

S1C17 Family Application Note

S1C17 シリーズ
PORT 多重割り込み
アプリケーションノート

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.....	1
1.1 機能.....	1
1.2 フォルダ構成.....	2
1.3 ファイル構成.....	2
2. サンプルソフトウェア機能詳細.....	3
2.1 サンプルソフトウェア仕様.....	3
2.2 動作概要.....	4
2.3 アプリケーションプログラムへの適用方法.....	5
3. 関数仕様.....	7
3. 動作時間.....	9
改訂履歴表.....	10

1. 概要

S1C17 シリーズの割り込みベクタテーブルは周辺モジュール毎に複数の割り込み要因をまとめる形で実装されていて、ポート割り込みでは全ポートの割り込み要因が一つの割り込みベクタに割り振られています。

そのため、複数のポートの割り込み要因に優先順位をつけて多段割り込みを行う処理は割り込みコントローラの制御だけでは不可能です。

そこで割り込みハンドラで割り込みベクタの起動状況と割り込み優先度を管理して、ポートの多段割り込みを実行する関数を提供します。

1.1 機能

ポートグループ毎 (P0X,P1X,P2X,...) に割り込みレベルを設定します。

現在実行中の割り込み関数より、優先度の高い割り込みが発生した場合は、そちらの処理を実行します。

逆に、優先度の低い割り込みが発生した場合は、優先度の高い割り込み処理が終了後、処理を実施します。

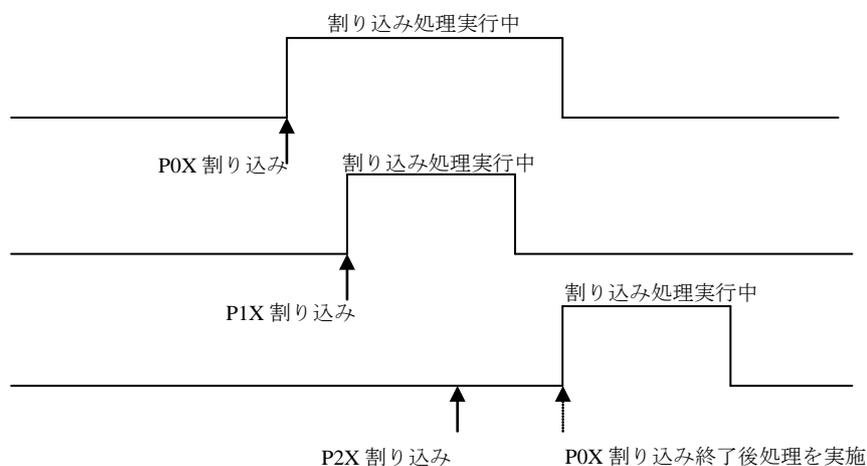
動作イメージは以下のようになります。

割り込みレベル

P0X=2

P1X=3

P2X=1



P0X 割り込み発生

→P0X 割り込み処理実施

P0X 割り込み処理中に P1X 割り込み発生

→割り込みレベルが $P0X < P1X$ なので P1X 割り込み処理実施

P1X 割り込み処理中に P2X 割り込み発生

→割り込みレベルが $P2X < P1X$ なので P2X 割り込み処理は P0X 割り込み処理終了後に実施

1. 概要

1.2 フォルダ構成

本パッケージのフォルダ構成は以下の通りです。

+c17_Port_Nested_Interrupt	
+c17_Port_Nested_Interrupt_gnu17v2	: GNU17Ver.2.x 用サンプルプログラム
+c17_Port_Nested_Interrupt_gnu17v3	: GNU17Ver.3.x 用サンプルプログラム
-s1c17_Port_Nested_Interrupt_j.pdf	: 説明書(日本語)
-s1c17_Port_Nested_Interrupt_e.pdf	: 説明書(英語)
-License_e.txt	: ソフトウェアライセンス契約書(英語)
-README.txt	: README ファイル

