

# ルネサスエレクトロニクス製 SH2Tiny の活用

安井 教郎

2011 年 5 月

## 1 はじめに

試験治具などで、自動制御できたら楽だろうと思うシーンは皆さん経験があるだろう。この記事ではルネサスエレクトロニクス製 SH2Tiny を使ったマイコンボードで実現するにはどんな準備が必要であるか解説する。

## 2 ターゲット

今回は、秋月電子通商製の SH7125F マイコンボードを使用する。このボードには SH7124F と 12.5MHz の水晶振動子が実装されているだけのいたってシンプルなボードである。SH7125F の全ての端子がコネクタに引き出されている為、非常に使いやすい。図 1 にボードの外観を示す。

- |   |  |
|---|--|
| ● 内蔵 ROM:128kBytes  | ● シリアルコミュニケーションインタフェース (SCI):3 チャンネル             |
| ● 内蔵 RAM:8kBytes  | ● A/D 変換器:10bit 8 チャンネル                          |
| ● デバック IF:H-UDI   | ● I/O ポート:37 本の IO 端子と 8 本の IN 端子                |
| ● マルチファンクションタイマパルスユニット 2(MTU2):16bit タイマ 6 チャンネル 最大 16 種類のパルス出力 | ● パッケージ:QFP64(0.8mm) LQFP64(0.5mm) VQFN64(0.4mm) |
| ● ウォッチドッグタイマ  | ● 電源電圧:4.0~5.5V                                  |
| ● コンペアマッチタイマ (CMT):16bit 2 チャンネル                                |  |

デバック用にシリアル I/F を PC と接続するためにはレベルコンバータやシリアル-USB 変換が必要。今回は同じ秋月電子通商製の FT232RL USB シリアル変換モジュールを使い、USB に変換する。図 2 に USB シリアル変換モジュール外観を示す。実際に使用する場合には、図 3 の様にピンヘッダーを全部取り除き、ケーブルを配線後、熱収縮チューブで絶縁。こうすることで使いやすい。

今回は、AD9953 を使用した DDS(Direct Digital Synthesizers) の制御に使用することとする。図 32 に回路図を示す。ボード上にパワーオンリセット回路が実装されていないので、ユーザ側で追加している。U101 がリセット IC。実機を図 4 に示す。

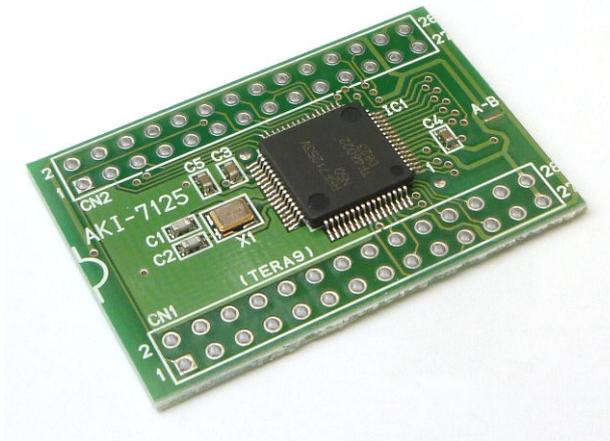


図1 SH7125F マイコンボード

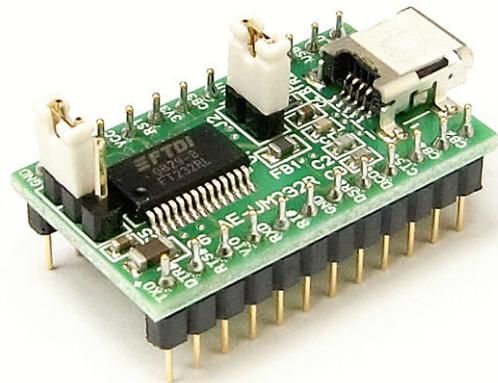
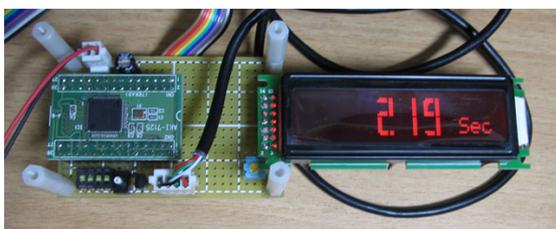


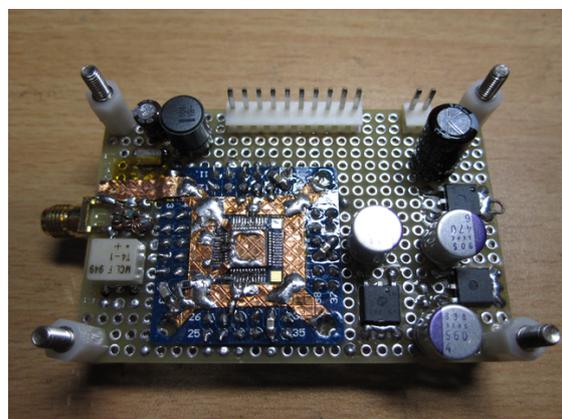
図2 FT232RL USB シリアル変換モジュール



図3 FT232RL モジュール加工例



(a) SH7125 部



(b) AD9953 部

図4 実機

### 3 開発環境

開発環境は、ルネサスエレクトロニクス製 HEW 評価版を使用する。試用期間が過ぎるとリンクサイズが 256kBytes 以内に制限される。しかし、SH7125F は 128kBytes の FlashROM 容量しか持たないため、実質的に制限とならない。現時点での最新版は、V9.04 Release00 である。

