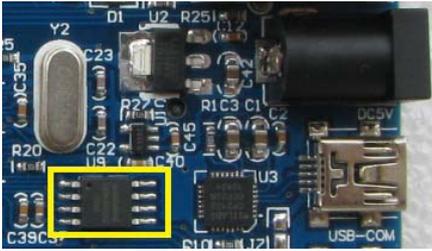
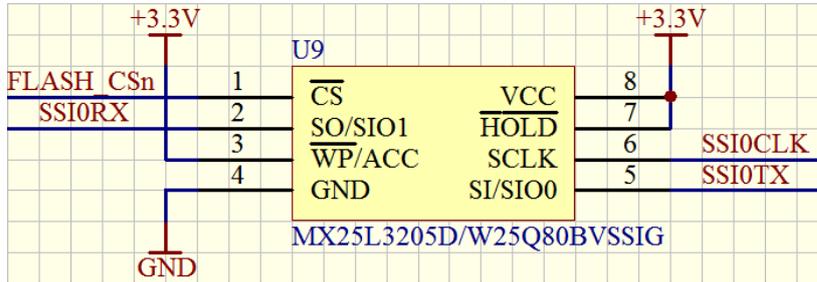
2.5 CAN



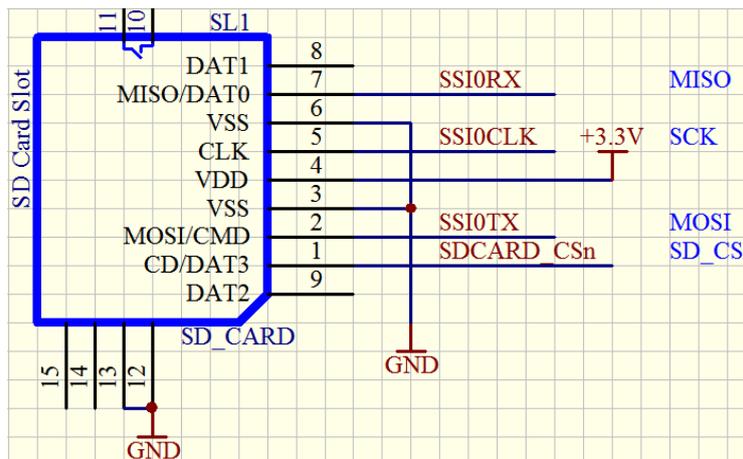
2.6 IIC EEPROM



2.7 SPI Flash

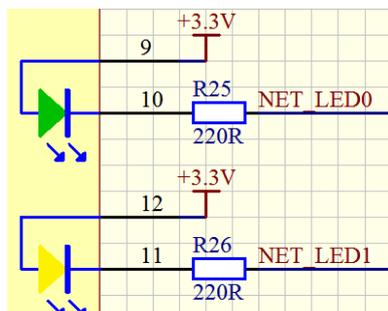


2.8 SD Card



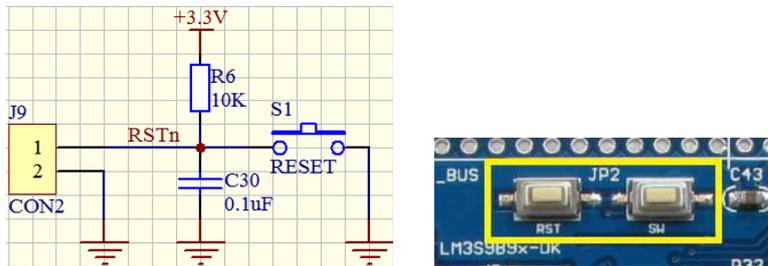
SPI インタフェースで SD カードと通信します。

2.9 LED



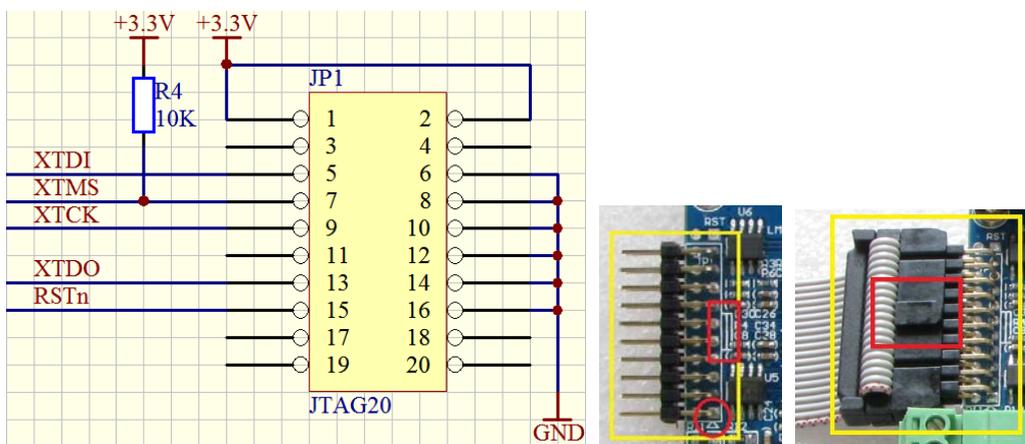
I/O を節約する為、イーサネットの緑色と黄色 LED を普通の LED と共用します。

2.10 Key

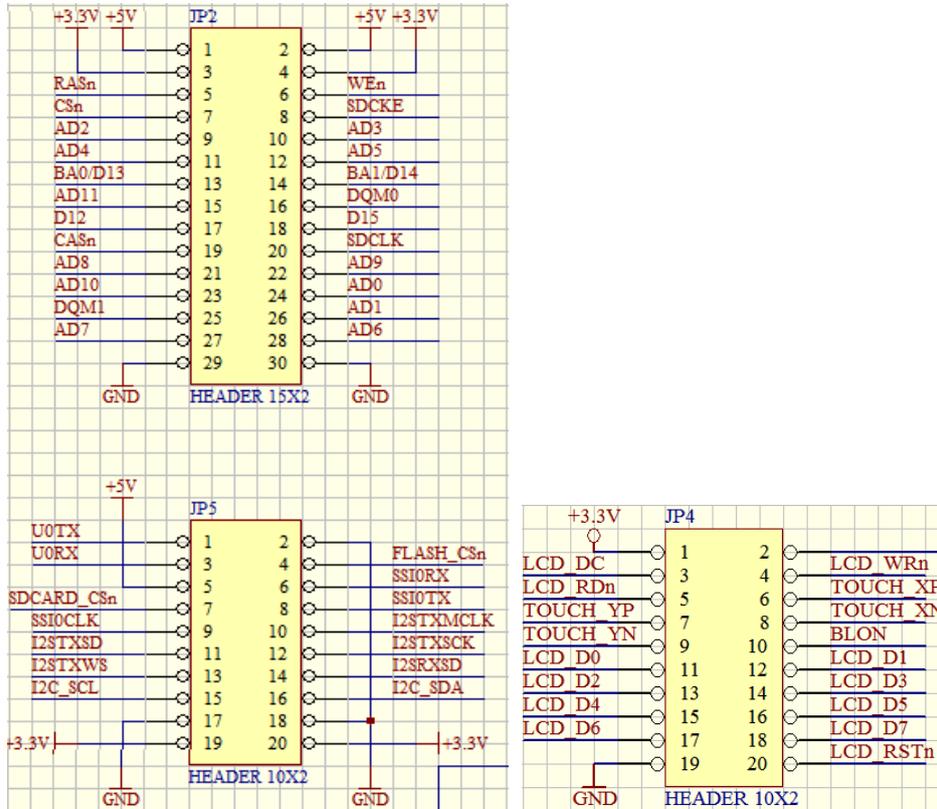


RST と標示しているのは Reset キー。SW は普通のユーザーキー、IIC インタフェースと共用。

2.11 JTAG



2.12 EXTEND I/O



本ボードは LM3S9B92 の大部分の I/O（イーサネット及びUSB 以外）を引き出しています。

第三章 開発ツールKEILの応用

本章ではご参考まで KEIL の無償評価版 MDK315B でインストール手順を説明します。

Keil社のHP (<http://www.keil.com/>) から最新版がダウンロード出来ます。

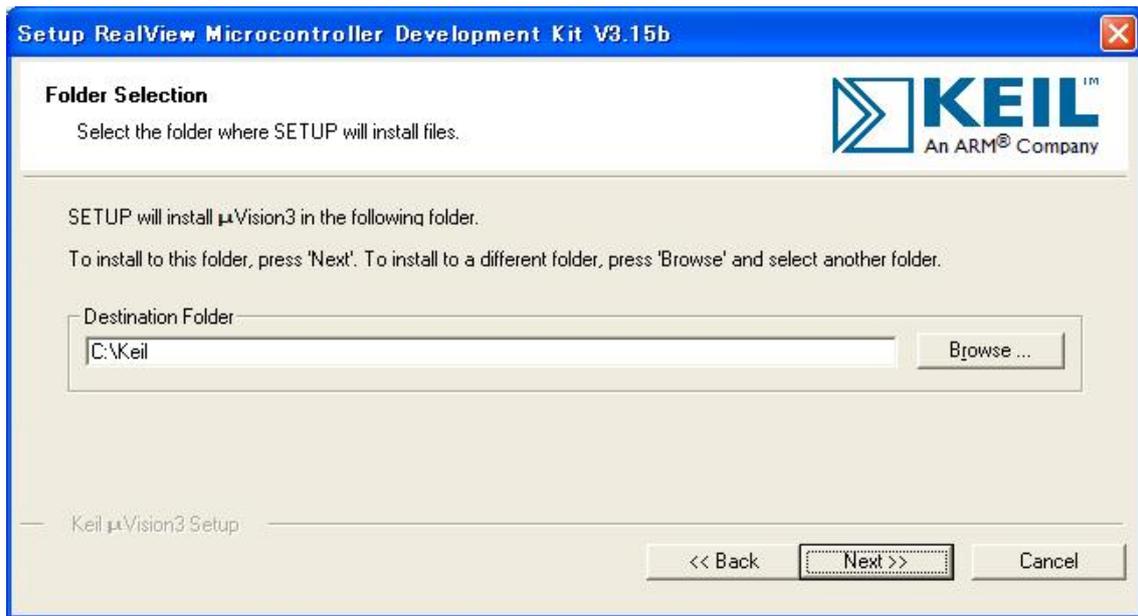
3.1 KEILのインストール

MDK315B.exe を実行して、KEIL3.15 をインストールする。

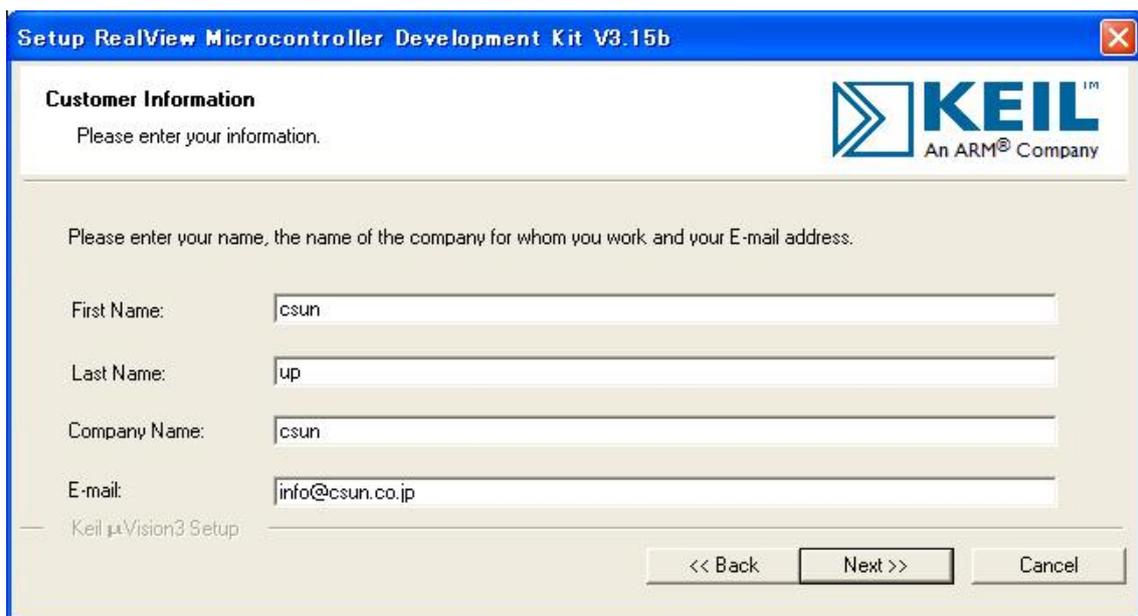


「Next」ボタンを押すと、英文のライセンス契約画面が表示される。同意できる場合は、「I accept the terms of the license agreement」を選択して、「Next」ボタンを押す。

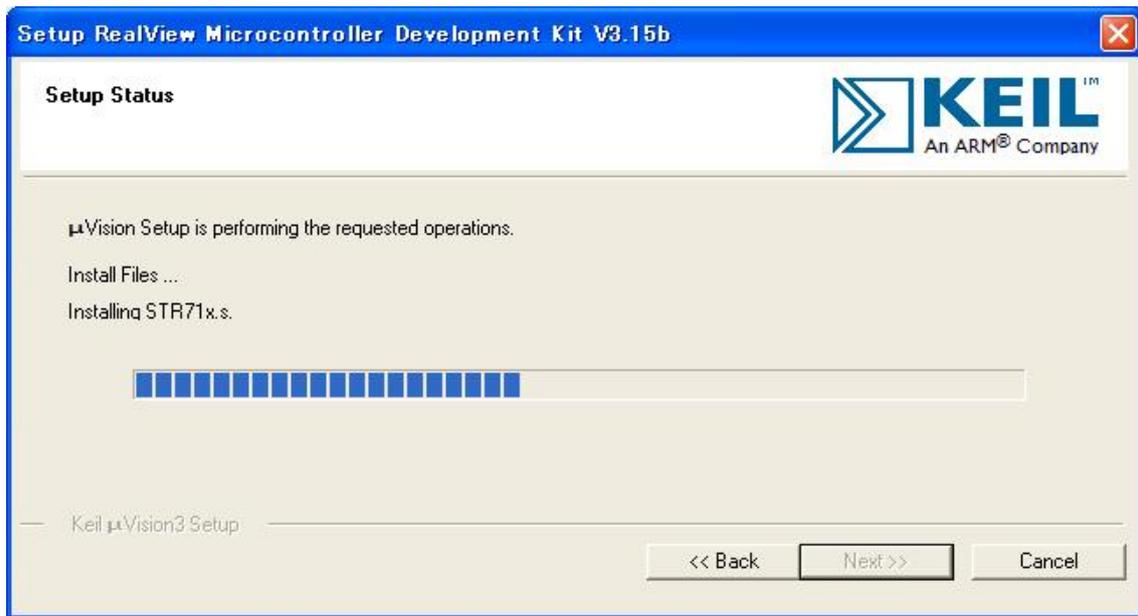




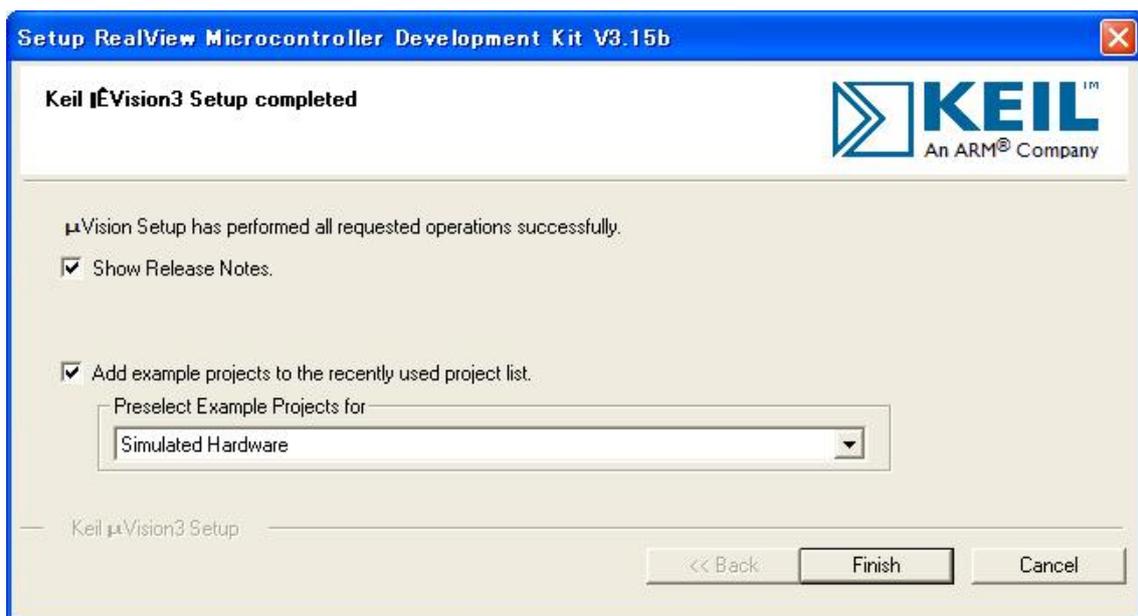
インストール先フォルダを変更せず、そのまま進んでください。



使用者の名前と所属会社名を入力するダイアログが表示される。名前は半角のアルファベットで入力してください。



インストール中の画面です。



最後に「Finish」をクリックすると、ウィザードが閉じられてインストール終了。
デモ版ではライセンスがないので、プログラムのサイズ制限があります。ライセンスを取得するにはKeil社の日本代理店にご連絡ください。

3.2 環境の設定

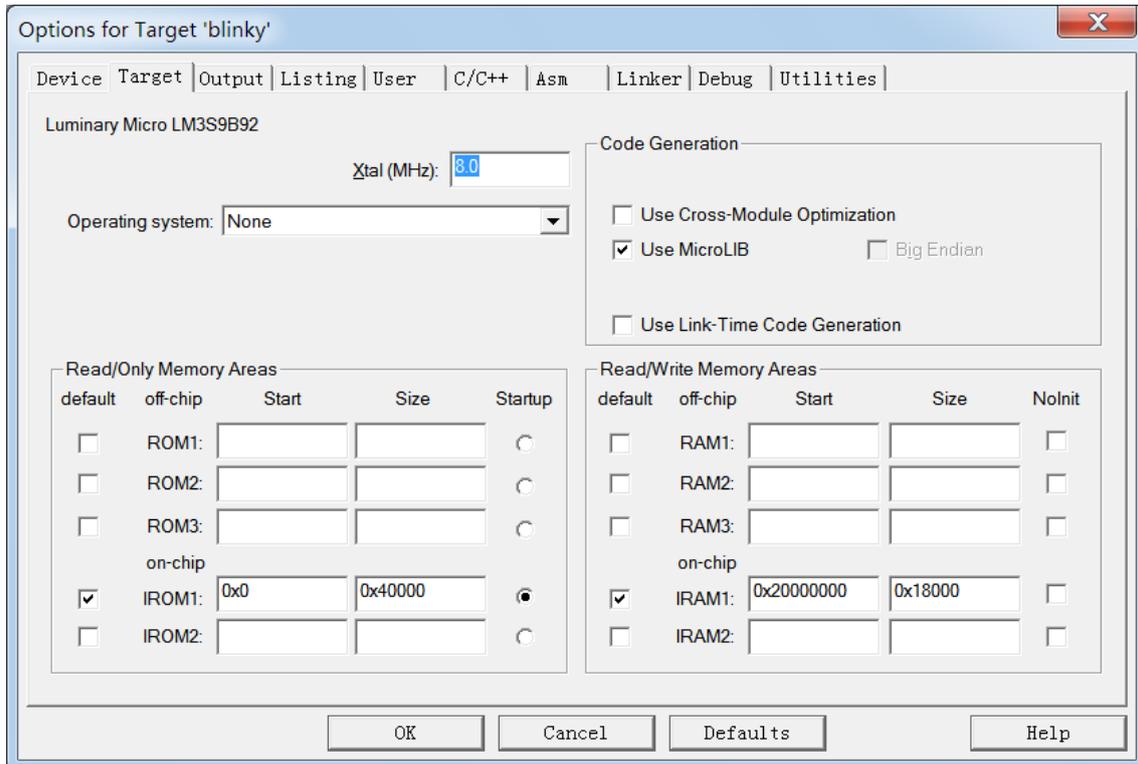
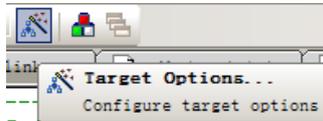
3.2.1 ダウンロード関連

