

S1C17 Family Application Note

S1C17 シリーズ

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。

©SEIKO EPSON CORPORATION 2017, All rights reserved.

目 次

1. 概要.....	1
1.1 動作環境	1
2. サンプルソフトウェア構成.....	2
2.1 ディレクトリ構成及びファイル構成	2
3. 実行方法	4
3.1 ソフトウェア開発環境	4
3.2 各種ツールのインストール	4
3.3 プロジェクトのインポート	6
3.3.1 サンプルソフトウェアのインポート（GNU17 version 2 / version 3 共通）	6
3.3.2 サンプルソフトウェアのインポート（GNU17 version 2）	7
3.3.3 サンプルソフトウェアのインポート（GNU17 version 3）	11
3.4 ターゲットとの接続.....	15
3.5 ICDmini とターゲットの操作	15
3.5.1 ICDmini ver2.0 の場合	15
3.5.2 ICDmini ver3.0 の場合	16
3.6 ビルドとデバッガの起動.....	17
3.6.1 GNU17 version 2 の場合	17
3.6.2 GNU17 version 3 の場合	18
4. サンプルソフトウェア機能詳細.....	19
4.1 機種、機能対応	19
4.2 周辺回路の初期設定	21
4.2.1 init_config.h 設定例	21
4.2.2 init_config.h 注意事項	22
4.3 入出力ポート（PPORT）	23
4.3.1 サンプルプログラム仕様	23
4.3.2 ハードウェア条件	23
4.3.3 動作概要	23
4.4 クロックジェネレータ（CLG）	24
4.4.1 サンプルプログラム仕様	24
4.4.2 ハードウェア条件	24
4.4.3 動作概要	24
4.5 16ビットタイマ（T16）	25
4.5.1 サンプルプログラム仕様	25
4.5.2 ハードウェア条件	25
4.5.3 動作概要	25
4.6 16ビットPWM タイマ（T16B）	26
4.6.1 サンプルプログラム仕様	26
4.6.2 ハードウェア条件	26
4.6.3 動作概要	26
4.7 リアルタイムクロック（RTCA）、（RTCA2）	27
4.7.1 サンプルプログラム仕様	27
4.7.2 ハードウェア条件	27
4.7.3 動作概要	27
4.8 ウォッチドッグタイマ（WDT）、（WDT2）	28

4.8.1	サンプルプログラム仕様	28
4.8.2	ハードウェア条件	28
4.8.3	動作概要	28
4.9	UART (UART)、(UART2)、(UART3)	29
4.9.1	サンプルプログラム仕様	29
4.9.2	ハードウェア条件	29
4.9.3	動作概要	30
4.9.3.1	マスターサンプル動作概要	30
4.9.3.2	スレーブサンプル動作概要	30
4.9.4	キャリブレーション機能	30
4.10	SPIA (SPIA)	31
4.10.1	サンプルプログラム仕様	31
4.10.2	ハードウェア条件	31
4.10.3	動作概要	32
4.10.3.1	マスターサンプル動作概要	32
4.10.3.2	スレーブサンプル動作概要	32
4.11	I2C (I2C)	33
4.11.1	サンプルプログラム仕様	33
4.11.2	ハードウェア条件	33
4.11.3	動作概要	34
4.11.3.1	マスターサンプル動作概要	34
4.11.3.2	スレーブサンプル動作概要	34
4.12	LCD ドライバー (LCD4A)	35
4.12.1	サンプルプログラム仕様	35
4.12.2	ハードウェア条件	35
4.12.3	動作概要	35
4.13	LCD ドライバー (LCD8A)	36
4.13.1	サンプルプログラム仕様	36
4.13.2	ハードウェア条件	36
4.13.3	動作概要	36
4.14	LCD ドライバー (LCD8B)	37
4.14.1	サンプルプログラム仕様	37
4.14.2	ハードウェア条件	37
4.14.3	動作概要	37
4.15	LCD ドライバー (LCD16A)	38
4.15.1	サンプルプログラム仕様	38
4.15.2	ハードウェア条件	38
4.15.3	動作概要	38
4.16	LCD ドライバー (LCD24A)	39
4.16.1	サンプルプログラム仕様	39
4.16.2	ハードウェア条件	39
4.16.3	動作概要	39
4.17	LCD ドライバー (LCD32B)	40
4.17.1	サンプルプログラム仕様	40
4.17.2	ハードウェア条件	40
4.17.3	動作概要	40
4.18	7セグメント LED コントロール回路 (LEDC)	41
4.18.1	サンプルプログラム仕様	41
4.18.2	ハードウェア条件	41
4.18.3	動作概要	41
4.19	R/F 変換機 (RFC)	42
4.19.1	サンプルプログラム仕様	42
4.19.2	ハードウェア条件	42

4.19.3 動作概要	42
4.20 サウンドジェネレータ (SNDA)	43
4.20.1 サンプルプログラム仕様	43
4.20.2 ハードウェア条件	43
4.20.3 動作概要	43
4.21 電源電圧検出回路 (SVD)、(SVD3)	44
4.21.1 サンプルプログラム仕様	44
4.21.2 ハードウェア条件	44
4.21.3 動作概要	44
4.22 パワージェネレータ (PWG)、(PWG2)	45
4.22.1 サンプルプログラム仕様	45
4.22.2 ハードウェア条件	45
4.22.3 動作概要	45
4.23 IR リモートコントローラ (REMC)、(REMC2)、(REMC3)	46
4.23.1 サンプルプログラム仕様	46
4.23.2 ハードウェア条件	46
4.23.3 動作概要	47
4.24 ソフトウェア IR リモートコントローラ (SOFT REMC)	48
4.24.1 サンプルプログラム仕様	48
4.24.2 ハードウェア条件	48
4.24.3 動作概要	48
4.25 12 ビット A/D 変換器 (ADC12)	49
4.25.1 サンプルプログラム仕様	49
4.25.2 ハードウェア条件	49
4.25.3 動作概要	49
4.26 オペアンプ/コンパレータ (OPCMP)	50
4.26.1 サンプルプログラム仕様	50
4.26.2 ハードウェア条件	50
4.26.3 動作概要	50
4.27 温度センサ/基準電圧生成回路 (TSRVR)	51
4.27.1 サンプルプログラム仕様	51
4.27.2 ハードウェア条件	51
4.27.3 動作概要	51
改訂履歴表	52

1. 概要

本マニュアルは S1C17 シリーズ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。

S1C17 シリーズサンプルソフトウェアは S1C17 シリーズ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

S1C17 シリーズ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。

- S1C17 シリーズ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、GNU17v2.4.0/GNU17v3.1.0 で動作確認を行っています。

注 GNU17 及び ICDmini には、複数のバージョンが存在します。ご使用になる GNU17 で使用可能な ICDmini については S5U1C17001C Manual を参照して下さい。

2. サンプルソフトウェア構成

2. サンプルソフトウェア構成

本章では S1C17 シリーズサンプルソフトウェアのファイル構成を記載します。

S1C17 シリーズ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 S1C17 シリーズサンプルソフトウェアのディレクトリ構成を示します。

注 xxx：機種名

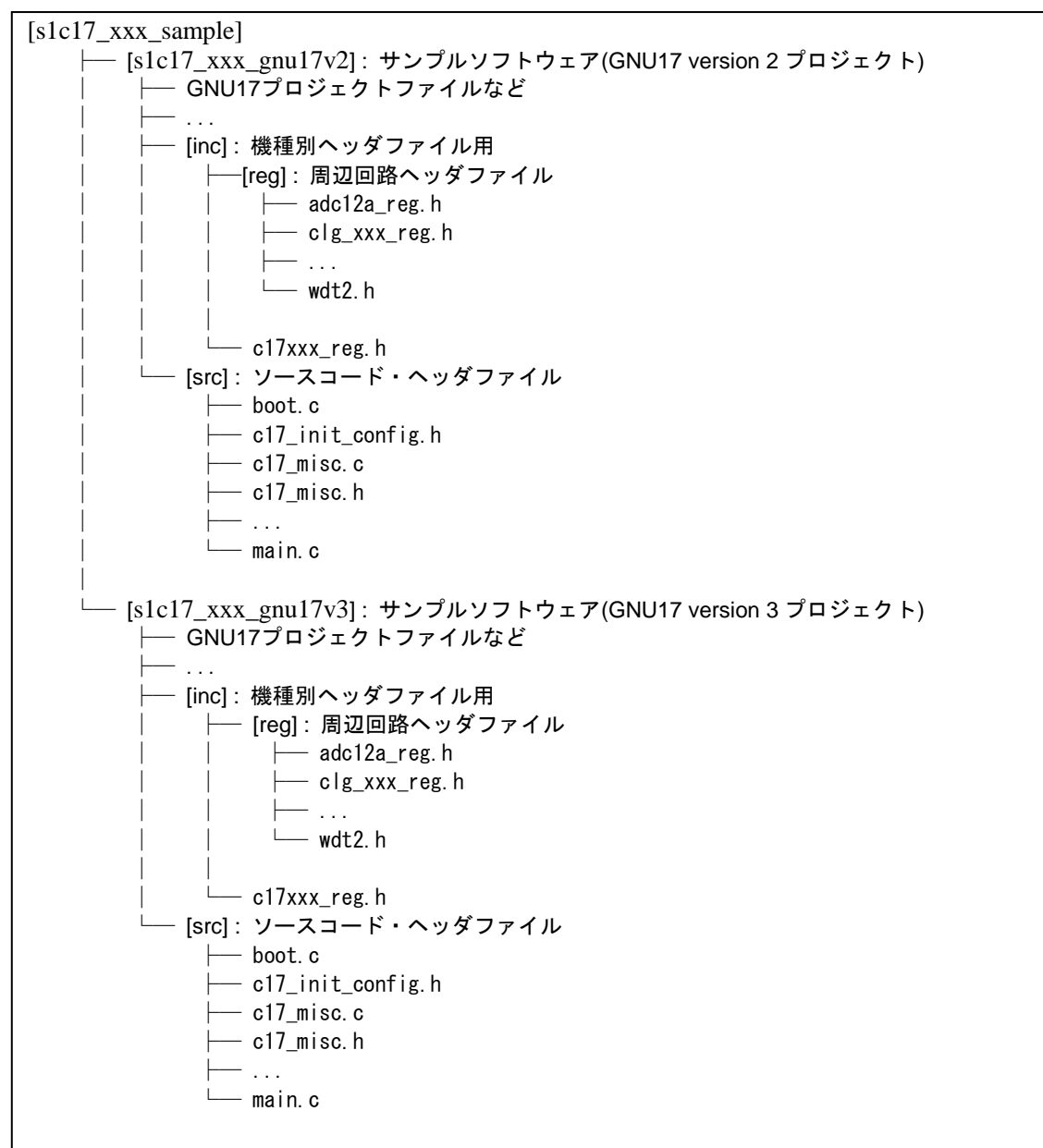


図 2.1.1 S1C17 シリーズサンプルソフトウェアディレクトリ構成図

(1) s1c17_xxx_sample ディレクトリ

サンプルソフトウェアが格納されているディレクトリです。

(2) s1c17_xxx_gnu17vx ディレクトリ

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です。

ディレクトリ名の最後の_gnu17v2/_gnu17v3 が GNU17 のバージョンを示しています。

(3) inc ディレクトリ

S1C17 シリーズのレジスタなど機種に依存する情報を定義したヘッダファイルが格納されているディレクトリです。

- S1C17xxx のレジスタアドレスなどを定義したヘッダファイル (c17xxx_reg.h)
- 使用する S1C17 シリーズを選択するためのヘッダファイル (c17_mcu_select.h)

(4) 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。

- 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_xxx_reg.h など)

(5) src ディレクトリ

マイコンの初期化处理、周辺回路毎のサンプルプログラム及び、サンプルドライバ、またそれら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。

- ベクターテーブルや初期化处理のファイル (boot.c)
- 周辺回路毎のサンプルプログラムファイル (main_clg.c や main_port.c)
- 周辺回路毎のサンプルドライバファイル (c17_misc.c や c17_clg.c など)
- 周辺回路毎のサンプルドライバのヘッダファイル (c17_misc.h や c17_clg.h など)
- 周辺回路への設定を行うためのヘッダファイル (c17_init_config.h)

3. 実行方法

3. 実行方法

3.1 ソフトウェア開発環境

ソフトウェアの開発環境について記載します。本書では、以下のツールを使用した方法について記載します。各ツールの詳細については、各ツールのマニュアルを参照して下さい。

- S5U1C17001C (GNU17) ver.2.4.0/ver.3.1.0
- 機種別情報ファイル (S1C17 シリーズ)
- S5U1C17001H (ICDmini) ver.2.0/ver3.0

3.2 各種ツールのインストール

(1) GNU17 version 2 / version 3 (S5U1C17001C)

- GNU17 の圧縮ファイルをエプソン マイコン ユーザーズサイトよりダウンロードして下さい。
http://www.epson.jp/prod/semicon/products/micro_controller/16bit/sw_tool.htm
- 入手した GNU17 の圧縮ファイルを任意のフォルダに解凍して下さい。
- Setup.exe を実行し、インストールを行ってください。