また、出来上がった HEX 形式の実行ファイルを SH7125F に書き込むには、シリアルポートを使った書き込みを行う。その為のツールもルネサスエレクトロニクス製 Flash Development Toolkit として用意されており、これまた評価版がある。最新版は V4.07 Release00 である。

評価版ソフトウェアツールのダウンロード URL は、[http://japan.renesas.com/products/tools/evaluation\\_software/evaluation\\_software\\_mid\\_level\\_landing.jsp](http://japan.renesas.com/products/tools/evaluation_software/evaluation_software_mid_level_landing.jsp) である。尚、ダウンロードにはルネサスエレクトロニクスにあらかじめ登録しておく必要がある。

## 4 開発環境のセットアップ

### 4.1 HEW のインストール

ルネサスエレクトロニクスの Web サイトからダウンロードした評価版ソフトウェアをインストール。 *shv9400\_ev.exe* をダブルクリックして、インストールマネージャが起動したら、「標準インストール (推奨)」をクリック。オートアップデートユーティリティにもチェックを入れておく。これで、ツールチェーンとシミュレータデバッグがインストールされる。

### 4.2 FDT のインストール

こちらも、ルネサスエレクトロニクスの Web サイトからダウンロードした評価版ソフトウェアをインストール。 *fdtv407r00.exe* をダブルクリックして、インストーラ起動後、適当にセットアップする。

### 4.3 FT232RL のドライバインストール

FDTI 社の Web サイトから、仮想 COM ポート用ドライバを入手して展開しておく。 URL は <http://www.ftdichip.com/Drivers/VCP.htm> である。Windows x86(32bit) の最新版は、2.08.14 である。ファイル名は、*CDM20814-WHQL-Certified.zip* である。そのまま展開してしまうと、カレントディレクトリにぶちまけてしまうので、フォルダを指定して展開しておく。

FT232RL モジュールと USB Cable で接続すると、ドライバのインストールを求められるので、先ほど展開しておいたドライバを指定する。

## 5 SH7125 向けサポートプログラム

このサポートプログラムのライセンスは GPL である。

### 5.1 内蔵モジュール向け

SH7125 に内蔵しているモジュールを簡単に使うため、サポートプログラムを用意した。

- SCI sci1.c, sci1.h : シリアル入出力 (文字入出力, 文字列出力, HEX 入出力)
- CMT cmt.c, cmt.h : コンペアマッチタイマ (周期タイマと時間待ちタイマ)
- MTU mtu.c, mtu.h :

又、割り込みを使用する場合は、割り込みハンドラの定義が必要。今回は、HEW で用意される *intprg.c* に直接組み込む。

ソースコード 1 割り込みハンドラの記述例

```
// 88 MTU2.0 TGIA0
void INT_MTU2_0_TGIA0(void);
#pragma interrupt INT_MTU2_0_TGIA0
void INT_MTU2_0_TGIA0(void) {

    MTU2.TSTR.BYTE &= ~0x07;           /* カウンタ0,1,2停止 */
    MTU20.TSR.BIT.TGFA = 0;          /* TGFAクリア */
}

// 184 CMT0 CMI0
void INT_CMT0_CMI0(void);
#pragma interrupt INT_CMT0_CMI0
void INT_CMT0_CMI0(void) {

    CMT0.CMCSR.BIT.CMF=0;
    tcount++;
    Interrupt_cmt0();
}

// 185 Reserved
```

### 5.2 LCD モジュール向け

キャラクタタイプの LCD モジュールのうち、コントローラが HD44780 コンパチブルな物がそのまま使用可能。リソース削減のため、4bit バス接続としている。4 倍角数字表示機能も備える。

- LCD モジュール制御 lcd.c, lcd.h

### 5.3 TYMON

メモリのダンプ, 変更, モトローラ S 形式のプログラムのローディング等の機能を備えたモニタープログラムを H8 向けから SH2 向けに移植した.

- TYMON copyright.c, copyright.h, motosload.c, motosload.h, tymon.c, tymon.h, version.h, COPYING.txt, readme.txt

## 6 サンプルプログラム

サンプルプログラムの作成の手順を示す。

### 6.1 プロジェクトの作成

High-performance Embedded Workshop をスタートメニューで選択し, HEW を起動する. 図 5 の様に, ようこそメニューが起動するので, 新規プロジェクトワークスペースの作成 (C) を選択し, OK を押す.

次に, 図 6 の様に, ワークスペース名を決定する.

次に, 図 7 の様に, 使用するデバイスを指定する.

次に, 図 8 の様に生成ファイルを指定する.

次に, 図 9 の様に標準ライブラリを指定する. デフォルトのまま.

次に, 図 10 の様にスタック領域を指定する. デフォルトのまま.

次に, 図 11 の様にベクタの定義を指定する. デフォルトのまま.