サンプルソースで例を挙げます。

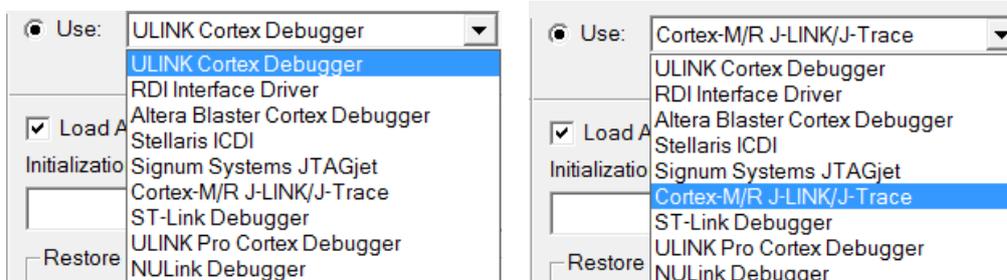
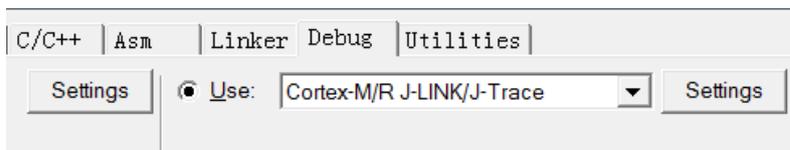
プロジェクトファイルCode¥ek-1m3s9b92¥blinky¥Blinky.uvprojをダブルクリックする。或

いはKEILのメニューでProject→Open Project…でBlinky.uvprojを選択する。

“Target Options” をクリックする。

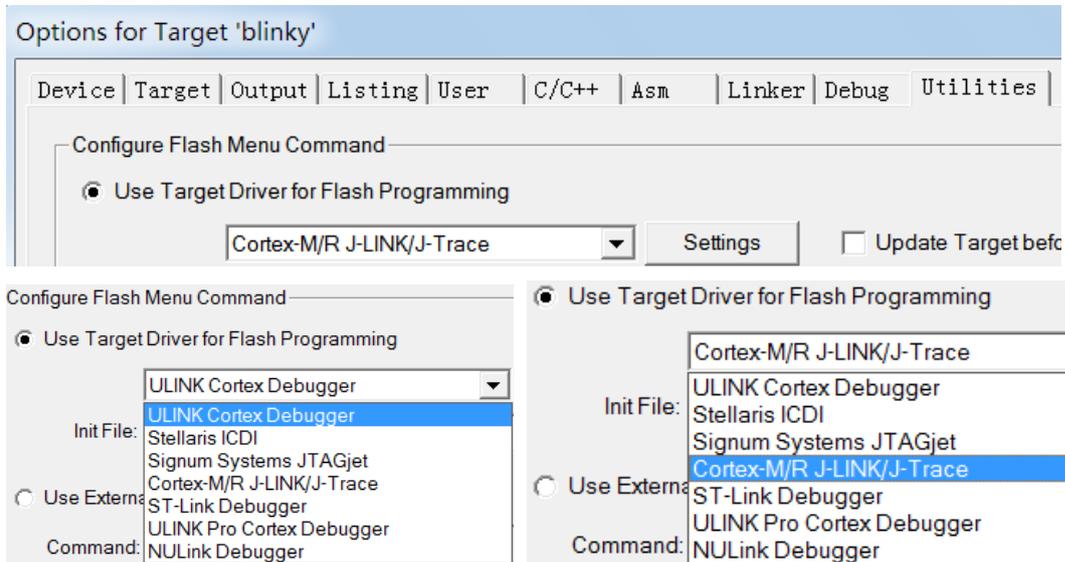


“Debug” タブを選択して、“Use Target Driver for Flash Programming” の所デバッグ型を選択します。ULINKを利用する場合は“ULINK Cortex Debugger”、JLINKを利用する場合は“Cortex-M3 J-LINK” を選択します。

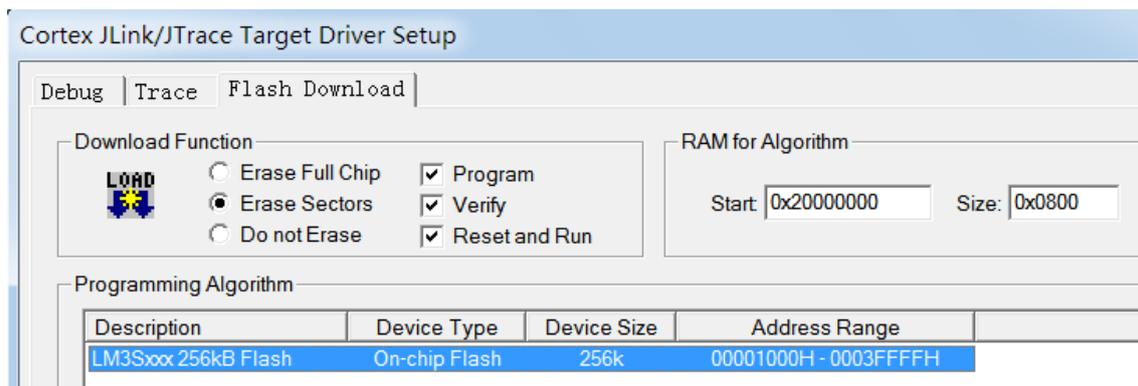


“Utilities” タブを選択して、“Use Target Driver for Flash Programming” の所デバッグ型を選択します。ULINKを利用する場合は“ULINK Cortex Debugger”、JLINKを利用する場合は“Cortex-M3 J-LINK” を選択します。

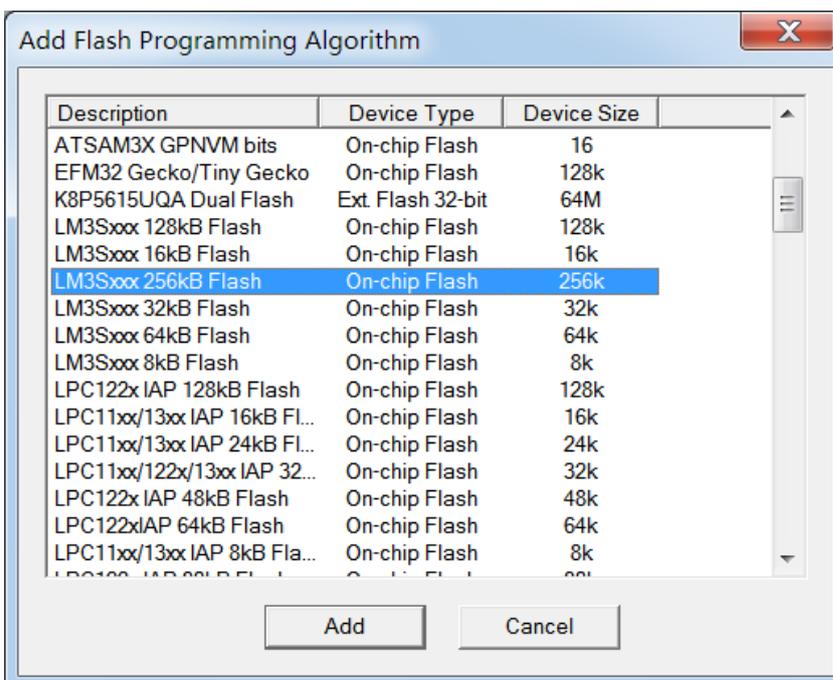
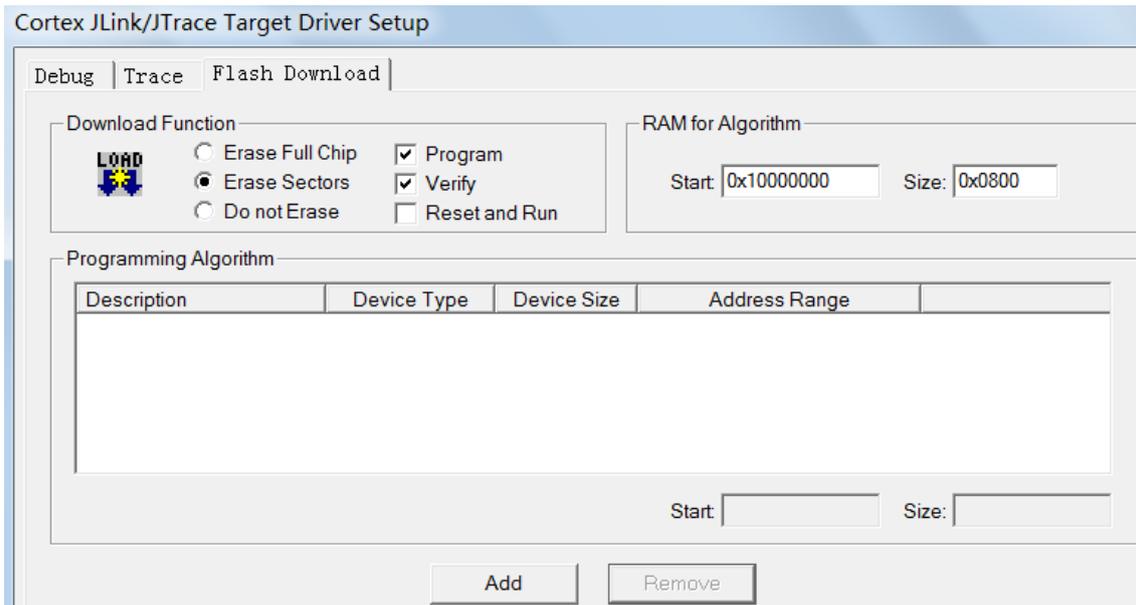
る場合は“Cortex-M3 J-LINK” を選択します。



「Settings」 ボタンを押す。JLINK を利用として時の例を挙げます。



“Flash Download” タブ画面の“Programming Algorithm” にデバイス情報がない場合、「Add」 ボタンを押す。

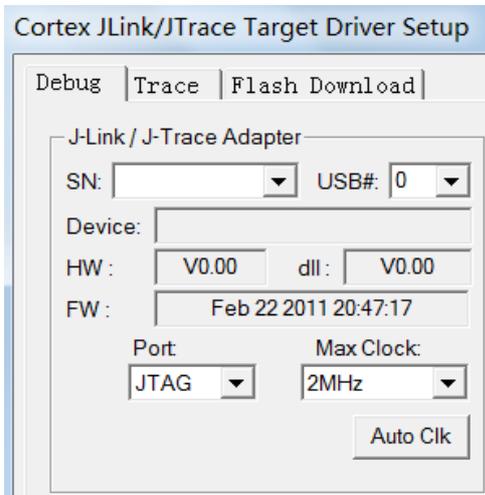


“Reset and Run” の所を選択すると、ダウンロード後すぐ実行します。

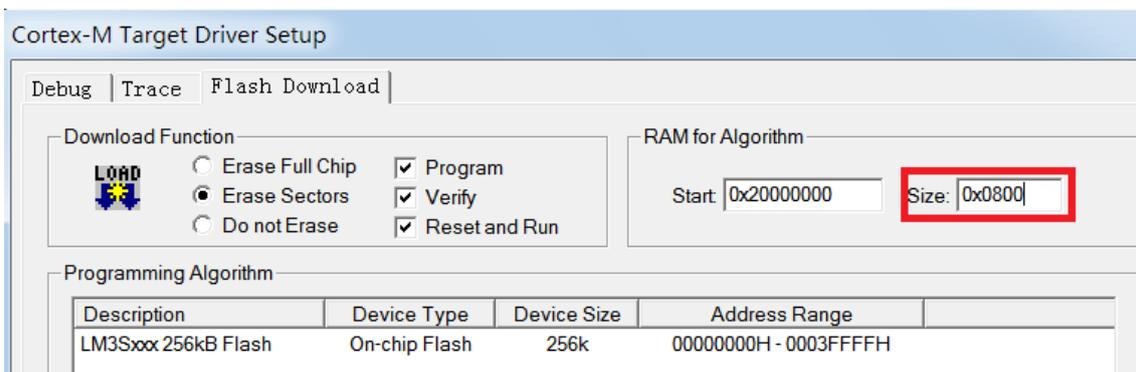


“Debug” タブを開いて設定します。ダウンロード失敗する時、スピードを下げて、500K或

いは以下に設定してお試してください。



RAM for Algorithmのサイズを0x0800になっているか確認する：

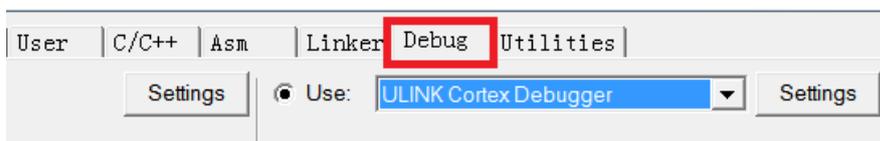


上記設定後、“Download to Flash Memory” を押してダウンロードができます。

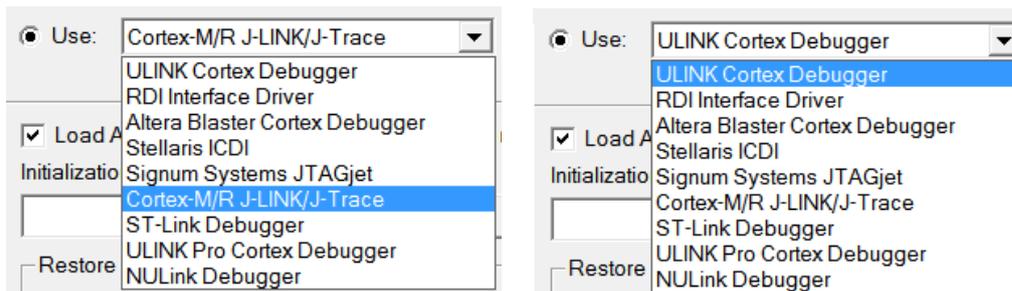


3.2.2 デバッグ関連

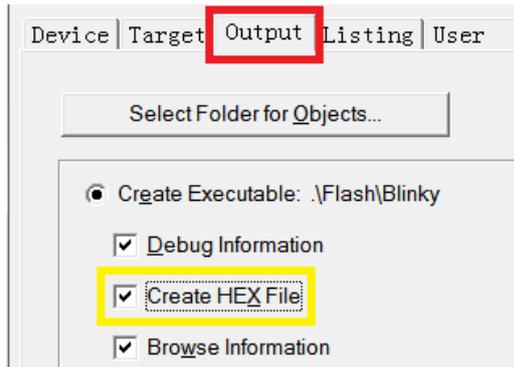
「Debug」タブを選択する。



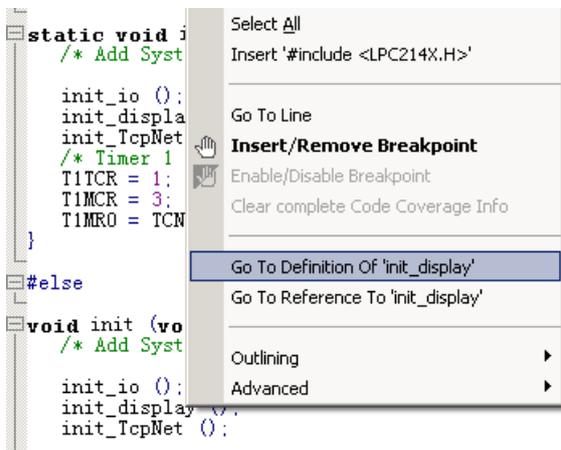
デバッグツールがULINKの場合は“ULINK Cortex Debugger”、JLINKの場合は“Cortex-M3 J-LINK”を選択する。



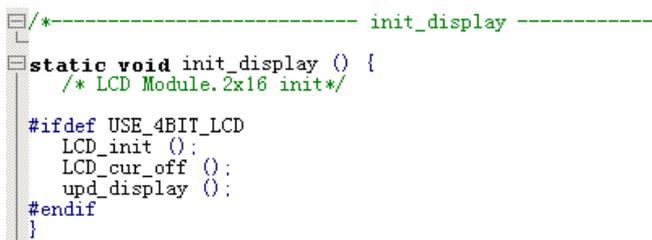
Hexファイル生成必要な場合は「Output」タブで設定します。



“Browse Information”の所も選択すると、ソースコードの検索が便利になります。例えば任意の所でinit_displayをコールした場合、マウスで右クリックすると“Go To Definition Of init_display”選択肢があります。



関数のソースと所に飛びます。



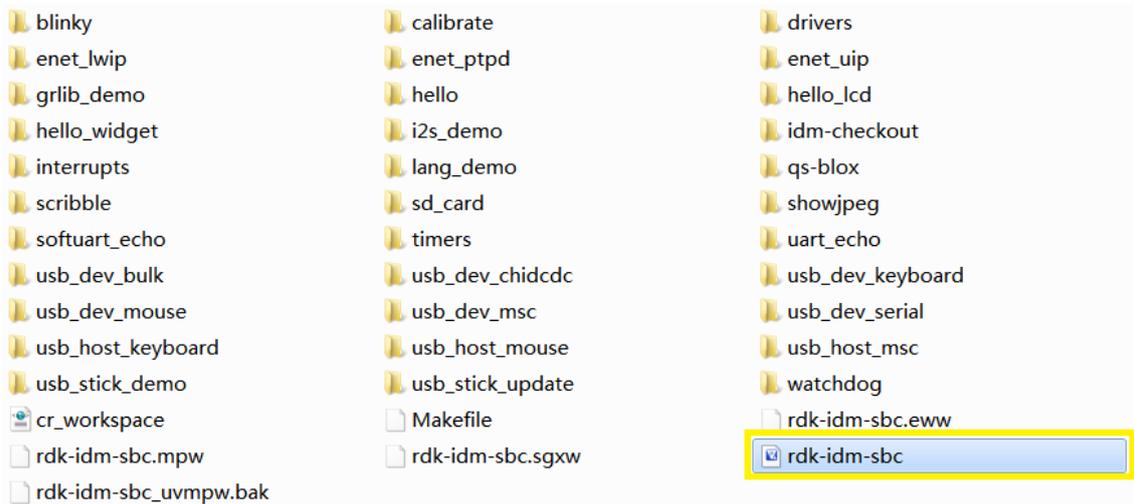
第四章 サンプルソースについて

4.1 CodeStellarisWareboardsSmart-9B92-C3

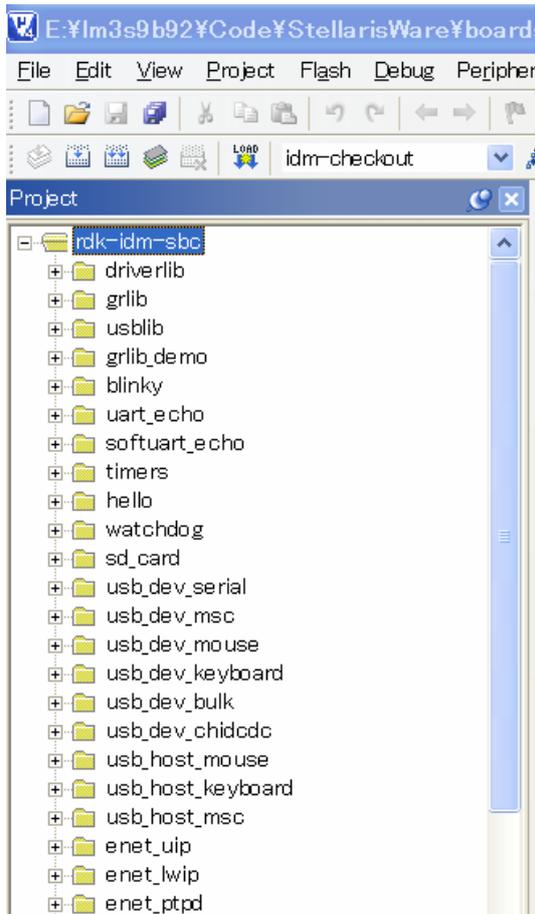
※MCU が C1 のバージョンの場合は CodeStellarisWareboardsSmart-9B92-C1 をご参照ください。



