

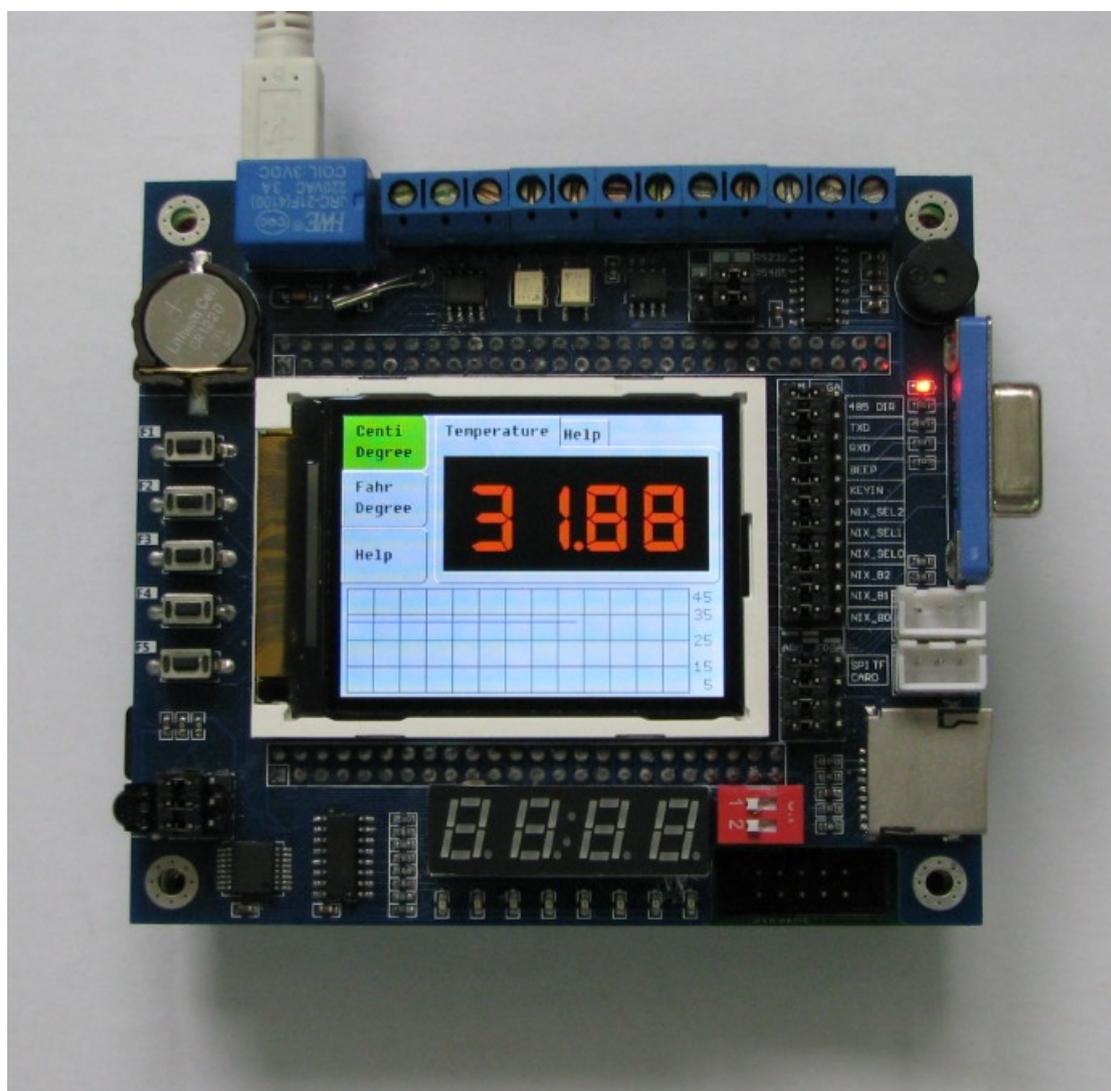
## STM32+CycloneIV 統合ボードマニュアル

株式会社日昇テクノロジー

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

info@csun.co.jp

作成日：2012/09/01



copyright@2012-2013



## ・ 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2012/09/01

※ 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。



1. 統合ボード紹介.....	4
2. 統合コアボードインタフェース.....	4
2.1 インタフェース説明.....	4
2.2 コアボードボタン定義.....	6
3. 拡張ボードインタフェース.....	7
3.1 インタフェース説明.....	7
3.2 詳細説明.....	8
4. システムブロック図.....	12
5. シミュレーション USB メモリモードに入る.....	12
6. LCD スクリーンキャプチャー.....	13
7. MiniLink と接続&デバッグ.....	14
8. USB シリアルポート変換モジュールで接続と通信.....	19
9. Altera ダウンロードで FPGA をダウンロードする.....	20
10. VGA インタフェースでモニタと接続する.....	21
11. iTool で色んな開発を行う.....	22
12. 拡張板と一緒に動く様子.....	23
13. 豊富なサンプル.....	24

## 1. 統合ボード紹介

本統合ボードは ARM と FPGA を搭載するものです、ARM の方、周波数が ST 社の Cortex-M3 72MHZ STM32F103VC ARM チップを搭載しています、豊富なインタフェースも含まれます。FPGA の方、Cyclone™ IV FPGA EP4CE6E22C8N を搭載しています。バスを通して ARM と FPGA を統合されています、ARM は主 CPU とし、FPGA がコプロセッサで使われて並列処理できます、そうすると、ARM と FPGA の利点をお互いに使って連携させ、組込システムの二つ種類チップの特徴を合わせて利用できます。統合ボードはコアボードと拡張ボードを分けられます。

