EPSON
EXCEED YOUR VISION

S1C17 Family Application Note

S1C17651/653
周辺回路サンプルソフトウェア

セイコーエプソン株式会社

Rev.1.0

評価ボード・キット、開発ツールご使用上の注意事項

1. 本評価ボード・キット、開発ツールは、お客様での技術的評価、動作の確認および開発のみに用いられることを想定し設計されています。それらの技術評価・開発等の目的以外には使用しないで下さい。本品は、完成品に対する設計品質に適合していません。

2. 本評価ボード・キット、開発ツールは、電子エンジニア向けであり、消費者向け製品ではありません。お客様において、適切な使用と安全に配慮願います。弊社は、本品を用いることで発生する損害や火災に対し、いかなる責も負いかねます。通常の使用においても、異常がある場合は使用を中止して下さい。

3. 本評価ボード・キット、開発ツールに用いられる部品は、予告無く変更されることがあります。

本資料のご使用につきましては、次の点にご留意願います。

本資料の内容については、予告無く変更することがあります。

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。

2. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これらに起因する第三者の知的財産権およびその他の権利侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の知的財産権およびその他の権利の実施権の許諾を行うものではありません。

3. 特性値の数値の大小は、数直線上の大小関係で表しています。

4. 製品および弊社が提供する技術を輸出等するにあたっては「外国為替および外国貿易法」を遵守し、当該法令の定める手続きが必要です。大量破壊兵器の開発等およびその他の軍事用途に使用する目的をもって製品および弊社が提供する技術を費消、再販売または輸出等しないでください。

5. 本資料に掲載されている製品は、生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。

6. 本資料に掲載されている会社名、商品名は、各社の商標または登録商標です。

目　次

1. 概要

本マニュアルは S1C17651、S1C17653 向けのサンプルソフトウェアの使い方とサンプルソフトウェアの動作について記載しています。

S1C17651/653 サンプルソフトウェアは S1C17651/653 マイコンに内蔵されている各周辺回路の使用例を示すことを目的としています。

S1C17651/653 サンプルソフトウェアはインストールの簡便さなどから、機種毎に提供していますが、各機能の基本的な動作は同じです。

各機種情報、各テクニカルマニュアル、S5U1C17001C Manual と合わせてご覧下さい。

1.1 動作環境

S1C17651/653 サンプルソフトウェアを動作させるにあたり、以下の機材をご用意下さい。

- S1C17651/653 の実装されたボード
- S5U1C17001H（以下 ICDmini とします。）
- S5U1C17001C（以下 GNU17 とします。）

注　本サンプルソフトウェアは、GNU17v2.2.0 で動作確認を行っています。

2. サンプルソフトウェア説明

本章では S1C17651/653 サンプルソフトウェアのファイル構成と実行方法を記載します。

S1C17651/653 サンプルソフトウェアは各周辺回路の動作を確認する“サンプルソフトウェア”と、各周辺回路のサンプルドライバである“サンプルドライバ”から成ります。

2.1 ディレクトリ構成及びファイル構成

以下に S1C17651/653 サンプルソフトウェアのディレクトリ構成を示します。

```
[s1c17651_653_xxx]
    |
    ├GNU17プロジェクトファイルなど
    ├…
    ├…
    |
    |
    ├── [inc]
    |    ├── c17651_653_reg.h       init.h
    |    …
    |    └──[reg]
    |          ├── clg_reg.h        ct_reg.h
    |          ├── flashc_reg.h     itc_reg.h
    |          ├── lcd_reg.h        misc_reg.h
    |          ├── mux_reg.h        port_reg.h
    |          ├── rtc_reg.h        snd_reg.h
    |          ├── spi_reg.h        svd_reg.h
    |          ├── t8_reg.h         t16a2_reg.h
    |          ├── tr_reg.h         uart_reg.h
    |          └── vd1_reg.h        wdt_reg.h
    |
    |
    └──  [src]
         ├── boot.c
         ├── init.c
         …
         └── main.c
```

図 2.1 S1C17651/653 サンプルソフトウェアディレクトリ構成図

（1）s1c17651_653_xxx ディレクトリ

本ディレクトリには GNU17 のプロジェクトに関係するファイルが格納されているディレクトリが配置されています。

（2）inc ディレクトリ

機種に依存する情報を定義したファイル及び、周辺回路毎のサンプルソフトウェア内で使用する定数等を定義したヘッダファイルが配置してあります。

- 対象機種のレジスタアドレスなどが定義されたヘッダファイル（c17651_653_reg.h）
- 周辺回路毎のヘッダファイル（init.h など）

（3）reg ディレクトリ

周辺回路毎のビットアサインなどを定義したヘッダファイが配置してあります。

- 周辺回路毎のビットアサインなどを定義されたヘッダファイル（clg_reg.h など）

（4）src ディレクトリ

マイコンの初期化、処理周辺回路毎のサンプルソフトウェア及び、サンプルドライバが配置してあります。

- 初期化処理のファイル（boot.c）
- 周辺回路毎のサンプルソフトウェアファイル（main.c）
- 周辺回路毎のサンプルドライバファイル（init.c など）

2.2 実行方法

以下の手順で S1C17651/653 サンプルソフトウェアを実行して下さい。

（1）プロジェクトをインポート

GNU17 を起動して S1C17651/653 サンプルソフトウェアのプロジェクトをインポートして下さい。

プロジェクトインポート方法の詳細につきましては S5U1C17001C Manual の“3 ソフトウェア開発手順”を参照して下さい。

（2）プロジェクトをビルド

GNU17 で S1C176xx プロジェクトをビルドして下さい。

ビルド方法の詳細につきましては S5U1C17001C Manual の“5 GNU17 IDE”を参照して下さい。

（3）ICDmini を接続

ICDmini を PC、開発ボードに接続して開発ボードの電源を投入して下さい。

（4）デバッガによるプログラムのロードと実行

GNU17 の“デバッグ”ボタンを押下してデバッグを開始して下さい。

プログラムが S1C17651/653 にロードされプログラムが起動します。

デバッガ使用方法の詳細につきましては S5U1C17001C Manual の“10 デバッガ”を参照して下さい。

3. サンプルソフトウェア機能詳細

本章では S1C17651/653 サンプルソフトウェアの機能詳細について記載します。

3.1 入出力ポート（P）

3.1.1 サンプルソフトウェア仕様

本サンプルソフトウェアは入出力ポートを使用して以下の動作を行います。

- ポートを入力割り込みに設定し、入力信号が Low レベルになったことを検出する。
- ポートを出力に設定し、High レベルや Low レベルの信号を出力する。

使用するポート設定とポート名は以下の通りです。

表 3.1.1 入出力ポート設定一覧

設定	ポート名
入力割り込みポート	P00 P01 P02 P03
出力ポート	P04 P05 P06 P07

3.1.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。
本サンプルプログラムはマイコンの各ポートを以下のように接続してご使用下さい。

