

S1C17 Family Application Note

S1C17656

周辺回路サンプルソフトウェア

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事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	3
1.1 動作環境	3
2. サンプルソフトウェア説明	4
2.1 ディレクトリ構成及びファイル構成	4
2.2 実行方法	5
3. サンプルソフトウェア機能詳細	6
3.1 入出力ポート (P)	6
3.1.1 サンプルソフトウェア仕様	6
3.1.2 ハードウェア条件	6
3.1.3 動作概要	7
3.2 クロックジェネレータ (CLG)	8
3.2.1 サンプルソフトウェア仕様	8
3.2.2 ハードウェア条件	8
3.2.3 動作概要	8
3.3 8 ビットタイマ (T8)	9
3.3.1 サンプルソフトウェア仕様	9
3.3.2 ハードウェア条件	9
3.3.3 動作概要	9
3.4 16ビットPWMタイマ (T16A2)	10
3.4.1 サンプルソフトウェア仕様	10
3.4.2 ハードウェア条件	10
3.4.3 動作概要	10
3.5 計時タイマ (CT)	11
3.5.1 サンプルソフトウェア仕様	11
3.5.2 ハードウェア条件	11
3.5.3 動作概要	11
3.6 リアルタイムクロック (RTC)	12
3.6.1 サンプルソフトウェア仕様	12
3.6.2 ハードウェア条件	12
3.6.3 動作概要	12
3.7 ウオッチドッグタイマ (WDT)	13
3.7.1 サンプルソフトウェア仕様	13
3.7.2 ハードウェア条件	13
3.7.3 動作概要	13
3.8 UART	14
3.8.1 サンプルソフトウェア仕様	14
3.8.2 ハードウェア条件	14
3.8.3 動作概要	14
3.9 SPI	16
3.9.1 サンプルソフトウェア仕様	16
3.9.2 ハードウェア条件	16
3.9.3 動作概要	16
3.10 LCD ドライバ(LCD)	18
3.10.1 サンプルソフトウェア仕様	18

3.10.2 ハードウェア条件	18
3.10.3 動作概要	18
3.11 SLEEP/HALT モード切替	19
3.11.1 サンプルソフトウェア仕様	19
3.11.2 ハードウェア条件	19
3.11.3 動作概要	19
3.12 サウンドジェネレータ (SND)	20
3.12.1 サンプルソフトウェア仕様	20
3.12.2 ハードウェア条件	20
3.12.3 動作概要	20
3.13 消費電流評価	21
3.13.1 サンプルソフトウェア仕様	21
3.13.2 ハードウェア条件	21
3.13.3 動作概要	21
3.14 電源電圧検出回路 (SVD)	22
3.14.1 サンプルソフトウェア仕様	22
3.14.2 ハードウェア条件	22
3.14.3 動作概要	22
3.15 MISC	23
3.15.1 動作概要	23
3.16 R/F 変換器 (RFC)	24
3.16.1 サンプルソフトウェア仕様	24
3.16.2 ハードウェア条件	24
3.16.3 動作概要	24
4. サンプルドライバ関数一覧	25
4.1 入出力ポート (P)	25
4.2 クロックジェネレータ (CLG)	25
4.3 8 ビットタイマ (T8)	26
4.4 16 ビット PWM タイマ (T16A2)	26
4.5 計時タイマ (CT)	27
4.6 リアルタイムクロック (RTC)	27
4.7 ウオッチドッグタイマ (WDT)	28
4.8 UART	28
4.9 SPI	29
4.10 LCD ドライバ (LCD)	29
4.11 サウンドジェネレータ (SND)	30
4.12 電源電圧検出回路 (SVD)	30
4.13 MISC	31
4.14 R/F 変換器 (RFC)	31
改訂履歴表	32

1. 概要

1. 概要

本マニュアルは S1C17656 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。

S1C17656 サンプルソフトウェアは S1C17656 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。

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

S1C17656 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。

- S1C17656 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、GNU17v2.3.0 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 S1C17656 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。

S1C17656 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S1C17656 サンプルソフトウェアのディレクトリ構成を示します。

