
SH7753 グループ

TMU ドライバソフトウェア

R01AN1612JJ0100
Rev.1.00
2013.7.1

要旨

本仕様書は、SH7753 グループの TMU (Timer Unit) ドライバについて説明します。

動作確認デバイス

SH7753

動作環境

本仕様書に示す TMU ドライバの動作環境を以下に示します。

- 評価ボード SH7753 グループ EVB ボード
- ソフトウェア : High-Performance Embedded Workshop- Ver 4.09.01.007
 : Toolchain - Ver 9.4.1.0
 : OptLinker - Ver 10.01.00
 : SH アセンブラ - Ver 7.01.02
 : SH C/C++コンパイラ- Ver 9.04.01
 : SH C/C++ライブラリジェネレータ - Ver 3.00.03
- エミュレータ : E10A エミュレータ Ver 3.03.00

目次

1.	機能概要	3
1.1	主な機能	3
1.2	関連ドキュメント	3
2.	ドライバ仕様	4
2.1	関数一覧	4
2.2	コールバック関数仕様一覧	4
3.	関数	5
	R_TMU_SetOpr	6
	R_TMU_SetCounter	7
	R_TMU_TimerStart	8
	R_TMU_MultiTimerStart	9
	R_TMU_TimerStop	10
	R_TMU_MultiTimerStop	11
	R_TMU_GetCounter	12
	R_TMU2_GetInputCapture	13
	R_TMU_GetFlag	14
	R_TMU_ClearFlag	15
	R_TMUn_Interrupt(n=0 to 5)	16
	R_TMU2_InCapInterrupt	17
4.	動作条件	18
4.1	割り込みについて	18
4.2	使用メモリ領域について	19
5.	付録	20
5.1	ドライバ関数の使用例	20
5.1.1	TMU周期割り込み使用例(単一チャネル)	20
5.1.2	TMU周期割り込み使用例(マルチスタート)	21
5.1.3	TMU経過時間測定の使用例	22

1. 機能概要

本デバイスドライバは、SH7753 に搭載されている Timer Unit (以下、TMU)モジュールを使用し、時間測定処理を実現する関数群で構成されています。

1.1 主な機能

TMU ドライバの主な機能を以下に示します。

- ・ TMU チャンネルの初期化処理
- ・ TMU タイマのカウント動作開始処理
- ・ TMU タイマのカウント動作停止処理

1.2 関連ドキュメント

SH7753 グループ ユーザーズマニュアル：ハードウェア編

SH7753 グループ アプリケーションノート DMAC ドライバソフトウェア

2. ドライバ仕様

2.1 関数一覧

表 1に、TMU ドライバが提供する関数一覧を示します。

表 1. TMU ドライバ関数一覧

分類	関数名	機能概要	スタック サイズ
TMU 設定関数	R_TMU_SetOpr	チャンネルの初期化処理	24 バイト
	R_TMU_SetCounter	カウンタの初期化処理	24 バイト
タイマカウント関数	R_TMU_TimerStart	カウント動作の開始処理	32 バイト
	R_TMU_MultiTimerStart	カウント動作の開始処理 (マルチスタート)	12 バイト
	R_TMU_TimerStop	カウント動作の停止処理	20 バイト
	R_TMU_MultiTimerStop	カウント動作の停止処理 (マルチストップ)	12 バイト
情報取得関数	R_TMU_GetCounter	カウンタ値の取得処理	16 バイト
	R_TMU2_GetInputCapture	インプットキャプチャレジスタ値の取得処理	4 バイト
	R_TMU_GetFlag	割り込みフラグの取得処理	20 バイト
	R_TMU_ClearFlag	割り込みフラグのクリア処理	24 バイト
割り込みハンドラ関数	R_TMUn_Interrupt(n=0 to 5)	アンダフロー割り込みハンドラ	12 バイト
	R_TMU2_InCapInterrupt	インプットキャプチャ割り込みハンドラ	8 バイト

2.2 コールバック関数仕様一覧

表 2に TMU ドライバのコールバック関数仕様一覧を示します。

表 2. TMU ドライバコールバック関数仕様一覧

関数名	コールバック名	仕様内容
R_TMU_TimerStart	void (*pv_unf_callback) (void)	アンダフロー発生時のコールバック関数。 (NULL の場合、コールバックなし)
	void (*pv_icpf_callback) (void)	インプットキャプチャ発生時のコールバック関数。 (NULL の場合、コールバックなし)