次に, 図 12 の様にデバッガを指定する. デフォルトのまま.

次に, 図 13 では生成ファイルが表示される. 完了をクリックする.

最後に, 図 14 ではプロジェクト概要が表示される. OK をクリックする.

ワークスペースの前準備が完了したら, 図 15 の様な画面で待機している.

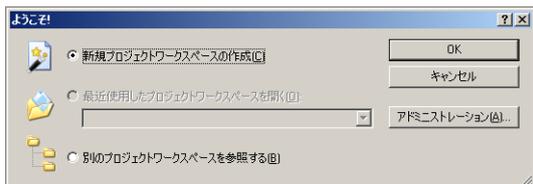


図 5 ワークスペースの作成



図 6 新規ワークスペーススペース

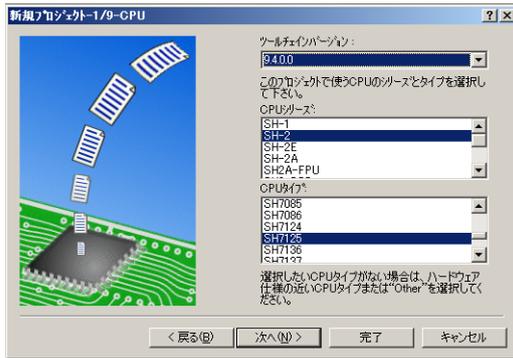


図 7 使用デバイス



図 8 生成ファイル



図 9 標準ライブラリ



図 10 スタック領域



図 11 ベクタの定義

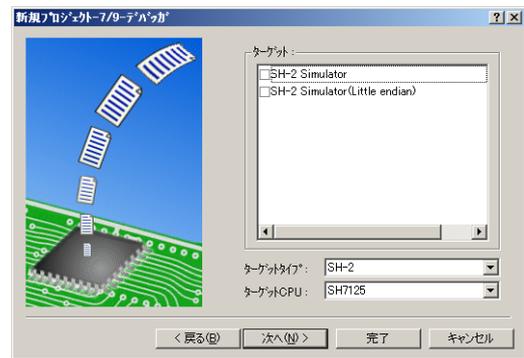


図 12 デバッグ指定

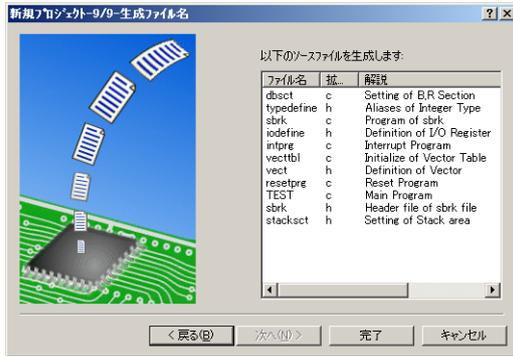


図 13 生成ファイル



図 14 プロジェクト概要

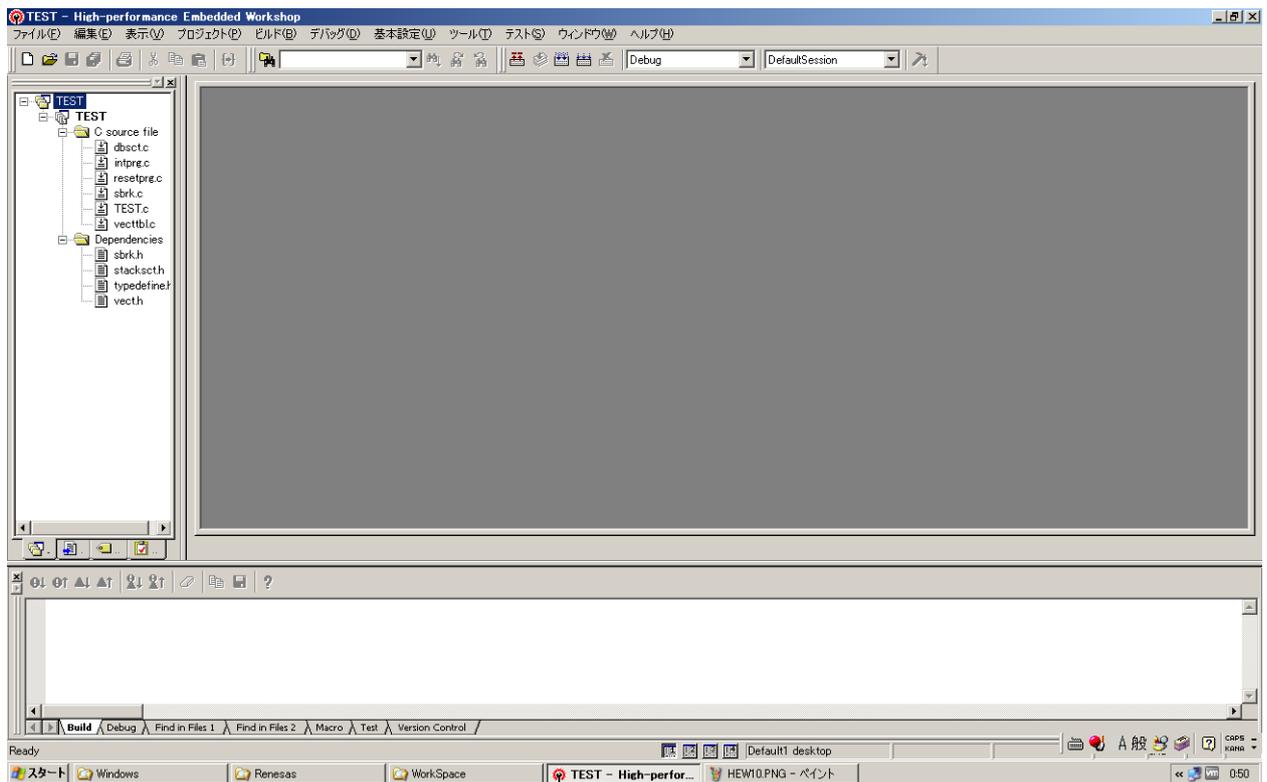


図 15 HEW 全景

## 6.2 既存プログラムの追加

無事にプロジェクトが出来上がったら、次はサポートプログラムをプロジェクトに追加する。図6の例では、*C:\WorkSpace\TEST\TEST* にプログラムが置かれている。

*C:\WorkSpace\TEST\TEST* のディレクトリ

<i>[.]</i>	<i>[..]</i>	<i>dbstc.c</i>
<i>[Debug]</i>	<i>DefaultSession.hsf</i>	<i>intprg.c</i>
<i>iodefine.h</i>	<i>Readme.txt</i>	<i>[Release]</i>
<i>resetprg.c</i>	<i>sbrk.c</i>	<i>sbrk.h</i>
<i>stacksct.h</i>	<i>TEST.c</i>	<i>TEST.hwp</i>
<i>TEST.nav</i>	<i>TEST.pgs</i>	<i>TEST.tps</i>
<i>typedefine.h</i>	<i>vect.h</i>	<i>vecttbl.c</i>

ここに必要な既存コードをコピーしておく。今回の例では、配布アーカイブ中の *support* を丸ごとコピーする。

次に、メニューのプロジェクト (P) → ファイルの追加 (A) と進み、図??の如く次のファイルを追加しておく。

- *iodefine.h*
- *support\cmt.c, cmt.h*
- *support\common.c, common.h*
- *support\lcd.c, lcd.h*
- *support\mtu.c, mtu.h*
- *support\sci1.c, sci1.h*
- *support\TYMON\copyright.c copyright.h*
- *support\TYMON\mosload.c mosload.h*
- *support\TYMON\tymon.c tymon.h*



図16 HEW プロジェクトへのファイル追加

### 6.3 main() の記述

RESET が解除されると真っ先に実行されるのが、*main()* である。この中に必要な処理を記述する。プロジェクトを新規作成したばかりの *TEST.C* はソースコード 2 の様になっている。

ソースコード 2 TEST.C のひな形