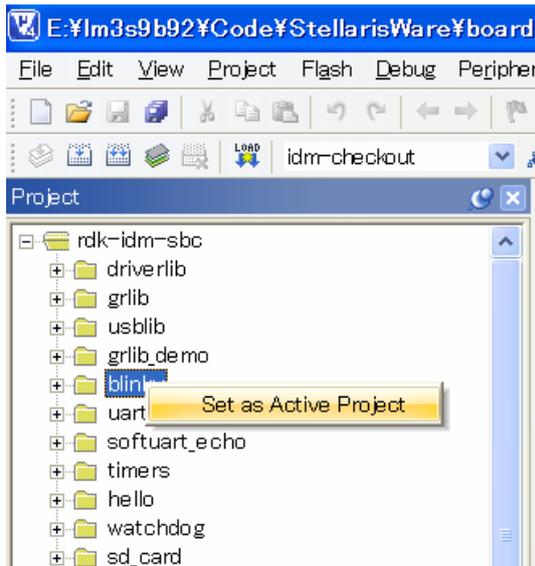
この図は C5 バージョンです。



rdk-idm-sbc_uvmpw は多プロジェクト管理ファイル、開くと全てのプロジェクトがリストされます：



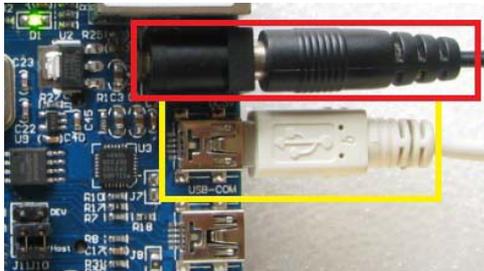
利用するプロジェクトをマウスで右クリックして“Set as Active Project”をクリックすると、そのプロジェクトが利用できます。



4.1.1 ¥blinky

RJ45 に付いている緑色と黄色 LED を点滅させます。

4.1.2 ¥uart_echo



シリアルポートのテストプログラムです。ハイパーターミナルから入力したデータをそのままシリアルポートから返送されます。

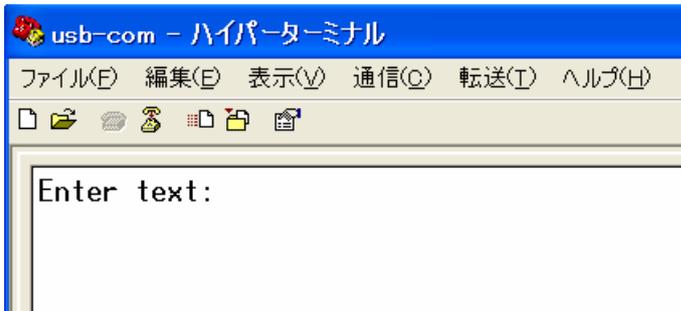
Step1、Mini USBケーブルでUSB-COMインタフェースと繋ぐ。デバイスマネージャで配分されたシリアルポートを確認します。例えばCOM6。



Step2、ハイパーターミナルを開いて COM6 ポートを選択して、下記の様に設定します：



Step3、本テストプログラムをボードにダウンロードすると、ハイパーターミナルに下記情報が提示されます：



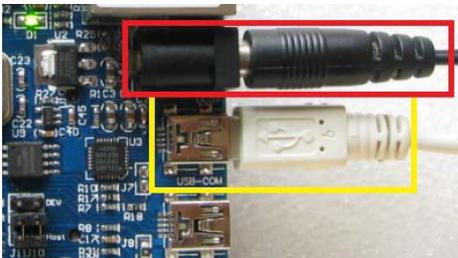
Step4、任意の文字を入力するとハイパーターミナルに表示されます：



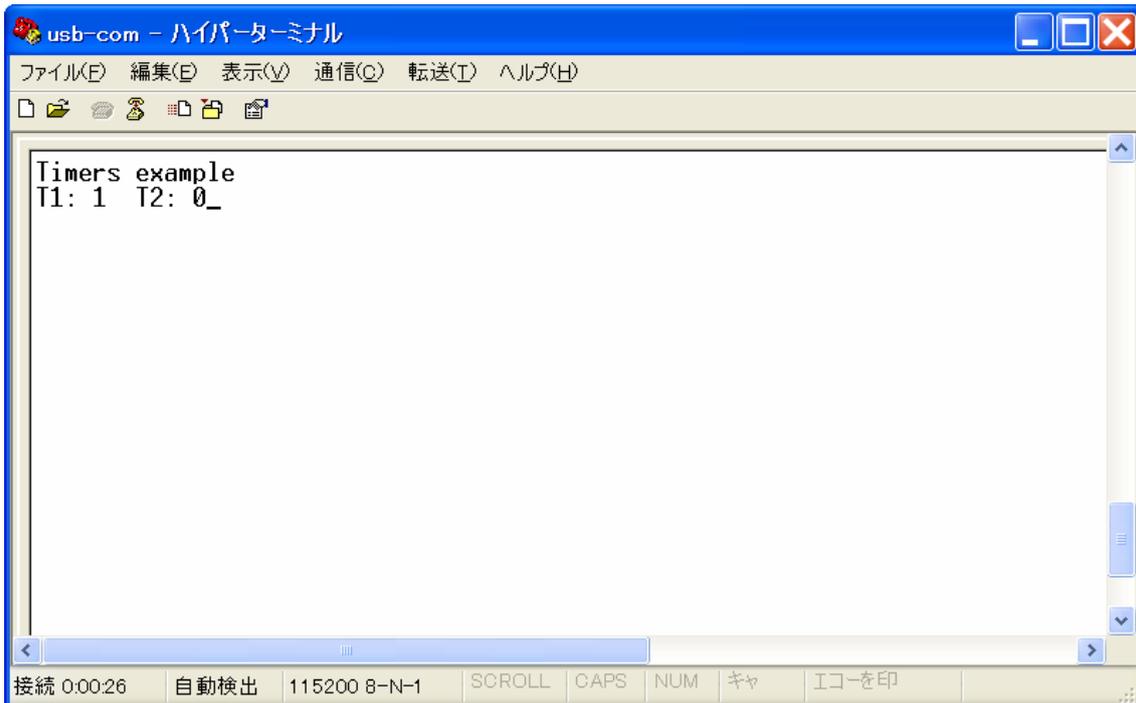
4.1.3 ¥softuart_echo

普通の I/O でシリアルポートを模擬します。機能とテスト方法は前例と同じ。

4.1.4 ¥timers



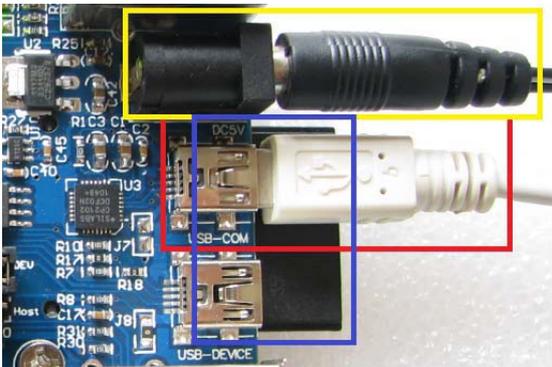
タイマー割込みテストプログラムです。タイマー1は1秒毎にシリアルポートにデータを出します。タイマー2は0.5秒毎にシリアルポートにデータを出します。



4.1.5 ¥watchdog

割り込みモードで実現しています。間隔は1秒。割り込みサービスプログラムでは毎回LED制御のIOをNOT演算します。結果としてはRJ45の緑色のLEDが点滅します。

4.1.6 ¥sd_card



本テストプログラムにはFatFs ファイルシステムを利用し、LCD 或いはハイパーターミナルでSDカード上のファイルを操作できます。

Step1、シリアルポートの設定はuart_echo例と同じ。ハイパーターミナルから下記の情報が表示されます：

```
SD Card Example Program
Type 'help' for help.

/>>
```

Step2、help を入力するとサポートするコマンドが表示されます：

```
/> help
Available commands
-----
help : Display list of commands
h    : alias for help
?    : alias for help
ls   : Display list of files
chdir: Change directory
cd   : alias for chdir
pwd  : Show current working directory
cat  : Show contents of a text file

/>> _
```

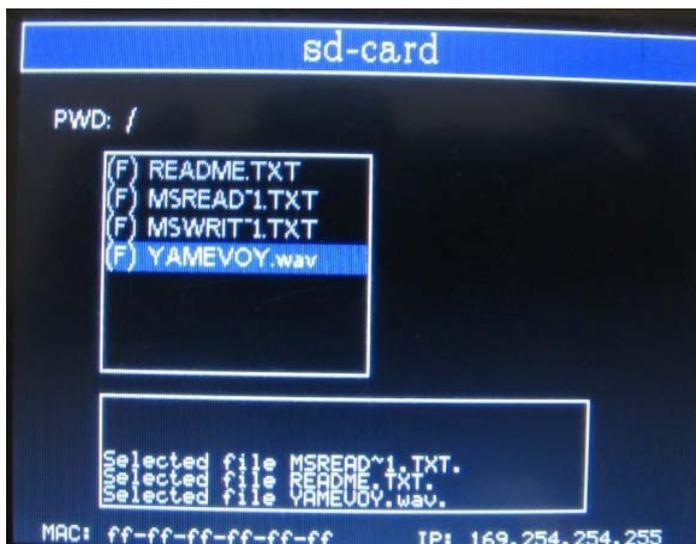
Step3、ls コマンドを入力すると SD カードの内容が表示されます：

```
/> ls
----A 2011/04/11 10:55      6 README.TXT
----A 2011/04/14 08:44      8 MSREAD~1.TXT
----A 2011/04/14 08:44      0 MSWRIT~1.TXT
----A 2011/05/26 16:09 1542844 YAMEVOY.wav

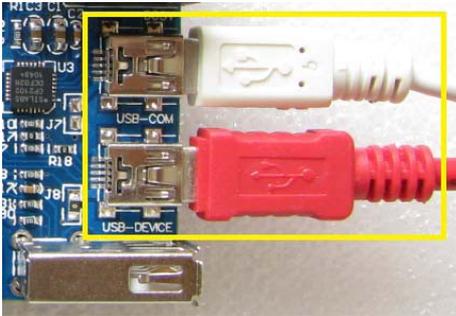
  4 File(s), 1542858 bytes total
  0 Dir(s),  7764384K bytes free

/>> _
```

Step4、LCD を接続している場合、下記の様な内容が表示されます：



4.1.7 ¥usb_dev_serial



USB 模擬シリアルポートプログラムです。USB デバイスをシリアルデバイスとして利用できますのでホスト側の USB ドライバが必要ないです。

Step1、テストプログラムをボードにダウンロードします。J10、J11を“DEV”側に設定して、Mini USBケーブルでUSB-DEVICEインタフェースと繋ぐ。PC側は新しいUSBデバイスが検出された提示が出ます。提示によって、¥Code¥StellarisWare¥windows_driversにあるドライバを指定します。

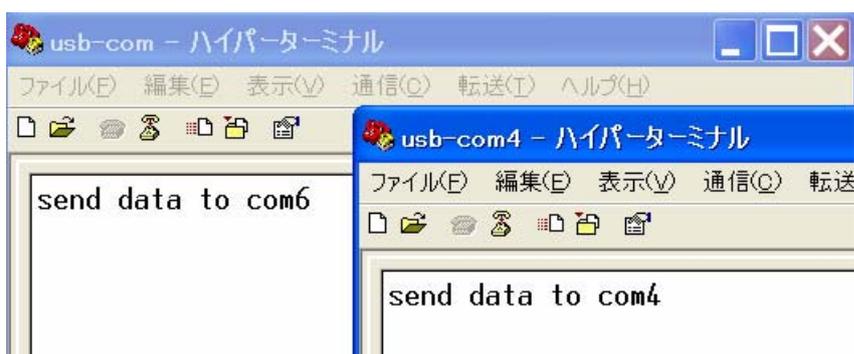
Step2、デバイスマネージャで配分されたシリアルポートを確認します。例えばCOM4。



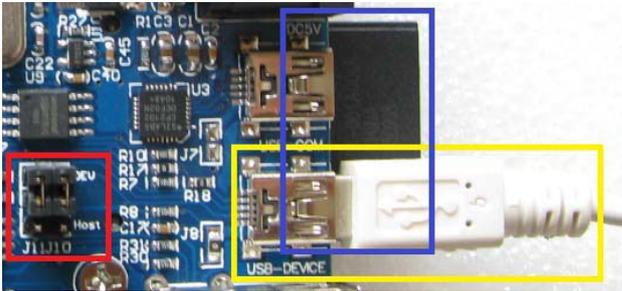
Step3、もう一本のMini USBケーブルでUSB-COMインタフェースと繋ぐ。デバイスマネージャで配分されたシリアルポートを確認します。例えばCOM6。



Step4、二つのハイパーターミナルを開いて、同じ属性で設定します。二つの仮シリアルポートがお互いに通信できます。



4.1.8 ¥usb_dev_msc

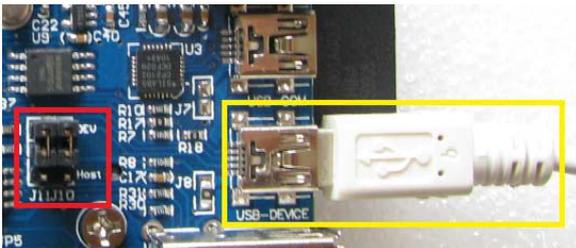


USB カードリーダーとして利用できます。

Step1、J10、J11を“DEV”側に設定して、Mini USBケーブルでUSB-DEVICEインタフェースと繋ぐ。PC側は新しいUSBデバイスが検出された提示が出ます。自動的にドライバーをインストールします。

Step2、新しい移動デバイスが追加されています。開くと中の内容が見えます。

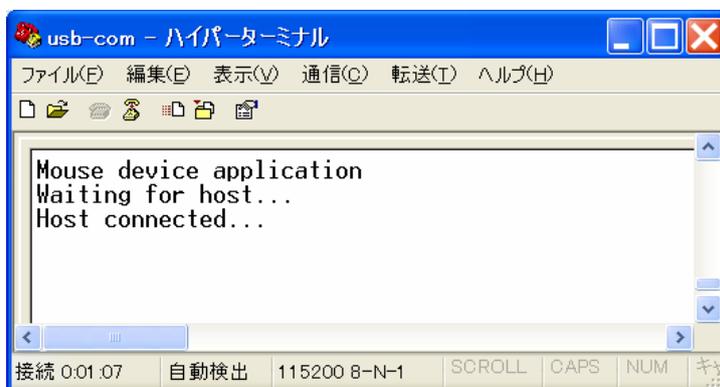
4.1.9 ¥usb_dev_mouse



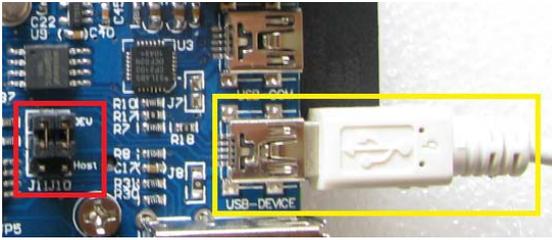
HID マウスの USB デバイスプログラムです。

Step1、J10、J11を“DEV”側に設定して、Mini USBケーブルでUSB-DEVICEインタフェースと繋ぐ。PC側は新しいUSBデバイスが検出された提示が出ます。自動的にドライバーをインストールします。

Step2、PC上のマウスが四角形で移動します。同時にハイパーターミナル側は下記情報が表示されます：



4.1.10 ¥usb_dev_keyboard



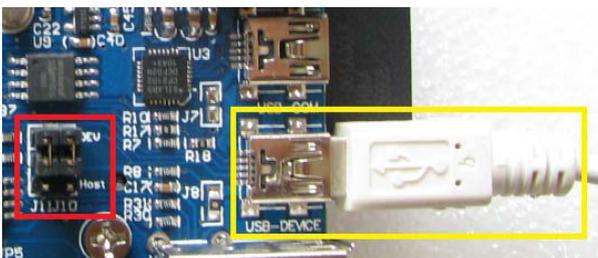
HID キーボードの USB デバイスプログラムです。

Step1、J10、J11を“DEV”側に設定して、Mini USBケーブルでUSB-DEVICEインタフェースと繋ぐ。PC側は新しいUSBデバイスが検出された提示が出ます。自動的にドライバーをインストールします。

Step2、Wordファイルを新規作成して開きます。ボードのSWキーを押すと、下記内容がファイルに書かれます：“Stellaris…SeeHowFarYourDollarCanGo”。シリアルポートからは下記情報が表示します：



4.1.11 ¥usb_dev_bulk

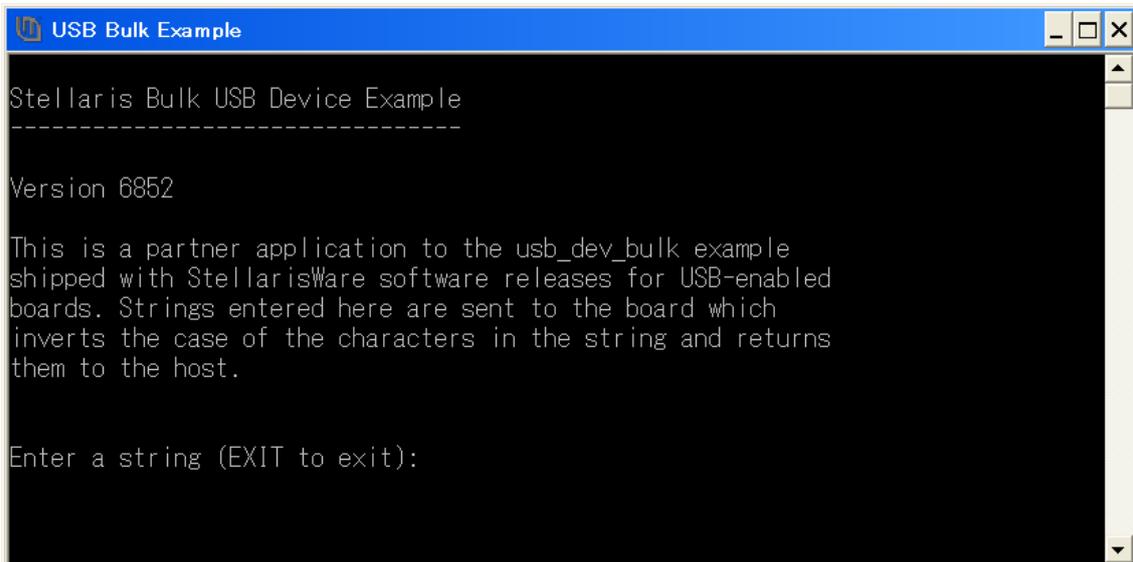


USB バルク転送プログラムです。

Step1、¥TOOLS¥SW-USB-win-6852.exeを実行してインストールします。インストール完了したプログラムはUSB Bulk Example.exeです。