1.3 ファイル構成

本パッケージのファイル構成は以下の通りです。

ファイル名	機能
src/c17_Port_Nested_Interrupt.c	ポート多重割り込み処理関数
src/c17_Port_Nested_Interrupt.h	ポート多重割り込み処理関数ヘッダ定義ファイル
src/main.c	メイン関数
src/boot.c	スタートアップモジュール
src/c17_port_m30_m31_m32_m33_m34.c	ポートサンプルドライバ
src/c17_port_m30_m31_m32_m33_m34.h	ポートサンプルドライバヘッダ定義ファイル
src/c17_misc.c	MISC サンプルドライバ
src/c17_misc.h	MISC サンプルドライバヘッダ定義ファイル
src/c17_clg.c	CLG サンプルドライバ
src/c17_clg.h	CLG サンプルドライバヘッダ定義ファイル
src/c17_init_config.h	周辺回路への設定を行うためのヘッダ定義ファイル
src/crt0.h	割り込みベクタ&ブート処理関数ヘッダ定義ファイル
inc	CPU レジスタ定義ファイルディレクトリ
inc/reg	CPU 周辺回路制御レジスタ定義ファイルディレクトリ

2. サンプルソフトウェア機能詳細

サンプルソフトウェアの機能詳細について記載します。

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 S1C17M33 用として実装されています。

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。

- ポートを入力に設定し、入力信号が High レベルから Low レベルになったとき、割り込みが発生します。
- 割り込み発生時にポートを出力に設定し、High レベルを出力します。
- 入力ポートが High レベルになるのを待ちます。
- 出力ポートに Low レベルを出力します。

使用するポート設定とポート名は以下の通りです。

表 2.1.1 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
出力ポート	P07
割り込みレベル	2

入力割り込みポート	P10
出力ポート	P17
割り込みレベル	3

入力割り込みポート	P20
出力ポート	P27
割り込みレベル	2

入力割り込みポート	P30
出力ポート	P37
割り込みレベル	1

2. サンプルソフトウェア機能詳細

2.2 動作概要

メイン処理

ポートを初期化します。

P0X ポートグループの割り込みレベルを設定します。

P00 ポートを立下りエッジで割り込みを許可します。

P07 から Low を出力します。

P1X ポートグループの割り込みレベルを設定します。

P10 ポートを立下りエッジで割り込みを許可します。

P17 から Low を出力します。

P2X ポートグループの割り込みレベルを設定します。

P20 ポートを立下りエッジで割り込みを許可します。

P27 から Low を出力します。

P3X ポートグループの割り込みレベルを設定します。

P30 ポートを立下りエッジで割り込みを許可します。

P37 から Low を出力します。

割り込み処理 (P0X)

割り込みハンドラよりポートの割り込み関数 (intInputP0x) が呼び出されます。

P07 を High 出力します。

P00 が High レベルになるまで待ちます。

P07 を Low 出力します。

他の割り込み関数(intInputP1x, intInputP2x,...)も制御するポートが異なりますが、処理は同じです。

2.3 アプリケーションプログラムへの適用方法

本プログラムをアプリケーションプログラムのプロジェクトに組み込む方法について説明します。

1. ファイルのプロジェクトへの追加

c17_Port_Nested_Interrupt.c/c17_Port_Nested_Interrupt.h をプロジェクトに追加します。

2. ヘッダファイル宣言

使用するソースファイル内に、以下をインクルード宣言します。

```
#include "c17_port_nested_interrupt.h"
```

3. ベクタテーブルへの登録

GNU17Ver.2.x の場合

割り込み関数 “c17portNestedInterruptPortInput” を PORT 割り込みとして登録します。

boot.c

```
#include "c17_port_nested_interrupt.h"
/** vector table */
func *const vector[] = {
    /* ----- Non-Maskable Interrupts ----- */
    VECTOR(boot),                // 0      00      Reset
    VECTOR(intAddrErr),          // 1      04      Address misaligned
    VECTOR(intReserved),        // 2      08      NMI
    VECTOR(emu_copro_process),   // 3      0C      Co-processor emulation

    /* ----- Maskable Interrupts ----- */
    VECTOR(c17intSvd),           // 4      10      SVD3 interrupt
    VECTOR(c17portNestedInterruptPortInput), // 5      14      PORT interrupt
};
```