## 2. 統合コアボードインタフェース

### 2.1 インタフェース説明

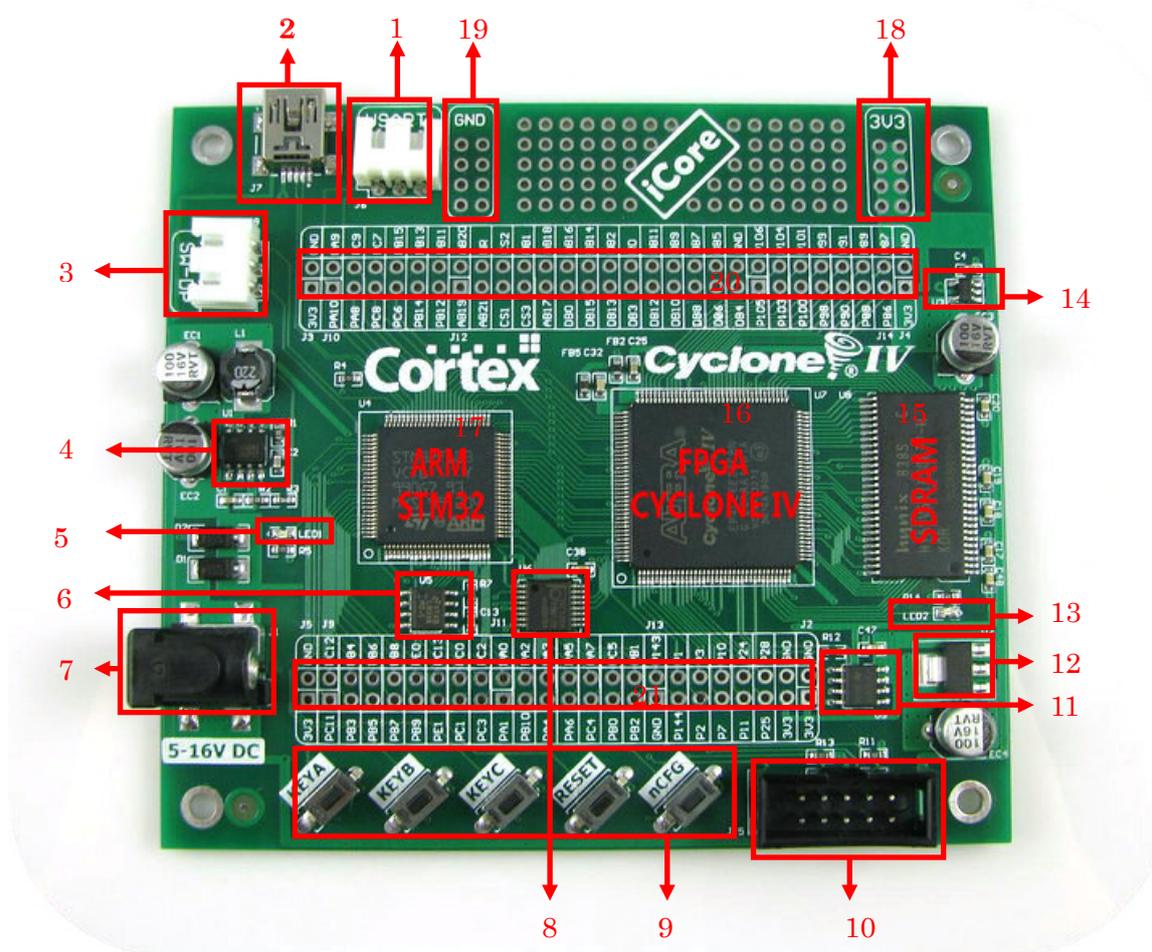


図 1 コアボード正面

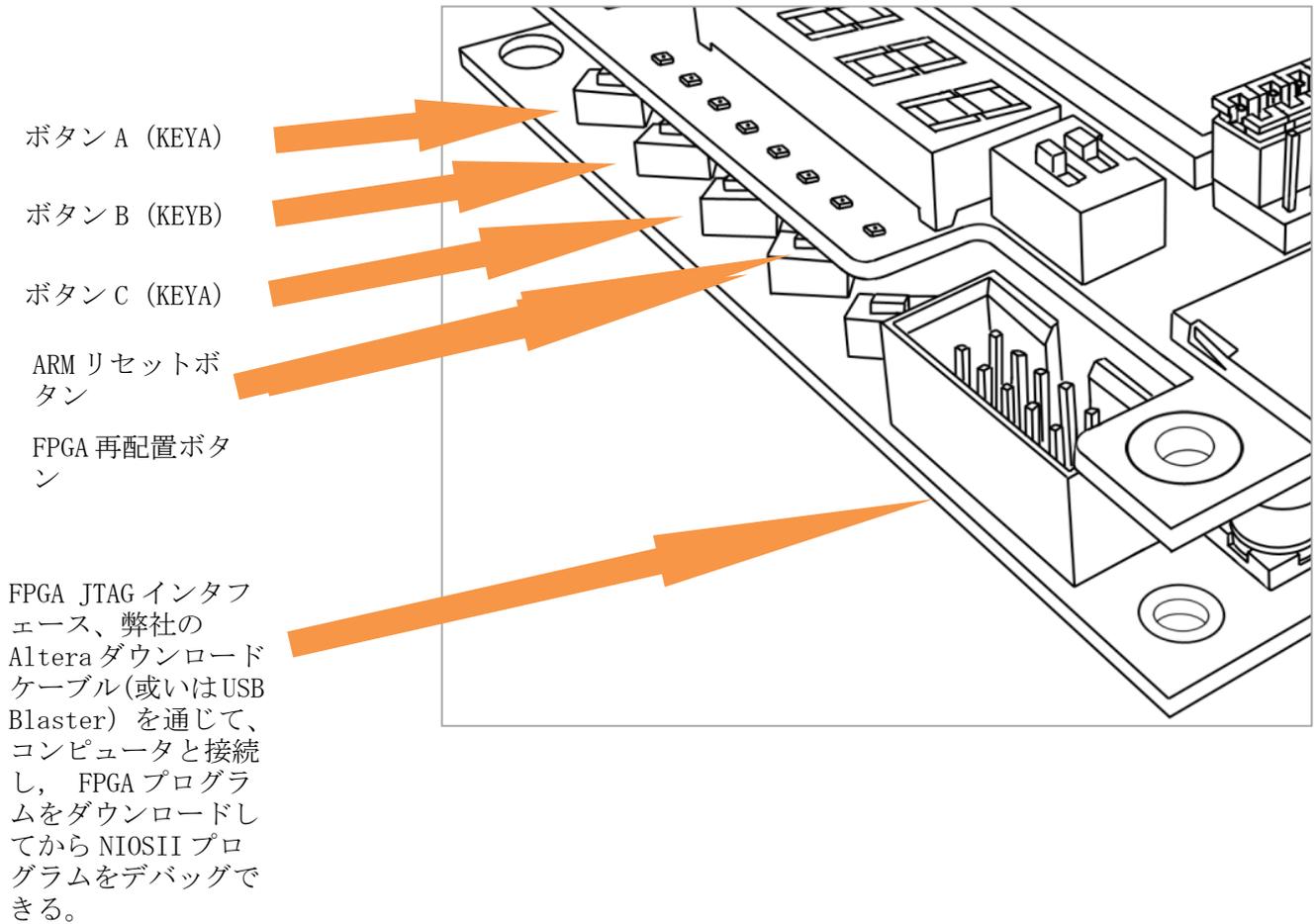


表1 コアボードインタフェース

NO	インタフェース	機能説明
1	UART インタフェース	XH2.54 コネクタ、USB から UART への変換モジュールを通じてコンピュータと通信する
2	USB インタフェース	MiniUSB インタフェース、二つ機能があり： 1. ボードに給電する； 2. コンピュータと USB で通信する；
3	ARM SWD デバッグインタフェース	XH2.54 コネクタ、プログラムをダウンロードまたデバッグする
4	パワーチップ	AOZ1016、BUCK DCDC コンバータ、3.3V を出力し、開発ボードに給電する
5	LED1	LED1:ARM 機能指示
6	2.5V 基準電圧源	LM336M-2.5 高精度基準電圧源、ADC/DAC で使える
7	オプション給電プラグ	2.1mm SMD DC 端子、DC 5~16V 幅広い範囲の電圧入力可能
8	デコーダ	74LVC138 アドレスデコーダ、FSMC チップ・セレクトを拡張する際使われる (CS0~CS3)
9	ボタン×5	3つ独立ボタン、ARM リセットボタン、FPGA 再配置ボタン
10	FPGA JTAG インタフェース	FPGA ダウンロード用、NIOSII デバッグ標準機能
11	EPCS 配置チップ	16M bit FPGA コンフィギュレーション情報、NIOS II コードを保存可能
12	2.5V LDO	AMS1117、2.5V 出力、FPGA を給電
13	LED2	LED2:FPGA 機能指示
14	1.2V LDO	RT9193、1.2V 出力、FPGA 位相ロックループを給電
15	SDRAM	HY57V641620HT、8M バイト SDRAM、SOPC 実験を行える
16	FPGA	Cyclone™ 四世代 FPGA EP4CE6E22C8N
17	ARM	32 ビット Cortex-M3 STM32F103VC、2 チャンネル 12bit DAC、16 チャンネル 12bit ADC を内蔵
18	3.3V 電源	複数電源
19	グランド拡張	グランド拡張ポート
20	ARM ピン拡張	複数 ARM IO 拡張、STM32 チップから SPI/CAN/UART/I2C をすべて引き出され、詳しくは回路図を参照する
21	FPGA ピン拡張	複数 FPGA IO 拡張、詳しくは回路図を参照する

■PCB 寸法図：90×100mm

## 2.2 コアボードボタン定義



### 3. 拡張ボードインタフェース

#### 3.1 インタフェース説明

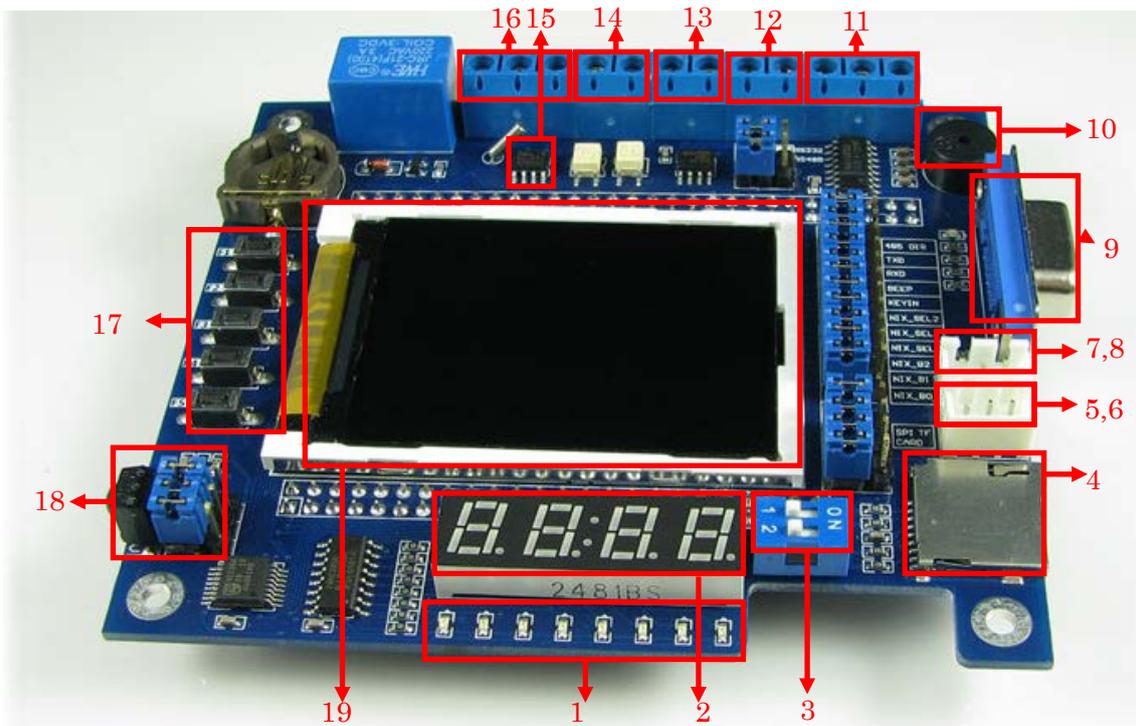


図 2 : 拡張ボード正面

表2 拡張ボードインタフェース

NO	インタフェース	STM32 接続可否	FPGA 接続可否
1	LED×8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	デジタルチューブ×4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	2チャンネルDIPスイッチ	<input checked="" type="checkbox"/>	
4	Micro SD カード	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	12ビットADC入力(裏のピン)	<input checked="" type="checkbox"/>	
6	12ビットDAC出力(外のピン)	<input checked="" type="checkbox"/>	
7	PWM出力(裏のピン)	<input checked="" type="checkbox"/>	
8	周波数測定入力(外のピン)		<input checked="" type="checkbox"/>
9	VGA インタフェース		<input checked="" type="checkbox"/>
10	ブザー	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	RS-232 インタフェース	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



12	RS-485 インタフェース	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	フォトカプラ出力	<input checked="" type="checkbox"/>	
14	フォトカプラ入力	<input checked="" type="checkbox"/>	
15	RTC リアルタイムクロック	<input checked="" type="checkbox"/>	
16	リレー (50m 抵抗、AC250V/3A、 DC30V/3A)	<input checked="" type="checkbox"/>	
17	ボタン×5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	赤外線受信器	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	2.2" 320x240 TFT	<input checked="" type="checkbox"/>	
20	2M SPI Flash(裏面)	<input checked="" type="checkbox"/>	
21	LM75A 温度センサー (裏面)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	24LC04 EEPROM (裏面)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

### 3.2 詳細説明

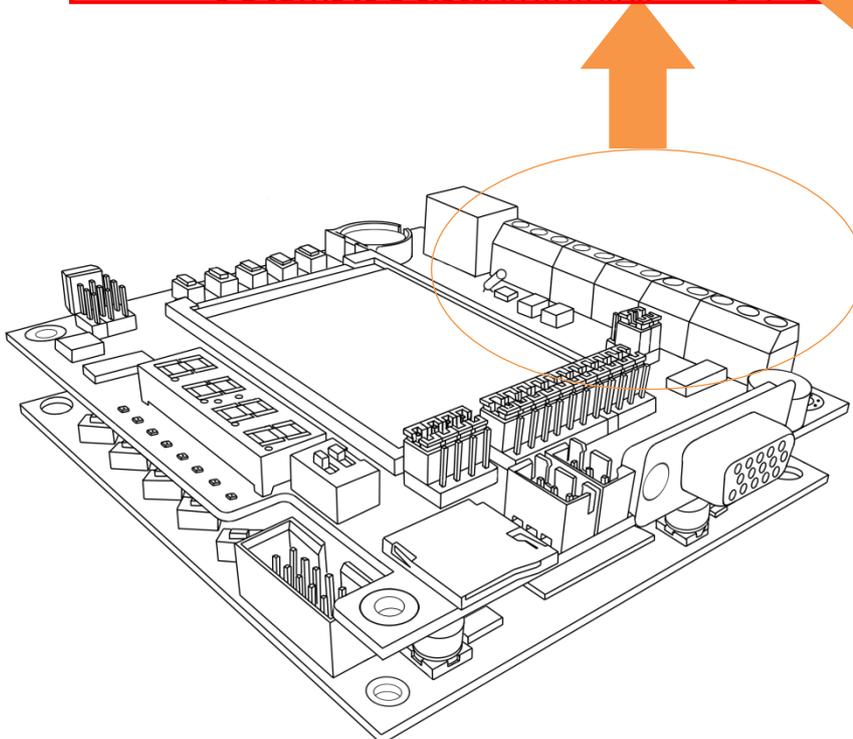
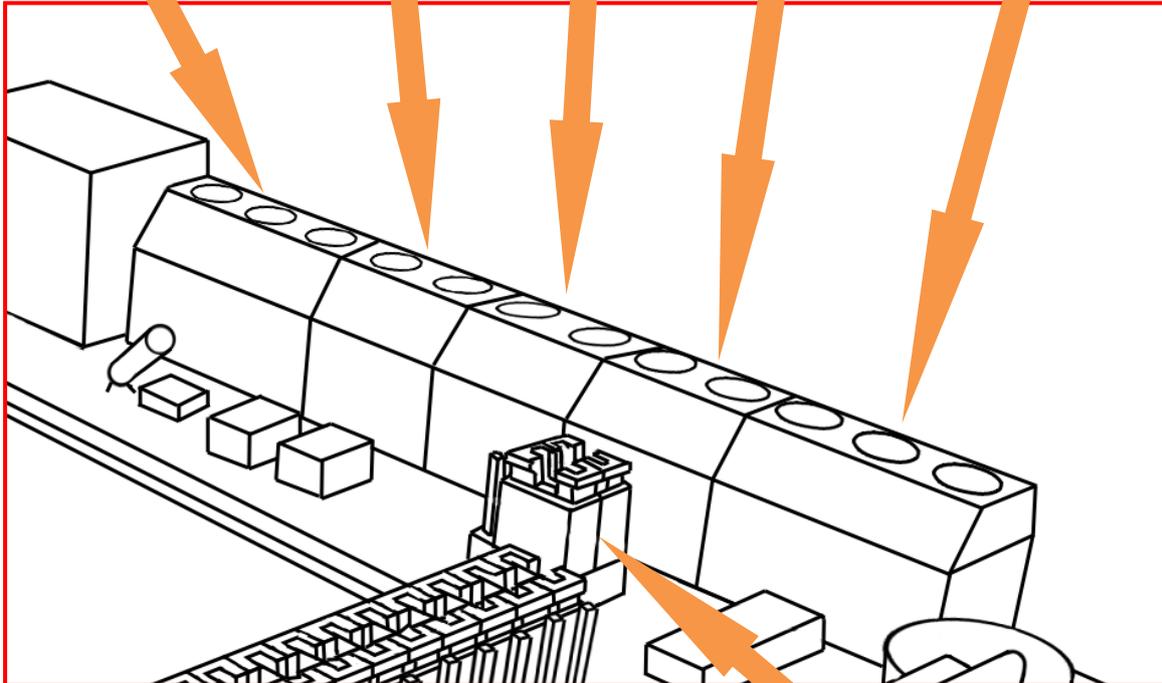
SPDT リレー出力  
ポート