```
/******  
/*                                                                 */  
/* FILE      :TEST.c                                             */  
/* DATE      :Sun, May 22, 2011                                   */  
/* DESCRIPTION :Main Program                                     */  
/* CPU TYPE   :SH7125                                           */  
/*                                                                 */  
/* This file is generated by Renesas Project Generator (Ver.4.18). */  
/* NOTE:THIS IS A TYPICAL EXAMPLE.                               */  
/******  
  
//#include "typedefine.h"  
#ifdef __cplusplus  
//#include <ios> // Remove the comment when you use ios  
//_SINT ios_base::Init::init_cnt; // Remove the comment when you use ios  
#endif  
  
void main(void);  
#ifdef __cplusplus  
extern "C" {  
void abort(void);  
}  
#endif  
  
void main(void)  
{  
  
}  
  
#ifdef __cplusplus  
void abort(void)  
{  
  
}  
#endif
```

## 6.4 割り込みルーチン

割り込みを使用する場合は、割り込みベクタに割り込みルーチンを登録する必要がある。配布アーカイブ中の `support\intprg_sample.c` を参考にしたい。今回の例では、ソースリスト 1 の様に `88 MTU2_0`, `184 CMT0` の 2 つの割り込みを使用する。行頭に `#include "mtu.h" #include "cmt.h"` を忘れない事。

### 6.4.1 サンプルプログラム

ビルド可能なサンプルプログラムも同梱してある。このサンプルプログラムでは、`SCI1` を使ったデバックシリアル、`CMT` を使った周期タイマ割り込みで秒数を計時する `LCD` 表示タイマ、`TYMON` を使ったモニタデバッグ、おまけに `MTU` を使ったビープ音をインプリメントしてある。

## 6.5 ビルド

ビルドの前に、ヘッダファイルのインクルードパスを設定しておく。メニューのビルド (B) → *Super H RISC engine Standard Toolchain...* をクリック。コンパイラタブのカテゴリ: ソース → オプション項目 (S): インクルードファイルディレクトリを選択後、次の 3 ディレクトリを指定する。図 17 を参照。