(2) 機種別情報ファイルのインストール

- 各機種の機種別情報ファイルをエプソン マイコン ユーザーズサイトよりダウンロードして下さい。
http://www.epson.jp/prod/semicon/products/micro_controller/16bit/sw_tool.htm
- 入手した機種別情報ファイルを解凍し、GNU17 をインストールしたフォルダの[mcu_model]フォルダに上書きコピーして下さい。
例：S1C17M01 の場合

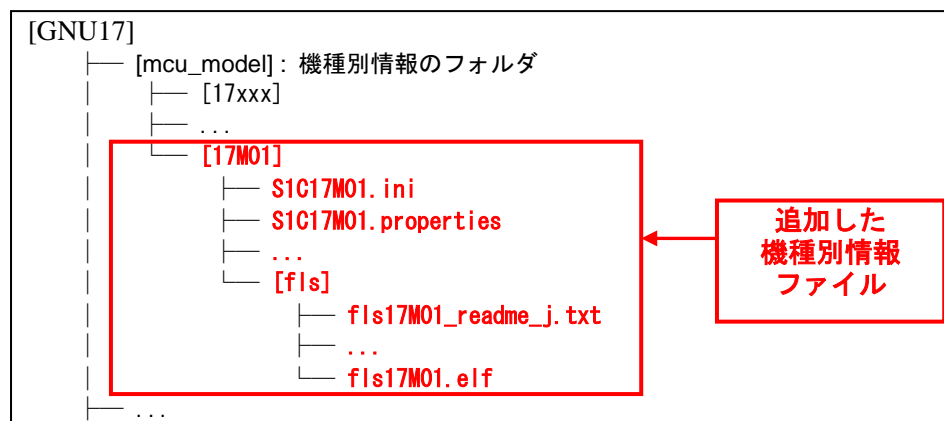


図 3.2.1 S1C17M01 の機種別情報ファイルの追加

(3) ICDmini ver2.0 (S5U1C17001H)

- GNU17 version 2 をインストール後、ICDmini と PC を USB ケーブルにて接続して下さい。
- 以下の場所から、USB ドライバーをインストールして下さい。USB ドライバーは、GNU17 をインストールしたフォルダにあります。以下に GNU17 を C:\EPSON\GNU17 にインストールした場合のフォルダ構成を記載します。
32bit OS の場合：C:\EPSON\GNU17\utility\drv_usb\OS_32bit
64bit OS の場合：C:\EPSON\GNU17\utility\drv_usb\OS_64bit

(4) ICDmini ver3.0 (S5U1C17001H)

- GNU17 version 3 をインストール後、ICDmini と PC を USB ケーブルにて接続して下さい。
- 以下の場所から、USB ドライバーをインストールして下さい。USB ドライバーは、GNU17 をインストールしたフォルダにあります。GNU17 を C:\EPSON\GNU17V3 にインストールした場合には、以下のフォルダとなります。

C:\EPSON\GNU17V3\utility\drv_usb\Mini3Driver

3. 実行方法

3.3 プロジェクトのインポート

サンプルソフトウェアを GNU17 にインポートする方法について記載します。

注 プロジェクトのインポートの詳細については、S5U1C17001C Manual を参照して下さい。

注 GNU17 version 2 と GNU17 version 3 でプロジェクトのインポート方法が異なります。

3.3.1 サンプルソフトウェアのインポート（GNU17 version 2 / version 3 共通）

（1）GNU17 の起動

GNU17 をインストールしたディレクトリ内の eclipse ディレクトリにある、eclipse.exe のアイコンをダブルクリックし、IDE を起動させます。Windows のスタートアップメニューから [EPSON MCU] > [GNU17] > [GNU17 IDE]（GNU17 version 3 の場合、[EPSON MCU] > [GNU17V3] > [GNU17V3 IDE]）を選択しても起動します。

（2）サンプルソフトウェアの解凍

エプソンマイコン ユーザーズサイトよりダウンロードしたサンプルソフトウェアを解凍します。

（3）ヘッダファイルの変更

ターゲットとなる CPU に合わせて、サンプルソフトウェアのヘッダファイルを変更します。変更するファイルは [regC17_mcu_select.h] となります。[regC17_mcu_select.h] ファイルの以下の行を、ターゲットとなる CPU に変更して下さい。

例：S1C17M01 のサンプルソフトウェアを使用する場合

```
// Please activate macro of used MCU.
```

```
#define C17_MCUSEL_C17M01
```

```
//#define C17_MCUSEL_W22
```

```
//#define C17_MCUSEL_W23
```

例：S1C17W22 のサンプルソフトウェアを使用する場合

```
// Please activate macro of used MCU.
```

```
//#define C17_MCUSEL_M01
```

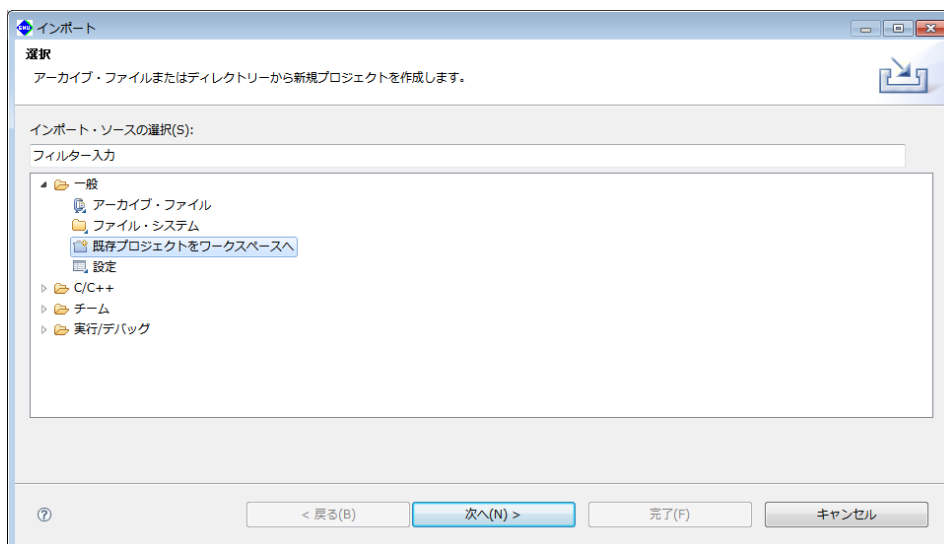
```
#define C17_MCUSEL_C17W22
```

```
//#define C17_MCUSEL_W23
```

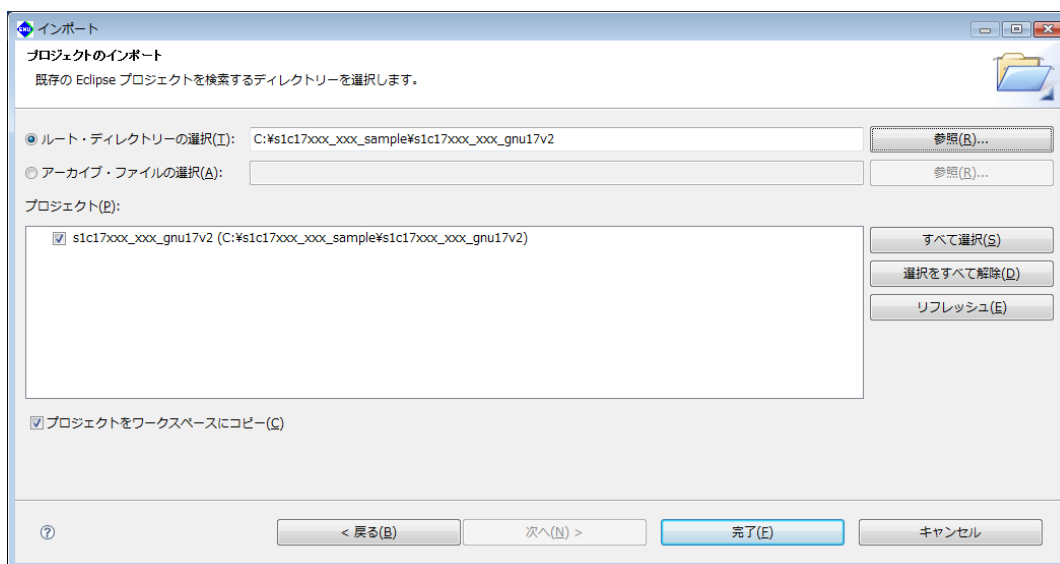
3.3.2 サンプルソフトウェアのインポート（GNU17 version 2）

（１）サンプルソフトウェアのインポート

GNU17 の[ファイル]メニューから[インポート...]を選択します。[インポート]ウィザードから、[一般]>[既存プロジェクトをワークスペースへ]を選択します。



[ルート・ディレクトリーの選択]の[参照...]ボタンで、3.3.1 の（２）で解凍したサンプルソフトウェアのフォルダ内の [xxx_gnu17v2] プロジェクトを選択します。

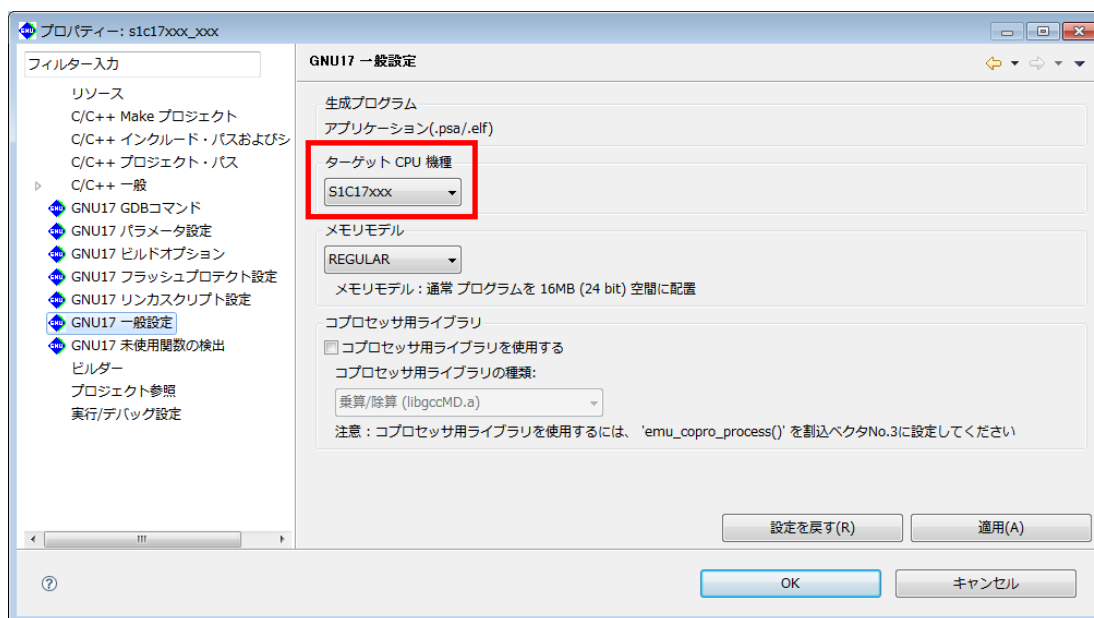


[プロジェクトをワークスペースにコピー]チェックボックスを選択します。これにより、プロジェクトのコピーがワークスペースディレクトリに作成され、オリジナルが変更されることはありません。最後に、[完了]ボタンを押します。GNU17にて、サンプルプロジェクトがインポートされたことを確認後、3.3.1 の（２）で解凍したフォルダは削除して下さい。

3. 実行方法

(2) ターゲットの変更

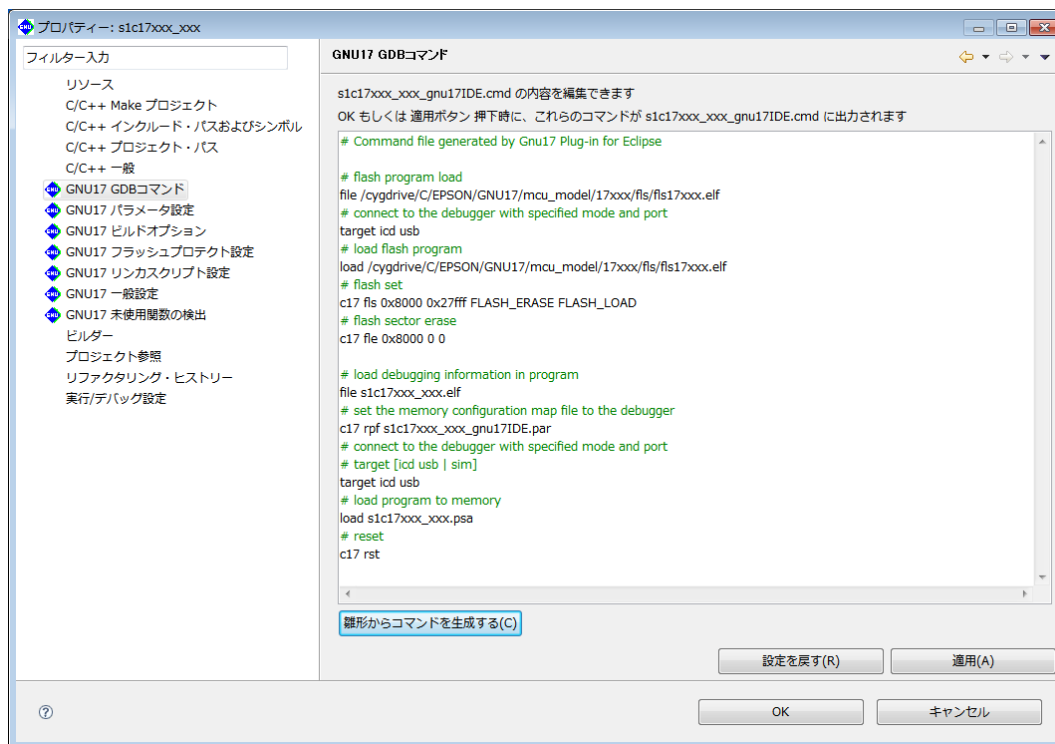
サンプルソフトウェアのターゲットを変更します。サンプルソフトウェアのプロジェクトを[C/C++プロジェクト]ビューで選択し、[プロジェクト]メニューから[プロパティ]を選択します。[プロパティ]ダイアログから、[GNU17 一般設定]を選択します。