フォトカ  
プラ入力。

フォトカ  
プラ出力。

RS-485 インタ  
フェース, 左か  
ら右への順  
番 : B, A。

RS-232 インタフェ  
ース, 左から右へ順  
番 : RXD, GND, TXD。

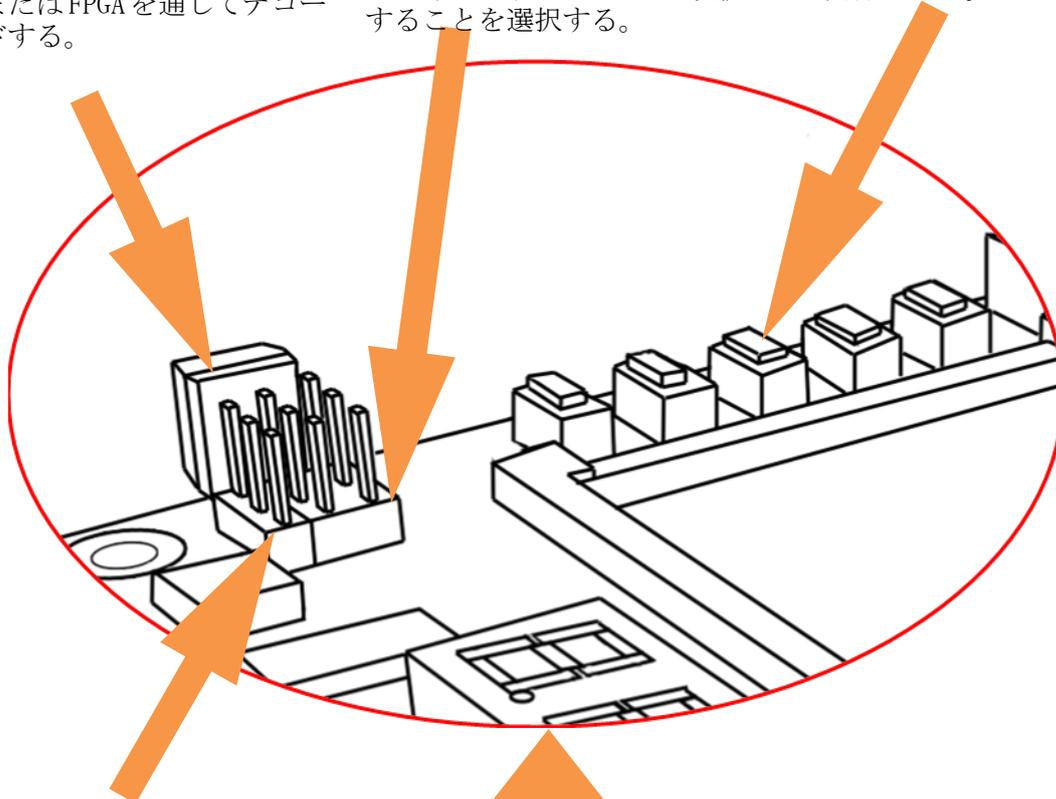


RS-232/RS-485 インタフェ  
ース選択、左側へジャンプする  
時 RS-485 インタフェ  
ースを選択し、右側へジャンプする  
時 RS-232 インタフェ  
ースを選択し、図のように RS-232 イ  
ンタフェースを選択する。

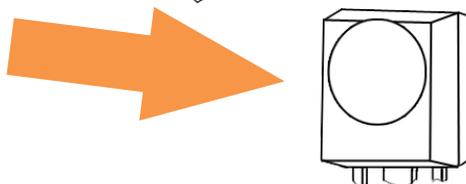
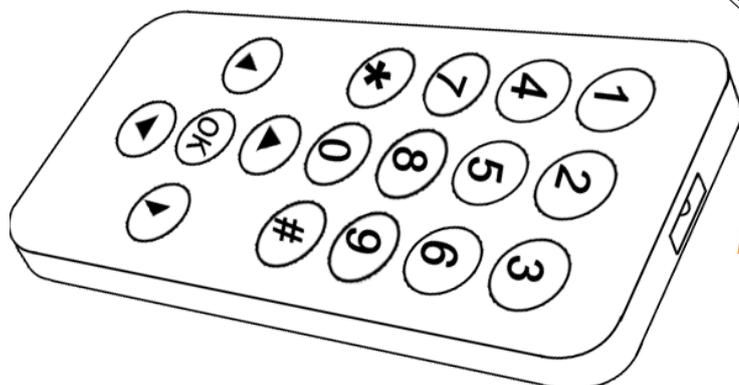
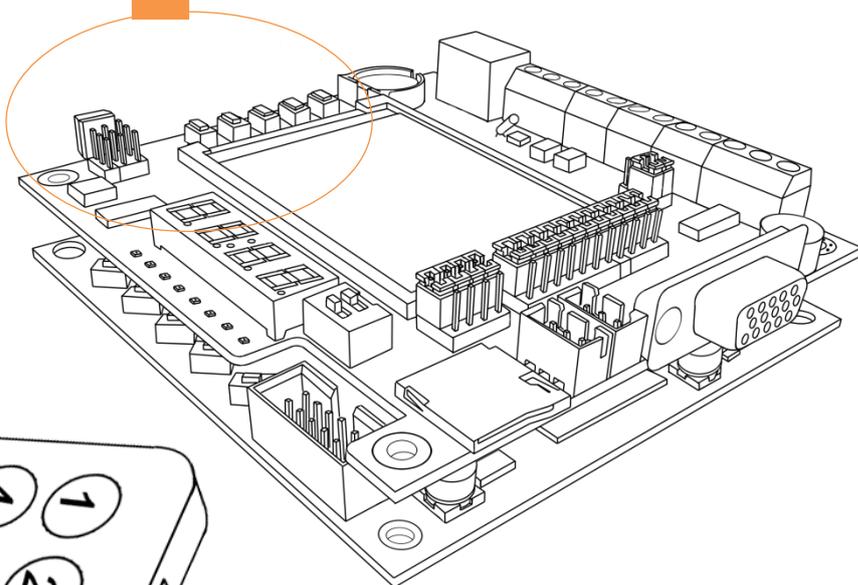
赤外線受信器、赤外線リモコン信号を受信でき、ARM または FPGA を通じてデコードする。

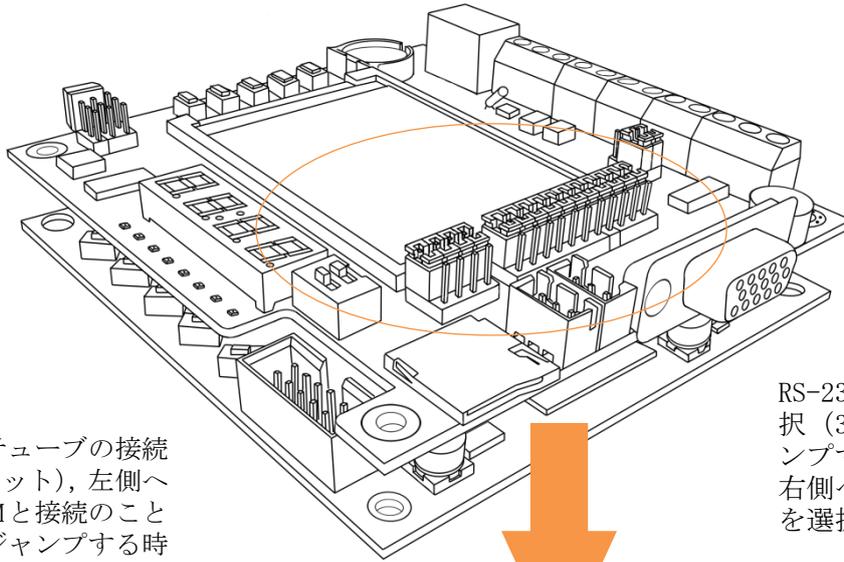
I<sup>2</sup>C バス・ジャンパー (2 ビット), 左側へジャンプする時 ARM を接続することを 選択し, 右側へジャンプする時 FPGA を接続 することを 選択する。

五個スクリーン仮想ボタ ン, それぞれ F1~F5 で, こ の機能はスクリーン表示と 関係がある。



赤外線受信器ジャンパー (1 ビット), 左側へジャンプす る時 ARM を接続するこ とを 選択し, 右側へジャンプす る時 FPGA を接続するこ とを 選択する。