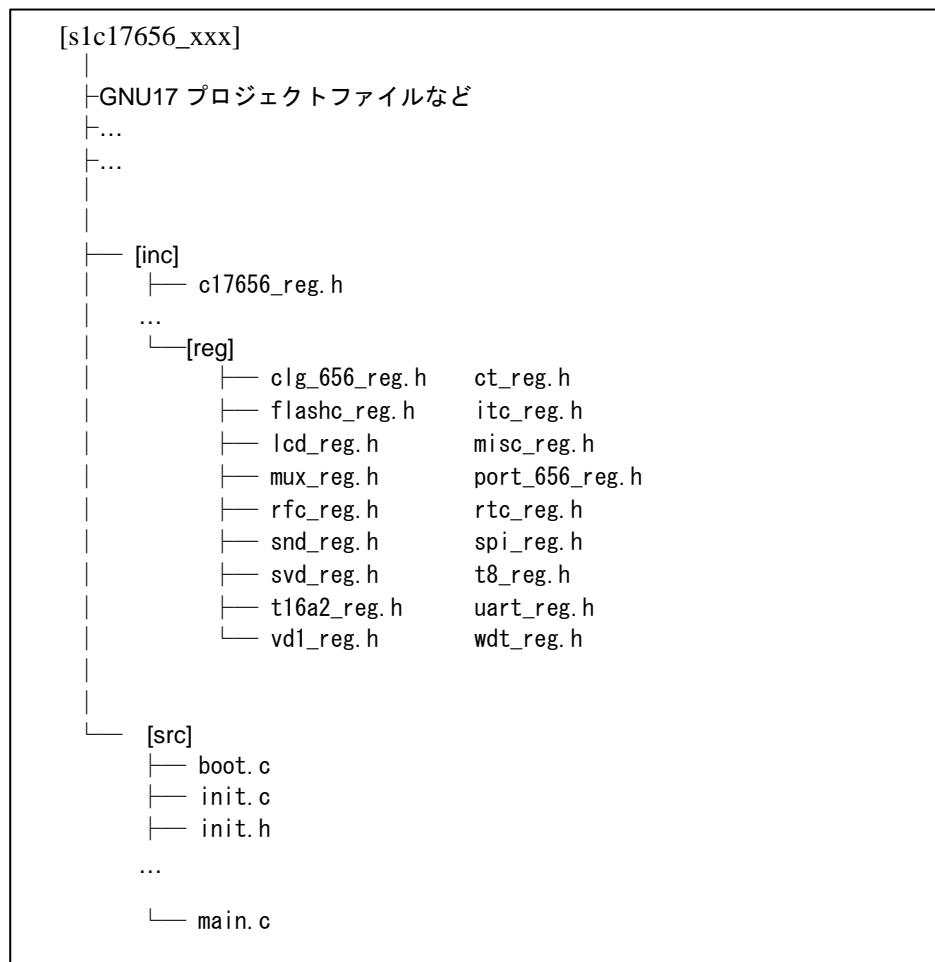


図2.1 S1C17656 サンプルソフトウェアディレクトリ構成図

(1) s1c17656_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 (c17656_reg.h)

(3) reg ディレクトリ 各周辺回路のビットアサインなどを定義したヘッダファイルが配置してあります。

2. サンプルソフトウェア説明

- 各周辺回路のビットアサインなどを定義したヘッダファイル (clg_656_reg.h など)
- (4) src ディレクトリ マイコンの初期化処理、各周辺回路のサンプルソフトウェア及び、サンプルドライバ、各周辺回路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。
- 初期化処理のファイル (boot.c)
 - 各周辺回路のサンプルソフトウェアファイル (main.c)
 - 各周辺回路のサンプルドライバファイル (init.c など)
 - 各周辺回路のヘッダファイル (init.h など)

2.2 実行方法

以下の手順でS1C17656 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S1C17656 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S1C17656 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 S1C17656 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S1C17656 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

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路で動作します。本サンプルソフトウェア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。

