

ARM Cortex-M3 STM32F407

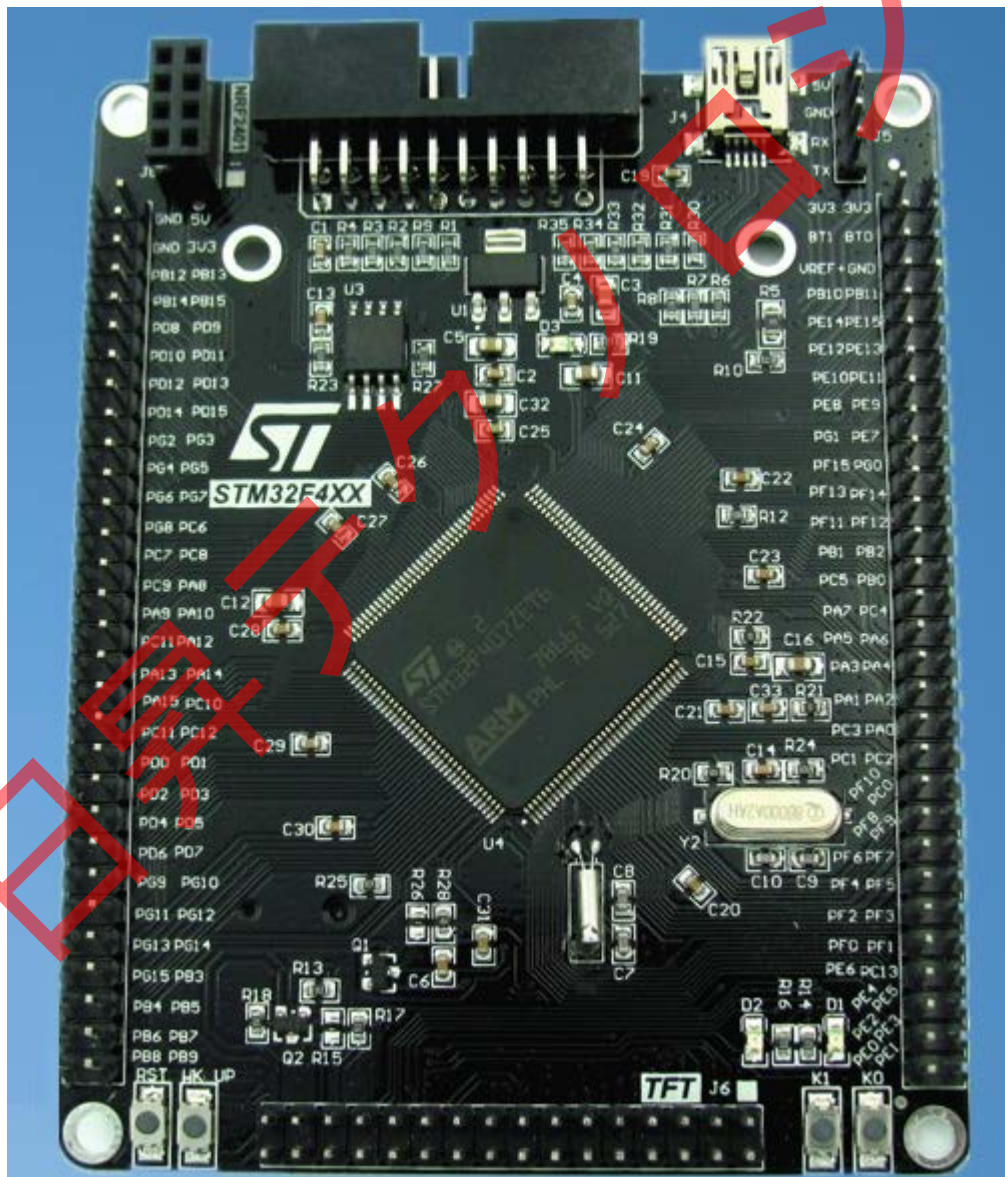
サンプルソース説明書

株式会社日昇テクノロジー

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

info@csun.co.jp

2015/12/22



copyright©2015



• 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2015/12/22

※ 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。

日昇テクノロジー



目 次

(1) Template	5
(2) WaterLED	5
(3) KEY	5
(4) UART	5
(5) INT	7
(6) Watchdog	7
(7) WinWatchdog	7
(8) Timer	8
(9) PWM	8
(10) Input	8
(11) TFT LCD	8
(12) USMART	9
(13) RTC	10
(14) RandomGen	11
(15) ADC	11
(16) Temperature	12
(17) DAC	12
(18) PWM DAC	13
(19) DMA	13
(20) SPI	13
(21) TouchPanel	14
(22) NRF24L01	14
(23) FLASH2EEPROM	15
(24) SRAM	16
(25) Memory	16
(26) SD	17
(27) FATFS	18
(29) IMAGE	19
(30) AVPlay	20
(31) FPU(Julia)	22
(32) DSP	23
(33) HandWriter	25
(35) USBCardReader(Slave)	26
(37) UCOSII1-1-EventCall	27
(38) UCOSII-1-2-EventOther	27



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

(39) UCOSII-2-SemaphoreMailbox 27

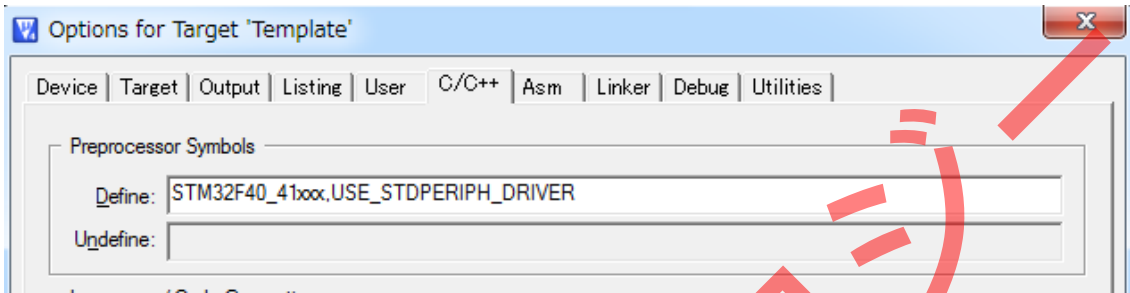
日昇テクノロジー株式会社

(1) Template

このサンプルはプロジェクトを新規作成時ご参考ください。

ベースはSTM32F4 のファームウェアの V1.3.0 のプロジェクト。

注意事項：プロジェクトを新規作成時、Option for target xxx の C/C++ タグの全てのマクロ定義の Define の所は STM32F40_41xxx, USE_STDPERIPH_DRIVER にする必要。



(2) WaterLED

このサンプルはSTM32F407開発ボードの2つのLED（DS0とDS1）をコントロールし、交替的に点滅する。

(3) KEY

このサンプルはSTM32F407 開発ボードの2つのボタン（KEY0、KEY1）を通じて、ボードの2つのLED（DS0 と DS1）をコントロールする。

KEY1 ボタンは DS1 をコントロールし、一回押すと点灯、再度押すと消灯。

KEY0 ボタンは DS0 をコントロールし、一回押すと点灯、再度押すと消灯。

(4) UART

このサンプルではシリアルポート1はメッセージをPCに送信し続ける、同時にシリアルポートからデータを受信して、受信したデータをPCに送信する。

注意：シリアルポートのボーレートを115200bpsに設定する。

ハードウェア：

STM32F407 開発ボード

[RS232C-TTL レベル変換基板](#)