GNU17Ver.3.x の場合

サンプルのポートドライバを使用している場合は、サンプルのポートドライバのベクタハンドラの登録をコメントアウトします。

c17_port_m30_m31_m32_m33_m34.c

```
/* --- interrupt handler (GNU17v3 Only) --- */
#ifdef CRT0_LINKED
//C17_INTERRUPT_HANDLER(_vector05_handler, c17intPortInput); // comment out
#endif /* CRT0_LINKED */
```

2. サンプルソフトウェア機能詳細

4. ユーザー割り込み処理の追加

c17_port_nested_interrupt.c の intInputP0x, intInputP1x,...に割り込み関数内で処理を追加します。

```
void intInputP0x(void)
{
    unsigned short interruptFlag;

    interruptFlag = pxintf_get[C17_PORT_GROUP_P0X]; //割り込み要因保存変数を取得
    pxintf_get[C17_PORT_GROUP_P0X] = 0;           //割り込み要因保存変数をクリア
    asm("ei");                                     //多重割り込み許可
    if(interruptFlag & 0x01){
        /**
        * @todo add interrupt function of P00          //ユーザー処理の追加
        */
    }
    if(interruptFlag & 0x02){
        /**
        * @todo add interrupt function of P01          //ユーザー処理の追加
        */
    }
};
```

5. ポートグループの割り込みレベルの登録

割り込みを使用するポートグループ (P0x,P1X,P2X,...) の割り込みレベルを関数 c17portNestedInterruptSetInterruptLevel で登録します。

```
c17portNestedInterruptSetInterruptLevel(C17_PORT_GROUP_P0X, 2);
c17enableIntP0x(0, C17_PORT_INT_EDGE_FALLING);

c17portNestedInterruptSetInterruptLevel(C17_PORT_GROUP_P1X, 3);
c17enableIntP1x(0, C17_PORT_INT_EDGE_FALLING);

c17portNestedInterruptSetInterruptLevel(C17_PORT_GROUP_P2X, 2);
c17enableIntP2x(0, C17_PORT_INT_EDGE_FALLING);

c17portNestedInterruptSetInterruptLevel(C17_PORT_GROUP_P3X, 1);
c17enableIntP3x(0, C17_PORT_INT_EDGE_FALLING);
```

3. 関数仕様

関数について記します。

ポートグループの割り込みレベル設定

関数名		
c17portNestedInterruptSetInterruptLevel		
引数		
portGr	unsigned short	ポートグループ
portIntLevel	unsigned short	割り込みレベル 1 ... 4
戻り値		
unsigned short	EXIT_SUCCESS : 成功 EXIT_FAILURE : 失敗	
機能		
引数で指定されたポートグループに割り込みレベルを指定します。		
備考		
割り込みレベルの有効範囲は 1~4 です。		

3. 関数仕様

ポートの割り込みハンドラ

関数名
c17portNestedInterruptPortInput
引数 ：なし
戻り値 ：なし
機能
ポートの割り込みハンドラです。 本関数より、ポートの割り込み関数（intInputP0x, intInputP1x, intInputP2x,...）を呼び出します。
備考
GNU17v2 の時本関数を PORT の割り込み関数としてベクタに登録します。

ポートの割り込み関数

関数名
intInputP0x, intInputP1x, intInputP2x,...
引数 ：なし
戻り値 ：なし
機能
ポート割り込みハンドラより呼び出されます。 ① 割り込み要因保存変数より、割り込み要因を取得します。 ② 割り込み要因保存変数をクリアします。 ③ ポートの多重割り込みを有効にします。 ④ 各ポートの処理を実行します。（ユーザー記述）
備考
各ポートの割り込み処理を記述してください。

3. 動作時間

ポート入力によるハードウェア割り込みが発生してから、ポートの割り込み関数（intInputPxx）が呼ばれるまでの時間は以下になります。

時間：228usec

参考

通常の M33 のサンプルプログラム

時間：208usec

条件

動作クロック：IOSC=700KHZ

アクセスウェイト：FLASHCWAIT=0

割り込みポート：P10

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 : 413541700
2017年 8月 作成