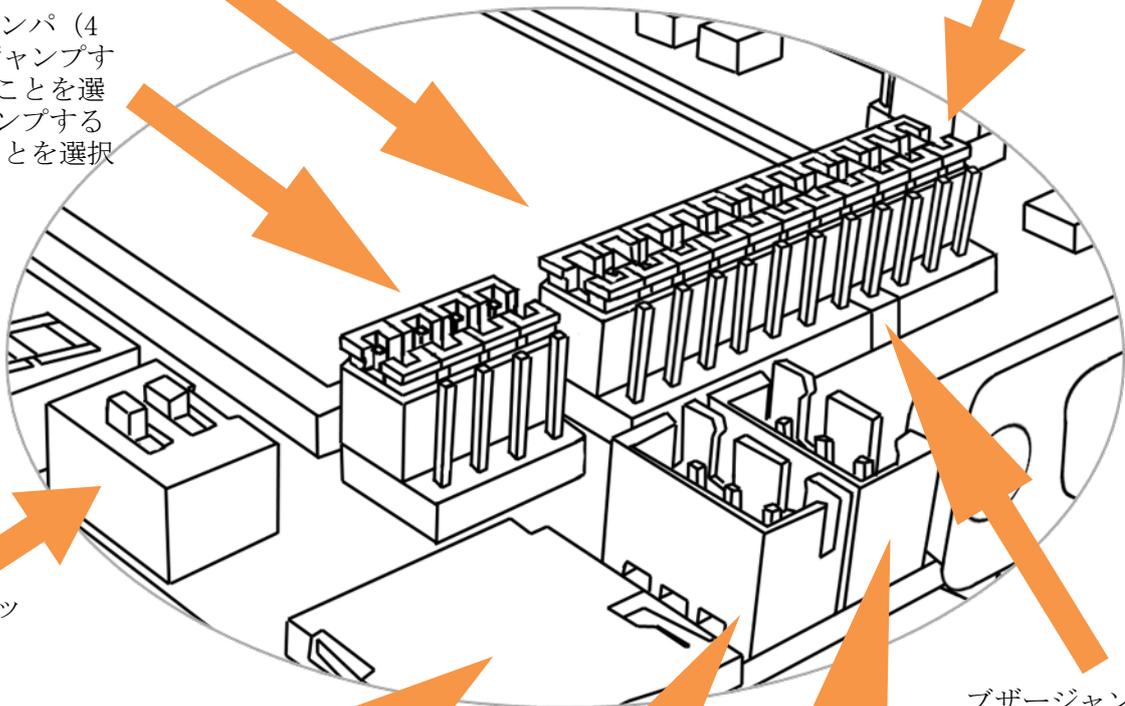




ボタン、デジタルチューブの接続  
ジャンパ選択 (7 ビット), 左側へ  
ジャンプする時 ARM と接続のこ  
とを選択し, 右側へジャンプする  
時 FPGA と接続のこ  
とを選択する。

RS-232/RS-485 ジャンパ選  
択 (3 ビット), 左側へジャン  
プする時 ARM を選択し,  
右側へジャンプする時 FPGA  
を選択する。

SD カード選択ジャンパ (4  
ビット), 左側へジャンプす  
る時 ARM と接続のこ  
とを選択し, 右側へジャンプす  
る時 FPGA と接続のこ  
とを選択する。



2 ビットディ  
ップスイッチ

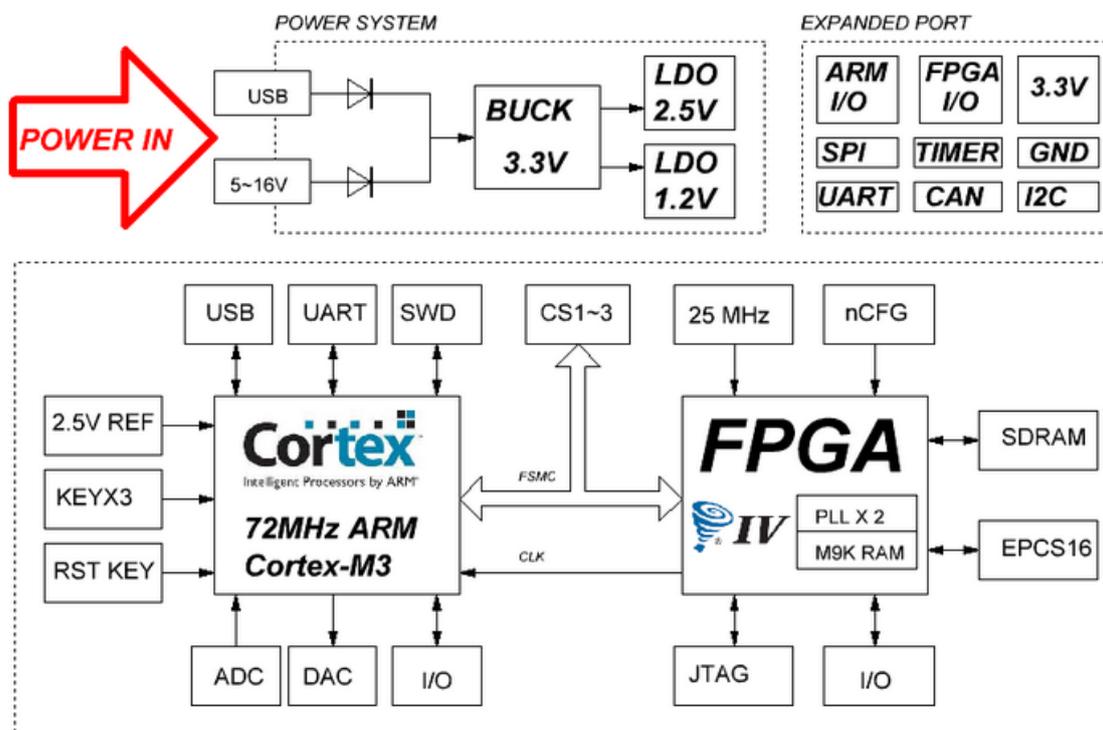
自動ポップアップ式  
SD カードスロット

ADC/DAC 端子, 左から右へそ  
れぞれ ADC 入力, GND と DAC  
出力

PWM 出力と周波数計測端  
子, 左から右へそれぞれ  
PWM 出力, GND, 周波数計  
測入力。

ブザージャンパー (1  
ビット), 左側へジャン  
プする時 ARM を選  
択し, 右側へジャン  
プする時 FPGA を選  
択する。

## 4. システムブロック図



## 5. シミュレーション USB メモリモードに入る

KEYA ボタンを押したまま、開発ボードに電源を入れ、または ARM プログラムをリセットし (ARM リセットボタンを押す)、拡張ボード LCD は“virtual u……” を表示されるまで、仮想 USB メモリモードに入ってしまう。USB インタフェースを通じて、コンピュータと接続し、拡張ボードにある 2M SPI Flash を USB メモリとしてシミュレーションされる。USB メモリにはフォントライブラリなどのシステム必要なファイルが含まれる。それを読み込み・書き換えることができる。シミュレーション USB メモリモードに入る二つ条件：

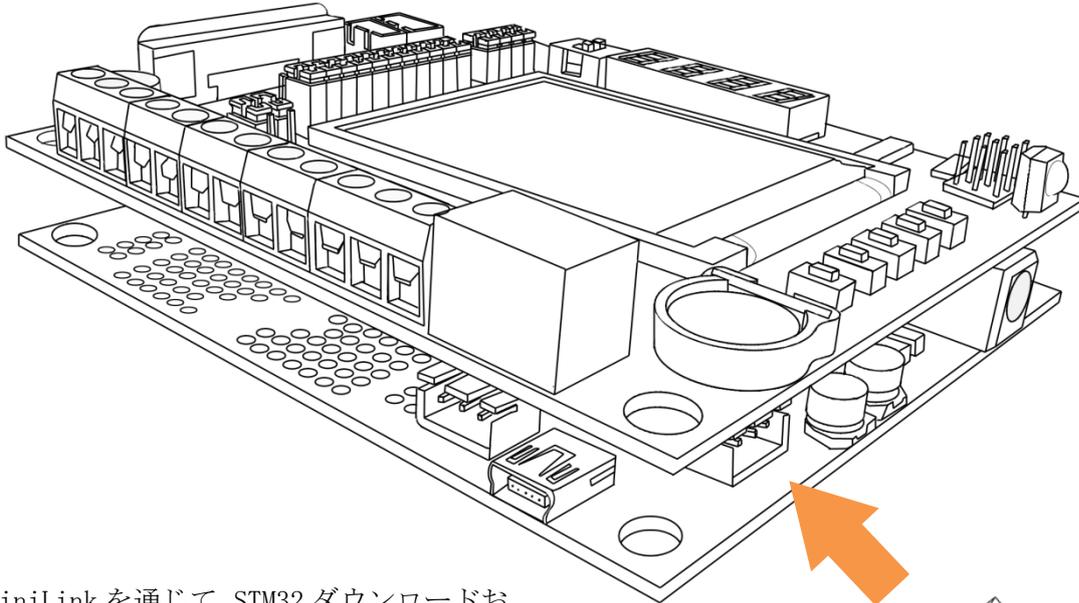
- 1、拡張ボードと接続しなければならない (拡張ボード自分が持っている 2M SPI Flash を利用する)
- 2、シミュレーション USB メモリ機能を含めるプログラムを実行しなければならない、多くの拡張ボードサンプルにはこの機能が含まれている。



## 6. LCD スクリーンキャプチャー

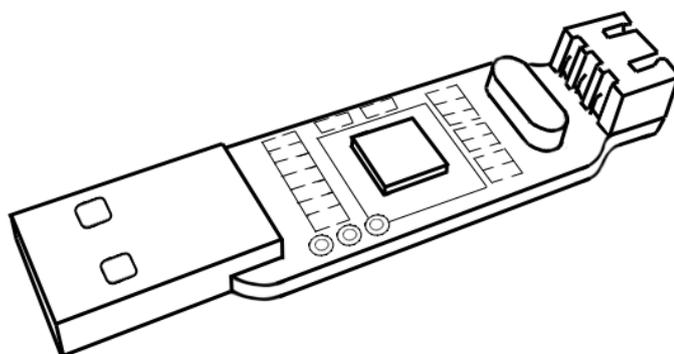
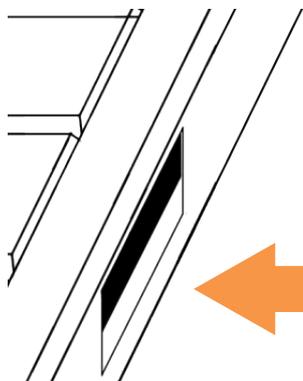
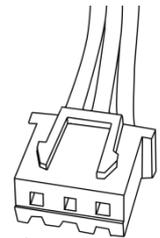
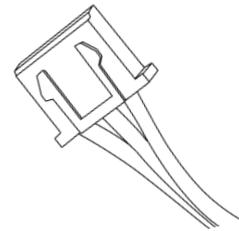
プログラムを実行する時、KEYB と KEYC を同時に押すことにより、スクリーンキャプチャー機能を起動させる。LCD スクリーンには Capture success を表示し、この時システムはスクリーンハードコピーを仮想 USB メモリの capture ディレクトリに保存し、シミュレーション USB メモリモードに入ることよりその画像を読み取る。容量制限のため、キャプチャーファイルをすぐ削除したほうが良い。

## 7. MiniLink と接続&デバッグ



弊社 MiniLink を通じて、STM32 ダウンロードおよびシミュレーション機能を行える。方法は次のとおりであり：

1. KEIL MDK IDE 開発環境をインストールする；
2. MiniLink ドライバをインストールする；
3. MiniLink の付属の 3P ケーブルを通じて MiniLink と開発ボードのデバッグポートを接続してからコンピュータの USB インタフェースに差し込む；
4. 開発ボードに供电する（miniUSB インタフェースまた DC 入力ポート）；
5. KEIL MDK でサンプルソースをオープンし、ARM を体験しよう。
6. ARM サンプルパスには漢字等を含めないと注意する