3. 関数

本章では、TMU ドライバの各関数仕様詳細を示します。各関数詳細の読み方は以下のとおりです。

関数名		分類
機能概要		
書式	関数の呼び出し形式を示します。#include “ヘッダファイル”で示すヘッダファイルは、この関数の実行に必要な標準ヘッダファイルであるため、必ずインクルードしてください。	
引数	I,O は、引数がそれぞれ入力データ、出力データであることを意味します。	
戻り値	関数の戻り値を示します。	
解説	関数の仕様について説明します。	
注意事項	注意事項があればここに示します。	

R_TMU_SetOpr

TMU 設定関数

チャンネルの初期化処理

- 書式**
- ```
#include "r_common.h"
#include "iodefine.h"
#include "r_tmu_if.h"
char_t R_TMU_SetOpr (uchar_t uc_ch_num, tmu_opr_t *pst_tmu_opr);
```
- 引数**
- |                        |  |                          |
|------------------------|--|--------------------------|
| uchar_t uc_ch_num      |  | TMU のチャンネル番号 (0~5)       |
| tmu_opr_t *pst_tmu_opr |  | TMU オペレーション処理管理構造体へのポインタ |
- 戻り値**
- |                |          |
|----------------|----------|
| RET_NORMAL     | 正常終了     |
| RET_ERR_PARAM1 | 第 1 引数不正 |
| RET_ERR_PARAM2 | 第 2 引数不正 |
- 解説**
- 本関数は、TMU のチャンネル番号で選択したチャンネルのカウント機能を選択します。

## 【TMU オペレーション処理管理構造体】 pst\_tmu\_opr

| メンバ                            | I/O | 機能                           | 説明                                                                                                                                                                       |
|--------------------------------|-----|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| uchar_t<br>uc_icpf_enable;     |     | インプットキャプチャ制御<br>(チャンネル 2 のみ) | 0 : インプットキャプチャ機能を使用しない<br>1 : インプットキャプチャ機能を使用するが、インプットキャプチャによる割り込み(TICPI2)を許可しない<br>2 : インプットキャプチャ機能を使用し、またインプットキャプチャによる割り込み(TICPI2)を許可する<br><br>※チャンネル 2 以外では 0 にしてください |
| uchar_t<br>uc_clock_edge;      |     | クロックエッジ                      | 0 : 立ち上がりエッジでカウント<br>1 : 立ち下りエッジでカウント<br>2 : 立ち上がり/立ち下りの両エッジでカウント                                                                                                        |
| uchar_t<br>uc_timer_prescaler; |     | タイマプリスケアラ                    | 0 : Pck/4でカウント<br>1 : Pck/16でカウント<br>2 : Pck/64でカウント<br>3 : Pck/256でカウント<br>4 : Pck/1024でカウント<br>5 : 外部クロックでカウント(※)<br>※チャンネル 3、4、5 では設定禁止                               |

- 注意**
- TMU の機能にはチャンネル依存があります。SH7753 グループユーザーズマニュアル ハードウェア編を参照の上、使用する機能が割り当てられたチャンネルを選択してください。
- 本関数を実行する前に、選択したチャンネルのカウント動作を停止させてください。カウント動作の停止には R\_TMU\_TimerStop()関数を実行してください。

