

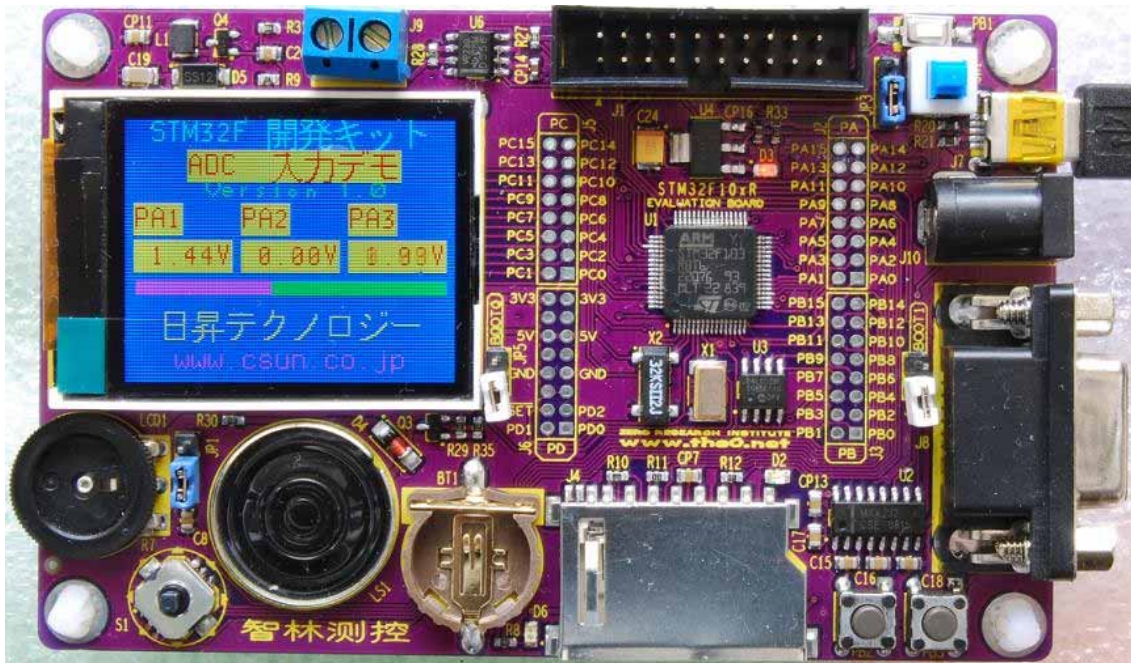
ARM Cortex-M3 STM32F103 開発キット (GCC TOPPERS/ASP 対応) マニュアル

株式会社日昇テクノロジー

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

info@csun.co.jp

2009/3/22



copyright@2009



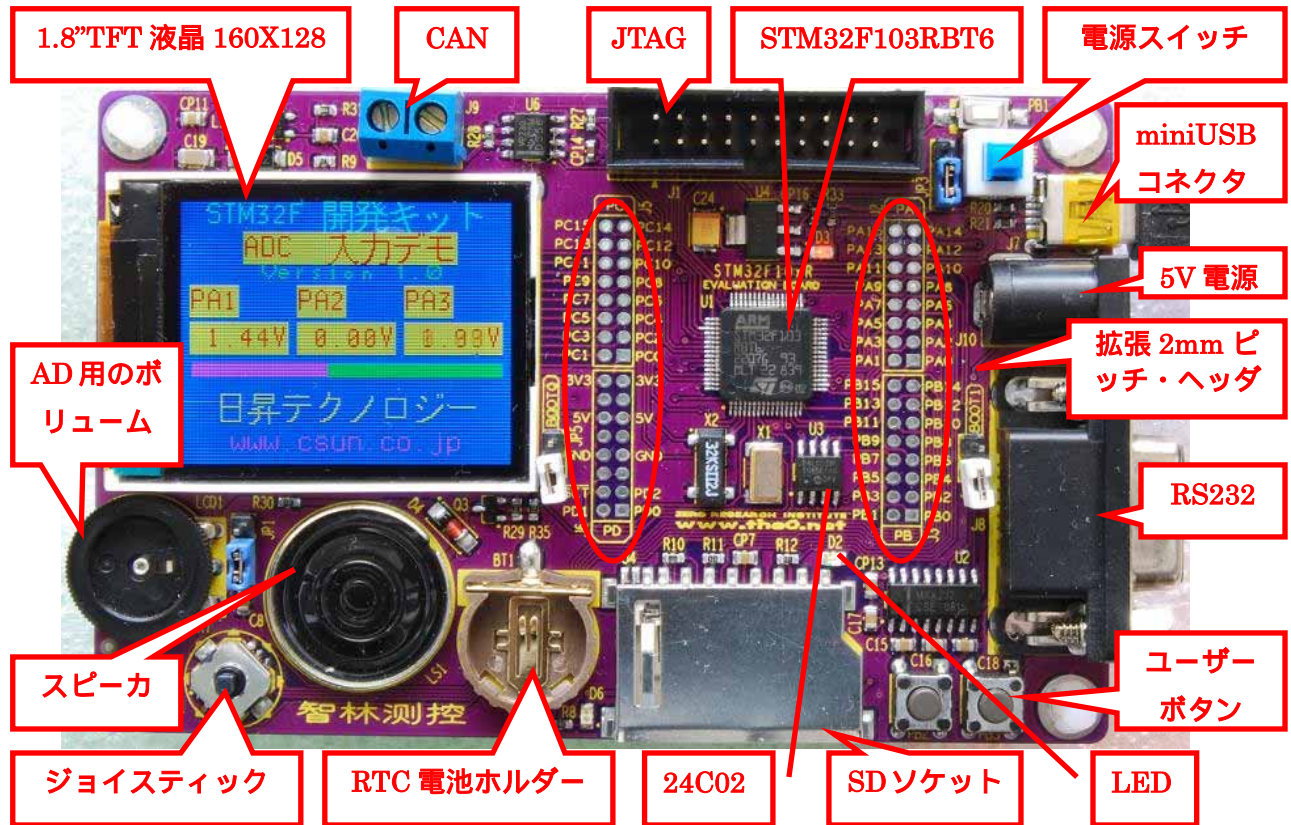
第一章STM32F103 開発キットの概要.....	4
第二章「STM32」シリーズの概要.....	5
第三章 初体験.....	9
3.1 デフォルトのサンプル.....	9
3.2 書き込みツールのインストール.....	10
3.3 書き込み.....	14
3.4 LED点灯.....	18
3.5 PWM.....	19
3.6 液晶LCD.....	20
3.7 漢字のディスプレイ.....	21
3.8 ピクチャのディスプレイ.....	21
3.9 GPIOの入力実験.....	22
3.10 ゲーム.....	23
3.11 ADC入力実験.....	23
3.12 PWM音声.....	24
3.13 I2C EEPROM.....	25
3.14 USB メモリ.....	25
3.15 SDカード.....	27
3.16 シリアル通信実験.....	28
3.17 CANシリアル通信実験.....	29
3.18 タイマーの割り込み実験.....	30
3.19 RTC.....	31
3.20 外部割り込み実験.....	31
3.21 Tamper実験.....	32
3.22 ウォッチドッグ・タイマ.....	33
3.23 ソフトウェア割り込み.....	34
3.24 uC/OS IIデモ.....	35
第四章 開発ツールKEILのインストール.....	36
4.1 KEILのインストール.....	36
4.2 ライブラリのインストール.....	39
4.3 ライセンス.....	42
4.4 既存のプロジェクトから.....	42
4.5 漢字の表示.....	44
第五章 Thumb-2 対応GCCクロス開発環境.....	46
第六章 TOPPERS/ASP.....	48



6.1 TOPPERS/ASPの初体験.....	48
6.2 TOPPERS/ASPのコンパイル.....	49
6.3 H-JTAGで書き込む.....	51

使用されたソースコードは<http://www.csun.co.jp/>からダウンロードできます。

第一章 STM32F103 開発キットの概要



CPU プロセッサー

- ARM コア新系列プロセッサー-Cortex-M3 を採用した ST マイクロエレクトロニクス社の STM32F103RBT6(周波数 72MHz, 128KB Flash, 20KB SRAM, 2 × SPI, 2 × I2C, USB2.0 デバイス, CAN, PWM, 2 × 12 ビット ADC 16ch , 3 × USART , 3 × 16 ビット・タイム, RTC, CAN2.0B, 温度センサ)

開発キットのインターフェース


- RS232 × 1
- USB2.0 device × 1
- JTAG/ICE
- CAN2.0B × 1
- SD カードソケット
- AD テスト用のボリューム

- I2C EEPROM
- ユーザーLED ×1
- ユーザーボタン×2
- スピーカー×1
- 1.8 インチ TFT 液晶、分解能 160×128

外形寸法

- 外形寸法: 110×70(mm) 突起物は除く

供給電源

- 5VDC 電源、プラグ 2.1mmφ、極性はセンタープラス  です。電源スイッチと電源指示 LED 付き

第二章「STM32」シリーズの概要

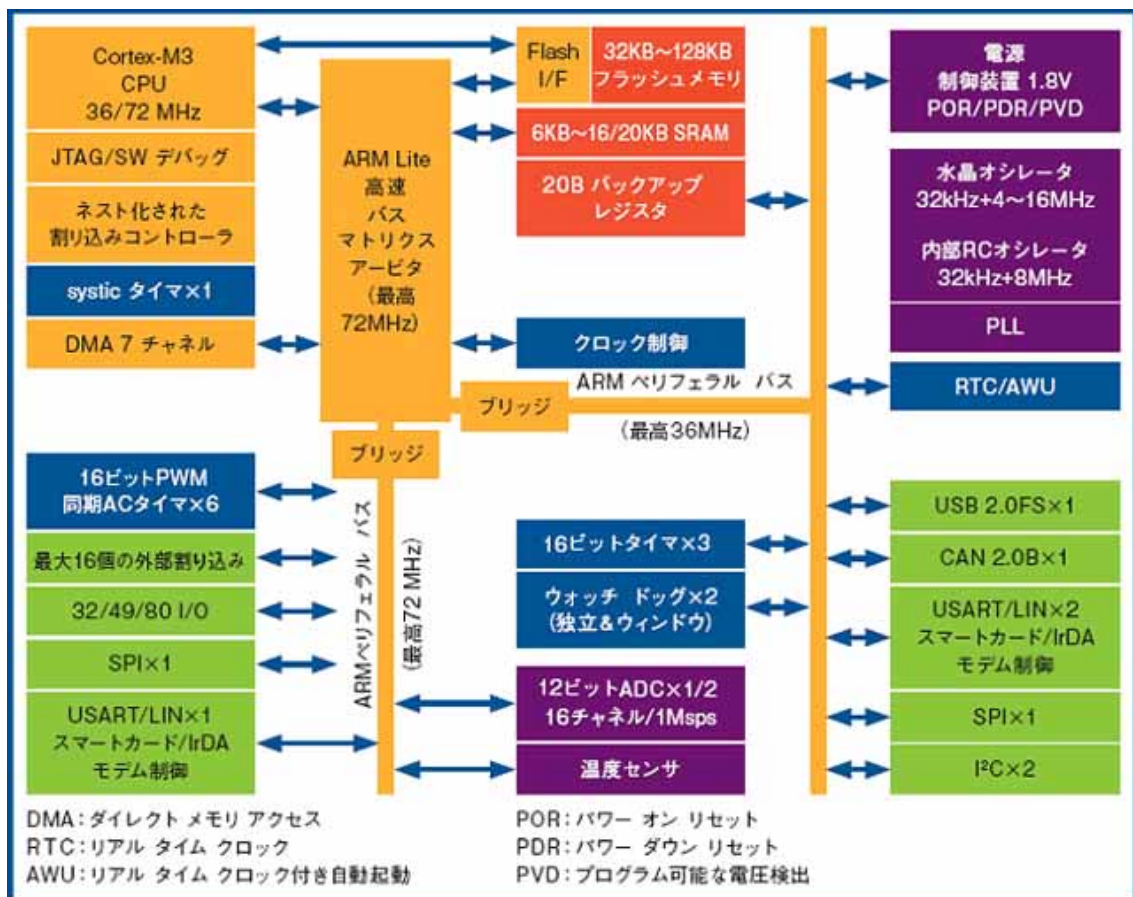
組み込み機器のプロセッサ(またはマイクロコントローラ)において、ARM プロセッサは高いシェアを占めている。ARM プロセッサとは、英 ARM が提供するプロセッサコア(Arm コア)を採用したプロセッサのことで、さまざまな半導体ベンダから数多くの種類の ARM プロセッサが出荷されている。

最近では、あらゆる組み込み機器において「低消費電力」と「高性能」という 2 つの要求を同時に実現することが極めて重要となっているが、ARM コアはこれを満たすことを特徴としている。また、各半導体ベンダにおいても、消費電力を抑えながらも性能を上げるために、さまざまなしくみを設けている。

ARM プロセッサの一例として、STMicroelectronics(以下、ST)の「STM32」シリーズに注目し、その低消費電力のためのしくみについて述べていく。STM32 シリーズは、ARM コアである「Cortex-M3」コアを採用した 32 ビットマイクロコントローラ(以下、マイコン)で、消費電力を 16 ビットマイコンと同等までに抑えていることを特徴としている。



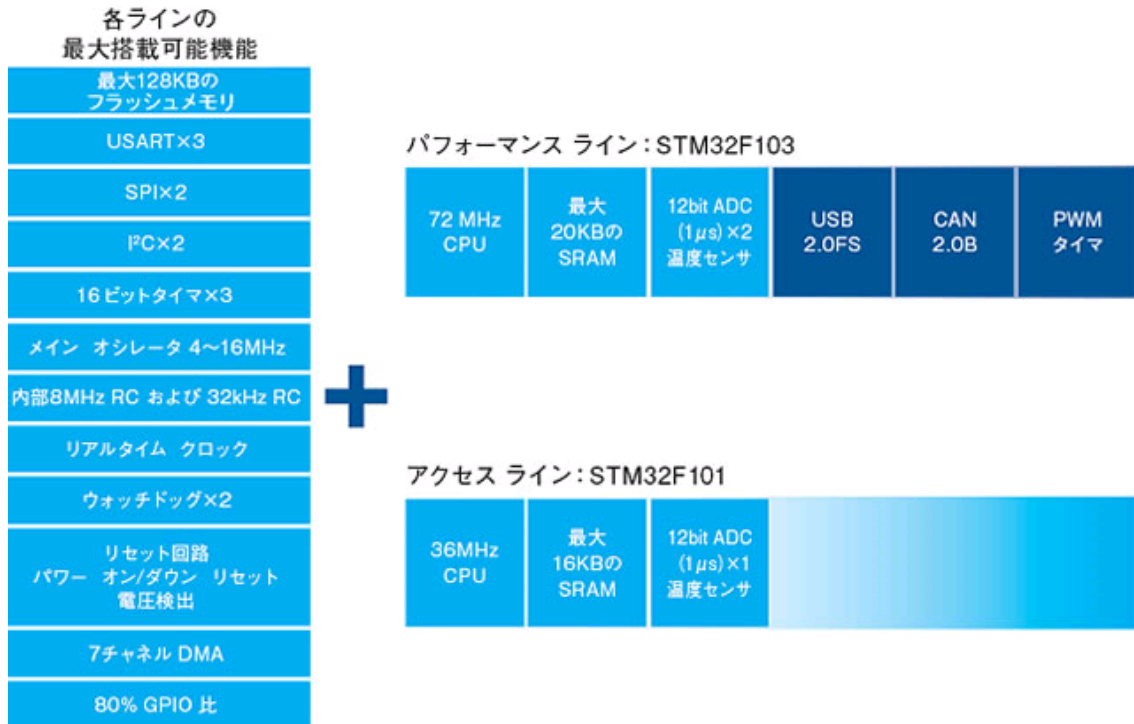
STM32 シリーズは、ARM の Cortex-M3 コアを採用した 32 ビットマイコンである。STM32 シリーズは、Cortex-M3 コアにより、パフォーマンスとコード密度の向上を実現する Thumb-2 命令セット、および割り込みに対する応答を大幅に改善するためにネスト化されたベクタ割り込みコントローラなどを搭載している。

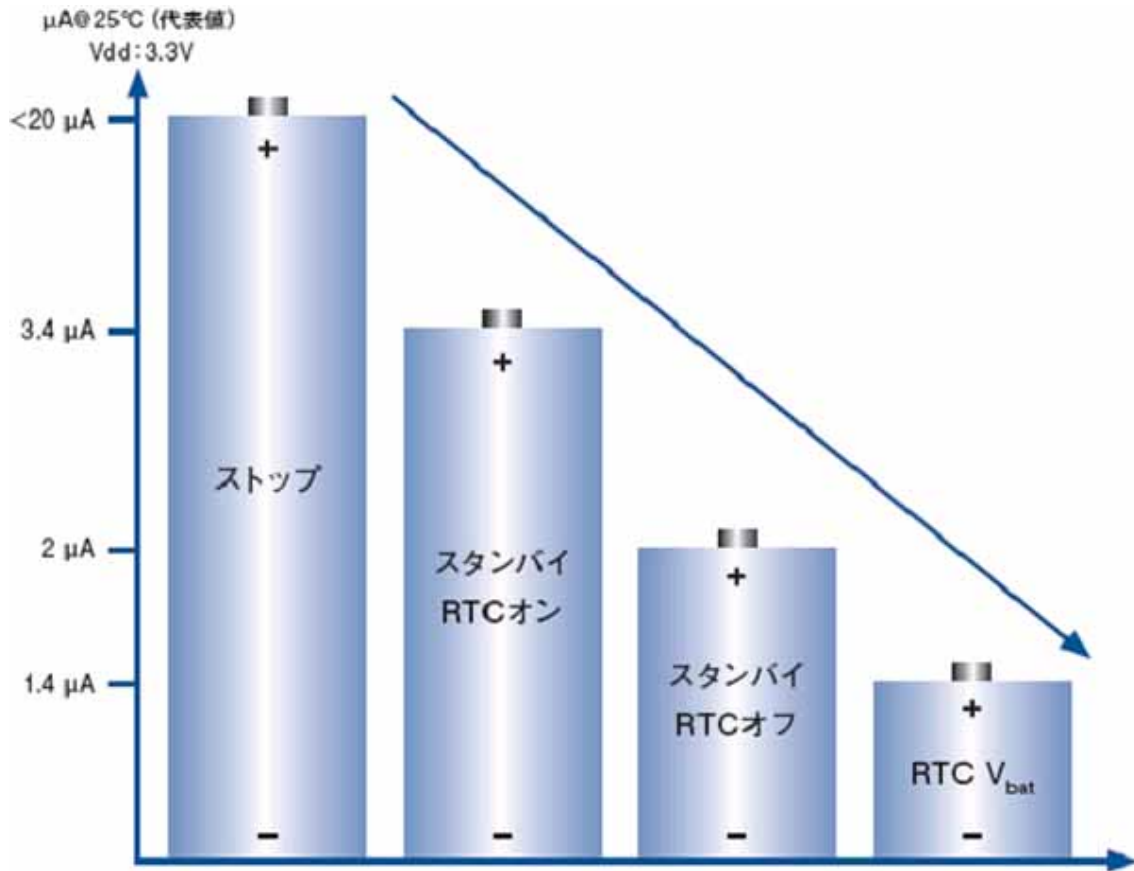


STM32 シリーズは、32K バイト~128K バイトのフラッシュメモリと、6K バイト~20K バイトの RAM を内蔵している。さらに、16 ビットマイコンクラスの「アクセスライン」と 32 ビットマイコンクラスの「パフォーマンスライン」の両方を用意しており、それぞれがピン配置の互換性をもっている。

「パフォーマンスライン」である STM32F103 は 72MHz で動作し、多くのオンチップ RAM

とペリフェラルを搭載。高性能でエネルギー効率に優れた 32 ビットマイコンである。また、「アクセスライン」である STM32F103 の動作クロックは 72MHz。32 ビットマイコンのパワーを持ちながら、コストは 16 ビットマイコン程度まで抑えている。





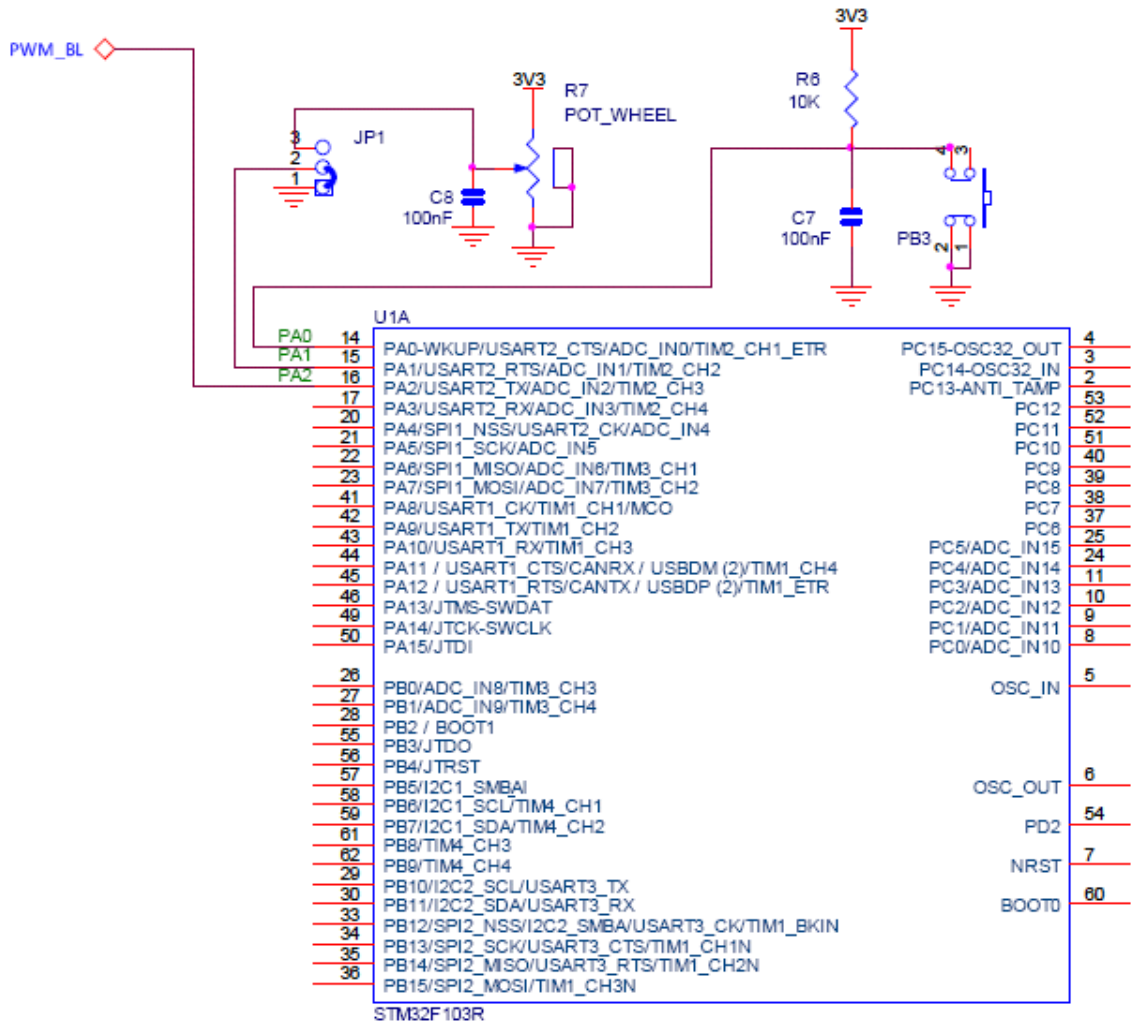
- 低電圧: 2.0V~3.6V
- フラッシュメモリからの実行モードで0.5mA/MHz
(ペリフェラルクロック有効)
- ストップモードからの立ち上がり時間: $<10\mu\text{s}$
スタンバイモードからの立ち上がり時間: $40\mu\text{s}$
- リセット回路は常時アクティブ

第三章 初体験

3.1 デフォルトのサンプル



STM32F103 開発キットに書き込んだデフォルトのサンプルは ADC デモプログラムです。TFT 液晶で AD の結果が表示されます。ボリュームを回すと、TFT 液晶でどんな変化が起きますか、やってみます。