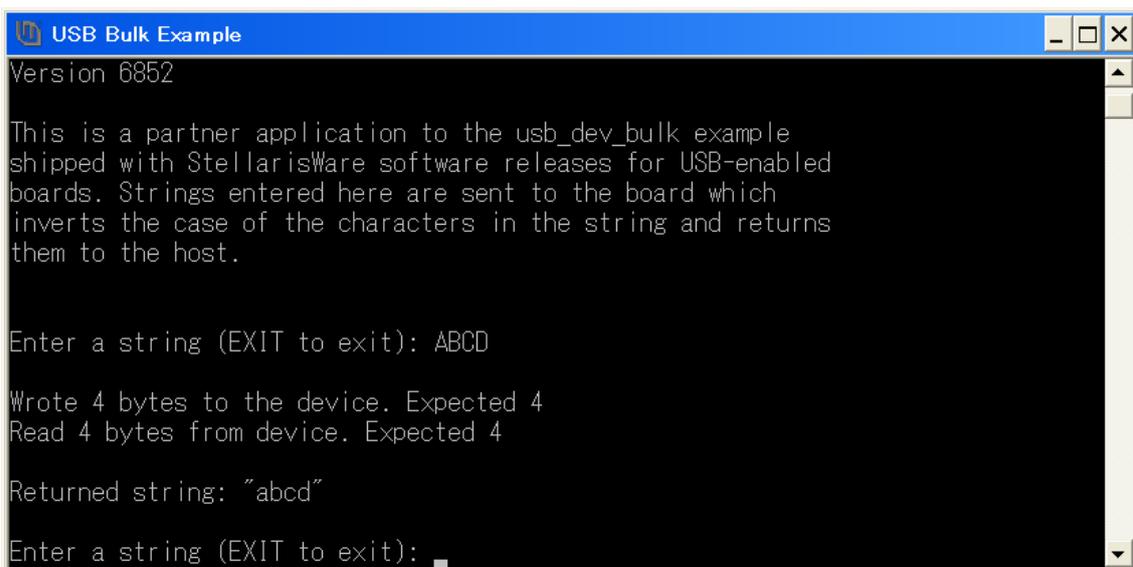
Step2、J10、J11を“DEV”側に設定して、Mini USBケーブルでUSB-DEVICEインタフェースと繋ぐ。PC側は新しいUSBデバイスが検出された提示が出ます。提示によって、¥Code¥StellarisWare¥windows_driversにあるドライバを指定します。

Step3、USB Bulk Example.exeを実行します。受信したキャラクタを大文字/小文字変換してホストに返送します。



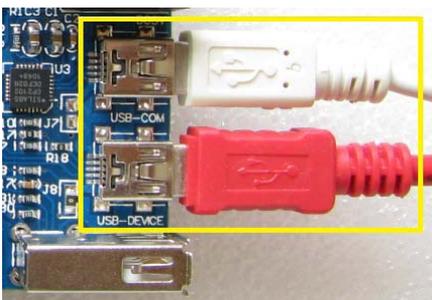
```
USB Bulk Example
Stellaris Bulk USB Device Example
-----
Version 6852
This is a partner application to the usb_dev_bulk example
shipped with StellarisWare software releases for USB-enabled
boards. Strings entered here are sent to the board which
inverts the case of the characters in the string and returns
them to the host.
Enter a string (EXIT to exit):
```

「ABCD」を入力してEnterを押すと、ボードから「abcd」を返送します。



```
USB Bulk Example
Version 6852
This is a partner application to the usb_dev_bulk example
shipped with StellarisWare software releases for USB-enabled
boards. Strings entered here are sent to the board which
inverts the case of the characters in the string and returns
them to the host.
Enter a string (EXIT to exit): ABCD
Wrote 4 bytes to the device. Expected 4
Read 4 bytes from device. Expected 4
Returned string: "abcd"
Enter a string (EXIT to exit):
```

4.1.12 ¥usb_dev_chidcdc



本プログラムを実行するとPC側はUSB HIDデバイスとUSB CDCデバイス（仮シリアルポート）が追加されます。

Step1、J10、J11を“DEV”側に設定して、Mini USBケーブルでUSB-DEVICEインタフェースと繋ぐ。PC側は新しいUSBデバイスが検出された提示が出ます。提示によって、¥Code¥StellarisWare¥windows_driversにあるドライバを指定します。

Step2、デバイスマネージャでポートとHIDデバイスの所にデバイスが追加された事を確認。例えばCOM6。

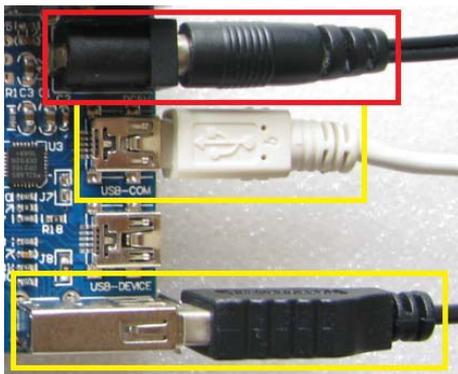
Step3、ハイパーターミナルを開いて、「?」とEnterを押すと、本プログラムがサポートするコマンドを提示します：

```
?
Available commands
-----
help : Display list of commands
h    : alias for help
?    : alias for help
mouse : turn (on|off) mouse movements
led   : Set LED mode (on|off|toggle|activity)
> _
```

Step4、led off、led on、led toggleなどのコマンドを入力してRJ45の緑色のLEDの変化を見ます。

Step5、mouse onを入力するとマウスが正方形を書きます。mouse offを入力するとマウスが正常に戻ります。

4.1.13 ¥usb_host_mouse

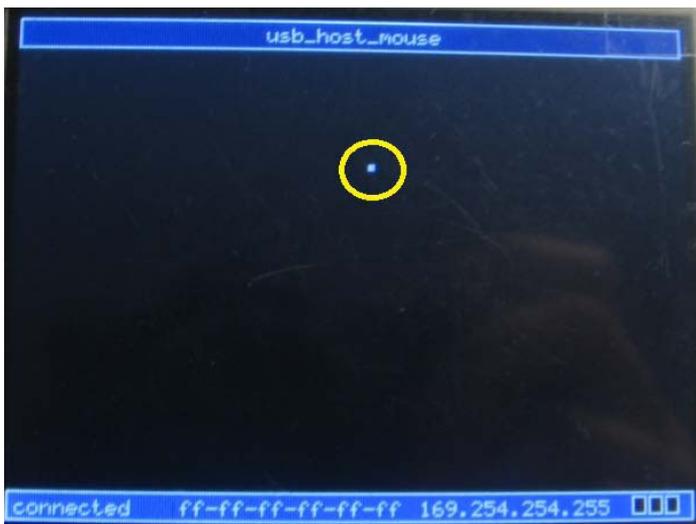


本テストプログラムはUSBマウスを接続して制御できます。

Step1、上記図の様にUSBマウスをボードのUSB Hostインタフェースに差し込む。USB-COMとPCを繋ぐ。プログラムをボードにダウンロードして実行すると、USBマウスの座標がシリアルポートから出力します：

```
X:02.  
Y:-16.  
X:09.  
X:07.  
Button Pressed 01  
Button Released 01  
Y:32.  
X:-2.  
Y:15.  
Y:16.  
Y:16.  
X:02.  
X:01.  
X:01.  
Button Pressed 02  
Button Released 02  
X:-2.  
Y:15.
```

Step2、LCD を繋いでいる場合、白い点も USB マウスの動きによって移動します：



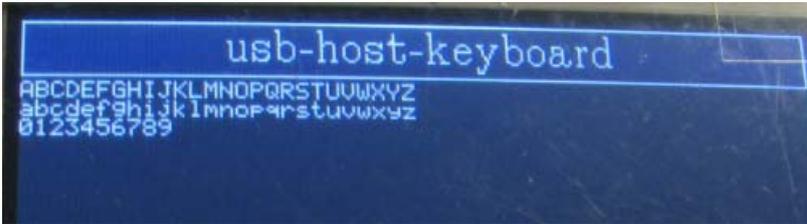
4.1.14 ¥usb_host_keyboard

本テストプログラムは USB キーボードを接続して制御できます。

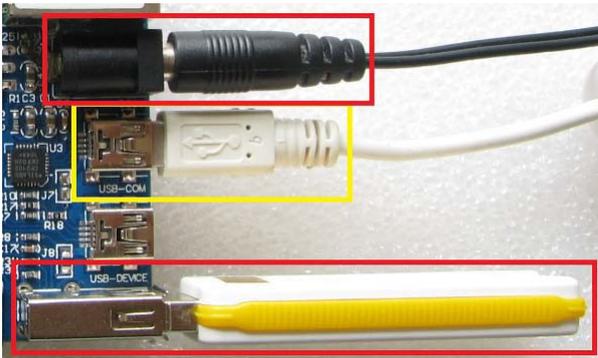
Step1、上記図の様に USB キーボードをボードの USB Host インタフェースに差し込む。
USB-COM と PC を繋ぐ。プログラムをボードにダウンロードして実行すると、USB キーボードから押したキーがシリアルポートから出力します：

```
Host Keyboard Application  
Keyboard Connected  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789
```

Step2、LCD を繋いでいる場合、LCD にも押したキーが表示されます：



4.1.15 ¥usb_host_msc



本テストプログラムは USB メモリを接続して制御できます。※全ての USB メモリが動作する保証はできない。本試験では Kingston 4G の USB メモリを利用しております。

Step1、上記図の様に USB メモリをボードの USB Host インタフェースに差し込む。USB-COM と PC を繋ぐ。プログラムをボードにダウンロードして実行すると、下記情報がシリアルポートから出力します：

```
USB Mass Storage Class Host Example Program
Type 'help' for help.

/> help

Available commands
-----
help : Display list of commands
h    : alias for help
?    : alias for help
ls   : Display list of files
chdir: Change directory
cd   : alias for chdir
pwd  : Show current working directory
cat  : Show contents of a text file

/>
```

Step2、ls コマンドを入力すると USB メモリの内容が表示されます：

```
----A 2010/08/08 11:01 1002097 KANEBE~1.PDF
----A 2010/07/30 17:04 3458141 FOXITP~1.ZIP
----A 2010/09/02 21:45 1241762 SD2440~1.PCB
----A 2010/09/02 21:41 1622687 SD2440~1.ASC
```

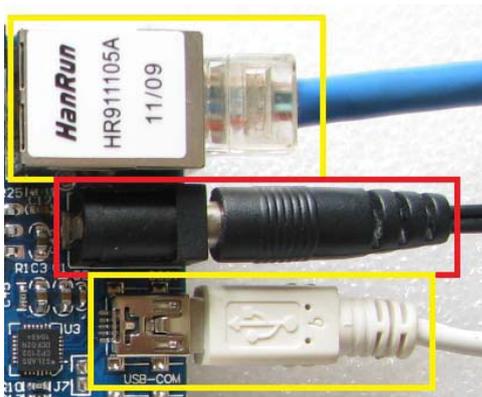
```
18 File(s), 190131176 bytes total
 2 Dir(s), 3652872K bytes free
```

```
/>
```

Step3、LCD を繋いでいる場合、LCD にも USB メモリの内容が表示されます：



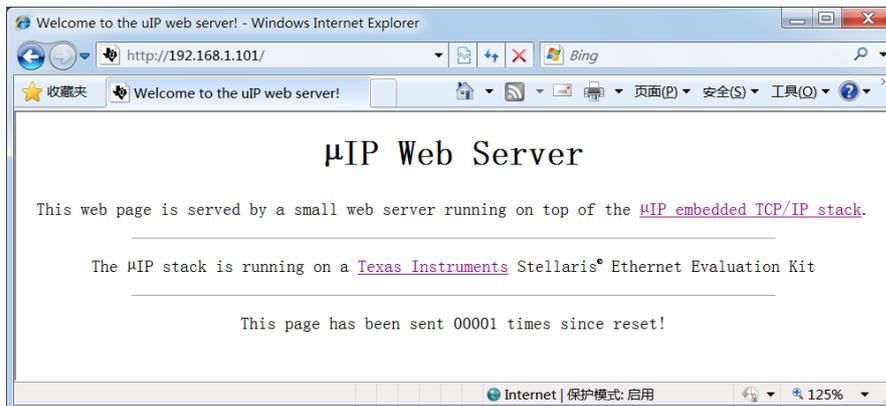
4.1.16 ¥enet_uip



本テストプログラムは uip ネットプロトコルで http ウェブサーバーの機能を実現します。Step1、上記図の様に LAN ケーブルでボードの ethernet インタフェースとルータを繋ぐ。USB-COM と PC を繋ぐ。プログラムをボードにダウンロードして実行すると、ボードが自動配分された IP アドレスがシリアルポートから出力します：

```
Ethernet with uIP
MAC Address Not Programmed!
Waiting for Link
Link Established
Waiting for IP address...
IP: 192.168.1.101
```

Step2、IEブラウザに<http://192.168.1.101>を入力するとボードのウェブ内容を見れます：



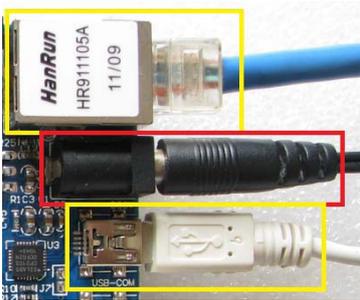
Step3、enet_uip.c を変更する事で固定 IP モードを実現できます。下記の図の様に USE_STATIC_IP 機能を有効にします：

```
enet_uip.c
133 // TODO: Uncomment the following #define
134 // instead of DHCP.
135 //
136 //*****
137 #define USE_STATIC_IP
138
139 #ifndef DEFAULT_IPADDR0
140 #define DEFAULT_IPADDR0          192
141 #endif
```

Step4、MAC アドレスは User Register 0-1 に保存されています。ROM_FlashUserGet 関数で取得できます。デフォルトでは MAC アドレスを書き込んでないので、読み出した MAC アドレスは全部 FF です。

```
enet_uip.c
596 // Read the MAC address from the user registers.
597 //
598 ROM_FlashUserGet(&ulUser0, &ulUser1);
599 if((ulUser0 == 0xffffffff) || (ulUser1 == 0xffffffff))
600 {
601 //
602 // We should never get here. This is an error if the MAC address has
603 // not been programmed into the device. Exit the program.
604 //
605 UARTprintf("MAC Address Not Programmed!\n");
606 // while(1)
607 // {
608 // }
609 }
```

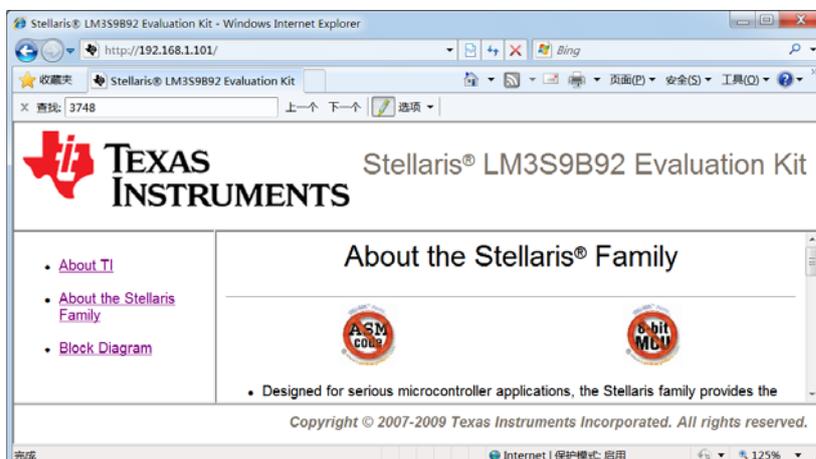
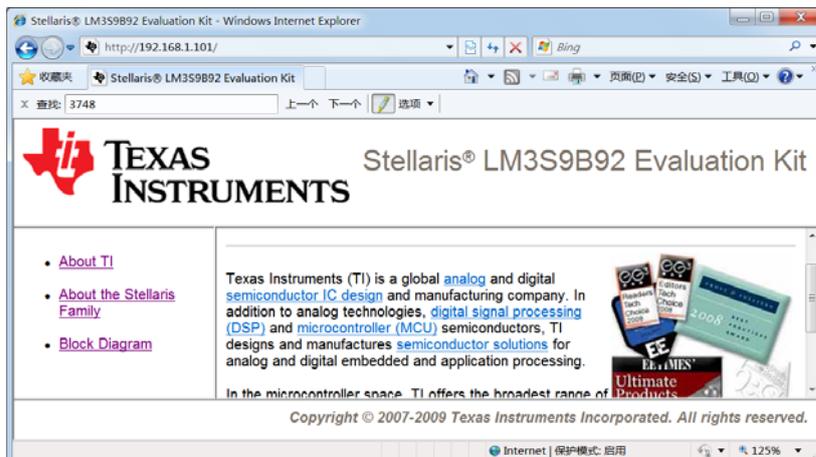
4.1.17 Yenet_lwip

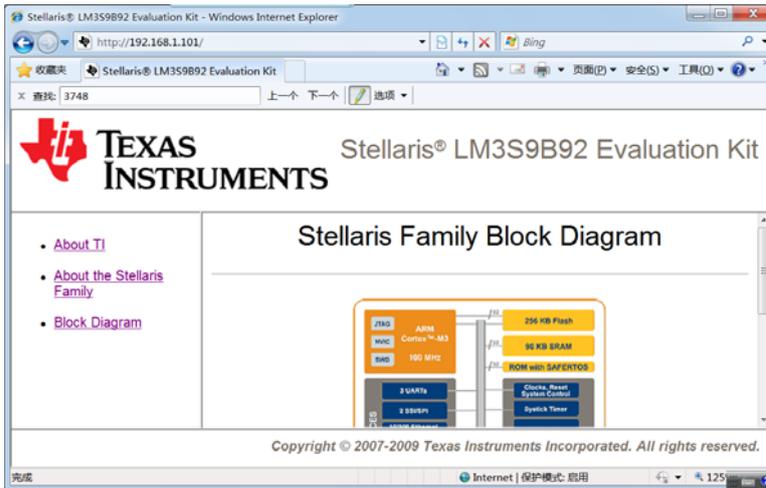


本テストプログラムは lwip ネットプロトコルで http ウェブサーバーの機能を実現します。Step1、上記図の様に LAN ケーブルでボードの ethernet インタフェースとルータを繋ぐ。USB-COM と PC を繋ぐ。プログラムをボードにダウンロードして実行すると、ボードが自動配分された IP アドレスがシリアルポートから出力します：

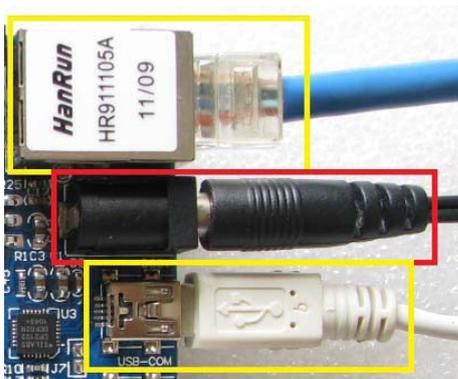
```
Ethernet with lwIP
MAC Address Not Programmed!
IP: 192.168.1.101
Netmask: 255.255.255.0
Gateway: 192.168.1.1
```