# R\_TMU\_SetCounter

TMU 設定関数

カウンタの初期化処理

|                     |                                                                                                                                                                                                                        |                    |      |                    |                     |  |           |                    |  |         |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|------|--------------------|---------------------|--|-----------|--------------------|--|---------|
| 書式                  | <pre>#include "r_common.h" #include "iodefine.h" #include "r_tmu_if.h" char_t R_TMU_SetCounter ( uchar_t uc_ch_num, ulong_t ul_constant, ulong_t ul_counter );</pre>                                                   |                    |      |                    |                     |  |           |                    |  |         |
| 引数                  | <table><tr><td>uchar_t uc_ch_num</td><td> </td><td>TMU のチャンネル番号 (0~5)</td></tr><tr><td>ulong_t ul_constant</td><td> </td><td>タイマコンスタント</td></tr><tr><td>ulong_t ul_counter</td><td> </td><td>タイマカウンタ</td></tr></table> | uchar_t uc_ch_num  |      | TMU のチャンネル番号 (0~5) | ulong_t ul_constant |  | タイマコンスタント | ulong_t ul_counter |  | タイマカウンタ |
| uchar_t uc_ch_num   |                                                                                                                                                                                                                        | TMU のチャンネル番号 (0~5) |      |                    |                     |  |           |                    |  |         |
| ulong_t ul_constant |                                                                                                                                                                                                                        | タイマコンスタント          |      |                    |                     |  |           |                    |  |         |
| ulong_t ul_counter  |                                                                                                                                                                                                                        | タイマカウンタ            |      |                    |                     |  |           |                    |  |         |
| 戻り値                 | <table><tr><td>RET_NORMAL</td><td>正常終了</td></tr><tr><td>RET_ERR_PARAM1</td><td>第 1 引数不正</td></tr></table>                                                                                                              | RET_NORMAL         | 正常終了 | RET_ERR_PARAM1     | 第 1 引数不正            |  |           |                    |  |         |
| RET_NORMAL          | 正常終了                                                                                                                                                                                                                   |                    |      |                    |                     |  |           |                    |  |         |
| RET_ERR_PARAM1      | 第 1 引数不正                                                                                                                                                                                                               |                    |      |                    |                     |  |           |                    |  |         |
| 解説                  | 本関数は、TMU のチャンネル番号で選択したチャンネルのタイマコンスタントとタイマカウンタを設定します。                                                                                                                                                                   |                    |      |                    |                     |  |           |                    |  |         |
| 注意                  | 本関数を実行する前に、選択したチャンネルのカウンタ動作を停止させてください。カウンタ動作の停止には R_TMU_TimerStop()関数を実行してください。                                                                                                                                        |                    |      |                    |                     |  |           |                    |  |         |

# R\_TMU\_TimerStart

タイマカウント関数

カウント動作の開始処理

|     |                                                                                                                                                                                                                                                                                 |                                                                                                                                                                  |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 書式  | <pre>#include "r_common.h" #include "iodefine.h" #include "r_tmu_if.h" char_t R_TMU_TimerStart ( uchar_t uc_ch_num, uchar_t uc_start_mode, void ( *pv_unf_callback )( void ), void ( *pv_icpf_callback )( void ) );</pre>                                                       |                                                                                                                                                                  |
| 引数  | <pre>uchar_t uc_ch_num uchar_t uc_multi_start void ( *pv_unf_callback )( void ) void ( *pv_icpf_callback )( void )</pre>                                                                                                                                                        | <pre>  TMU のチャンネル番号 (0~5)   マルチスタート (0 : 無効 1 : 有効)   アンダフロー発生時のコールバックアドレス   (NULL の場合、コールバックなし)   インพุットキャプチャ発生時のコールバックアドレス   (チャンネル 2 以外では NULL にしてください)</pre> |
| 戻り値 | <pre>RET_NORMAL RET_ERR_PARAM1</pre>                                                                                                                                                                                                                                            | <pre>正常終了 第 1 引数不正</pre>                                                                                                                                         |
| 解説  | <p>本関数は、TMU のチャンネル番号で選択したチャンネルのカウント動作を開始します。<br/>アンダフローが発生した場合と、インพุットキャプチャが発生した場合(チャンネル 2 のみ)、当該コールバック関数がコールされます。</p> <p>チャンネル 0、1、2、またはチャンネル 3、4、5 のカウント動作を同時に開始する場合、マルチスタートを有効にしてください。この場合、選択したチャンネルの開始準備のみ行い、カウント動作は開始しません。カウント動作の開始には R_TMU_MultiTimerStart()関数を使用します。</p> |                                                                                                                                                                  |
| 注意  | なし                                                                                                                                                                                                                                                                              |                                                                                                                                                                  |



# R\_TMU\_MultiTimerStart

タイマカウント関数

カウント動作の開始処理(マルチスタート)

|     |                                                                                                                                     |                                   |
|-----|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| 書式  | <pre>#include "r_common.h" #include "iodefine.h" #include "r_tmu_if.h" char_t R_TMU_MultiTimerStart ( uchar_t uc_ch_select );</pre> |                                   |
| 引数  | uchar_t uc_ch_select                                                                                                                | I チャンネルセレクト(H'01~H'07, H'81~H'87) |
| 戻り値 | RET_NORMAL<br>RET_ERR_PARAM1                                                                                                        | 正常終了<br>第 1 引数不正                  |