出荷時のデモの書き込みを例として説明する。

サンプルをダウンロードして解凍後、「¥icore-ex¥example¥demo¥stm32」に出荷時のデモがある

※次の手順を実施する前、前ページに記載された通り、ドライバーインストール、MiniLink とボードの接続、 MiniUSB でボードを PC と接続をする

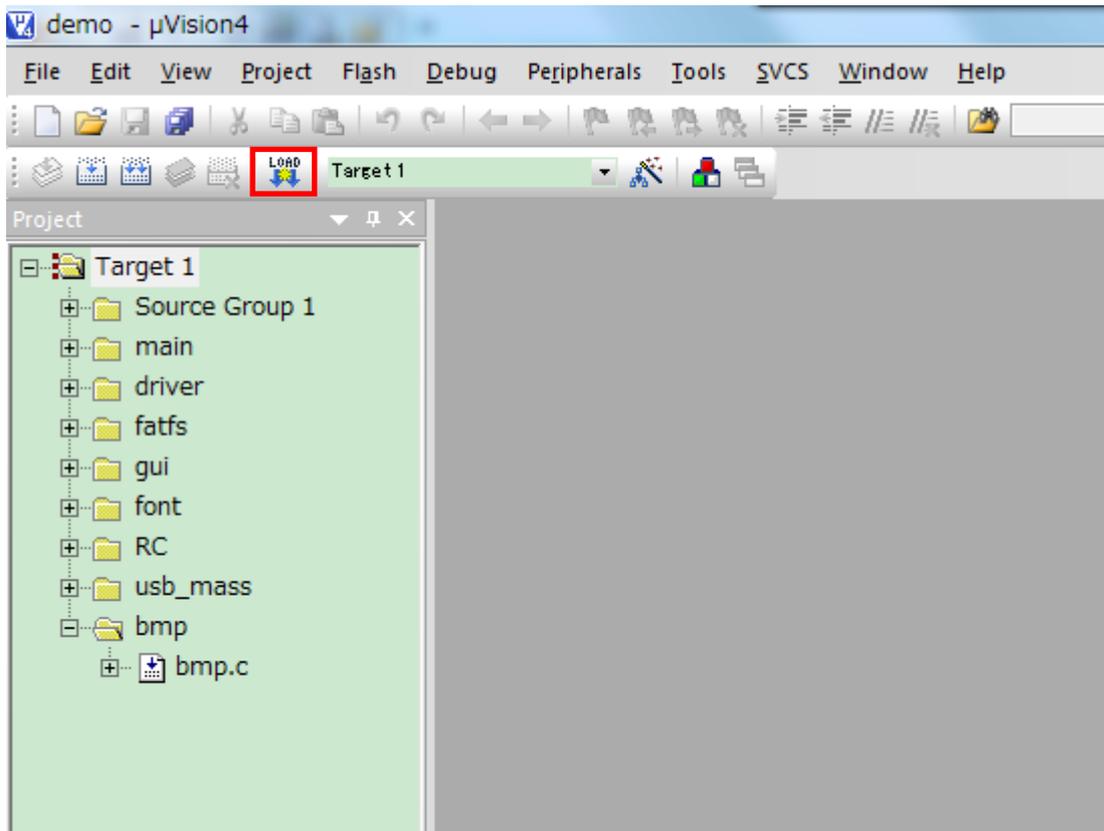
KEIL MDK IDE で Keil 用プロジェクトを開く (Keil のバージョンが 4.0 以上)

名前	更新日時	種類	サイズ
bmp	2012-10-28 16:45	ファイル フォル...	
CAPTURE	2012-10-28 19:56	ファイル フォル...	
driver	2012-10-28 16:45	ファイル フォル...	
fatfs	2012-10-28 16:45	ファイル フォル...	
Font	2012-10-28 16:45	ファイル フォル...	
GUI	2012-10-28 16:45	ファイル フォル...	
include	2012-10-28 16:45	ファイル フォル...	
main	2012-10-28 16:45	ファイル フォル...	
obj	2012-11-01 21:51	ファイル フォル...	
Resource	2012-10-28 16:45	ファイル フォル...	
usb_mass	2012-10-28 16:45	ファイル フォル...	
clear.bat	2012-08-27 16:36	Windows バッチ...	1 KB
demo.uvopt	2012-11-01 21:44	UVOPT ファイル	76 KB
demo.uvproj	2012-11-01 21:44	oVision4 Project	24 KB
demo_Target 1.dep	2012-11-01 21:51	DEP ファイル	22 KB
demo_uvopt.bak	2012-11-01 21:41	BAK ファイル	77 KB
demo_uvproj.bak	2012-11-01 21:41	BAK ファイル	24 KB
JLinkLog.txt	2012-11-01 21:51	TXT ファイル	306 KB
JLinkSettings.ini	2012-09-03 1:53	INI ファイル	1 KB
startup_stm32f10x_hd.s	2012-08-28 11:48	Assembler Source	16 KB

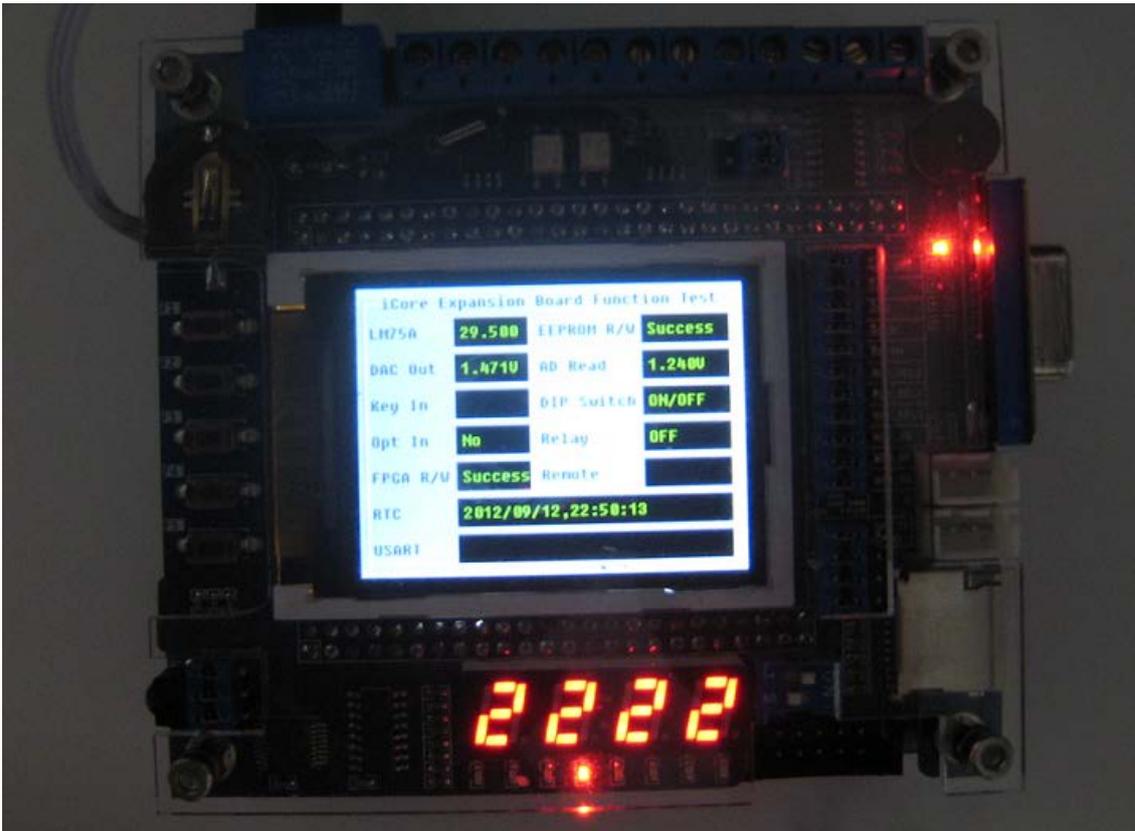
開いた後、下図のように示されます。

基本的に、書き込み用の Hex ファイルがありますので、そのまま書き込むことができる

アイコン「」をクリックし、或いは、メニュー「Flash」→「Download」をクリックする



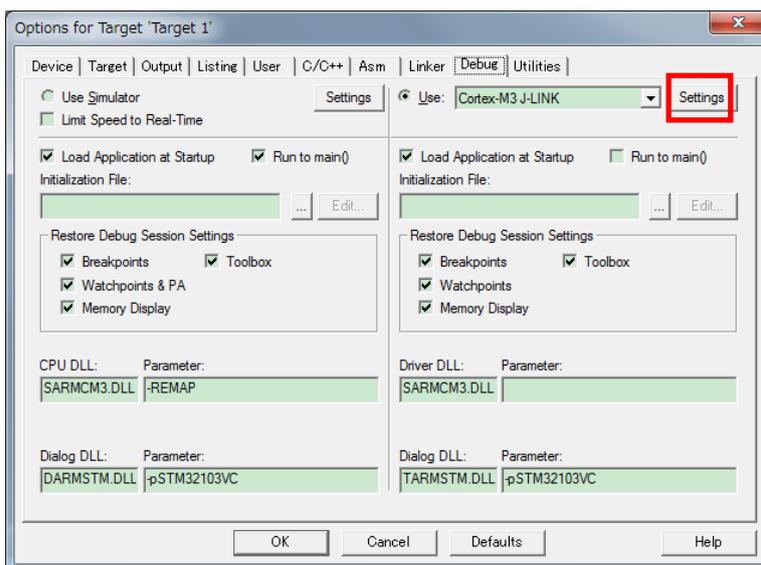
正常に書き込み後の様子：

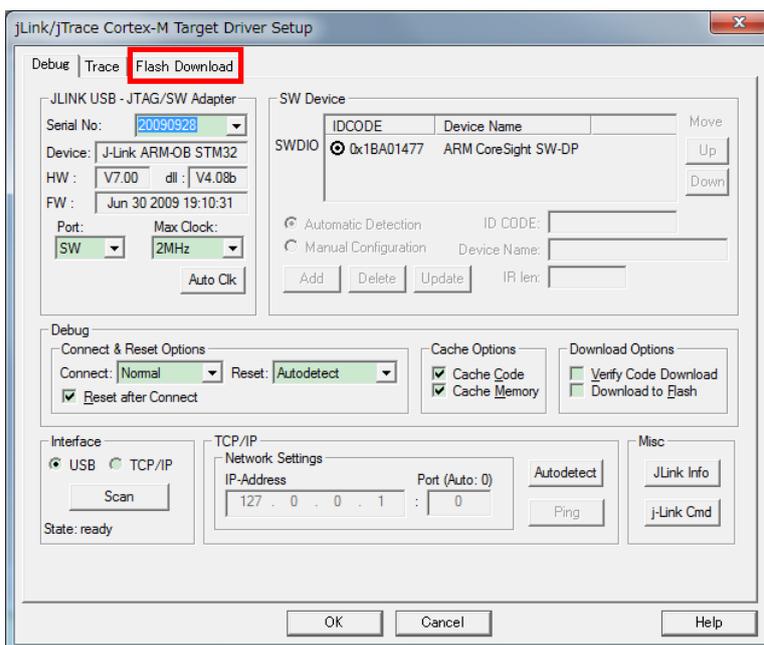


勿論、KEIL MDK IDE で修正、コンパイル、デバッグ等も行える。

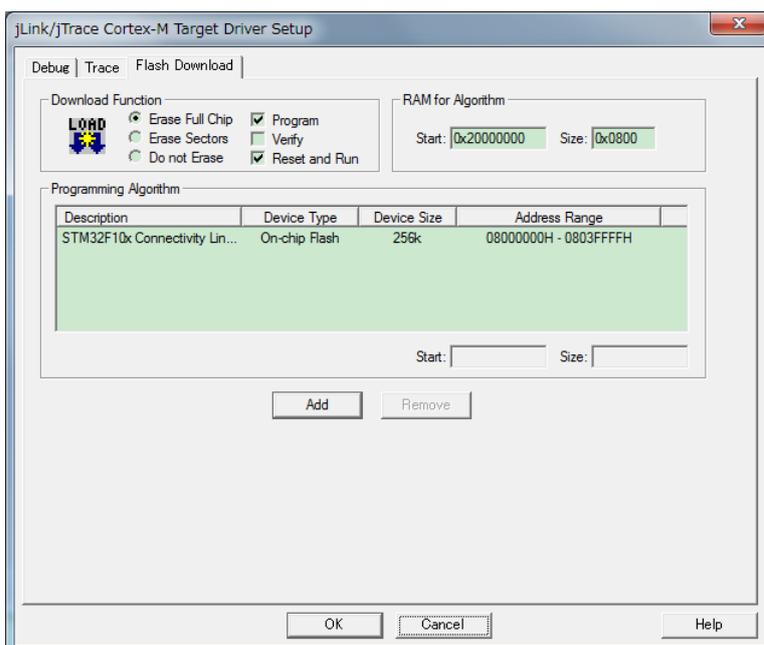
デバッグ等に関する MiniLink の設定：（基本修正不要、サンプルソースに既に設定済み）

