EPSON
EXCEED YOUR VISION

S1C17 Family Application Note

S1C17700 シリーズ周辺回路サンプルソフトウェア２

セイコーエプソン株式会社

Rev.1.0

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. 本資料に掲載されている製品のうち「外国為替及び外国貿易法」に定める戦略物資に該当するものについては、輸出する場合、同法に基づく輸出許可が必要です。

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 S1C17703/706 マイコンで新たに追加された周辺機器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。S1C17700 シリーズ共通の周辺回路については、以下のマニュアルを参照してください。

・S1C17700 シリーズ周辺回路サンプルソフトウェア

S1C17700 シリーズ周辺回路サンプルソフトウェア２は S1C17703/706 マイコンで新たに追加された周辺回路の使用例を示すことを目的としています。

S1C17700 シリーズ周辺回路サンプルソフトウェア２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。

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

S1C17700 シリーズ周辺回路サンプルソフトウェア２を動作させるにあたり、以下の機材をご用意下さい。

- S1C17703/706 が実装されたボード
- 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、GNU17v2.0.0 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S1C17700シリーズ周辺回路サンプルソフトウェア２のファイル構成と実行方法を記載します。

S1C17700 シリーズ周辺回路サンプルソフトウェア２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 収録されているサンプルソフトウェア

以下に収録されているサンプルソフトウェアの一覧を示します。

表 2.1 収録されているサンプルソフトウェア一覧

周辺回路	サンプルソフトウェア
入出力ポート（P）	○
クロックジェネレータ（CLG）	○
16 ビットタイマ（T16）	○
アドバンスドタイマ（T16A）	○
計時タイマ（CT）	○
ストップウォッチタイマ（SWT）	○
ウォッチドッグタイマ（WDT）	○
OSC3 を使用した UART	○
IOSC を使用した UART	○
SPI マスタ	○
SPI スレーブ	○
I2C マスタ（I2CM）	○
I2C スレーブ（I2CS）	○
LCD ドライバ（LCD）	○
電源電圧検出回路（SVD）	○
R/F 変換器（RFC）	○
A/D 変換器（ADC10）	○
リモートコントローラ送信（REMC）	○
リモートコントローラ受信（REMC）	○
Sleep/Halt	○
アドバンスドタイマ（T16A2）	◎
LCD ドライバ（LCD32A）	◎
電源電圧検出回路（SVD2）	◎
サウンドジェネレータ（SND）	◎
リアルタイムクロック（RTC2）	◎
電流測定	◎

◎:新規、変更 ○:シリーズ共通

2.2 収録されているサンプルドライバ

以下に収録されているサンプルドライバの一覧を示します。

表 2.2 収録されているサンプルドライバー覧

周辺回路	サンプルドライバ
入出力ポート(P)	○
クロックジェネレータ(CLG)	○
16 ビットタイマ(T16)	○
アドバンスドタイマ(T16A)	○
計時タイマ(CT)	○
ストップウォッチタイマ(SWT)	○
ウォッチドッグタイマ(WDT)	◎
UART	○
SPI	○
I2C マスタ(I2CM)	○
I2C スレーブ(I2CS)	○
LCD ドライバ(LCD)	◎
電源電圧検出回路(SVD)	○
R/F 変換器(RFC)	○
A/D 変換器(ADC10)	○
リモートコントローラ(REMC)	○
プリスケーラ(PSC)	○
MISC	○
マルチプレクサ(MUX)	◎
アドバンスドタイマ(T16A2)	◎
LCD ドライバ(LCD32A)	◎
電源電圧検出回路(SVD2)	◎
サウンドジェネレータ(SND)	◎
リアルタイムクロック(RTC2)	◎
電源制御回路(VD1)	◎