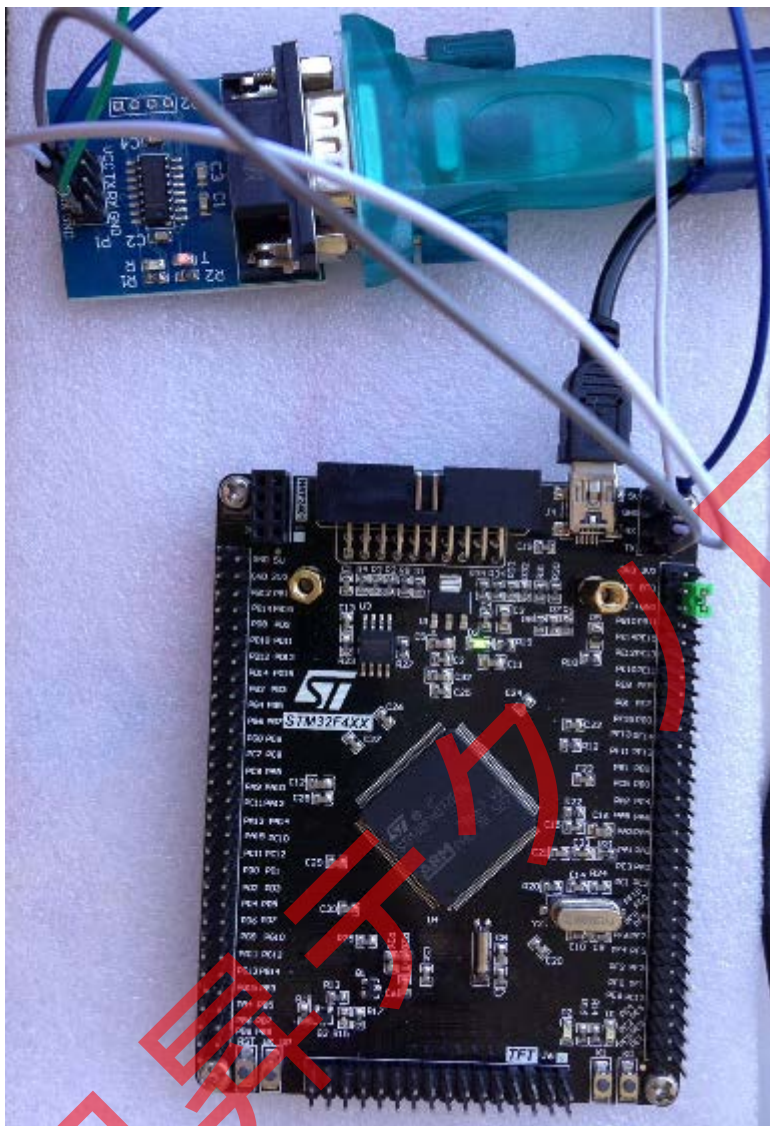
[USB RS232 変換ケーブル\(D サブオス\)](#)

[4 ピン配列変換ケーブル](#)

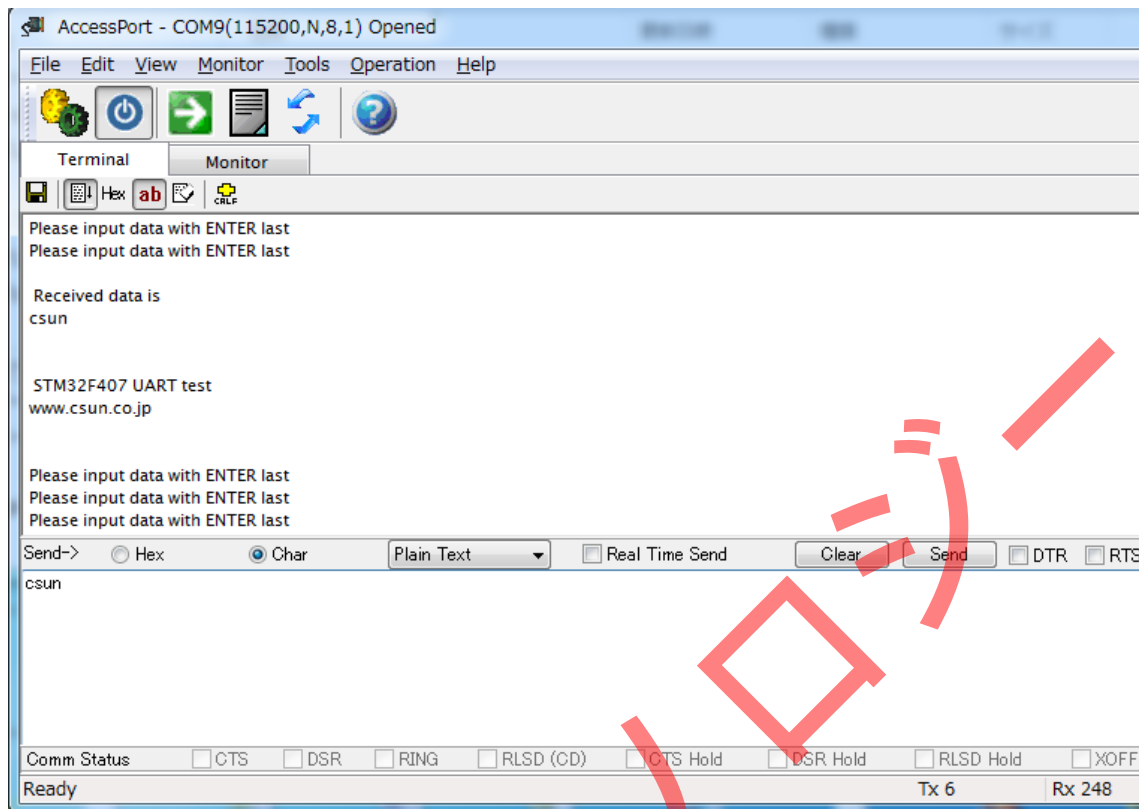
接続方法：STM32F407 開発ボードのシリアルポート1のRXとRS232C-TTL レベル変換基板のRXを繋ぐ、シリアルポート1のTXと変換基板のTXを繋ぐ。VCCとGNDは5V電源とGNDと

接続する。

ハードウェア接続イメージ：



シリアルポート出力イメージ：



(5) INT

このサンプルは外部割込みで STM32F407 開発ボードの 2 つのボタン (KEY0、KEY1) を通じて、ボードの 2 つの LED (DS0 と DS1) をコントロールする。

KEY1 ボタンは DS1 をコントロールし、一回押すと点灯、再度押すと消灯。

KEY0 ボタンは DS0 と DS1 を同時にコントロールし、一回押すと点灯、再度押すと消灯。

(6) Watchdog

このサンプルはウォッチドッグ(watchdog)をリセットしなければ、DS0 はずっと点灯する。WK_UP ボタンを押すと、フィードする。WK_UP ボタンを続けて押す場合、watchdog はずっとリセットせず、DS0 もずっと点灯する。一旦 watchdog 設置時間 (1 秒) を超えて WK_UP ボタンを押さなかった場合、プログラムは再起動になり、DS0 は一度消灯になる。

(7) WinWatchdog

このサンプルは DS0 を通じて、STM32 がリセットされたかを示す。もしリセットされたら DS0 を 300ms 点灯する。

DS1 は割込みウォッチドッグを示す。割込み発生する度 DS1 を一回回転する。

STM32 がリセットしなければ、DS0 はずっと消灯する。

(8) Timer

このサンプルは DS0 でプロジェクト実行を示し、周期は 400ms。DS1 はタイマー割込みの実行を示す。割込みで回転する。周期は 1000ms。実行後の現象は、DS0 は速く点滅し、DS1 は少しゆっくり点滅する。

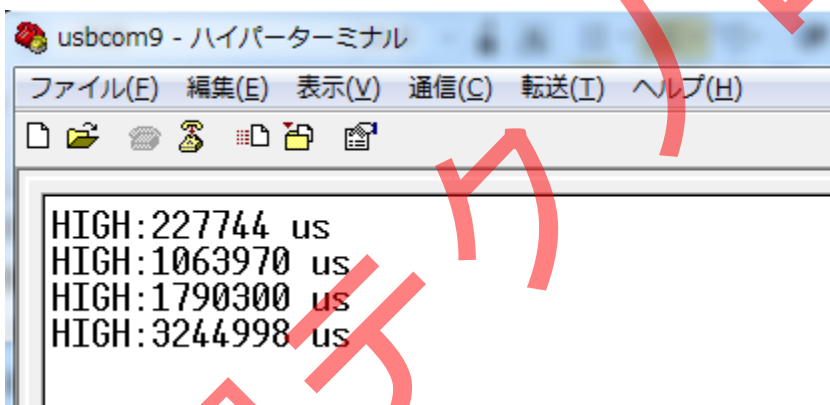
(9) PWM

このサンプルは TIM14_CH1 を使用して PWM を作成し、DS0 の輝度をコントロールする。実行後の現象は暗く→明るく→暗く→明るく→の循環である。

(10) Input

このサンプルは TIM15_CH1 を使用して PA0 のハイレベルを取得する。WK_UP を押してハイレベルを作成する。シリアルポートからハイレベルのパルス幅を出力する。
前節と同じ PWM の処理も残している。