Step2、IEブラウザに<http://192.168.1.101>を入力するとボードのウェブ内容を見れます：





4. 1. 18 Yenet_ptpd

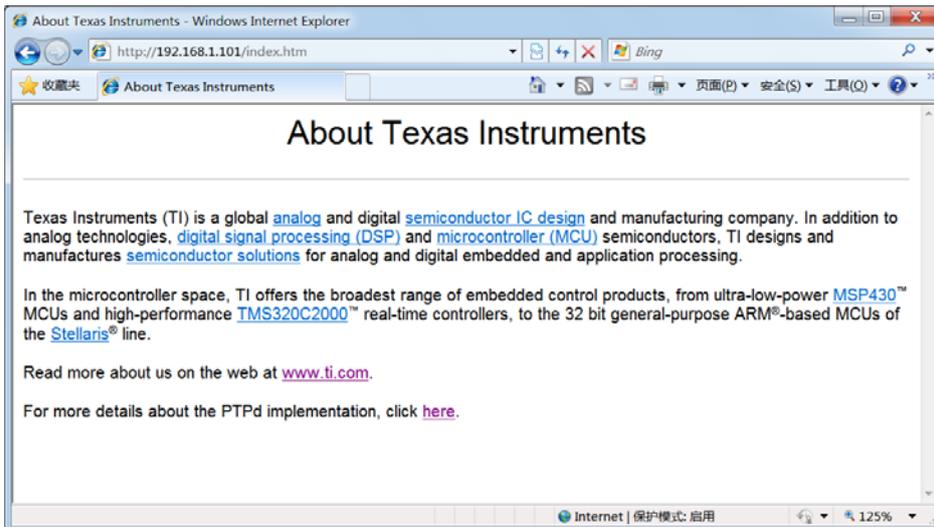


本テストプログラムは lwip ネットプロトコルを元に IEEE1588 プロトコルで同期イーサネット機能を実現します。

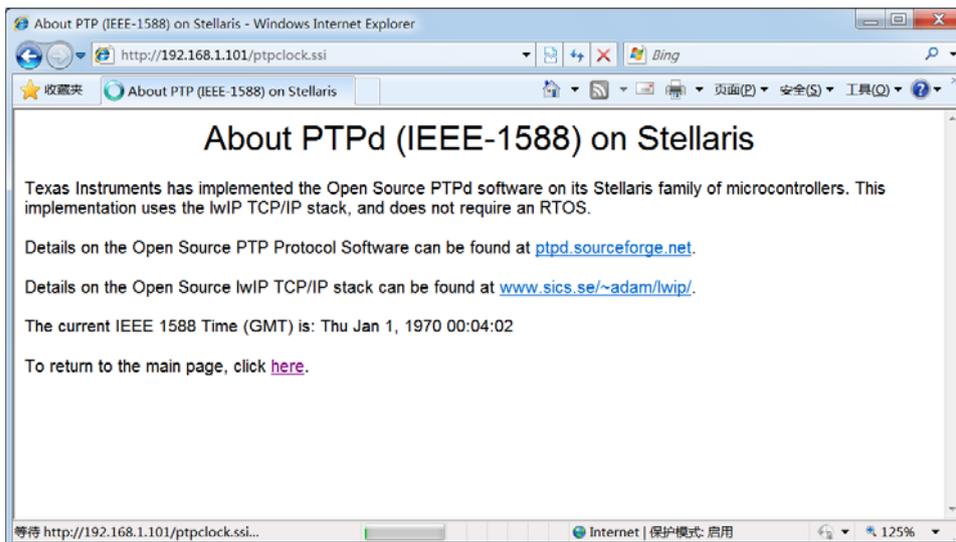
Step1、上記図の様に LAN ケーブルでボードの ethernet インタフェースとルータを繋ぐ。USB-COM と PC を繋ぐ。プログラムをボードにダウンロードして実行すると、ボードが自動配分された IP アドレスがシリアルポートから出力します。赤線の枠内はボード上の日付と時間です。

```
Ethernet with PTPd
MAC Address Not Programmed!
IP: 192.168.1.101
Netmask: 255.255.255.0
Gateway: 192.168.1.1
Thu Jan 1, 1970 00:00:32 (GMT)
```

Step2、IEブラウザに<http://192.168.1.101>を入力するとボードのウェブ内容を見れます：



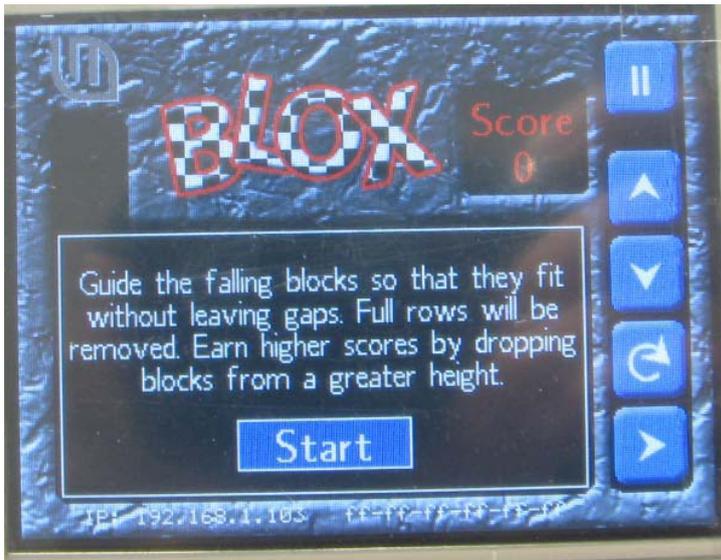
Step3、“For more details about the PTPd implementation, click here”の“here”をクリックしてイーサネット機能テストに入ります。1秒毎に時間を更新します。



4.1.19 ¥qs-blox

本試験はテトリスゲームを実現します。LCDを接続する必要があります。

Step1、「Start」を押してゲーム開始します。



Step2、画面下にある5つのボタンで各種操作できます。



4.1.20 ¥idm-checkout



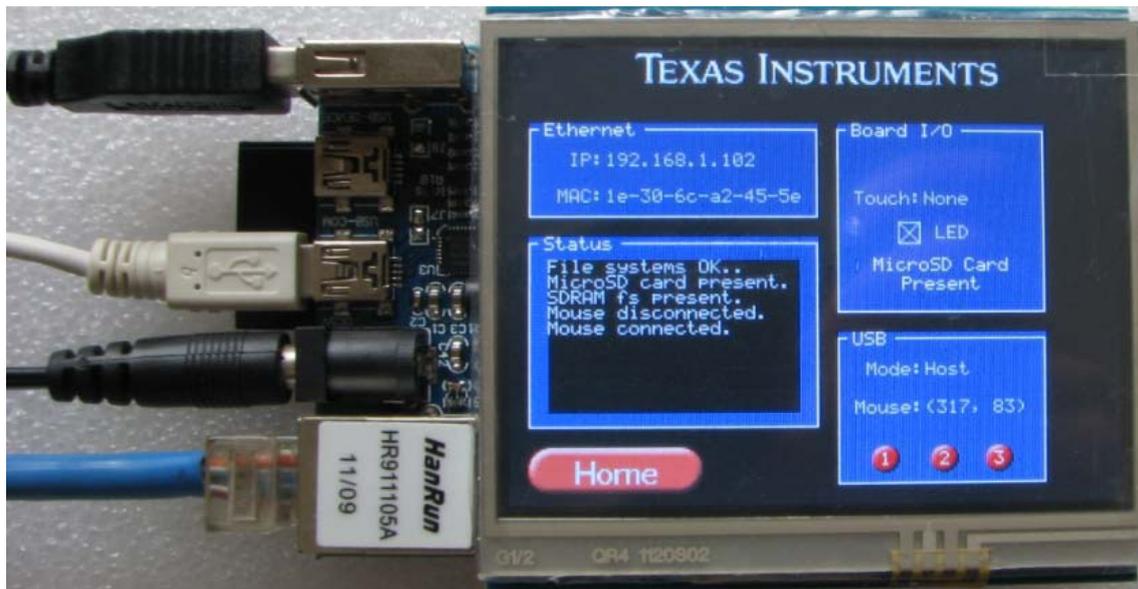
出荷時のデモソースです。

試験前の準備：

- 1、 Tool フォルダにある ramfs_data.bin を PC の C ドライブのルートフォルダにコピーします。
- 2、 Tool フォルダにある*.wav ファイル及び SD Web Sit フォルダ内のファイルを全て SD カードのルートフォルダにコピーします。

Step1、 LANケーブル、 Mini USB(USB-COM)、 USBマウス、 SDカード、 イヤホンをボードの対応インタフェースに接続。プログラムをボードにダウンロードして電源を入れると、四つのテスト項目が表示されます：“IO Examples”、“Graphics Demo”、“Audio Player”、“Image Viewer”。

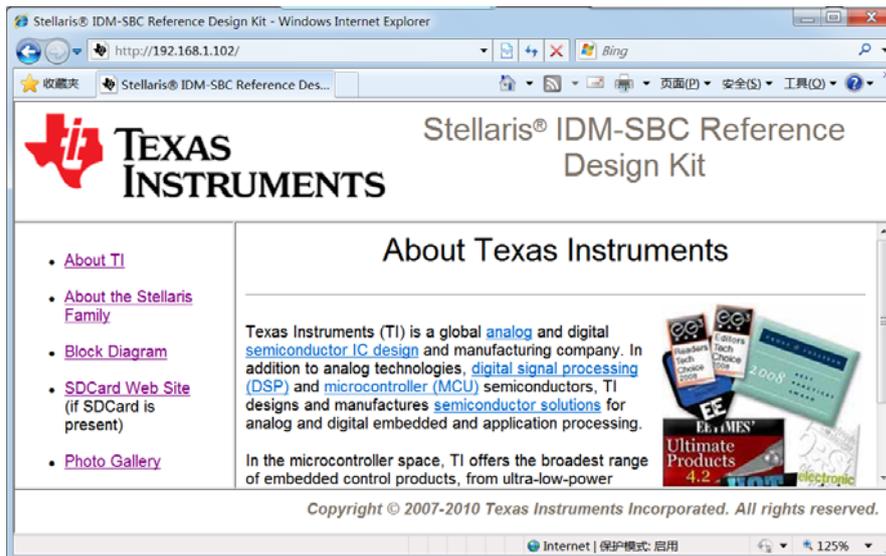
Step2、“IO Examples”をタッチします：



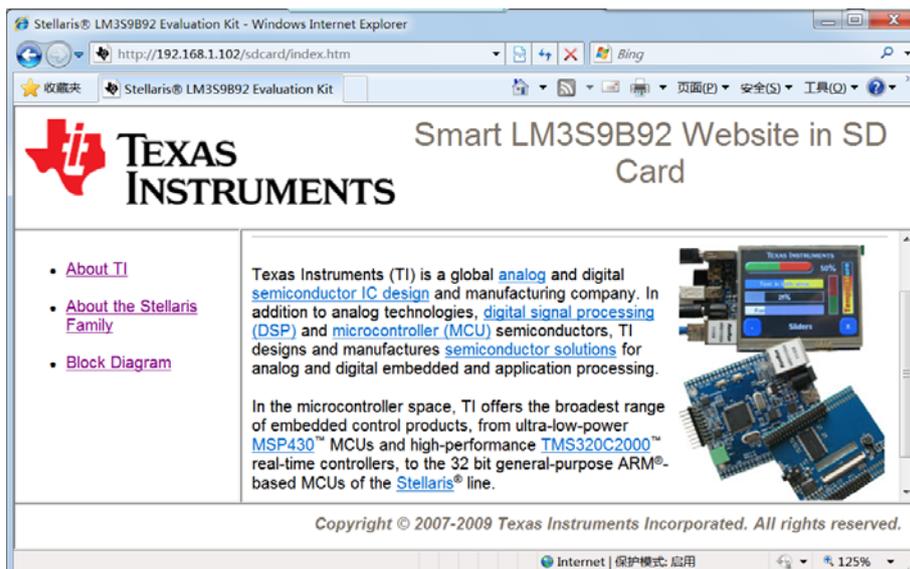
- 1、“Ethernet”項目ではボードに配分されたIPアドレス「192.168.1.102」が表示します。
- 2、“Board I/O”項目の”Touch”は指先がLCDにタッチしている位置の座標を表示します。
- 3、“USB”項目の”Mouse”はマウス移動の座標を表示します。1、2、3ボタンでマウスの三つのボタンの状態を示します。
- 4、“Status”項目欄はファイルシステムを表示します。SDカード、SDRAM上のファイル及びUSBマウスのカレント状態を表示します。
- 5、“SDRAM fs”について、画像ファイルはSPIインタフェースのFlashに保存されていて、電源入れるとFlashからSDRAMにコピーして解凍処理を行います。SPI Flashに何のファイルもない場合は”SDRAM fs absent”と表示します。
- 6、画像ファイル”ramfs_data.bin”はTFTPツールでボードにアップロードできます。WinXPでコマンドウィンドウでCドライブのルートフォルダに入って、“TFTP -i 192.168.1.102 PUT ramfs_data.bin EEPROM”コマンドでファイルをボードのSPI Flashにアップロードします（およそ20秒かかります。Win7ではサポートしません）。

```
C:\> C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\CXP>cd \
C:\>tftp -i 192.168.1.102 put ramfs_data.bin eeprom
Transfer successful: 922820 bytes in 20 seconds, 46141 bytes/s
C:\>_
```

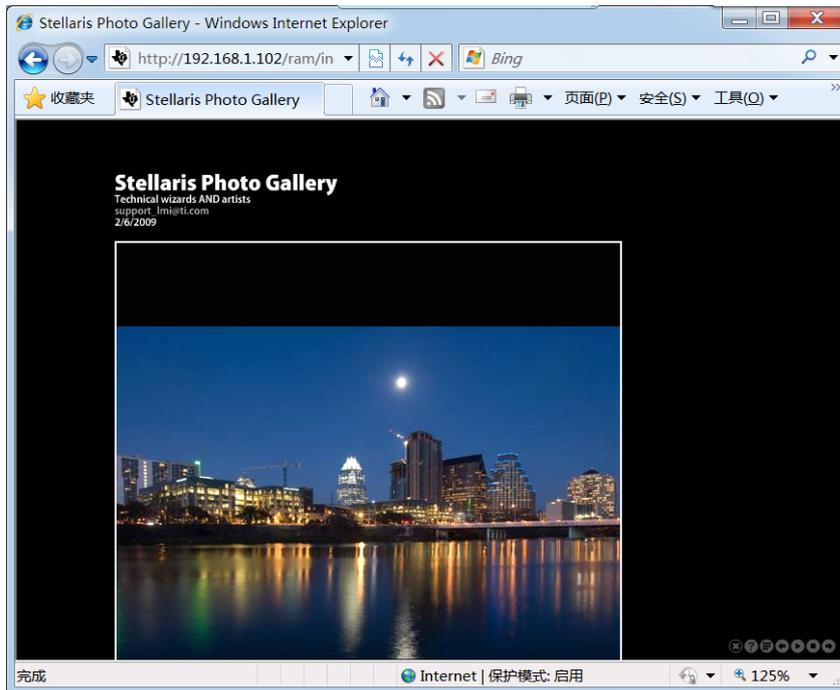
- 7、再度ボードに電源を入れて”Status”項目欄の”SDRAM fs absent”が”SDRAM fs present”に変更します。
- 8、IEブラウザでhttp://192.168.1.102を入力（Ethernet項目に表示しているIPアドレス）してボードのウェブサーバーに入ります：



9、” SDCard Web Site” をクリックして SD カードに保存しているウェブファイルが見えます：

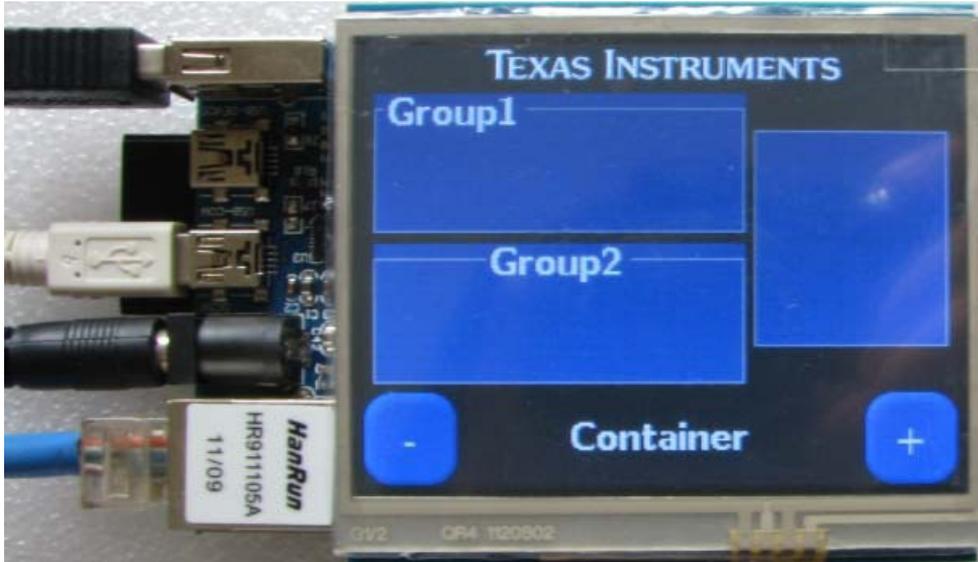
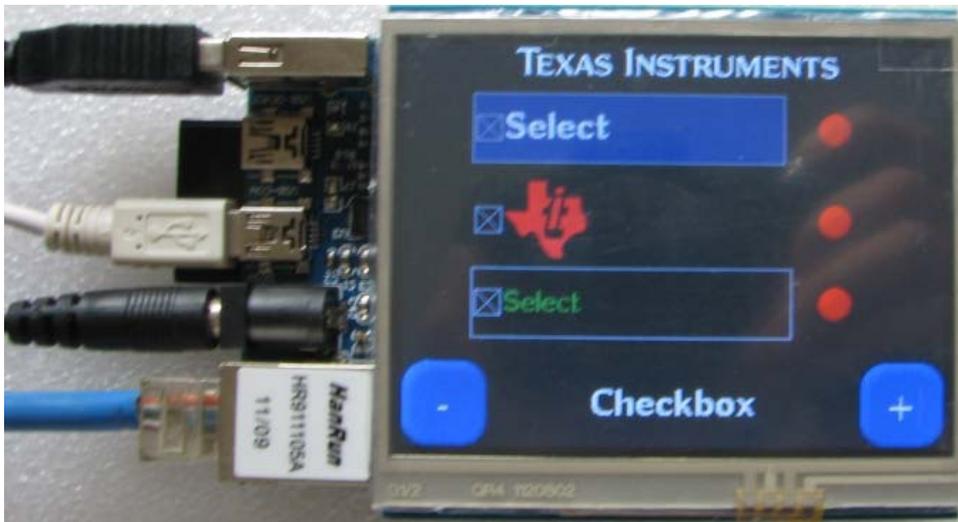
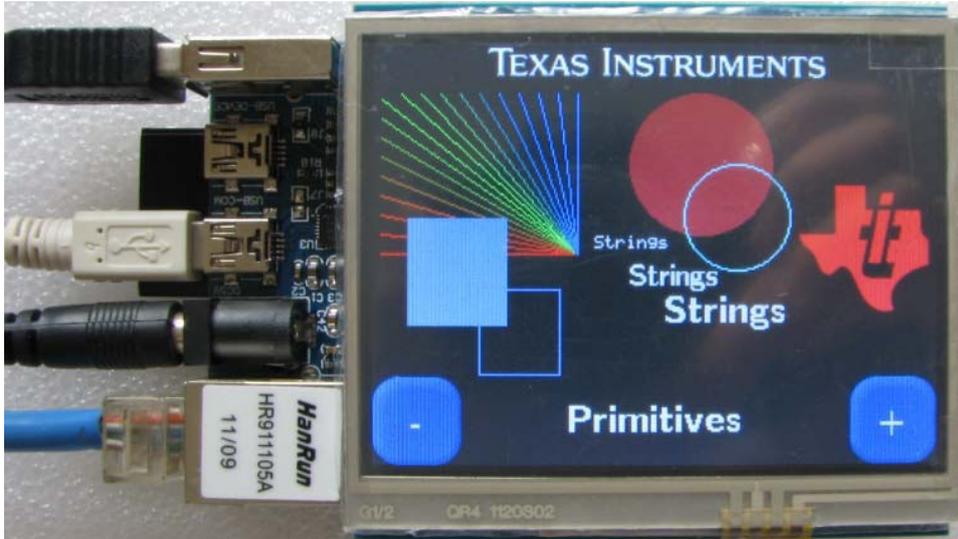