メニュー「Project」→「Options for Target 'Target 1'」をクリックし、「Debug」タブを選べ、



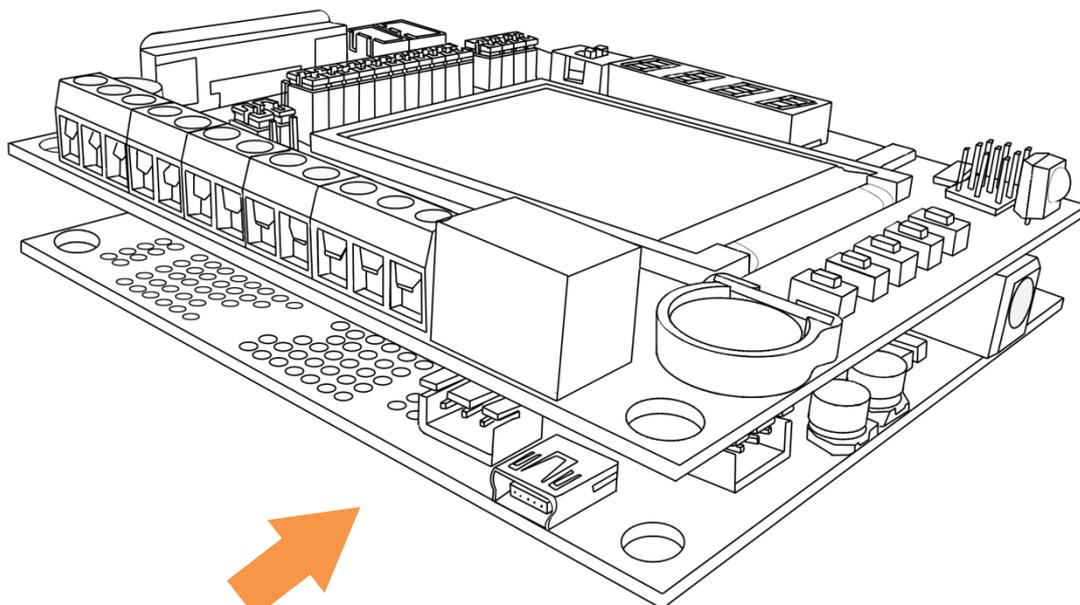


タブ「Flash Download」をクリック



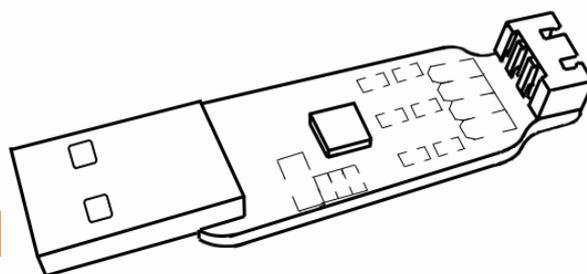
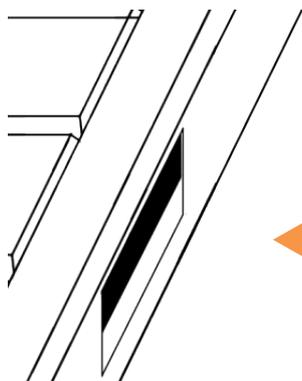
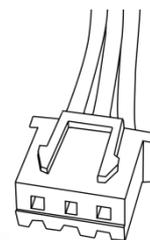
特に修正必要がないので、「Cancel」ボタンを押下して戻る  
他のサンプルも同じように体験できる。

## 8. USB シリアルポート変換モジュールで接続と通信

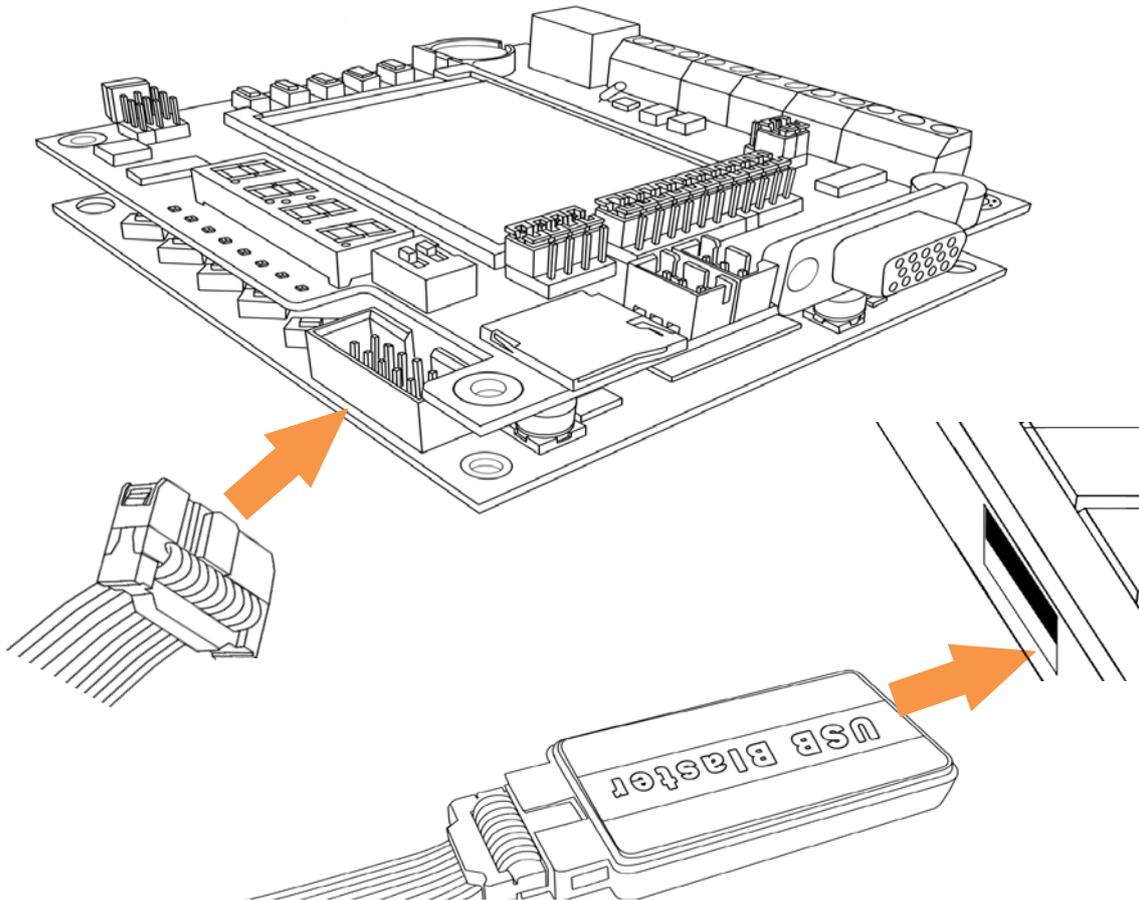


USB シリアルポート変換モジュールを利用して、コンピュータとの通信を実現する。方法は次のとおりであり：

1. CP2102 ドライバプログラムをインストールする；
2. 3p ケーブルを通じて、USB をシリアルポートに変換し、開発ボードの UART インタフェースと接続し、それから USB シリアルポート変換モジュールをコンピュータの USB インタフェースに差し込む；
3. ハイパーターミナルを使用し（或いは他のシリアル通信ツール）、仮想シリアルポートをオープンする；
4. 関連サンプルをオープンし、通信デモを行える。



## 9. Altera ダウンロードケーブルで FPGA をダウンロードする



Altera ダウンロードケーブルを利用して、FPGA 配置とデバッグを実現する。  
方法は次のとおりであり：

※ダウンロード方法は弊社 [CycloneIV ボードのマニュアル](#) を参照しても良い。  
(Note：FPGA チップが異なる、本ボード型番： EP4CE6E22C8N)