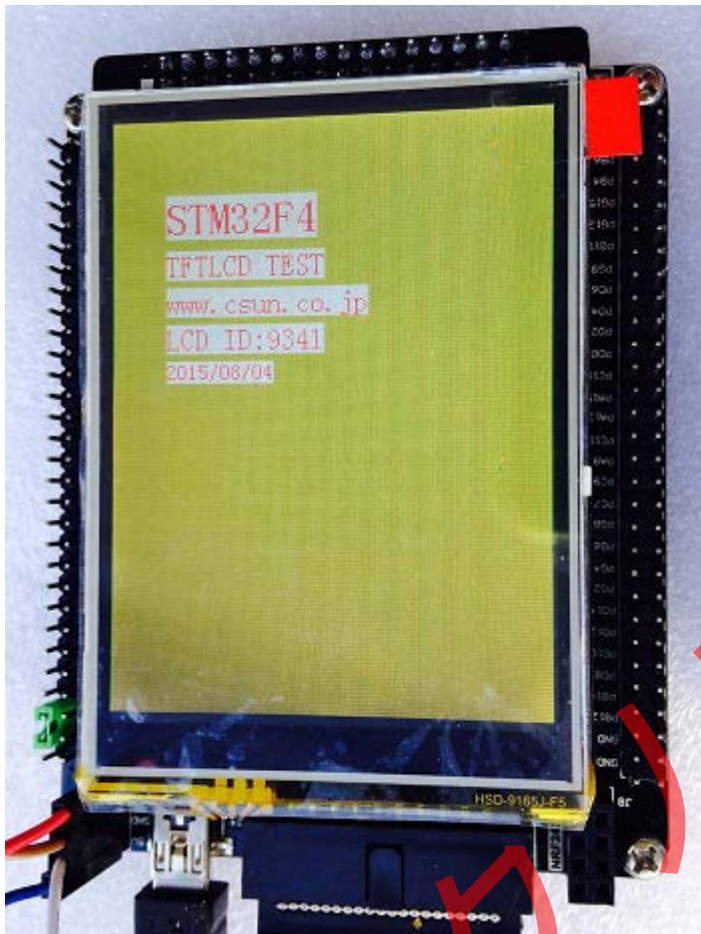
シリアルポートの出力例（WK_UP キーを押された時間が出力される）：



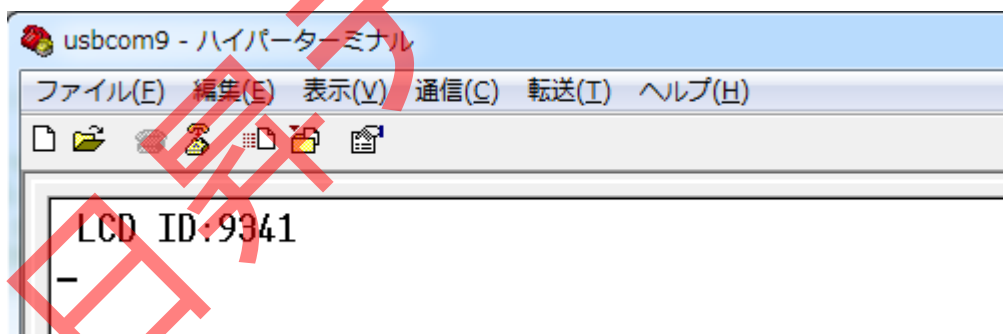
(11) TFT LCD

このサンプルは TFTLCD の表示を実現する。実行後、LCD にテキスト情報を表示し、背景色を自動で切り替える。またシリアルポートからリセットする度に LCD ドライバの ID を出力する。

実行イメージ：



シリアルポート出力：

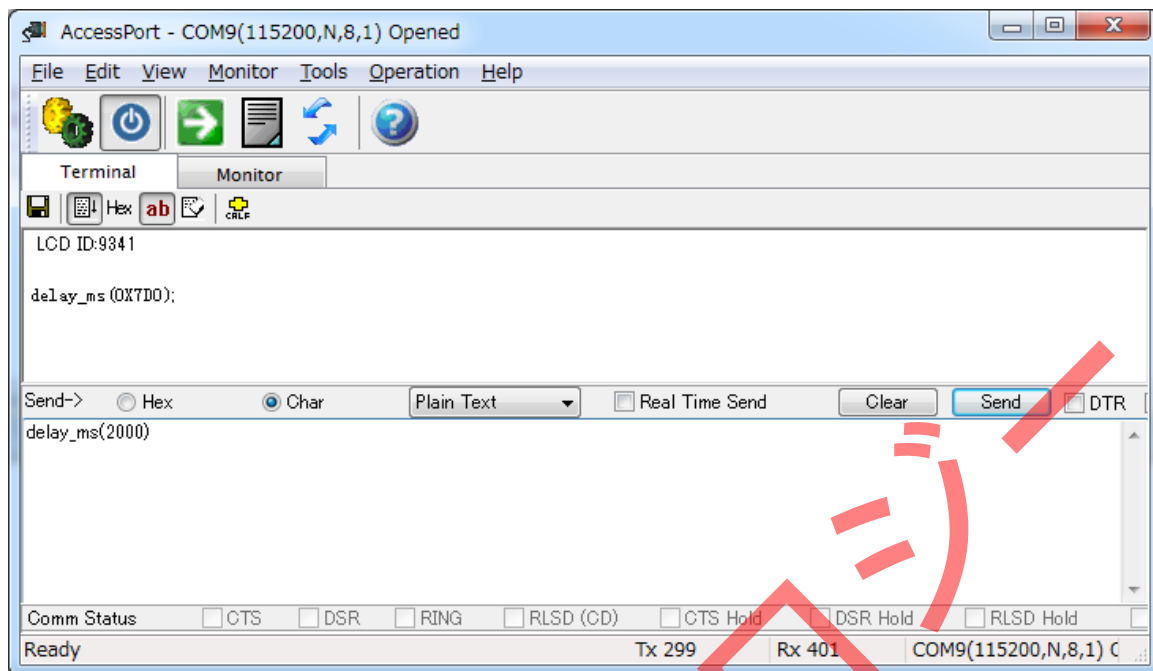


(12) USMART

このサンプルは usmart を使用して MCU 内蔵関数をコールして LCD と LED の表示及び遅延をコントロールする。

実行例：シリアルポートから delay_ms(2000)を送送すると、DS0 の状態は延長される。

シリアルポート出力状態：

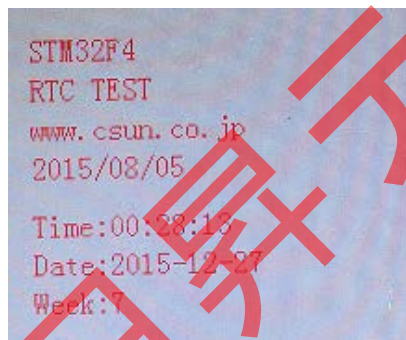


(13) RTC

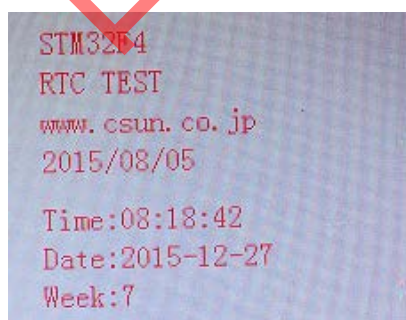
このサンプルは TFTLCD で RTC 時間を表示する。また usmart を使用して RTC 時間を設定できる。