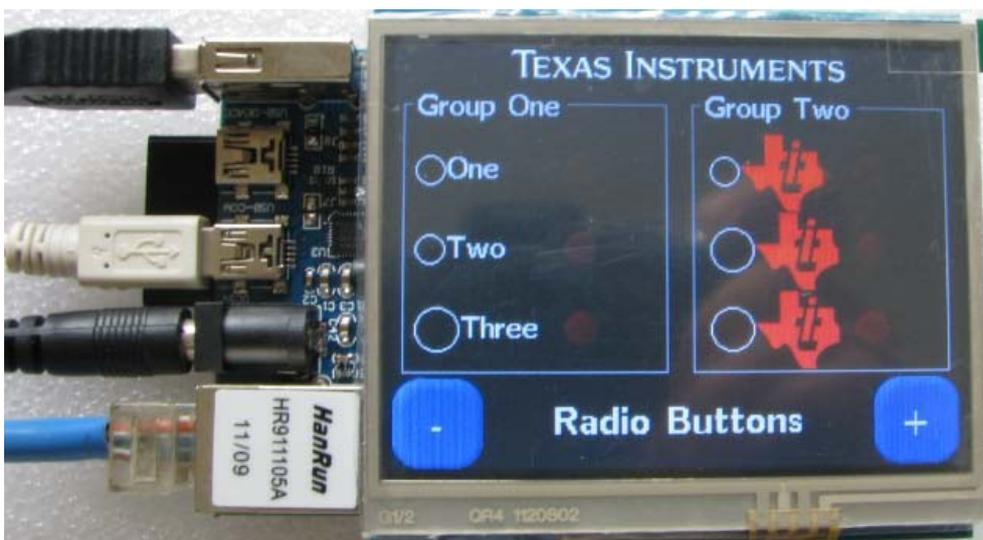
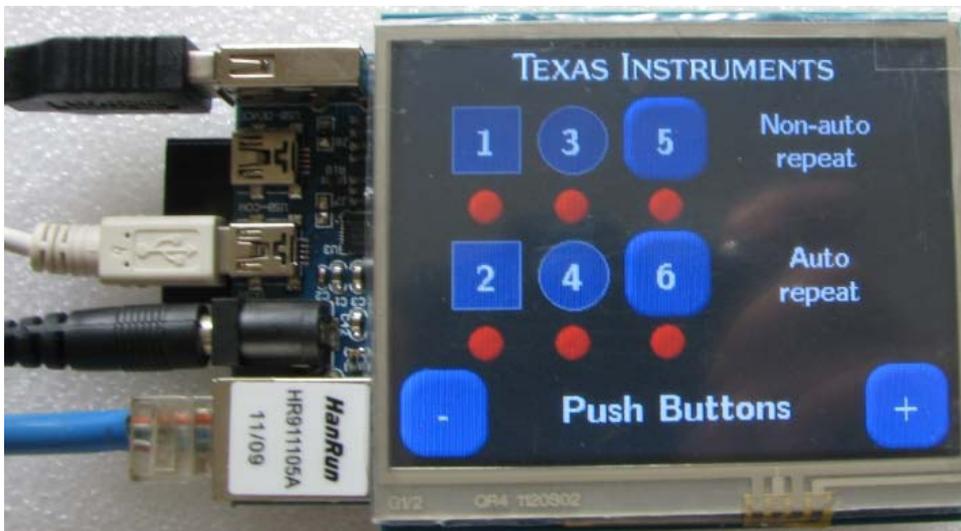
10、8 の画面から” Photo Gallery” をクリックすると、TFTP からアップロードした画像ファイル(ramfs_data. bin)が見えます：

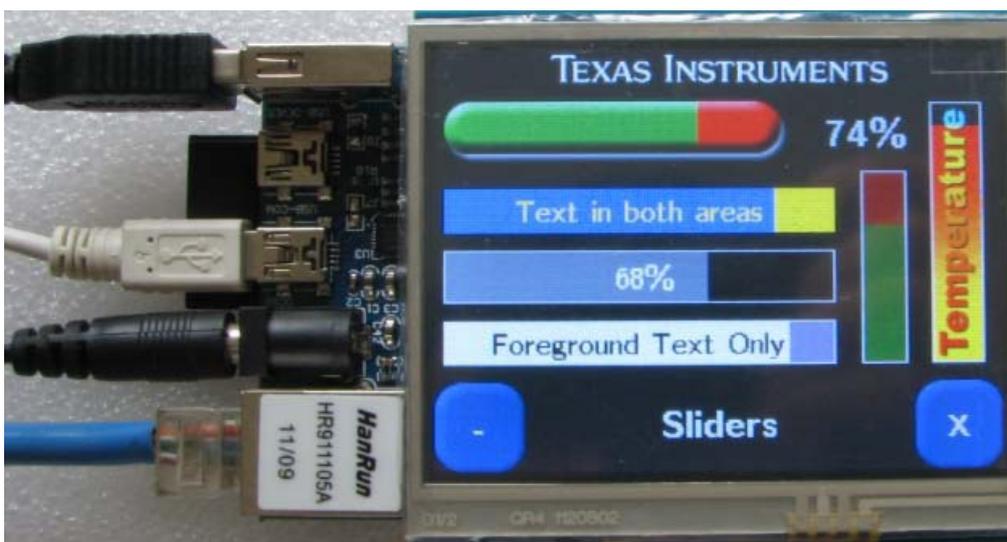
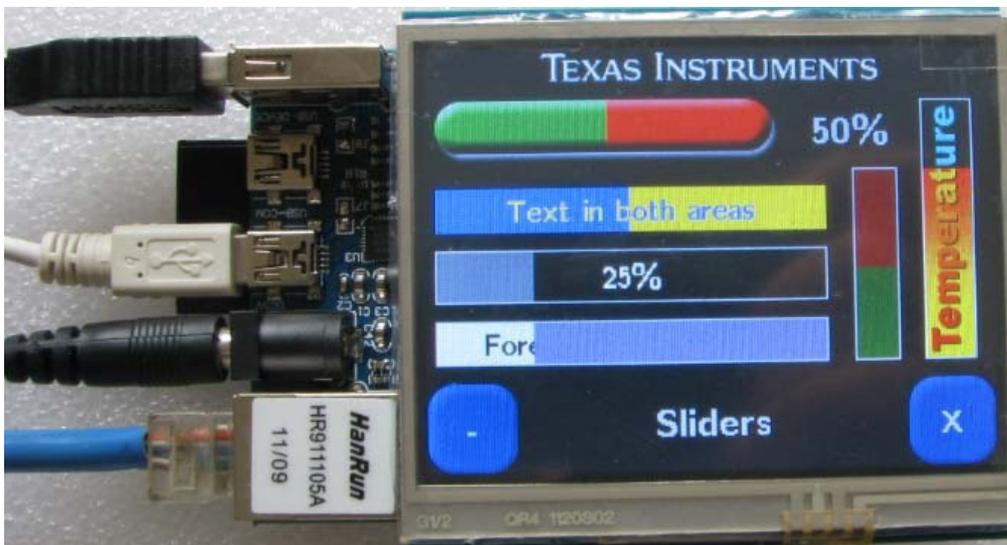
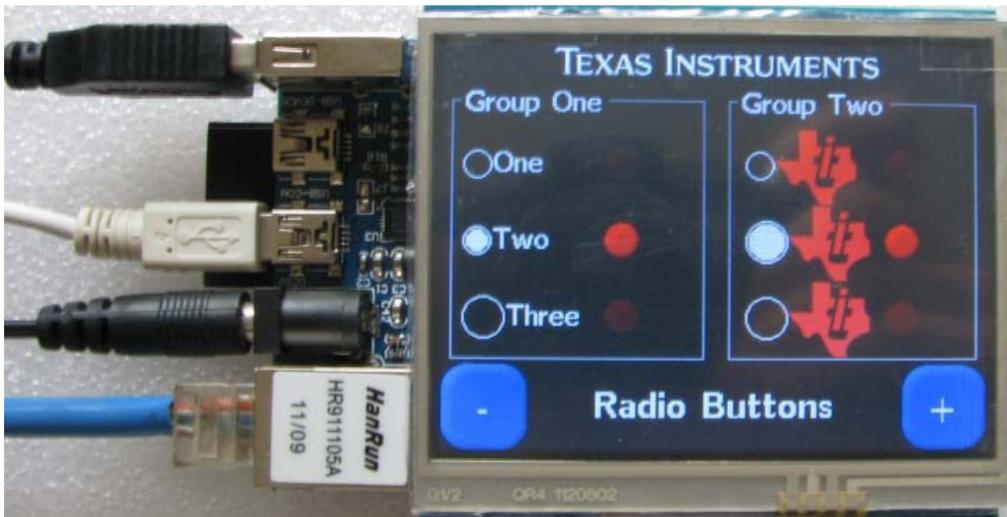


- 11、右下にあるボタン  125%  で閲覧モードを変更できます。
 - 12、” IO Examples” 項目テスト完了後、” Home” をタッチしてメイン画面に戻ります。
- Step3、” Graphics Demo” アイコンをタッチして、TI グラフィック機能をテストします。

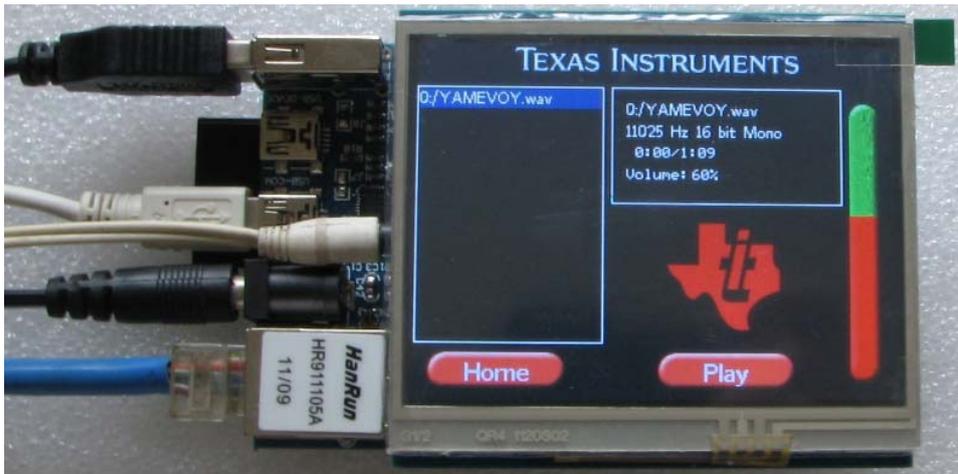




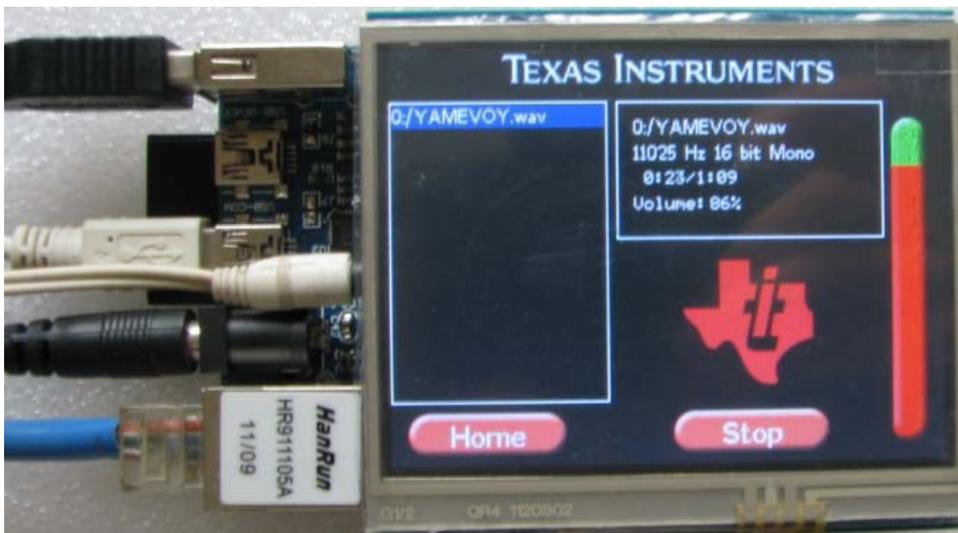




Step4、” Audio Player” アイコンをタッチしてオーディオ機能をテストします。

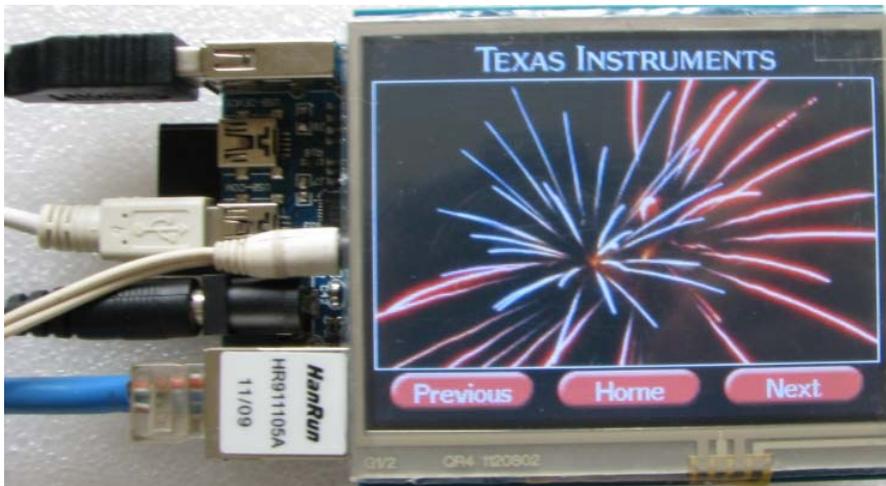


プレイする曲 (.wav ファイル) をタッチして、” Play” をタッチします。右側のボリュームバーをタッチして音量を調整できます。



“Home” をタッチしてメイン画面に戻します。

Step5、” Image Viewer” をタッチして TFTP から SPI Flash にアップロードした画像ファイル (ramfs_data.bin) を見えます。



4.1.21 ¥hello_lcd

本テストは簡単なLCDテストプログラムです。LCD上”Hello World!”を表示します。

4.1.22 ¥hello_widget

本テストは簡単な LCD 及びタッチパネルテストプログラムです。電源を入れると LCD に” Show Welcome” の表示があります。指先でタッチすると下に” Hello World!” が表示される同時に” Show Welcom” が” Hide Welcome” に変更します。

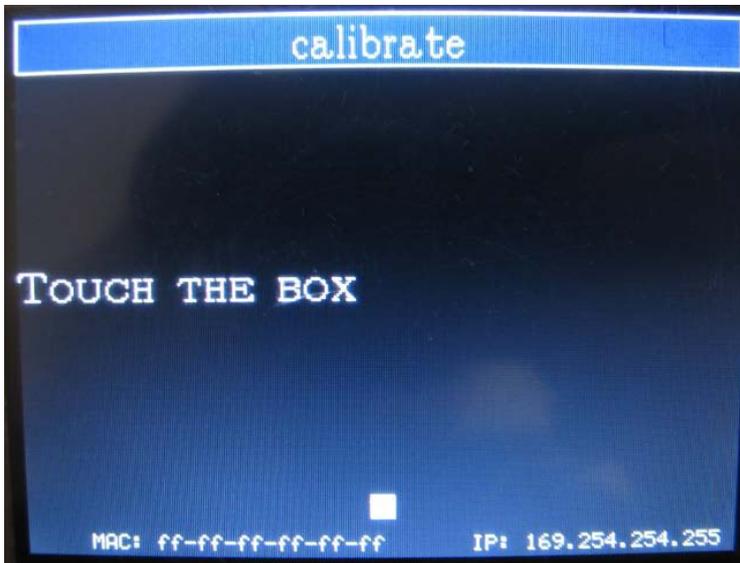
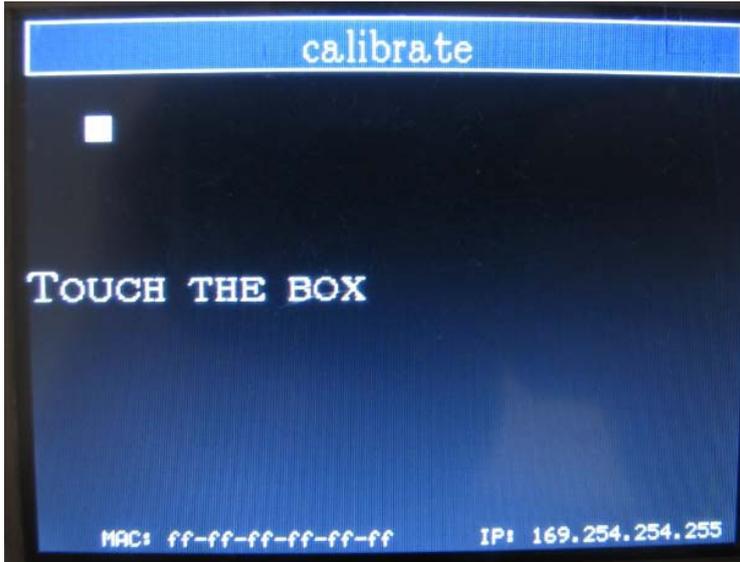


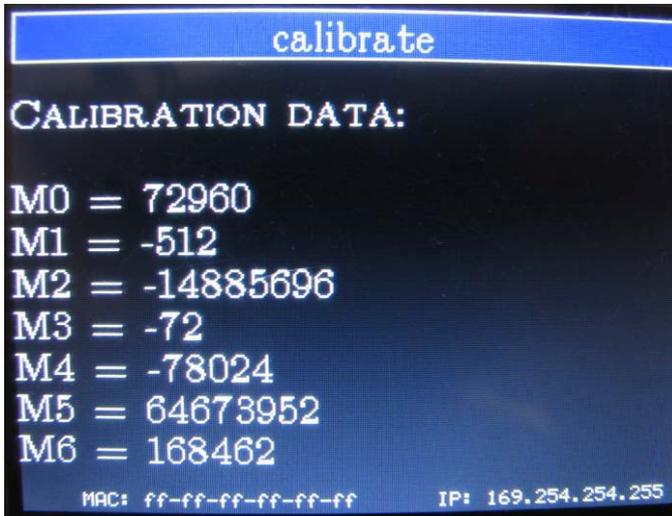
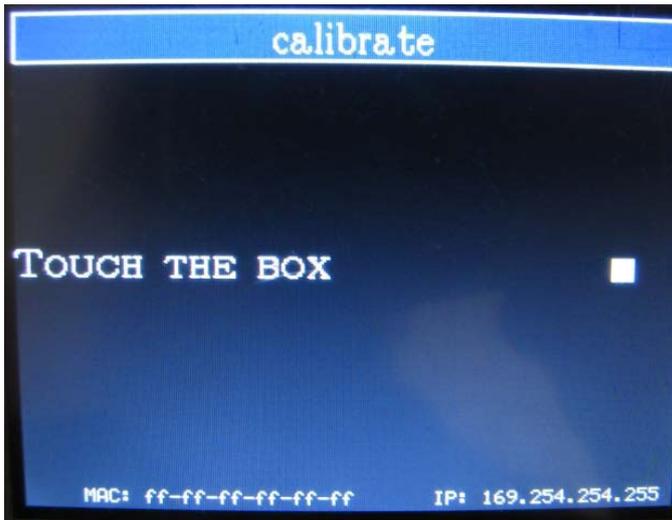
4.1.23 ¥calibrate

本テストはタッチパネルのキャリブレーションプログラムです。LCD の提示通り白い点をタッチして、補正後のデータが表示されます。このデータで元のデータ (M0-M6) に切り替える事でタッチ効果がより精確になります。



```
touch.c  
118 #ifdef LANDSCAPE  
119 #define M0 85888  
120 #define M1 -384  
121 #define M2 -18005888  
122 #define M3 264  
123 #define M4 -76008  
124 #define M5 66659640  
125 #define M6 193184
```