- `$(PROJDIR)`
- `$(PROJDIR)\support`
- `$(PROJDIR)\support\TYMON`

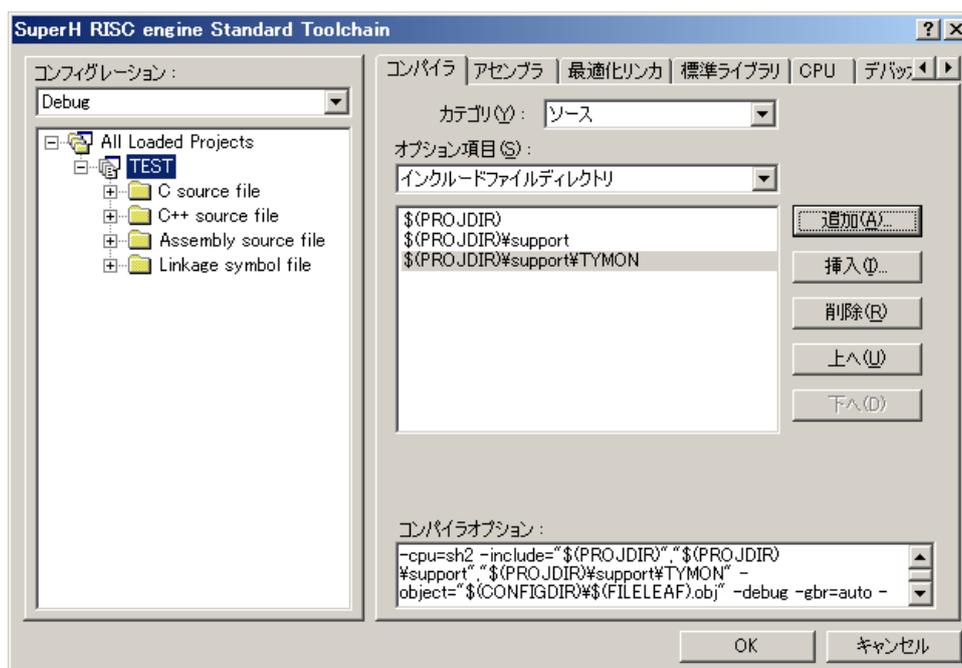


図 17 HEW インクルードファイルのパス追加

## 7 プログラム書き込み

いよいよ実機の *FlashMemory* への書き込みだ。実際の書き込みを始める前に、プロジェクトワークスペースを作成する。その後、実機への書き込みを行う。

### 7.1 プロジェクトワークスペースの作成

実機に接続してから、*Flash Development Toolkit* をスタートメニューで選択し、*FDT* 起動する。この時、*SH7125F* の *FWE* 端子は *H* レベル、*MD1* 端子は *L* レベルに設定しておく。添付の回路図では *SW102 1-8 OFF, 2-7 ON* とする。

図 18 の様に、ようこそ!メニューが起動するので、新規プロジェクトワークスペースの作成 (*C*) を選択し、*OK* を押す。尚、このプロジェクトワークスペースは *HEW* とは無関係である点に注意のこと。

次に、図 19 の様にワークスペース名を決定する。次へ (*N*) をクリックする。

次に、図 20 の様にデバイスを決定する。ここでは *SH7125F* を選択する。次へ (*N*) をクリックする。

次に、図 21 の様に通信ポートを決定する。この通信ポートは *FT232RL* のポートを選択しておくこと。*FDT* 起動前に *USB* 接続していない場合、リストに現れないので、後で設定しなおす。次へ (*N*) をクリックする。

次に、図 22 の様に実機に接続を促すダイアログが出る。*OK* をクリックする。

すると、図 23 の様に通信を開始する。図 24 の様にデバイス選択ダイアログが開くので、*R5F7125* を選び、*OK* をクリックする。

次に、図 25 の様に汎用でバイスの確認が終了する。*OK* をクリックする。

次に、図 26 の様に周波数の設定を行う。周辺クロックの通倍比 (*CKP*) を 2 に変更しておく。次へ (*N*) をクリックする。

次に、図 27 の様に接続タイプの設定を行う。デフォルトのまま *OK* である。次へ (*N*) をクリックする。

最後に、図 28 の様に書き込みオプションの設定を行う。*Readback Verification* は *Yes* に変更しておく。完了をクリックする。その後、プロジェクトをファイル (*F*) → 上書き保存 (*S*) をクリックして、保存しておく。

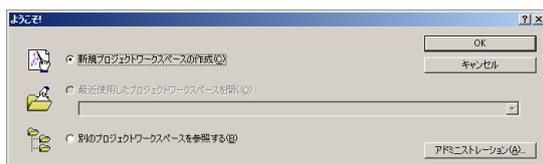


図 18 ようこそ!