◎:新規、変更 ○:シリーズ共通

2.3 ディレクトリ構成及びファイル構成

以下に S1C17700 シリーズ周辺回路サンプルソフトウェア２のディレクトリ構成を示します。

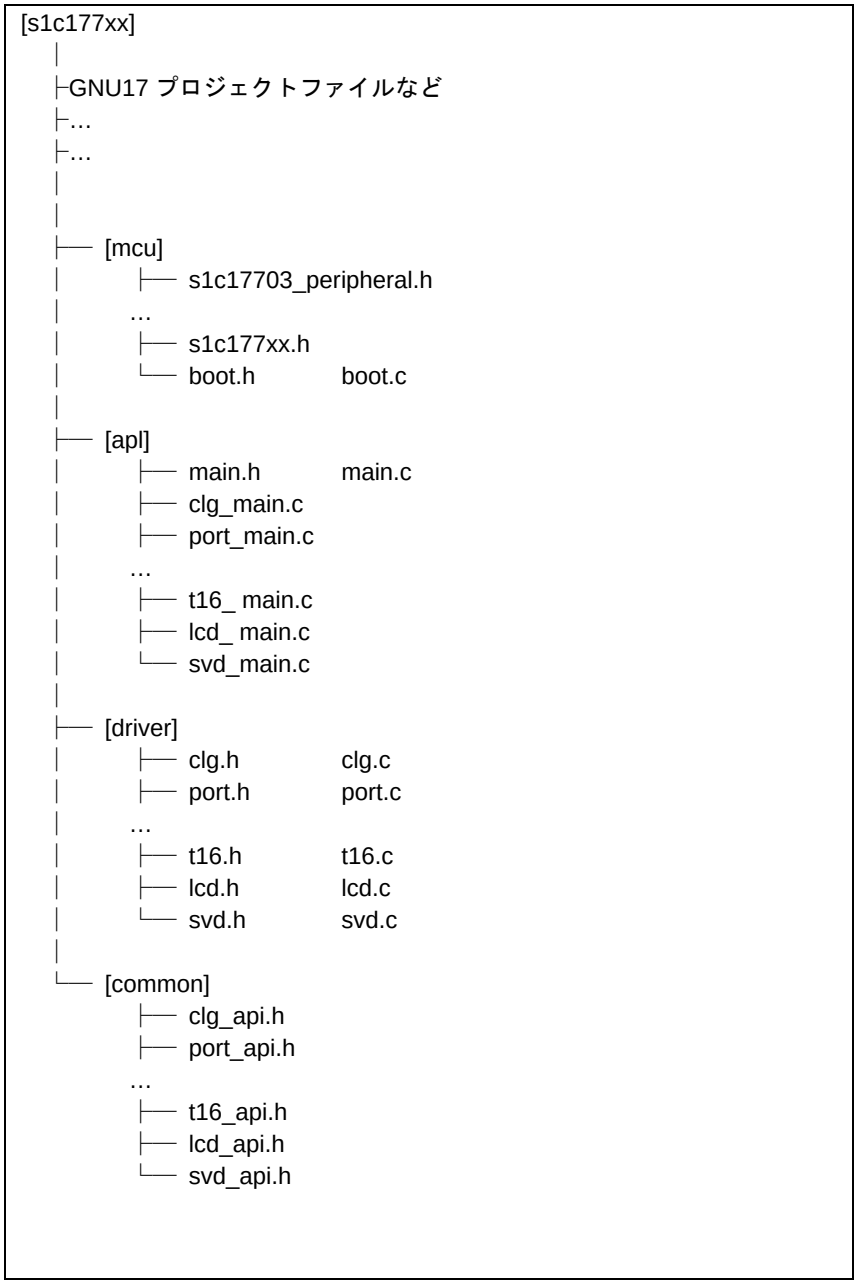

図 2.1　S1C17700 シリーズ周辺回路サンプルソフトウェア２ディレクトリ構成図

（1）s1c177xx 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と、サンプルソフトウェアのソースコードが格納されているディレクトリが配置されています。

（2）mcu ディレクトリ

マイコンの初期化処理と各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（s1c17703_peripheral.h など）
- 機種共通のヘッダファイル（s1c177xx.h）
- 初期化処理のファイル（boot.c）

（3）apl ディレクトリ

周辺回路毎のサンプルソフトウェアと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。

- 周辺回路毎のヘッダファイル（xxx.h）
- 周辺回路毎のサンプルソフトウェア（xxx.c）

（4）driver ディレクトリ

周辺回路毎のサンプルドライバが配置してあります。

- 周辺回路毎のレジスタアドレスやビットアサインを定義したヘッダファイル（xxx.h）
- 周辺回路毎のプログラム（xxx.c）

（5）common ディレクトリ

周辺回路毎のサンプルドライバが外部に提供する関数のプロトタイプを定義したヘッダファイルが配置してあります。

- 周辺回路毎のサンプルドライバが外部に提供する関数プロトタイプと引数の定数定義をしたヘッダファイル（xxx.h）

サンプルドライバを使用するソフトウェアは common ディレクトリに配置してあるヘッダファイルをインクルードしてサンプルドライバの関数を呼び出します。

2.4 実行方法

以下の手順で S1C17700 シリーズ周辺回路サンプルソフトウェア 2 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 S1C17700 シリーズ周辺回路サンプルソフトウェア 2 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 S1C177xx 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。

ICDmini 使用方法の詳細については S5U1C17001H User Manual を参照して下さい。

（4）デバッガによるプログラムのロードと実行

GNU17 の“デバッグの構成”ボタンを押下してデバッガを起動し、デバッグビューの“再開”ボタンを押下して下さい。

プログラムが S1C177xx 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いては S5U1C17001C Manual の“10 デバッガ”を参照して下さい。