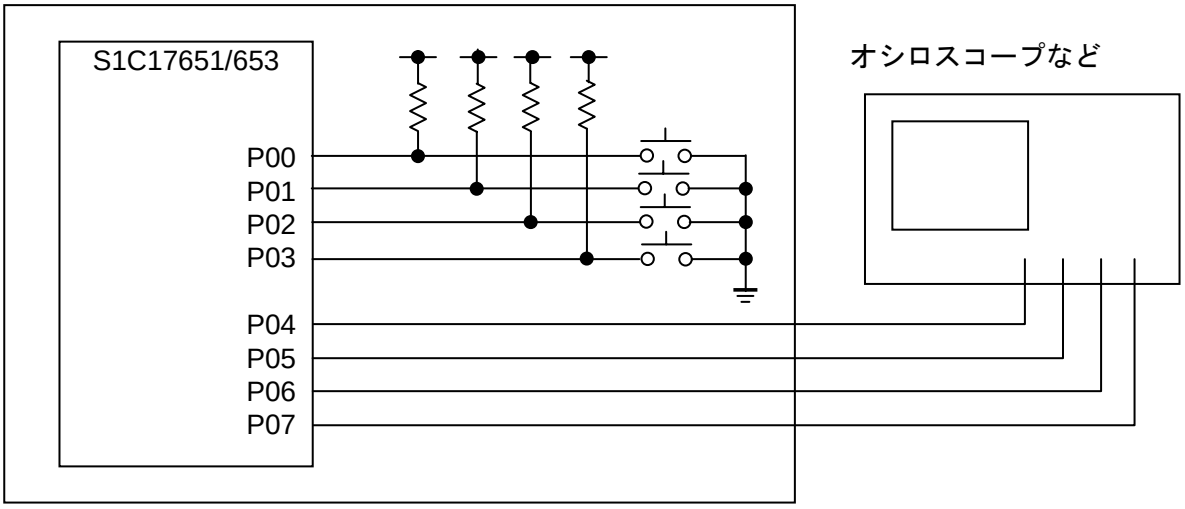

図 3.1.1 入出力ポート（P）サンプルソフトウェアハードウエア接続図

3.1.3 動作概要

1. P00 ポートの入力信号を Low レベルにすると Simulated I/O に「P00 Interrupt」と表示し、P04 ポートの出力を反転させます。（High レベルであれば Low レベル、Low レベルであれば High レベルにします。）
2. P01 ポートの入力信号を Low レベルにすると Simulated I/O に「P01 Interrupt」と表示し、P05 ポートの出力を反転させます。（High レベルであれば Low レベル、Low レベルであれば High レベルにします。）
3. P02 ポートの入力信号を Low レベルにすると Simulated I/O に「P02 Interrupt」と表示し、P06 ポートの出力を反転させます。（High レベルであれば Low レベル、Low レベルであれば High レベルにします。）
4. P03 ポートの入力信号を Low レベルにすると Simulated I/O に「P03 Interrupt」と表示し、P07 ポートの出力を反転させます。（High レベルであれば Low レベル、Low レベルであれば High レベルにします。）

```
<<< Port demonstration start >>>
*** P00 Interrupt ***
*** P01 Interrupt ***
*** P02 Interrupt ***
*** P03 Interrupt ***
<<< Port demonstration finish >>>
```

図 3.1.2 入出力ポートサンプルプログラム画面表示例

3.2 クロックジェネレータ（CLG）

3.2.1 サンプルソフトウェア仕様

発振回路サンプルプログラムは OSC を使用して以下の動作を行います。

- OSC1 の発振と停止を行う。(本サンプルプログラムでは OSC1A を選択。以後「OSC1」と表記する。)
- OSC3A の発振と停止を行う。
- OSC3B の発振と停止を行う。
- システムクロックを OSC3B から OSC3A へ切り替える。
- システムクロックを OSC3A から OSC1 へ切り替える。
- システムクロックを OSC1 から OSC3B へ切り替える。

3.2.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1A と OSC3A に水晶振動子またはセラミック振動子が接続されている状態で動作します。
振動子の接続方法につきましては、S1C17651/S1C17653 テクニカルマニュアル「クロックジェネレータ (CLG)」をご参照下さい。

3.2.3 動作概要

本サンプルプログラムは OSC3B を使用した状態で動作を開始します。
1. 一定間隔で Simulated I/O に「1」、「2」、「3」…、「9」と表示した後、OSC3A の発振を開始し、システムクロックを OSC3B から OSC3A に切り替え、OSC3B を停止します。
2. 一定間隔で Simulated I/O に「1」、「2」、「3」…、「9」と表示した後、OSC1 の発振を開始し、システムクロックを OSC3A から OSC1 に切り替え、OSC3A を停止します。
3. 一定間隔で Simulated I/O に「1」、「2」、「3」…、「9」と表示した後、OSC3B の発振を開始し、システムクロックを OSC1 から OSC3B に切り替え、OSC1 を停止します。
4. 一定間隔で Simulated I/O に「1」、「2」、「3」…、「9」と表示した後、サンプルプログラムを終了します。

```
<<< CLG(OSC) demonstration start >>>
OSC3B *** 1 ***
OSC3B *** 2 ***
…
OSC3B *** 9 ***
*** Change from OSC3B to OSC3A ***
OSC3A *** 1 ***
OSC3 A*** 2 ***
…
OSC3A*** 9 ***
*** Change from OSC3A to OSC1 ***
OSC1 *** 1 ***
OSC1*** 2 ***
…
OSC1*** 9 ***
*** Change from OSC1 to OSC3B ***
OSC3B *** 1 ***
OSC3B *** 2 ***
…
OSC3B *** 9 ***
<<< CLG(OSC) demonstration finish >>>
```

図 3.2.1　CLG_OSC サンプルプログラム画面表示例

3.3 8 ビットタイマ（T8）

3.3.1 サンプルソフトウェア仕様

8 ビットタイマサンプルプログラムは 8 ビットタイマを使用して以下の動作を行います。

- 8 ビットタイマ割り込みを発生させ、タイマのカウンター値を取得する。
- 割り込み待機時は CPU を halt モードにして消費電力を低減する。

3.3.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。

3.3.3 動作概要

1. 8 ビットタイマ割り込みを開始し、CPU を Halt 状態にします。
2. 8 ビットタイマ割り込みが発生すると CPU は Halt 状態が解除します。
3. 8 ビットタイマのカウンターデータを内部変数に保存して再び CPU を Halt 状態にします。
4. 8 ビットタイマ割り込みが 10 回発生すると 8 ビットタイマを停止します。
5. 各割り込みが発生したときのカウンターデータ値を Simulated I/O に表示し、サンプルプログラムを終了します。

```
<<< T8 timer demonstration start >>>
*** T8 interrupt 1 time, count data at this time : 128 ***
*** T8 interrupt 2 time, count data at this time : 128 ***
*** T8 interrupt 3 time, count data at this time : 128 ***
*** T8 interrupt 4 time, count data at this time : 128 ***
…
*** T8 interrupt 10 time, count data at this time : 128 ***
<<< T8 timer demonstration finish >>>
```