[ターゲット CPU 機種]から、ターゲットとなる CPU を選択し、[OK]ボタンを押して下さい。
例えば、ターゲットとなる CPU が S1C17W22 の場合、[S1C17W22]に変更して下さい。

(3) デバッガ起動オプションの変更

デバッガ起動オプションを変更します。サンプルソフトウェアのプロジェクトを[C/C++プロジェクト]ビューで選択し、[プロジェクト]メニューから[プロパティ]を選択します。[プロパティ]ダイアログから、[GNU17 GDB コマンド]を選択します。



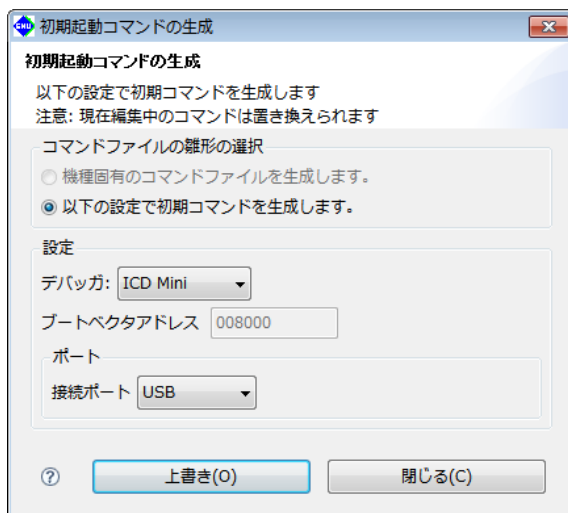
ここには、IDE が生成するデバッガ起動用コマンドファイルの内容が表示されます。デバッガは使用するマイコンや ICD などに合わせてモードに設定して動作させる必要があります。

3. 実行方法

デバッガ起動用コマンドファイルは、以下の手順により作成を行います。

[雛形からコマンドを生成する]ボタンをクリックして、[初期起動コマンドの生成]ダイアログボックスを表示させます。

[デバッガ:]コンボボックスから"ICD Mini"を選択します。



[上書き]ボタンをクリックし、その後表示される[現在のコマンドを上書き]ダイアログボックスでも[OK]ボタンをクリックします。

コマンドの修正や追加が必要な場合は、テキストボックス内で直接編集が可能です。

※：シミュレーテッド I/O を利用する場合の注意

シミュレーテッド I/O（シリアルインタフェースなどの外部入出力機能を標準入出力（stdin、stdout）あるいはファイルの入出力により評価すること）を使用する場合は、GDB コマンドの最後に以下の行を追加して下さい。

最後の行「c17 rst」の後に、以下の 3 行を追加する

u main

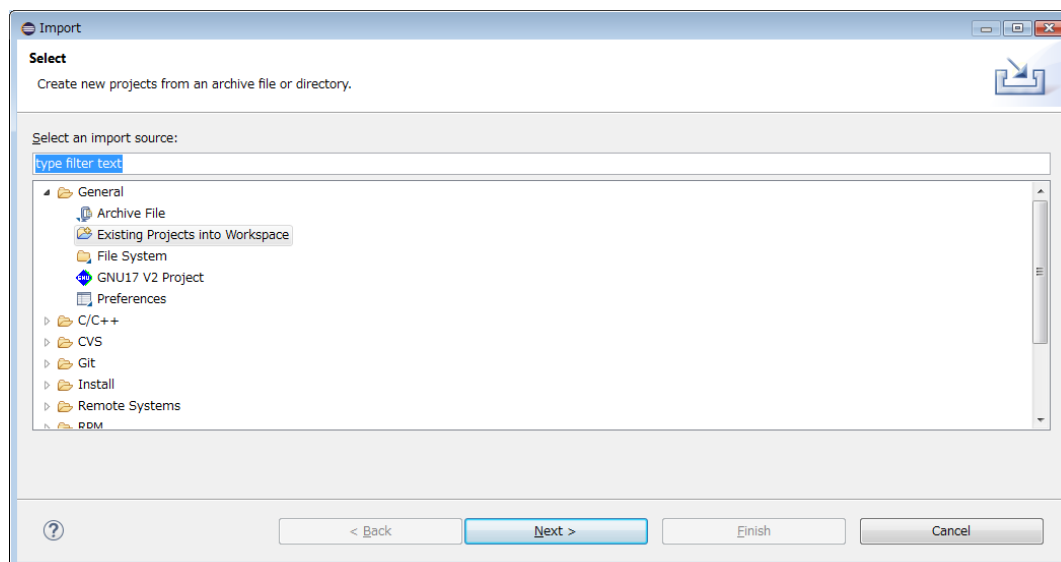
c17 stdout 1 WRITE_FLASH WRITE_BUF

c17 stdin 1 READ_FLASH READ_BUF

3.3.3 サンプルソフトウェアのインポート（GNU17 version 3）

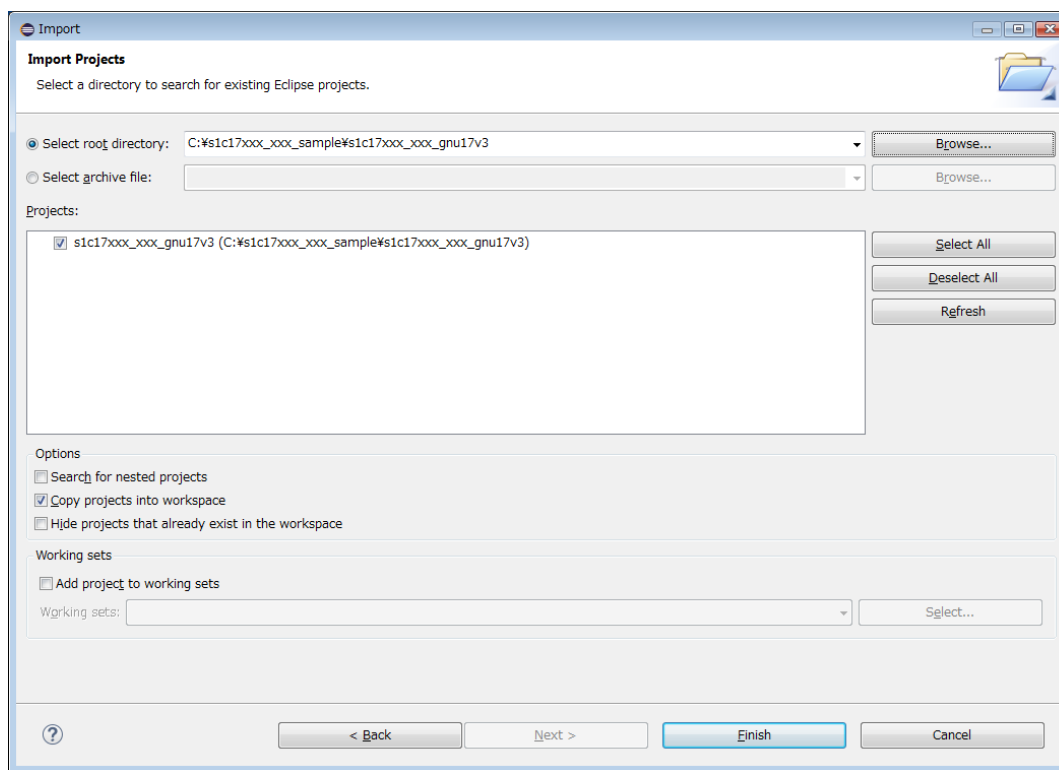
（１）サンプルソフトウェアのインポート

GNU17 の[File]メニューから[Import...]を選択します。[Import]ウィザードから、[General]>[Existing Projects into Workspace]を選択します。



3. 実行方法

[Select root directory]の[Browse...]ボタンで、3.3.1 の（２）で解凍したサンプルソフトウェアのフォルダ内の [xxx_gnu17v3] プロジェクトを選択します。

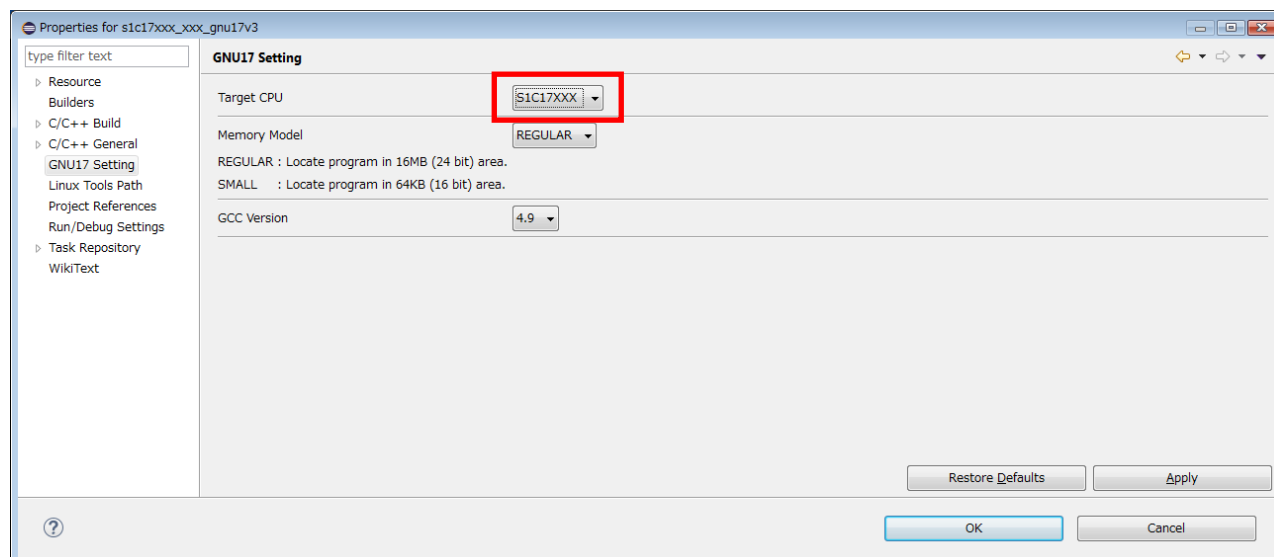


[Copy project into workspace]チェックボックスを選択します。これにより、プロジェクトのコピーがワークスペースディレクトリに作成され、オリジナルが変更されることはありません。最後に、[Finish]ボタンを押します。GNU17 にて、サンプルプロジェクトがインポートされたことを確認後、3.3.1 の（２）で解凍したフォルダは削除して下さい。

(2) ターゲットの変更

サンプルソフトウェアのターゲットを変更します。サンプルソフトウェアのプロジェクトを [Project Explorer] ビューで選択し、[Project]メニューから[Properties]を選択します。