2.5 サンプルソフトウェアメニュー

サンプルソフトウェアを起動すると、GNU17 の Simulated I/O(以下 SimI/O)にメニュー画面が表示されます。

Simulated I/O については S5U1C17001C Manual の“10.4.11 [シミュレーテッド I/O]ビュー”を参照して下さい。

プログラム番号を入力して Enter キーを押下すると選択したサンプルソフトウェアが起動します。

各サンプルソフトウェアの詳細は 3 章を参照して下さい。

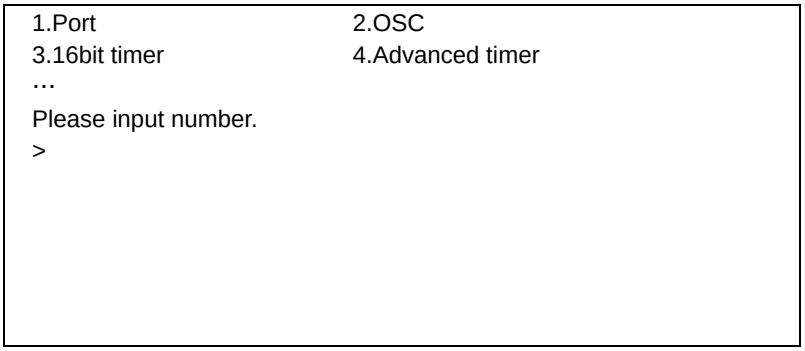

図 2.2 メニュー画面表示例

2.6 特定モジュールのビルド方法

S1C177xx サンプルソフトウェアは複数のプログラムがビルドされる状態で配布しております。

サンプルソフトウェアのソースコードを修正することで必要な周辺モジュールのサンプルソフトウェアだけをビルドすることができます。以下に手順を示します。

（1）修正対象ファイル

機種別の定義ヘッダを修正します。

S1C17706 サンプルソフトウェアの場合は s1c17706_periperal.h を修正します。

（2）修正箇所

ファイル下部にある以下の箇所を修正します。

```
//#undef PE_PORT
//#undef PE_CLG
//#undef PE_T16
//#undef PE_T16A
//#undef PE_T16A2
//#undef PE_CT
//#undef PE_SWT
//#undef PE_RTC2
//#undef PE_WDT
#undef   PE_UART
#undef   PE_UART_OSC3
#undef   PE_UART_IOSC
#undef   PE_SPI
#undef   PE_SPI_MASTER
#undef   PE_SPI_SLAVE
#undef   PE_I2CM
#undef   PE_I2CS
#undef   PE_LCD
#undef   PE_LCD32A
#undef   PE_SVD
#undef   PE_SVD2
#undef   PE_RFC
#undef   PE_ADC
#undef   PE_REMC
#undef   PE_REMC_TX
#undef   PE_REMC_RX
#undef   PE_SND
#undef   PE_SLEEP_HALT
#undef   PE_VD1
#undef   PE_CURRENT_MEASURE
```

図 2.3 特定モジュールの定義変更例

例えば入出力ポートサンプルソフトウェアのみビルドする場合、“#undef PE_PORT”の定義を無効にし、それ以外の“#undef PE_XXX”定義を有効にします。

ビルドする周辺モジュールのサンプルソフトウェアが他の周辺モジュールを使用する場合、使用する周辺モジュールサンプルソフトウェアもビルドする必要があります。

例えば I2CM サンプルソフトウェアは 16 ビットタイマを使用しますので、I2CM サンプルソフトウェアをビルドする際は、“#undef PE_I2CM”と“#undef PE_T16”の定義を無効にする必要があります。

3　サンプルソフトウェア機能詳細

本章では S1C17700 シリーズ周辺回路サンプルソフトウェア 2 の機能詳細について記載します。

3.1　アドバンスドタイマ（T16A2）

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アドバンスドタイマを使用して以下の動作を行います。

- アドバンスド**タイマ**コンペア A マッチ割り込みをノーマルモードで 5 回発生させ、**タイマ**のカウンター値を取得します。
- アドバンスドタイマコンペア B マッチ割り込みをノーマル/ハーフクロックモードでそれぞれ 5 回発生させ、**タイマ**のカウンター値を取得します。
- TOUTA0 端子に PWM 波形をノーマル/ハーフクロックモードで出力します。
- 割り込み待機時は CPU を halt モードにして消費電力を低減します。