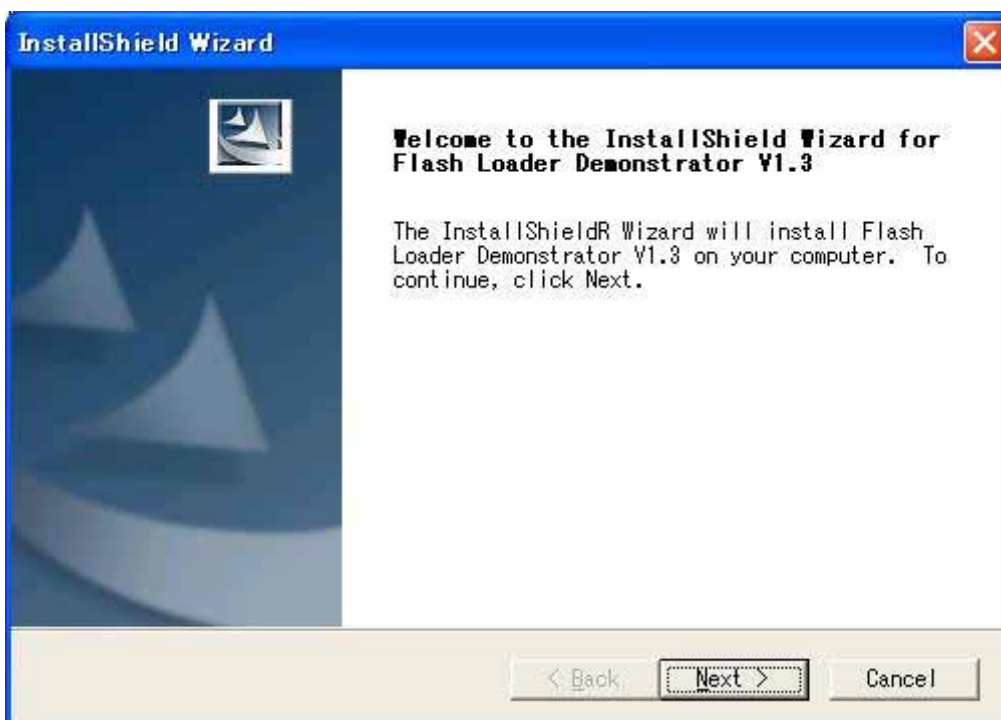
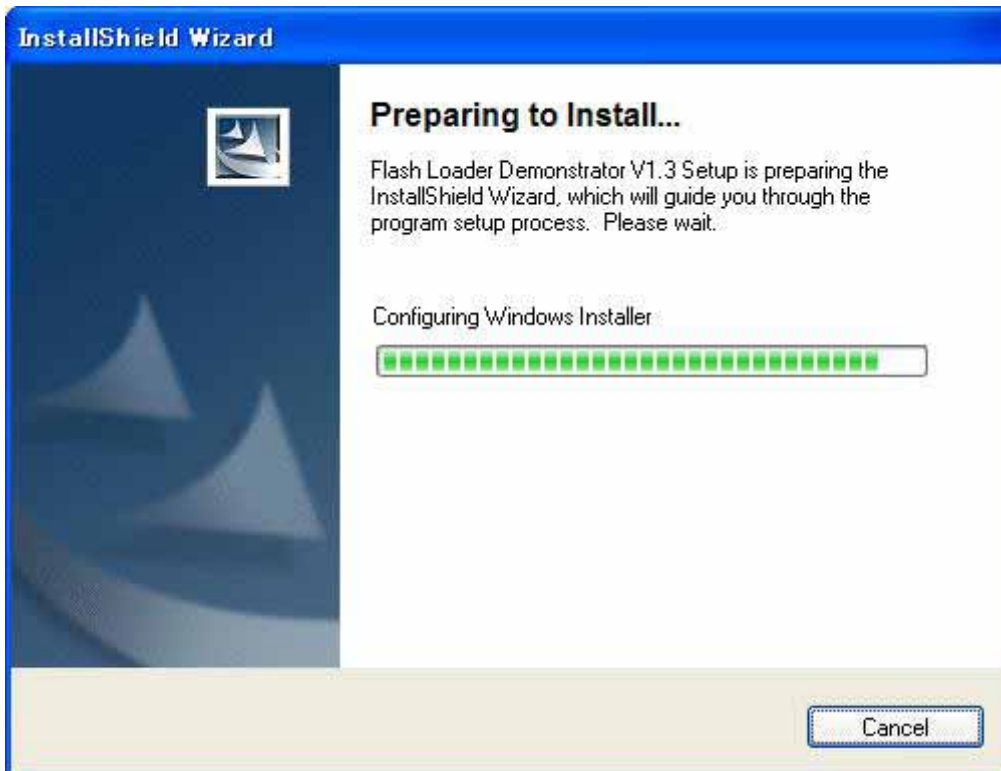
ADの回路図です。PA0, PA1, PA2 ピンの電圧を AD で測って、液晶で表示します。

stm32example.rar は STM32F103 開発キットのサンプルです。ソースコードも含まれます。なかのほかのサンプルを体験してみよう。

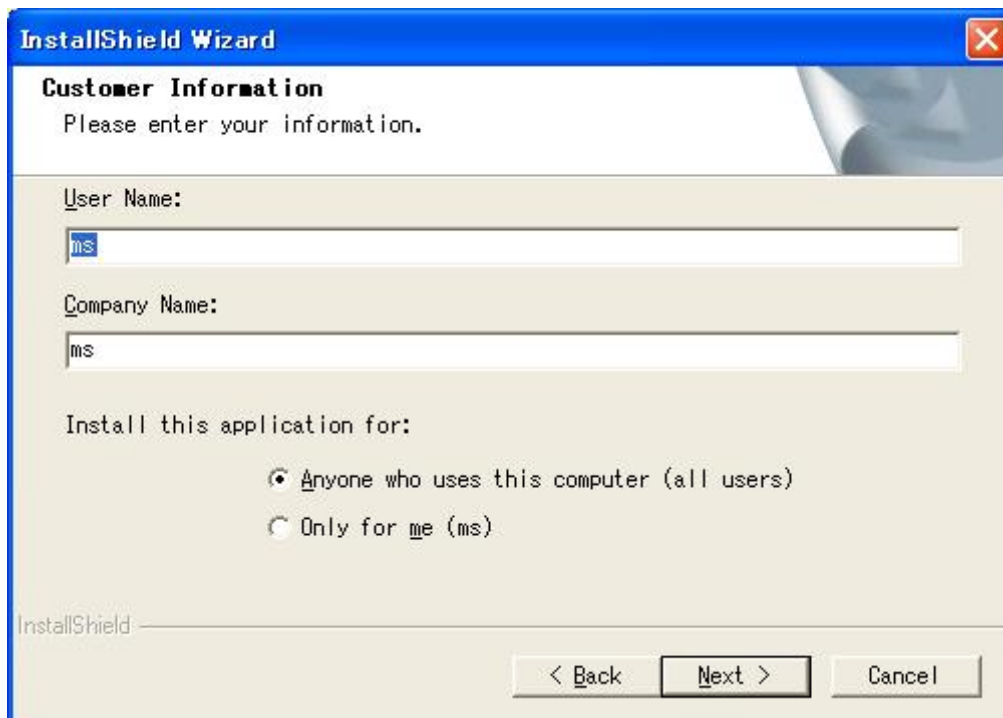
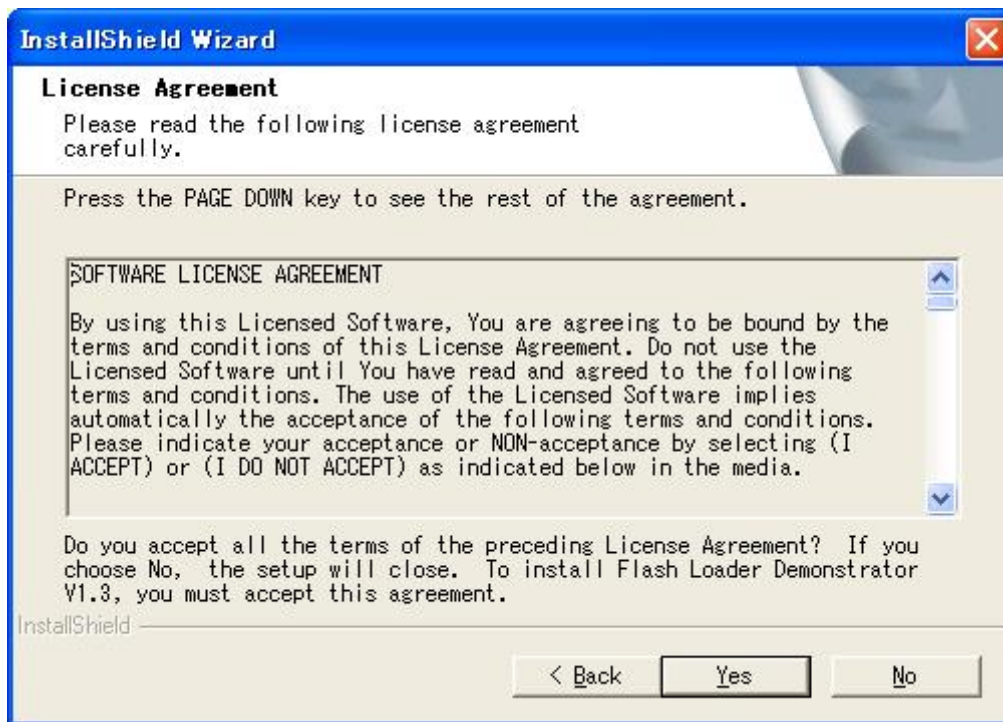
3.2 書き込みツールのインストール

Flash_Loader_Demonstrator_V1.3_Setup.exe はシリアルポートで STM32 マイコンの Flash を更新するツールです。

Flash_Loader_Demonstrator_V1.3_Setup.exe を実行すると、



「Next」ボタンを押すと、英文のライセンスが出てきます。同意できる場合は、「Yes」ボタンを押します。



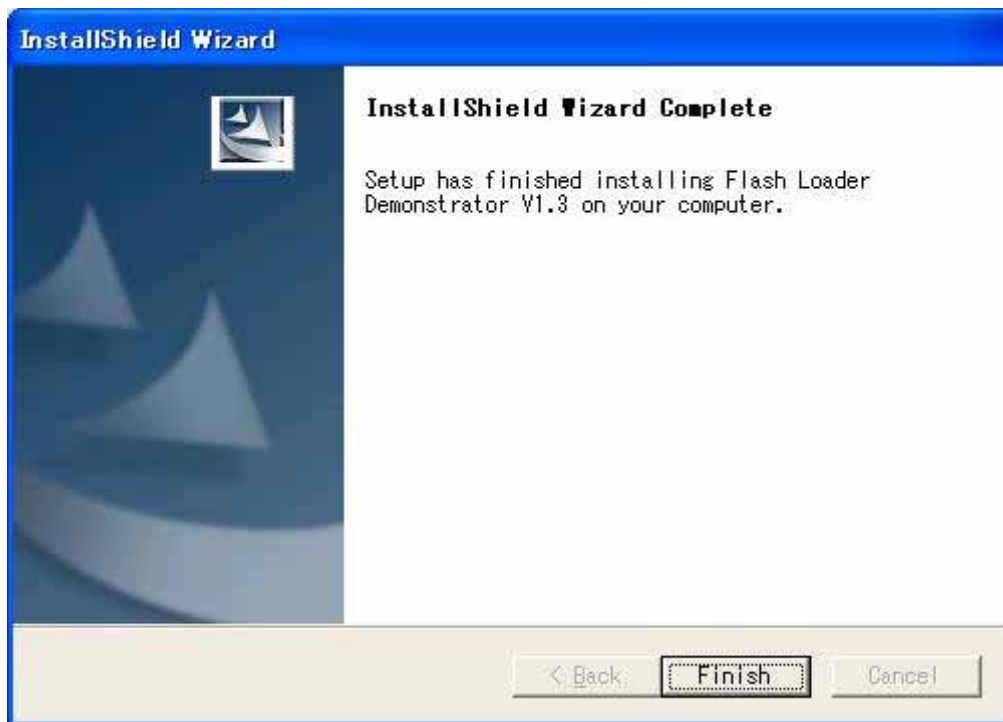
ユーザー名と会社名を入力して、「Next」ボタンを押します。



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。

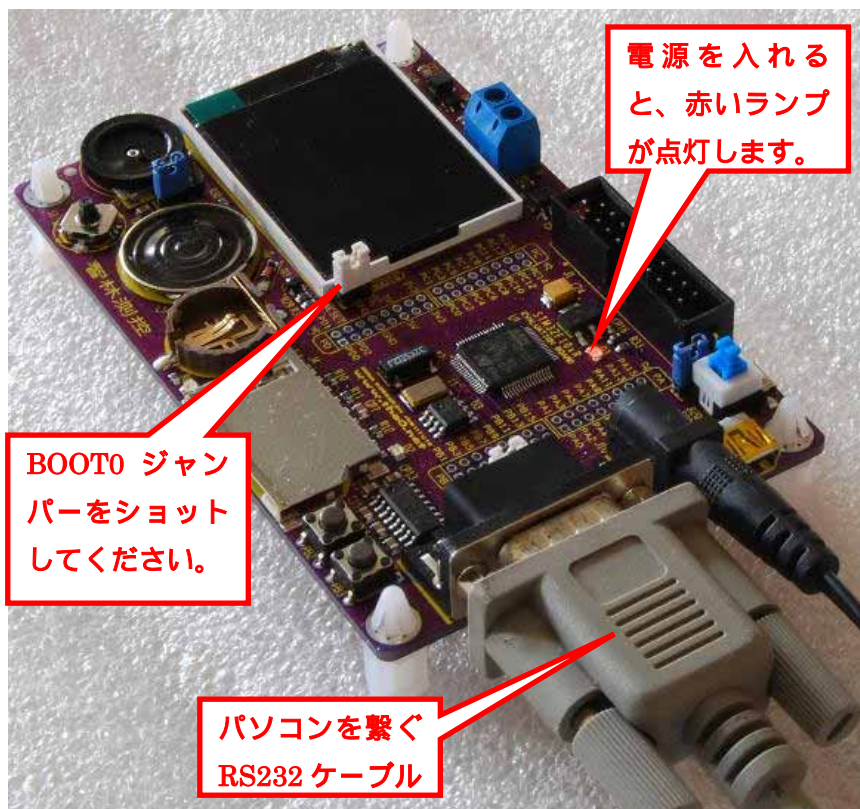


インストール中の画面です。



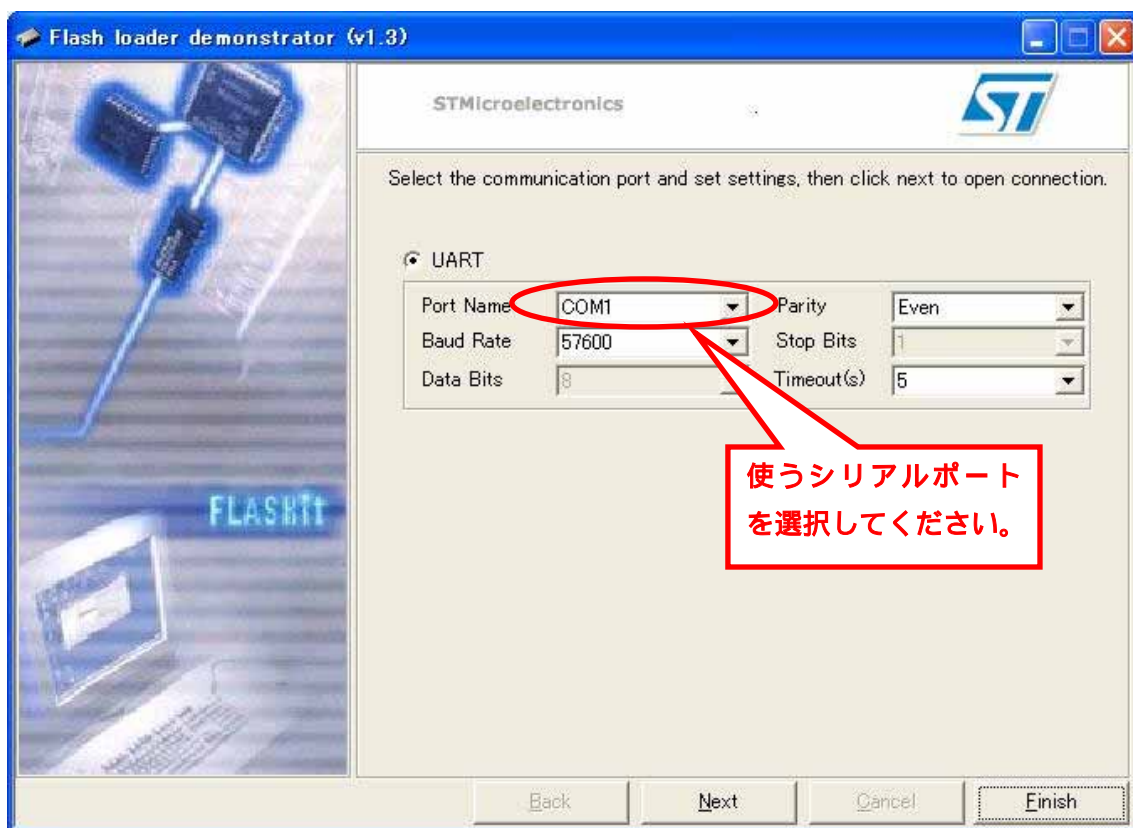
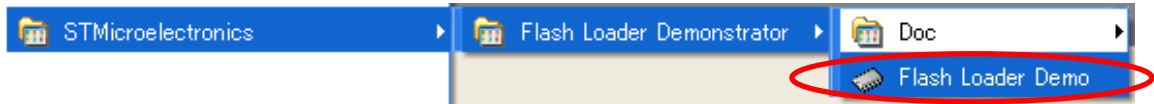
最後に「Finish」をクリックすると、ウィザードが閉じてインストールが終了します。

3.3 書き込み

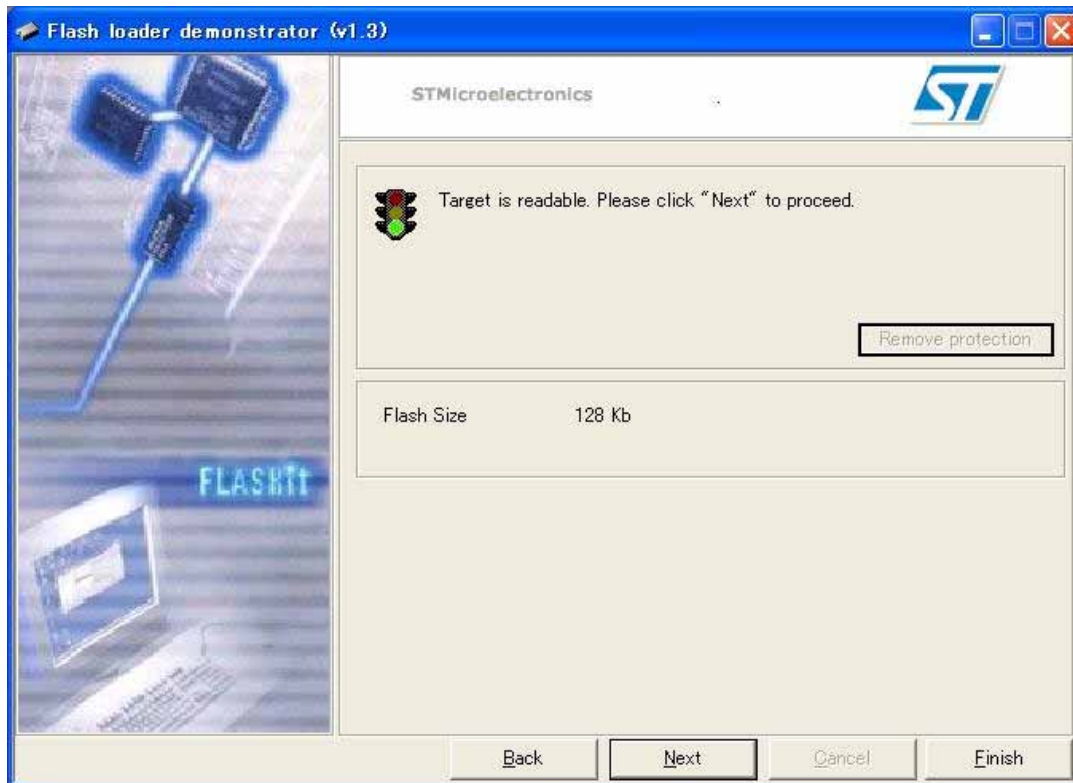


左の写真はSTM32F103 開発キットの書き込み状態の設定です。

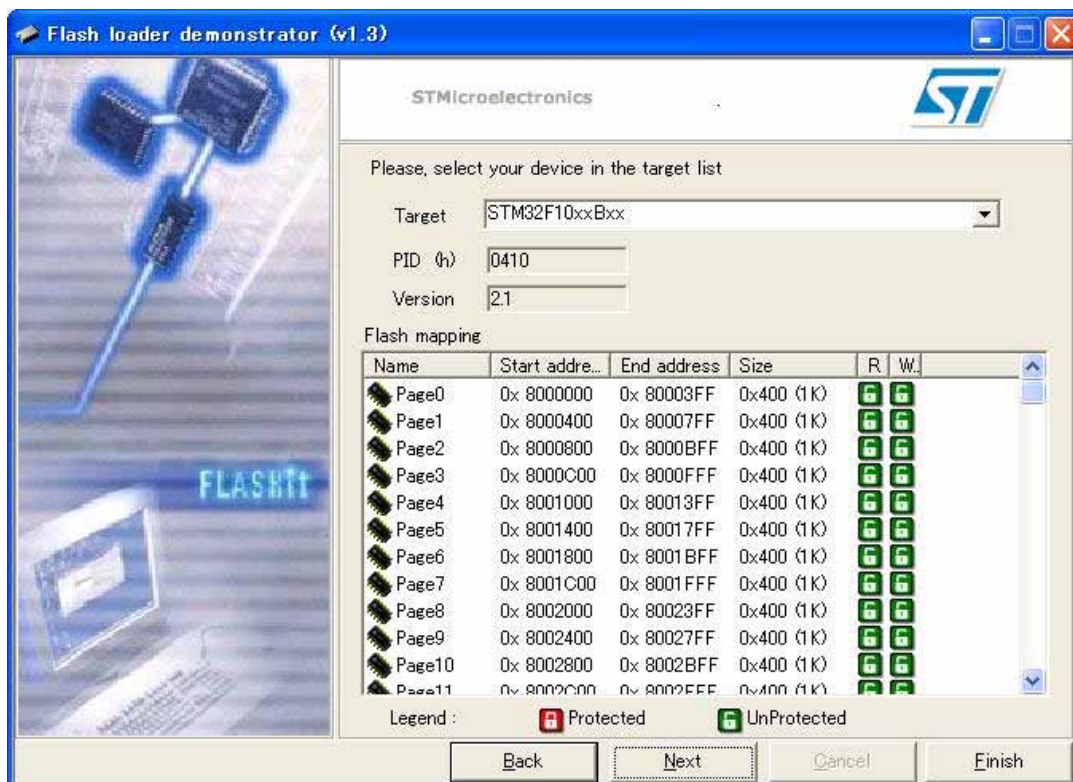
Windows のメニュー「スタート」→「STMicroelectronics」→「Flash Loader Demonstrator」
→「Flash Loader Demo」を選択してください。



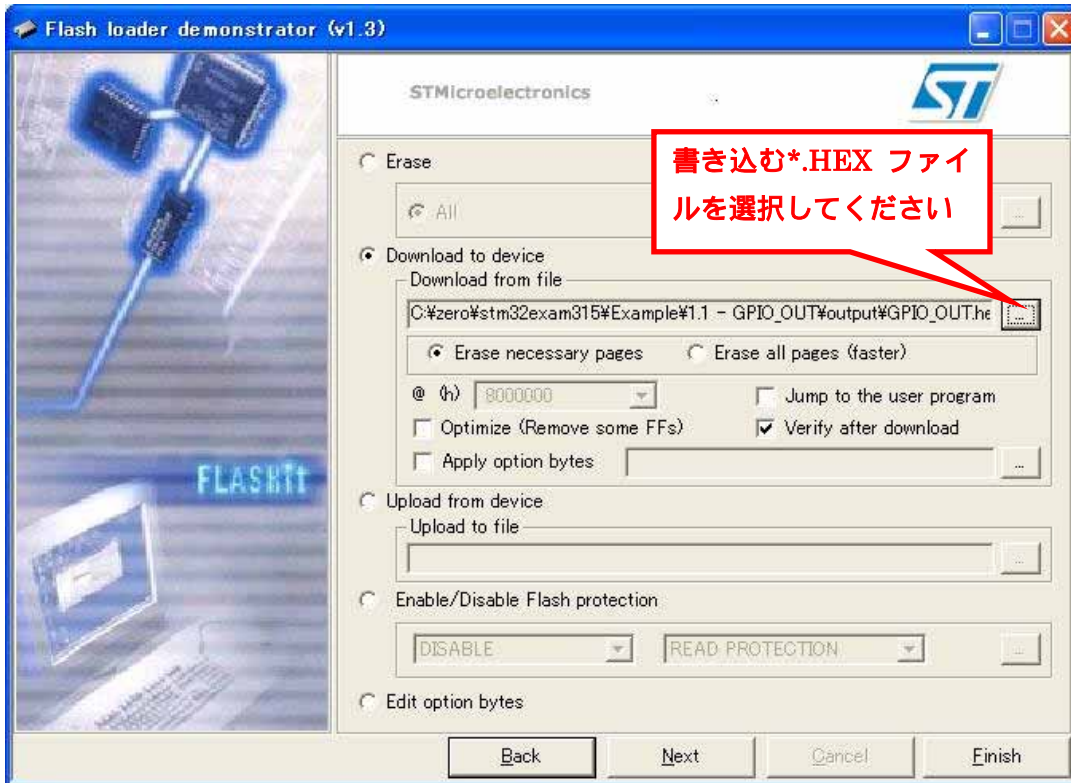
書き込み用のシリアルポートを選択して、「Next」ボタンを押します。



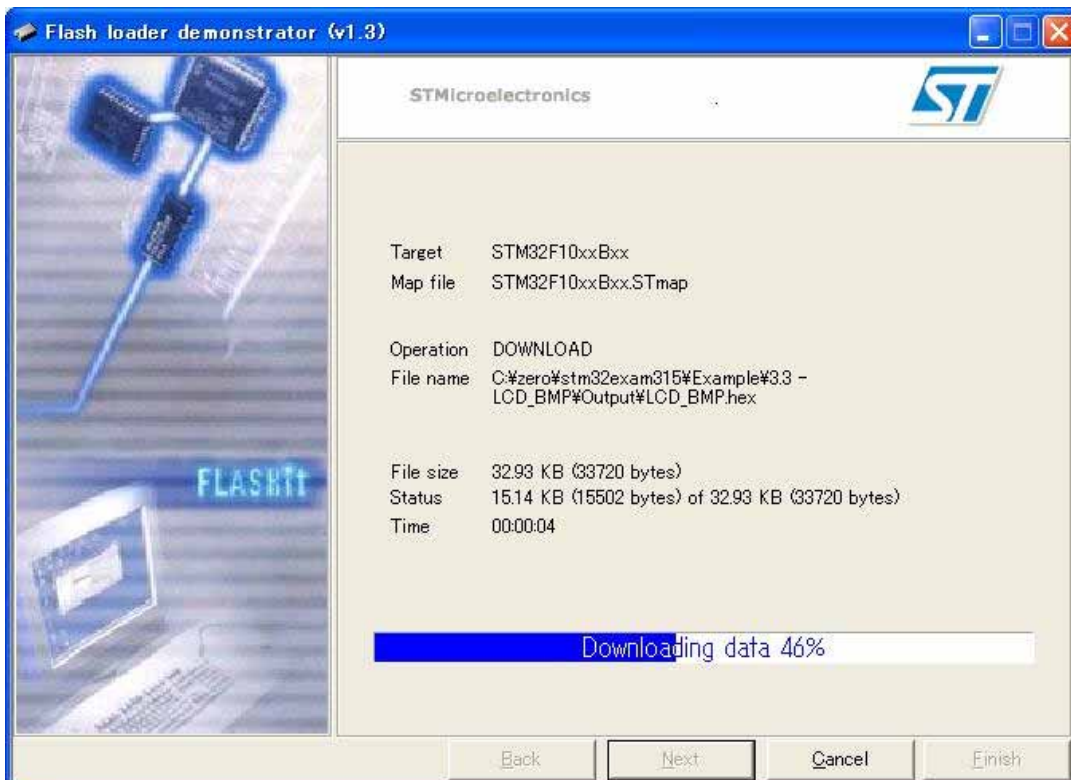
この画面があらわすと、パソコンは開発キットを繋ぎました。「Next」ボタンを押します。