図 19 ワークスペース名指定

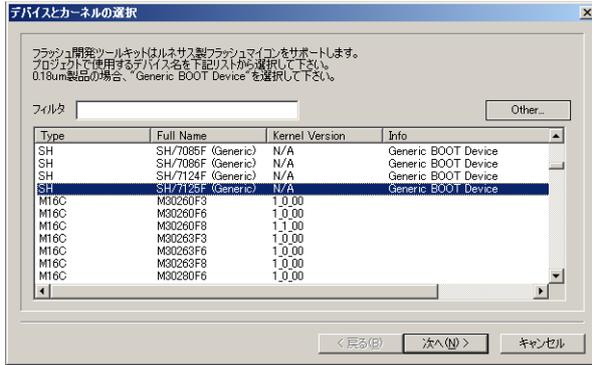


図 20 デバイス選択



図 21 通信ポート指定



図 22 実機接続指示



図 23 通信中



図 24 デバイス指定



図 25 デバイス確認完了

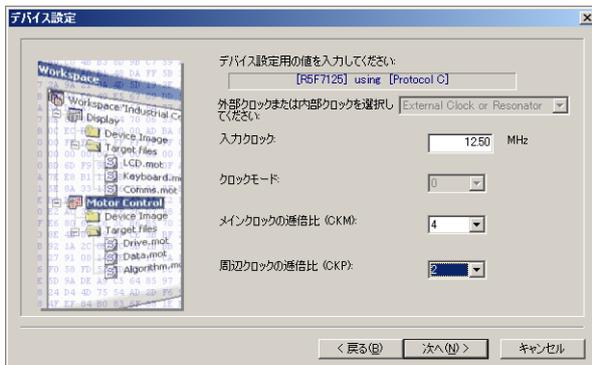


図 26 クロック周波数指定

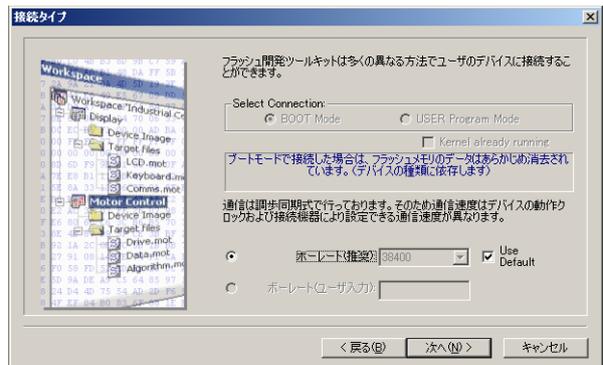


図 27 接続タイプ指定

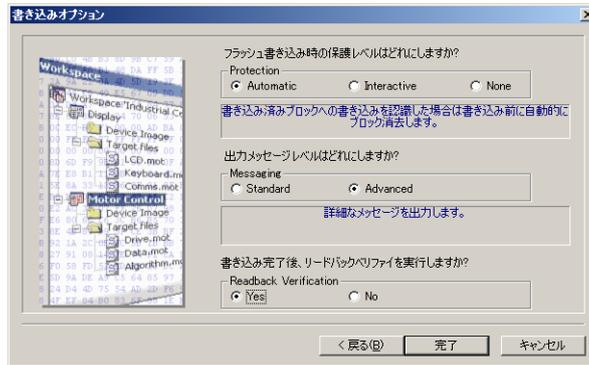


図 28 書き込みオプション設定

## 7.2 FlashMemory への書き込み

書き込みしたいファイルをプロジェクトに追加する。今回、書き込むファイル形式はモトローラ  $S(*.MOT)$  である。このファイルは HEW のプロジェクトの *Debug* ディレクトリにある。図 29 の様に指定する。

書き込む際にまず、図 30 のプロジェクトウィンドウの *S-Record Files* にある先ほど指定したモトローラ  $S$  ファイルをダブルクリックする。するとその内容がオープンされる。

次に、メニューのデバイス ( $D$ ) → デバイスとの接続 ( $C$ ) をクリックする。うまくいけばログウィンドウに接続が成功しましたと表示される。

次に、メニューのデバイス ( $D$ ) → 対象ファイルのダウンロード ( $L$ ) をクリックする。うまくいけばログウィンドウに図 31 の様に書き込みが完了しました、ベリファイ成功しましたと表示される。