実行例：

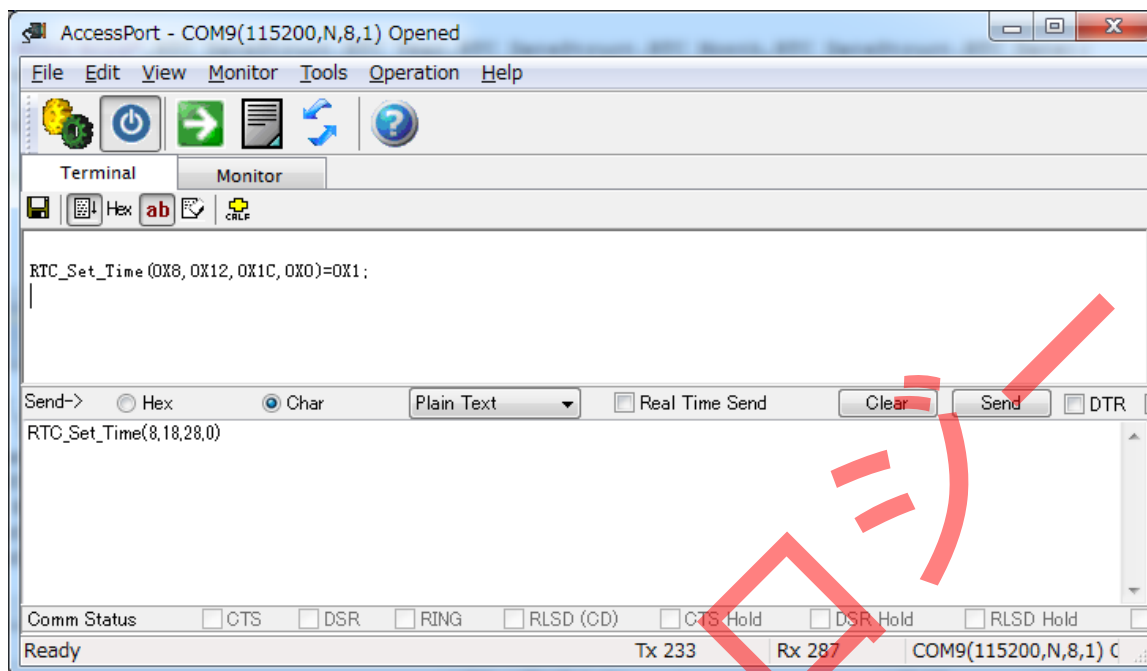
時間設定前の表示：



シリアルポートから時間設定後の表示：



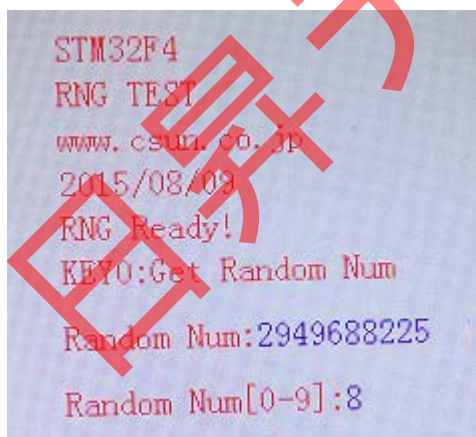
シリアルポート情報：



(14) RandomGen

このサンプルはSTM32F4 内蔵のハードウェアのランダム数生成器（RNG）でランダム数を生
成して、LCD に表示する。KEY0 キーを押してランダム数を取得する。また 0～9 の範囲内で
ランダム数を取得して、LCD に表示する。D0 はプログラム実行状態を示す。

実行イメージ：

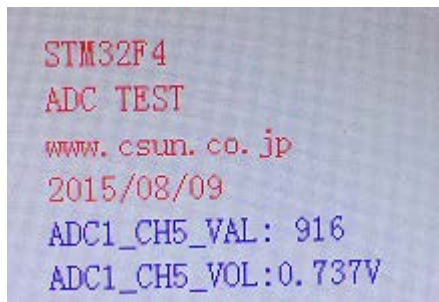


(15) ADC

このサンプルはADC1 でチャンネル 5（PA5）の電圧を取得して、LCD に ADC 変換値及び変換後
の電圧値を表示する。

注意：本テストの参考電圧は 3.3V。他の参考電圧を使用する場合、STM32F4 ボードの P7 ポートで設定できる。他の参考電圧を設定した後、入力電圧は参考電圧の最大値を超えない様に注意する必要。

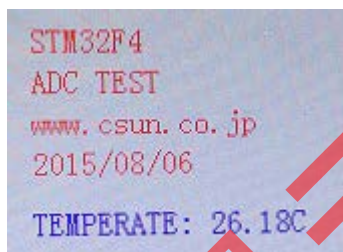
実行イメージ：



(16) Temperature

このサンプルは ADC1 のチャンネル 16 で STM32F4 内部温度センサーの電圧値を取得して温度に変換し、LCD に表示する。

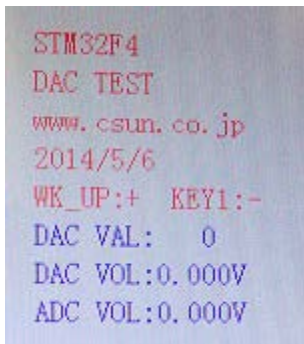
実行イメージ：



(17) DAC

このサンプルはキー或いは USMART で STM32F4 内蔵 DAC のチャンネル 1 の出力電圧をコントロールする。ADC1 のチャンネル 5 で DAC の出力電圧を採取して LCD に表示する。また Usmart で Dac1_Set_Vol 関数をコールして DAC の出力電圧を設定できる。

注意：ボードの PA4 と PA5 ピンをショートする必要。



(18) PWM DAC

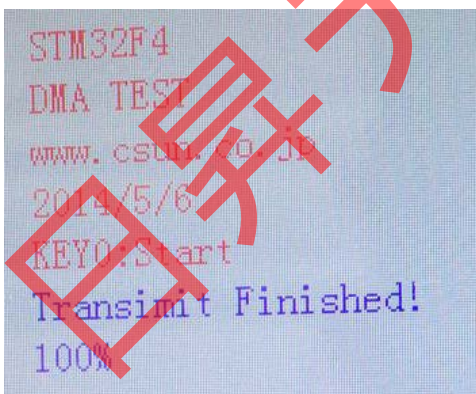
このサンプルはキー或いはUSMARTでSTM32F4のTIM9_CH2のPWM出力をコントロールする。RC フィルターした後 DAC 出力に変換して ADC1 のチャンネル 5 で PWM DAC の出力電圧を採取して LCD に表示する。

注意：ボードの PA3 と PA5 ピンをショートする必要。

(19) DMA

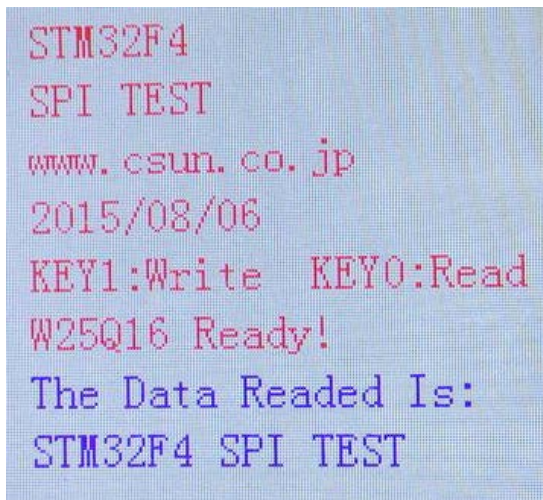
このサンプルはKEY0 キーで DMA シリアルポート 1 のデータ送信をコントロールする。KEY0 を押したら、DMA 転送が始まる、同時に LCD 上に転送進捗を表示する。シリアルデバッグツールで DMA 転送の内容を受信できる。

注意 1：シリアルのボーレートは 115200。



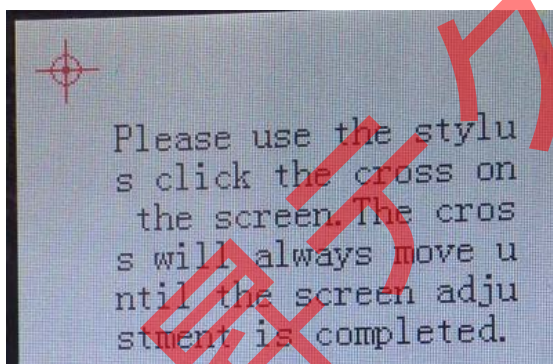
(20) SPI

このサンプルはKEY1 キーで W25Q16 への書き込をコントロールする。KEY0 キーで W25Q16 からの読出しをコントロールする。同時に LCD 上に情報を表示する。