[Properties]ダイアログから、[GNU17 Settings] を選択します。

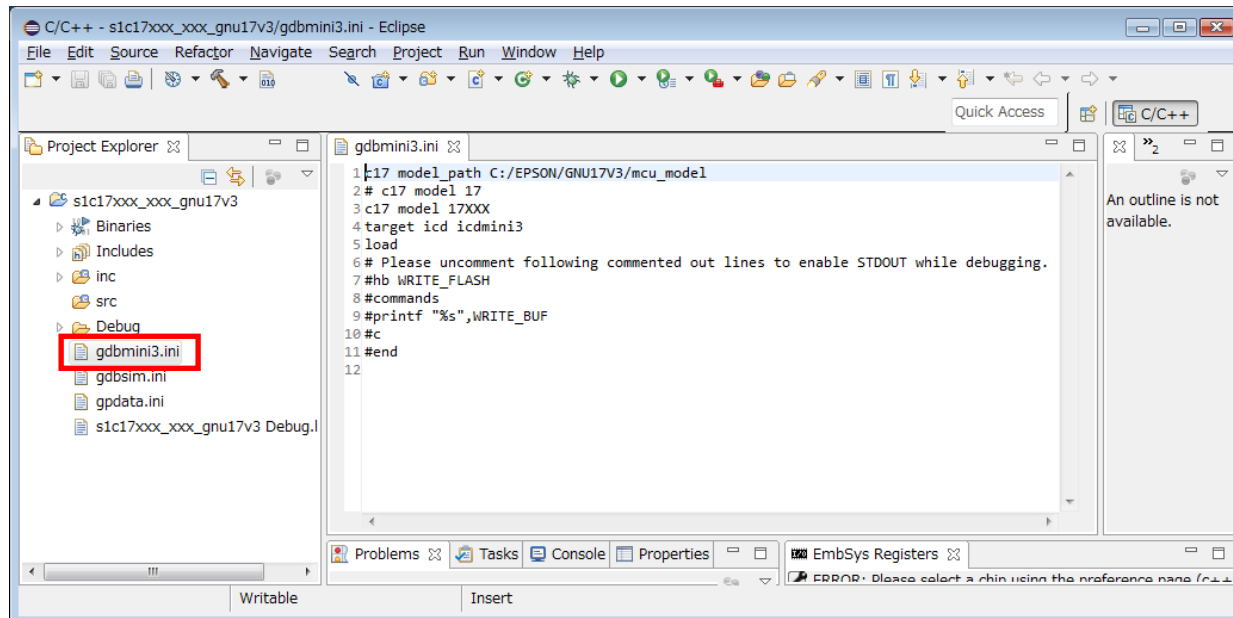


[Target CPU]から、ターゲットとなる CPU を選択し、[Apply]ボタンを押した後、[OK]ボタンを押して下さい。例えば、ターゲットとなる CPU が S1C17W22 の場合、[S1C17W22]に変更して下さい。

3. 実行方法

(3) デバッガ起動オプションの変更

デバッガ起動オプションを変更します。サンプルソフトウェアのプロジェクトを[[Project Explorer]ビューで選択し、プロジェクト内の[gdbmini3.ini]を選択してエディタで開きます。



ここには、デバッガ起動用コマンドファイルの内容が表示されます。デバッガは使用するマイコンやICDなどに合わせたモードに設定して動作させる必要があります。

※標準出力を利用する場合の注意

標準出力を使用するにはGDB コマンドファイル内の赤字の部分の#を外して有効にしてください。

GDB コマンドファイル [gdbmini3.ini]

Please uncomment following commented out lines to enable STDOUT while debugging.

hb WRITE_FLASH

commands

printf "%s",WRITE_BUF

c

end

3.4 ターゲットとの接続

GNU17 をインストールした PC と ICDmini を USB ケーブルで接続し、S1C17 シリーズを搭載したボードと ICDmini を接続して下さい。

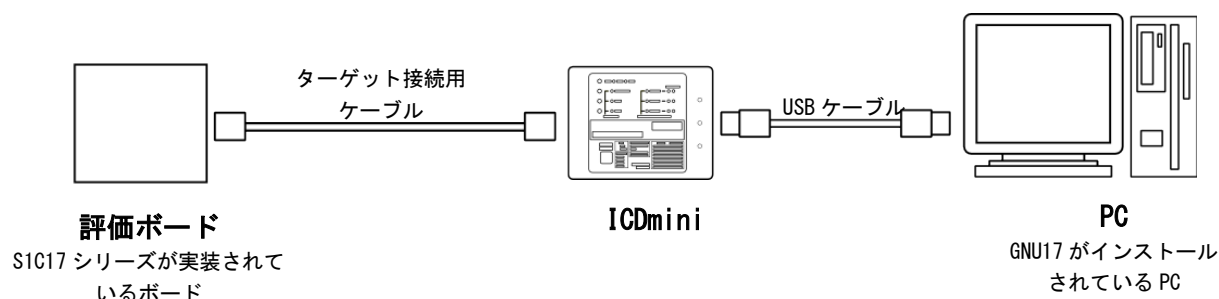


図 3.4.1 ターゲット、ICDmini、PC との接続図

3.5 ICDmini とターゲットの操作

3.5.1 ICDmini ver2.0 の場合

ターゲットと ICDmini、PC を接続後、以下の順序にリセットを行い、ターゲットと ICDmini の接続が確立していることを確認して下さい。なお、ICDmini の DIP Switch の設定は以下の通りです。

表 3.5.1 ICDmini の DIP Switch

DIP Switch	説明	S5U1C17M01T1 接続時	S5U1C17W23T 接続時
1	Target CPU (C17 / C33)	OPEN ↑	OPEN ↑
2	FLASH WRITER (OFF / WRITE)	OPEN ↑	OPEN ↑
3	FLASH WRITER (OFF / VERIFY)	OPEN ↑	OPEN ↑
4	DSIO LVL (OPEN / TRGET VCC)	OPEN ↑	ON ↓
5	DSIO LVL (3.3V / 1.8V)	OPEN ↑	OPEN ↑
6	CONNECTION DIAG (OFF / ON)	OPEN ↑	OPEN ↑
7	FW UPDATE MODE (OFF / ON)	OPEN ↑	OPEN ↑
8	FLASH VCC OUT (OFF / ON)	ON ↓	ON ↓

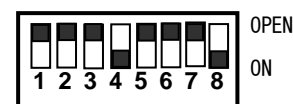


図 3.5.1.1 ICDmini の DIP Switch: S5U1C17M01T1 接続時[中央], S5U1C17W23T 接続時[右]

（１）ICDmini のリセット

ICDmini の DIP Switch を以下のように設定し、ICDmini の「RESET/START」ボタンを押して下さい。

3. 実行方法

い。

(2) ターゲットのリセット

ターゲットのリセットを行ってください。各ターゲットのリセット方法は、ターゲットのマニュアルを参照して下さい。

(3) ターゲットと ICDmini の接続確立の確認

ICDmini の LED が以下のように点灯していることを確認して下さい。

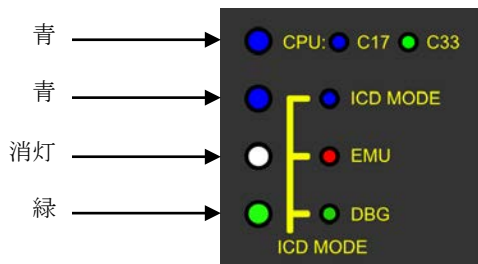


図 3.5.1.2 接続確立時の ICDmini の LED 表示

3.5.2 ICDmini ver3.0 の場合

ターゲットと ICDmini、PC を接続後、ターゲットと ICDmini の接続が確立していることを確認して下さい。

(1) ターゲットと ICDmini の接続確立の確認

ICDmini の LED が以下のように点灯していることを確認して下さい。

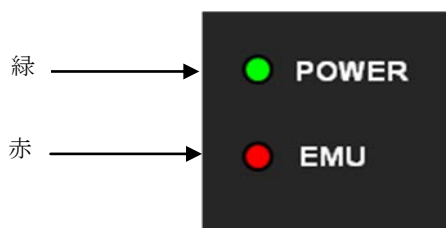


図 3.5.2.1 接続確立時の ICDmini の LED 表示

3.6 ビルドとデバッグの起動

サンプルソフトウェアのビルドを行い、デバッグを起動する方法について記載します。

- 注 ビルドとデバッグの起動の詳細については、S5U1C17001C Manual を参照して下さい。
- 注 クロック周りを操作中のプログラムを停止する場合、正常動作が期待できませんのでご注意ください。当該コードのデバッグにあたっては、S5U1C17001C Manual の DCLK 切替モードコマンドを参照してください。

3.6.1 GNU17 version 2 の場合

(1) ビルドの実行

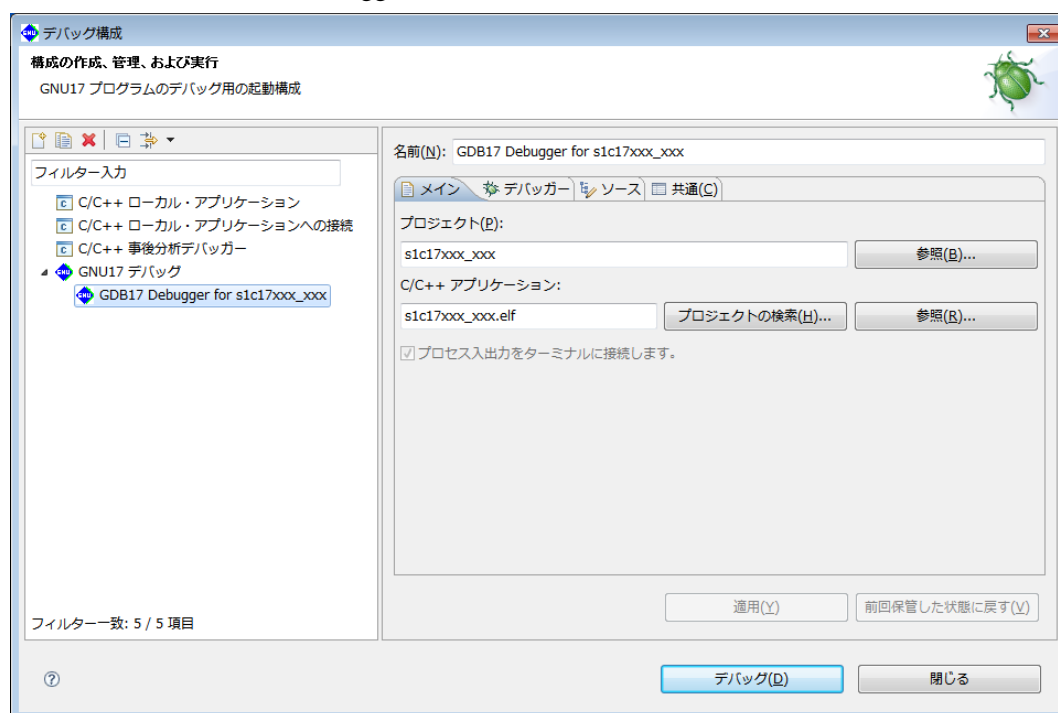
GNU17 の[C/C++ プロジェクト]ビューからビルドを実行したいプロジェクトを選択します。GNU17 の[プロジェクト]メニューから[プロジェクトのビルド]を選択し、ビルドを実行します。[C/C++ プロジェクト]ビューからビルドを実行したいプロジェクトを選択し、コンテキストメニュー（右クリックにより表示）の[プロジェクトのビルド]からも実行することができます。

(2) デバッグの起動構成ダイアログの表示

GNU17 の [実行]メニューから[デバッグの構成...]を選択し、起動構成ダイアログを表示させます。ツールバーの[デバッグ]ボタンのメニューからも表示させることができます。

(3) 起動するサンプルソフトウェアの選択

ツリーリストから[GDB17 Debugger for <プロジェクト名>]を選択します。



(4) デバッグの起動

[デバッグ]ボタンをクリックします。デバッグ gdb が起動して指定のコマンドファイルを実行します。

3. 実行方法

3.6.2 GNU17 version 3 の場合

(1) ビルドの実行

GNU17 の[Project Explorer]ビューからビルドを実行したいプロジェクトを選択します。GNU17 の[Project]メニューから[Build Project]を選択し、ビルドを実行します。[Project Explorer]ビューからビルドを実行したいプロジェクトを選択し、コンテキストメニュー（右クリックにより表示）の[Build Project]からも実行することができます。

(2) デバッガの起動構成ダイアログの表示

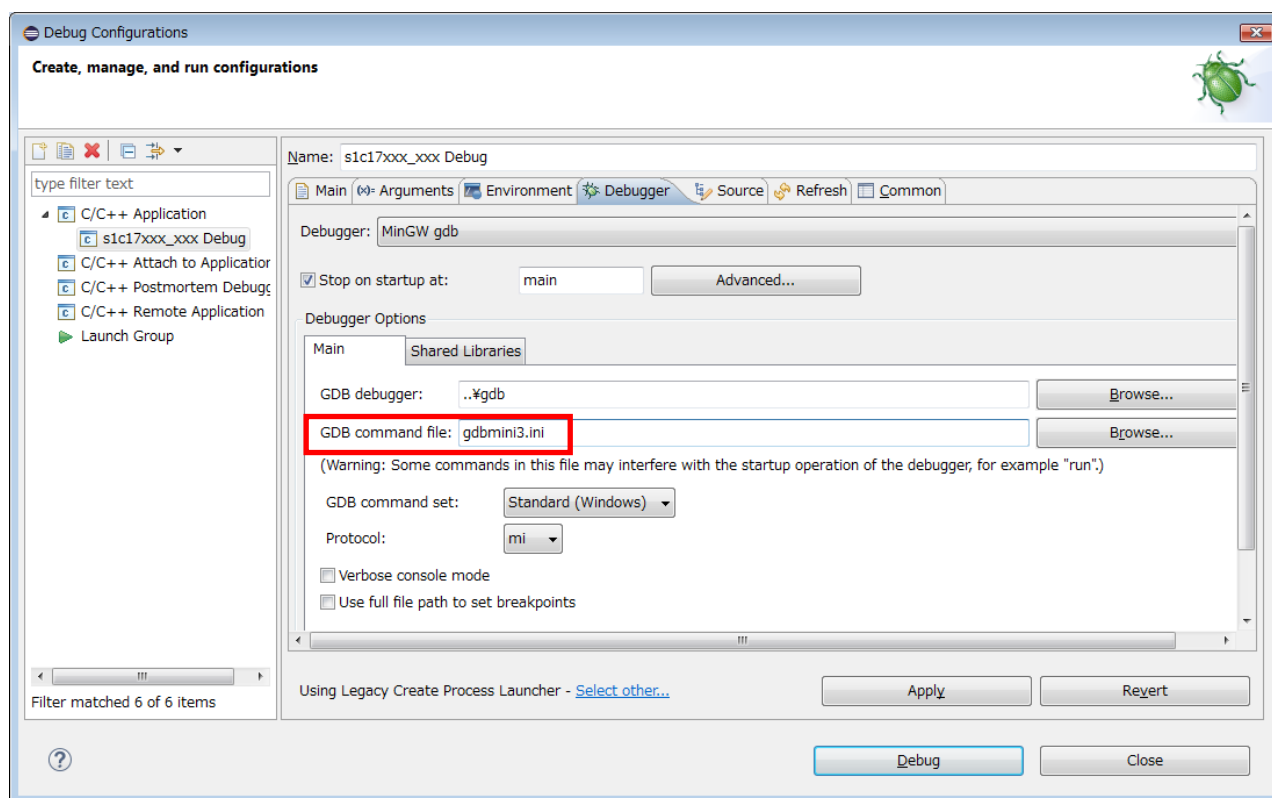
GNU17 の [Run]メニューから[Debug Configurations...]を選択し、Debug Configurations ダイアログを表示させます。ツールバーの[Debug]ボタンのメニューからも表示させることができます。