図 3.3.1 8 ビットタイマサンプルプログラム画面表示例

3.4 T16 ビット PWM タイマ（T16A2）

3.4.1 サンプルソフトウェア仕様

T16 ビット PWM タイマサンプルプログラムは T16 ビット PWM タイマを使用して以下の動作を行います。

- T16 ビット PWM タイマコンペア A マッチ割り込みを発生させ、タイマのカウンター値を取得する。
- T16 ビット PWM タイマコンペア B マッチ割り込みを発生させ、タイマのカウンター値を取得する。
- TOUTA0、TOUTB0 端子に波形を出力する。
- 割り込み待機時は CPU を halt モードにして消費電力を低減する。

3.4.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。

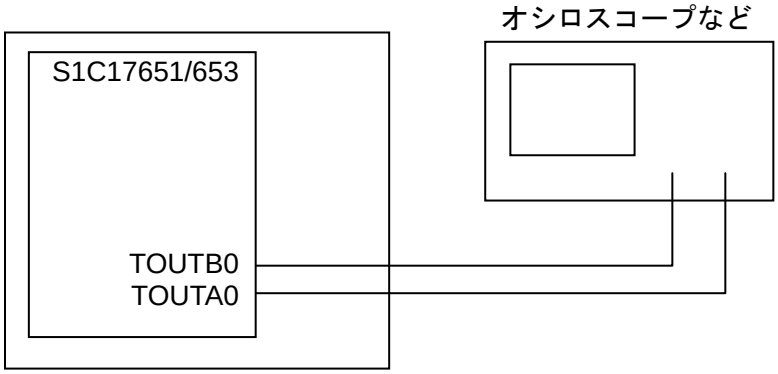

図 3.4.1 T16 ビット PWM タイマサンプルプログラムハードウェア接続図

3.4.3 動作概要

1. コンペア A マッチ割り込みとコンペア B マッチ割り込みを有効にし、T16 ビット PWM タイマを開始します。
2. コンペア A マッチ割り込みおよびコンペア B マッチ割り込みが発生すると、アップカウンターのカウンター値を取得します。
3. コンペア B マッチ割り込みが 5 回発生すると T16 ビット PWM タイマを停止し、割り込みの種類とカウンター値を Simulated I/O に表示してサンプルプログラムを終了します。

```
<<< T16 PWM timer demonstration start >>>
*** T16 PWM timer compare A interrupt :30 ***
*** T16 PWM timer compare B interrupt :61  ***
*** T16 PWM timer compare A interrupt :30 ***
…
*** T16 PWM timer compare B interrupt: 61 ***
<<< T16 PWM timer demonstration finish >>>
```

図 3.4.2　T16 ビット PWM タイマサンプルプログラム画面表示例

3.5 計時タイマ（CT）

3.5.1 サンプルソフトウェア仕様

計時タイマサンプルプログラムは計時タイマを使用して以下の動作を行います。

- 時計タイマ割り込みを発生させ、経過時間を算出する。
- 割り込み待機時は CPU を halt モードにして消費電力を低減する。

3.5.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路及び、OSC1B(32kHz)で動作します。

3.5.3 動作概要

1. 計時タイマを開始します。
2. 計時タイマ割り込みが発生すると、プログラム開始時からの経過時間を算出し、Simulated I/O に経過時間を表示します。
3. 計時タイマ割り込みが 10 回発生するとサンプルプログラムを終了します。

```
<<< Clock timer demonstration start >>>
*** 1.0 sec ***
*** 2.0 sec ***
*** 3.0 sec ***
…
*** 10.0 sec ***
<<< Clock timer demonstration finish >>>
```

図 3.5.1　計時タイマサンプルプログラム画面表示例

3.6 リアルタイムクロック（RTC）

3.6.1 サンプルソフトウェア仕様

リアルタイムクロックサンプルプログラムはリアルタイムクロックを使用して以下の動作を行います。

- リアルタイムクロックのサンプルプログラムメニューを表示する。
- リアルタイムクロックの時刻を取得する。
- リアルタイムクロックの時刻を設定する。
- リアルタイムクロックの割り込み回数を表示する。

3.6.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路及び、OSC1B(32kHz)で動作します。

3.6.3 動作概要

1. RTC サンプルプログラムのメニューを Simulated I/O に表示します。
2. Simulated I/O にてユーザーが時刻の設定を行います。
3. 設定後、RTC 割り込みの発生が 1 秒おきに発生し、Simulated I/O に RTC 割り込みの発生回数を表示します。
4. RTC 割り込みが 10 回発生すると RTC を停止します。

```
<<< Real Time Clock demonstration start >>>
> Input BCD format.
> 24H/12H (0/1):
0
> Hour (00 - 23) :
01
> Minute (00 - 59) :
23
> Second (00 - 59) :
45
interrupt count value = 1
interrupt count value = 2
.
.
.
interrupt count value = 10
<<< Real Time Clock demonstration finish >>>
```

図 3.6.1 リアルタイムクロックサンプルプログラム画面表示例

3.7 ウォッチドッグタイマ（WDT）

3.7.1 サンプルソフトウェア仕様

ウォッチドッグタイマサンプルプログラムはウォッチドッグタイマを使用して以下の動作を行います。

- ウォッチドッグタイマのクリアを確認する。
- ウォッチドッグタイマによる NMI 割り込みを発生させる。

3.7.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路及び、OSC1B(32kHz)で動作します。

3.7.3 動作概要

1. ウォッチドッグタイマと 8 ビットタイマを開始します。
2. 8 ビットタイマ割り込みが発生するとウォッチドッグタイマをクリアします。
3. 8 ビットタイマ割り込みが 10 回発生すると 8 ビットタイマを停止しします。
4. ウォッチドッグタイマによる NMI 割り込みが発生すると Simulated I/O にメッセージを表示し、サンプルプログラムを終了します。

```
<<< Watchdog timer demonstration start >>>
*** T8 timer : reset watchdog timer ***
*** T8 timer : reset watchdog timer ***
*** T8 timer : reset watchdog timer ***
...
*** T8 timer : reset watchdog timer ***
*** stop T8 timer ***
*** NMI occurred ***
<<< Watchdog timer demonstration finish >>>
```