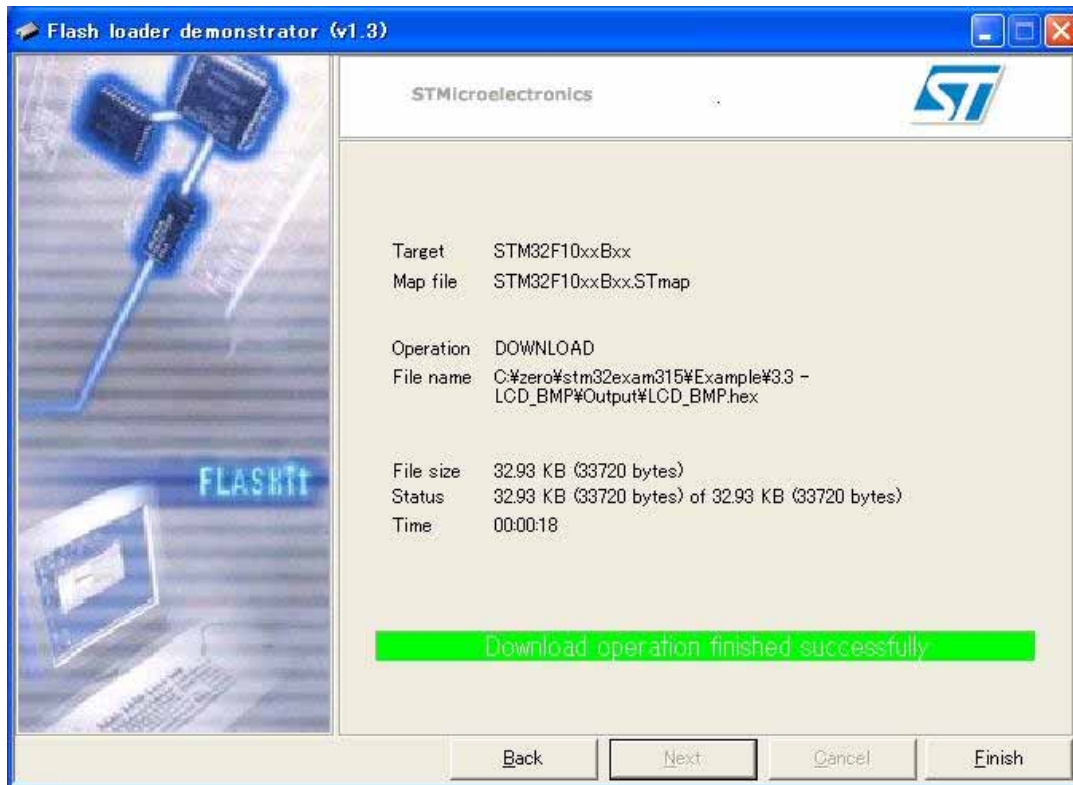
そのまま「Next」ボタンを押します。



書き込む*.HEX ファイルを選択して、「Next」ボタンを押します。



書き込み中です。

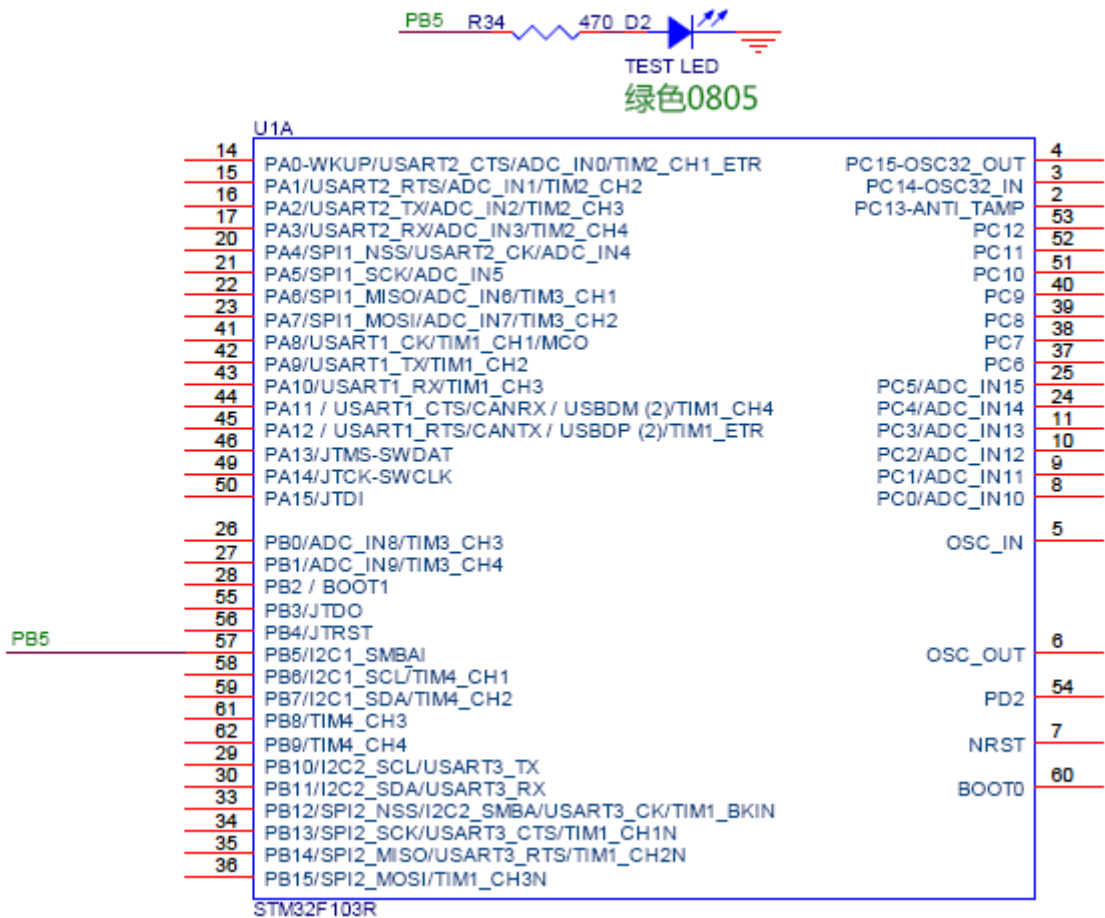


最後に「Finish」をクリックすると、ウィザードが閉じて書き込みが終了します。

3.4 LED 点灯

サンプル : Example/1.1 - GPIO_OUT/output/GPIO_OUT.hex

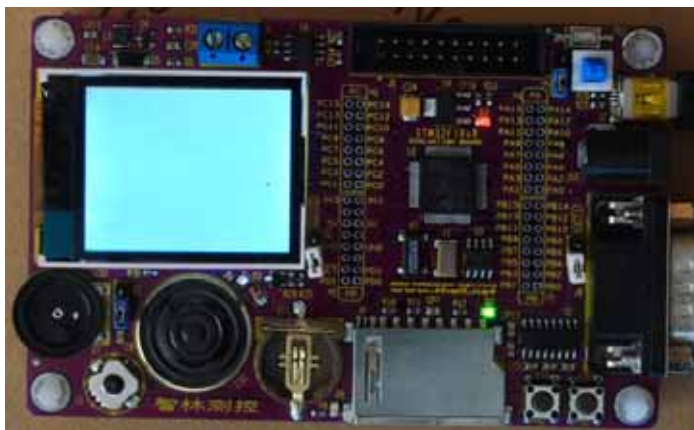




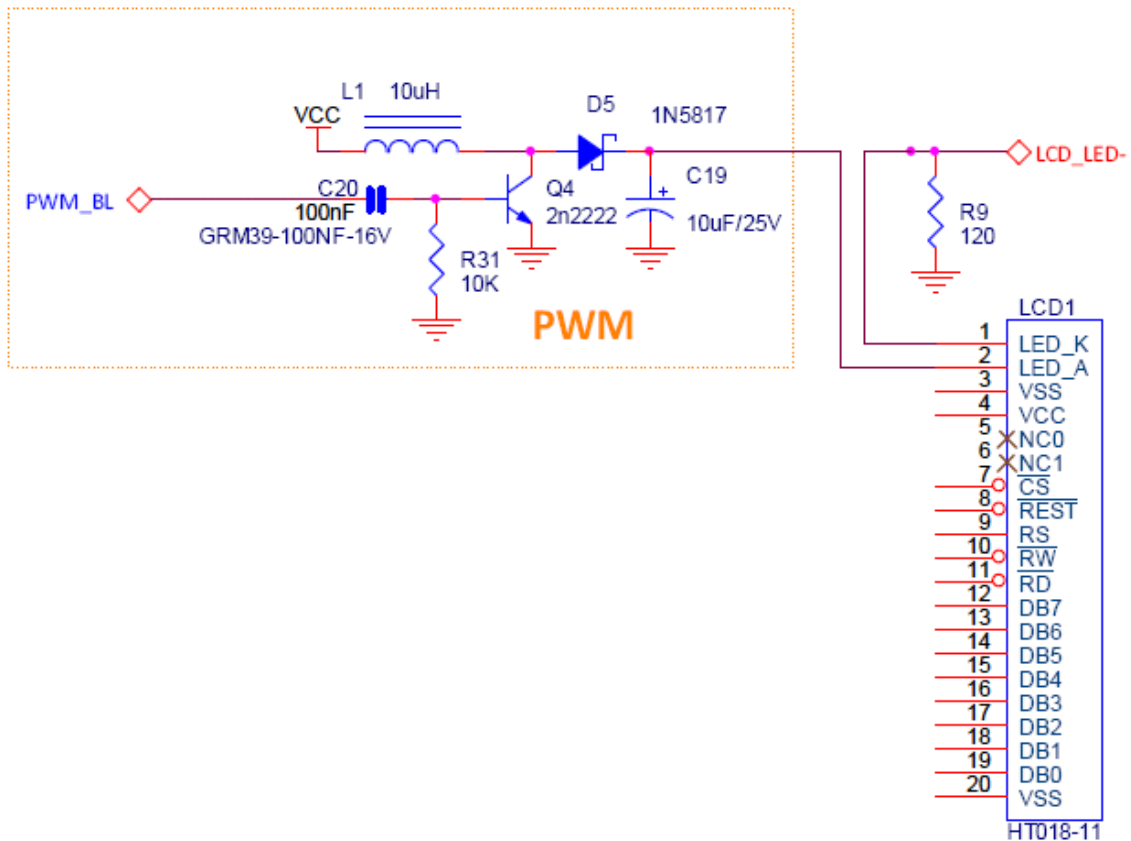
LED 点灯の回路です。STM32F103 の PB5 は LED を繋ぎます。

3.5 PWM

サンプル : Example/2.1 - PWM/output/PWM.hex



PWMで液晶のバックライトを点灯します。



PWMで液晶のバックライトを点灯する回路です。液晶のバックライトの電圧は6-7Vが必要です。PWMで5V電源から昇圧します。

3.6 液晶 LCD

サンプル : Example/3.1 - LCD/output/LCD.hex



液晶 LCD
で英語を
表示します。

3.7 漢字のディスプレイ

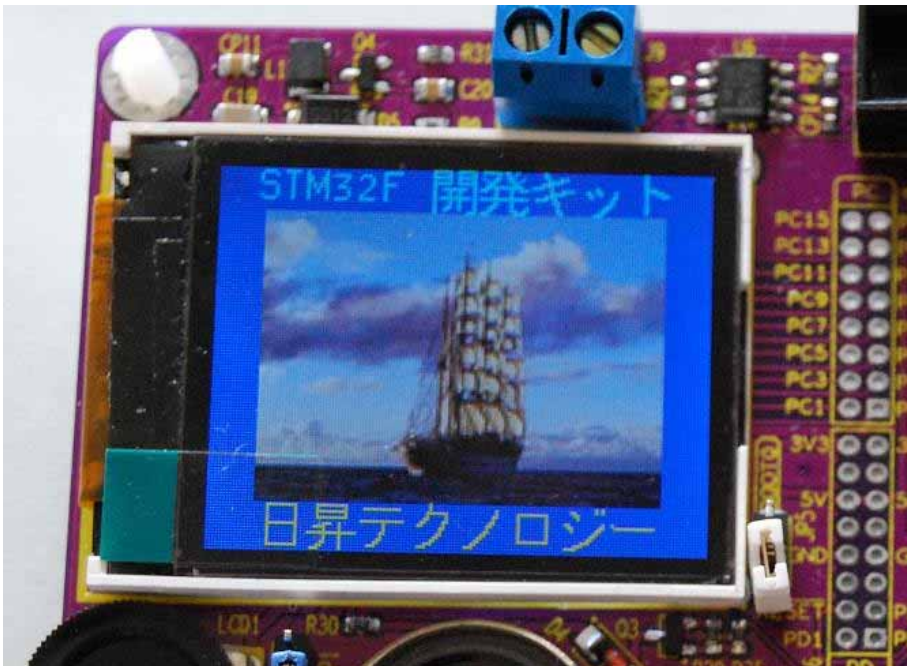
サンプル : Example/3.2 - LCD_CN/output/LCD.hex



液晶 LCD で漢字を表示します。

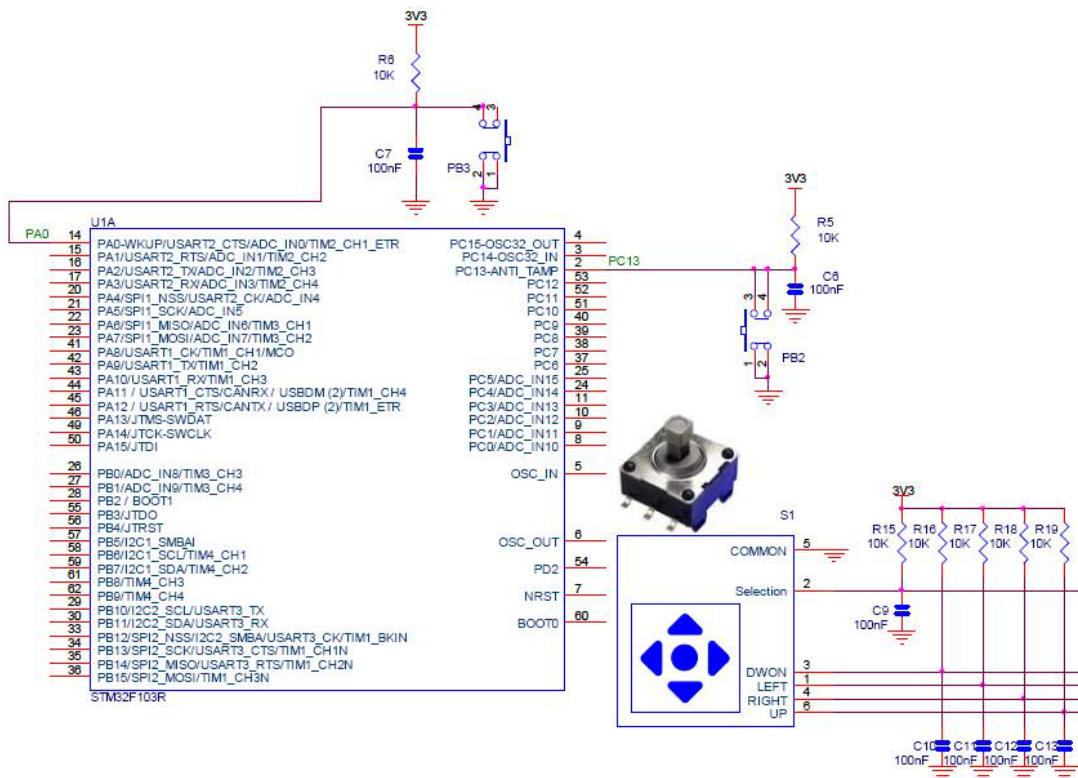
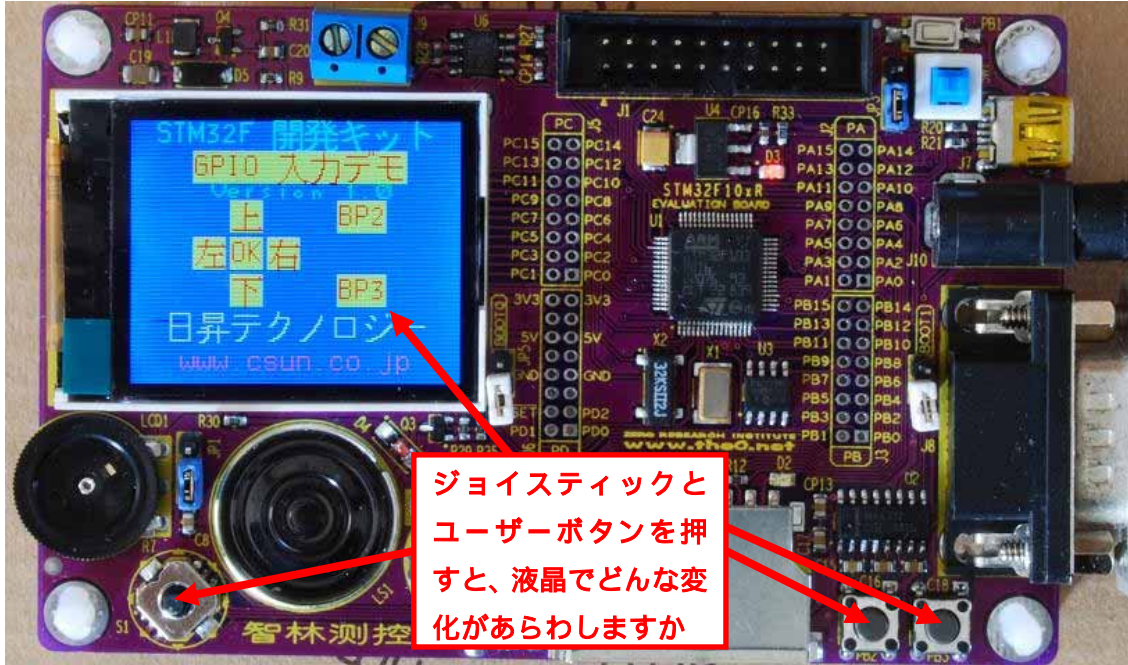
3.8 ピクチャのディスプレイ

サンプル : Example/3.3 - LCD_BMP/Output/LCD_BMP.hex



3.9 GPIO の入力実験

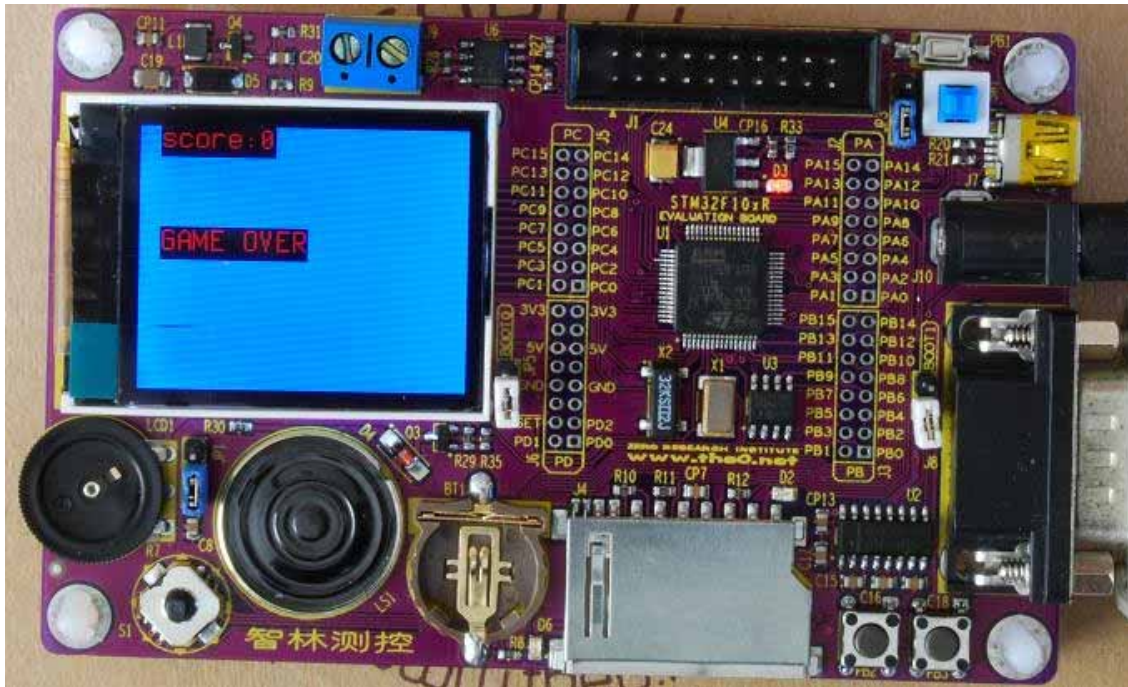
サンプル : Example/4.1 - GPIO_IN/Output/GPIO_IN.hex



GPIO 入力の回路です。

3.10 ゲーム

サンプル：Example/4.2 - GameSnake/Output/GameSnake.hex



3.7 節に基づくゲームです、やってみましょう。

3.11 ADC 入力実験

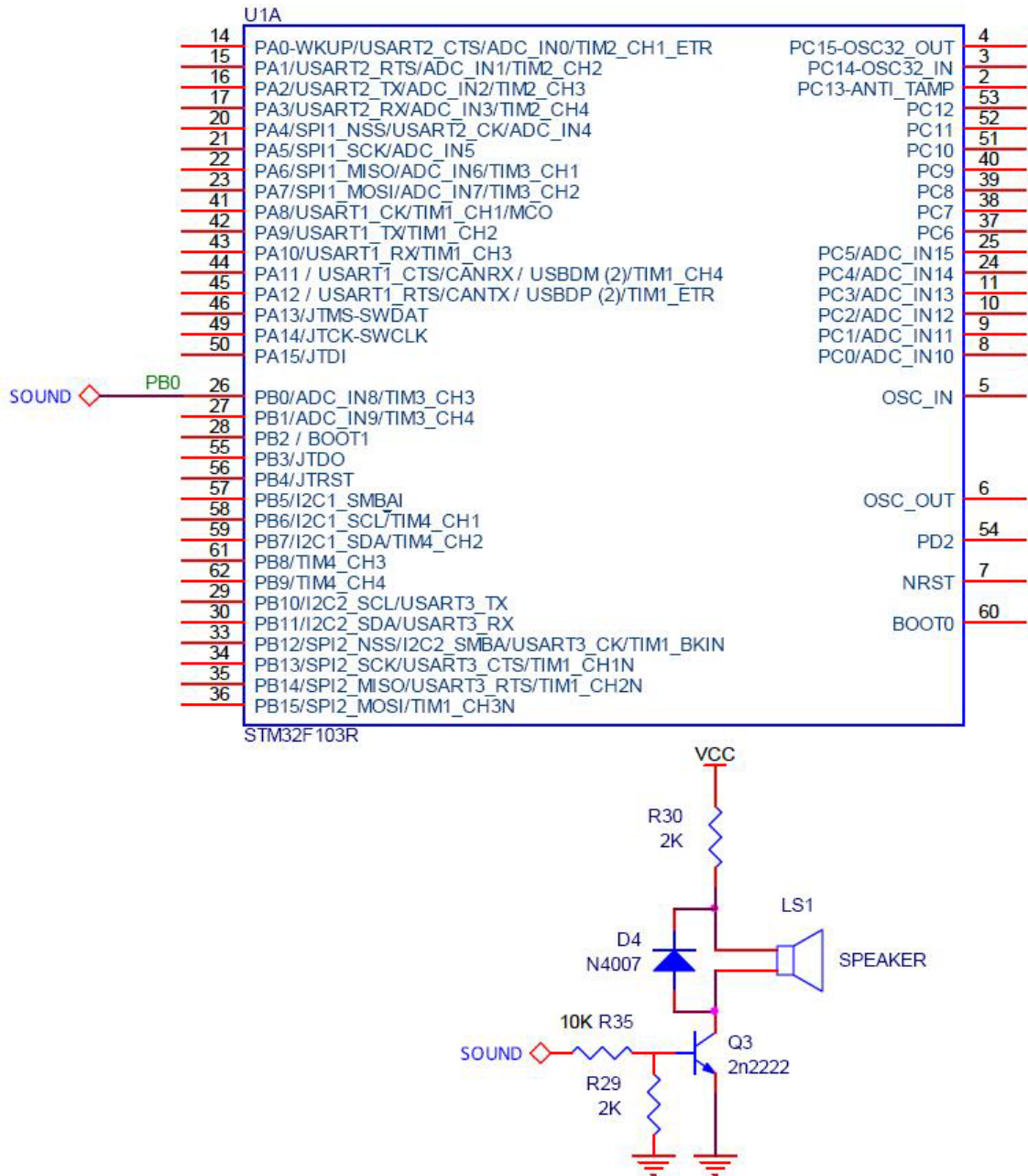
サンプル：Example/5.1 - ADC/Output/ADC.hex

これはデフォルトのサンプルです。



3.12 PWM 音声

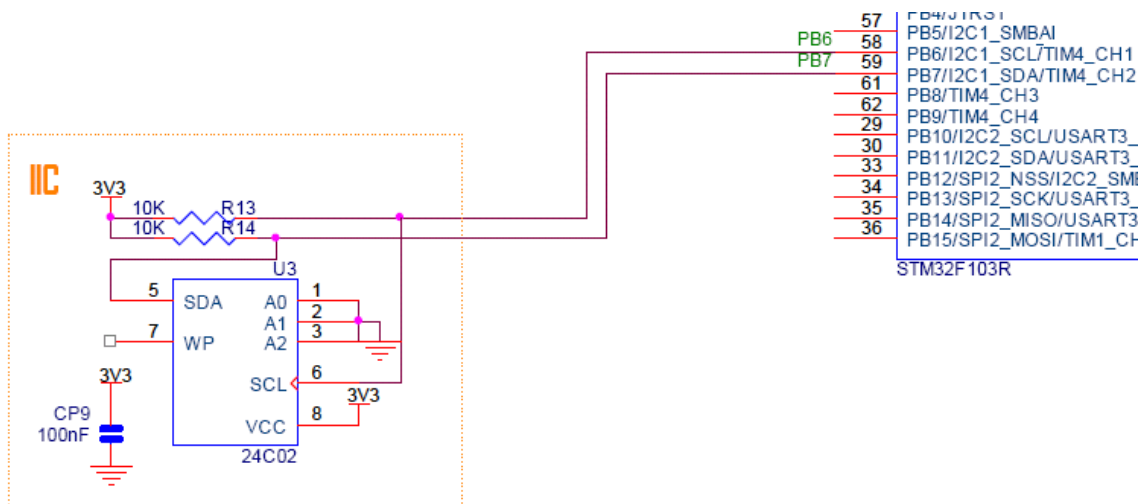
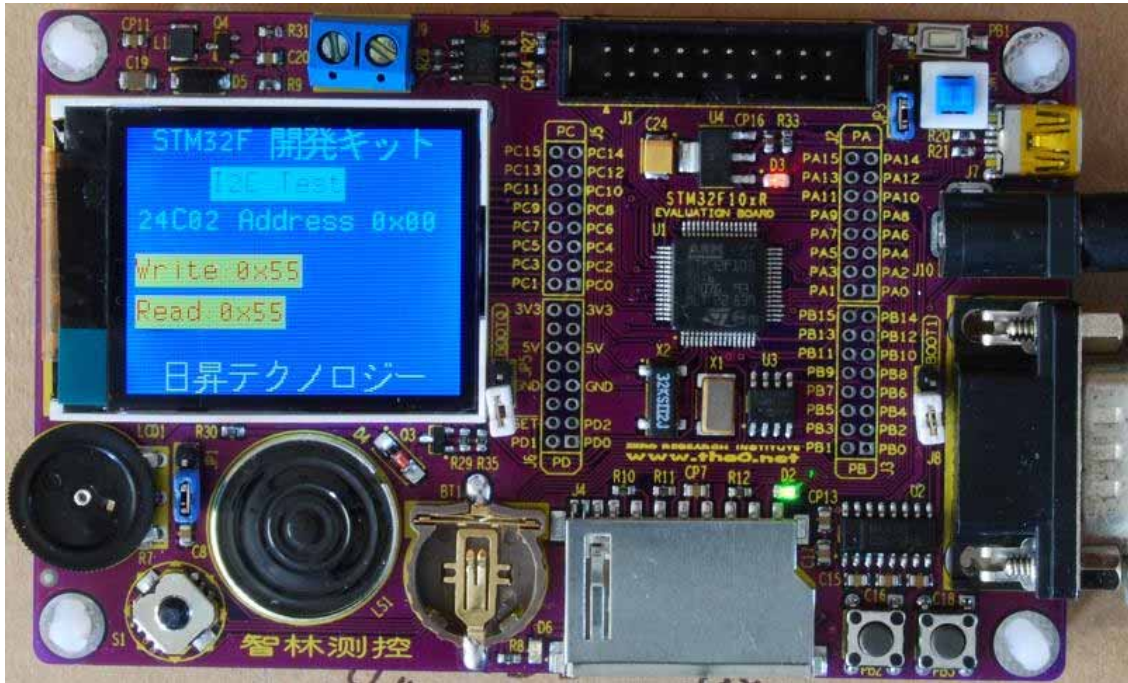
サンプル : Example/6.1 - PWM_Sound/Output/PWM_Sound.hex