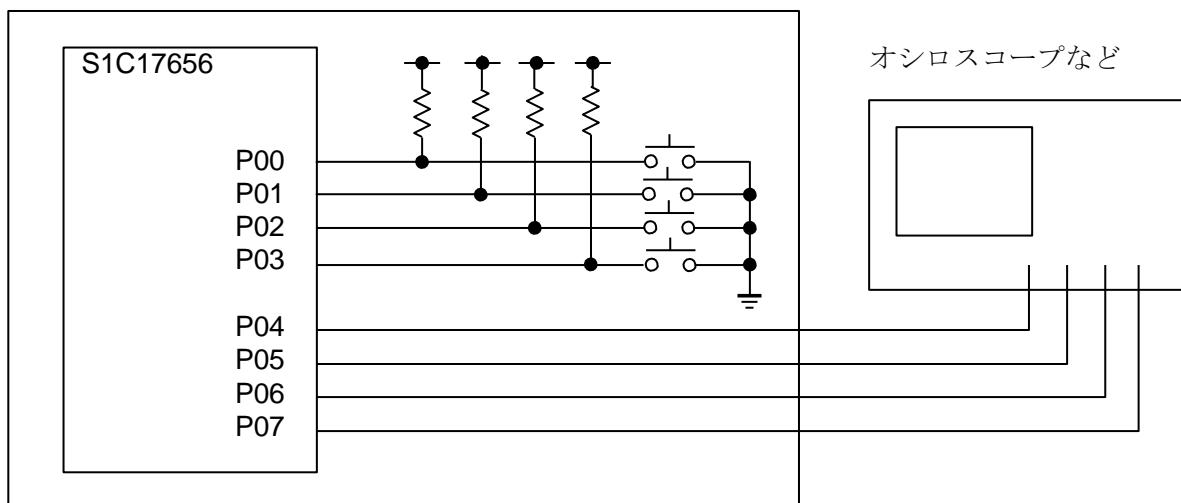


図3.1.1 入出力ポート (P) サンプルソフトウェアハードウェア接続図

3. サンプルソフトウェア機能詳細

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(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。)た後、サンプルソフトウェアを終了します。

```
<<< 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ソフトウェア画面表示例

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

本サンプルソフトウェアはクロックジェネレータを使用して以下の動作を行います。

- OSC1A 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OSC1A へ切り替える。
- システムクロックをOSC1A からOSC3B へ切り替える。

3.2.2 ハードウェア条件

本サンプルソフトウェアはマイコンのOSC1A に水晶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、S1C17656 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

本サンプルソフトウェアは OSC3B を使用した状態で動作を開始します。

- 一定間隔で Simulated I/O に「1」、「2」、「3」...、「9」と表示した後、OSC1A の発振を開始し、システムクロックを OSC3B から OSC1A に切り替え、OSC3B を停止します。
- 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 OSC1A から OSC3B に切り替え、OSC1A を停止します。
- 一定間隔で Simulated I/O に「1」「2」「3」...、「9」と表示した後、サンプルソフトウェアを終了します。

```
<<< CLG(OSC) demonstration start >>>
OSC3B *** 1 ***
OSC3B *** 2 ***
...
OSC3B *** 9 ***
*** Change from OSC3B 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 サンプルソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

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

本サンプルソフトウェアは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

本サンプルソフトウェアはマイコンのOSC3B(2MHz)内蔵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データ値を Simulated I/O に表示し、サンプルソフトウェアを終了します。

```
<<< 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ソフトウェア画面表示例

3.4 16ビットPWMタイマ (T16A2)

3.4.1 サンプルソフトウェア仕様

本サンプルソフトウェアは 16 ビット PWM タイマを使用して以下の動作を行います。

- 16 ビット PWM タイマコンペア A 割り込みを発生させ、タイマのカウンタ値を取得する。
- 16 ビット PWM タイマコンペア B 割り込みを発生させ、タイマのカウンタ値を取得する。
- 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。

3.4.2 ハードウェア条件

本サンプルソフトウェアはマイコンのOSC3B(2MHz)内蔵発振回路で動作します。

3.4.3 動作概要

1. コンペア A 割り込みとコンペア B 割り込みを有効にし、16 ビット PWM タイマを開始します。
2. コンペア A 割り込みおよびコンペア B 割り込みが発生すると、アップカウンタのカウンタ値を取得します。
3. コンペア B 割り込みが 5 回発生すると 16 ビット PWM タイマを停止し、割り込みの種類とカウンタ値を Simulated I/O に表示してサンプルソフトウェアを終了します。

```
<<< 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 3.4.1 16 ビットPWM タイマサンプルソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

3.5 計時タイマ (CT)