図 3.7.1　ウォッチドッグタイマサンプルプログラム画面表示例

3.8 UART

3.8.1 サンプルソフトウェア仕様

OSC3A のクロックを使用した UART サンプルプログラムは UART を使用して以下の動作を行います。

- UART を使いデータを送信する。
- UART を使いデータを受信する。

3.8.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3A 内臓発振回路で動作します。
CPU の各ポートを以下のように接続してご使用下さい。

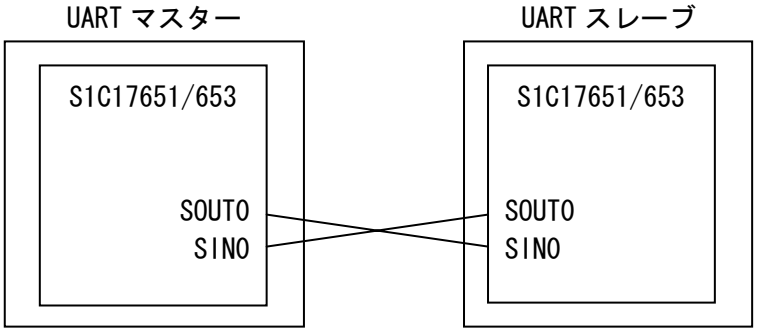

図 3.8.1 UART サンプルプログラムハードウェア接続図

3.8.3 動作概要

3.8.3.1 マスターサンプル動作概要

1. UART ポートを以下の値に初期化します。
   - 通信速度　　：最大のボーレート
   - データ長　　：8bit
   - ストップビット　：1bit
   - パリティ　　：無し
2. 計時タイマを 1 秒間隔で割り込みを発生させるように設定します。
3. 接続確認フラグ「0x7F」を受信するまで 1 秒おきに「0x7F」を送信し続けます。
4. 接続確認フラグ受信すると「0x7F」の送信を停止し、ASCII コード 0x21～0x7E のデータを UART ポートに送信します。
5. 送信後、UART ポートからデータを受信し、受信データを Simulated I/O に表示します。

```
<<< UART OSC3A demonstration start >>>
waiting connection.
connected
*** receive data ***
ABCDEFG…
<<< UART OSC3A demonstration finish >>>
```

図 3.8.2　UART（OSC3A）マスターサンプルプログラム画面表示例

3.8.3.2 スレーブサンプル動作概要

1. UART ポートを以下の値に初期化します。
   - 通信速度　　：最大のボーレート
   - データ長　　：8bit
   - ストップビット　：1bit
   - パリティ　　：無し
2. 接続確認フラグ「0x7F」を受信するまで待機します。
3. 接続確認フラグ受信すると「0x7F」を送信し、マスターからデータを受信します。
4. 受信後、UART ポートから ASCII コード 0x21～0x7E のデータを送信し、受信データを Simulated I/O に表示します。

```
<<< UART OSC3A demonstration start >>>
waiting connection.
connected
*** receive data ***
ABCDEFG…
<<< UART OSC3A demonstration finish >>>
```

図 3.8.3　UART（OSC3A）スレーブサンプルプログラム画面表示例

3.9 SPI

3.9.1 サンプルソフトウェア仕様

3.9.1.1 マスターサンプルソフトウェア仕様

SPI マスターサンプルプログラムは SPI マスターを使用して以下の動作を行います。

- 8byte のデータを SPI スレーブに送信する。
- 8byte のデータを SPI スレーブから受信する。
- 割り込み待機時は CPU を halt モードにして消費電力を低減する。

3.9.1.2 スレーブサンプルソフトウェア仕様

SPI スレーブサンプルプログラムは SPI スレーブを使用して以下の動作を行います。

- 8byte のデータを SPI マスターから受信する。
- 8byte のデータを SPI マスターに送信する。
- 割り込み待機時は CPU を halt モードにして消費電力を低減する。

3.9.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。
マイコンの各ポートを以下のように接続してご使用下さい。
本サンプルプログラムは SPI スレーブサンプルプログラムが動作している S1C176xx を SPI スレーブとして接続して使用して下さい。
各ポートは以下のように接続してご使用下さい。

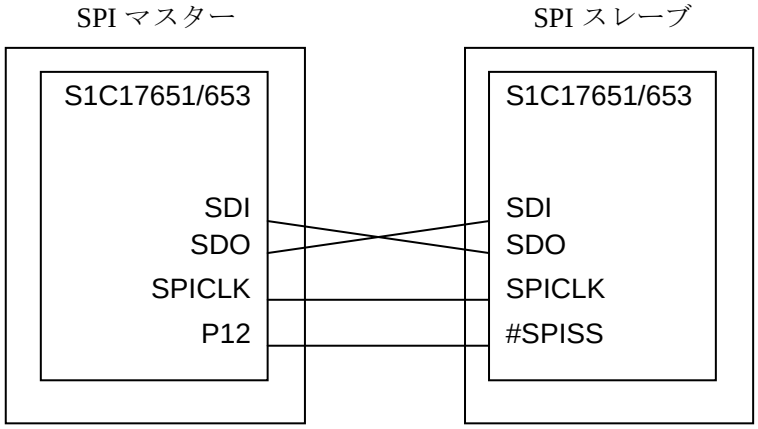

図 3.9.1　SPI マスター、スレーブサンプルプログラムハードウェア接続図

3.9.3 動作概要

3.9.3.1 マスターサンプル動作概要

1. SPI マスターの初期設定を行います。
2. SPI スレーブへ 8byte の ASCII データ「FROM MST」を送信します。
3. SPI スレーブへのデータ送信を終了すると、SPI スレーブへ SPI クロックを出力してデータを受信します。
4. SPI スレーブから受信したデータを Simulated I/O に表示し、サンプルプログラムを終了します。

```
<<< SPI master demonstration start >>>
Transmitted data : FROM MST
Received data : FROM SLV
<<< SPI master demonstration finish >>>
```

図 3.9.2 SPI マスターサンプルプログラム画面表示例

3.9.3.2 スレーブサンプル動作概要

1. SPI スレーブの初期設定を行います。
2. SPI マスターからのデータ受信を待ちます。
3. SPI マスターからデータを受信すると、受信データを Simulated I/O に表示します。
4. SPI マスターへ 8byte の ASCII データ「FROM SLV」を送信し、サンプルプログラムを終了します。

```
<<< SPI slave demonstration start >>>
Received data : FROM MST
Transmitted data : FROM SLV
<<< SPI slave demonstration finish >>>
```

図 3.9.3 SPI スレーブサンプルプログラム画面表示例

3.10 LCD ドライバー（LCD）

3.10.1 サンプルソフトウェア仕様

LCD ドライバーサンプルプログラムは LCD ドライバーを使用して以下の動作を行います。

- 通常表示における全点灯と全消灯を行う。
- 全点灯／全消灯機能における全点灯と全消灯を行う。

3.10.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。
マイコンの各ポートを以下のように接続してご使用下さい。