スピーカーは STM32F103 のタイマー3 の 3ch を繋ぎます。ボリュームを回すと、音声の周波数が変化します。

3.13 I2C EEPROM

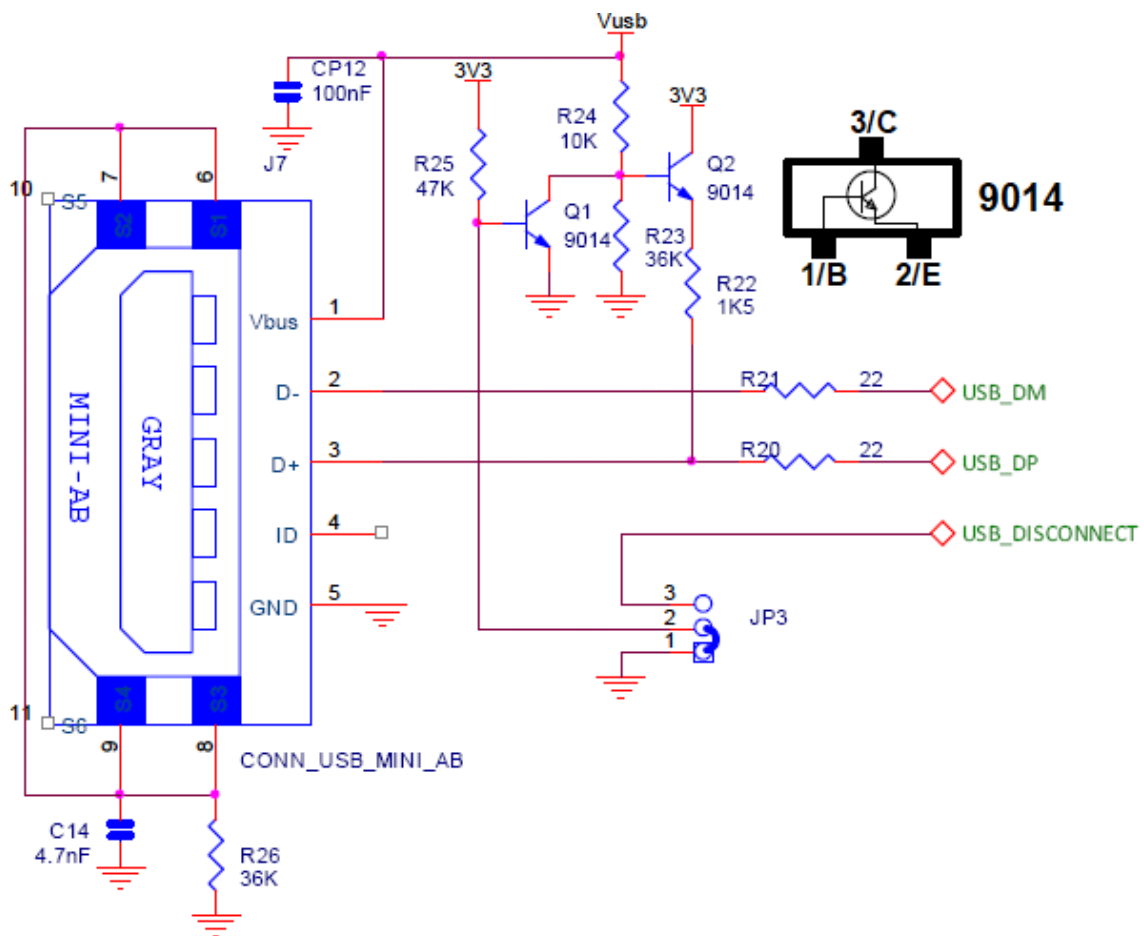
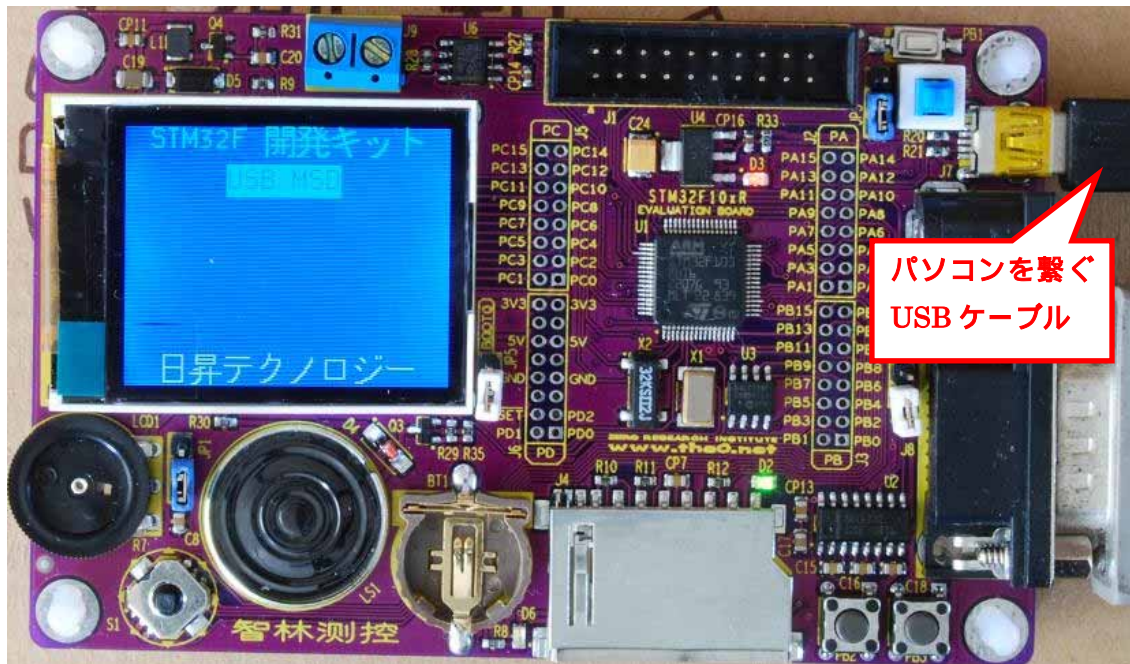
サンプル : Example/7.1 - I2C/Output/I2C.hex



STM32F103 開発キットの上に 2kbit の I2C EEPROM 24C02 があります。

3.14 USB メモリ

サンプル : Example/8.1 - USBMem/Output/USBmem.hex

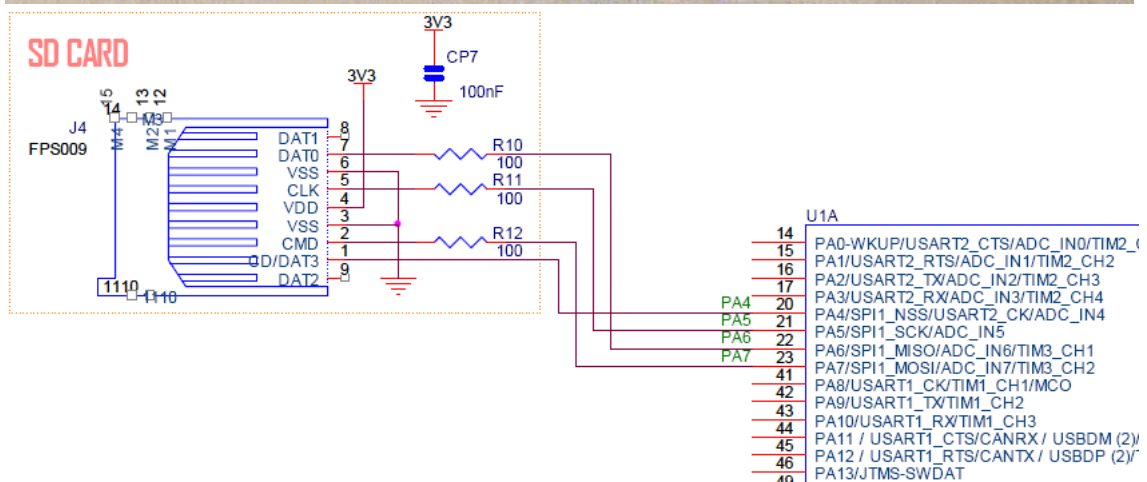
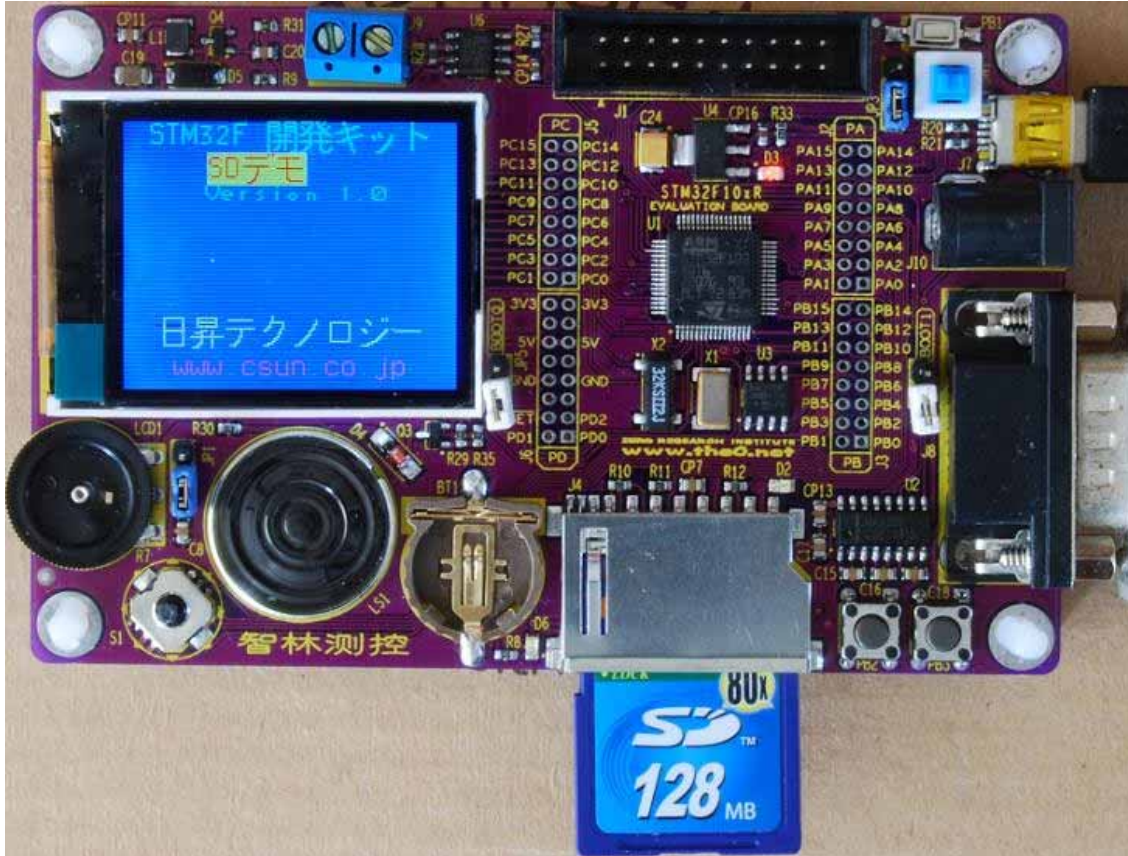


STM32F103 は USB2.0 デバイスインターフェースがあります。このサンプルを STM32F103 開発キットに書き込んで、USB ケーブルでパソコンを繋ぐと、開発キットは

USB メモリとして認識されます。

3.15 SD カード

サンプル : Example/9.1 - SDcard/Output/SDcard.hex

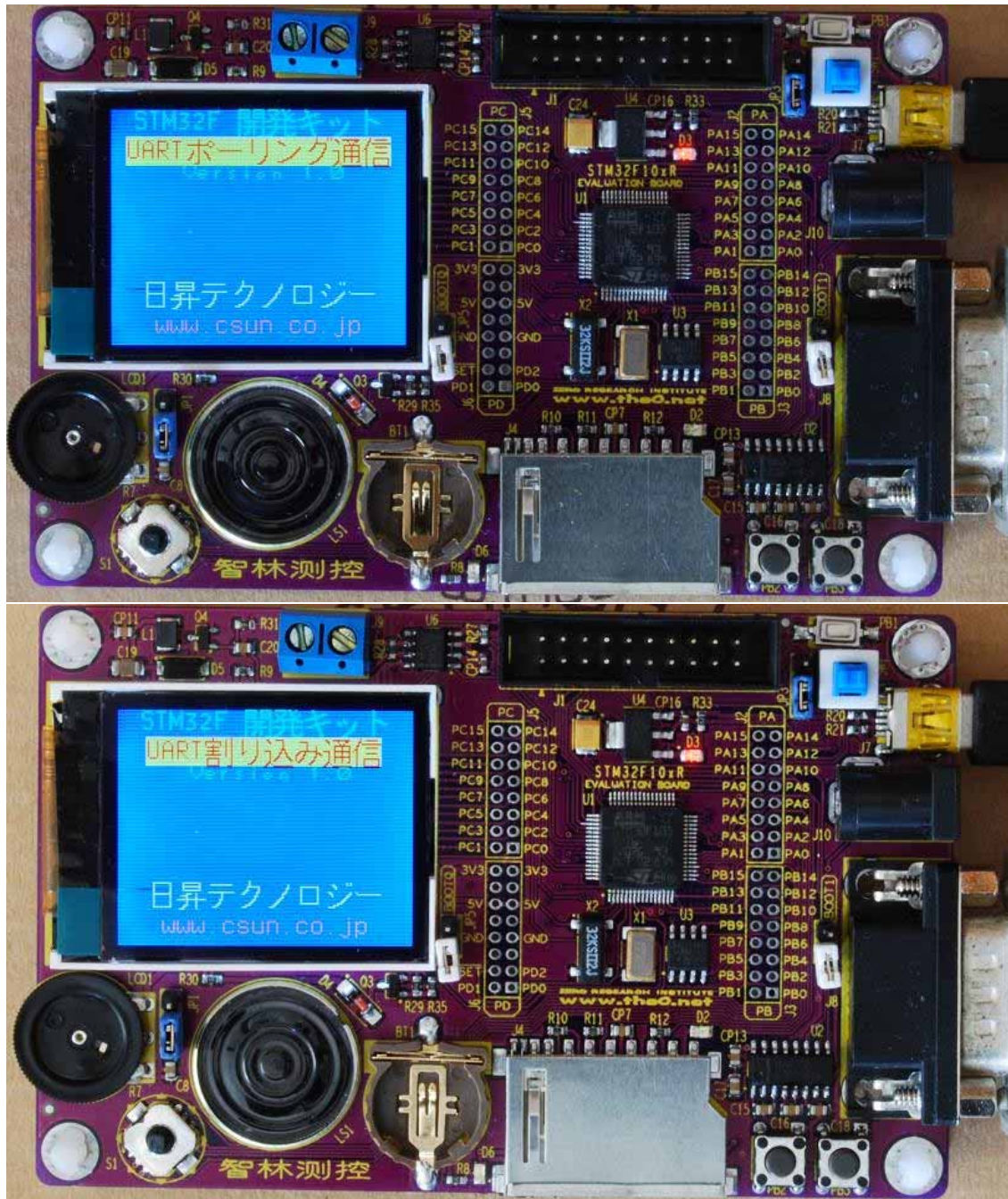


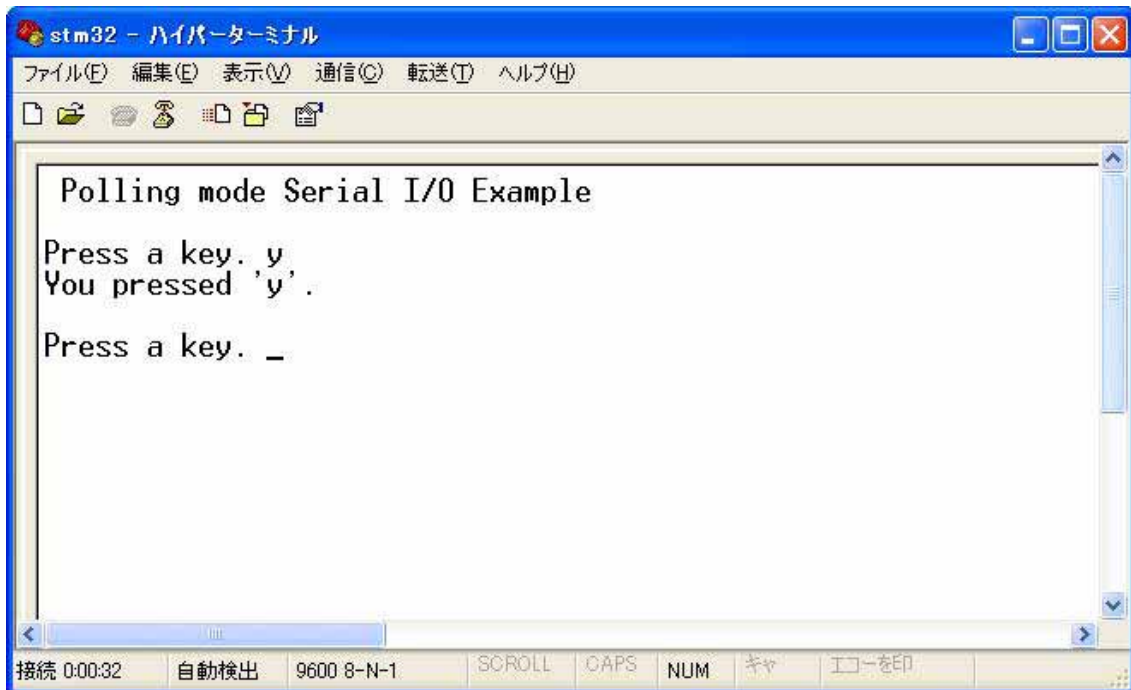
このサンプルはFatFS(http://elm-chan.org/fsw/ff/00index_j.html)を利用して、SDカードへアクセスします。SDカードを挿入すると、自動的にSDカードでtest.txtというファイルを生成します。

3.16 シリアル通信実験

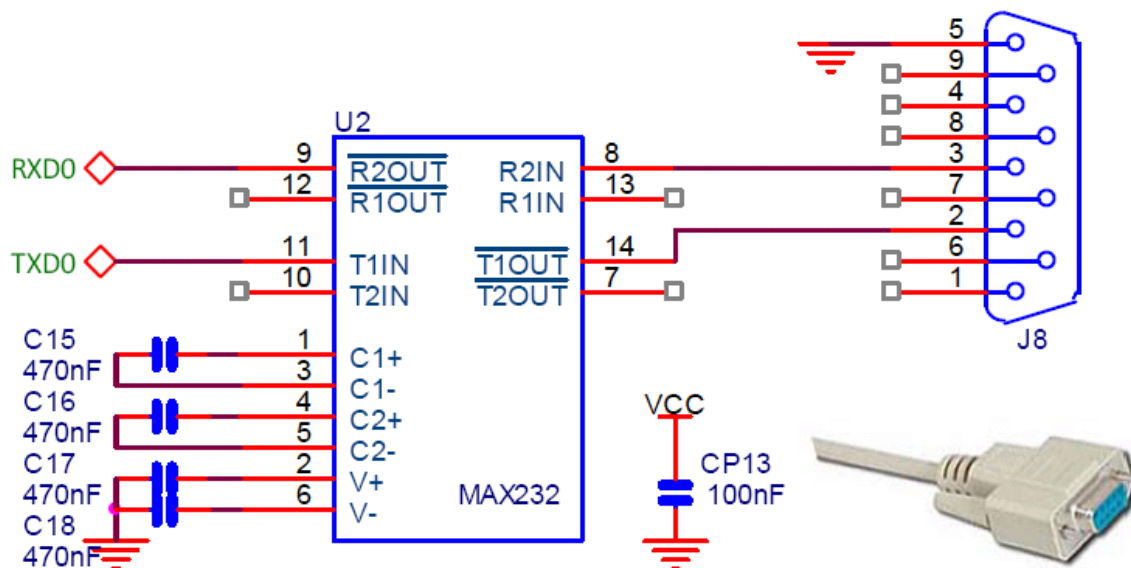
サンプル : Example/10.1 - UART/Output/USART_Pol.hex(ポーリング)

サンプル : Example/10.1 - UART/Output/USART_Pol.hex(割り込み)



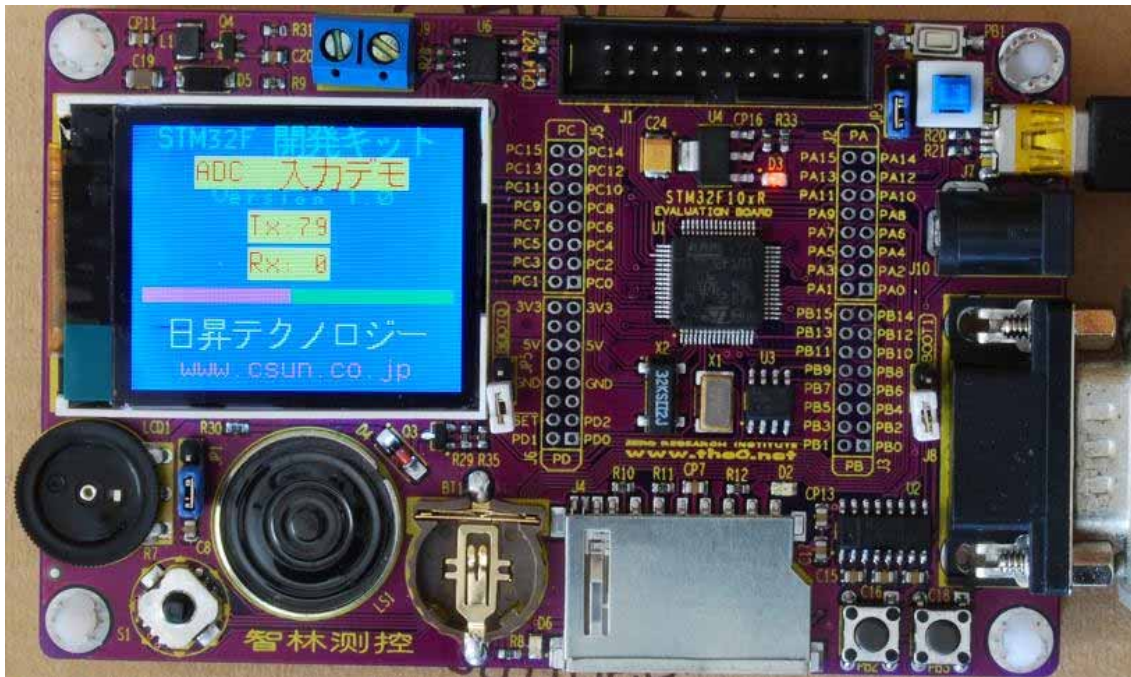


パソコンのハイパーターミナルの画面です。ハイパーターミナルの設定はボーレート 9600 ビット/秒、フロー制御なしです。

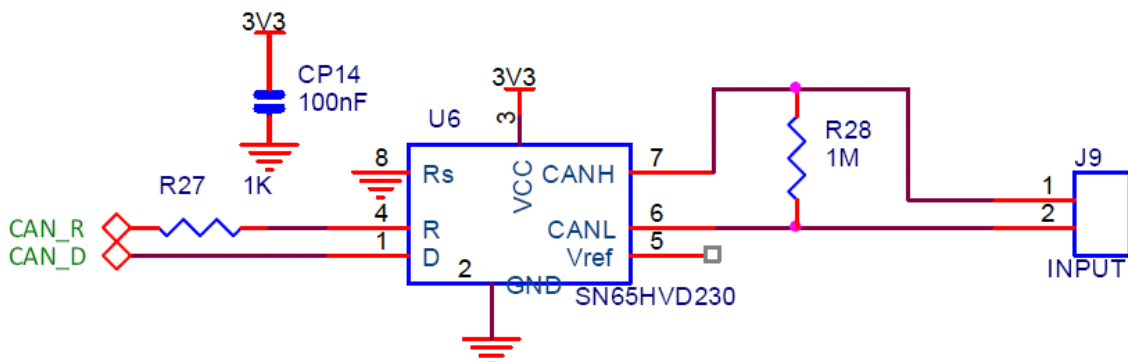


3.17 CAN シリアル通信実験

サンプル : Example/11.1 - CAN/Output/CAN.hex

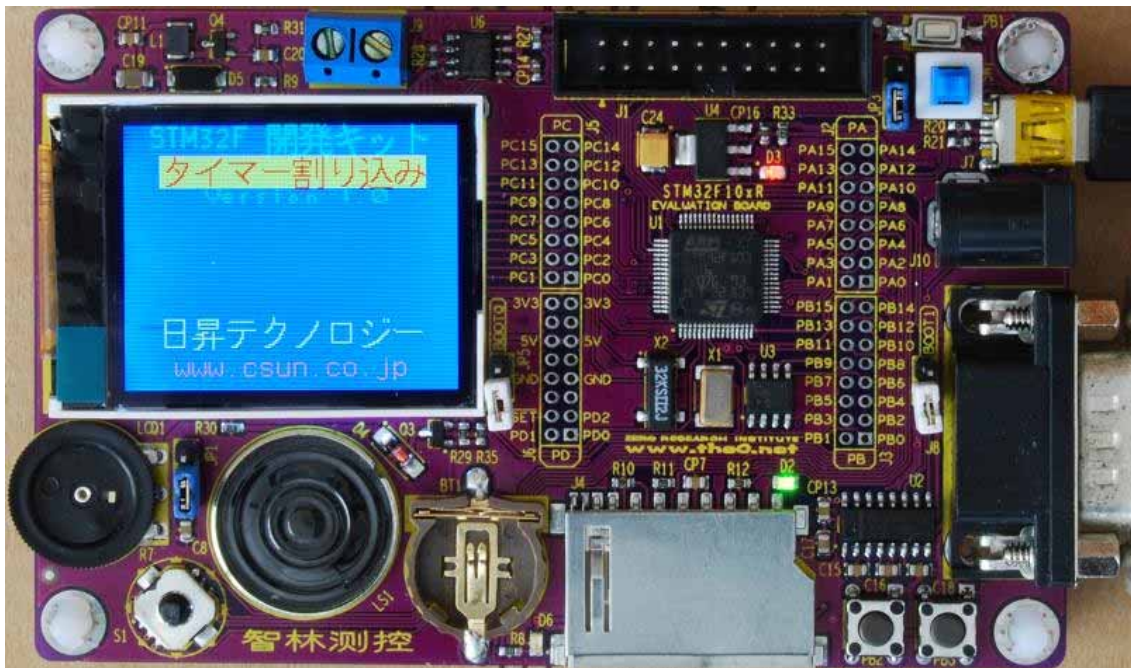


CAN 通信実験は二つの STM32F103 開発キットが必要です。STM32F103 開発キットはボリュームを測って、情報を CAN で送信します。同時に、ほかの STM32F103 開発キットからのボリューム情報を受信して、液晶で表示します。