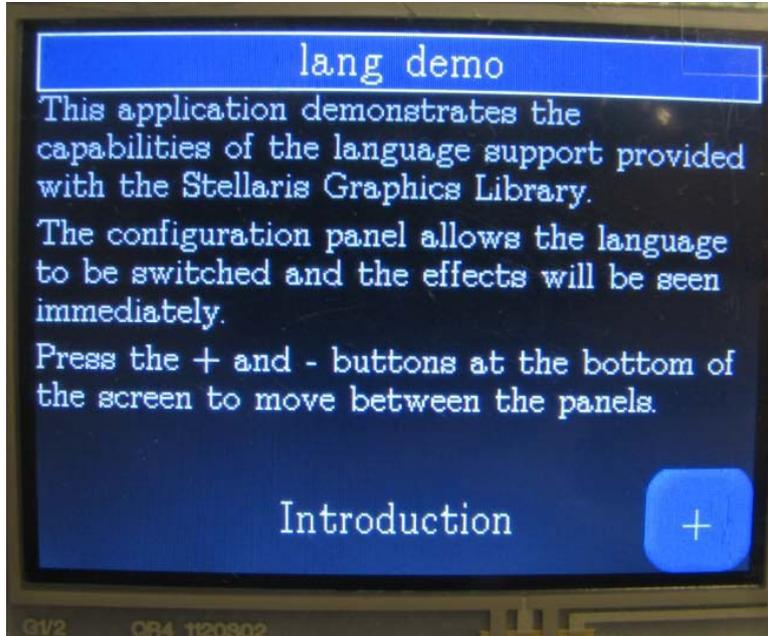
4.1.24 ¥scribble

本テストは描画プログラムです。タッチペンでLCDに画像を描けます。



4.1.25 ¥lang_demo

本テストはグラフィックライブラリのストリングの使用を表示します。



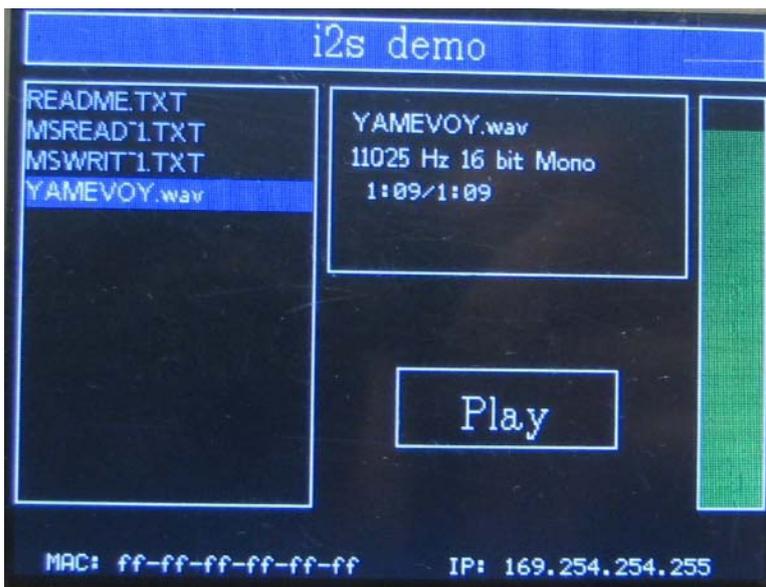
4.1.26 ¥showjpeg

本テストは JPEG 画像を表示します。JPEG 画像ファイルは SDRAM にコピーして解凍処理を行います。画像の表示範囲が LCD より大きい場合、スクロールで移動して閲覧できます。



4.1.27 ¥i2s_demo

本テストはボードのオーディオ機能を実現します。SD カードに保存している.wav ファイルをプレーします。右側のヴォリュームバーを移動して音量を調整できます。



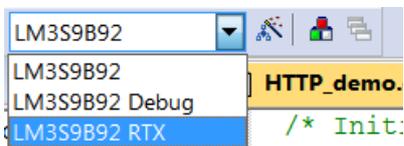
4.2 Code¥ek-1m3s9b92-C3¥RL¥TCPnet¥

※MCU が C1 のバージョンの場合は Code¥ek-1m3s9b92¥RL¥TCPnet¥をご参照ください。



この図は C5 バージョンです。

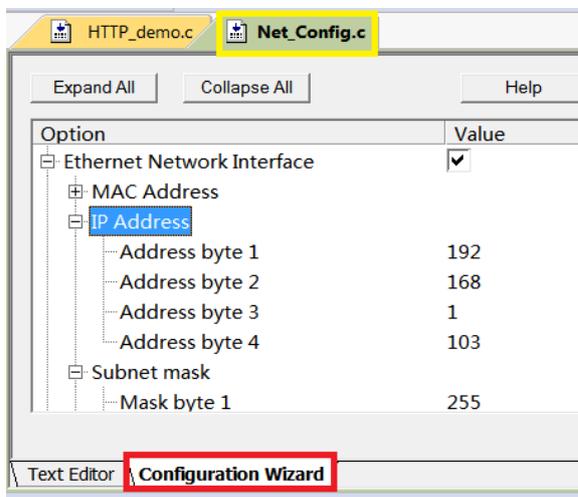
このフォルダにあるサンプルソースは Keil の lib で提供しているネットプロトコルを利用しています。一部のプログラムでは RTX の OS モードを選択できます。



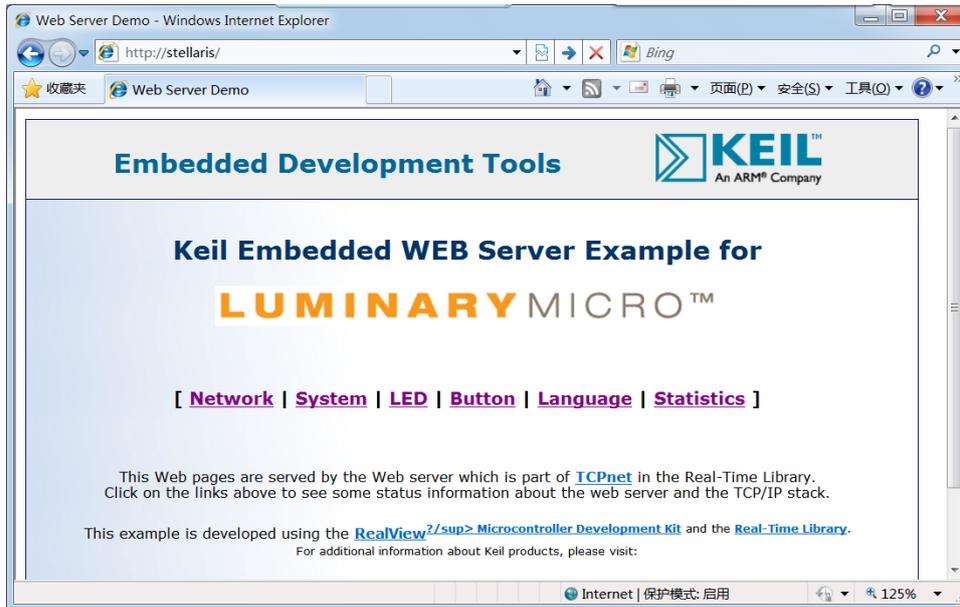
4.2.1 HTTP_demo

本テストは http サーバープログラムです。ウェブからボード上の LED 制御、キー状態の取得、ネット接続状態の取得などが出来ます。

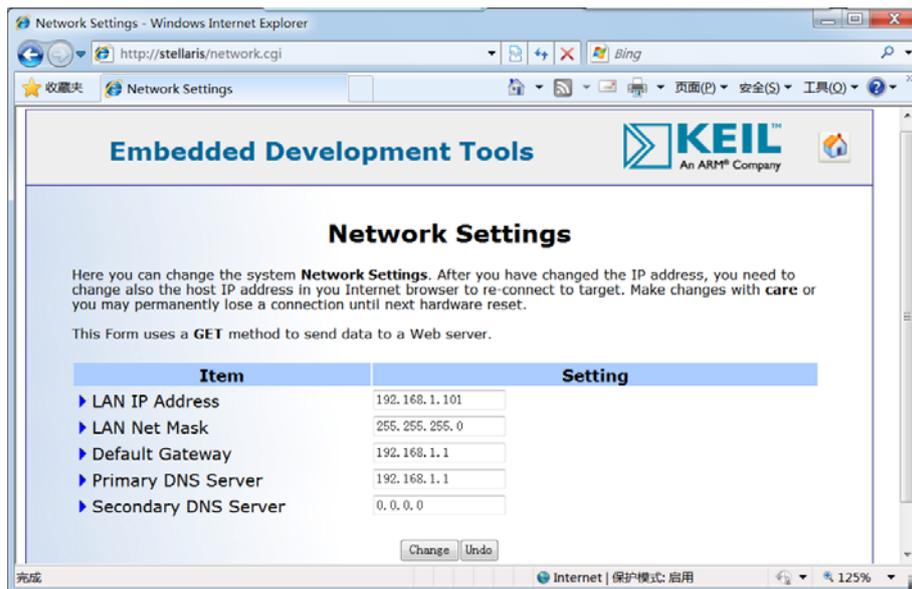
Step1、デフォルトでは DHCP 機能でルータから自動で IP アドレスを取得しますが、取得できない場合は Net_Config.c に設定している固定 IP アドレスを利用します。



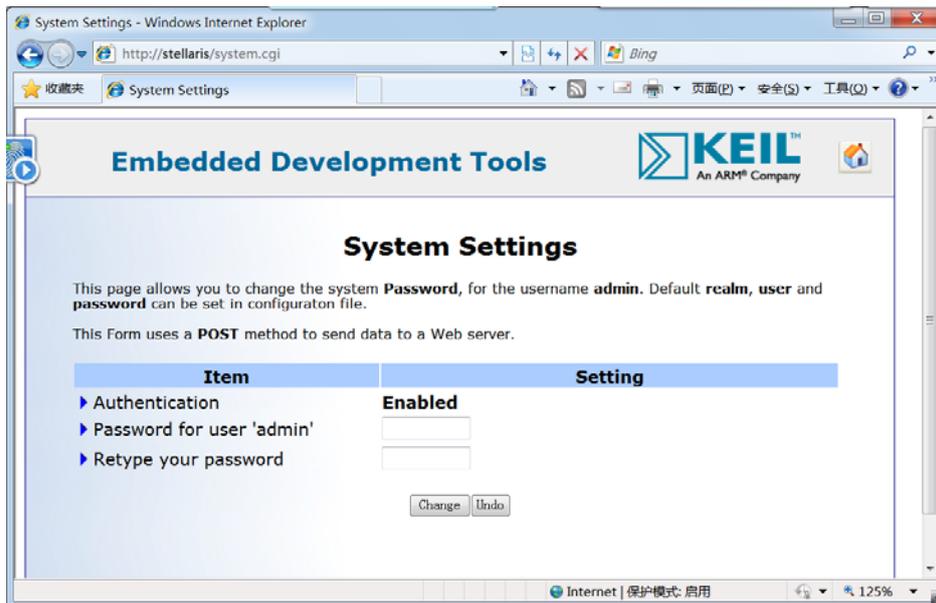
Step2、ブラウザで <http://stellaris> を入力するとボードのウェブに登録できます。ユーザー名は admin、パスワードはないです。



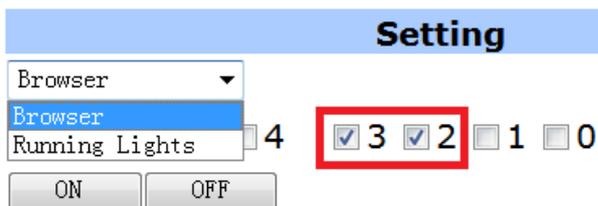
Step3、” Network” をクリックしてボードのネット関連設定を確認及び修正ができます：



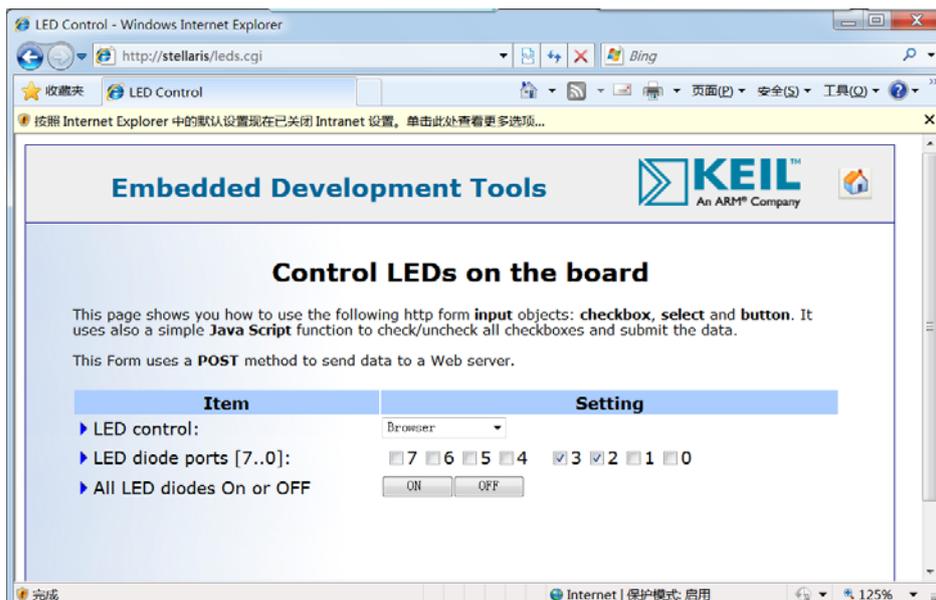
Step4、” System” をクリックして登録時のユーザー名とパスワードを修正できます：



Step5、” LED” をクリックして LED 制御画面に入ります。

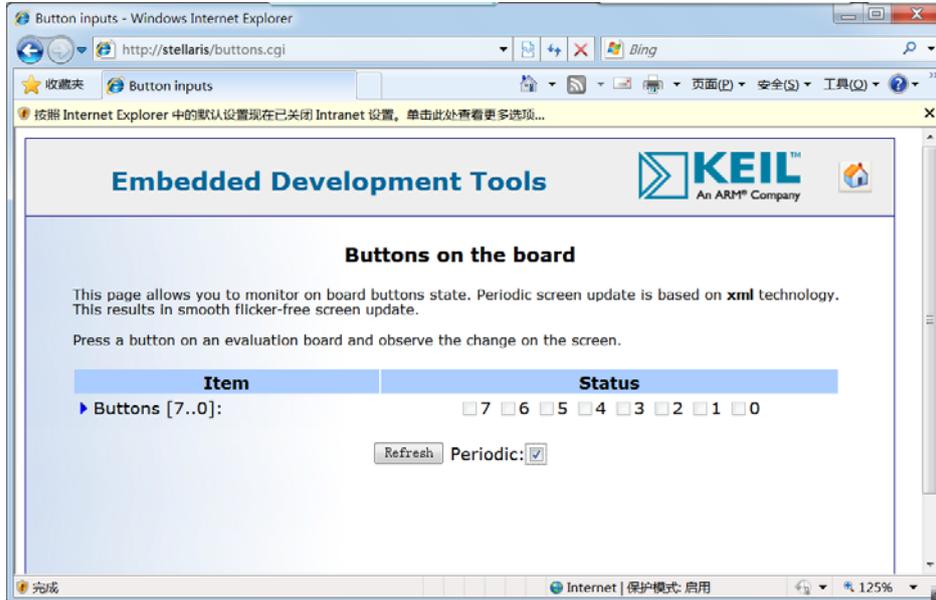


Running Lights は点滅モード。Brower は一個づつ LED の点灯/消灯を制御できます。Brower モードを選択して、2 と 3 の所にチックを入れる或いは消す事で RJ45 の緑色と黄色の LED を点灯/消灯させます。



Step6、” Button” をクリックしてボードのキーの状態を制御できます。SW キーは 0 位と対

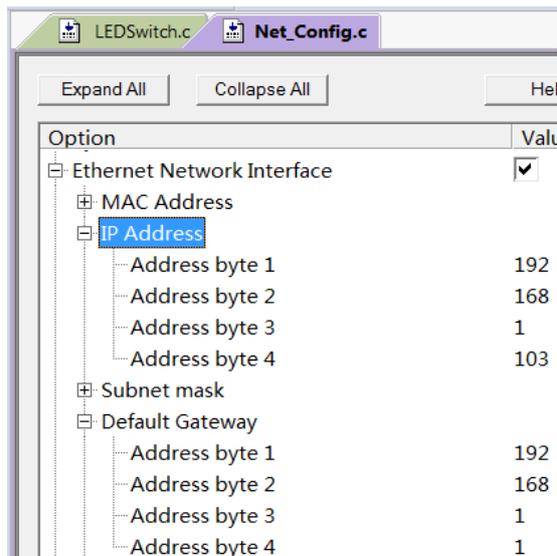
応じていて、押下時にチェックが表示されます。



4.2.2 ¥LEDSwitch

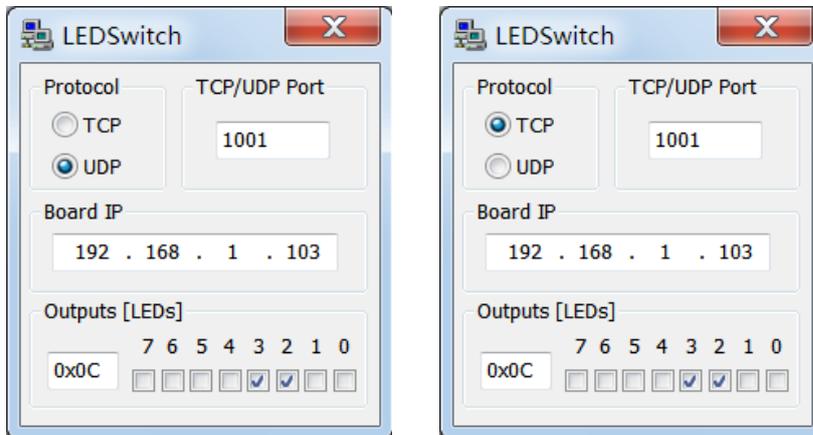
本プログラムはUDP及びTCPポートで¥Code¥PC-SoftWare¥LEDSwitch¥ReleaseフォルダにあるLEDSwitch.exeから発送したコマンドを検出して処理を行います。LEDSwitch.exeからボードのLEDを点灯或いは消灯制御できます。¥Code¥PC-SoftWare¥LEDSwitchフォルダにはVC++ソースコードです。

Step1、手動でボードのIPアドレスとGatewayを変更します。ボードとPCを同じIPアドレスセグに設定する必要があります。Net_Config.cファイルを編集します。



Step2、LANケーブルでルータとボードを繋ぐ。プログラムをボードにダウンロードして、LEDSwitch.exeを実行します。“TCP/UDP Port”を1001に設定し、“Board IP”を

192.168.1.103に設定します。“Outputs[LEDs]”の2-3位の所にチェックを入れるとボードの対応LEDが点灯、消灯します。



4.2.3 ¥DNS_demo

本プログラムではサーバー名でサーバーの IP アドレスを解析します。5 秒毎に以下のサーバーを解析します：

www.google.com

www.keil.com

www.microsoft.com

www.yahoo.com

dns_cbfunc関数で解析結果を判断して、解析情報をシリアルポートから出力します。



```
DNS Resolver Demo Example
Resolving host: www.google.com
IP Address      : 74.125.71.104

Resolving host: www.keil.com
IP Address      : 217.140.108.95

Resolving host: www.microsoft.com
IP Address      : 207.46.19.190

Resolving host: www.yahoo.com
IP Address      : 98.137.149.56

Resolving host: www.notexisting.site
IP Address      : 220.250.64.24

Resolving host: www.google.com
IP Address      : 74.125.71.104

Resolving host: www.keil.com
IP Address      : 217.140.108.95
```