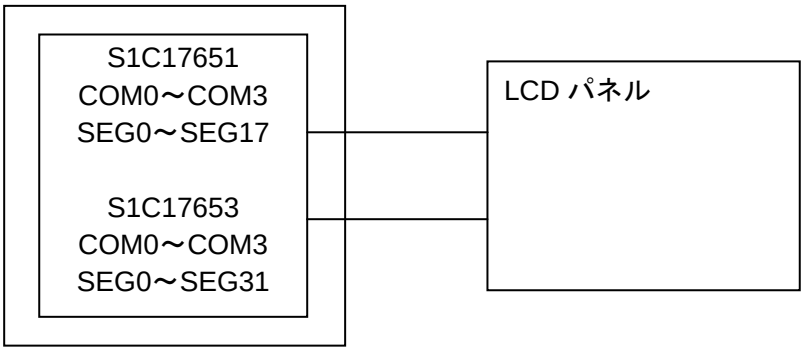

図 3.10.1　LCD ドライバーサンプルプログラムハードウェア接続図

3.10.3 動作概要

1. Simulated I/O に「on0」と入力して ENTER キーを押下すると、通常表示における LCD の全点灯を行います。
2. Simulated I/O に「on1」と入力して ENTER キーを押下すると、全点灯機能における LCD の全点灯を行います。
3. Simulated I/O に「off0」と入力して ENTER キーを押下すると、通常表示における LCD の全消灯を行います。
4. Simulated I/O に「off1」と入力して ENTER キーを押下すると、全消灯機能における LCD の全消灯を行います。

```
<<< LCD driver demonstration start >>>
on0
on1
off0
off1
exit
<<< LCD driver demonstration finish >>>
```

図 3.10.2　LCD ドライバーサンプルプログラム画面表示例

3.11 SLEEP/HALT モード切替

3.11.1 サンプルソフトウェア仕様

Sleep/Halt モード切替サンプルプログラムは以下の動作を行います。

- halt 命令を実行し CPU を halt モードにする。
- 8 ビットタイマ割り込みを使用し CPU の halt モードを解除する。
- sleep 命令を実行し CPU を sleep モードにする。
- ポート割り込みを使用し CPU の sleep モードを解除する。
- sleep 命令を実行し CPU を sleep モードにする。
- RTC 割り込みを使用し CPU の sleep モードを解除する。

3.11.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。
マイコンの各ポートを以下のように接続してご使用下さい。

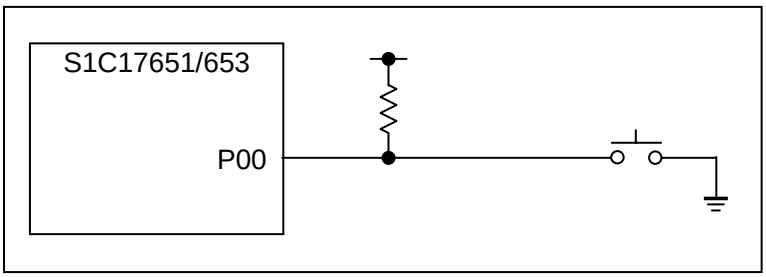

図 3.11.1　Sleep/Halt サンプルプログラム画面表示例

3.11.3 動作概要

1. 8 ビットタイマをセットし CPU を halt モードにします。
2. 8 ビットタイマ割り込みが発生すると halt モードを解除し、Simulated I/O にメッセージを表示します。
3. 8 ビットタイマ割り込みが 5 回発生すると CPU を sleep モードにします。
4. P00 ポートが Low レベルになると sleep モードを解除します。
5. RTC による割り込みが発生すると sleep モードを解除します。

```
<<< Sleep/halt demonstration start >>>
go to halt mode
return from halt mode
...
go to sleep mode(SW P00)
return from sleep mode
go to sleep mode(RTC)
return from sleep mode
<<< Sleep/halt demonstration finish >>>
```

図 3.11.2　Sleep/Halt モード切替サンプルプログラム画面表示例

3.12 サウンドジェネレータ（SND）

3.12.1 サンプルソフトウェア仕様

SND サンプルプログラムは SND を使用して以下の動作を行います。

- SND のエンベロープモードを使用して、周波数を上げてながら出力した波形を BZOUT 端子に出力します。

3.12.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路及び、OSC1A(32.768kHz)で動作します。

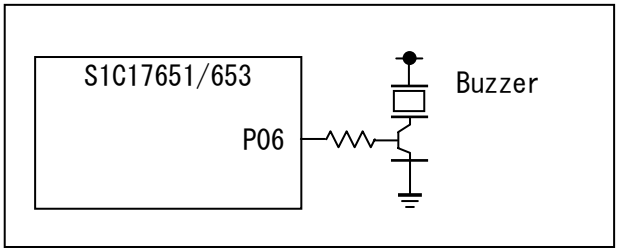

図 3.12.1 サウンドジェネレータサンプルプログラム ハードウェア接続図

3.12.3 動作概要

1. ブザー周波数を「4096.0Hz」に設定します。Simulated I/O に設定したブザー周波数を表示します。
2. Simulated I/O にブザー周波数を表示します。
3. エンベロープモードに設定後、ブザーを出力します。
4. エンベロープモードにより、デューティ比がレベル 1（最大）からレベル 8（最小）まで自動的に変化し、レベル 8 になるとブザーが停止します。
5. ブザー周波数を下げて設定します。Simulated I/O に設定したブザー周波数を表示します。
6. ブザーを 1 秒間出力して「2」に戻ります。この処理をブザー周波数が「1170.3Hz」になるまで繰り返します。
7. ブザー周波数を「1170.3Hz」に設定し、ブザーの出力が終了すると、サンプルプログラムを終了します

```
<<<Sound Generator demonstration start >>>
***Buzzer Frequncy : 4096.0Hz***
***Buzzer Frequncy : 3276.8Hz***
...
***Buzzer Frequncy : 1170.3Hz***
<<< Sound Generator demonstration finish >>>
```

図 3.12.2　サウンドジェネレータサンプルプログラム画面表示例

3.13 消費電流

3.13.1 サンプルソフトウェア仕様

電流測定サンプルプログラムは以下の状態の評価するためのプログラムです。

- OSC1A のみを発振させ、RTC および P クロックを停止させた状態でマイコンを Halt 状態にする。
- OSC1A のみを発振させ、RTC を起動、P クロックを停止させた状態でマイコンを Halt 状態にする。
- RTC を停止させた状態でマイコンを Sleep 状態にする。
- RTC を起動させた状態でマイコンを Sleep 状態にする。