3.5.1 サンプルソフトウェア仕様

本サンプルソフトウェアは計時タイマを使用して以下の動作を行います。

- 計時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

本サンプルソフトウェアはマイコンのOSC3B(2MHz)内蔵発振回路及び、OSC1A(32.768kHz)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、計時タイマ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ソフトウェアを終了します。

```
<<< 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ソフトウェア画面表示例

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

本サンプルソフトウェア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ソフトウェア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

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路及び、OSC1A(32.768kHz)で動作します。

3.6.3 動作概要

- RTC サンプルソフトウェアのメニューを Simulated I/O に表示します。
- Simulated I/O にてユーザーが時刻の設定を行います。
- 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。
- 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ソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

3.7 ウオッチドッグタイマ (WDT)

3.7.1 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

本サンプルソフトウェアはマイコンのOSC3B(2MHz)内蔵発振回路及び、OSC1A(32.768kHz)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リセットします。
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、サンプルソフトウェアを終了します。

```
<<< 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ソフトウェア画面表示例

3.8 UART

3.8.1 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

本サンプルソフトウェアはマイコンのOSC3B(4MHz)内蔵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。

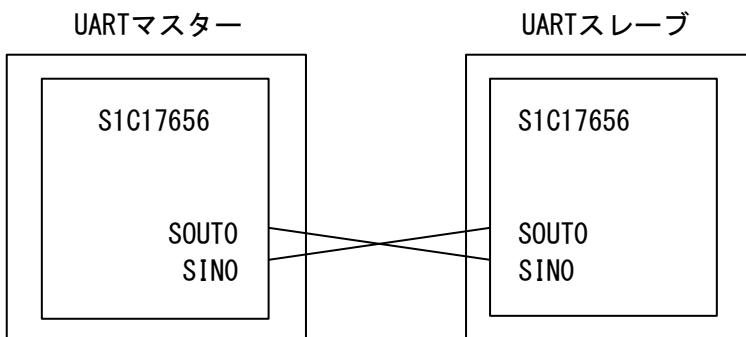


図3.8.1 UARTサンプルソフトウェアハードウェア接続図

3.8.3 動作概要

3.8.3.1 マスターサンプルソフトウェア動作概要

1. UARTポートを以下の値に初期化します。
 - 通信速度 : 22222bps
 - 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 OSC3B demonstration start >>>
waiting connection.
connected
*** receive data ***
!#$%& ()*+, -./0...
<<< UART OSC3B demonstration finish >>>
```

図 3.8.2 UART (OSC3B) マスターサンプルソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

3.8.3.2 スレーブサンプルソフトウェア動作概要

1. UART ポートを以下の値に初期化します。
 - 通信速度 : 222222bps
 - 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 OSC3B demonstration start >>>
waiting connection.
connected
*** receive data ***
! "#$%&' ()*+, -./0...
<<< UART OSC3B demonstration finish >>>
```

図 3.8.3 UART (OSC3B) スレーブサンプルソフトウェア画面表示例

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

本サンプルソフトウェアは 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

本サンプルソフトウェアは 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

各サンプルソフトウェアはマイコンのOSC3B(2MHz)内蔵発振回路で動作します。

なお、SPI スレーブサンプルソフトウェアが動作しているS1C17656 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。

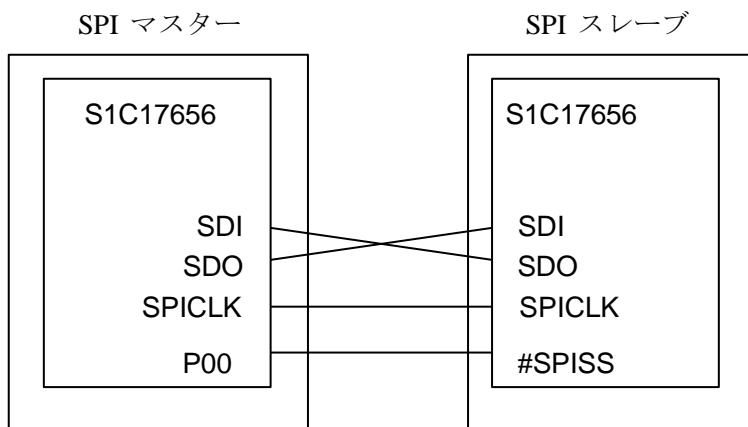


図3.9.1 SPI マスター、スレーブサンプルソフトウェアハードウェア接続図

3.9.3 動作概要

3.9.3.1 マスターサンプルソフトウェア動作概要

- SPI マスターの初期設定を行います。
- SPI スレーブへ 8byte の ASCII データ「FROM MST」を送信します。
- これに続き、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。