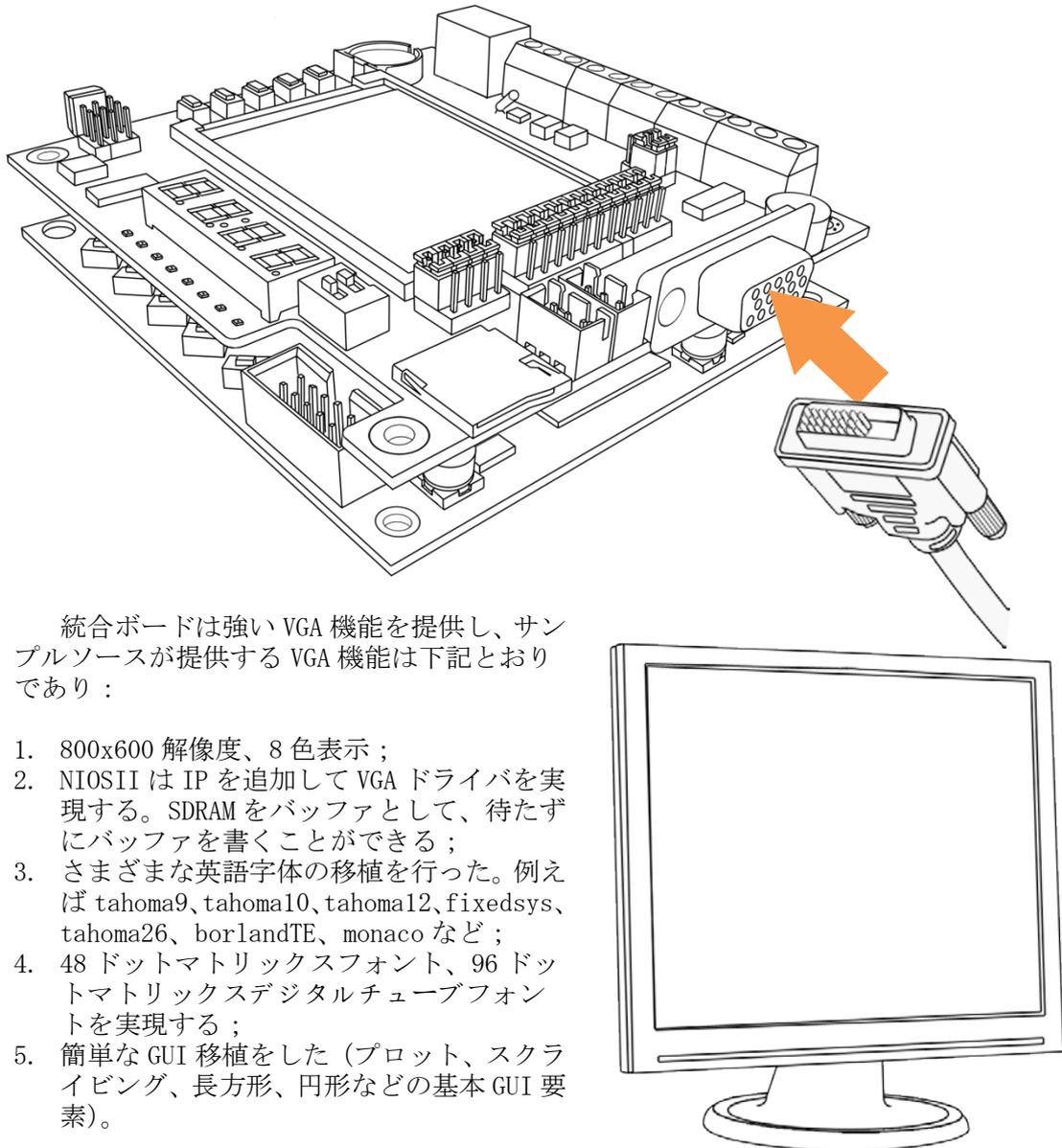
P16～P37:開発ツールをインストール

P104～P113：プログラムダウンロード

他にハードウェア・ソフトウェア開発にて該当内容を参照

1. QuartusII ソフトウェアをインストールする；
2. QuartusII の Device サポートパッケージ (Cyclone パッケージ) をインストールする；
3. Altera ダウンロードはコア基板の FPGA JTAG と接続してから上図のように PC と接続する；
4. 開発ボードに電源を入れる；
5. QuartusII プロジェクトをオープンして NIOSII プロジェクトをデバッグ

## 10、VGA インタフェースでモニタと接続する

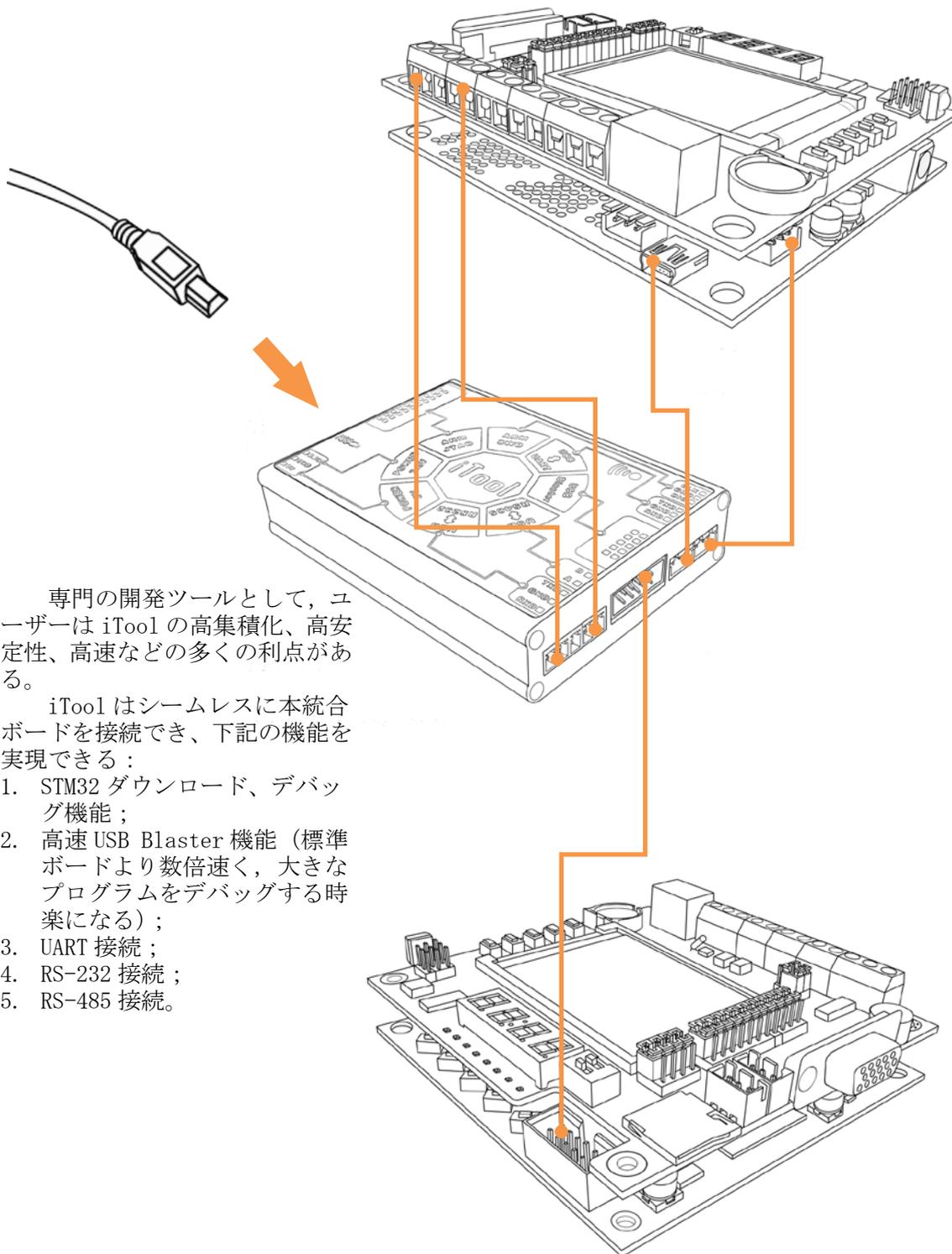


統合ボードは強い VGA 機能を提供し、サンプルソースが提供する VGA 機能は下記とおりであり：

1. 800x600 解像度、8 色表示；
2. NIOSII は IP を追加して VGA ドライバを実現する。SDRAM をバッファとして、待たずにバッファを書くことができる；
3. さまざまな英語字体の移植を行った。例えば tahoma9、tahoma10、tahoma12、fixedsys、tahoma26、borlandTE、monaco など；
4. 48 ドットマトリックスフォント、96 ドットマトリックスデジタルチューブフォントを実現する；
5. 簡単な GUI 移植をした（プロット、スクライビング、長方形、円形などの基本 GUI 要素）。