3.13.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1A（32.768kHz）で動作します。

3.13.3 動作概要

本サンプルプログラムは、表 3.13.1 の定義を行うことにより動作を変更することができます。デフォルトでは、プロジェクトのシンボル定義にて HALT_OSC1A が設定されています。動作を変更する場合は、シンボル定義を変更してください。シンボル定義の変更は S5U1C17001C Manual を参照してください。

表 3.13.1 消費電流評価プログラム設定一覧

設定	動作
HALT_OSC1A	OSC1A のみを発振させ、RTC および P クロックを停止させた状態でマイコンを Halt 状態にする。
HALT_OSC1A_RTC	OSC1A のみを発振させ、RTC を起動、P クロックを停止させた状態でマイコンを Halt 状態にする。
SLEEP_ONLY	RTC を停止させた状態でマイコンを Sleep 状態にする。
SLEEP_RTC	RTC を起動させた状態でマイコンを Sleep 状態にする。

1. 測定を行うサンプルプログラムをあらかじめフラッシュメモリに書き込みます。
2. マイコンの電源を ON にし、サンプルプログラムを実行させてから電流値を測定します。

3.14 電源電圧検出回路（SVD）

3.14.1 サンプルソフトウェア仕様

リアルタイムクロックサンプルプログラムはリアルタイムクロックを使用して以下の動作を行います。

- OSC1A のみを発振させ、RTC 及び、P クロックを停止させた状態でマイコンを Halt 状態にする。
- リアルタイムクロックのサンプルプログラムメニューを表示する。
- リアルタイムクロックの時刻を取得する。
- リアルタイムクロックの時刻を設定する。
- リアルタイムクロックの割り込み回数を表示する。

3.14.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路で動作します。
本サンプルプログラムは以下のように接続してご使用下さい。

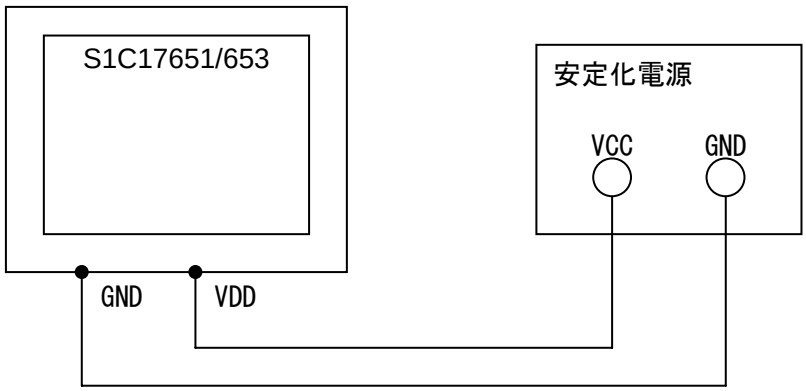

図 3.14.1　電源電圧検出回路ポートサンプルプログラムハードウェア接続図

3.14.3 動作概要

1. 比較電圧を 3.20V に設定します。
2. Simulated I/O に比較電圧を 3.20V に設定するメッセージを表示します。
3. 比較電圧との電圧を比較後、結果を Simulated I/O に表示します。比較電圧より小さい場合は「VDD is smaller than comparison voltage.」を表示し、比較電圧より大きい場合は「VDD is more than comparison voltage.」を表示します。

```
<<< SVD demonstration start >>>
Comparison Voltage:3.20V
VDD is smaller than comparison voltage.
Or
VDD is more than comparison voltage.
<<< SVD demonstration finish >>>
```

図 3.14.2 電源電圧検出回路サンプルプログラム画面表示例

3.15 MISC

3.15.1 動作概要

デバッグ時、サンプルプログラムの最初に以下の設定を行います。
1. フラッシュメモリのウェイトサイクル数を設定します。
2. デバッグモード時における PCLK で動作する周辺回路の状態を選択します。
3. デバッグモード時における PCLK 以外で動作する周辺回路の状態を選択します。
4. システムクロックを減速するギア比を設定します。
5. 内臓周辺モジュールへのクロックの供給を設定します。
6. 入出力端子機能の初期値を設定します。

ターゲット単体での動作では、「6」において、DCLK/DSIO/DST2 を P11/P12/P13 の入出力ポートに設定します。

3.16 論理暖急（TR）

3.16.1 サンプルソフトウェア仕様

論理緩急サンプルプログラムは以下の動作を行います。
計時タイマを起動させ、REGMON 端子に補正実行後のクロック(F256)を出力させる。

3.16.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC3B(2MHz)内臓発振回路及び、OSC1A(32.768kHz)で動作します。
本サンプルプログラムは以下のように接続してご使用下さい。

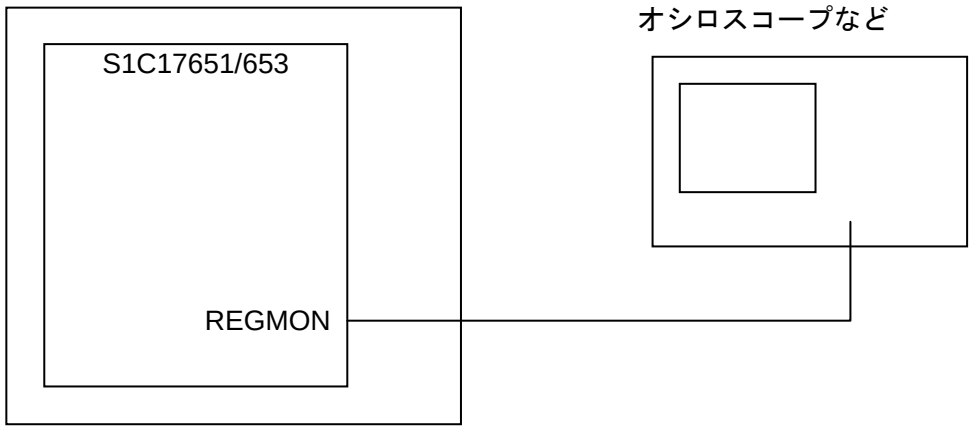

図 3.16.1　論理緩急サンプルプログラムハードウェア接続図

3.16.3 動作概要

1. サンプルプログラムのメニューを Simulated I/O に表示します。
2. メニューが表示された状態で Simulated I/O に「1」を入力して ENTER キーを押下すると、論理緩急調整値「-15/32768」を設定します。論理緩急調整値は「1」～「32」(-15/32768 ～ +16/32768)です。
3. 論理緩急設定値を設定すると、1 秒おきに論理緩急を実行します。
4. 論理緩急を 10 秒間実行します。10 秒経過すると、論理緩急調整値の選択が可能になります。

```
<<< TR demonstration start >>>
1.  -15/32768   2. -14/32768   3. -13/32768   4. -12/32768
5.  -11/32768   6. -10/32768   7. -9/32768    8. -8/32768
9.  -7/32768   10. -6/32768   11. -5/32768   12. -4/32768
13. -3/32768   14. -2/32768   15. -1/32768   16. 0/32768
17. +1/32768   18. +2/32768   19. +3/32768   20. +4/32768
21. +5/32768   22. +6/32768   23. +7/32768   24. +8/32768
25. +9/32768   26. +10/32768  27. +11/32768  28. +12/32768
29. +13/32768  30. +14/32768  31. +15/32768  32. +16/32768
Please input number
<<< TR demonstration finish >>>
```