3.1.2　ハードウェア条件

本サンプルソフトウェアは OSC1 と OSC3 に水晶振動子またはセラミック振動子が接続された状態で動作します。

振動子の接続方法につきましては、各テクニカルマニュアル「クロックジェネレータ（CLG）発振回路」をご参照下さい。

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。

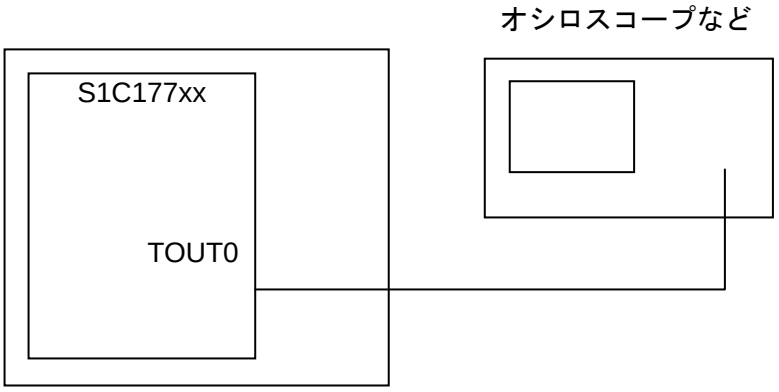

図 3.1　アドバンスドタイマ（T16A2）サンプルソフトウェアハードウェア接続図

3.1.3　動作概要

（1）サンプルソフトウェア動作概要

- 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てアドバンスドタイマを開始します。
- 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したときにアップカウンターのカウンター値を取得します。
- コンペア B マッチ割り込みが 5 回発生したところでアドバンスドタイマを停止し、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。

```
<<< PWM timer(T16A2) demonstration start >>>
Normal clock mode start
*** PWM compare A interrupt :633 ***
*** PWM compare B interrupt :    0 ***
*** PWM compare A interrupt :633 ***
…
*** PWM Interrupt B interrupt:      0 ***

Half clock mode start
*** PWM Interrupt B interrupt:      0 ***
・・・
<<< PWM timer demonstration finish >>>
```

図 3.2　アドバンスドタイマ（T16A2）サンプルソフトウェア画面表示例

（2）サンプルソフトウェアの停止方法

上記“サンプルソフトウェア動作概要”に記載された全ての動作を終了すると、サンプルソフトウェアを終了してメニュー画面に戻ります。

3.2　LCD ドライバー（LCD32A）

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 LCD ドライバー(LCD32A)を使用して以下の動作を行います。

- 全点灯と全消灯を行います。
- 任意の COM/SEG の点灯/消灯を行います。
- LCD 表示の白黒反転を行います。
- LCD のコントラストを変更して、4 階調表示を行います。
- Gray-scale 機能を使用して、4 階調表示を行います。

3.2.2　ハードウェア条件

本サンプルソフトウェアは OSC1 と OSC3 に水晶振動子またはセラミック振動子が接続された状態で動作します。

振動子の接続方法につきましては、各テクニカルマニュアル「クロックジェネレータ（CLG）発振回路」をご参照下さい。

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。

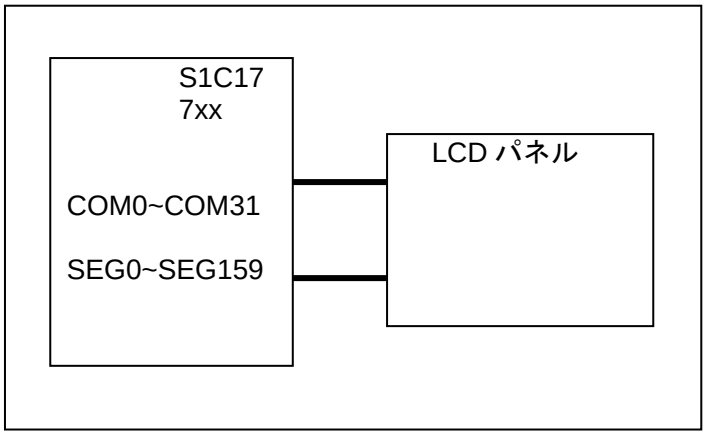

図 3.3　LCD ドライバー（LCD32A）サンプルソフトウェアハードウェア接続図

3.2.3　動作概要

（1）サンプルソフトウェア動作概要

- メニューから「1」を入力して ENTER キーを押下すると LCD を全点灯します。
- メニューから「2」を入力して ENTER キーを押下すると LCD を全消灯します。
- メニューから「3」を入力して ENTER キーを押下すると SEG/COM 番号入力待ちになり、「(SEG 番号)、(COM 番号)」を入力して ENTER キーを押下すると指定されたセグメントを点灯します。
- メニューから「4」を入力して ENTER キーを押下すると SEG/COM 番号入力待ちになり、「(SEG 番号)、(COM 番号)」を入力して ENTER キーを押下すると指定されたセグメントを消灯します。
- メニューから「5」を入力して ENTER キーを押下すると市松模様を表示し、一定時間間隔で LCD の白黒反転を 2 回実施します。
- メニューから「6」を入力して ENTER キーを押下すると LCD32A 割り込みを利用して LCD の 4 階調表示を行います。(1,5 行目は黒、2,6 行目は 75%グレー、3,7 行目は 50%グレー、4,8 行目は 25%グレーにします。)
- メニューから「7」を入力して ENTER キーを押下すると Gray-scale 機能を利用して LCD の 4 階調表示を行います。(1 行目は Gray-scale-0、2 行目は Gray-scale-1、3 行目は Gray-scale-2、4 行