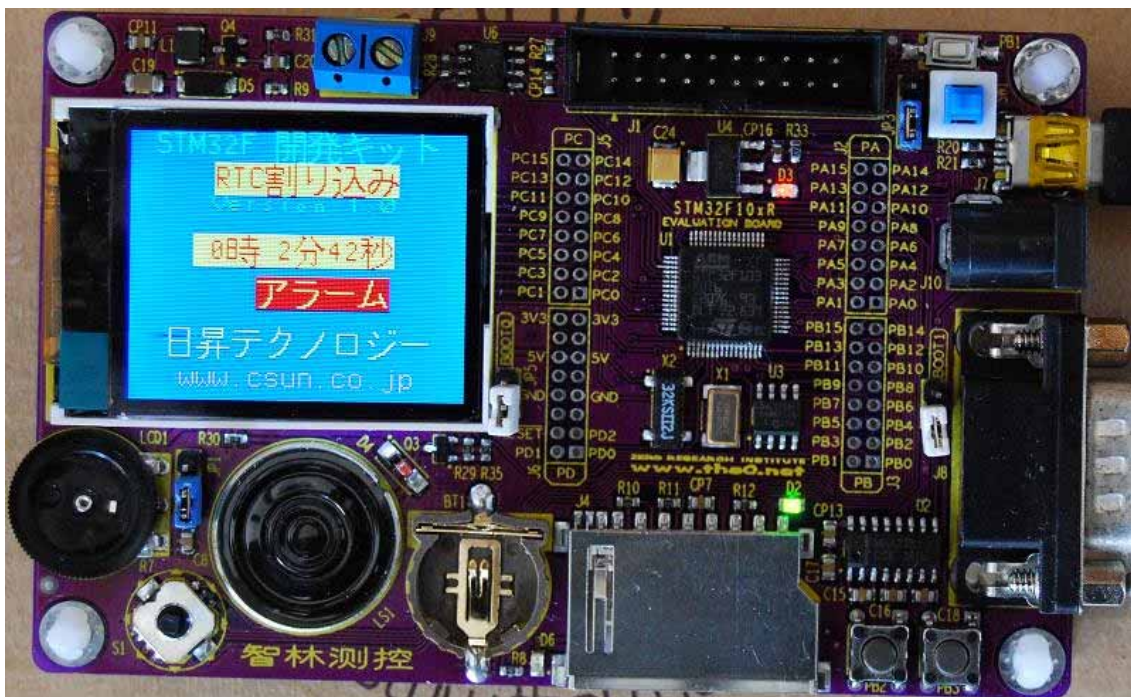
3.18 タイマーの割り込み実験

サンプル : Example/12.1 - Timer/Output/Timer.hex



3.19 RTC

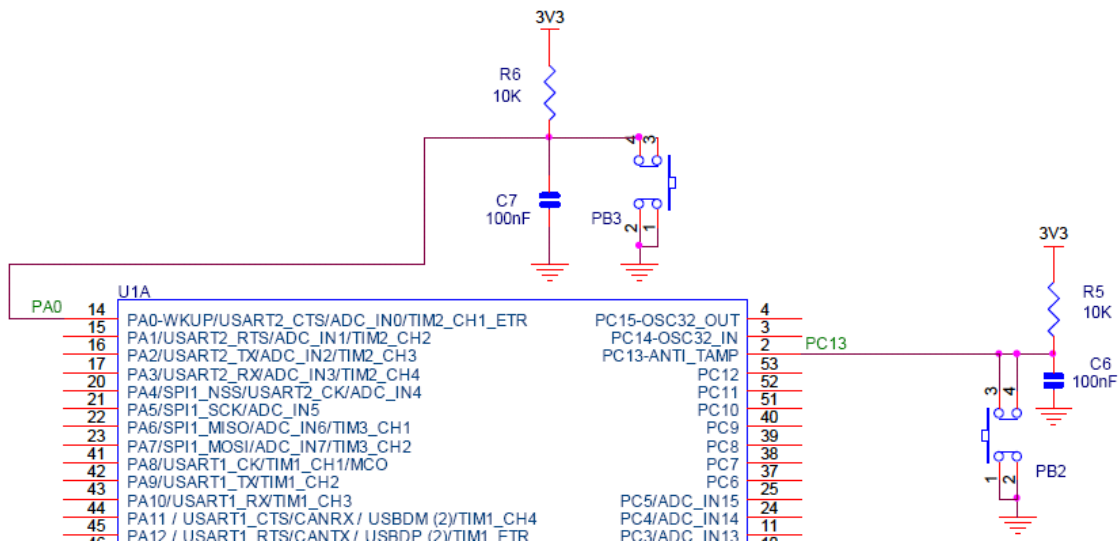
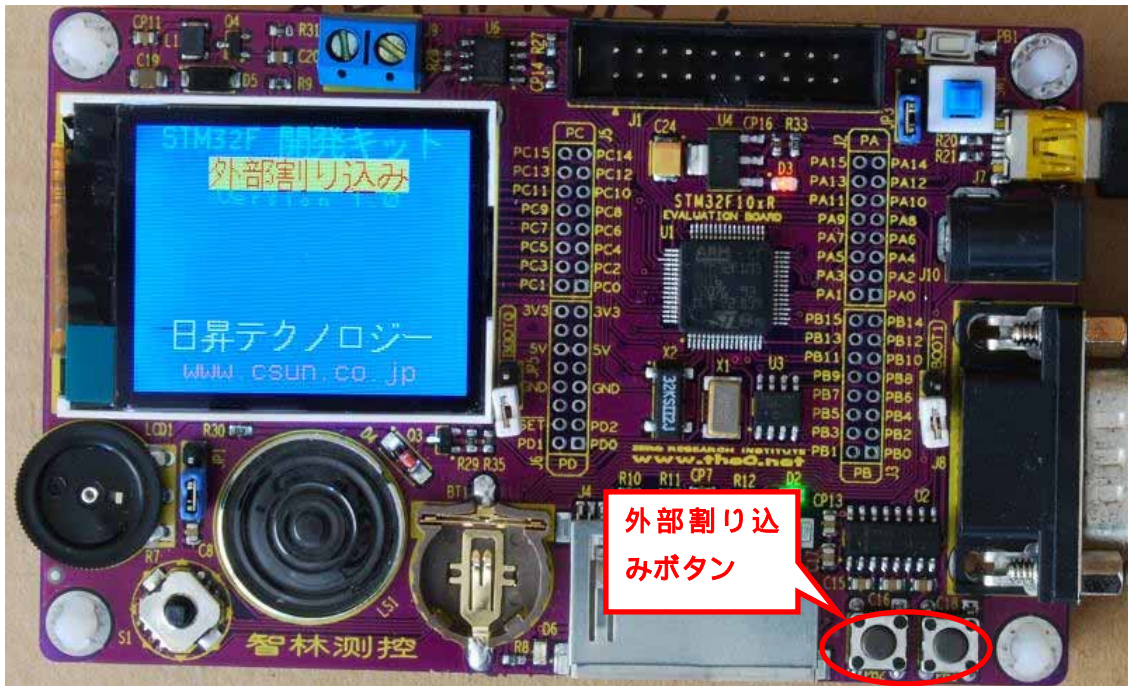
サンプル : Example/13.1 - RTC/Output/RTC.hex



STM32F103 は RTC が内蔵しています。

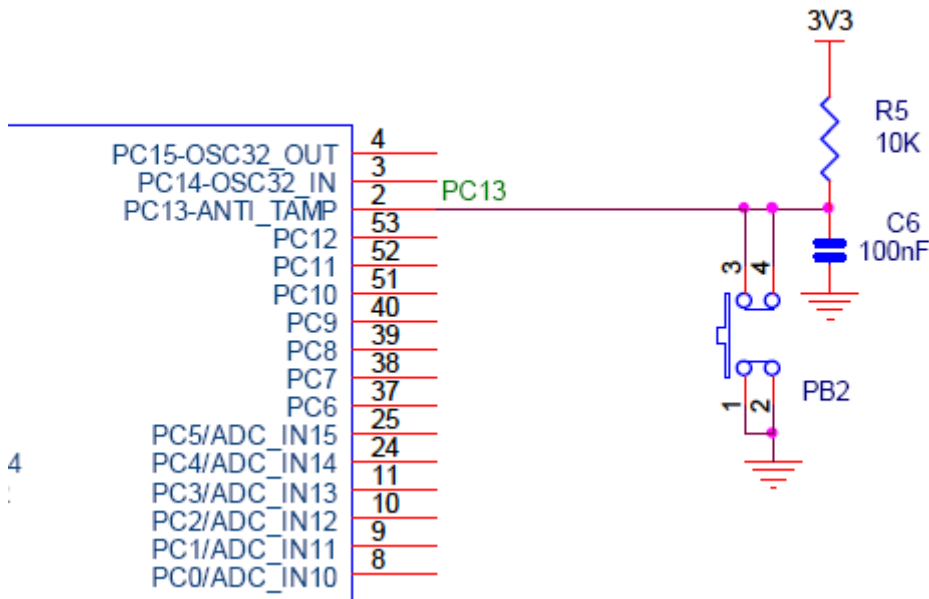
3.20 外部割り込み実験

サンプル : Example/14.1 - Exti/Output/Exti.hex



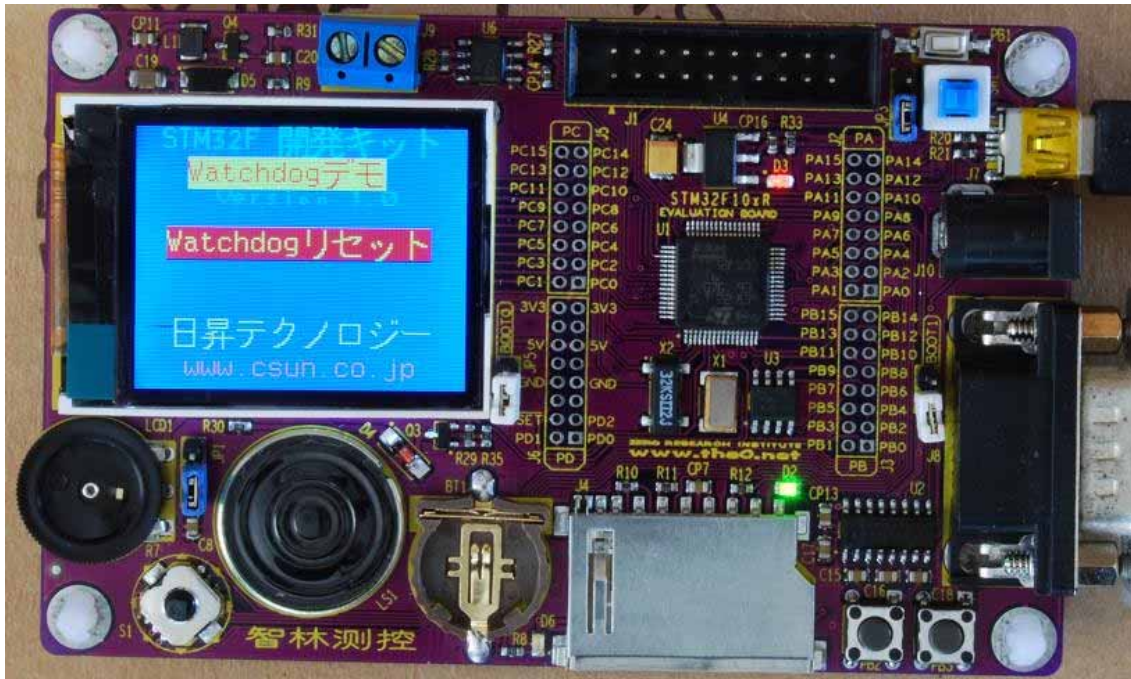
3.21 Tamper 実験

サンプル : Example/15.1 - Tamper/Output/Tamper.hex



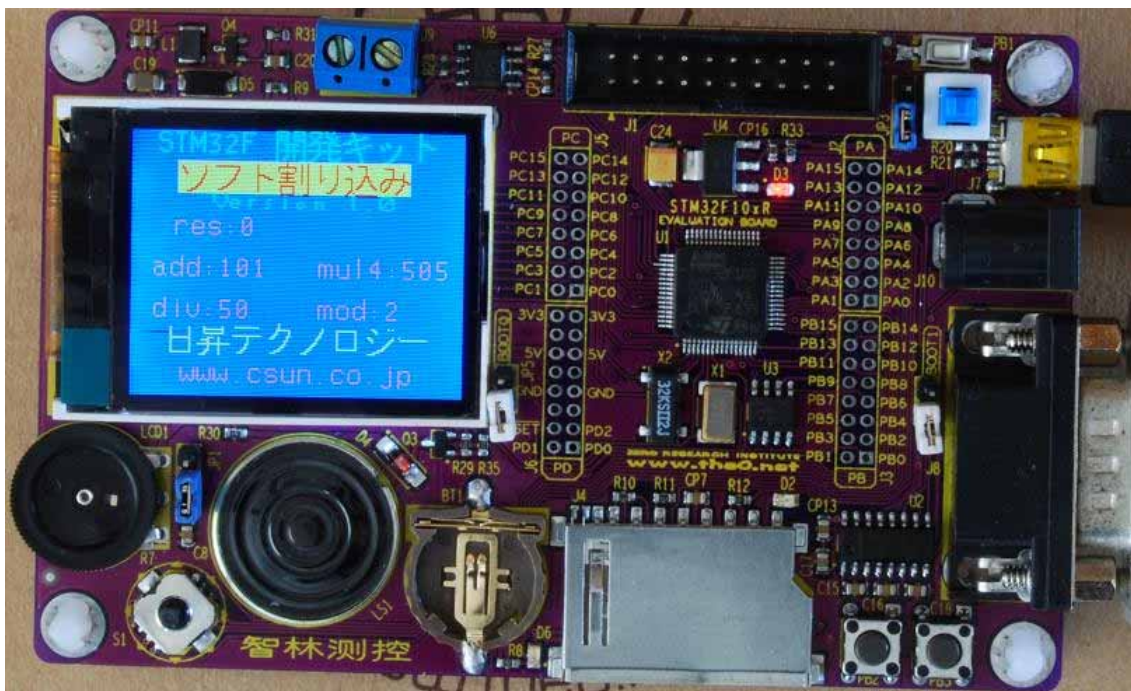
3.22 ウォッチドッグ・タイマ

サンプル : Example/16.1 - IWDG/Output/IWDG.hex



3.23 ソフトウェア割り込み

サンプル : Example/17.1 – SVC/Output/SVC.hex



3.24 uC/OS II デモ

サンプル : Example/18.1 - uCOS/APP/Output/uCOSii.hex



uC/OS II は小さいリアルタイム OS です。uC/OS II はフリーソフトではありません。これはデモです。二つのタスクを生成しました。一つは LED を点滅させます。もう一つは液晶でカウンターを表示します。

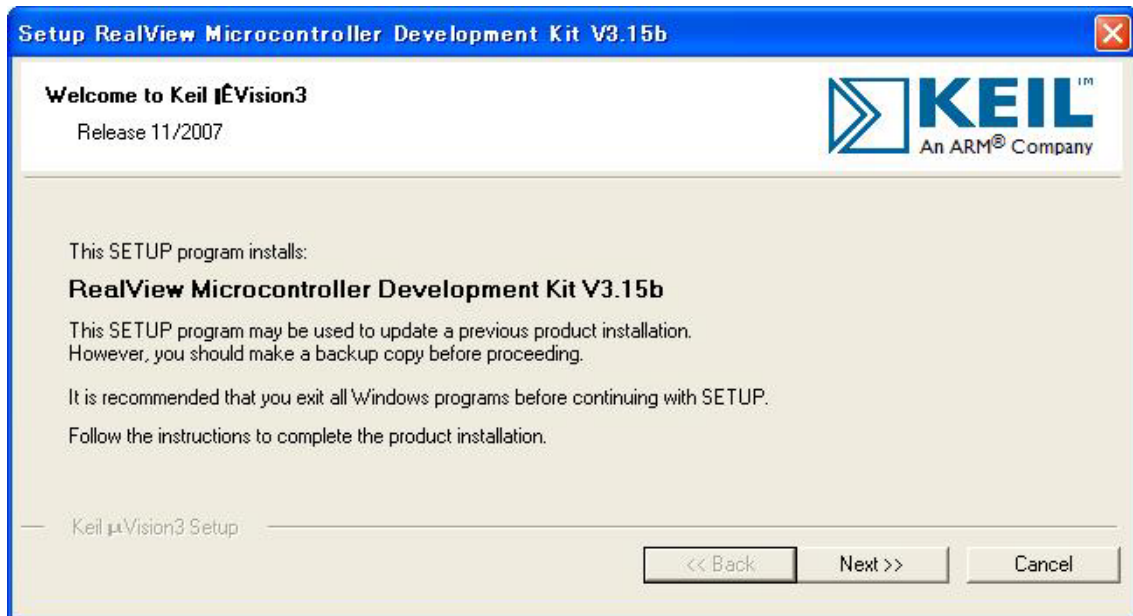
第四章 開発ツール KEIL のインストール

MDK315B.exe は開発ツール KEIL のデモ版です。

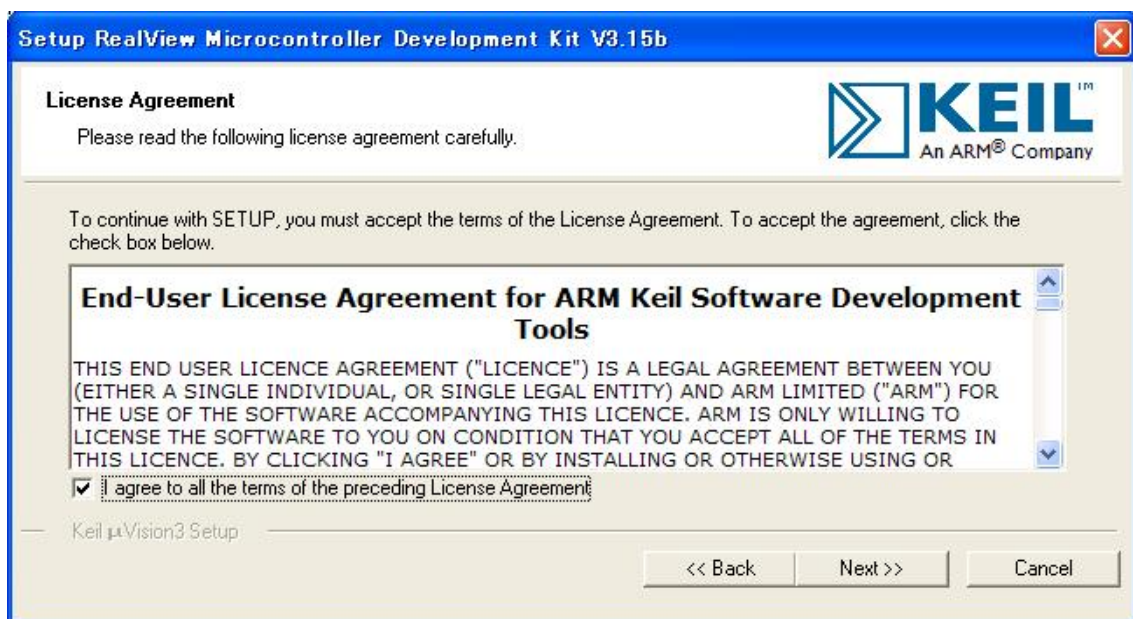
rlarm313a.exe は開発ツール KEIL のライブラリです。

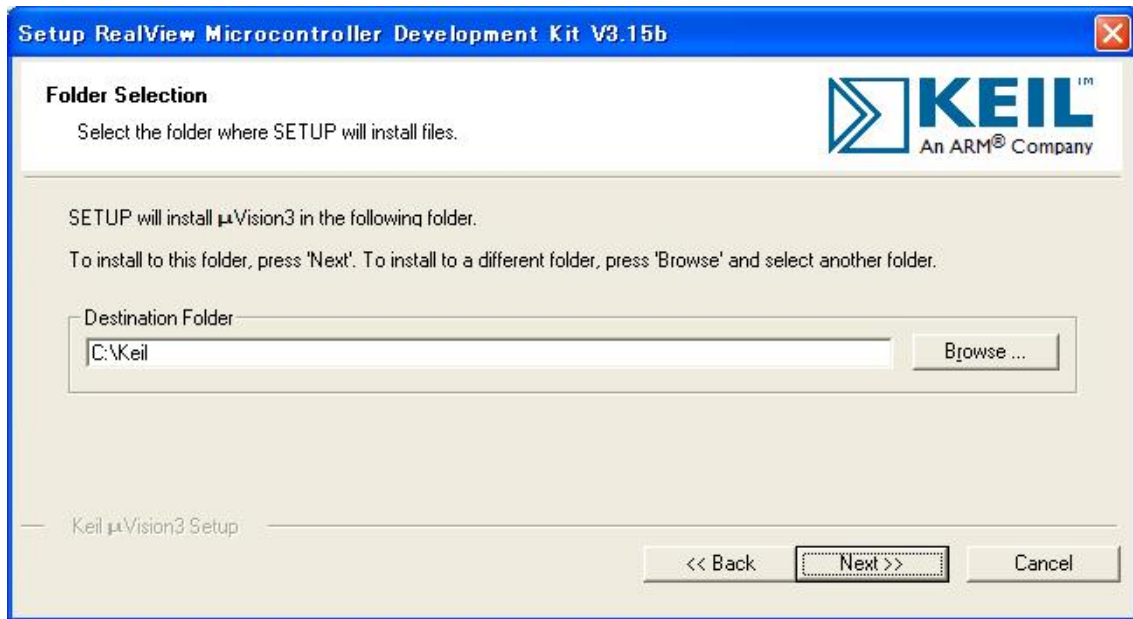
4.1 KEIL のインストール

まず、MDK315B.exe をクリックして、KEIL3.15 をインストールしてください。

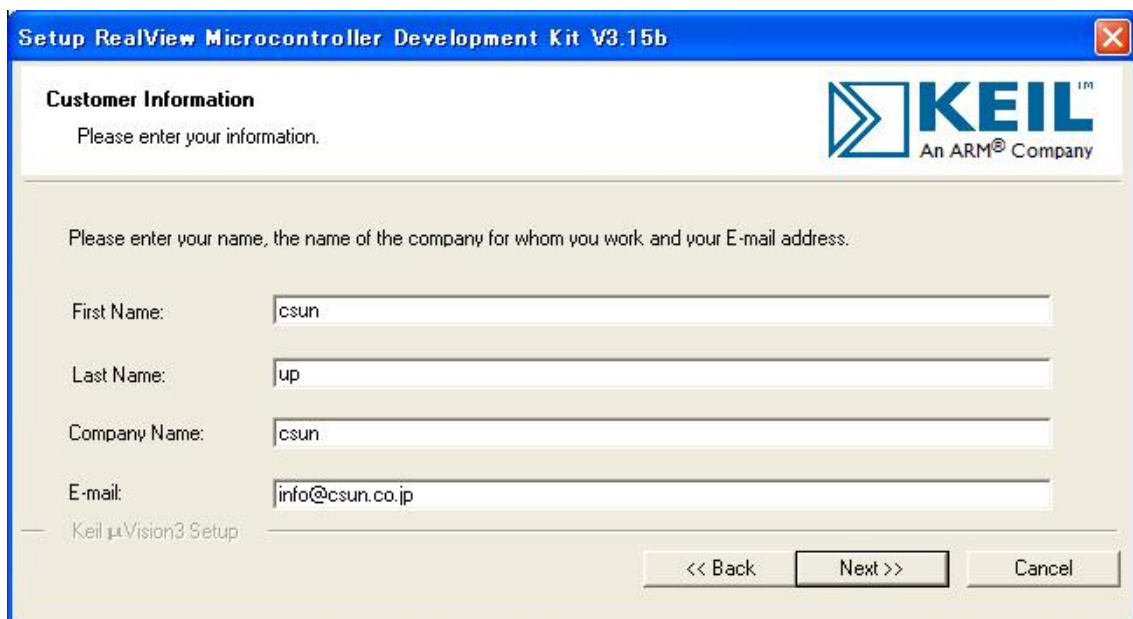


「Next」ボタンを押すと、英文のライセンスが出てきます。同意できる場合は、「I accept the terms of the license agreement」を選択して、「Next」ボタンを押します。