(3) 起動するサンプルソフトウェアの選択とデバッガの設定

[Debugger]タブの[GDB command file:]に[gdbmini3.ini]を指定します。

(サンプルでは、デフォルトで [gdbmini3.ini] が指定されています。)

設定後、[Apply]ボタンをクリックした後に[Debug]ボタンをクリックして指定のコマンドファイルを実行します。



4. サンプルソフトウェア機能詳細

本章では S1C17 シリーズサンプルソフトウェアの機能詳細について記載します。

4.1 機種、機能対応

機種とモジュールの対応一覧表を下記に示します。

各周辺回路をクリックすると、動作説明のページにジャンプします。

表 4.1.1 機種、機能対応表 S1C17W00 シリーズ

	周辺回路																		
機種名	CLG	PPOINT	WDT	RTC	T16	T16B	SVD	UART	SPI	I2C	SNDA	REMC	SOFT REMC	LCD	RFC	PWG	ADC	TSRVR	OPCMP
S1C17W03	○	○	○	○	○	○	○	○	○	○	○	○	—	—	○	○	○	—	—
S1C17W04	○	○	○	○	○	○	○	○	○	○	○	○	—	—	○	○	○	—	—
S1C17W12	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	—	—	—
S1C17W13	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	—	—	—
S1C17W14	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	—	—	—
S1C17W15	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	—	—	—
S1C17W16	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	—	—
S1C17W18	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17W22	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	—	—
S1C17W23	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	—	○
S1C17W34	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17W35	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17W36	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—

4. サンプルソフトウェア機能詳細

表 4.1.2 機種、機能対応表 S1C17M00 シリーズ

機種名	周辺回路																		
	CLG	PPORT	WDT	RTC	T16	T16B	SVD	UART	SPI	I2C	SNDA	REMC	SOFT REMC	LCD	RFC	PWG	ADC	TSRVR	OPCMP
S1C17M01	○	○	○	○	○	—	○	○	○	○	—	—	○	○	○	○	—	—	—
S1C17M10	○	○	○	○	○	○	○	○	○	○	—	—	○	○	—	○	—	—	—
S1C17M12	○	○	○	—	○	○	○	○	○	○	—	○	—	○	—	○	—	—	—
S1C17M13	○	○	○	—	○	○	○	○	○	○	—	○	—	○	—	○	○	—	—
S1C17M20	○	○	○	○	○	○	○	○	○	○	○	○	—	—	—	○	○	—	—
S1C17M21	○	○	○	○	○	○	○	○	○	○	○	○	—	—	—	○	○	—	—
S1C17M22	○	○	○	○	○	○	○	○	○	○	○	○	—	—	○	○	○	—	—
S1C17M23	○	○	○	○	○	○	○	○	○	○	○	○	—	—	—	○	○	—	—
S1C17M24	○	○	○	○	○	○	○	○	○	○	○	○	—	—	—	○	○	—	—
S1C17M25	○	○	○	○	○	○	○	○	○	○	○	○	—	—	○	○	○	—	—
S1C17M30	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17M31	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17M32	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17M33	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—
S1C17M34	○	○	○	○	○	○	○	○	○	○	○	○	—	○	○	○	○	○	—

4.2 周辺回路の初期設定

本サンプルソフトウェアは[src¥c17_init_config.h]ファイルを使用して以下の動作を行います。

- マクロで周辺回路の設定値を決め、c17_init_xxx()関数を実行すると 初期設定が行われます。
- 割り込みレベルや周辺回路設定を変更する際に[src¥c17_init_config.h]ファイルを修正してください。
- (#define C17_PE_xxx)が有効になっているモジュールをアクティブにしてビルドします。

4.2.1 init_config.h 設定例

- 入出力ポートサンプルプログラムを実行する場合

s1c17xxx_sample を解凍したフォルダにある[src]フォルダ配下の c17_init_config.h を編集します。

(1) ポートモジュールの有効化

#define C17_PE_PORT を有効にします。

例：ポートサンプルプログラムの有効化する場合

```
#define C17_PE_PORT
```

例：ポートサンプルプログラムの無効化する場合

```
// #define C17_PE_PORT
```

(2) PORT 割り込みレベルの変更

PORT 割り込みのレベルを変更します。[src¥c17_init_config.h]ファイルの

#define INTERRUPT_LEVEL_PORT を、変更して下さい。

例：割り込みレベルを 4 に変更する場合

```
#define INTERRUPT_LEVEL_PORT (4) // Set PORT interrupt level 0 to 7
```

(3) P00 ポート入出力設定の変更

P00 ポートの入出力設定を変更します。[src¥c17_init_config.h]ファイルの

#define PORT00_INIT PORT_INIT 内の引数(disable, input, chatta_disable, none)を変更して下さい。

第 1 引数：P00 の設定を行うか否か。

第 2 引数：P00 の Input or Output 設定

第 3 引数：P00 のチャタリングフィルター機能の有効、無効を設定。

第 4 引数：P00 の PullUp or PullDown or 無しの設定。

例：P00 = OUTPUT, chattering filter = Disable, pull-up = None にする場合

```
#define PORT00_INIT PORT_INIT(enable, output, chatta_disable, none)
```

4. サンプルソフトウェア機能詳細

4.2.2 init_config.h 注意事項

(1) 複数機種サンプルの場合

デフォルトの機種でサポートしているモジュールとの間に違いがある場合があります。

例：s1c17m01_w22_w23_sample_gnu17v2

#define C17_PE_LCD8A を有効化しても W23 設定時は使用できません。

(2) 通信系の MASTER、SLAVE は排他的です。

MASTER と SLAVE の同時使用は出来ません。

必ずモジュールの一方を有効化したら、もう一方は無効化して下さい。

4.3 入出力ポート (PPORT)

4.3.1 サンプルプログラム仕様

入出力ポートサンプルプログラムは入出力ポートを使用して以下の動作を行います。

- ポートを入力に設定し、入力信号が **High** レベルまたは **Low** レベルになったとき、割り込みが発生します。
- ポートを出力に設定し、**High** レベルまたは **Low** レベルの信号を出力します。

使用するポート設定とポート名は以下の通りです。

表 4.3.1.1 入出力ポート設定一覧

設定	ポート名
出力ポート	P01
入力割り込みポート	P02

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、PORT(#define C17_PE_PORT)をアクティブにしてください。

4.3.2 ハードウェア条件

本サンプルプログラムはマイコンの各ポートを以下のように接続してご使用下さい。

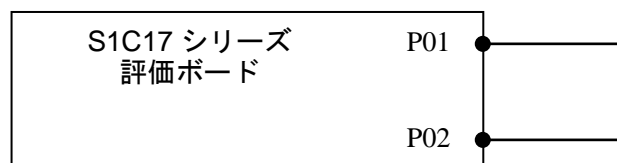


図 4.3.2.1 入出力ポートサンプルプログラムハードウェア接続図

4.3.3 動作概要

1. ポートを初期化します。
2. P01 ポートを出力ポートに設定して **Low** レベルを出力します。
3. P02 ポートを入力ポートに設定し、**Low** レベルから **High** レベルにて割り込みが発生するように設定します。
4. P01 ポートから **High** レベルの信号を出力します。
5. P02 ポートの割り込みにて P02 ポートが **High** レベルであることを確認します。
6. P02 ポートを **High** レベルから **Low** レベルにて割り込みが発生するように設定します。
7. P01 ポートから **Low** レベルの信号を出力します。
8. P02 ポートの割り込みにて P02 ポートが **Low** レベルであることを確認します。

4. サンプルソフトウェア機能詳細

4.4 クロックジェネレータ (CLG)

4.4.1 サンプルプログラム仕様

発振回路サンプルプログラムは、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。

- OSC1 の発振を行う (OSC1 が存在する機種のみ)。
- OSC3 の発振を行う (OSC3 が存在する機種のみ)。発振周波数の設定も可能。
- IOSC の発振を行う。 (IOSC が存在する機種のみ)。発振周波数の設定も可能。
- システムクロックを OSC1 へ切り替える (OSC1 が存在する機種のみ)。
- システムクロックを OSC3 へ切り替える (OSC3 が存在する機種のみ)。
- システムクロックを IOSC へ切り替える (IOSC が存在する機種のみ)。
- 各システムクロック変更時、FOUT へ対象クロックを出力する。
- HALT モード、SLEEP モード間の動作モードの遷移を行う。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)をアクティブにしてください。

4.4.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振を使用して動作開始した後、OSC1、OSC3、IOSC に切り替える動作を行います。従って、発振のために外付け振動子が必要な場合があります。詳細については、S1C17 シリーズのテクニカルマニュアル「クロックジェネレータ (CLG)」を参照して下さい。

表 4.4.2.1 PORT 割り付け対応表

機種名	FOUT
S1C17M01	P32
S1C17M10	P04
S1C17M12/M13	P16
S1C17M20/M21/ M22/ M23/ M24/ M25	P13
S1C17M30/M31/M32/M33/M34	P10
S1C17W03/W04	P06
S1C17W12/W13	P10
S1C17W14/W16	P34
S1C17W15	P16
S1C17W18	P16
S1C17W22/W23	P15
S1C17W34/W35/W36	P12

4.4.3 動作概要

1. 存在する発振回路を発振開始し、16 ビットタイマを初期化します。
2. 現在の確認クロックを FOUT へ出力します。
3. CPU を停止 (HALT) し、1 秒毎に 16 ビットタイマ割り込みが起こる動作を 2 回繰り返します。
4. CPU を停止 (SLEEP) し、1 秒毎に 16 ビットタイマ割り込みが起こる動作を 2 回繰り返します。
5. 上記 1、2、3 の動作を IOSC、OSC1、OSC3 で行います。(機種ごとの存在する発振回路)
6. 最後に FOUT 出力を停止させ、一連の動作を終了します。

4.5 16 ビットタイマ (T16)

4.5.1 サンプルプログラム仕様

16 ビットタイマサンプルプログラムは 16 ビットタイマを使用して以下の動作を行います。

- 16 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)をアクティブにしてください。

4.5.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

4.5.3 動作概要

1. 16 ビットタイマのカウンターを 1 秒毎に 16 ビットタイマ割り込みが起こるように設定します。
2. 16 ビットタイマの開始し、CPU を Halt 状態にします。
3. 16 ビットタイマの割り込みが発生すると CPU は Halt 状態を解除します。
4. 16 ビットタイマの割り込み回数を取得します。
5. 16 ビットタイマ割り込み回数が 10 回未満のときは、再び CPU を Halt 状態にします。
6. 16 ビットタイマ割り込み回数が 10 回以上のときは、16 ビットタイマを停止し、サンプルプログラムを終了します。

4. サンプルソフトウェア機能詳細

4.6 16 ビット PWM タイマ (T16B)

4.6.1 サンプルプログラム仕様

16 ビット PWM タイマサンプルプログラムは 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 タイマのコンパレータ割り込みを発生させる。
- 16 ビット PWM タイマのキャプチャ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。
- TOUT へ PWM 出力を行う。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、T16B(#define C17_PE_T16B)をアクティブにしてください。