目は Gray-scale-3 にします。）
- メニューから「8」を入力して ENTER キーを押下すると漢字の表示を行います。
- メニューから「9」を入力して ENTER キーを押下するとサンプルプログラムを終了します。

```
<<< LCD driver demonstration start >>>
1.All on                2.All off
3.Turn dot on           4.Turn dot off
5.Reverse               6.Grayscale1
7.Grayscale2            8.Kanji
9.exit

<<< LCD driver demonstration finish >>>
```

図 3.4　LCD ドライバ（LCD32A）サンプルソフトウェア画面表示例

（2）サンプルソフトウェアの停止方法

メニューから「9」を入力して ENTER キーを押下することで、サンプルプログラムを終了してメニュー画面に戻ります。

3.3　電源電圧検出回路（SVD2）

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電源電圧検出回路（以下、SVD2 回路）を使用して以下の動作を行います。

- SVD2 回路を用いて電源電圧の検出を行います。

3.3.2　ハードウェア条件

本サンプルソフトウェアは OSC1 と OSC3 に水晶振動子またはセラミック振動子が接続された状態で動作します。

任意の電源電圧を設定して動作させてください。

3.3.3　動作概要

（1）サンプルソフトウェア動作概要

- SVD2 回路を用いて電源電圧（VDD）の検出を行い、現在の VDD 電圧を SimI/O に表示します。比較電圧は 1.8V～3.2V です。
- 電源電圧値 1.8V 未満、3.2V 以上の場合、SimI/O に「SVD interrupt did not occur」と表示されます。

```
<<< SVD demonstration start >>>
Vdd=2.5V
<<< SVD demonstration finish >>>
```

図 3.5　電源電圧検出回路（SVD2）サンプルソフトウェア画面表示例

注 検出電圧は機種によって変更がある場合があります。各機種のソースコードを参照してご確認下さい。

（2）サンプルソフトウェアの停止方法

上記“サンプルソフトウェア動作概要”に記載された全ての動作を終了すると、サンプルソフトウェアを終了してメニュー画面に戻ります。

3.4　サウンドジェネレータ（SND）

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 のノーマルモードを使用して、周波数を上げながら出力した波形を BZOUT 端子に出力します。

3.4.2　ハードウェア条件

本サンプルソフトウェアは OSC1 と OSC3 に水晶振動子またはセラミック振動子が接続された状態で動作します。

振動子の接続方法につきましては、各テクニカルマニュアル「クロックジェネレータ（CLG）発振回路」をご参照下さい。

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。

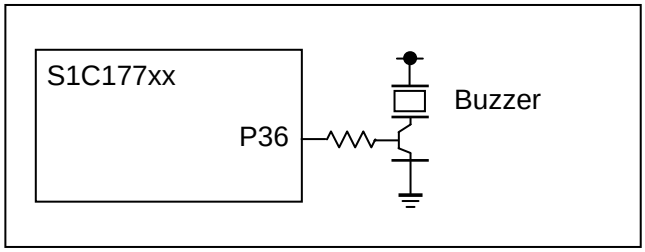

図 3.6　サウンドジェネレータ（SND）サンプルソフトウェアハードウェア接続図

3.4.3　動作概要

（1）サンプルソフトウェア動作概要

- 計時タイマ割り込みを開始します。
- ブザーを周波数「1170.8Hz」で出力します。
- CPU を HALT 状態にします。
- 計時タイマ割り込みが発生したら HALT 状態が解除され、ブザーを停止します。
- ブザー周波数を上げブザーを 1 秒間出力して HALT 状態に戻ります。ブザー周波数が「4096.0Hz」になり出力するまで繰り返します。（Simulated I/O にブザー周波数を表示します。）
- ブザー周波数の「4096.0Hz」出力が終了したら、計時**タイマ**を終了させて、サンプルプログラムを終了します。

```
<<<Sound Generator demonstration start >>>
***Buzzer Frequency : 1170.8Hz***
***Buzzer Frequency : 1365.3Hz***
…
***Buzzer Frequency : 4096.0Hz***
<<< Sound Generator demonstration finish >>>
```