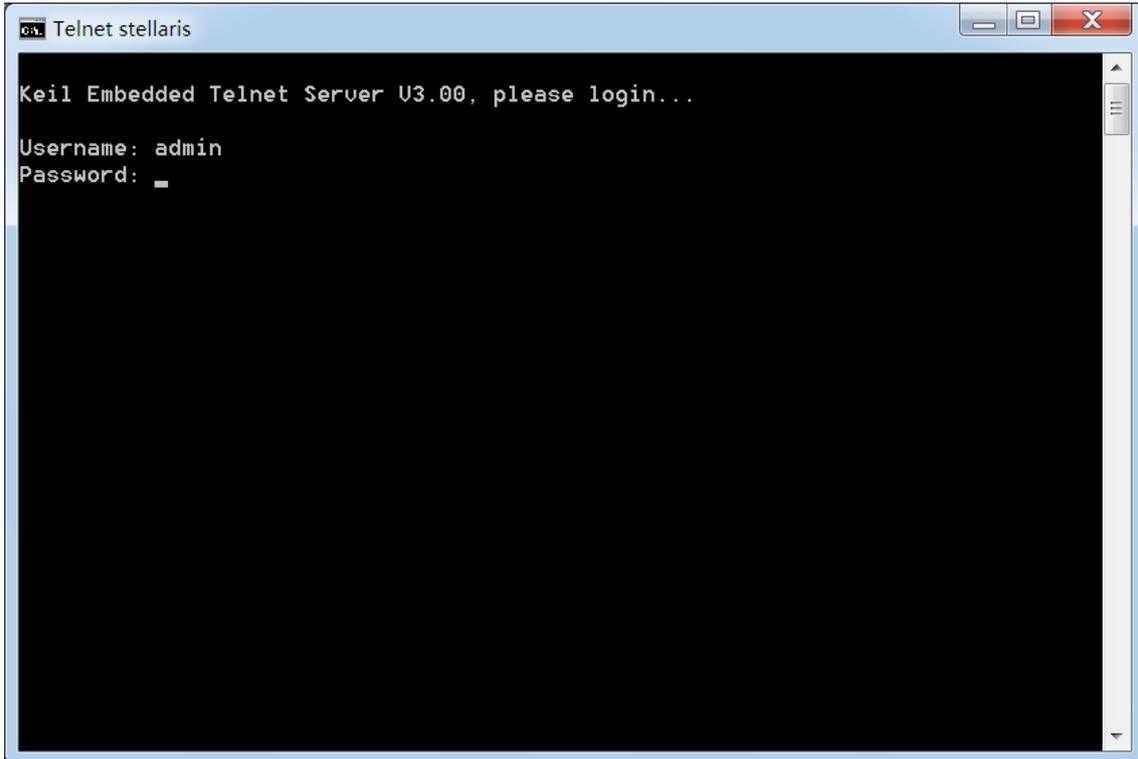
4.2.4 ¥Telnet_demo

本テストはTelnet関連アプリのプログラムです。Telnetソフトでボードを制御できます。

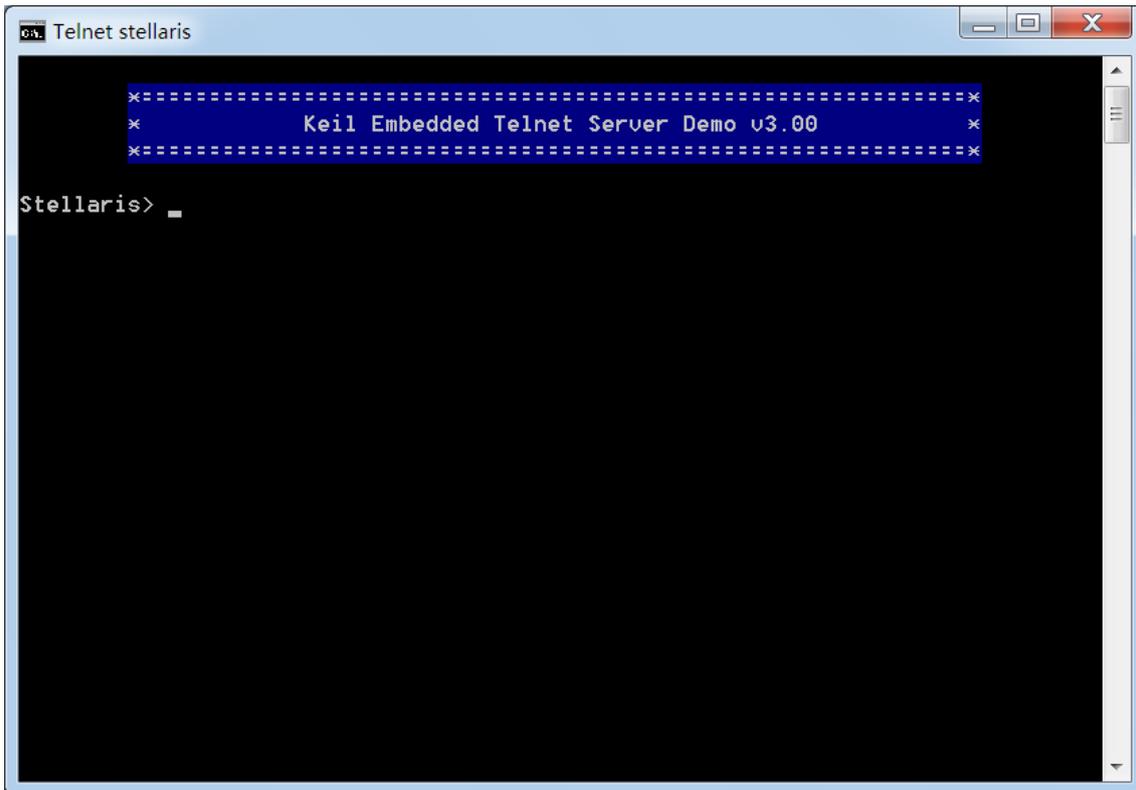
Step1、LANケーブルを繋いで電源をいれる。プログラムをダウンロードします。

Step2、PC側でコマンドウィンドウを開いて（CMD.exe）、telnet stellarisを実行します。

ユーザー名はadmin、パスワード無し。

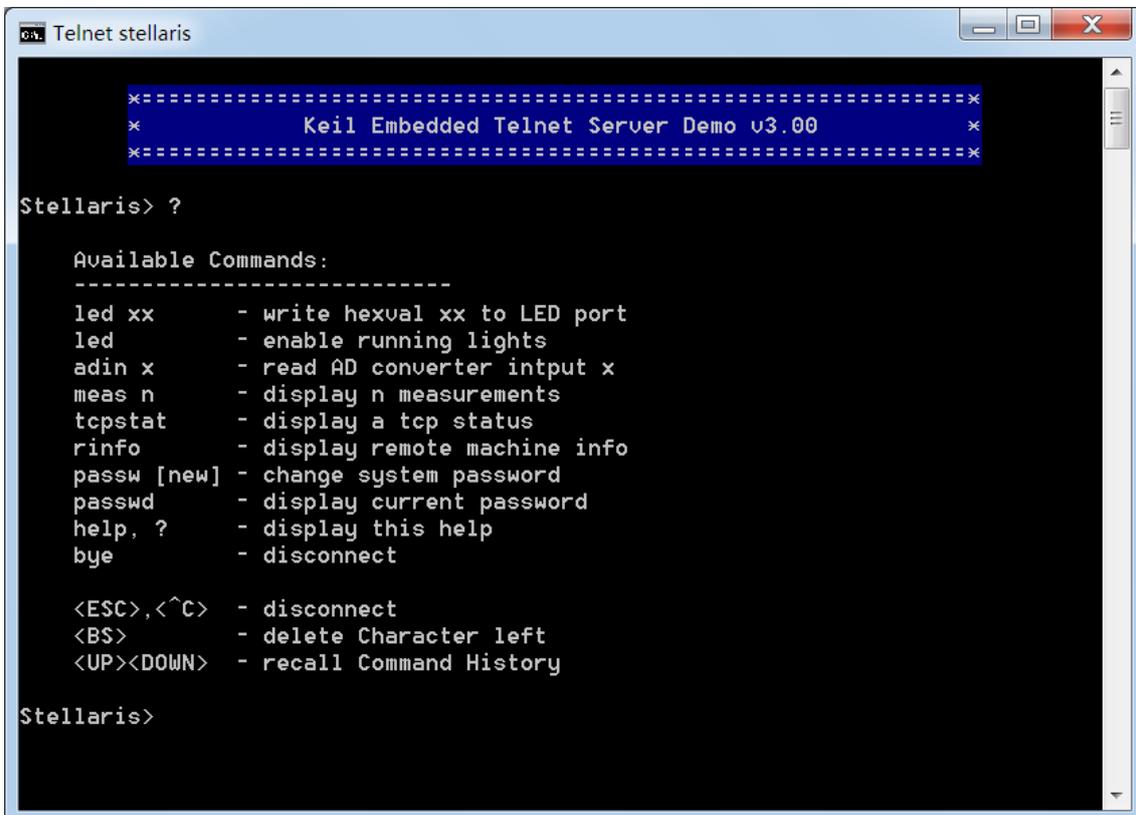


```
Keil Embedded Telnet Server U3.00, please login...
Username: admin
Password: _
```



```
ca. Telnet stellaris
*====*
* Keil Embedded Telnet Server Demo v3.00 *
*====*
Stellaris> _
```

Step3、?を入力してEnterキーを押す。提供している機能を確認します。



```
ca. Telnet stellaris
*====*
* Keil Embedded Telnet Server Demo v3.00 *
*====*
Stellaris> ?

Available Commands:
-----
led xx      - write hexval xx to LED port
led         - enable running lights
adin x     - read AD converter input x
meas n     - display n measurements
tcpstat    - display a tcp status
rinfo      - display remote machine info
passwd [new] - change system password
passwd     - display current password
help, ?    - display this help
bye        - disconnect

<ESC>,<^C> - disconnect
<BS>      - delete Character left
<UP><DOWN> - recall Command History

Stellaris>
```



Step4、led 0x0Cを入力してEnterを押すと二つのLED共消灯させます。Led 0x08を入力してEnterを押すと一つのLEDが点灯します。

```
Telnet stellaris
<ESC>,<^C> - disconnect
<BS>       - delete Character left
<UP><DOWN> - recall Command History

Stellaris> ?

Available Commands:
-----
led xx      - write hexval xx to LED port
led         - enable running lights
adin x     - read AD converter input x
meas n     - display n measurements
tcpstat    - display a tcp status
rinfo     - display remote machine info
passwd [new] - change system password
passwd     - display current password
help, ?   - display this help
bye       - disconnect

<ESC>,<^C> - disconnect
<BS>       - delete Character left
<UP><DOWN> - recall Command History

Stellaris> led 0x0C --> Running Lights OFF
Stellaris> led 0x00
Stellaris> led 0x08
Stellaris>
```



付録：OpenLink で実行ファイルの書き込む

弊社は **OpenLink** のハードウェアを提供しております（製品紹介 URL：<http://www.csun.co.jp/SHOP/2009121901.html>）。

1 ドライバのインストール

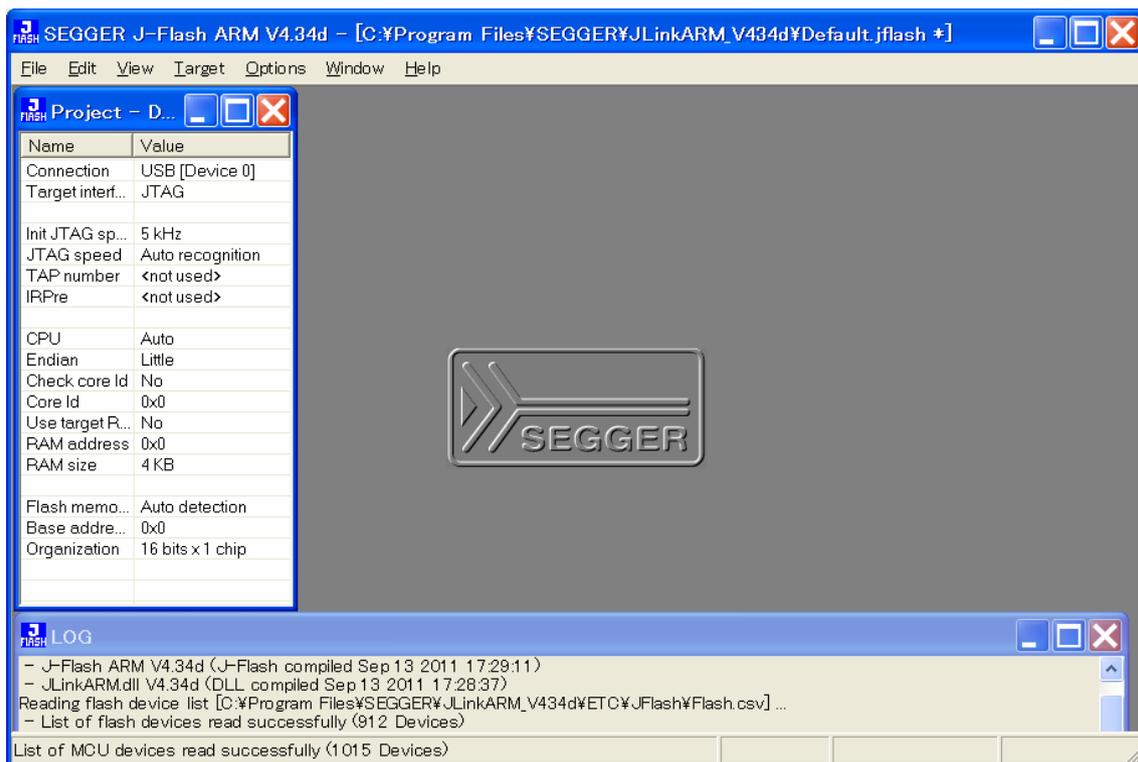
SEGGER 社様のソフトウェアを利用しておりますので、直接 SEGGER 社様ホームページから最新の USB ドライバをダウンロードしてください。

<http://www.segger.com/cms/jlink-software.html>

インストールの際に、ダウンロードした ZIP ファイルを解凍し、デフォルトのままで行ってください。

2 J-FLASH ARMで実行ファイルを書き込む

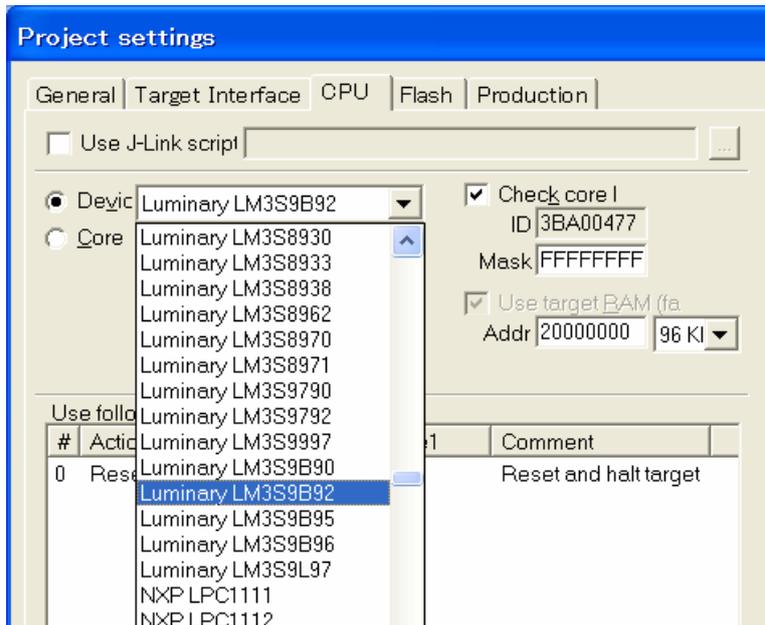
J-FLASH ARM を実行する。



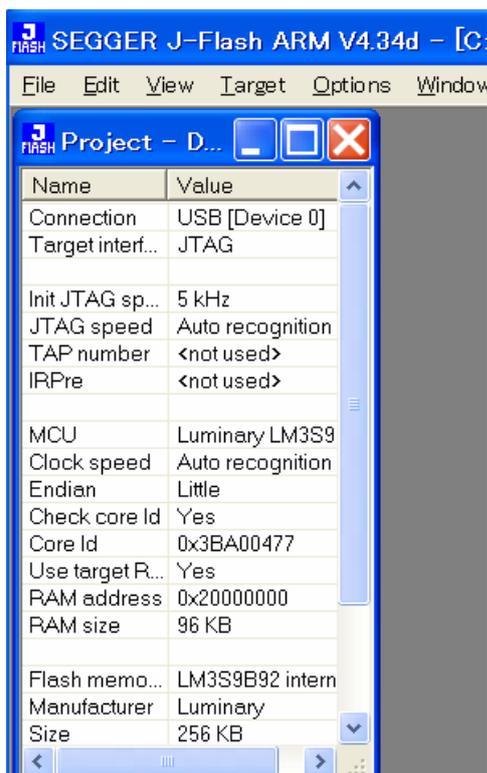
書き込む前に必要な設定 (Options->project settings...) :



ボードの CPU 型番を選択する。



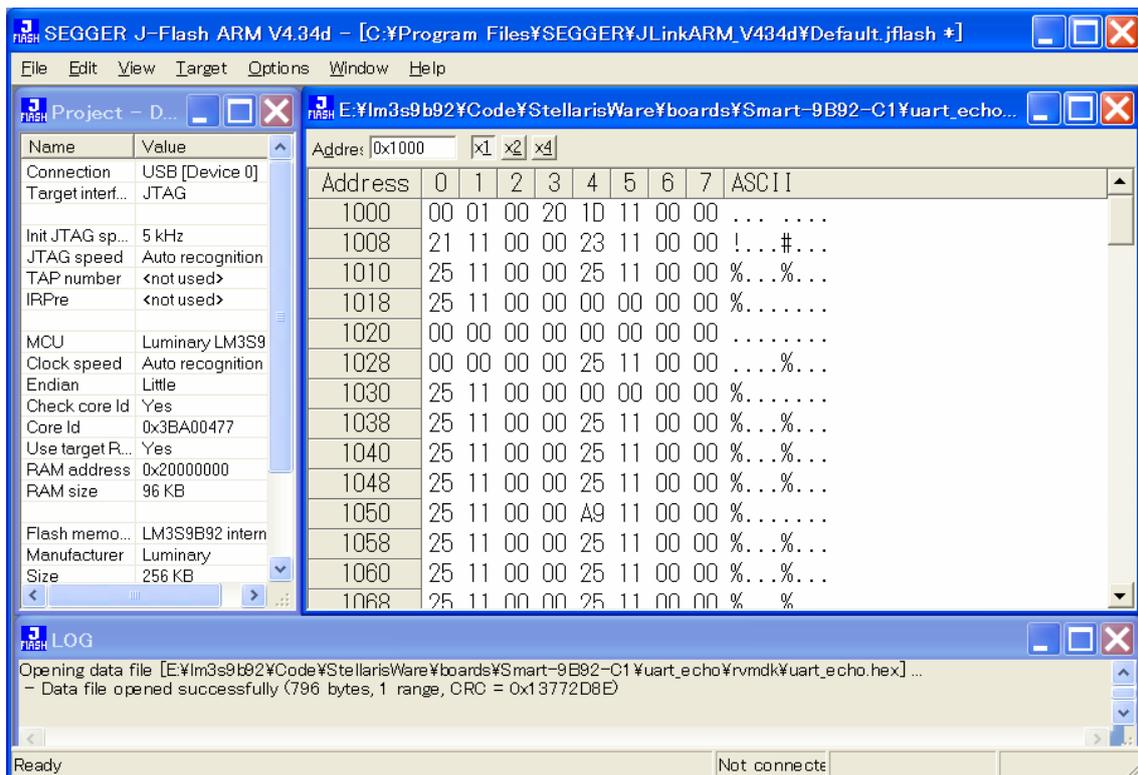
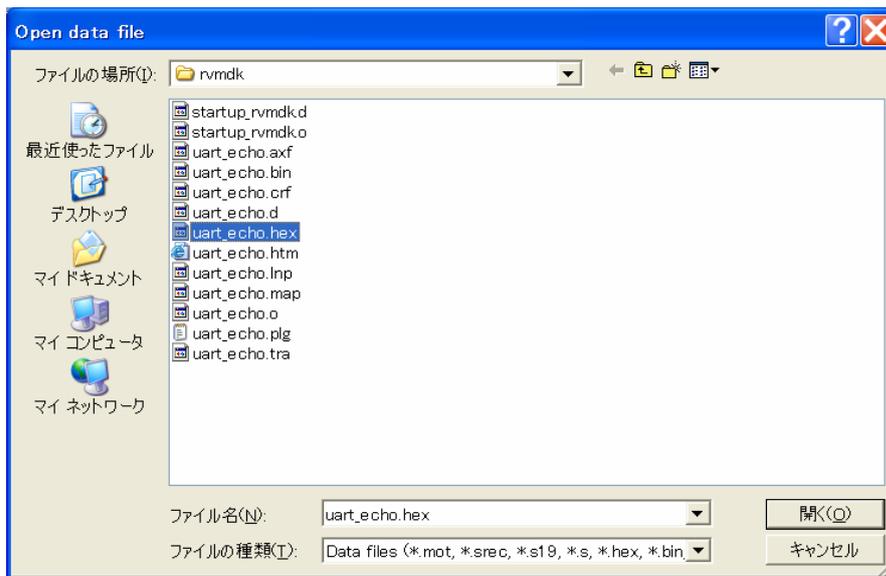
設定後、左側に書き込み情報が表示される。



ボードを接続する。

ソフト側も接続する (Target→Connect)。

File→open で実行ファイルを選択する。



Target->Auto あるいは F7 で書き込み開始する。

書き込み完了 :