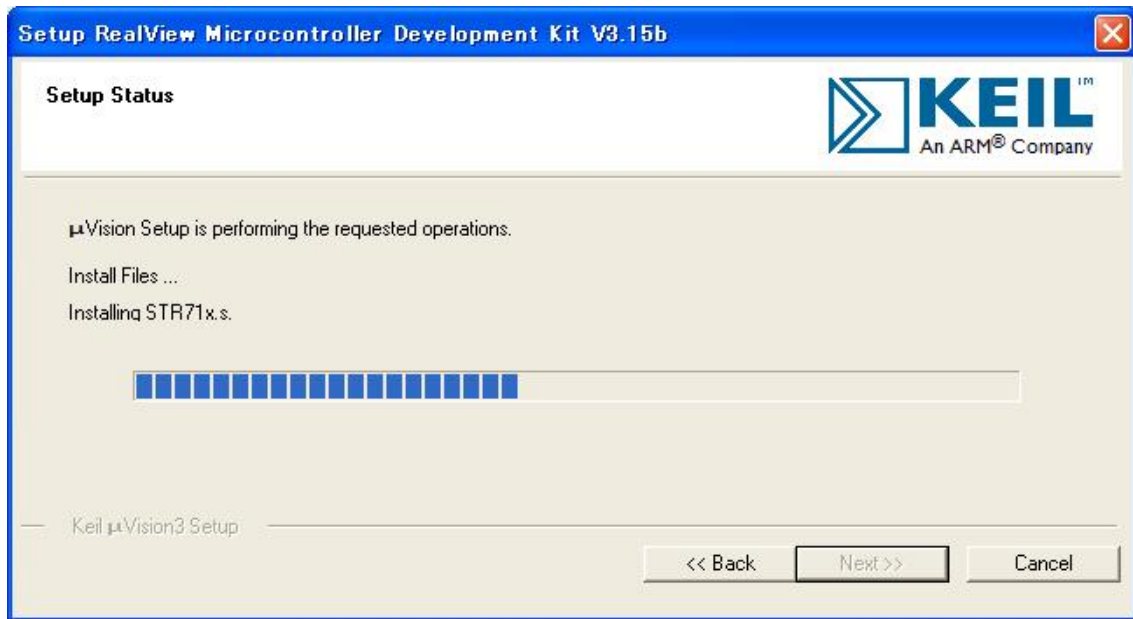




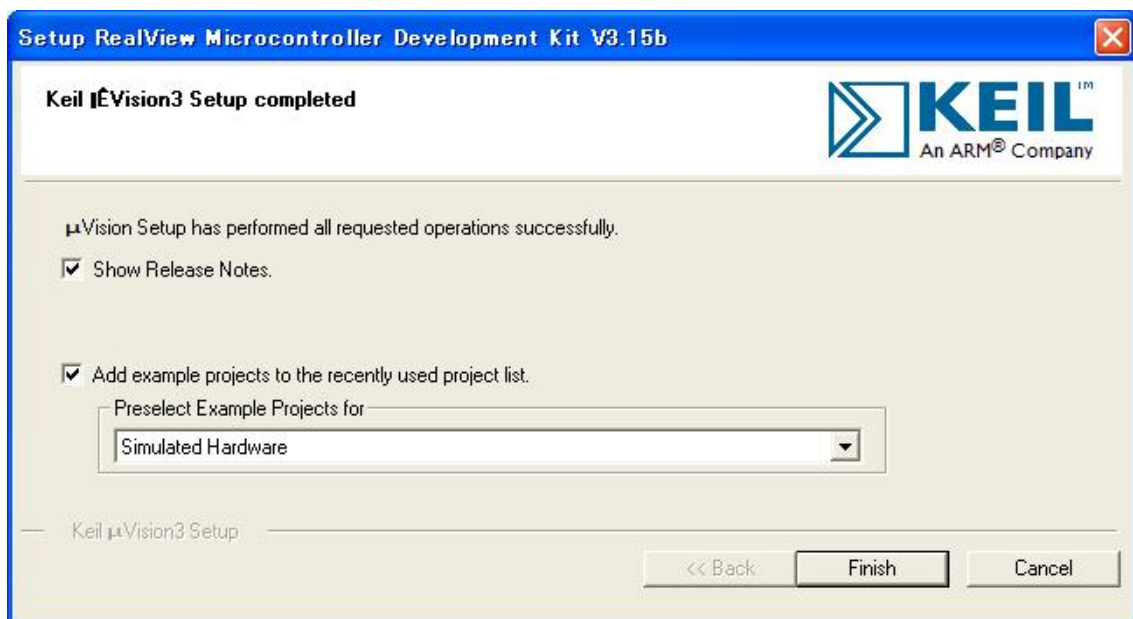
インストール先フォルダを変更せず、そのまま進んでください。



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ます。名前は半角のアルファベットで入力しましょう。



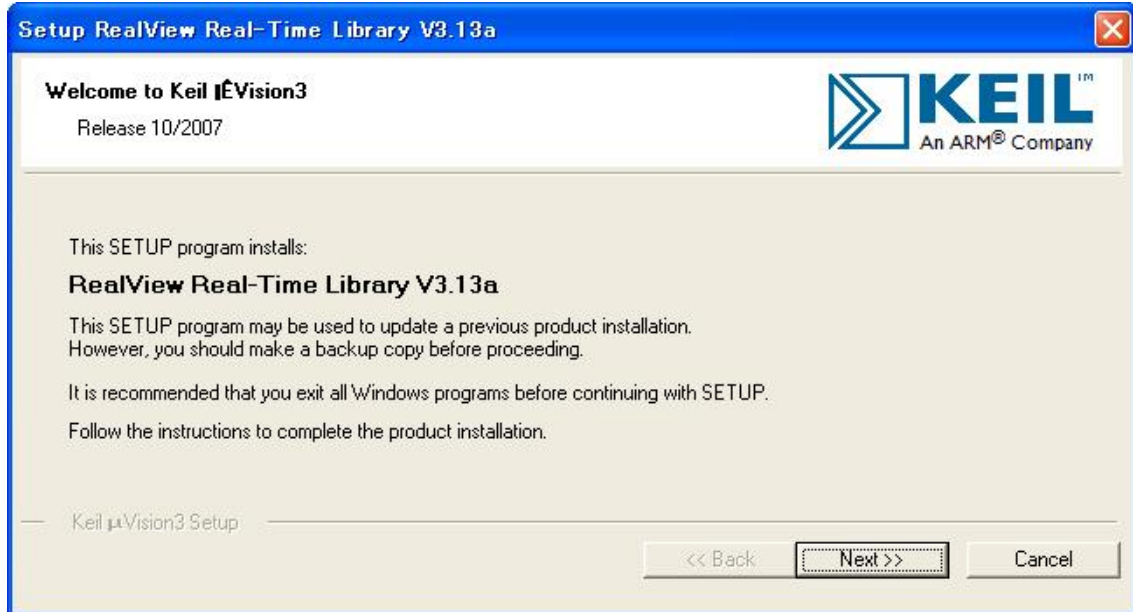
インストール中の画面です。



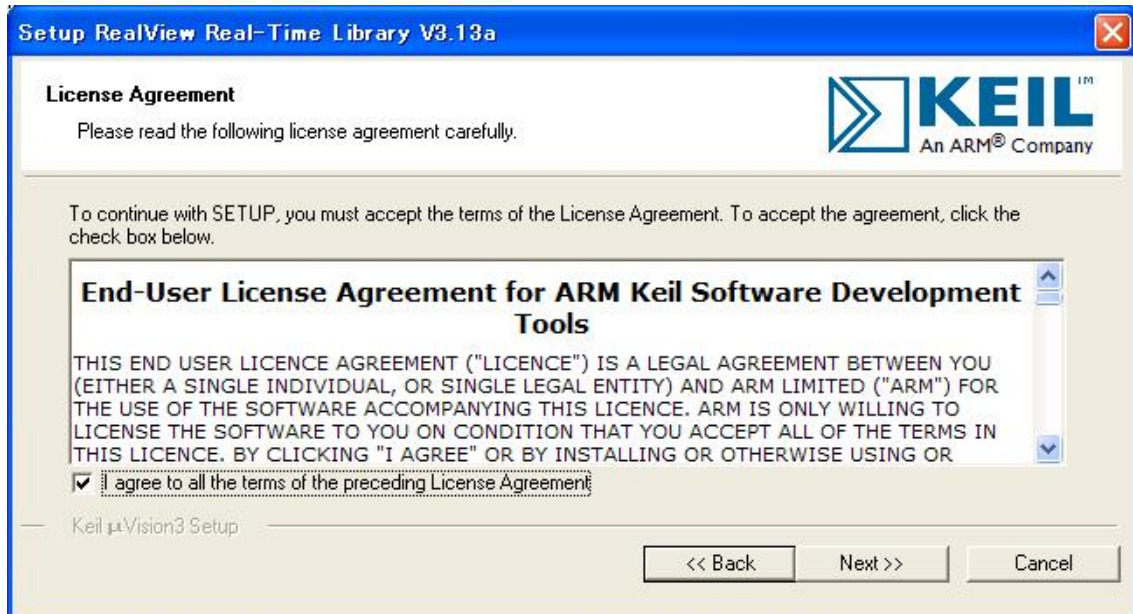
最後に「Finish」をクリックすると、ウィザードが閉じてインストールが終了します。

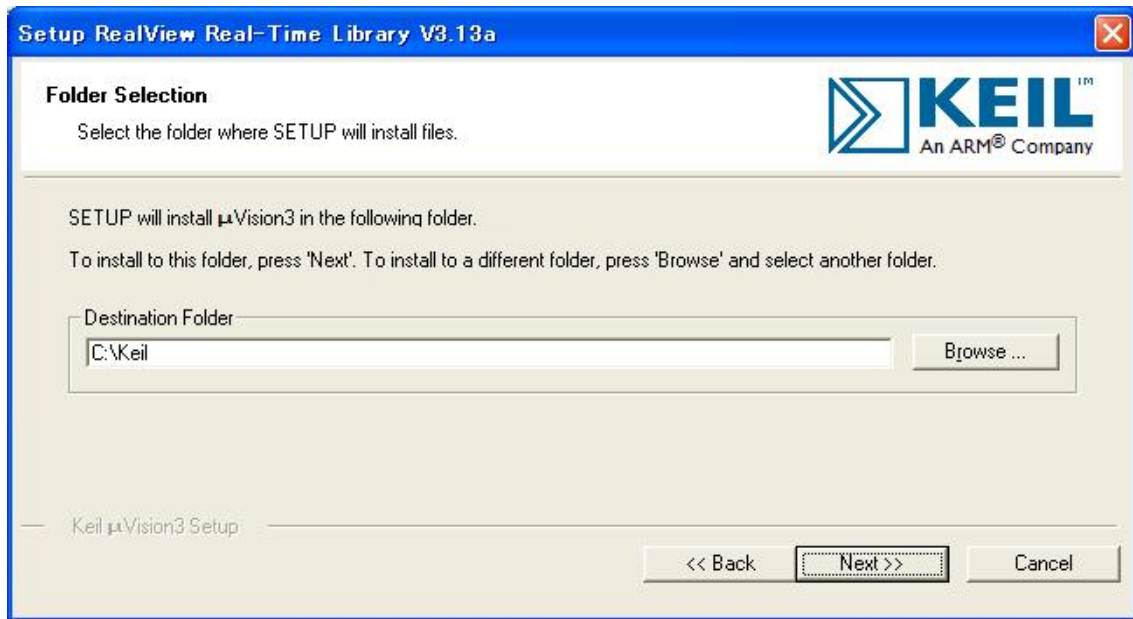
4.2 ライブラリのインストール

rlarm313a.exe をクリックして、KEIL のライブラリをインストールしてください。

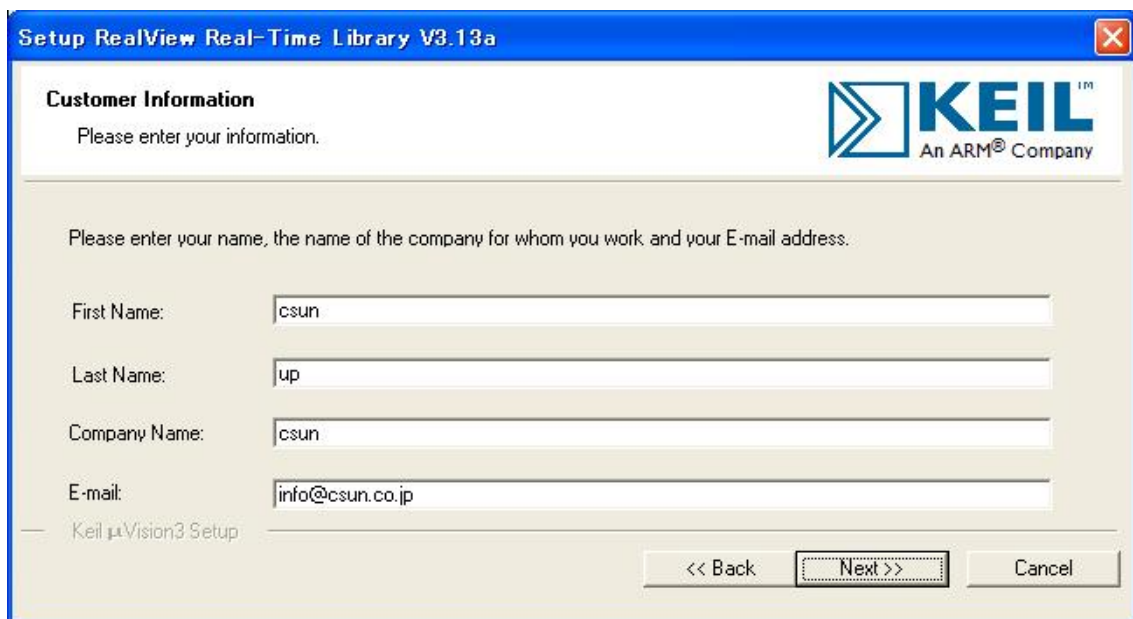


「Next」ボタンを押すと、英文のライセンスが出てきます。同意できる場合は、「I accept the terms of the license agreement」を選択して、「Next」ボタンを押します。

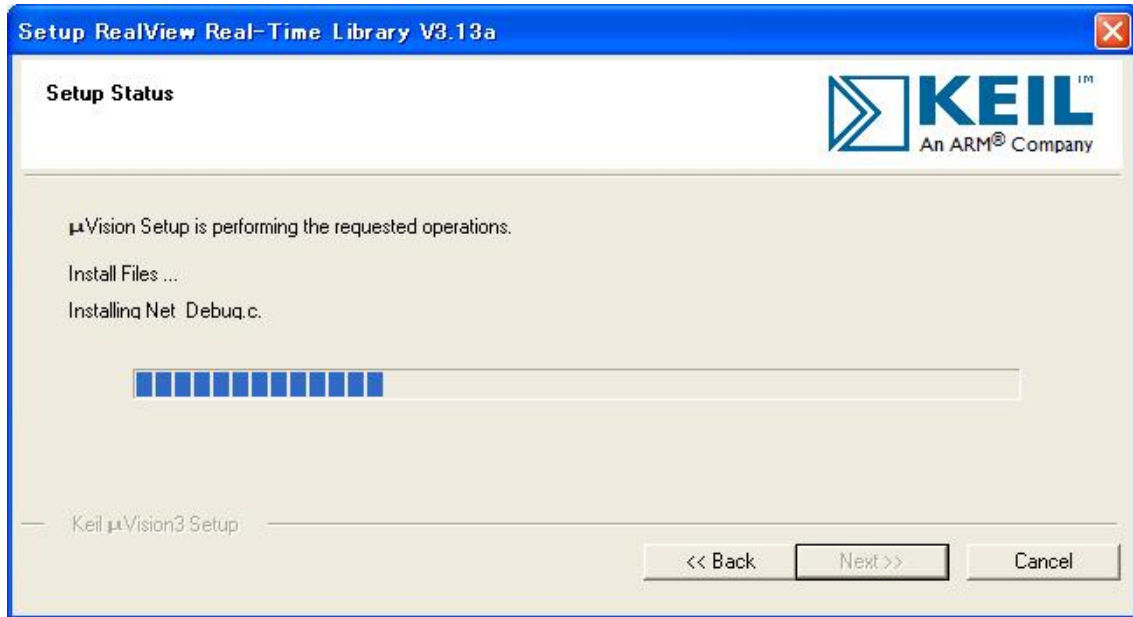




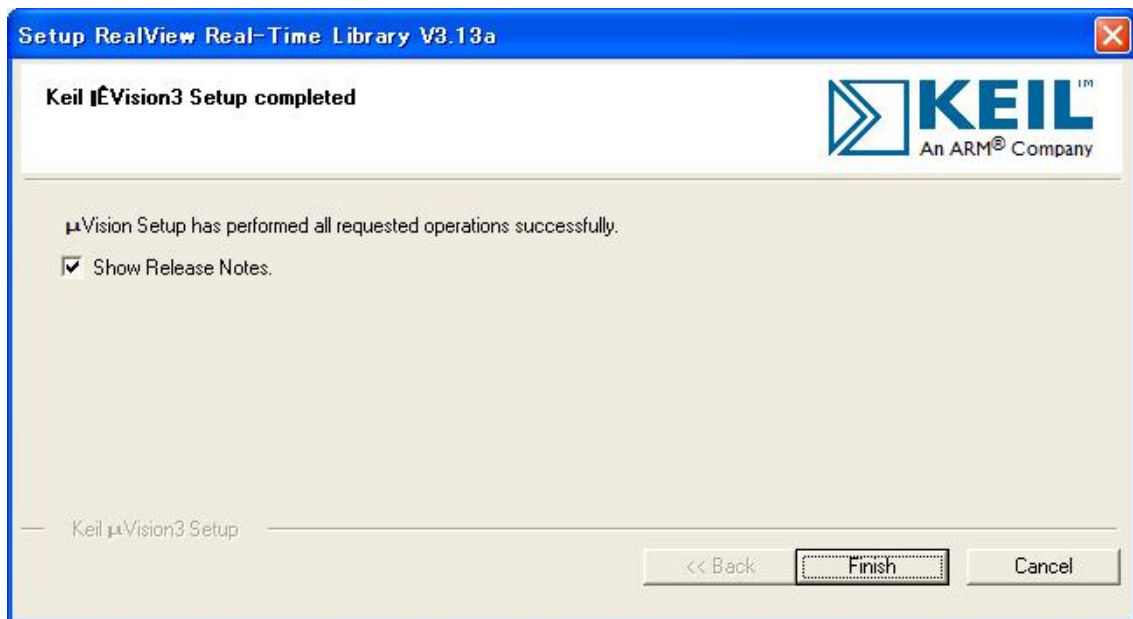
インストール先フォルダを変更せず、そのまま進んでください。



使用者の名前と所属会社名をを変更せず、そのまま進んでください。



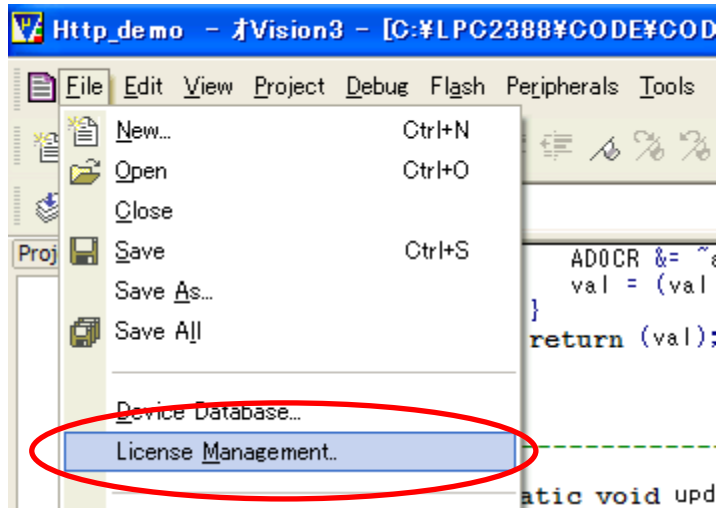
インストール中の画面です。



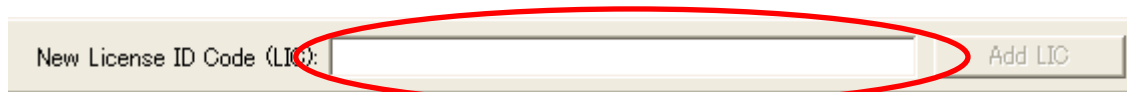
最後に「Finish」をクリックすると、ウィザードが閉じてインストールが終了します。