3. サンプルソフトウェア機能詳細

4. SPI スレーブから受信したデータを Simulated I/O に表示し、サンプルソフトウェアを終了します。

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

図3.9.2 SPI マスターサンプルソフトウェア画面表示例

3.9.3.2 スレーブサンプルソフトウェア動作概要

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、SPI マスターへ 8byte の ASCII データ「FROM SLV」を送信します。
4. 受信データを Simulated I/O に表示し、サンプルソフトウェアを終了します。

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

図 3.9.3 SPI スレーブサンプルソフトウェア画面表示例

3.10 LCD ドライバ(LCD)

3.10.1 サンプルソフトウェア仕様

本サンプルソフトウェアは LCD ドライバ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。
- 表示 Off 機能における全消灯を行う。

3.10.2 ハードウェア条件

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵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。

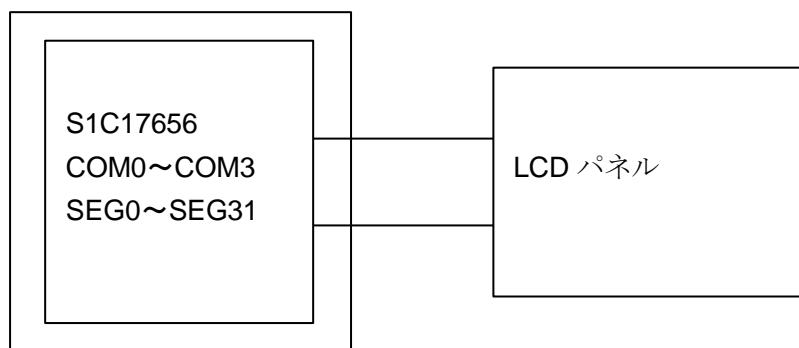


図3.10.1 LCD ドライバサンプルソフトウェアハードウェア接続図

3.10.3 動作概要

- 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。
- 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。
- 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。
- 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。
- Simulated I/O に「off2」と入力して ENTER キーを押下すると、表示 Off 機能における LCD の全消灯を行います。

```
<<< LCD driver demonstration start >>>
on0
on1
off0
off1
off2
exit
<<< LCD driver demonstration finish >>>
```

図 3.10.2 LCD ドライバサンプルソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

3.11 SLEEP/HALT モード切替

3.11.1 サンプルソフトウェア仕様

本サンプルソフトウェアは以下の Sleep/Halt モード切替動作を行います。

- 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。
- 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

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路及び、OSC1A(32.768kHz)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。

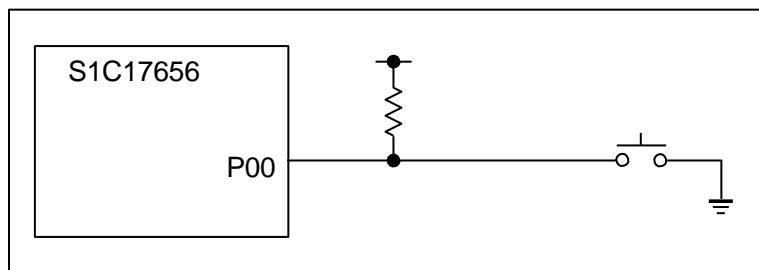


図3.11.1 Sleep/Halt サンプルソフトウェアハードウェア接続図

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. CPU を再び sleep モードにします。
6. 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ソフトウェア画面表示例

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

本サンプルソフトウェアはサウンドジェネレータ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、周波数を上げながら出力した波形を BZ 端子に出力します。

3.12.2 ハードウェア条件

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路及び、OSC1A(32.768kHz)で動作します。

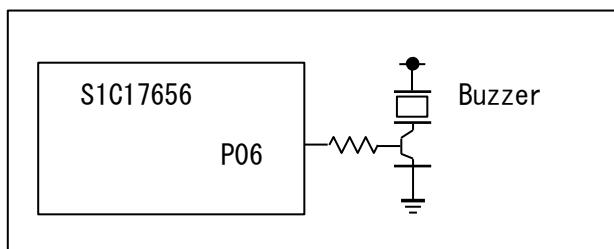


図3.12.1 サウンドジェネレータサンプルソフトウェアハードウェア接続図

3.12.3 動作概要

- ブザー周波数を「4096.0Hz」に設定します。
- Simulated I/O にブザー周波数を表示します。
- ブザーを 1 秒間出力します。
- エンベロープモードに設定後、ブザーを出力します。
- 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。
- ブザー周波数を下げて設定し「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。
- 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、サンプルソフトウェアを終了します。