最後に、メニューのデバイス ( $D$ ) → デバイスとの切断 ( $D$ ) をクリックする。以上で書き込み終了である。

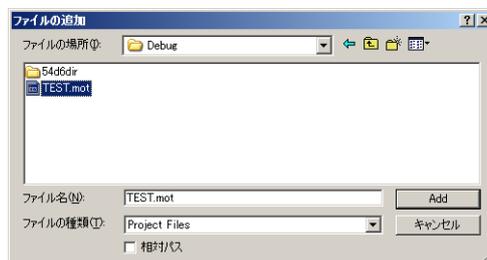


図 29 FDT プロジェクトへのファイル追加



図 30 ファイルオープン



図 31 書き込み・ベリファイ完了

## 8 実行!

さて、書き込んだプログラムを実行するには、SH7125FのFWE端子はLレベル、MD1端子はHレベルに設定を変更する。その後、リセットすると書き込んだユーザプログラムが動き始める。

```

——— Serial Menu ———
x.. TYMON
1.. LCD Banner
d.. DekaNum Display Check
i.. Beep doremi check
Select?x
Welcome to TY Monitor!
Dist.Version : TYmonitor Version a6, 01/03/2000(Mon)
tymon.c Revision : Rev.01 2010/05/14 T.YAsui, Joe for SH7260 Series

=>
C
Copyright (C) 1999-2010 T.YAsui(yasui.takao@nifty.com)
Copyright (C) 1999 Joe(ohkubo@ist.aichi-pu.ac.jp)

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published
by the Free Software Foundation; either version 2 of the License,
or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

=>
?
Dist. Version : TYmonitor Version a6, 01/03/2000(Mon)
tymon.c Revision : Rev.01 2010/05/14 T.YAsui, Joe for SH7260 Series
? | This Help Message
C | Display copyright
m address:8[;l,w,b] | Edit Memory
d address:8[;l,w,b] | Dump Memory
f top_address:8 end_address:8 data:2 | Fill Memory
g address:8 | Jump
l | Program Load(Motorola S format, SCI1 port)

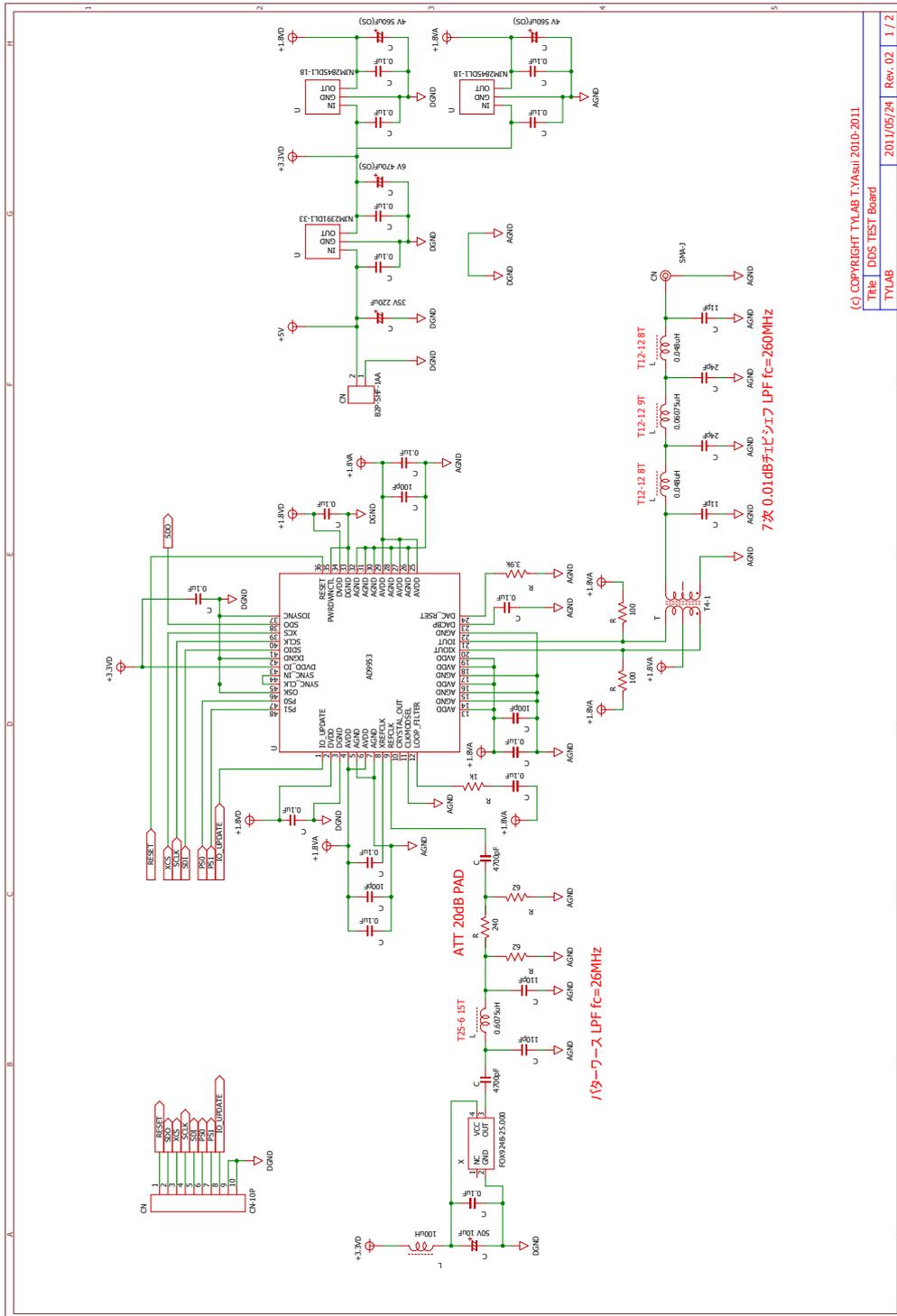
=>
d 00000000