4.3 ライセンス

パソコンのメニュー：スタート →すべてのプログラム →Keil uVision3 を選択して、Keil を開きます。



Keil のメニュー「File」→「License Management」を選択してください。

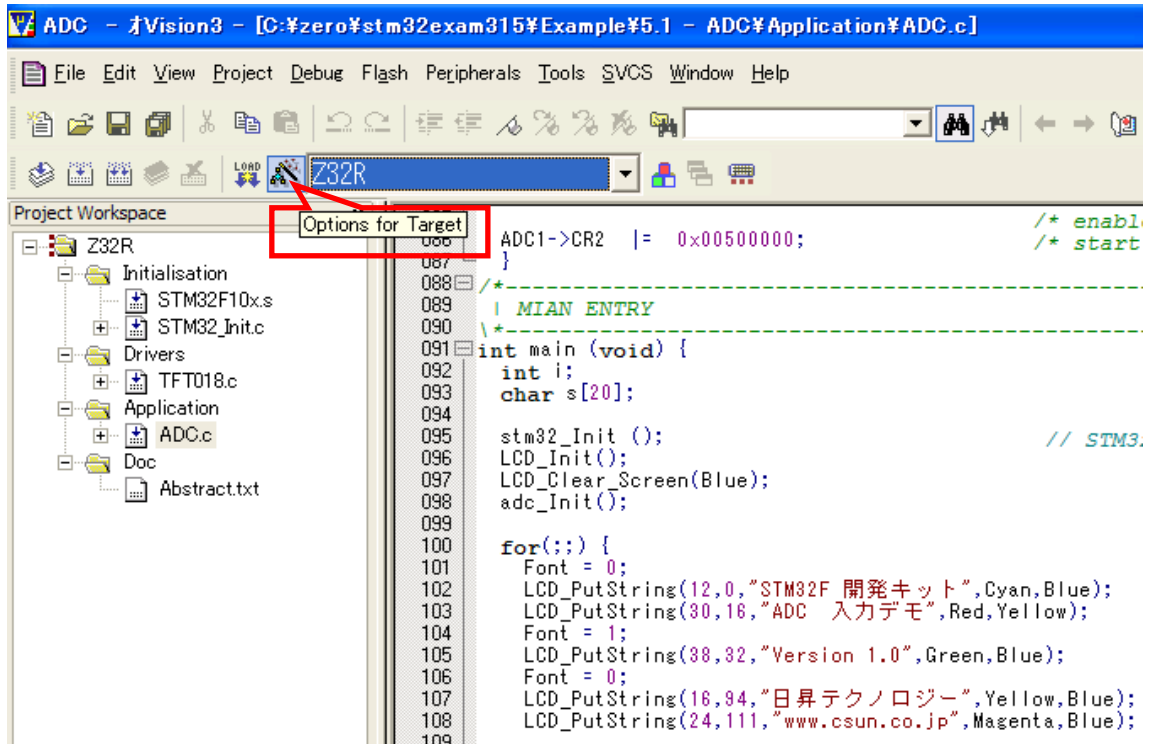


Keil 社からのライセンスを入力してください。

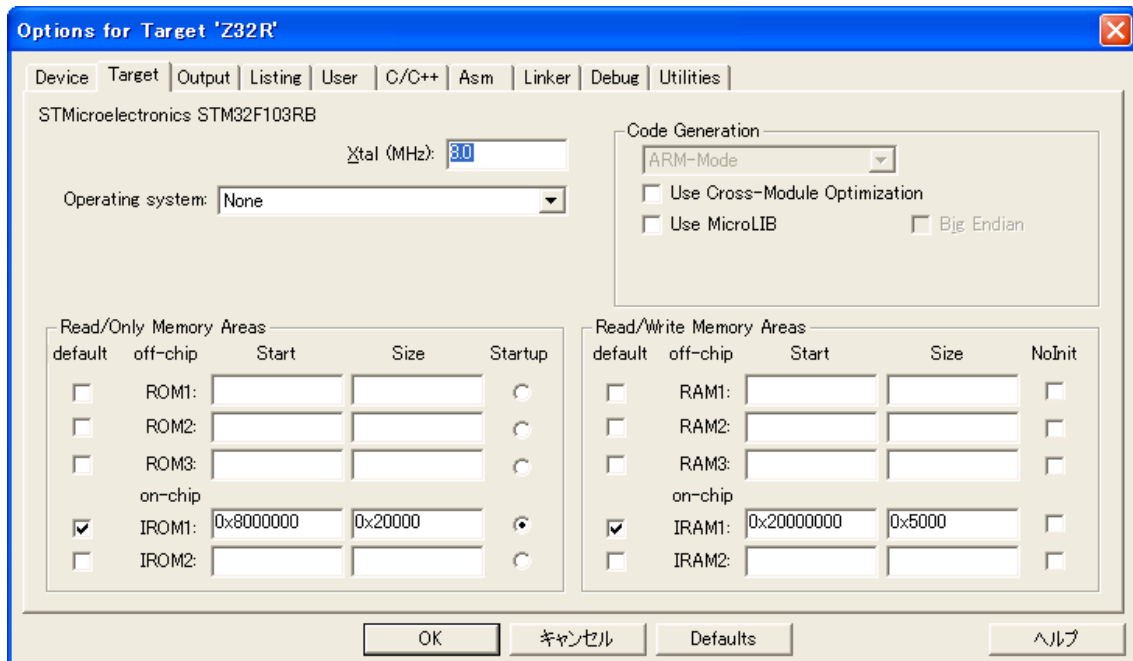
Keil社の日本代理店は<http://axe-inc.co.jp/>

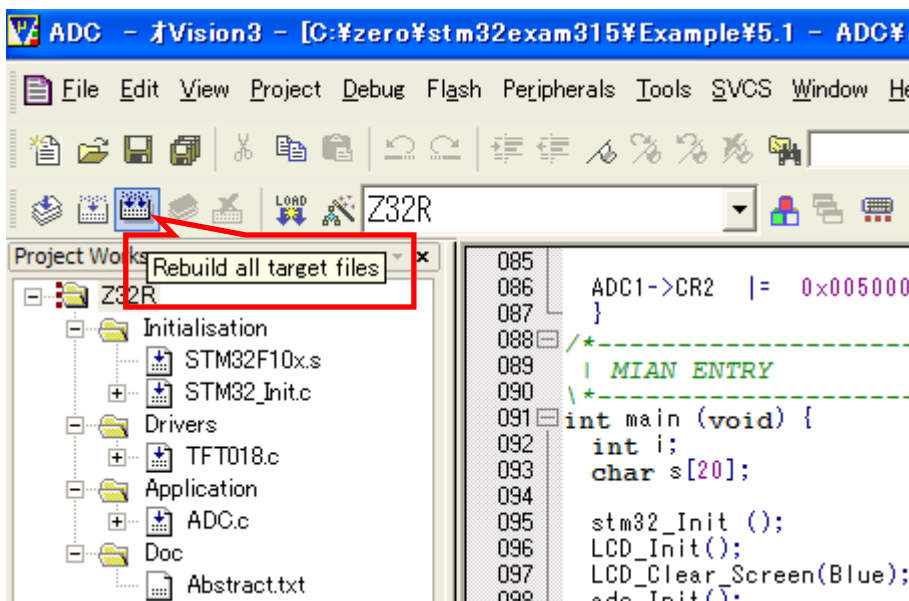
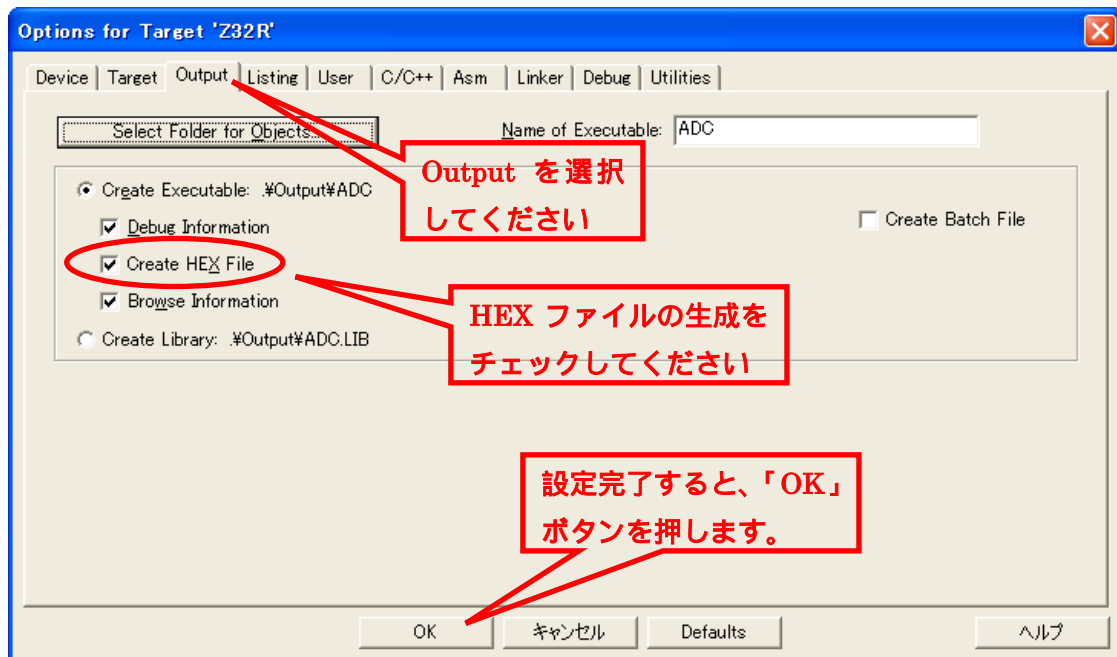
4.4 既存のプロジェクトから

プロジェクトファイル Example/5.1 - ADC/ ADC.Uv2.Uv2 をダブルクリックして、開きます。



ツールバーの「Options for Target」を押します。





ツールバーの「Rebuild all target files」を押すと、ビルドが開始します。ビルドが成功したら、プロジェクトの Output フォルダで ADC.hex ファイルを生成させます。この HEX ファイルを STM32F103 ボードに書き込んでください。

4.5 漢字の表示

STM32F103 の Flash は 128KB しかありません。第 1,2 水準の漢字フォントを入れることが不可能です。サンプルの漢字表示は小さい専用漢字フォントを使います。

プロジェクトの Drivers/フォルダに漢字フォントとドライバがあります。直せば、自分の漢字を表示できます。

GB1616.h は漢字フォントです。GB1616.h の一部の内容 :

```
struct typFNT_GB16
{
    unsigned char Index[3];           // JIS 漢字コード
    char Msk[32];                    // 漢字のドットマップ
};

const struct typFNT_GB16 codeGB_16[] =
{
    "日", 0x00,0x00,0x1f,0xf8,0x10,0x08,0x10,0x08,0x10,0x08,0x10,0x08,0x10,0x08,0x1f,0xf8,
    "昇", 0x00,0x00,0x1f,0xf8,0x10,0x08,0x10,0x08,0x10,0x08,0x10,0x08,0x10,0x08,0x1f,0xf8,0x00,0x07,0x20,
    "テ", 0x00,0x00,0x00,0x00,0x3f,0xfc,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7f,0xfe,0x00,0x80,
    "ク", 0x01,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    "ノ", 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    "口", 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    "ジ", 0x00,0x00,0x00,0x0a,0x0c,0x0a,0x06,0x00,0x02,0x00,0x00,0x00,0x30,0x02,0x18,0x06,
    "ー", 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

```

C 言語のフォーマット
で漢字のドットマップ
をコピーしてください。

```
STARTCHAR 01-38-92
ENCODING 18044
$WIDTH 1024 0
DWIDTH 16 0
EBX 16 16 0 -2
BTMAP
0000
1ff8
1008
1008
1008
1008
1008
1ff8
1008
1008
1008
1008
1008
1008
1008
1008
1ff8
0000
ENDCHAR
```

漢字「日」の
JIS コード

漢字「日」
16X16 ドット
マップ

jiskan16-2004-1.bdf.gz は第 1,2 水準の漢字フォントです。左の表はこのフォントの一部です。

漢字の JIS コードで漢字のドットマップを探して、C 言語のフォーマットで GB1616.h ファイルにコピーしてください。

```
#include "GB1616.h" //16*16 漢字ドット
void PutGB1616(unsigned short x, unsigned short y, unsigned char c[2], unsigned int f, unsigned int b){
    unsigned int i,j,k;

    LCD_SetArea(x, y, x+16-1, y+16-1);
    LCD_Inst(0x2C);

    LCD_CS(0);
    LCD_RD(1);
    LCD_RS(1);

    for (k=0;k<22;k++) {
        if ((codeGB_16[k].Index[0]==c[0])&&(codeGB_16[k].Index[1]==c[1])){
            for(i=0;i<32;i++) {
                unsigned short m=codeGB_16[k].Msk[i];
                for(j=0;j<8;j++) {
```

漢字の数量に
よって、直し
てください

これは TFT 液晶ドライバ TFT018.c の一部です。小さい漢字フォントの漢字の数量によって、ループの終了条件を直してください。

第五章 Thumb-2 対応 GCC クロス開発環境

KEIL 社の開発ツールが便利ですが、無償評価版は最大コード・サイズ 16K バイトの制限があります。

「Interface」誌 2008 年 11 月号で Thumb-2 対応 GCC クロス開発環境の構築を紹介しました。クロス開発ツールの構築環境には Cygwin を用いています。まだ Cygwin を入手していない読者は、

<http://www.cygwin.com/setup.exe>

からインストールしてください。この際に、Devel カテゴリのツールをすべてインストールしてください。

GNU 開発環境の構築は時間がかかる作業なので、「Interface」誌の Web ページ <http://www.cqpub.co.jp/interface/download/contents.htm> から、直接にダウンロードしてください。このパッケージを /usr/local に展開すると、使えるようになります。

```
$ cd /usr/local
$ tar jxvf arm-tools-new-20080625.tar.bz2
```

Web ページから GCC 環境のサンプルと GDB でのデバッグ用スタブもダウンロードできます。ご覧ください。

GDB スタブ : gdb-stub_20080830.tar.bz2

GCC サンプル : gcc_sample_20080830.tar.bz2

gcc_example_for_STM32F103.rar は STM32F103 開発キット用のサンプルです。

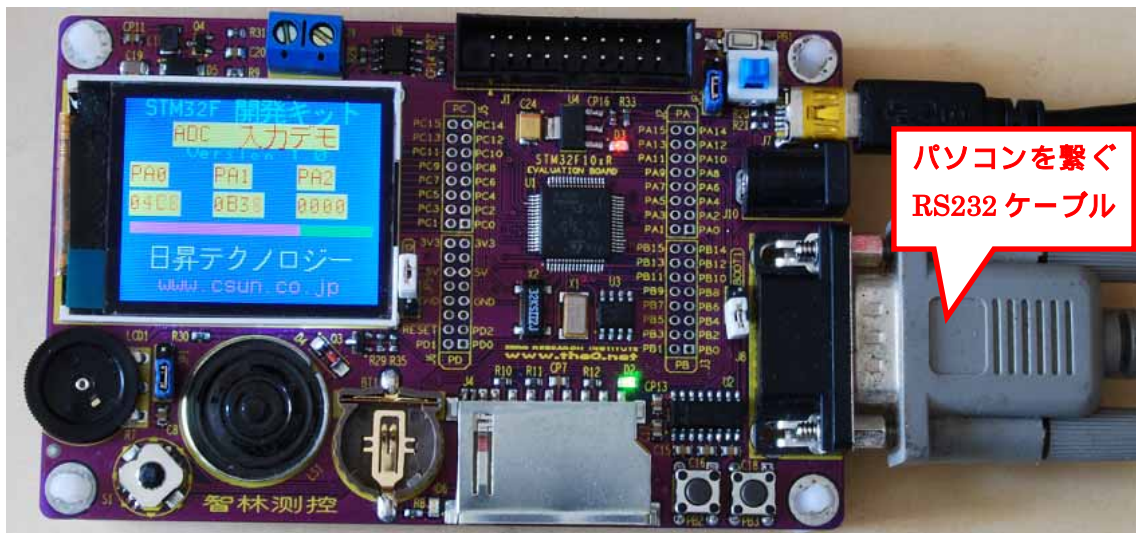
gcc_example_for_STM32F103/cortex_m3 は「Interface」誌のサンプルから改修したものです。一つの変更点は外部 12MHz 水晶を使い、CPU を 72MHz 駆動します。もう一つは LED 点灯のポートを変更しました。

gcc_example_for_STM32F103/ADC-LCD は Keil 環境のサンプル「5.1 - ADC」から移植したものです。

サンプルのコンパイル：

```
$ cd gcc_example_for_STM32F103/ADC-LCD
$ make
```

生成した*.HEX ファイルを STM32F103 ボードに書き込んでください。



GCC 版 ADC デモが動く様子

パソコンのハイパーターミナルで AD の数値が見えます。LED も制御できます。ハイパーターミナルの設定：115200 ビット/秒、フロー制御なしです。



```
arm9 - ハイパーターミナル
ファイル(E) 編集(E) 表示(V) 通信(C) 転送(T) ヘルプ(H)
0B36
OFF
0B38
0B38
0B34
0B38
ON
0B33
ON
0B35
0B33
0B33
0B33
0B39
-
```

キー「0」を入力すると、LED が消灯。キー「1」を入力すると、LED が点灯。

第六章 TOPPERS/ASP

TOPPERS/ASPカーネル<http://www.toppers.jp/asp-kernel.html>(以下、ASPカーネル)は、TOPPERS新世代カーネルの基盤(出発点)となるものとして、TOPPERS新世代カーネル統合仕様に準拠した最初のリアルタイムカーネルです。ASP(Advanced Standard Profile)の名前が示す通り、 μ ITRON4.0仕様のスタンダードプロファイル準拠のリアルタイムカーネルであるTOPPERS/JSPカーネルを拡張・改良する形で開発しました。

6.1 TOPPERS/ASP の初体験

STM32F103 開発キット用のパッケージ [asp_csun_starm_gcc-20090322.tar.gz](http://www.csun.co.jp/asp_csun_starm_gcc-20090322.tar.gz)
CQ-STARM (CQ 出版) 簡易パッケージ [asp_cq_starm_gcc-20081003.tar.gz](http://www.csun.co.jp/asp_cq_starm_gcc-20081003.tar.gz) から改修した
のです。

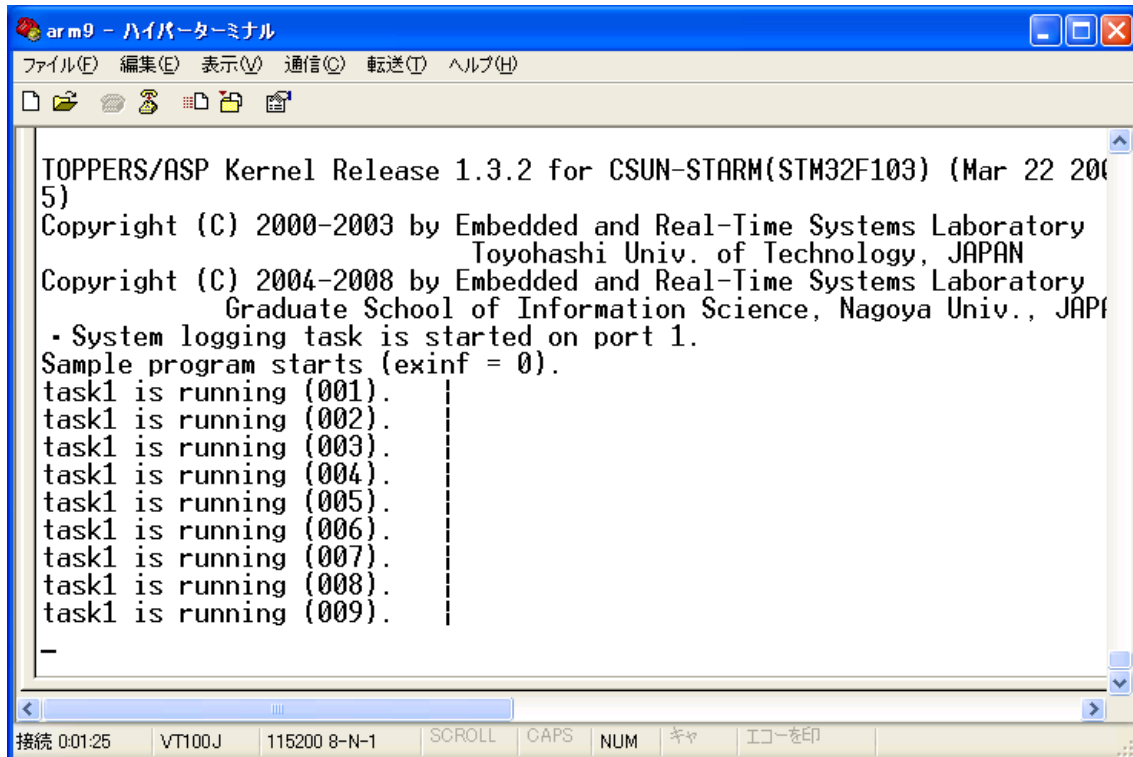
STM32F103 開発キットと CQ-STARM の異なる点 :

- 発振子周波数 12MHz
- LED 用のポート(PB5)

パッケージの展開：

```
$ tar zxvf asp_csun_starm_gcc-20090322.tar.gz
```

asp/CSUN_STM32F103/asp.hex は生成された TOPPERS/ASP サンプルです。asp.hex を STM32F103 開発キットに書き込んで、動く様子：



```
arm9 - ハイパーターミナル
ファイル(F) 編集(E) 表示(V) 通信(C) 転送(T) ヘルプ(H)
TOPPERS/ASP Kernel Release 1.3.2 for CSUN-STARM(STM32F103) (Mar 22 200
5)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
Toyohashi Univ. of Technology, JAPAN
Copyright (C) 2004-2008 by Embedded and Real-Time Systems Laboratory
Graduate School of Information Science, Nagoya Univ., JAPAN
- System logging task is started on port 1.
Sample program starts (exinf = 0).
task1 is running (001).
task1 is running (002).
task1 is running (003).
task1 is running (004).
task1 is running (005).
task1 is running (006).
task1 is running (007).
task1 is running (008).
task1 is running (009).
-
```

パソコンのハイパーターミナルの設定：115200 ビット/秒、フロー制御なし

STM32 シリーズの System memory エリアには、工場出荷時に USART 経由でプログラムが可能なブート・ローダが予め書き込まれているので、前節紹介した書き換えツール **Flash Loader Demonstrator** はシリアルポートで内蔵フラッシュ ROM を更新できます。しかし、TOPPERS/ASP はこのブート・ローダを破壊しましたので、内蔵フラッシュ ROM が JTAG でしか更新できません。

6.2 TOPPERS/ASP のコンパイル

バイナリ版の開発ツールは

<http://www.codesourcery.com/sgpp/lite/arm/portal/subscription?@template=lite>

からダウンロードできます。GCC のバージョンは 4.3.2 です。

Cygwin 環境のツール：arm-2008q3-66-arm-none-eabi-i686-mingw32.tar.bz2

GNU/Linux 環境のツール : arm-2008q3-66-arm-none-eabi-i686-pc-linux-gnu.tar.bz2
このパッケージを展開すると、使えるようになります。

```
$ cd /  
$ tar jxvf arm-2008q3-66-arm-none-eabi-i686-mingw32.tar.bz2
```

~/bashrc ファイルを編集して、次の行を入れてください。

```
export PATH=$PATH:/arm-2008q3/bin
```

カーネルを構築する前に、まず、コンフィギュレータをコンパイルする必要がある。コンフィギュレータ (cfg プログラム) は、cfg ディレクトリに移動し、makedepend で依存関係ファイル (Makefile.depend) を生成した後、make コマンドにより構築される。

```
$ cd cfg  
$ make depend  
$ make
```

次に、ASP カーネル上で動作するサンプルプログラムを構築する方法を説明する。

まず、サンプルプログラムのオブジェクトファイルを置くディレクトリを作成し、コンフィギュレーションスクリプトを実行する。例えば、オブジェクトファイルを置くディレクトリを、ASP カーネルのソースファイルを展開したディレクトリの下に CSUN_STM32F103 という名称のディレクトリにする場合には、次のコマンドを実行する (ディレクトリの場所と名称は任意に決めてよい)。

```
$ mkdir CSUN_STM32F103  
$ cd CSUN_STM32F103  
$ perl ../configure -T csun_starm_gcc
```

コンフィギュレーションスクリプトの実行により、カレントディレクトリには、サンプルプログラムを構築するための Makefile、サンプルプログラム用のコンフィギュレーションファイル (sample1.cfg)、サンプルプログラム本体 (sample1.h および sample1.c) が生成される。

コンフィギュレーションスクリプトの実行後、必要であれば Makefile を修正する。

```
$(OBJCOPY) -O ihex $(OBJFILE) $(OBJNAME).hex
```

この行を Makefile に入れて、書き込むツール用の HEX ファイルを生成します。

その後、make depend で依存関係ファイル (Makefile.depend) を生成した後、make コマンドによりサンプルプログラムのロードモジュール (asp.srec または asp.hex) が生成できる。依存関係ファイルの生成には若干時間がかかる。

```
$ make depend  
$ make
```

ここで構築したサンプルプログラム (sample1.h、sample1.c、sample1.cfg) は、ASP カーネルの基本的な動作を確認するためのものである。このプログラムの概要説明は、sample1.c の先頭のコメントにある。

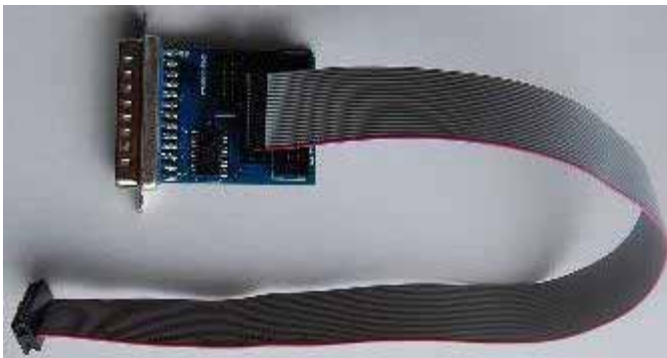
詳しい説明は doc/user.txt ファイルを参照してください。

6.3 H-JTAG で書き込む

ホームページ<http://www.hjtag.com>から最新版をダウンロードできます。

H-JTAGの特性：

- a. RDI 1.5.0 & 1.5.1 をサポートします；
- b. ARM7 & Contex-M3 & ARM9 (ARM9E-SとARM9EJ-Sを含む)；
- c. thumb & thumb 2 & arm 命令；
- d. little-endian & big-endian；
- e. semihosting；
- f. 実行環境WINDOWS 9.X/NT/2000/XP；
- g. flashの書き込み

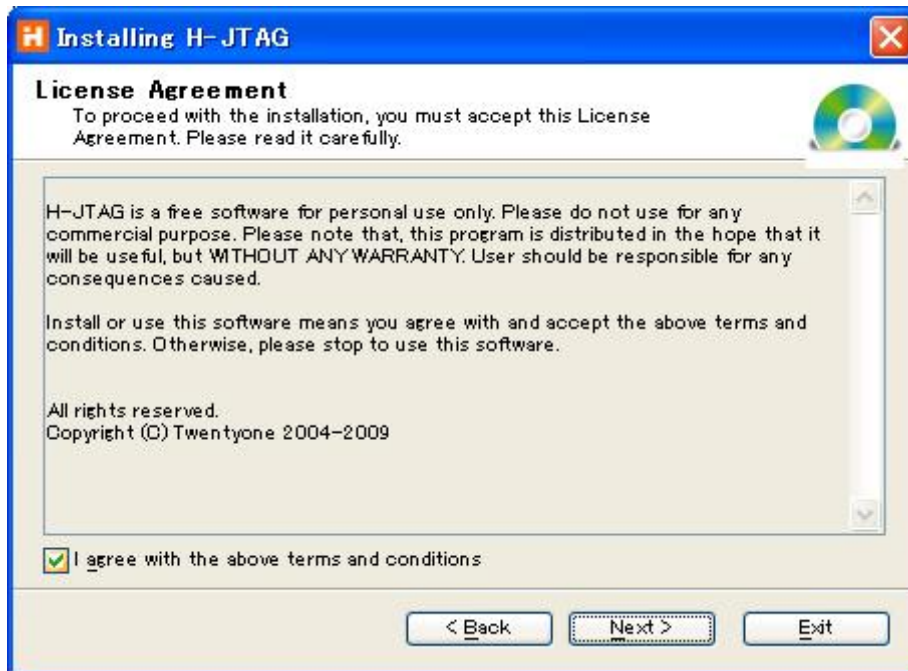


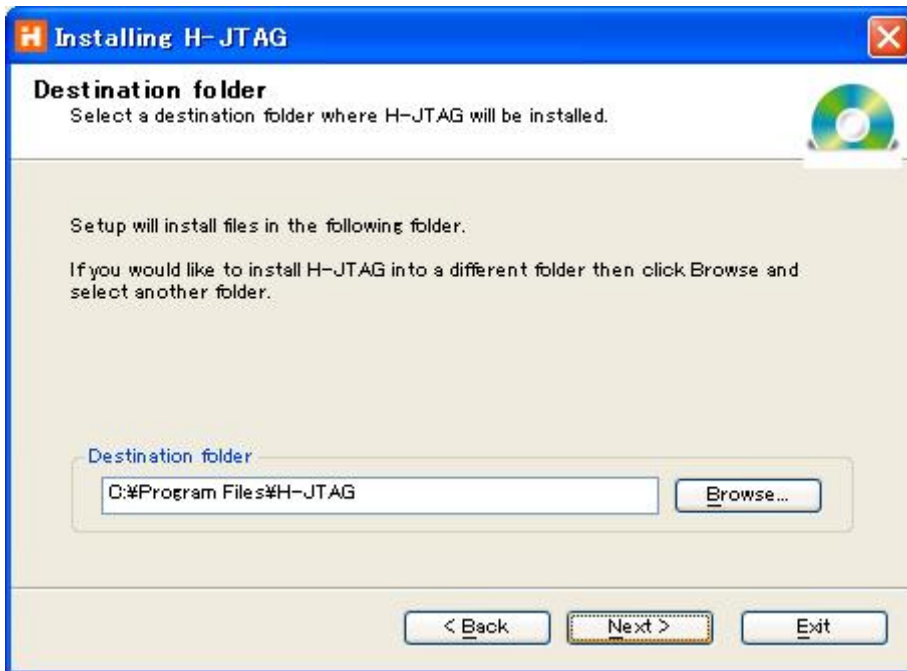
弊社は H-JTAG のハードウェアを提供しております。パソコンはLTPが必要です。