4.6.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

表 4.6.2.1 PORT 割り付け対応表

機種名	TOUT
S1C17 シリーズ	P00

4.6.3 動作概要

1. 16 ビット PWM タイマを以下のように設定し、T16 ビット PWM タイマを開始します。
 - モード：リピータアップカウントモード
 - カウンター最大値：0x5000
 - コンパレータ/キャプチャ回路 0：コンパレータモード(コンペアバッファ：0x2000)
 - コンパレータ/キャプチャ回路 1：キャプチャモード(トリガ信号：LOW)キャプチャ割り込み回数
 - TOUT 出力許可
2. 16 ビット PWM タイマを開始し、CPU を Halt 状態にします。
3. 16 ビット PWM タイマの割り込みが発生すると CPU は Halt 状態を解除します。
4. 16 ビット PWM タイマの割り込みが発生したとき、以下の割り込み回数を取得します
 - コンペア割り込み回数
 - キャプチャ割り込み回数
 - カウント MAX 割り込み回数
 - カウンタゼロ割り込み回数
5. コンペア割り込みが発生したとき、キャプチャ用トリガ信号を HIGH に設定します。
6. カウント MAX 割り込みが発生したとき、キャプチャ用トリガ信号を LOW に設定します。
7. カウント MAX 割り込み回数が 10 回未満のときは、再び CPU を Halt 状態にします。
8. カウント MAX 割り込み回数が 10 回以上のときは、16 ビットタイマを停止し、割り込みが発生したときのカウンタ値を保存し、サンプルプログラムを終了します。

4.7 リアルタイムクロック (RTCA)、(RTCA2)

4.7.1 サンプルプログラム仕様

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時刻を設定する。
- 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割り込み回数を取得する。
- 論理緩急機能を使用する。
- ストップウォッチを使用する。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)をアクティブにしてください。

4.7.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.7.3 動作概要

1. リアルタイムクロックを初期化します。
2. リアルタイムクロックの 1 秒割り込み、1/2 秒割り込みを有効にします。
3. リアルタイムクロックの論理緩急機能（論理緩急実行間隔：4,096 秒、補正率：-58.2ppm）を設定します。
4. リアルタイムクロックの日時を「2016 年 12 月 31 日(土) 23 時 59 分 51 秒」に設定し、24 時間モードに設定します。
5. リアルタイムクロックを開始し、CPU を Halt 状態にします。
6. リアルタイムクロックの割り込みが発生すると CPU は Halt 状態を解除します。
7. リアルタイムクロックの 1 秒割り込みの回数を取得します。
8. 1 秒割り込みの回数が 10 回未満のときは、再び CPU を Halt 状態にします。
9. 1 秒割り込みの回数が 10 回以上のときは、12 時間モードに設定し、リアルタイムクロックの日時を取得します（取得した値は、「2017 年 1 月 1 日(日) AM 12 時 00 分 01 秒」になります）。
10. リアルタイムクロックの 1 秒割り込みを無効にします。
11. リアルタイムクロックの開始から 4,096 秒間隔で、論理緩急を実行します。
12. ストップウォッチを初期化します。
13. ストップウォッチの割り込みを有効にします。
14. ストップウォッチを開始し、CPU を Halt 状態にします。
15. ストップウォッチの割り込みが発生すると CPU は Halt 状態を解除します。
16. ストップウォッチ 1Hz 割り込みの回数を取得します。
17. ストップウォッチ 1Hz 割り込みの回数が 10 回未満のときは、再び CPU を Halt 状態にします。
18. ストップウォッチ 1Hz 割り込みの回数が 10 回以上のときは、ストップウォッチカウント(10Hz 桁、100Hz 桁)を読み出します。
19. ストップウォッチを停止し、ストップウォッチの割り込みを無効にします。

4. サンプルソフトウェア機能詳細

4.8 ウォッチドッグタイマ (WDT)、(WDT2)

4.8.1 サンプルプログラム仕様

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リセット割り込みを発生させる。

本サンプルプログラムを使用する場合は T16(`#define C17_PE_T16`)、WDT(`#define C17_PE_WDT`)をアクティブにして下さい。

4.8.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1 (32.768kHz) または、内蔵発振回路で動作します。

4.8.3 動作概要

1. ウォッチドッグタイマと 16 ビットタイマを開始します。
2. 16 ビットタイマの 1 秒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. 16 ビットタイマの 1 秒割り込みが 10 回発生すると 16 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リセット割り込みが発生すると本サンプルプログラムの先頭(`boot.c`)に処理が移ります。

4.9 UART (UART)、(UART2)、(UART3)

4.9.1 サンプルプログラム仕様

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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、
UART(#define C17_PE_UART_MASTER or #define C17_PE_UART_SLAVE)をアクティブにしてください。

4.9.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。
本サンプルプログラムは、UART マスターサンプルプログラムと UART スレーブサンプルプログラムが動作している評価ボードを接続し、スレーブから起動して下さい。各ポートは以下のように接続してご使用下さい。

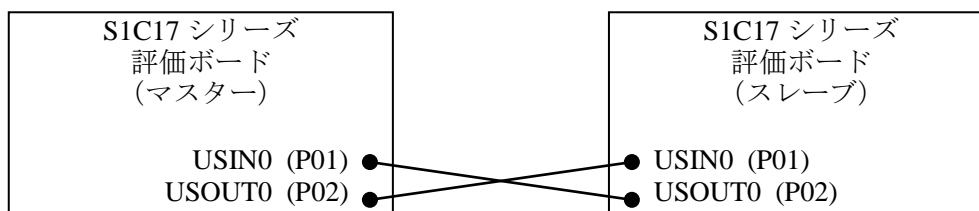


図 4.9.2.1 UART サンプルプログラムハードウェア接続図(デフォルト割り付け)

4. サンプルソフトウェア機能詳細

4.9.3 動作概要

4.9.3.1 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。
 - データ長 : 8bit
 - ストップビット : 1bit
 - パリティ : 無し
2. 「A(0x41)」から「Z(0x5A)」までのデータと「¥0」をスレーブに送信する。
3. スレーブから 27 バイトのデータを受信する。
4. 受信したデータが、送信したデータと同じか比較し、サンプルプログラムを終了します。

4.9.3.2 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。
 - データ長 : 8bit
 - ストップビット : 1bit
 - パリティ : 無し
2. マスターから 27 バイトのデータを受信する。
3. 「A(0x41)」から「Z(0x5A)」までのデータと「¥0」をマスターに送信する。
4. 受信したデータが、送信したデータと同じか比較し、サンプルプログラムを終了します。

4.9.4 キャリブレーション機能

本サンプルプログラムでは UART の動作クロックに IOSC を選択した場合、16 ビットタイマ CH.0 を使用したキャリブレーション機能を使用することができます。

16 ビットタイマの動作クロックには OSC1 または OSC3 を設定して下さい。

16 ビットタイマの動作クロックに OSC3 を選択した場合、32kHz に近づくよう分周比を設定して下さい。

1. c17_init_config.h の以下を変更します。
 - #define IOS_CALIBRATION_ENABLE : enable = 1
 - #define IOS_CALIBRATION_CLOCK : OSC1 or OSC3 の周波数[Hz]
 - #define T16CH0_INIT T16_INIT : enable = 1
: OSC1 or OSC3
: OSC1 = CLKDIV_1、OSC3 = 32kHz に近い値
: mode = oneshot
2. c17_uartX.c の以下を変更します。
 - baudrateCalibration 関数 発振開始 : OSC1 = c17startClgOSC1
: OSC3 = c17startClgExtOSC3
 - baudrateCalibration 関数 発振停止 : OSC1 = c17stopClgOSC1();
: OSC3 = c17stopClgOSC3();

4.10 SPIA (SPIA)

4.10.1 サンプルプログラム仕様

SPIA サンプルプログラムは SPIA を使用して以下の動作を行います。

- SPI を使いデータを送信する。
- SPI を使いデータを受信する。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、
SPIA(#define C17_PE_SPIA_MASTER or #define C17_PE_SPIA_SLAVE)をアクティブにしてください。

4.10.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。
本サンプルプログラムは、SPIA マスターサンプルプログラムと SPIA スレーブサンプルプログラムが動作している評価ボードを接続し、スレーブから起動して下さい。各ポートは以下のように接続してご使用下さい。

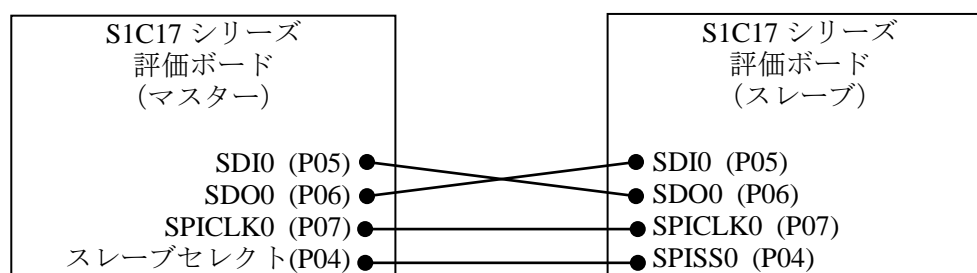


図 4.10.2.1 SPI マスター、スレーブサンプルプログラムハードウェア接続図(デフォルト割り付け)

表 4.10.2.1 PORT 例外割り付け対応表

機種名	SDI0	SDO0	SPICLK0	SPISS0 or SS
S1C17W15	P15	P16	P17	P14
S1C17M20/M21/M22/M23/M24/M25	P01	P02	P03	P00

4. サンプルソフトウェア機能詳細

4.10.3 動作概要

4.10.3.1 マスターサンプル動作概要

1. SPI ポートを以下の値に初期化します。
 - データ長 : 8bit
 - データフォーマット : MSB 先頭
2. 「A(0x41)」から「Z(0x5A)」までのデータと「¥0」をスレーブに送信する。
3. スレーブから 27 バイトのデータを受信する。
4. 受信したデータが、送信したデータと同じか比較し、サンプルプログラムを終了します。

4.10.3.2 スレーブサンプル動作概要

1. SPI ポートを以下の値に初期化します。
 - データ長 : 8bit
 - データフォーマット : MSB 先頭
2. マスターから 27 バイトのデータを受信する。
3. 「A(0x41)」から「Z(0x5A)」までのデータと「¥0」をマスターに送信する。
4. 受信したデータが、送信したデータと同じか比較し、サンプルプログラムを終了します。

4.11 I2C (I2C)

4.11.1 サンプルプログラム仕様

I2C サンプルプログラムは I2C を使用して以下の動作を行います。

- I2C を使いデータを送信する。
- I2C を使いデータを受信する。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、
I2C(#define C17_PE_I2C_MASTER or #define C17_PE_I2C_SLAVE)をアクティブにしてください。

4.11.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。
本サンプルプログラムは、I2C マスターサンプルプログラムと I2C スレーブサンプルプログラムが動作している評価ボードを接続し、スレーブから起動して下さい。デバッグ時は、スレーブ側のプログラムを先に走らせてください。又、両評価ボードの GND は共通としてください。各ポートは以下のように接続してご使用下さい。

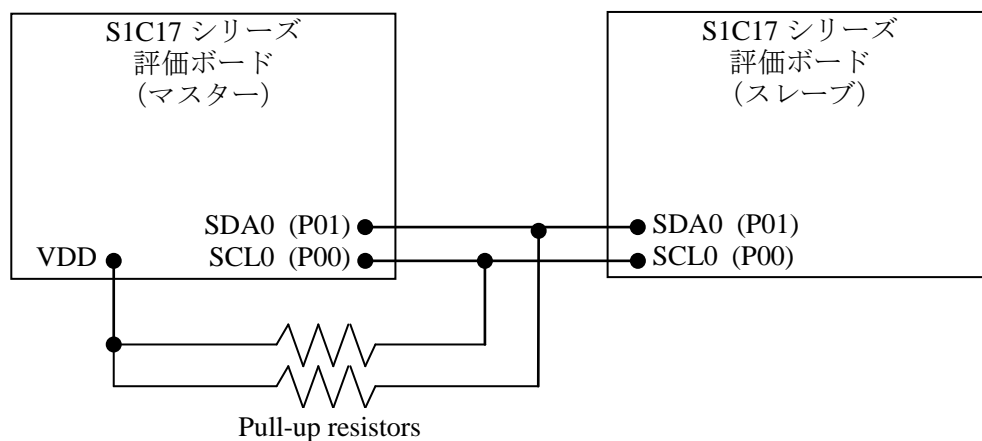


図 4.11.2.1 I2C マスター、スレーブサンプルプログラムハードウェア接続図(デフォルト割り付け)

表 4.11.2.1 PORT 例外割り付け対応表

機種名	SCL0	SDA0
S1C17M01	P03	P04
S1C17W22/W23	P03	P04

4. サンプルソフトウェア機能詳細

4.11.3 動作概要

4.11.3.1 マスターサンプル動作概要

1. I2C ポートを初期化します。
2. 「A(0x41)」から「Z(0x5A)」までのデータと「¥0」をスレーブに送信する。
3. スレーブから 27 バイトのデータを受信する。
4. 受信したデータが、送信したデータと同じか比較し、サンプルプログラムを終了します。

4.11.3.2 スレーブサンプル動作概要

1. I2C ポートを以下の値に初期化します。
 - アドレスモード : 7bit
 - スレーブアドレス : 0x2A
 - ジェネラルコール : 無効
2. マスターから 27 バイトのデータを受信する。
3. 「A(0x41)」から「Z(0x5A)」までのデータと「¥0」をマスターに送信する。
4. 受信したデータが、送信したデータと同じか比較し、サンプルプログラムを終了します。