図 3.16.2　SVD サンプルプログラム画面表示例

4. サンプルドライバ関数一覧

ここではサンプルソフトウェアの各サンプルドライバの一覧を記述します。

4.1 入出力ポート（P）

表 4.1 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード port.c を参照して下さい。

表 4.1 入出力ポート（P）サンプルドライバ関数一覧

関数名	機能名
SetPortInput	Px ポートのプルアップ設定
SetPortOutput	Px ポートの出力設定
SetPortOutputData	Px ポートのデータ出力設定
SetP0Chat	P0 ポートのチャタリング除去設定
InitP0Int	P0 ポート割り込みの初期化設定
EnableP0Int	P0 ポート割り込みの許可設定
DisableP0Int	P0 ポート割り込みの禁止設定
isP0Int	P0 ポート割り込み要因の確認
ClrP0IntFlg	P0 ポート割り込み要因のフラグクリア設定

本サンプルドライバはport.cとport.hに記述しています。

本サンプルドライバを使用するプログラムは port.h をインクルードして下さい。

4.2 発振回路（OSC）

表 4.2 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード clg.c を参照して下さい。

表 4.2 発振回路（OSC）サンプルドライバ関数一覧

関数名	機能名
StopOSC1	OSC1 の発振停止処理
StartOSC1	OSC1 の発振開始処理
StopOSC3A	OSC3A の発振停止処理
StartOSC3A	OSC3A の発振開始処理
StopOSC3B	OSC3B の発振停止処理
StartOSC3B	OSC3B の発振開始処理
SetWaitCycleClg	クロック発振開始時の待ち時間設定
ChgOSC	システムクロックを切り替える

本サンプルドライバはclg.cとclg.hに記述しています。

本サンプルドライバを使用するプログラムは clg.h をインクルードして下さい。

4.3 8 ビットタイマ（T8）

表 4.3 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード t8.c を参照して下さい。

表 4.3 8 ビットタイマ（T8）サンプルドライバ関数一覧

関数名	機能名
InitT8	8 ビットタイマの初期化
GetT8Count	カウンターデータの取得
StartT8	8 ビットタイマの開始設定
StopT8	8 ビットタイマの停止設定
InitT8Int	8 ビットタイマ割り込みの初期化設定
EnableT8Int	8 ビットタイマ割り込みの許可設定
DisableT8Int	8 ビットタイマ割り込みの禁止設定
isT8Int	8 ビットタイマ割り込みの確認
ClrT8IntFlg	8ビットタイマ割り込み要因のフラグクリア設定

本サンプルドライバはt8.cとt8.hに記述しています。
本サンプルドライバを使用するプログラムは t8.h をインクルードして下さい。

4.4 T16PWM タイマ（T16A2）

表 4.4 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード t16a2.c を参照して下さい。

表 4.4 T16PWM タイマ（T16A2）サンプルドライバ関数一覧

関数名	機能名
InitT16A2	T16 ビット PWM タイマの初期化
DataInitT16A2	T16 ビット PWM タイマカウンター値の設定
GetT16A2Count	カウンターデータの取得
StartT16A2	T16 ビット PWM タイマの開始設定
StopT16A2	T16 ビット PWM タイマの停止設定
InitT16A2Int	T16 ビット PWM タイマ割り込みの初期化設定
EnableT16A2Int	T16 ビット PWM タイマ割り込みの許可設定
DisableT16A2Int	T16 ビット PWM タイマ割り込みの禁止設定
isT16A2Int	T16 ビット PWM タイマ割り込みの確認
ClrT16A2IntFlg	T16ビットPWMタイマ割り込み要因のフラグクリア設定

本サンプルドライバはt16a2.cとt16a2.hに記述しています。

本サンプルドライバを使用するプログラムは t16a2.h をインクルードして下さい。

4.5 計時タイマ（CT）

表 4.5 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード ct.c を参照して下さい。

表 4.5 計時タイマ（CT）サンプルドライバ関数一覧

関数名	機能名
ResetCTCount	計時タイマの初期化
StartCT	計時タイマの開始設定
StopCT	計時タイマの開始設定
InitCTInt	計時タイマ割り込みの初期化設定
EnableCTInt	計時タイマ割り込みの許可設定
DisableCTInt	計時タイマ割り込みの禁止設定
isCTInt	計時タイマ割り込みの確認
ClrCTIntFlg	計時タイマ割り込み要因のフラグクリア設定

本サンプルドライバはct.cとct.hに記述しています。
本サンプルドライバを使用するプログラムは ct.h をインクルードして下さい。

4.6 リアルタイムクロック（RTC）

表4.6に本サンプルドライバの関数一覧を示します。関数の詳細はソースコードrtc.cを参照して下さい。

表 4.6 リアルタイムクロック（RTC）サンプルドライバ関数一覧

関数名	機能名
InitRTC	RTC の初期化
StartRTC	RTC の開始設定
StopRTC	RTC の開始設定
InitRTCInt	RTC 割り込みの初期化設定
EnableRTCInt	RTC 割り込みの許可設定
DisableRTCInt	RTC 割り込みの禁止設定
isRTCInt	RTC 割り込みの確認
ClrRTCIntFlg	RTC割り込み要因のフラグクリア設定

本サンプルドライバはrtc.cとrtc.hに記述しています。

本サンプルドライバを使用するプログラムは rtc.h をインクルードして下さい。

4.7 ウォッチドッグタイマ（WDT）

表 4.7 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード wdt.c を参照して下さい。

表 4.7 ウォッチドッグタイマ（WDT）サンプルドライバ関数一覧

関数名	機能名
InitWDT	WDT の初期化
StartWDT	WDT の開始設定
StopWDT	WDT の開始設定
ResetWDT	WDT のリセット設定
isWDTnmi	WDT の NMI 発生の確認

本サンプルドライバはwdt.cとwdt.hに記述しています。

本サンプルドライバを使用するプログラムは wdt.h をインクルードして下さい。

4.8 UART

表 4.8 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード uart..c を参照して下さい。

表 4.8 UART サンプルドライバ関数一覧

関数名	機能名
InitUART	UART の初期化
SendDataUART	送信データの設定
ReceiveDataUART	受信データの取得
StartUART	UART 送受信の開始設定
StopUART	UART 送受信の停止設定
EnableUARTInt	UART 割り込みの許可設定
DisableUARTInt	UART 割り込みの禁止設定
InitUARTInt	UART 割り込みの初期化設定
isUARTInt	UART 割り込みの確認
ClrUARTIntFlg	UART割り込み要因のフラグクリア設定