図 3.7　サウンドジェネレータ（SND）サンプルソフトウェア画面表示例

（2）サンプルソフトウェアの停止方法

上記“サンプルソフトウェア動作概要”に記載された全ての動作を終了すると、サンプルソフトウェアを終了してメニュー画面に戻ります。

3.5　リアルタイムクロック（RTC2）

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します。
- リアルタイムクロックの時刻を取得します。
- リアルタイムクロックの時刻を設定します。
- リアルタイムクロックの割り込み回数を表示します。

3.5.2　ハードウェア条件

本サンプルソフトウェアはOSC1とOSC3に水晶振動子またはセラミック振動子が接続された状態で動作します。

振動子の接続方法につきましては、各テクニカルマニュアル「クロックジェネレータ（CLG）発振回路」をご参照下さい。

3.5.3　動作概要

（1）サンプルソフトウェア動作概要

- プログラム開始後、RTC2 サンプルプログラムのメニューが表示されます。
- メニューから「1」を入力して ENTER キーを押下することで RTC2 の時刻を取得し、24 時間モードで表示します。
- メニューから「2」を入力して ENTER キーを押下することで、RTC2 の時刻を設定します。
- メニューから「3」を入力して ENTER キーを押下することで、RTC2 割り込みの回数を表示します。
- メニューから「4」を入力して ENTER キーを押下することで、RTC2 サンプルプログラムを終了します。

```
<<< Real Time Clock demonstration start >>>
1.get RTC                          2.set RTC
3.indicate the count of interrupt  4.exit
Please input number.
> 1
10/03/01(Mon) 10:00:00

1.get RTC                          2.set RTC
3.indicate the count of interrupt  4.exit
・・・
> 4
<<< Real Time Clock demonstration finish >>>
```

図 3.8　リアルタイムクロック（RTC）サンプルソフトウェア画面表示例

（2）サンプルソフトウェアの停止方法

メニューから「4」を入力して ENTER キーを押下することで、サンプルプログラムを終了してメニュー画面に戻ります。

3.6　電流計測

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電流を計測するために以下の状態で CPU を実行するためのプログラムです。

- CPU を sleep モードにする。
- OSC1 のみを発振させた状態で CPU を halt モードにする。
- OSC1/OSC3 を発振させた状態で CPU を halt モードにする。
- OSC1/OSC3/IOSC を発振させた状態で CPU を halt モードにする

3.6.2　ハードウェア条件

本サンプルソフトウェアは OSC3 に水晶振動子またはセラミック振動子が接続された状態で動作します。

振動子の接続方法につきましては、S1C177xx シリーズ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。

3.6.3　実行方法

本サンプルソフトウェアは以下の手順でビルドすることで使用可能となります。

（1）プログラムの変更

機種別の定義ヘッダを変更します。

例として S1C17706 の場合は s1c17706_peripheral.h を変更します。

（2）変更箇所

ファイル下部にある以下の箇所を変更します。

```
//#undef   PE_SND
//#undef PE_SLEEP_HALT
//#undef   PE_VD1
//#undef PE_CURRENT_MEASURE

/*****************************************************************************/
/* Current measurement mode selecting                                        */
/*****************************************************************************/
// Comment out "#undef         PE_CURRENT_MEASURE", when using current measure program.
#ifdef PE_CURRENT_MEASURE
#define CURRENT_MEASURE_SLEEP                                   (0x00)
#define CURRENT_MEASURE_HALT_OSC1                               (0x01)
#define CURRENT_MEASURE_HALT_OSC1_OSC3                          (0x02)
#define CURRENT_MEASURE_HALT_OSC1_OSC3_IOSC                     (0x03)
// Remove the comment on the selected measurement mode.
//#define CURRENT_MEASURE_MODE CURRENT_MEASURE_SLEEP
//#define CURRENT_MEASURE_MODE CURRENT_MEASURE_HALT_OSC1
#define CURRENT_MEASURE_MODE CURRENT_MEASURE_HALT_OSC1_OSC3
//#define CURRENT_MEASURE_MODE CURRENT_MEASURE_HALT_OSC1_OSC3_IOSC
#endif
```