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

図3.12.2 サウンドジェネレータサンプルソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

3.13 消費電流評価

3.13.1 サンプルソフトウェア仕様

本サンプルソフトウェアは消費電流評価のために以下のいずれかの状態を用意します。

- OSC1A のみを発振させ、RTC および PCLK が停止した状態でマイコンを Halt させた状態。
- OSC1A のみを発振させ、RTC は動作、PCLK は停止した状態でマイコンを Halt させた状態。
- RTC が停止した状態でマイコンを Sleep させた状態。
- RTC が動作した状態でマイコンを Sleep させた状態。

3.13.2 ハードウェア条件

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路及び、OSC1A (32.768kHz) で動作します。

3.13.3 動作概要

本サンプルソフトウェアは、表 3.13.1 の設定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 消費電流評価サンプルソフトウェア設定一覧

設定	動作
HALT_OSC1A	OSC1A のみを発振させ、RTC および PCLK が停止した状態でマイコンを Halt させた状態にする。
HALT_OSC1A_RTC	OSC1A のみを発振させ、RTC は動作、PCLK は停止した状態でマイコンを Halt させた状態にする。
SLEEP_ONLY	RTC が停止した状態でマイコンを Sleep させた状態にする。
SLEEP_RTC	RTC が動作した状態でマイコンを Sleep させた状態にする。

- 測定を行うサンプルソフトウェアをあらかじめフラッシュメモリに書き込みます。
- マイコンの電源を ON にし、サンプルソフトウェア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

本サンプルソフトウェアは電源電圧検出回路を使用して以下の動作を行います。

- 比較電圧を設定し、VDD が比較電圧未満か否かを検出します。

3.14.2 ハードウェア条件

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路で動作します。本サンプルソフトウェアは以下のように接続してご使用下さい。

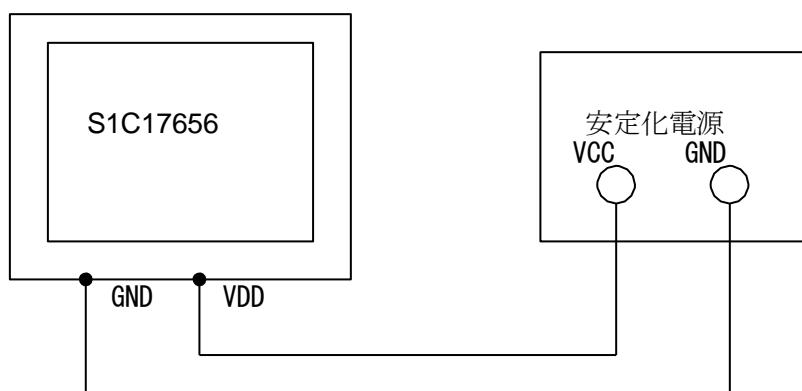


図3.14.1 電源電圧検出回路サンプルソフトウェアハードウェア接続図

3.14.3 動作概要

- 比較電圧を 3.20V に設定します。
- 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。
- 比較電圧と検出電圧を比較後、結果を Simulated I/O に表示します。

```
<<< 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ソフトウェア画面表示例

3. サンプルソフトウェア機能詳細

3.15 MISC

3.15.1 動作概要

サンプルソフトウェアの最初に以下の設定を行います。

1. Flash メモリリード時のウェイト数を設定します。
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. 内部周辺モジュールへのクロックの供給を設定します。

3.16 R/F 変換器 (RFC)

3.16.1 サンプルソフトウェア仕様

本サンプルソフトウェアは R/F 変換器を使用して以下の動作を行います。

- センサの抵抗値を CR 発振させ、発振クロックをカウントしてデジタル値に変換する。
- 接続例) 基準抵抗 $10\text{k}\Omega$ 抵抗性センサ $15\text{k}\Omega$ 基準キャパシタ 1000pF

3.16.2 ハードウェア条件

本サンプルソフトウェアはマイコンの OSC3B(2MHz)内蔵発振回路及び、OSC1A(32.768kHz)で動作します。本サンプルソフトウェアは以下のように接続してご使用下さい。

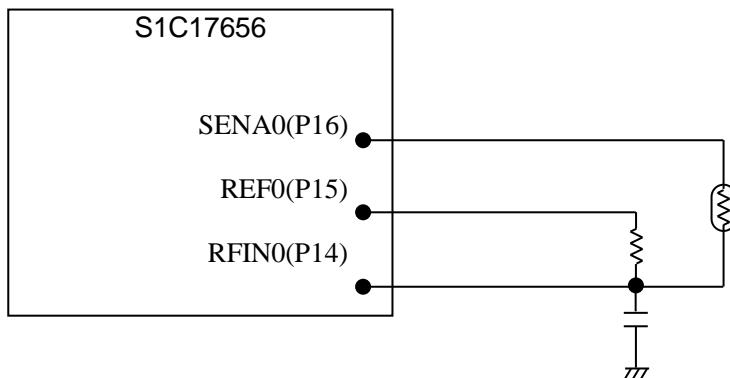


図 3.16.1 R/F 変換器サンプルソフトウェアハードウェア接続図

3.16.3 動作概要

- RFC の外部クロック入力モード、連続発振を設定します。
- R/F 変換器の計測カウンタの初期値を `0x00fc0000` とし、計測を開始します。
- 基準発振、センサ発振の制御により計測カウンタ値を取得し、サンプルソフトウェアを終了します。