出荷デモプログラムは VGA 表示機能を含め、標準モニターとの接続を通じて開発ボードの強い機能をデモンストレーションする。

## 11. iTool で色々な開発を行う

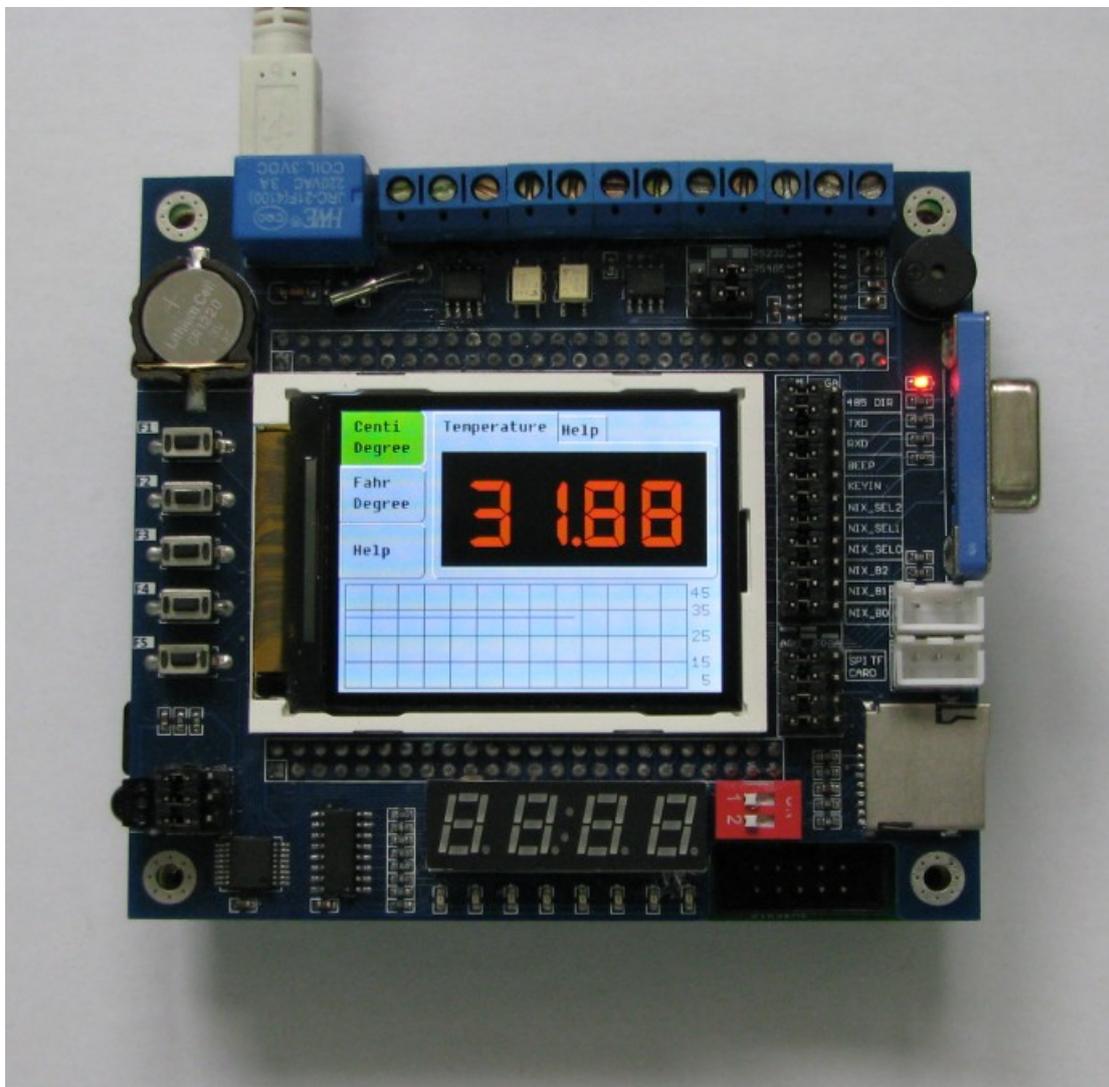


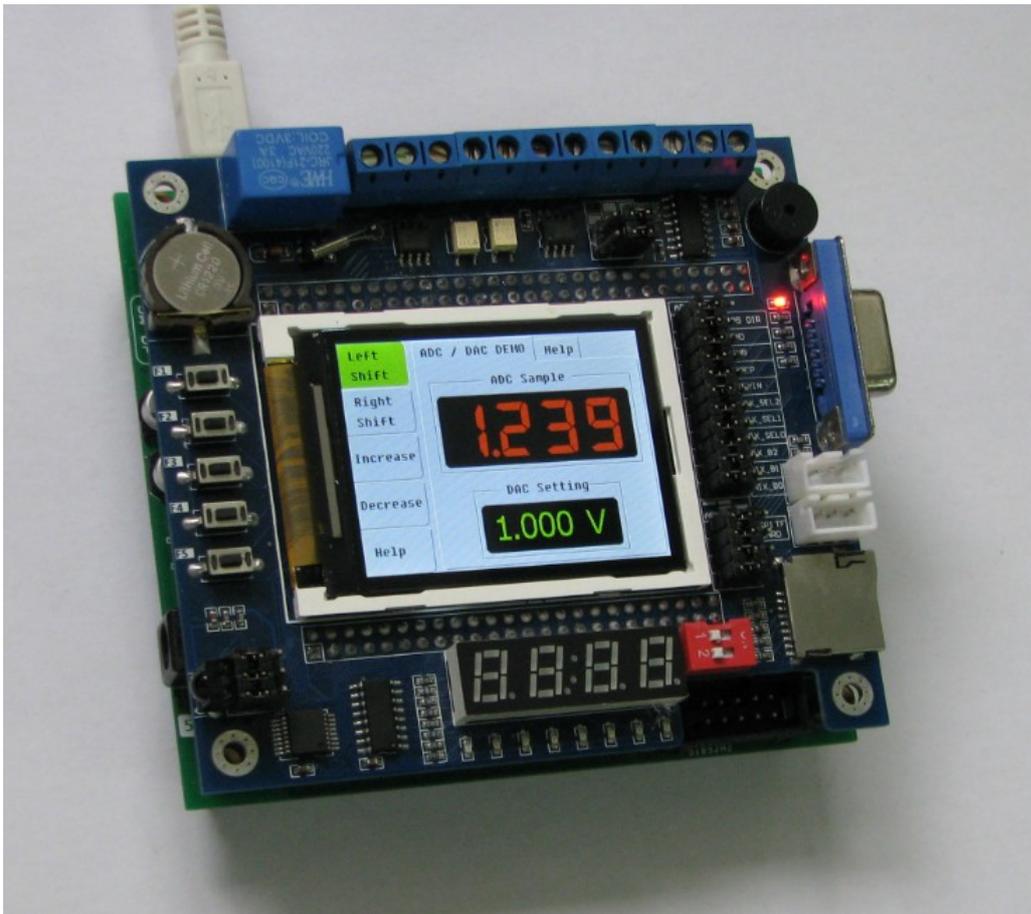
専門の開発ツールとして、ユーザーは iTool の高集積化、高安定性、高速などの多くの利点がある。

iTool はシームレスに本統合ボードを接続でき、下記の機能を実現できる：

1. STM32 ダウンロード、デバッグ機能；
2. 高速 USB Blaster 機能（標準ボードより数倍速く、大きなプログラムをデバッグする時楽になる）；
3. UART 接続；
4. RS-232 接続；
5. RS-485 接続。

## 12. 拡張板と一緒に動く様子





### 13. 豊富なサンプル

#### ①コアボードのみの場合



サンプルソース説明：

¥code

1_led	サイクル LED 点灯サンプル
2_uart	シリアルポートサンプル
3_shell	STM32 でシリアルポートの発受信機能を実現する
4_systick	STM32 のタイマーでデジタルチューブから 0000～9999 を繰り返して表示する
5_key	ユーザーボタンサンプル

- 6\_key\_interrupt ユーザーボタン割り込みサンプル
- 7\_fsmc STM32 から FSMC を利用して FPGA を読み込む・書込む事をテストする
- 8\_dac D/A 変換サンプル
- 9\_adc A/D 変換サンプル
- 10\_device\_id デバイス ID を読み取るサンプル

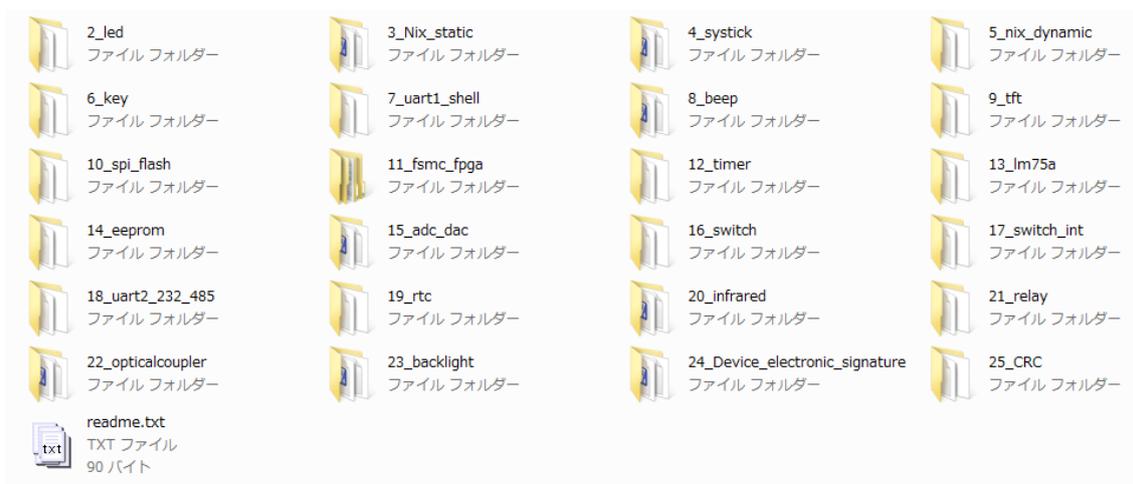
¥demo

- bin FPGA で SDRAM の読み込む・書込むテストデモ  
\*JTAG を利用して EPCS に書き込む
- LOGIC FPGA デモ
- stm32 STM32 のデモ (出荷時基本的に拡張板用のデモを書き込まれている)

¥fpga\_tcl

- iCore.tcl コアボードの FPGA ピン配置 TCL ファイル

## ②拡張ボードと合わせての場合



サンプルソース説明 :

¥code

- 2\_led STM32 で LED 点灯を制御する
- 3\_Nix\_static STM32 でデジタルチューブを制御する
- 4\_systick STM32 のタイマーでデジタルチューブから 0000~9999 を繰り返して表示する
- 5\_nix\_dynamic STM32 でデジタルチューブの動的表示を制御する
- 6\_key STM32 からユーザーボタンを認識し、押されたキーのコードをデジタルチューブで表示される (五つキーのコード : 1, 2, 4, 8, 16)



7_uart1_shell	STM32 でシリアルポートの発受信機能を実現する
8_beep	STM32 で異なる周波数の音をブザーから生み出す事を制御する
9_tft	STM32 で TFT 表示を制御する
10_spi_flash	STM32 で SPI バスを通して Flash 情報を読み込む
11_fsmc_fpga	STM32 から FSMC を利用して FPGA を読み込む・書込む事をテストする
12_timer	STM32 で普通タイマーTimer2 のアラームとカウンタ機能を実現し、結果を液晶に表示させる
13_lm75a	STM32 で LM75A 温度センサーを読み込み、温度値も液晶に表示される
14_eeeprom	STM32 で EEPROM を読み込む・書込むサンプル
15_adc_dac	STM32 で A/D 変換機能と D/A 変換機能を実現する
16_switch	ディップスイッチで STM32 の I/O 出力レベル（高あるいは低）を制御する
17_switch_int	ディップスイッチで STM32 の割り込みを制御する
18_uart2_232_485	STM32 シリアルポートで RS232 と RS485 を利用して発受信するサンプル
19_rtc	STM32 で時計チップ DS1302 内部情報を読み込む・制御し、同時に時間情報を液晶に表示される
20_infrared	STM32 でリモコンから赤外線情報を受信し、赤外線情報を解析してからリモコンの押されたキーのコードが得られ、このコードを液晶に表示される
21_relay	STM32 でリレーを制御するサンプル
22_opticalcoupler	STM32 でフォトカプラ入力・出力を制御する
23_backlight	STM32 でバックライトの明るさを制御する
24_Device_electronic_signature	STM32 で CPU と電子署名情報を液晶に表示する
25_CRC	STM32 で CRC ユニットの演算機能を実現する

## ¥demo

fpga	FPGA デモ
stm32	出荷時のデモ

## ¥fpga\_tcl

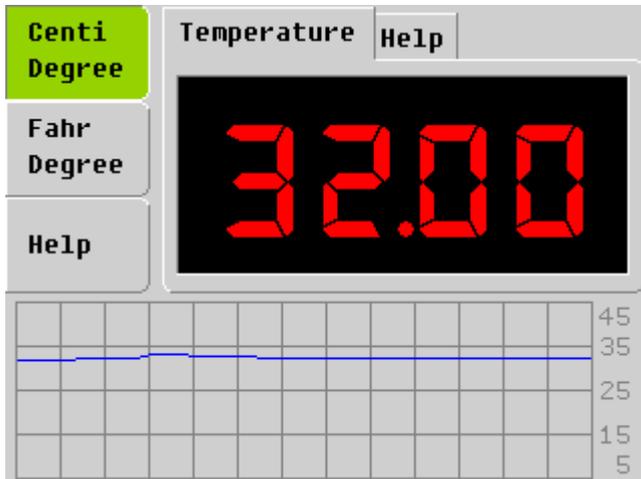
iCore_ex01_pin.tcl	コアボードと拡張ボードの FPGA ピン配置 TCL ファイル
--------------------	---------------------------------

### ③一部サンプルの動く様子

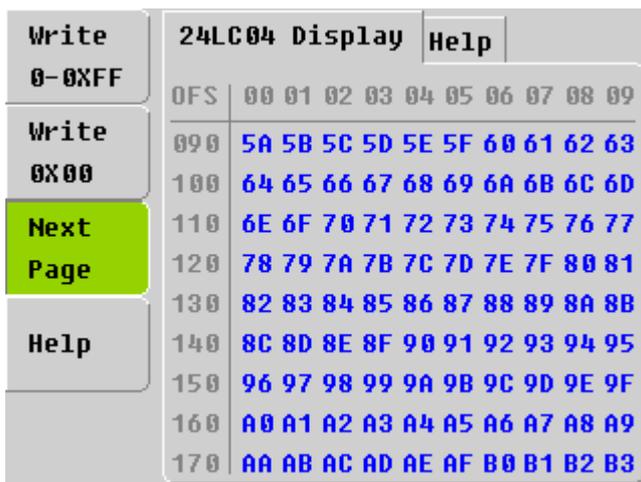
## 1) タイマー :



2) 温度計測 :



3) EEPROM :



OFS	00	01	02	03	04	05	06	07	08	09
090	5A	5B	5C	5D	5E	5F	60	61	62	63
100	64	65	66	67	68	69	6A	6B	6C	6D
110	6E	6F	70	71	72	73	74	75	76	77
120	78	79	7A	7B	7C	7D	7E	7F	80	81
130	82	83	84	85	86	87	88	89	8A	8B
140	8C	8D	8E	8F	90	91	92	93	94	95
150	96	97	98	99	9A	9B	9C	9D	9E	9F
160	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
170	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3

4) ADC/DAC :



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