(21) TouchPanel

このサンプルはまず LCD ID によって静電気タッチパネルか、抵抗式タッチパネルかを確認して関連の検査を行う。デフォルトは抵抗式タッチパネルです。校正したかを確認して、してなければ校正を行う。校正した場合は手書きプログラムに入る。スクリーン上にクリアエリア (RST) があり、ここをクリックすると全てクリアされる。また KEY0 で校正を実行する。

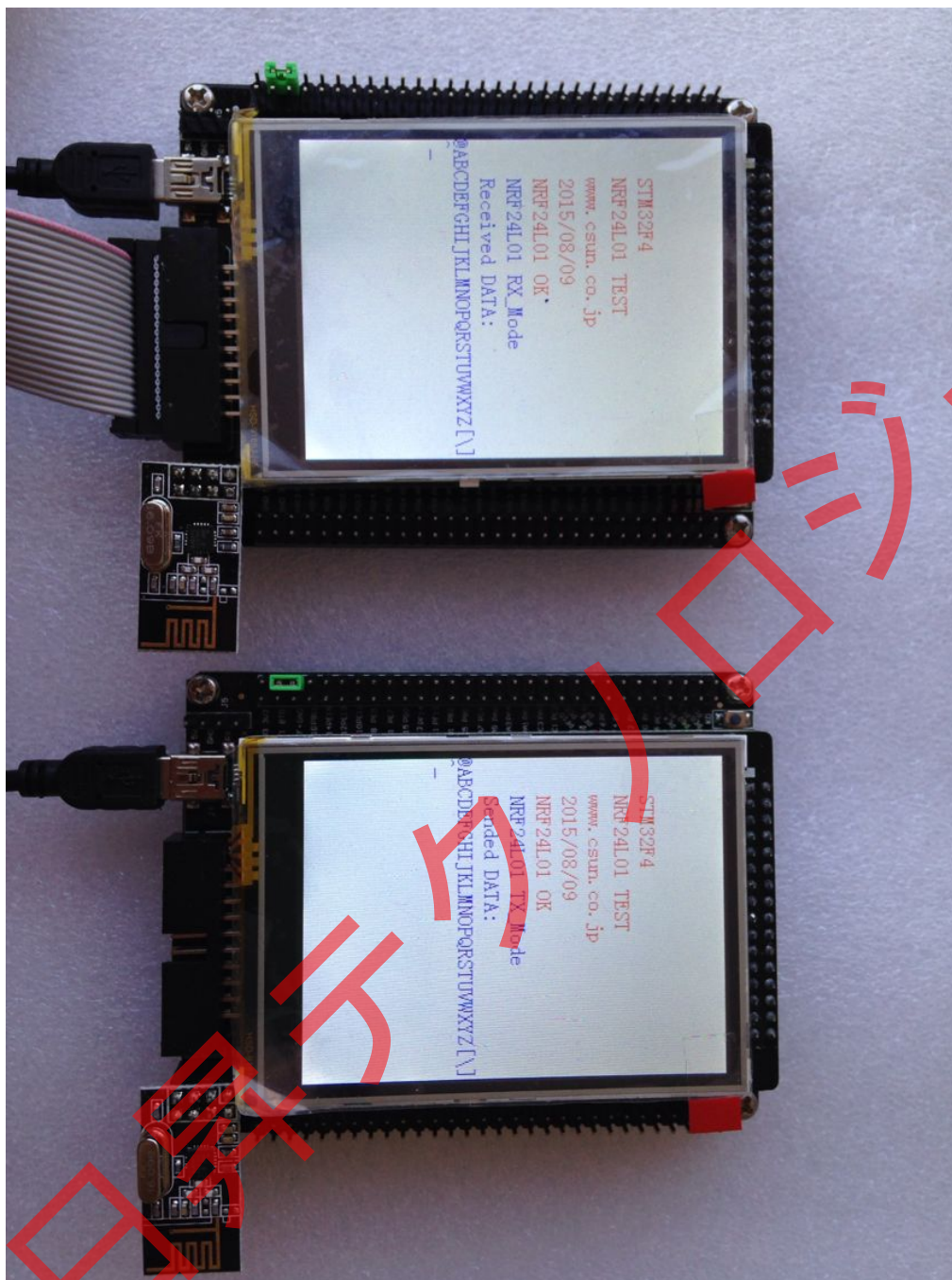


(22) NRF24L01

このサンプルは起動する時はまず、NRF24L01 モジュールが存在するか確認する。NRF24L01 モジュールを検出した後、KEY0 と KEY1 の設置によってモジュールの動作モードを確認し、動作モードを正確に設定した後、継続的にデータを送信/受信することができ、同時に DS0 で実行していることを示す。

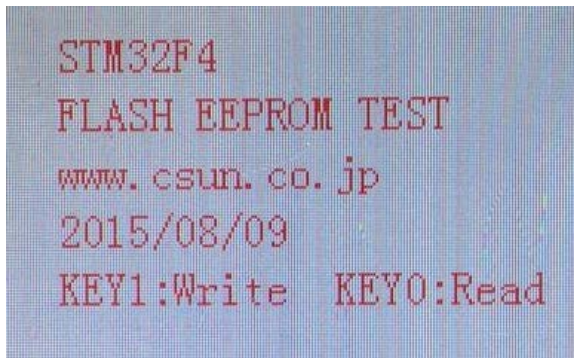
注意：

本テストは2セットの開発ボード+2つの NRF24L01 無線モジュールで、正常にテストすることができる。1つの開発ボードと1つのモジュールではテストすることができない。



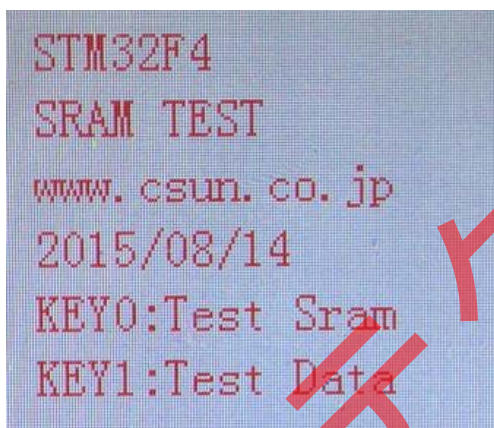
(23) FLASH2EEPROM

このサンプルは起動する時はまず、提示の情報を画面に表示し、それからメインループの中で2つのボタンを測定し、1つのボタン (KEY1) はFLASHの書き込を実行する。もう1つのボタン (KEY0) は読み出しを実行する。TFTLCD上で関連情報を表示する。DS0で実行していることを示す。



(24) SRAM

このサンプルは起動した後に、提示の情報を画面に表示して、KEY0 キーを押したら、外部 SRAM 容量のサイズを測定し、LCD 上に表示する。KEY1 キーを押したら、予め保存した外部 SRAM のデータを表示する。DS0 で実行していることを示す。



(25) Memory

このサンプルは起動した後に、提示の情報を画面に表示し、外部入力を待つ。KEY0 はメモリを申し込む。毎回 2K バイトのメモリを申請する。KEY1 の機能は申請したのメモリの中にデータを書く。KEY_UP は操作メモリエリア（内部 SRAM メモリ/外部 SRAM メモリ/内部 CCM メモリ）を切り替えることを実現する。同時に DS0 で実行していることを示す。

```
STM32F4
MALLOC TEST
www.csun.co.jp
2015/08/15
KEY0:Malloc
KEY_UP:SRAMx KEY1:Write
SRAMIN
SRAMIN USED: 0%
SRAMEX USED: 0%
SRAMCCM USED: 0%
```