本サンプルドライバはuart.cとuart.hに記述しています。

本サンプルドライバを使用するプログラムは uart.h をインクルードして下さい。

4.9 SPI

表 4.9 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード spi.c を参照して下さい。

表 4.9 SPI サンプルドライバ関数一覧

関数名	機能名
InitSPI	SPI の初期化
SendDataSPI	送信データの設定
ReceiveDataSPI	受信データの取得
StartSPI	SPI 送受信の開始設定
StopSPI	SPI 送受信の停止設定
EnableSPIInt	SPI 割り込みの許可設定
DisableSPIInt	SPI 割り込みの禁止設定
InitSPIInt	SPI 割り込みの初期化設定
isSPIInt	SPI 割り込みの確認
ClrSPIIntFlg	SPI割り込み要因のフラグクリア設定

本サンプルドライバはspi.cとspi.hに記述しています。

本サンプルドライバを使用するプログラムは spi.h をインクルードして下さい。

4.10 LCD ドライバー（LCD）

表 4.10 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード lcd.c を参照して下さい。

表 4.10 LCD サンプルドライバ関数一覧

関数名	機能名
InitLCDPower	LCD 電源の初期化
InitLCD	LCD の初期化
SetLCDDisplay1Seg	1 セグメントの表示
StartLCDClock	LCD クロックの供給開始設定
StopLCDClock	LCD クロックの供給停止設定
InitLCDInt	LCD 割り込みの初期化設定
EnableLCDInt	LCD 割り込みの許可設定
DisableLCDInt	LCD 割り込みの禁止設定
isLCDInt	LCD 割り込みの確認
ClrLCDIntFlg	LCD割り込み要因のフラグクリア設定

本サンプルドライバはlcd.cとlcd.hに記述しています。

本サンプルドライバを使用するプログラムは lcd.h をインクルードして下さい。

4.11 サウンドジェネレータ（SND）

表 4.11 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード snd.c を参照して下さい。

表 4.11 SND サンプルドライバ関数一覧

関数名	機能名
InitSND	SND の初期化
StartSND	SND ブザーの開始設定
StopSND	SND ブザーの停止設定

本サンプルドライバはsnd.cとsnd.hに記述しています。

本サンプルドライバを使用するプログラムは snd.h をインクルードして下さい。

4.12 電源電圧検出回路（SVD）

表 4.12 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード svd.c を参照して下さい。

表 4.12 SVD サンプルドライバ関数一覧

関数名	機能名
SetSVDCompVolt	SVD 比較電圧の設定
StartSVD	SVD の検出開始設定
StopSVD	SVD の検出停止設定
GetSVDResult	SVD検出結果の取得

本サンプルドライバはsvd.cとsvd.hに記述しています。

本サンプルドライバを使用するプログラムは svd.h をインクルードして下さい。

4.13 MISC

表 4.13 に本サンプルドライバの関数一覧を示します。関数の詳細はソースコード init.c を参照して下さい。

表 4.13 MISC サンプルドライバ関数一覧

関数名	機能名
DebugModePsc	デバッグモード時における PCLK で動作する周辺回路の状態設定
ControlClg	内臓周辺モジュールへのクロックの供給設定
SetClockGear	システムクロックを減速するギア比の設定
SetFlashcAccessCycle	フラッシュメモリのウェイトサイクル数の設定
DebugModeMisc	デバッグモード時における PCLK 以外で動作する周辺回路の状態設定
ProtectMisc	MISC レジスタ書き込み保護・解除の設定
SetMisclramSize	内臓 RAM のサイズの設定
SetMiscVecAddress	ベクタテーブルアドレスの設定
SetMUXTargetOnly	ターゲット単体時における MUX の設定

本サンプルドライバはinit.cとinit.hに記述しています。

本サンプルドライバを使用するプログラムは init.h をインクルードして下さい。

4.14 論理緩急（TR）

表4.14に本サンプルドライバの関数一覧を示します。関数の詳細はソースコードtr.cを参照して下さい。

表 4.14 TR サンプルドライバ関数一覧

関数名	機能名
InitTR	TR の初期化
StartTR	TR のクロックモニタ出力開始設定
StopTR	TR のクロックモニタ出力停止設定
SetTRLogicAdjustVal	論理緩急調整値の設定
SetTRtrig	論理緩急実行の設定

本サンプルドライバはtr.cとtr.hに記述しています。
本サンプルドライバを使用するプログラムは tr.h をインクルードして下さい。

Appendix A 乗除算器

ここでは乗除算器の使い方について説明します。

A.1　乗除算器を使った乗算と除算

乗除算器を使った乗算と除算を行うために、GNU17 にはコプロセッサ用ライブラリが用意されています。

コプロセッサ用ライブラリの使い方につきましては S5U1C17001C Manual を参照して下さい。

A.1　乗除算器を使った積和演算

乗除算器を使って積和演算を行うためのプログラムを以下に示します。

本プログラムは“0x1204×0x1080＋0x28A00”の積和演算を行います。

```
asm ( "ld.cw %r0, 0x0" ); /* clear */
asm ( "ld.cw %r0, 0x2" ); /* setup mode */
asm ( "xld %r0, 0x0002" ); /* set 0x28A00 */
asm ( "xld %r1, 0x8A00" );
asm ( "ld.cf %r0, %r1" );
asm ( "ld.cw %r0, 0x7" ); /* setup mode */
asm ( "xld %r0, 0x1204" ); /* 0x1204 */
asm ( "xld %r1, 0x1080" ); /* 0x1080 */
asm ( "ld.ca %r0, %r1" );
asm ( "ld.cw %r0, 0x13" ); /* read */
asm ( "ld.ca %r1, %r0" );
asm ( "ld.cw %r0, 0x03" ); /* read */
asm ( "ld.ca %r2, %r0" );
/* result = 0x12BCC00 */
```

改訂履歴表

Rev. No.	日付	ページ	種別	改訂内容（旧内容を含む） および改訂理由
Rev 1.0	2012/07/06	全ページ	新規	新規制定

セイコーエプソン株式会社
マイクロデバイス事業部　IC 営業部

東京　〒191-8501　東京都日野市日野 421-8
TEL（042）587-5313（直通）　FAX（042）587-5116

大阪　〒541-0059　大阪市中央区博労町 3-5-1　エプソン大阪ビル 15F
TEL（06）6120-6000（代表）　FAX（06）6120-6100

ドキュメントコード：412394100
2012 年 7 月 作成