図 3.9　電流測定サンプルソフトウェアのための定義変更例

例えば OSC1/OSC3 を発振させた状態で CPU を halt モードにする場合、

“#undef PE_CURRENT_MEASURE”の定義を無効にし、

“#define CURRENT_MEASURE_MODE CURRENT_MEASURE_HALT_OSC1_OSC3”の定義を有効にし

てビルドを行います。

（3）デバッガによるプログラムのロード

デバッガを起動し、サンプルソフトウェアを S1C177xx へロードします。

（4）プログラムの実行

ICDmini からの電流の影響を除外するため、ICDmini を開発ボードから外し S1C177xx をリセットしてください。

（5）動作概要

- リセット後、電流計測サンプルソフトウェアが実行されます。

別の状態で電流計測を行いたい場合は、（2）～（4）の手順を行ってください。

（6）サンプルソフトウェアの停止方法

停止方法はありません。開発ボードへの電源供給を止めてください。

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の一覧を記述します。

4.1 アドバンスドタイマ（T16A2）

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 t16a2.c を参照して下さい。

表 4.1 アドバンスドタイマ（T16A2）サンプルドライバ関数一覧

関数名	機能名
T16A2_setInputClock	入力クロック設定
T16A2_controlInputClock	入力クロック供給許可/禁止設定
T16A2_init	アドバンスドタイマ初期化
T16A2_setTimerMode	アドバンスドタイマモード設定
T16A2_setComparatorCapture	コンパレータ/キャプチャ設定
T16A2_getCounterData	カウンターデータ取得
T16A2_setCompareData	コンペアデータ設定
T16A2_getCaptureData	キャプチャデータ取得
T16A2_resetTimer	アドバンスドタイマリセット
T16A2_setTimerRun	アドバンスドタイマ開始/停止設定
T16A2_initInt	アドバンスドタイマ割り込み初期化
T16A2_controlInt	アドバンスドタイマ割り込み許可/禁止設定
T16A2_resetIntFlag	アドバンスドタイマ割り込み要因フラグリセット
T16A2_checkIntFlag	アドバンスドタイマ割り込み要因フラグチェック

本サンプルドライバは t16a2.c と t16a2.h、t16a2_api.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_api.h をインクルードして下さい。

4.2 LCD ドライバ（LCD）

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 lcd.c を参照して下さい。

表 4.2 LCD ドライバ（LCD）サンプルドライバ関数一覧

関数名	機能名
LCD_initPower	LCD 電源初期化
LCD_controlBooster	電源電圧昇圧回路設定
LCD_setRegulatorSource	LCD 系定電圧回路の電源電圧選択
LCD_init	LCD 初期化
LCD_setSEGAssignment	SEG 端子メモリ割り当て設定
LCD_setCOMAssignment	COM 端子メモリ割り当て設定
LCD_setDisplayArea	LCD 表示領域設定
LCD_setDisplayReverse	LCD 表示白黒反転設定
LCD_controlDisplay	LCD 表示制御
LCD_setContrast	LCD コントラスト設定
LCD_display1Seg	1 セグメント表示
LCD_initInt	LCD 割り込み初期化
LCD_controlInt	LCD 割り込み許可/禁止設定
LCD_resetIntFlag	LCD 割り込み要因フラグリセット
LCD_checkIntFlag	LCD 割り込み要因フラグチェック
LCD_setCOMPinAssignment	COM 端子ピン割り付け設定
LCD_displayDraw	矩形表示
LCD_setLCDClock	LCD クロック設定
LCD_controlLCDClock	LCD クロック供給許可/禁止設定
LCD_allClear	LCD 表示エリア全クリア

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

4.3 LCD ドライバ（LCD32A）

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 lcd32a.c を参照して下さい。

表 4.3 LCD ドライバ（LCD32A）サンプルドライバ関数一覧

関数名	機能名
LCD32A_initPower	LCD 電源初期化
LCD32A_controlBooster	電源電圧昇圧回路設定
LCD32A_setRegulatorSource	LCD 系定電圧回路の電源電圧選択
LCD32A_init	LCD32A 初期化
LCD32A_setSEGAssignment	SEG 端子メモリ割り当て設定
LCD32A_setCOMAssignment	COM 端子メモリ割り当て設定
LCD32A_setDisplayArea	LCD32A 表示領域設定
LCD32A_setDisplayReverse	LCD32A 表示白黒反転設定
LCD32A_controlDisplay	LCD32A 表示制御
LCD32A_setContrast	LCD32A コントラスト設定
LCD32A_controlGrayscale	グレースケール機能設定
LCD32A_display1Seg	1 セグメント表示
LCD32A_initInt	LCD32A 割り込み初期化
LCD32A_controlInt	LCD32A 割り込み許可/禁止設定
LCD32A_resetIntFlag	LCD32A 割り込み要因フラグリセット
LCD32A_checkIntFlag	LCD32A 割り込み要因フラグチェック
LCD32A_setCOMPinAssignment	COM 端子ピン割り付け設定
LCD32A_displayDraw	矩形表示
LCD32A_controlComDrive	COM 端子駆動許可/禁止設定
LCD32A_controlSegDrive	SEG 端子駆動許可/禁止設定
LCD32A_setLCDClock	LCD32A クロック設定
LCD32A_controlLCDClock	LCD32A クロック供給許可/禁止設定
LCD32A_allClear	LCD32A 表示エリア全クリア

本サンプルドライバは lcd32a.c と lcd32a.h、lcd32a_api.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 lcd32a_api.h をインクルードして下さい。