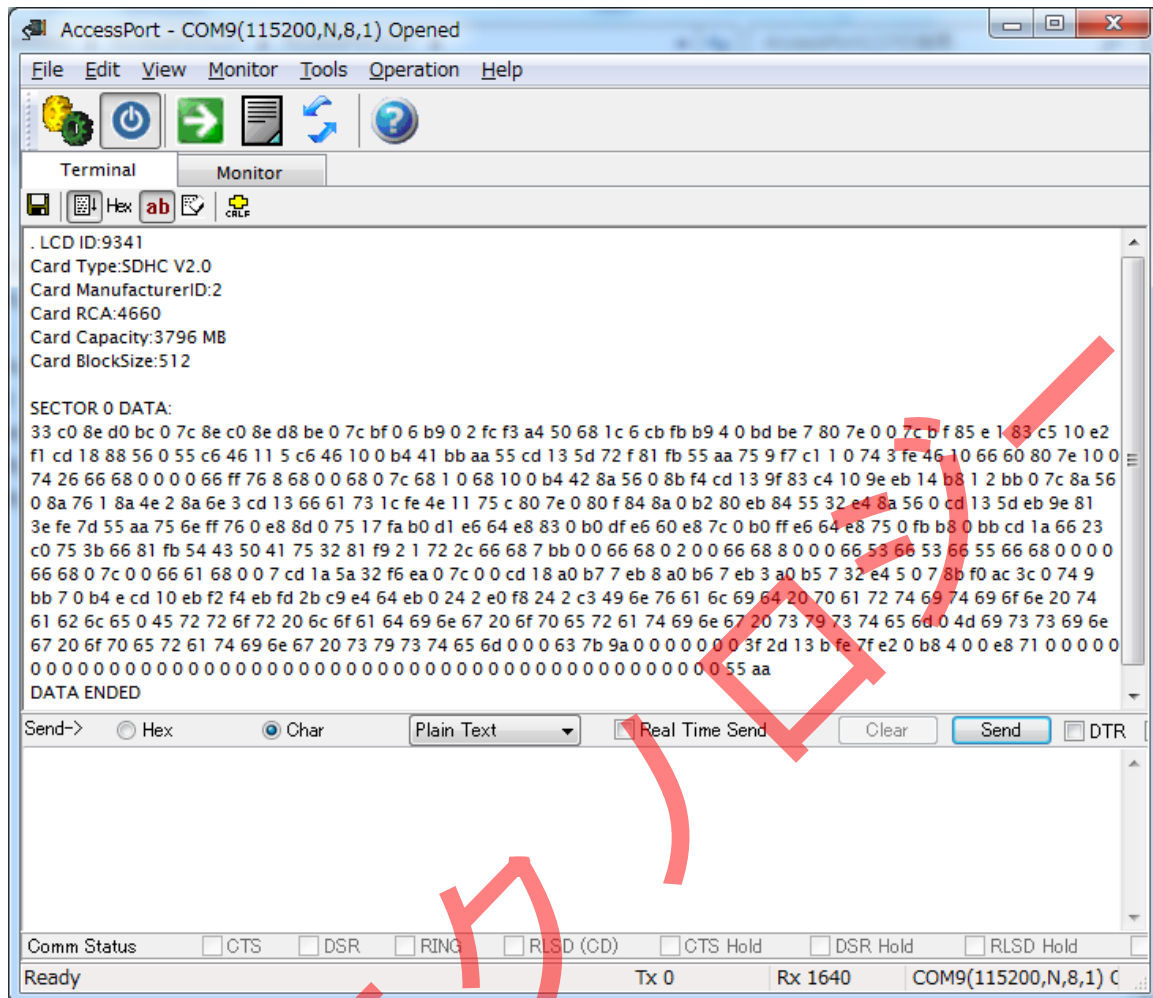
(26) SD

このサンプルは起動する時は先に SD カードを初期化する。成功すれば、LCD が初期化することを提示し、KEY0 を押し、SD カードのセクター0 のデータを読み取って、シリアルポートで PC に発送する。もし初期化できないと、LCD の上で失敗した情報を表示する。DS0 で実行していることを示す。

LCD 表示内容：

```
STM32F4
SD CARD TEST
www.csun.co.jp
2015/08/15
KEY0:Read Sector 0
SD Card OK
SD Card Size: 3796MB
```

シリアルポート出力イメージ (KEY0 を押した場合)：



(27) FATFS

このサンプルは起動する時は先に SD カードを初期化する。成功すれば、2つのワークエリアを登録し（一つは SD カード用、一つは SPIFLASH 用のため）、SD カードの容量と余裕空間を測定し、LCD 上に表示し、最後 USMART からのコマンドでテストを行う。DS0 で実行していることを示す。

注意：

- 1、一つの SD カードを用意してください。
- 2、USMART を通じ、各種の fatfs をコールしてテストする。

```
STM32F4
FATFS TEST
www.csun.co.jp
2015/08/15
Use USMART for test
FATFS OK!
SD Total Size: 3637MB
SD Free Size: 3088MB
```

(29) IMAGE

このサンプルは起動する時 SD カードの存在するかどうかを確認する。存在する場合、SD カードのルート・ディレクトリの下に PICTURE フォルダを探す。見つけたらこのフォルダの下に画像のファイル (bmp、jpg、jpeg あるいは gif をサポートする) をループで表示し、KEY0 と KEY1 で、PICTURE を閲覧できる。WK_UP キーは一時停止/再生の機能で、DS1 は当面の状態を一時停止かどうかを指示する。もし PICTURE フォルダ/画像のファイルが見つからないと、エラーの提示を表示する。本テストは DS0 で実行していることを示す。

注意：

- 1、本テストは一つの SD カードをご用意ください。そして SD カードでルート・ディレクトリで PICTURE フォルダ作り、いくつかの画像 (BMP/JPG/JPEG/GIF) を入れてください。
- 2、もし一部 jpg/jpeg を読み取れなかったら、Windows XP のペイントツールで開いて保存してください。
- 3、JPEG/JPG/BMP は LCD の解像度によって自動的にズームすることができる。GIF は LCD の解像度の以下でないと読み取れない。



(30) AVPlay

このサンプルソースは次の機能を実現する：立ち上げた後、まずは周辺装置を初期化する。若し問題がなければ、TF カードにある VIDEO ファイルの中のビデオ（avi フォーマット）を再生しはじめる。

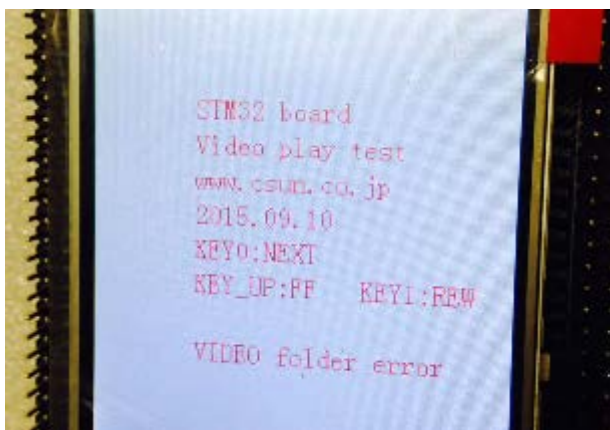
ビデオを再生する時、TFT LCD 上に当ビデオの名前、番号、ビデオの総数、サウンドトラック数、オーディオサンプリングレート、フレームレート、再生時間と総時間などの情報も示す。KEY0 キーを押して次のビデオを再生する。WK_UP キーは早送り、KEY1 キーは早戻しができる。

注意：

- 1、TF カード一枚を用意する必要。
- 2、TF カードのルート目録に VIDEO フォルダを作成し、AVI ビデオファイル（ビデオは MJPG しかサポートしない、オーディオは PCM でなければならない。そして、ビデオの解像度は LCD の解像度より小さい又は同じでなければならない）を入れる。
- 3、本ボードでは、オーディオ・デコードが搭載してないので、音声は確認できない。