4.12 LCD ドライバー (LCD4A)

4.12.1 サンプルプログラム仕様

LCD ドライバーサンプルプログラムは LCD4A が存在する機種を対象とするサンプルプログラムです。本サンプルプログラムは LCD4A 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、LCD4A(#define C17_PE_LCD4A)をアクティブにしてください。

4.12.2 ハードウェア条件

本サンプルプログラムは LCD4A が存在する機種を対象とするサンプルプログラムです。LCD パネルとの接続などについては、各機種のマニュアルを参照してください。本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.12.3 動作概要

1. 全点灯を 2 秒間行います。
2. 全消灯を 2 秒間行います。
3. 表示領域 0 に市松模様を設定します。
4. 表示領域 0 を選択し、通常表示を 2 秒間行います。
5. 反転表示を 2 秒間行います。
6. LCD の消灯を行い、サンプルプログラムを終了します。

4. サンプルソフトウェア機能詳細

4.13 LCD ドライバー (LCD8A)

4.13.1 サンプルプログラム仕様

LCD ドライバーサンプルプログラムは LCD8A が存在する機種を対象とするサンプルプログラムです。本サンプルプログラムは LCD8A 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、LCD8A(#define C17_PE_LCD8A)をアクティブにしてください。

4.13.2 ハードウェア条件

本サンプルプログラムは LCD8A が存在する機種を対象とするサンプルプログラムです。LCD パネルとの接続などについては、各機種のマニュアルを参照してください。本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.13.3 動作概要

3~18 の動作は S5U17M01 ボード専用 LCD を接続した場合に行われます。

1. 全点灯を 2 秒間行います。
2. 全消灯を 2 秒間行います。
3. セグメントに「00000000」の表示を 2 秒間行います。
4. セグメントに「12345678」の表示を 2 秒間行います。
5. 「battery warning」の点灯を 2 秒間行います。
6. 「overflow warning」の点灯を 2 秒間行います。
7. 「water-leak warning(up)」の点灯を 2 秒間行います。
8. 「water-leak warning(down)」の点灯を 2 秒間行います。
9. 「water-leak warning(up-down)」の点灯を 2 秒間行います。
10. 「reverse flow warning(inside)」の点灯を 2 秒間行います。
11. 「reverse flow warning(outside)」の点灯を 2 秒間行います。
12. 「reverse flow warning(inside-outside)」の点灯を 2 秒間行います。
13. 「unit("m3/h")」の点灯を 2 秒間行います。
14. 「communication」の点灯を 2 秒間行います。
15. 「pilot」の点灯を 2 秒間行います。
16. セグメントの消灯を 2 秒間行います。
17. セグメントに「Error」の表示を 2 秒間行います。
18. セグメントに「87654321」の表示を 2 秒間行います。
19. LCD の消灯を行い、サンプルプログラムを終了します。

4.14 LCD ドライバー (LCD8B)

4.14.1 サンプルプログラム仕様

LCD ドライバーサンプルプログラムは LCD8B が存在する機種を対象とするサンプルプログラムです。本サンプルプログラムは LCD8B 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、LCD8B(#define C17_PE_LCD8B)をアクティブにしてください。

4.14.2 ハードウェア条件

本サンプルプログラムは LCD8B が存在する機種を対象とするサンプルプログラムです。LCD パネルとの接続などについては、各機種のマニュアルを参照してください。本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.14.3 動作概要

1. 全点灯を 2 秒間行います。
2. 全消灯を 2 秒間行います。
3. 表示領域 0 に市松模様を設定します。
4. 表示領域 0 を選択し、通常表示を 2 秒間行います。
5. 反転表示を 2 秒間行います。
6. LCD の消灯を行い、サンプルプログラムを終了します。

4. サンプルソフトウェア機能詳細

4.15 LCD ドライバー (LCD16A)

4.15.1 サンプルプログラム仕様

LCD ドライバーサンプルプログラムは LCD16A が存在する機種を対象とするサンプルプログラムです。本サンプルプログラムは LCD16A 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、LCD16A(#define C17_PE_LCD16A)をアクティブにしてください。

4.15.2 ハードウェア条件

本サンプルプログラムは LCD16A が存在する機種を対象とするサンプルプログラムです。LCD パネルとの接続などについては、各機種のマニュアルを参照して下さい。本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.15.3 動作概要

1. 全点灯を 2 秒間行います。
2. 全消灯を 2 秒間行います。
3. 表示領域 0 に市松模様を設定します。
4. 表示領域 0 を選択し、通常表示を 2 秒間行います。
5. 反転表示を 2 秒間行います。
6. LCD の消灯を行い、サンプルプログラムを終了します。

4.16 LCD ドライバー (LCD24A)

4.16.1 サンプルプログラム仕様

LCD ドライバーサンプルプログラムは LCD24A が存在する機種を対象とするサンプルプログラムです。本サンプルプログラムは LCD24A 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、LCD24A(#define C17_PE_LCD24A)をアクティブにしてください。

4.16.2 ハードウェア条件

本サンプルプログラムは LCD24A が存在する機種を対象とするサンプルプログラムです。LCD パネルとの接続などについては、各機種のマニュアルを参照してください。本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.16.3 動作概要

1. 全点灯を 2 秒間行います。
2. 全消灯を 2 秒間行います。
3. 表示領域 0 に市松模様 (小) を設定します。
4. 表示領域 0 を選択し、通常表示を 2 秒間行います。
5. 反転表示を 2 秒間行います。
6. 表示領域 1 に市松模様 (中) を設定します。
7. 表示領域 1 を選択し、通常表示を 2 秒間行います。
8. 反転表示を 2 秒間行います。
9. 表示領域 0 に市松模様 (大) を設定します。
10. 表示領域 0 を選択し、通常表示を 2 秒間行います。
11. 反転表示を 2 秒間行います。
12. LCD の消灯を行い、サンプルプログラムを終了します。

4. サンプルソフトウェア機能詳細

4.17 LCD ドライバー (LCD32B)

4.17.1 サンプルプログラム仕様

LCD ドライバーサンプルプログラムはLCD32Bが存在する機種を対象とするサンプルプログラムです。本サンプルプログラムはLCD32B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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、LCD32B(#define C17_PE_LCD32B)をアクティブにしてください。

4.17.2 ハードウェア条件

本サンプルプログラムは LCD32B が存在する機種を対象とするサンプルプログラムです。LCD パネルとの接続などについては、各機種のマニュアルを参照して下さい。本サンプルプログラムはマイコンの OSC1 (32.768kHz) 内蔵発振回路で動作します。

4.17.3 動作概要

1. 全点灯を 2 秒間行います。
2. 全消灯を 2 秒間行います。
3. 表示領域 0 に市松模様 (小) を設定します。
4. 表示領域 0 を選択し、通常表示を 2 秒間行います。
5. 反転表示を 2 秒間行います。
6. 表示領域 1 に市松模様 (中) を設定します。
7. 表示領域 1 を選択し、通常表示を 2 秒間行います。
8. 反転表示を 2 秒間行います。
9. 表示領域 0 に市松模様 (大) を設定します。
10. 表示領域 0 を選択し、通常表示を 2 秒間行います。
11. 反転表示を 2 秒間行います。
12. LCD の消灯を行い、サンプルプログラムを終了します。

4.18 7セグメント LED コントロール回路 (LEDC)

4.18.1 サンプルプログラム仕様

7セグメント LED コントローラ回路サンプルプログラムは7セグメント LED コントローラ回路を使用して以下の動作を行います。

- 接続された7セグ LED の表示を行う。

本サンプルプログラムを使用する場合は T16(`#define C17_PE_T16`)、RTCA(`#define C17_PE_RTCA`)、LEDC(`#define C17_PE_LEDC`)をアクティブにしてください。

4.18.2 ハードウェア条件

本サンプルプログラムは LEDC が存在する機種を対象とするサンプルプログラムです。7セグ LED との接続などについては、各機種のマニュアルを参照してください。本サンプルプログラムはマイコンの OSC3 内蔵発振回路で動作します。

4.18.3 動作概要

1. COM0 に接続された7セグ LED に「0」表示を行います。
2. COM1 に接続された7セグ LED に「1」表示を行います。
3. COM2 に接続された7セグ LED に「2」表示を行います。
4. COM3 に接続された7セグ LED に「3」表示を行います。
5. COM4 に接続された7セグ LED に「4」表示を行います。
6. LEDC を停止させ、サンプルプログラムを終了します。

4. サンプルソフトウェア機能詳細

4.19 R/F 変換機（RFC）

4.19.1 サンプルプログラム仕様

R/F 変換機サンプルプログラムは R/F 変換機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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RFC(#define C17_PE_RFC)をアクティブにしてください。

4.19.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1（32.768kHz）内蔵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。

SENBO 端子は OPEN にしてください。

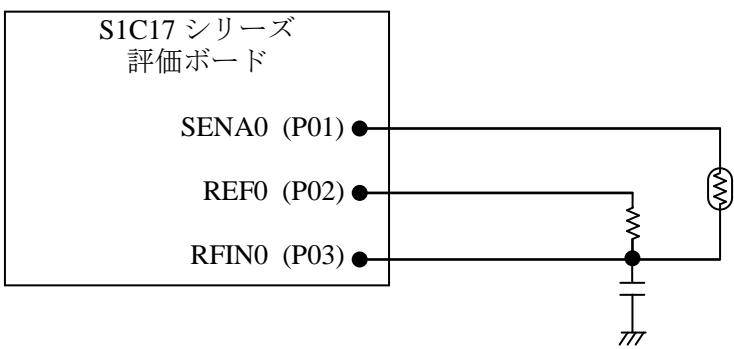


図 4.19.2.1 R/F 変換機サンプルプログラム ハードウェア接続図(デフォルト割り付け)

表 4.19.2.1 PORT 例外割り付け対応表

機種名	SENA0	REF0	REFIN
S1C17M01	P04	P05	P06
S1C17M22/M25	P34	P35	P36
S1C17W03/W04	P16	P15	P14
S1C17W12/W13	P05	P06	P07

4.19.3 動作概要

- RFC の外部クロック入力モード、発振モードを設定します。
- R/F 変換機の計測カウンターの初期値を 0x00fc0000 とし、計測を開始します(計測時間 約 8 秒)。
- 基準発振、センサ発振により計測値を取得し、サンプルプログラムを終了します。

4.20 サウンドジェネレータ (SNDA)

4.20.1 サンプルプログラム仕様

サウンドジェネレータサンプルプログラム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。

- ノーマルブザーモード機能によるブザーの鳴動。
- ワンショットブザーモード機能によるブザーの鳴動。
- メロディモード機能によるブザーの鳴動。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、RTCA(#define C17_PE_RTCA)、SNDA(#define C17_PE_SNDA)をアクティブにして下さい。

4.20.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1 (32.768kHz) または、内蔵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。

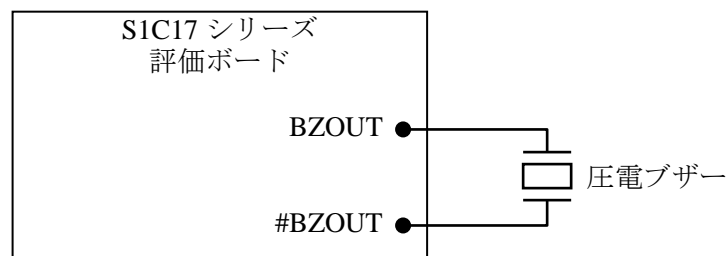


図 4.20.2.1 サウンドジェネレータサンプルプログラム ハードウェア接続図

表 4.20.2.1 PORT 割り付け対応表

機種名	BZOUT	#BZOUT
S1C17M20/M21/M22/M23/M24/M25	P02	P03
S1C17M30/M31/M32/M33/M34	P33	P34
S1C17W03/W04	P04	P05
S1C17W12/W13	P02	P03
S1C17W14/W16	P06	P07
S1C17W15	P15	P14
S1C17W18	P15	P14
S1C17W22/W23	P14	P13
S1C17W34/W35/W36	P23	P22

4.20.3 動作概要

1. ノーマルブザーモード機能によるブザーの鳴動をします。
2. ワンショットブザーモード機能によるブザーの鳴動をします。
3. メロディモード機能によるブザーの鳴動をします。

4. サンプルソフトウェア機能詳細

4.21 電源電圧検出回路 (SVD)、(SVD3)

4.21.1 サンプルプログラム仕様

電源電圧検出回路サンプルプログラムは電源電圧検出回路を使用して以下の動作を行います。

- 電源電圧検出回路にて、電源電圧と比較電圧との比較を行う。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、SVD(#define C17_PE_SVD)をアクティブにしてください。

4.21.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵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。