```
<<< R/F Converter demonstration start >>>
*** Measurement Counter : 262166 ***
<<< R/F Converter demonstration finish >>>
```

図3.16.2 R/F 変換器サンプルソフトウェア画面表示例

4. サンプルドライバ関数一覧

4. サンプルドライバ関数一覧

ここでは各周辺回路のサンプルドライバ一覧を記述します。

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
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 クロックジェネレータ (CLG)

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 クロックジェネレータ (CLG) サンプルドライバ関数一覧

関数名	機能
StopOSC1	OSC1 の発振停止処理
StartOSC1	OSC1 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	システムクロックの切り替え処理

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
InitT8	8 ビットタイマの初期化
GetT8Count	カウンタデータの取得
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 16 ビット PWM 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 16 ビット PWM タイマ (T16A2) サンプルドライバ関数一覧

関数名	機能
InitT16A2	16 ビット PWM タイマの初期化
DataInitT16A2	16 ビット PWM タイマカウンタ値の設定
GetT16A2Count	カウンタデータの取得
StartT16A2	16 ビット PWM タイマの開始設定
StopT16A2	16 ビット PWM タイマの停止設定
InitT16A2Int	16 ビット PWM タイマ割り込みの初期化設定
EnableT16A2Int	16 ビット PWM タイマ割り込みの許可設定
DisableT16A2Int	16 ビット PWM タイマ割り込みの禁止設定
isT16A2Int	16 ビット PWM タイマ割り込みの確認
ClrT16A2IntFlg	16 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. サンプルドライバ関数一覧

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
ResetCTCount	計時タイマのリセット
StartCT	計時タイマの開始設定
StopCT	計時タイマの停止設定
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
InitRTC	RTC の初期化
StartRTC	RTC の開始設定
StopRTC	RTC の停止設定
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 ウオッチドッグタイマ (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 ウオッチドッグタイマ (WDT) サンプルドライバ関数一覧

関数名	機能
InitWDT	WDT の初期化
StartWDT	WDT の開始設定
StopWDT	WDT の停止設定
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
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. サンプルドライバ関数一覧

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
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

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 (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
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
InitSND	SND の初期化
StartSND	SND ブザー出力の開始設定
StopSND	SND ブザー出力の停止設定

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
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. サンプルドライバ関数一覧

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
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	内部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	Flash メモリリード時のウェイト数の設定
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
SetMiscIramSize	内蔵 RAM のサイズの設定
SetMiscVecAddress	ペクターブルベースアドレスの設定

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 R/F 変換器 (RFC)

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 `rfc.c` を参照して下さい。

表4.14 RFC サンプルドライバ関数一覧

関数名	機能
initRfc	RFCの初期化
startRfc	RFC の開始設定
stopRfc	RFC の停止設定
setRfcMeasurementCounter	計測カウンタ値の設定
getRfcMeasurementCounter	計測カウンタ値の読み出し
setRfcTimeBaseCounter	タイムベースカウンタ値の設定
getRfcTimeBaseCounter	タイムベースカウンタ値の読み出し
runRfcConvertingOperation	基準発振/センサ発振の変換動作制御

本サンプルドライバは `rfc.c` と `rfc.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 `rfc.h` をインクルードして下さい。

改訂履歴表

改訂履歴表

付-1

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	2014/09/16	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 デバイス営業部

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: 412867600
2014 年 10 月 作成