実行する際のイメージ

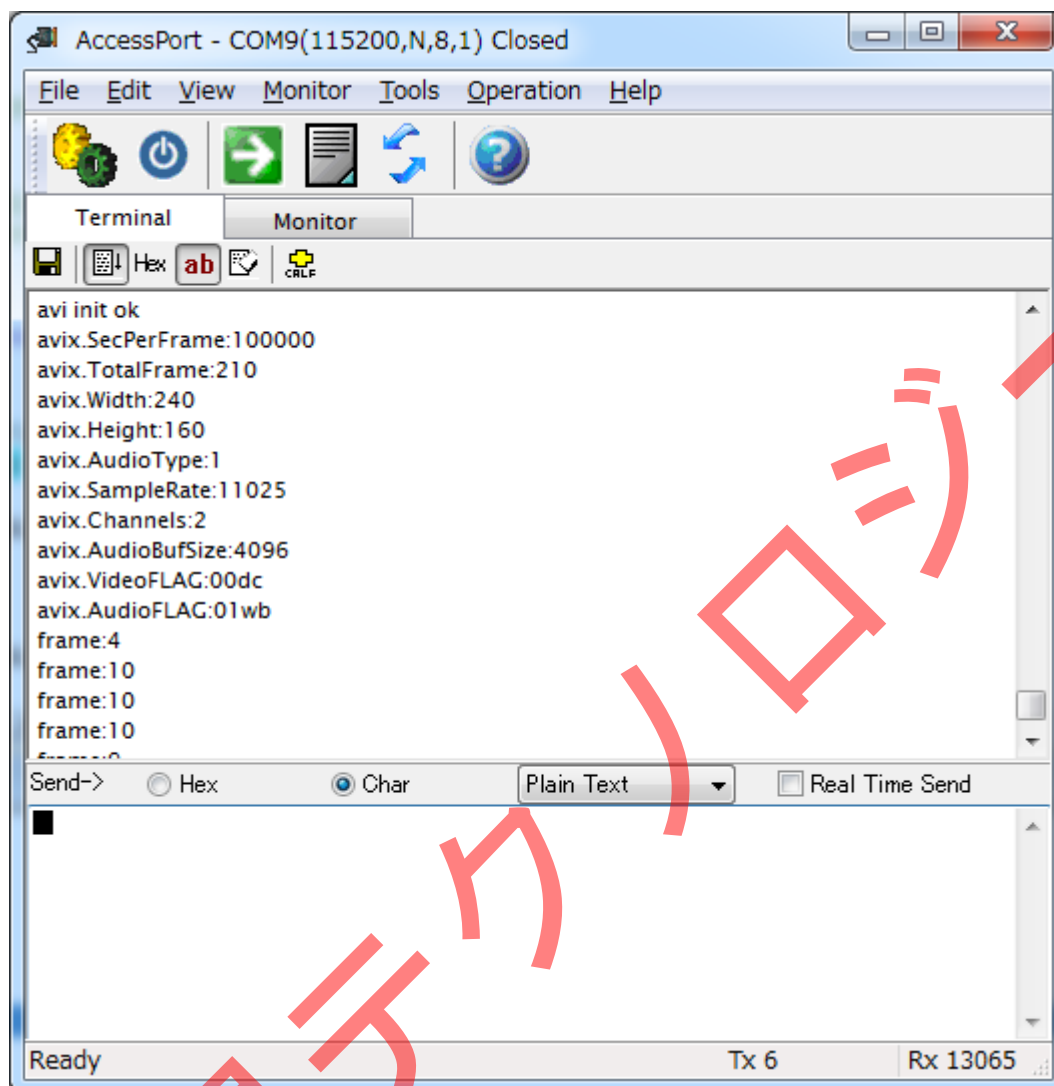
- 1、TF カード挿入してない場合：



2、avi ファイル保存してある TF カード挿入した場合：



シリアルポートから出力した情報：



(31) FPU(Julia)

このサンプルソースは次の機能を実現する：立ち上げた後、反復回数よりカラーテーブル（RGB565）を作成し、そして、ジュリア分形を計算して LCD 上に表示する。また、観察比較をし易いように、1フレームがかかる時間を統計する為に、プログラムはタイマー3を起動し、1フレームのジュリア分形図面を表示し終わった後、プログラムの実行時間、FPU が使用されているかどうかとズーム倍率などの情報を示す。KEY0/KEY1 でズーム倍率を調節することができる。WK_UP で自動ズームと手動ズームを設定する。DS0 はプログラムの実行状態を示す。

注意：

1、「46_1_FPU(Julia)_openHardwareFPU」と「46_2_FPU(Julia)_closeHardwareFPU」のソースコードは全く同じ、ただハードウェア FPU をオン・オフしただけ。

2、テストする時、まずは一つをダウンロードして、対応する時間とパラメーターを記録する。そしてもう一つをダウンロードして、同じパラメーター、特に時間を見て、比較できる。

46_1_FPU(Julia)_openHardwareFPU を実行時：
を実行時：

46_2_FPU(Julia)_closeHardwareFPU



(32) DSP

47_1_DSP-BasicMath について

STM32F4 の DSP ライブラリの基本数学関数：arm_cos_f32 と arm_sin_f32、と標準ライブラリの基本数学関数：cosf と sinf のスピード差を確認して、二つの計算にかかる時間を LCD に表示する。DS0 はプログラムが実行状態である事を指示する。

実行イメージ：



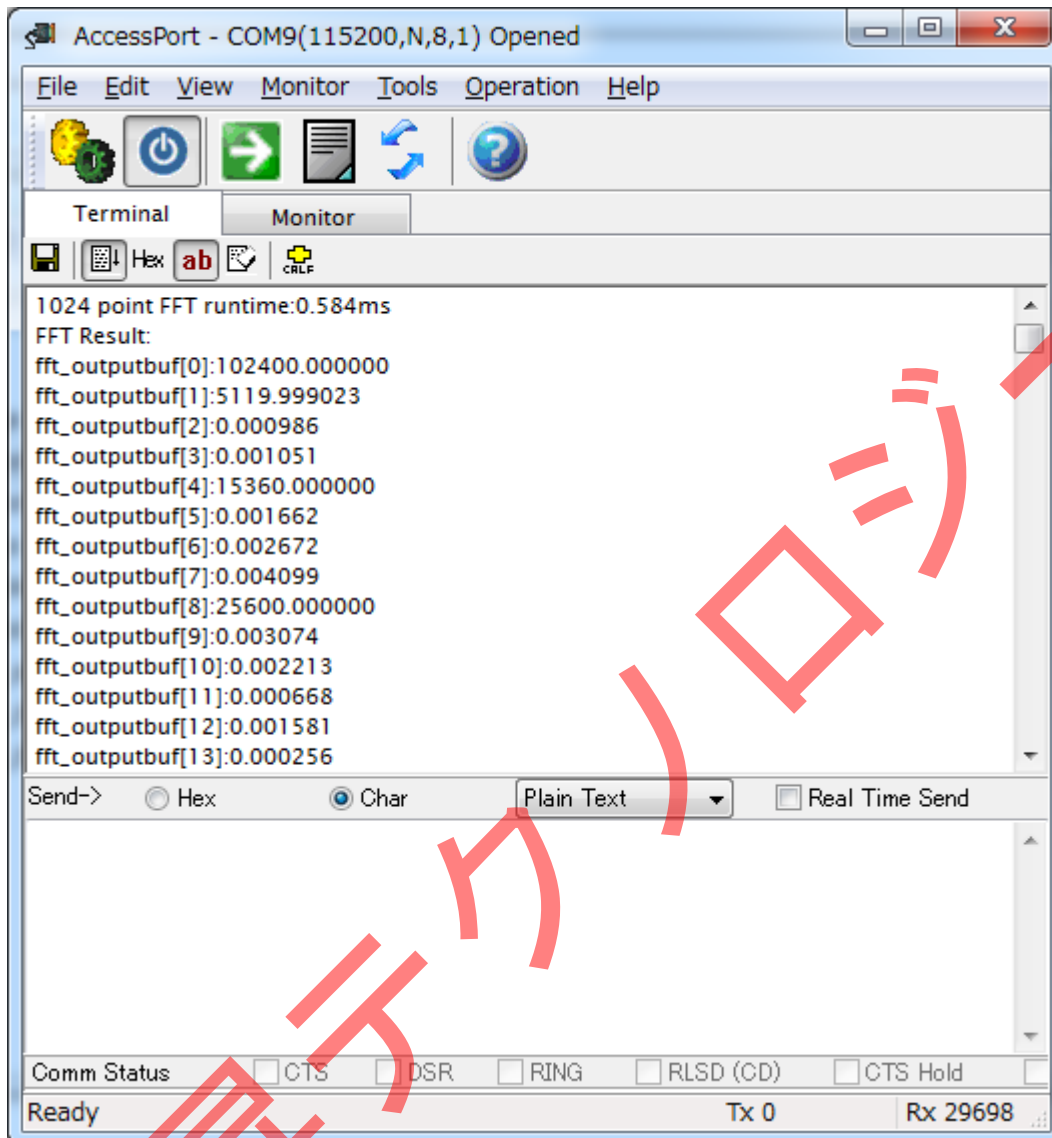
47_2_DSP-FFT について

STM32F4 の DSP ライブラリの FFT 関数をテストする。プログラム実行後、1024 点検査序列が自動的に作成する。そして、KEY0 を押すたびに、DSP ライブラリの FFT 計算法（基 4 法）を呼び出して FFT 計算を実行する。LCD に計算時間を表示して、同時に FFT の結果をシリアルポートからも出力する。DS0 はプログラムが実行状態である事を指示する。