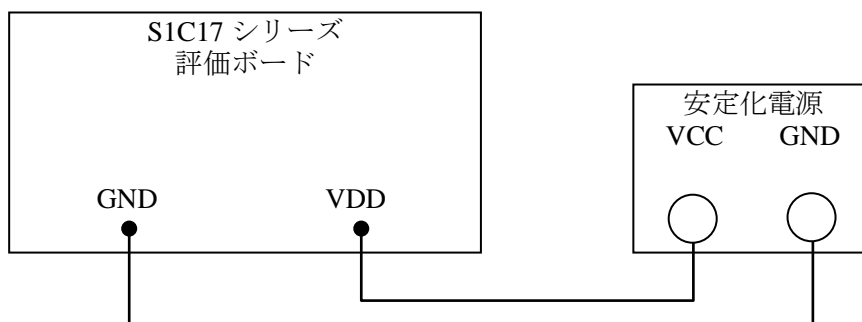


図 4.21.2.1 電源電圧検出回路サンプルプログラム ハードウェア接続図

4.21.3 動作概要

- 16 ビットタイマを 1 秒毎に割り込みが発生するように設定し、開始します。
- 電源電圧検出回路を 2.7v 以下の電圧で、電源電圧低下を検出するように設定し、開始します。
- 電圧電圧の低下が発生したかどうかを確認します。
- 10 秒間、電源電圧の低下が発生したか計測し、サンプルプログラムを終了します。

4.22 パワージェネレータ (PWG)、(PWG2)

4.22.1 サンプルプログラム仕様

パワージェネレータサンプルプログラムはパワージェネレータを使用して以下の動作を行います。

- 省電力モードを切り替える。

本サンプルプログラムを使用する場合は T16(`#define C17_PE_T16`)、PWG(`#define C17_PE_PWG`)をアクティブにしてください。

4.22.2 ハードウェア条件

本サンプルプログラムはマイコンの OSC1 (32.768kHz) または内蔵発振回路で動作します。

4.22.3 動作概要

1. OSC1 を起動します。
2. PWG の動作モードを設定します。

4. サンプルソフトウェア機能詳細

4.23 IR リモートコントローラ (REMC)、(REMC2)、(REMC3)

4.23.1 サンプルプログラム仕様

IR リモートコントローラサンプルプログラムは IR リモートコントローラを使用して以下の動作を行います。

- NEC format でデータ信号を発生させます。
- AEHA format でデータ信号を発生させます。
- SONY format でデータ信号を発生させます。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、REMC(#define C17_PE_REMC)をアクティブにしてください。

4.23.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

図 4.23.2.1 に示した回路例との接続などについては、SVT17 シリーズのマニュアルを参照してください。

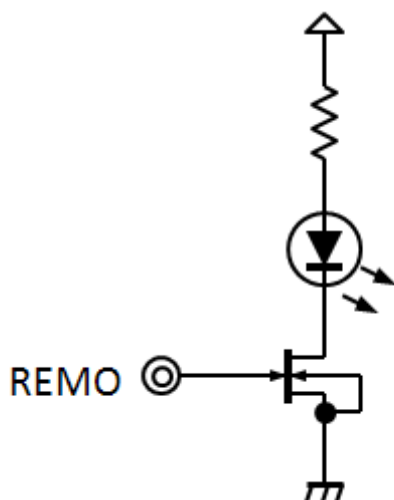


図 4.23.2.1 REMC サンプルプログラム 赤外発光ダイオード接続回路例

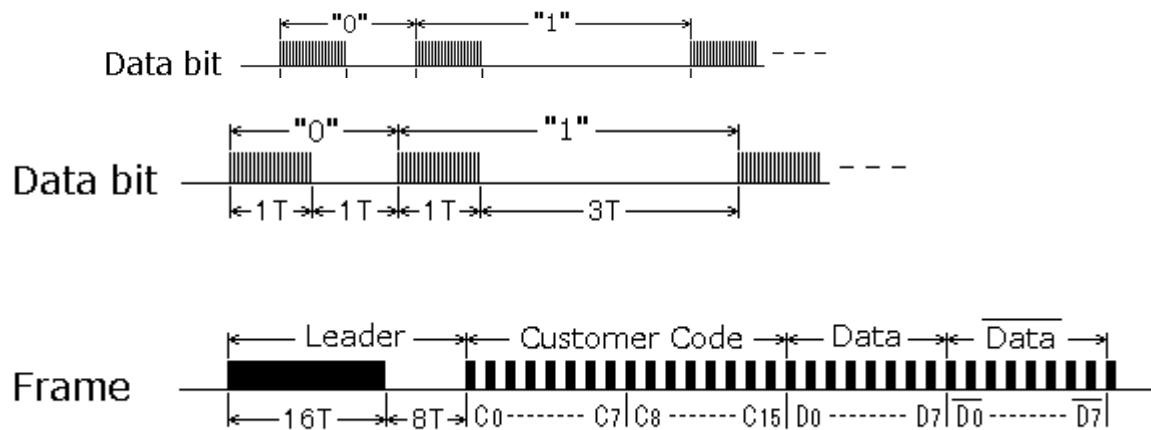
表 4.23.2.1 PORT 割り付け対応表

機種名	REMO
S1C17M12/M13	P50
S1C17M20/M21/M22/M23/M24/M25	P12
S1C17M30/M31/M32/M33/M34	P07
S1C17W03/W04	P00
S1C17W12/W13	P00
S1C17W14/W16	P16
S1C17W18	P62
S1C17W22/W23	P16
S1C17W34/W35/W36	P20

4.23.3 動作概要

- 1 例として NEC フォーマットの仕様の概略を以下と図 4.23.3.1 に示します。

キャリア: 赤外線 ($\lambda_p = 940\text{nm}$)
 サブキャリア: $f_{sc} = 38\text{kHz}$, $1/3\text{duty}$
 $T = 562\mu\text{s}$
 固定長フレーム (32bit)
 16bit のカスタマーコード
 8bit のデータ + 8bit の反転データ



4. サンプルソフトウェア機能詳細

4.24 ソフトウェア IR リモートコントローラ (SOFT REMC)

4.24.1 サンプルプログラム仕様

ソフトウェア IR リモートコントローラサンプルプログラムは T16、16B を使用し IR リモートコントローラの動作をソフトウェアで疑似的に実現し、以下の動作を行います。

- NEC format でデータ信号を発生させます。
- AEHA format でデータ信号を発生させます。
- SONY format でデータ信号を発生させます。

T16B が実装されている MCU の場合は T16B×2 チャンネルか T16B×1 チャンネル+T16×2 チャンネルを、T16B が実装されていない MCU の場合は T16×4 チャンネルを使用します。使用するタイマは c17_init_config.h で指定しますが、本プログラムを実行中は該当タイマを他の用途で使わないで下さい。また、本プログラムを実行すると該当タイマの設定レジスタが変更されます。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、SOFTREMC(#define C17_PE_SOFTREMC)をアクティブにしてください。

4.24.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

表 4.24.2.1 PORT 割り付け対応表

機種名	REMO
S1C17 シリーズ	P00

4.24.3 動作概要

- 1 例として NEC フォーマットの仕様の概略を以下と図 4.24.3.1 に示します。

キャリア: 赤外線 ($\lambda p = 940\text{nm}$)
サブキャリア: fsc = 38kHz, 1/3duty
T = 562 μs
固定長フレーム (32bit)
16bit のカスタマーコード
8bit のデータ + 8bit の反転データ

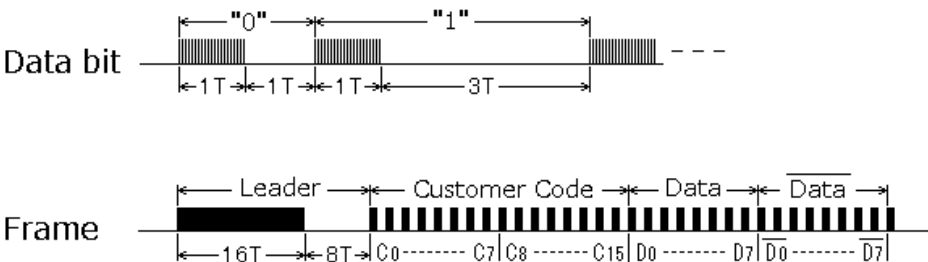


図 4.24.3.1 NEC フォーマットの信号波形

- 2 NEC format でデータ信号を発生させます。
3 AEHA format でデータ信号を発生させます。
4 SONY format でデータ信号を発生させます。

4.25 12 ビット A/D 変換器 (ADC12)

4.25.1 サンプルプログラム仕様

12 ビット A/D 変換器サンプルプログラムは 12 ビット A/D 変換器を使用して以下の動作を行います。

- 1 秒毎に A/D 変換を行い、A/D 変換の結果値を取得する。
- 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、ADC(#define C17_PE_ADC)をアクティブにしてください。

4.25.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

4.25.3 動作概要

1. 12 ビット A/D 変換器の設定をします。
2. 16 ビットタイマのカウンターを設定します。
3. 16 ビットタイマの開始します。
4. 12 ビット A/D 変換器のトリガ待ちを開始し、CPU を Halt 状態にします。
5. 1 秒毎に 16 ビットタイマの割り込みが発生し、A/D 変換トリガを行います。
6. 16 ビットタイマの割り込みが発生すると CPU は Halt 状態を解除します。
7. 12 ビット A/D 変換器の変換結果、割り込み回数を取得します。
8. 12 ビット A/D 変換割り込み回数が 10 回未満のときは、再び CPU を Halt 状態にします。
9. 12 ビット A/D 変換割り込み回数が 10 回以上のときは、12 ビット A/D 変換器、16 ビットタイマを停止し、サンプルプログラムを終了します。

4. サンプルソフトウェア機能詳細

4.26 オペアンプ/コンパレータ（OPCMP）

4.26.1 サンプルプログラム仕様

オペアンプ/コンパレータサンプルプログラムはオペアンプ/コンパレータ回路を使用して以下の動作を行います。

- OPCMP のコンパレータによる比較結果を取得する。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、OPCMP(#define C17_PE_OPCMP) をアクティブにしてください。

4.26.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

表 4.26.2.1 PORT 割り付け対応表

機種名	OPIN0P	OPIN0N	OPOUT0
S1C17W23	P06	P07	P10

4.26.3 動作概要

1. オペコンプ/コンパレータの設定をします。
2. オペコンプ/コンパレータの比較結果を取得します。
3. オペコンプ/コンパレータを停止し、サンプルプログラムを終了します。

4.27 温度センサ/基準電圧生成回路 (TSRVR)

4.27.1 サンプルプログラム仕様

温度センサ/基準電圧生成回路サンプルプログラムは温度センサ/基準電圧生成回路と内蔵 A/D 変換器を使用して以下の動作を行います。

- 1 秒毎に内蔵温度センサの出力電圧を A/D 変換し、計測された温度を算出する。
- 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。

本サンプルプログラムを使用する場合は T16(#define C17_PE_T16)、ADC(#define C17_PE_ADC) 、TSRVR(#define C17_PE_TSRVR)をアクティブにしてください。

4.27.2 ハードウェア条件

本サンプルプログラムはマイコンの内蔵発振回路で動作します。

4.27.3 動作概要

1. 温度センサ/基準電圧生成回路の設定をします。
2. 12 ビット A/D 変換器の設定をします。
3. 16 ビットタイマのカウンターを設定します。
4. 16 ビットタイマの開始します。
5. 12 ビット A/D 変換器のトリガ待ちを開始し、CPU を Halt 状態にします。
6. 1 秒毎に 16 ビットタイマの割り込みが発生し、A/D 変換トリガを行います。
7. 16 ビットタイマの割り込みが発生すると CPU は Halt 状態を解除します。
8. 12 ビット A/D 変換器の変換結果、割り込み回数を取得します。
9. 12 ビット A/D 変換割り込み回数が 10 回未満のときは、再び CPU を Halt 状態にします。
10. 12 ビット A/D 変換割り込み回数が 10 回以上のときは、各 12 ビット A/D 変換器の変換結果から、温度を算出します。
11. 温度センサ/基準電圧生成回路、12 ビット A/D 変換器、16 ビットタイマを停止し、サンプルプログラムを終了します。

改訂履歴表

付ー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	2017/02/28	全ページ	新規	新規制定
Rev 1.1	2018/01/10	全ページ	変更	以下のように語句を統一 サンプルソフトウェア：プログラム全体 サンプルプログラム：個々のモジュールのプログラム
		2	変更	図 2.1.1 reg ディレクトリ位置変更
		3	変更	項目順番変更
		4	変更	Web リンクアドレス変更
		19-20	追加	図 4.1.1、図 4.1.2 機種名追加：W12, M10, M20-M25 周辺回路追加：SOFT REMC
		22	削除	4.2.2 (3)項削除
		23-51	追加	4.x.1 項にサンプルプログラム使用時にアクティブにするモジュールの記述を追加
		38	追加	4.15 項に LCD16A モジュールの記述を追加
		47	変更	図 4.23.3.1 Repeat 以降を削除
		48	追加	4.24 項に SOFT REMC モジュールの記述を追加
		50	変更	内容が ADC サンプルプログラムの記述だったので、OPCMP サンプルプログラムの記述に変更
		51	変更	内容が ADC サンプルプログラムの記述だったので、TSRVR サンプルプログラムの記述に変更

セイコーエプソン株式会社

営業本部 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 御堂筋グランタワー15F
TEL (06) 6120-6000 (代表) FAX (06) 6120-6100

ドキュメントコード : 413445701
2017 年 3 月 作成
2018 年 1 月 改訂