```

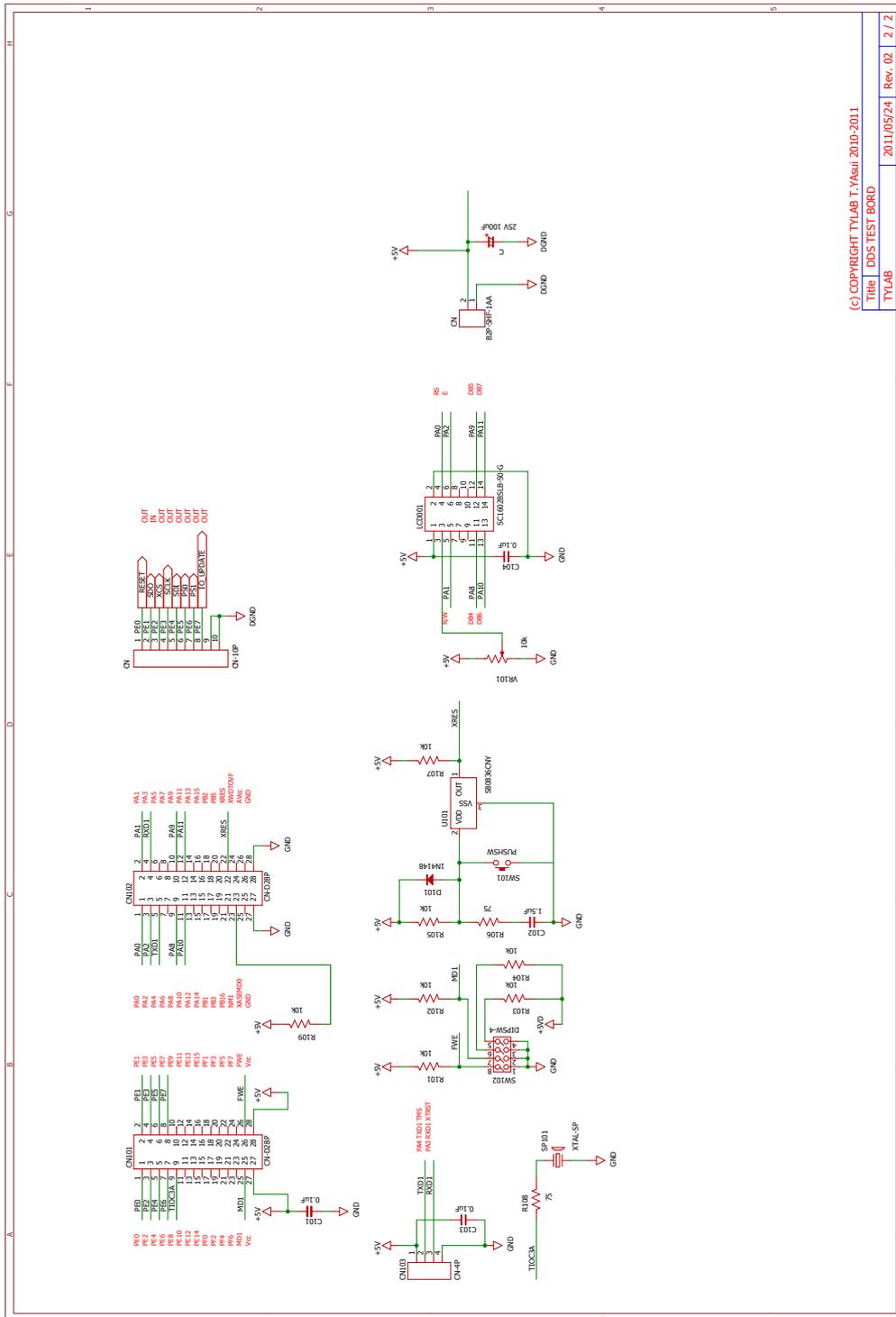
8 実行!

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	—— ASCII ——
00000000	: 00	00	08	00	FF	FF	C0	00	00	00	08	1E	FF	FF	C0	00	.....
00000010	: 00	00	04	00	00	00	05	C2	00	00	04	04	00	00	05	C2	.....
00000020	: 00	00	05	C2	00	00	04	08	00	00	05	C2	00	00	04	0C	.....
00000030	: 00	00	04	10	00	00	05	C2	00	00	05	C2	00	00	05	C2	.....
00000040	: 00	00	05	C2	.....												
00000050	: 00	00	05	C2	.....												
00000060	: 00	00	05	C2	.....												
00000070	: 00	00	05	C2	.....												
00000080	: 00	00	04	14	00	00	04	18	00	00	04	1C	00	00	04	20	.....
00000090	: 00	00	04	24	00	00	04	28	00	00	04	2C	00	00	04	30	... \ \$ . . . ( . . . , . . . 0
000000A0	: 00	00	04	34	00	00	04	38	00	00	04	3C	00	00	04	40	... 4 . . . 8 . . . < . . . @
000000B0	: 00	00	04	44	00	00	04	48	00	00	04	4C	00	00	04	50	... D . . . H . . . L . . . P
000000C0	: 00	00	04	54	00	00	04	58	00	00	04	5C	00	00	04	60	... T . . . X . . . \ . . . '
000000D0	: 00	00	04	64	00	00	04	68	00	00	04	6C	00	00	04	70	... d . . . h . . . l . . . p
000000E0	: 00	00	04	74	00	00	04	78	00	00	04	7C	00	00	04	80	... t . . . x . . .   . . .
000000F0	: 00	00	04	84	00	00	04	88	00	00	04	8C	00	00	04	90	.....

⇒



(a) AD9953 部



(b) SH7125 部

図 32 回路図

## 参考文献

- [1] ルネサスエレクトロニクス株式会社 『SH7125 グループ,SH7124 グループ ハードウェアマニュアル Rev.5.00』 (ルネサスエレクトロニクス株式会社 2009.03)