4.4 電流電圧検出回路（SVD2）

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 svd2.c を参照して下さい。

表 4.4 電源電圧検出回路（SVD2）サンプルドライバ関数一覧

関数名	機能名
SVD2_setCompareVoltage	SVD2 比較電圧設定
SVD2_setCompareMode	SVD2 比較モード/回数設定
SVD2_setResetMode	SVD2 リセットモード設定
SVD2_controlDetection	SVD2 検出開始/停止設定
SVD2_setDetectionResult	SVD2 検出結果取得
SVD2_initInt	SVD2 割り込み初期化
SVD2_controlInt	SVD2 割り込み許可/禁止設定
SVD2_resetIntFlag	SVD2 割り込み要因フラグリセット
SVD2_checkIntFlag	SVD2 割り込み要因フラグチェック
SVD2_setSVD2Clock	SVD2 クロック設定
SVD2_controlSVD2Clock	SVD2 クロック供給許可/禁止設定

本サンプルドライバは svd2.c と svd2.h、svd2_api.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 svd2_api.h をインクルードして下さい。

4.5 サウンドジェネレータ（SND）

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 snd.c を参照して下さい。

表 4.5 サウンドジェネレータ（SND）サンプルドライバ関数一覧

関数名	機能名
SND_init	SND の初期化
SND_controlSND	SND ブザー開始/停止設定
SND_setSNDMode	SND モード設定
SND_setSNDQuality	SND 周波数設定
SND_setSNDDuty	SND 音量設定
SND_setSNDClock	SND クロック設定
SND_controlSNDClock	SND クロック供給許可/禁止設定

本サンプルドライバは snd.c と snd.h、snd_api.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_api.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（RTC2）

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 rtc2.c を参照して下さい。

表 4.6 リアルタイムクロック（RTC2）サンプルドライバ関数一覧

関数名	機能名
RTC2_init	RTC の初期化
RTC2_setTimerRun	RTC 開始/停止設定
RTC2_setTimerMode	RTC24/12H モード設定
RTC2_setAdj	RTC の 30 秒補正設定
RTC2_setTime	RTC の時刻を設定
RTC2_getTime	RTC の時刻を取得
RTC2_checkBusy	RTC のビジーチェック
RTC2_initInt	RTC 割り込みを初期化
RTC2_controlInt	RTC 割り込みを許可/禁止
RTC2_resetIntFlag	RTC2 割り込み要因フラグをリセット
RTC2_checkIntFlag	RTC2 割り込み要因フラグをチェック
RTC2_get1HzCounter	RTC2 1Hz カウンターを取得

本サンプルドライバは rtc2.c と rtc2.h、rtc2_api.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 rtc2_api.h をインクルードして下さい。

4.7 電源制御回路（VD1）

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 vd1.c を参照して下さい。

表 4.7 電源制御回路（VD1）サンプルドライバ関数一覧

関数名	機能名
VD1_setMode	重負荷モードの設定

本サンプルドライバは vd1.c と vd1.h、vd1_api.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 vd1_api.h をインクルードして下さい。

4.8 マルチプレクサ（MUX）

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 mux.c を参照して下さい。

表 4.8 マルチプレクサ（MUX）サンプルドライバ関数一覧

関数名	機能名
MUX_init	MUX 初期化
MUX_setREMCport	REMC ポート設定
MUX_setADCport	ADC ポート設定
MUX_setSPIport	SPI ポート設定
MUX_setUARTport	UART ポート設定
MUX_setRFCport	RFC ポート設定
MUX_setI2CMport	I2C マスタポート設定
MUX_setI2CSport	I2C スレーブポート設定
MUX_setLCDport	LCD ポート設定
MUX_setDBGport	デバッグポート設定
MUX_setCLGport	CLG ポート設定
MUX_setT16Aport	アドバンスドタイマポート設定
MUX_setT16A2port	アドバンスドタイマ（T16A2）ポート設定
MUX_setSNDport	サウンドジェネレータ（SND）ポート設定

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

4.9 ウォッチドッグタイマ（WDT）

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 wdt.c を参照して下さい。

表 4.9 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
WDT_resetTimer	ウォッチドッグタイマリセット
WDT_setTimerRun	ウォッチドッグタイマ開始/停止設定
WDT_setTimerMode	ウォッチドッグチタイマモード設定
WDT_checkNMI	ウォッチドッグタイマ NMI 発生チェック
WDT_setCountClock	ウォッチドッグタイマクロック設定
WDT_controlCountClock	ウォッチドッグタイマクロック供給許可/禁止設定

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

改訂履歴表

付－1

コードNo.	ページ	改訂内容（旧内容を含む） および改訂理由
412023900	全ページ	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本部　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：412023900
2010 年 10 月 作成