実行イメージ：



シリアルポート出力：



(33) HandWriter

このサンプルソースは次の機能を実現する：立ち上げて、タッチパネルの校正モードに入る。画面提示に従って、十字マークを4回クリックする。10秒間入力しない場合は自動で終わる。そして、入力待つ状態になる。LCDの手書きエリア内に数字あるいはキャラクタを入力する。入力し終わったたびに、自動的に識別状態に入って識別する。また識別の結果をLCDに表示する（同時にシリアルポートにも出力する）。KEY0を押すと、モードを切り替えることができる（4種類のモードがある）、KEY1を押すと、タッチパネルの校正モードに入る。DS0はプログラムが実行している状態を指示する。

注意：静電容量式タッチスクリーンを調整する必要がないので、静電容量式スクリーンを使用する時、KEY1を押しても反応が出ない。

実行イメージ：

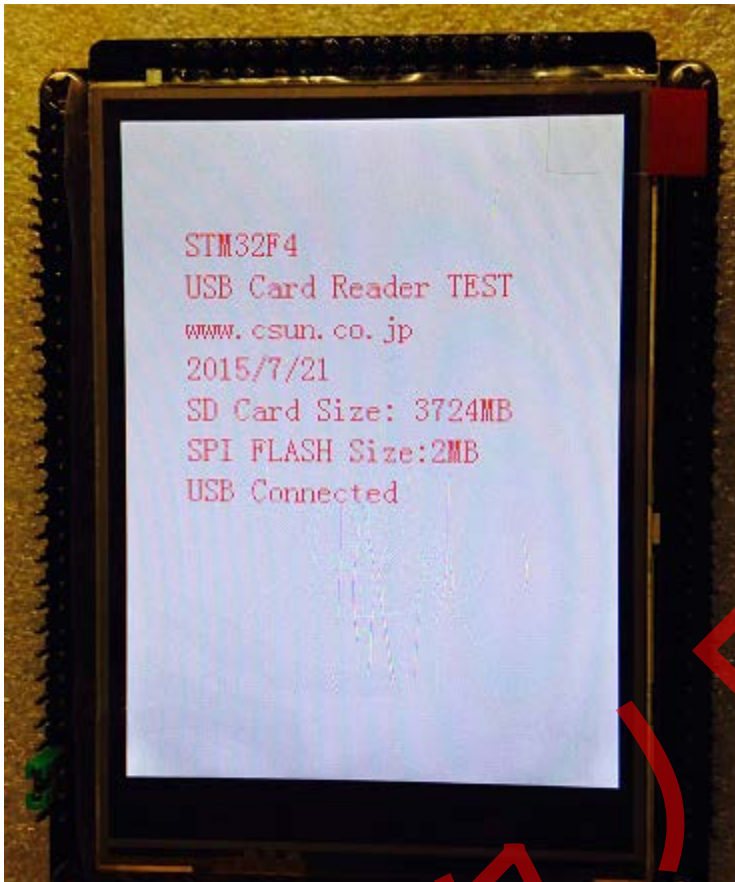


(35) USBCardReader(Slave)

このサンプルソースは次の機能を実現する：立ち上げる時、SD カードと SPI FLASH が存在するかどうかを検査する、もし存在する場合、その容量を LCD に表示する。(存在しない場合、エラーを知らせる)。その後、USB を設定し始める。設定が成功したあと、PC 上に 2 つのリムーバブルディスクが見つける。DS1 で USB がアクセスしていることを示して、LCD にも表示する。DS0 はプログラムが実行している状態を示す。

注意：

- 1、SD カード 1 枚を準備してください。(もしないなら、ボードに搭載した SPI FLASH ディスクしか使えない)。
- 2、ボードと PC を USB ケーブルで繋ぐ必要。



(37) UCOSII1-1-EventCall

このサンプルは下記機能を実現する。UCOSII で3つのタスクを作成する：スタートタスク、LED0 タスクと LED1 タスク。スタートタスクはその他のタスク（LED0 タスクと LED1 タスク）を作成後ハングする。LED0 タスクは DS0 LED をコントロールする、1 秒間 80ms 点灯する。LED1 タスクは DS1 LED をコントロールする、300ms 点灯して 300ms 消灯する。本テストで使用した ucossii バージョンは V2.91 である。

(38) UCOSII-1-2-EventOther

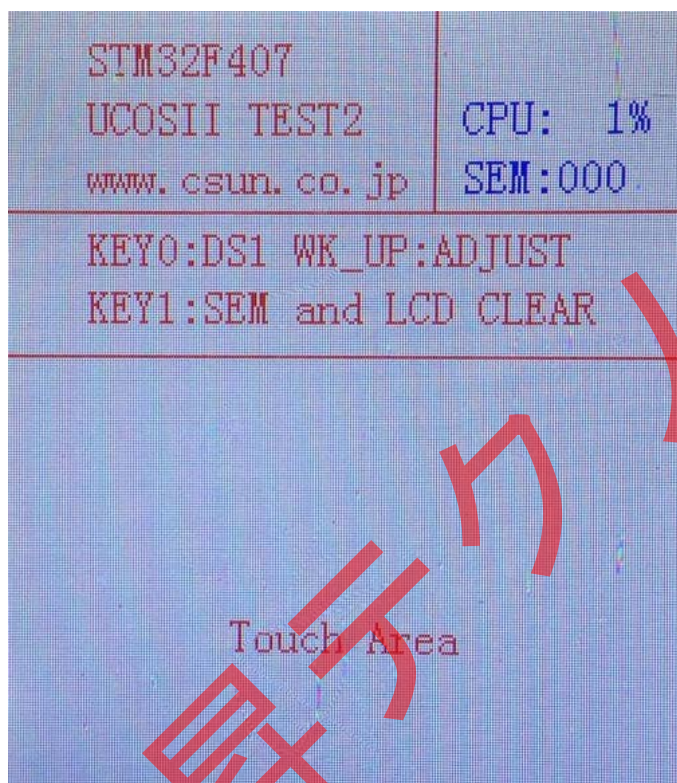
このサンプルは下記機能を実現する。UCOSII で3つのタスクを作成する：スタートタスク、LED タスクと KEY タスク。スタートタスクはその他のタスク（LED タスクと KEY タスク）を作成後ハングする。LED タスクは DS0/DS1 LED をコントロールする。KEY0 を押して LED タスクをハングして消灯する。KEY1 を押して LED タスクを再起動して点灯する。本テストで使用した ucossii バージョンは V2.91 である。

(39) UCOSII-2-SemaphoreMailbox

このサンプルは下記機能を実現する。UCOSII で6つのタスクを作成する：スタートタスク、

LED タスク、タッチパネルタスク、BEEP タスク、メインタスクと KEY タスク。スタートタスクはその他のタスクを作成後ハングする。LED タスクは DS0 LED をコントロールする。BEEP タスクはセマフォを申請する。タッチパネルタスクは Handwritting と CPU の使用量をテストする。KEY タスクはキーをスキャンする、優先順位が一番高い、キーをスキャン後、Mailbox で発送する。メインタスクは Mailbox でキーを検索して各種のタスクをコントロールする。

KEY0 で DS1 の点滅をコントロールする。KEY1 でセマフォを申請する、LCD でカレント値を表示する、同時に Handwritting エリアの表示をクリアする。WK_UP でタッチパネルの校正を行う。



本テストで使用したucosii バージョンは V2.91 である。

以上