現時点最新版 : **H-JTAG V0.9.1.EXE**

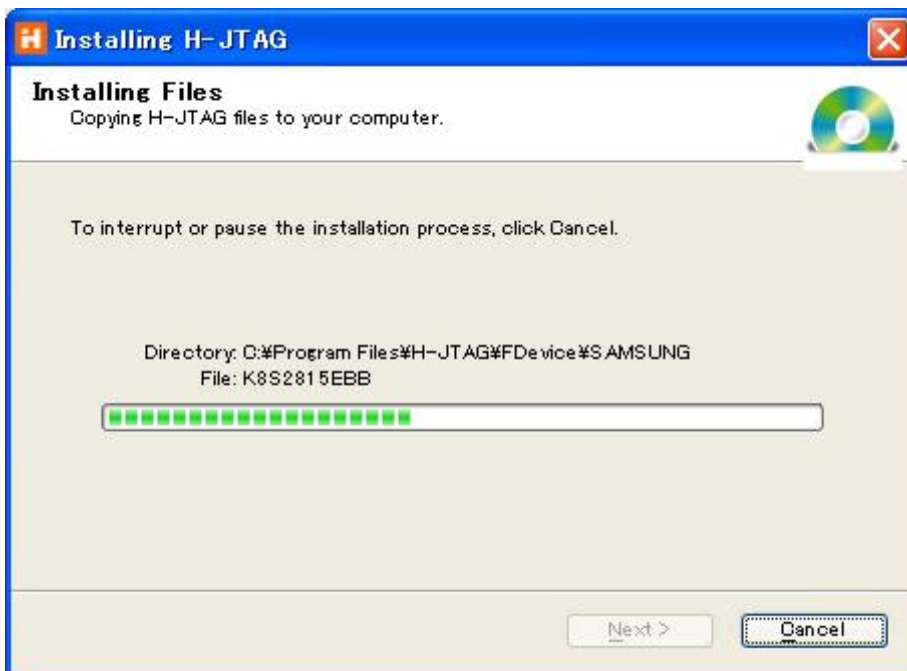


「Next」ボタンを押すと、英文のライセンスが出てきます。同意できる場合は、「Next」ボタンを押します。





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。

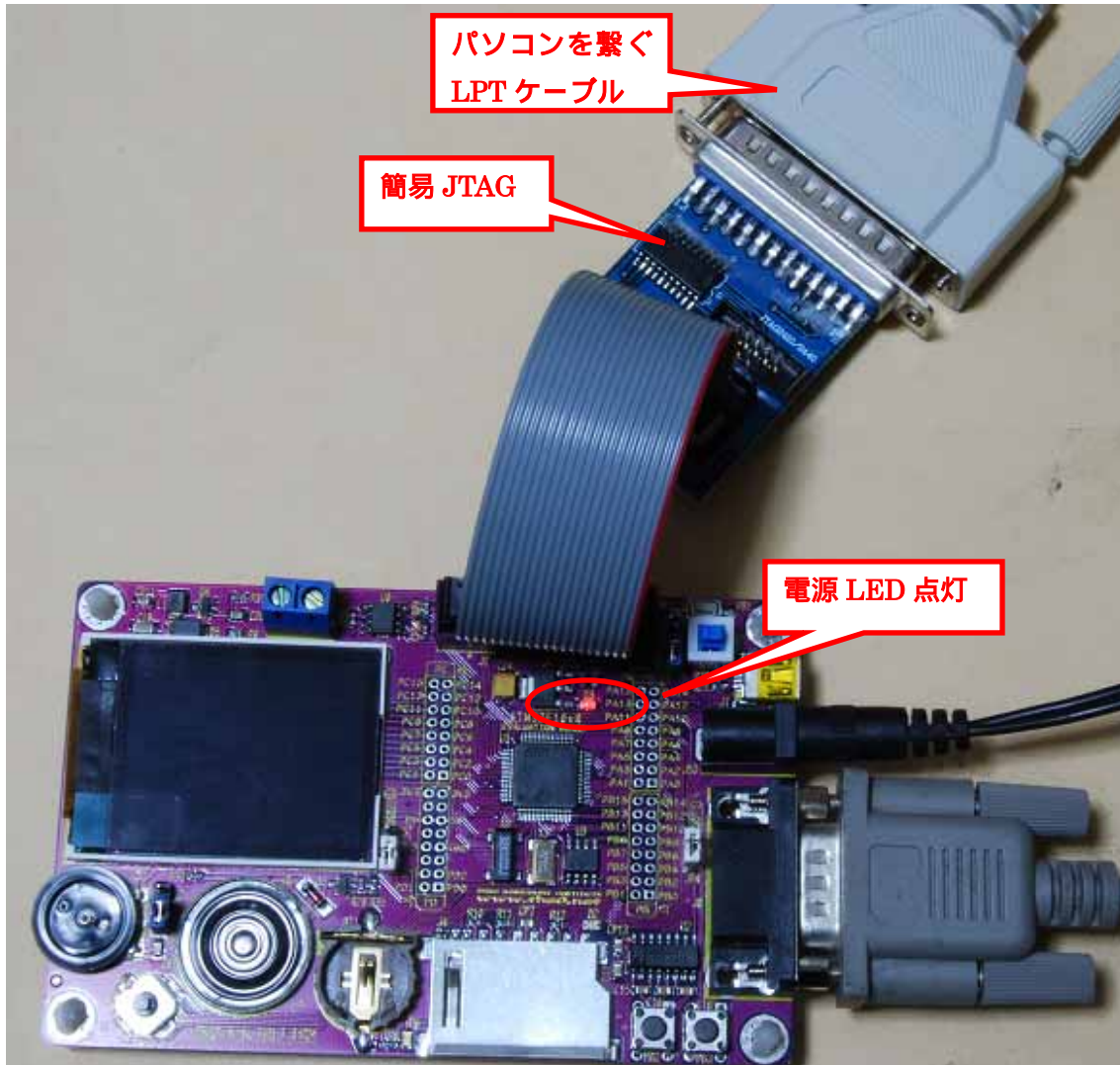


インストール中の画面です。

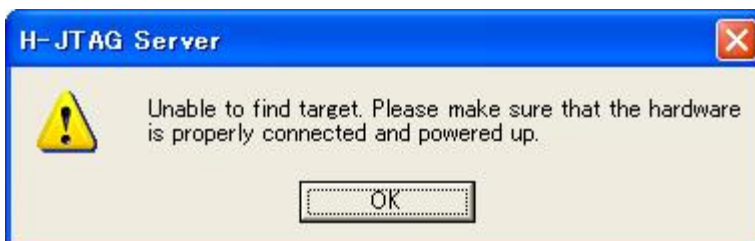


最後に「Finish」をクリックすると、ウィザードが閉じてインストールが終了します。

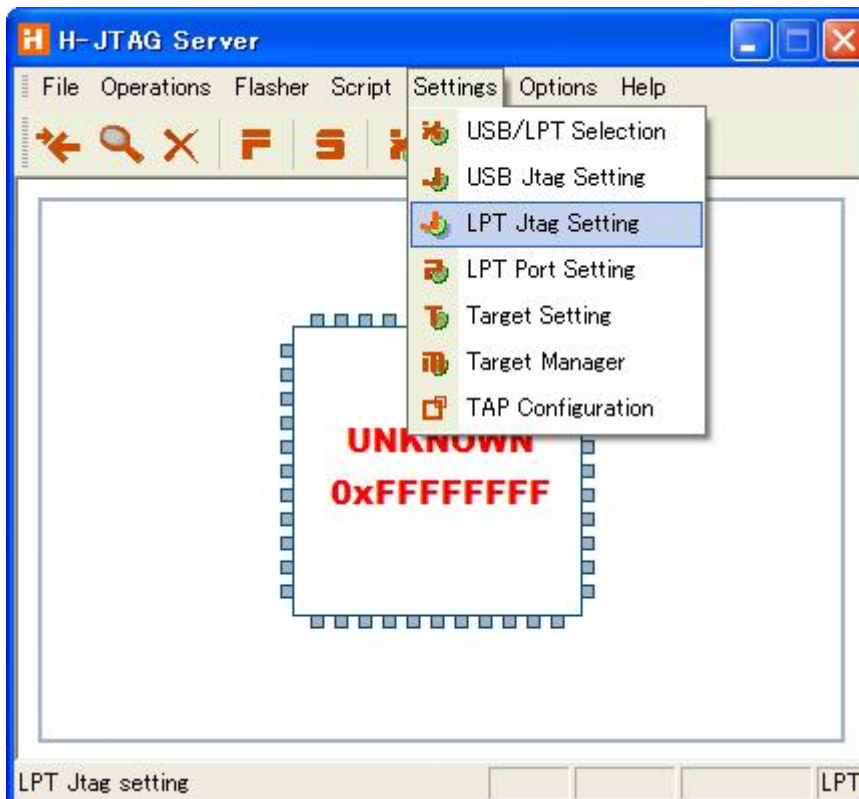
H-JTAG を実行する前に、まず、簡易 JTAG で STM32F103 開発キットとパソコンを繋ぎます。STM32F103 開発キットに電源を入れてください。



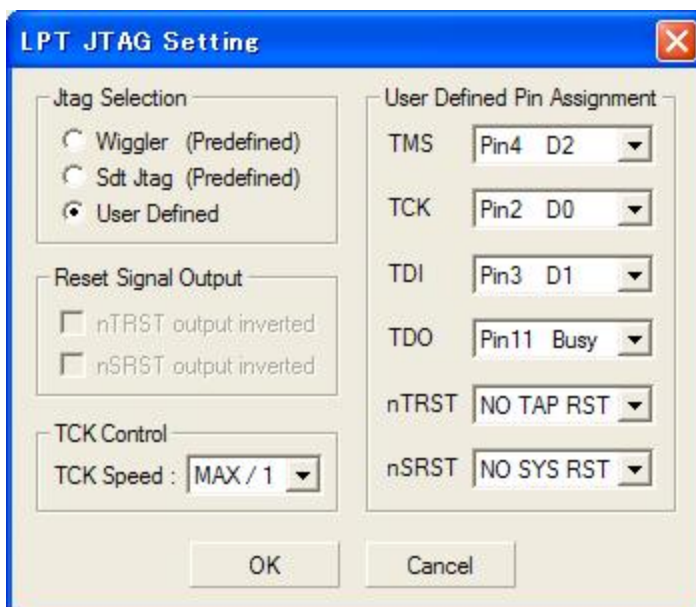
始めて H-JTAG を実行すると、このエラーメッセージが出てきます。



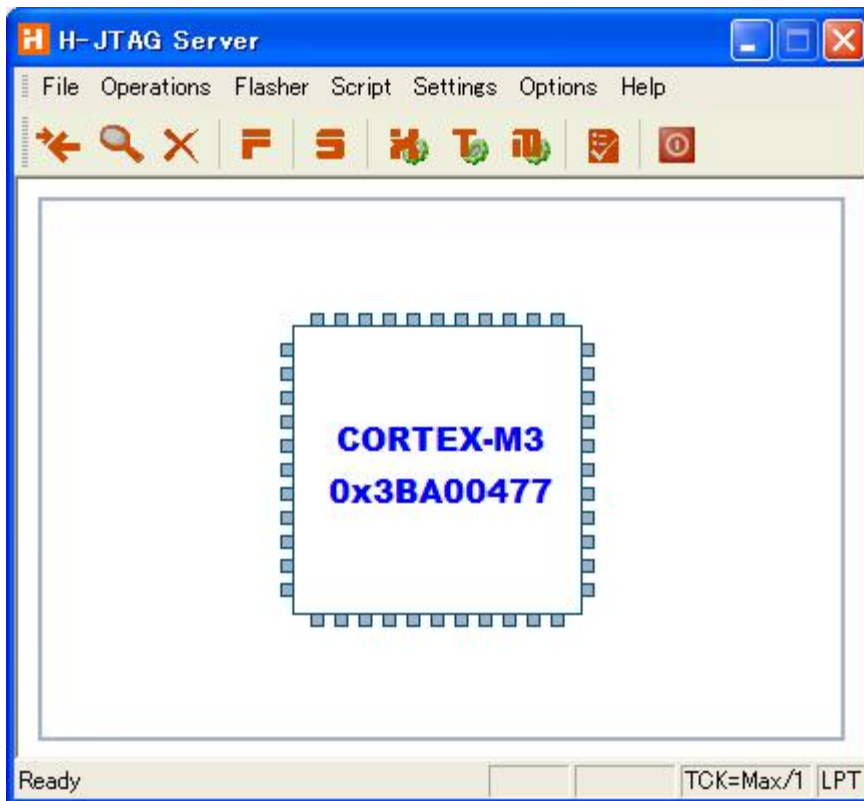
"Ok"ボタンを押すと、初の画面が出てきます。



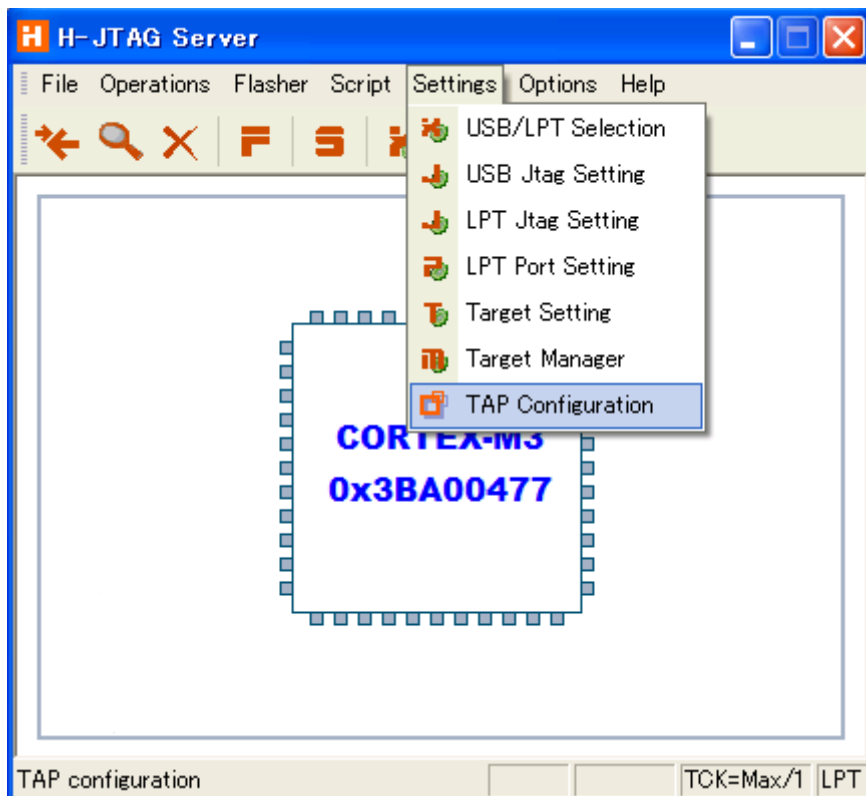
メニュー「Settings」→「LPT Jtag Setting」を選択してください。



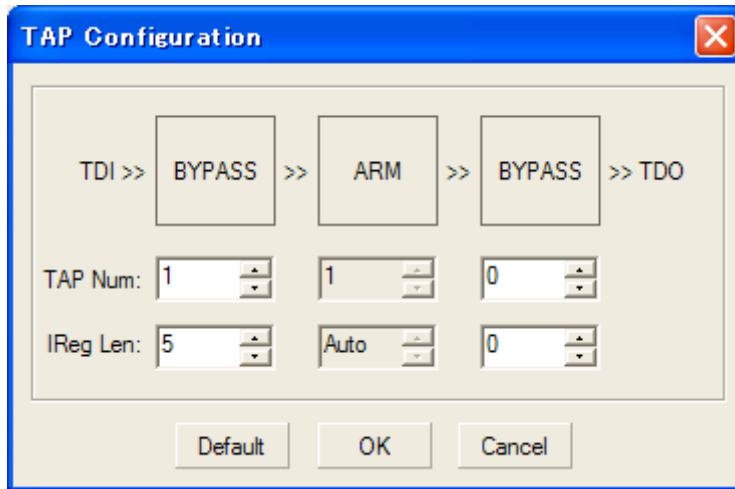
画面の通りに設定してください。「OK」ボタンを押すと、



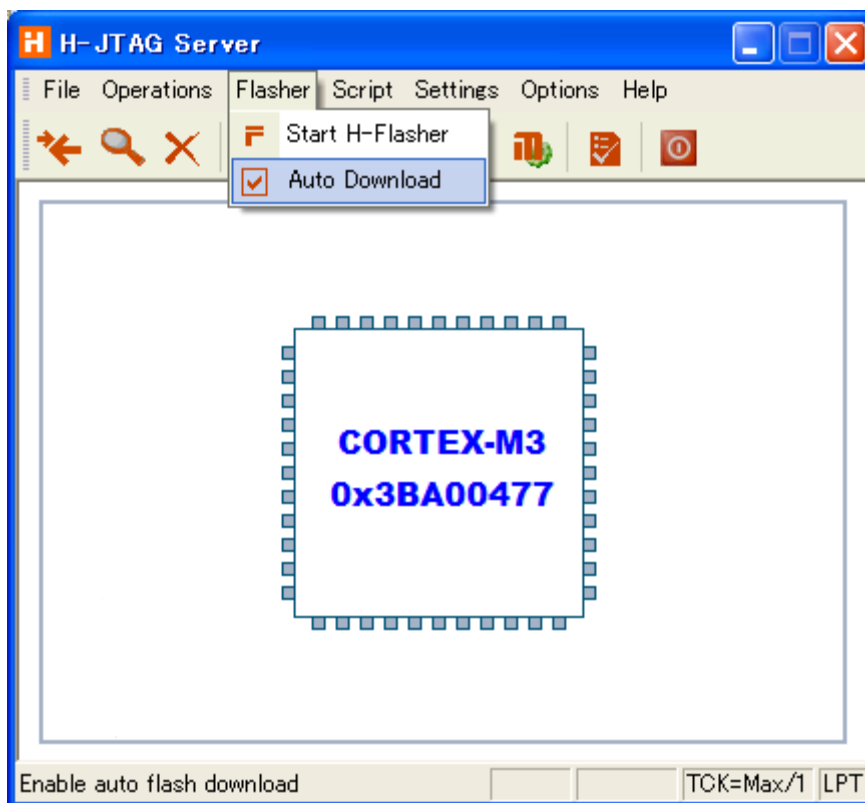
CORTEX-M3 は認識されました。



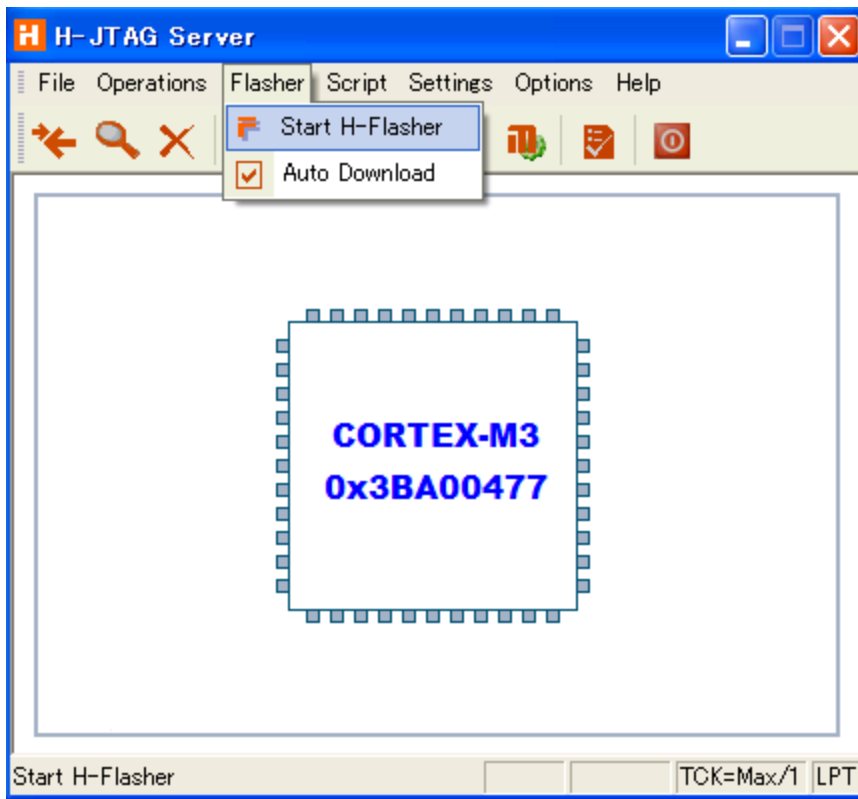
メニュー「Settings」→「TAP Configuration」を選択してください。



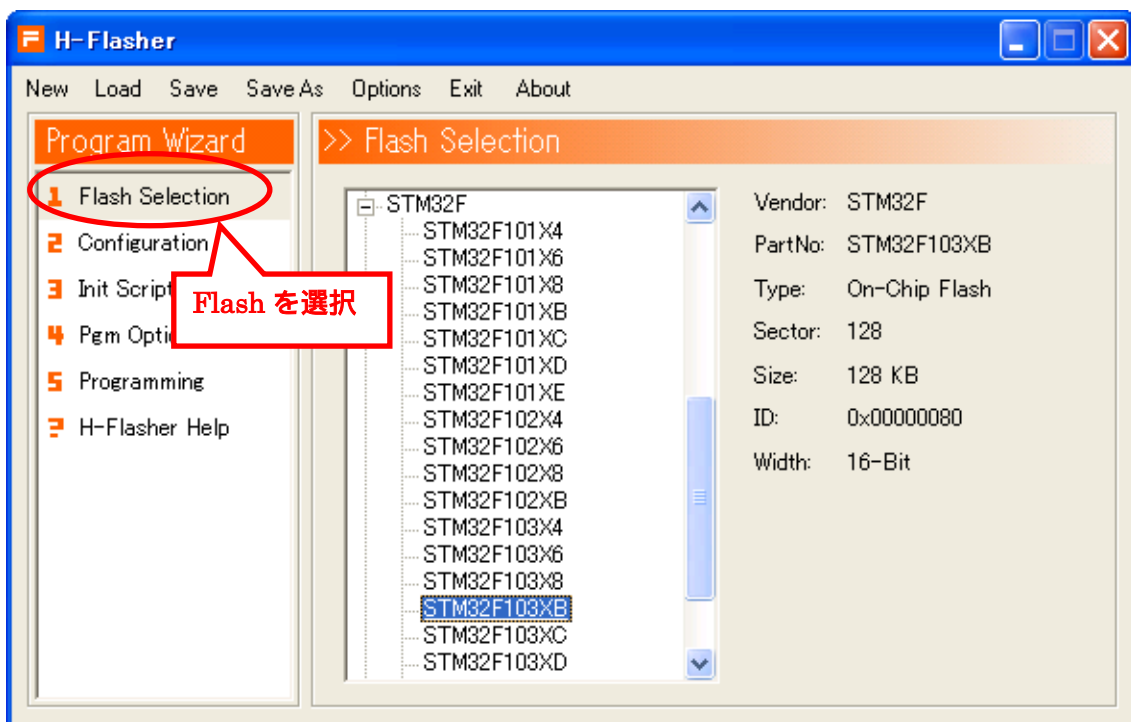
画面の通りに設定してください。



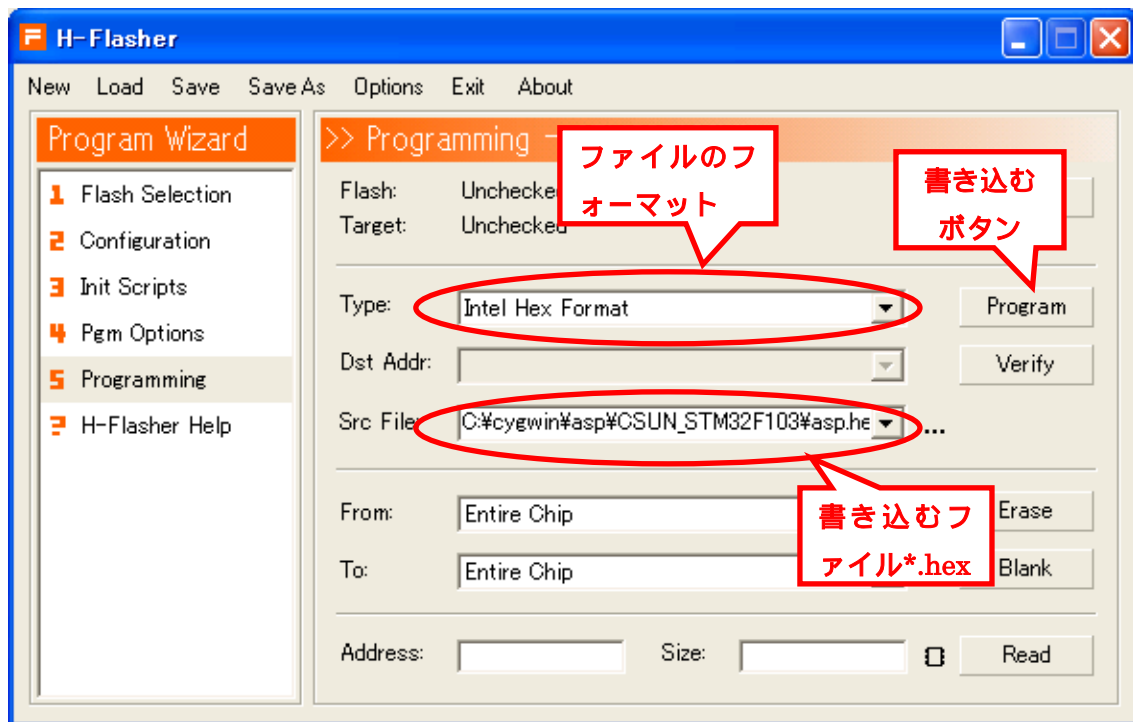
メニュー「Flasher」→「Auto Download」をチェックしてください。



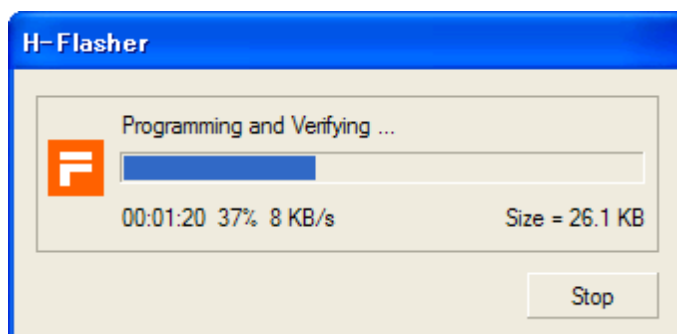
メニュー「Flasher」→「Start H-Flasher」を選択してください。



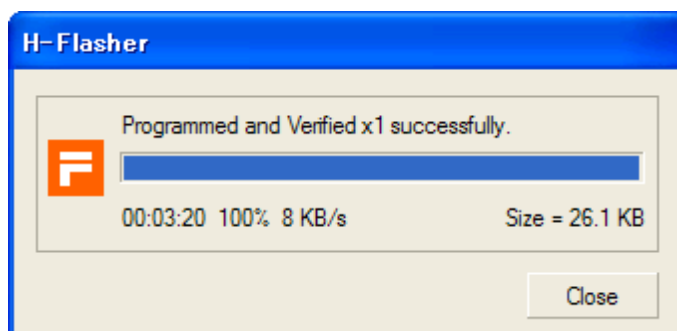
STM32F103XB を選択してください。



ファイルのフォーマットを「Intel Hex Format」を設定して、書き込む*.hex ファイルを選択して、書き込むボタンを押してください。



書き込中です。



最後に「Close」をクリックすると、ウィザードが閉じてインストールが終了します。