**解説** 本関数は、チャンネルセレクトで選択したチャンネルのカウント動作を開始します。  
チャンネル 0、1、2、またはチャンネル 3、4、5 のカウント動作は同時に開始することができます。

### 【チャンネルセレクト】

| ビット      | チャンネル 0、1、2                           | チャンネル 3、4、5                           |
|----------|---------------------------------------|---------------------------------------|
| bit[7]   | 0: チャンネル 0、1、2 を選択する                  | 1: チャンネル 3、4、5 を選択する                  |
| bit[6:3] | 未使用<br>B'0000 にしてください                 | 未使用<br>B'0000 にしてください                 |
| bit[2]   | 0: チャンネル 2 を開始しない<br>1: チャンネル 2 を開始する | 0: チャンネル 5 を開始しない<br>1: チャンネル 5 を開始する |
| bit[1]   | 0: チャンネル 1 を開始しない<br>1: チャンネル 1 を開始する | 0: チャンネル 4 を開始しない<br>1: チャンネル 4 を開始する |
| bit[0]   | 0: チャンネル 0 を開始しない<br>1: チャンネル 0 を開始する | 0: チャンネル 3 を開始しない<br>1: チャンネル 3 を開始する |

※チャンネルセレクトにて選択していないチャンネルの設定は変更しません。

**注意** 本関数を実行する前に、選択したチャンネルの開始準備を行ってください。開始準備は R\_TMU\_TimerStart()関数で行います。

---

## R\_TMU\_TimerStop

タイマカウント関数

カウント動作の停止処理

---

|     |                                                                                                                           |
|-----|---------------------------------------------------------------------------------------------------------------------------|
| 書式  | <pre>#include "r_common.h" #include "iodefine.h" #include "r_tmu_if.h" char_t R_TMU_TimerStop( uchar_t uc_ch_num );</pre> |
| 引数  | uchar_t uc_ch_num                         TMU のチャンネル番号 (0~5)                                                              |
| 戻り値 | RET_NORMAL                                 正常終了<br>RET_ERR_PARAM1                             第 1 引数不正                    |
| 解説  | 本関数は、TMU のチャンネル番号で選択したチャンネルのカウント動作を停止します。                                                                                 |
| 注意  | なし                                                                                                                        |

## R\_TMU\_MultiTimerStop

タイマカウント関数

カウント動作の停止処理(マルチストップ)

```
書式 #include "r_common.h"
 #include "iodefine.h"
 #include "r_tmu_if.h"
 char_t R_TMU_MultiTimerStop(uchar_t uc_ch_select);
```

引数 uchar\_t uc\_ch\_select | チャンネルセレクト(H'01~H'07, H'09~H'0F)

戻り値 RET\_NORMAL 正常終了  
 RET\_ERR\_PARAM1 第1引数不正

解説 本関数は、チャンネルセレクトで選択したチャンネルのカウント動作を停止します。  
 チャンネル0、1、2、またはチャンネル3、4、5のカウント動作を同時に停止することができます。

## 【チャンネルセレクト】

| ビット      | チャンネル0、1、2                        | チャンネル3、4、5                        |
|----------|-----------------------------------|-----------------------------------|
| bit[7]   | 0: チャンネル0、1、2を選択する                | 1: チャンネル3、4、5を選択する                |
| bit[6:3] | 未使用<br>B'0000にしてください              | 未使用<br>B'0000にしてください              |
| bit[2]   | 0: チャンネル2を停止しない<br>1: チャンネル2を停止する | 0: チャンネル5を停止しない<br>1: チャンネル5を停止する |
| bit[1]   | 0: チャンネル1を停止しない<br>1: チャンネル1を停止する | 0: チャンネル4を停止しない<br>1: チャンネル4を停止する |
| bit[0]   | 0: チャンネル0を停止しない<br>1: チャンネル0を停止する | 0: チャンネル3を停止しない<br>1: チャンネル3を停止する |

※チャンネルセレクトにて選択していないチャンネルの設定は変更しません。

注意 なし

---

## R\_TMU\_GetCounter

情報取得関数

カウンタ値の取得処理

---

- 書式**            `#include "r_common.h"`  
                  `#include "iodefine.h"`  
                  `#include "r_tmu_if.h"`  
                  `ulong_t R_TMU_GetCounter( uchar_t uc_ch_num );`
- 引数**            `uchar_t uc_ch_num`                    |    TMU のチャンネル番号 (0~5)
- 戻り値**        `0~H'FFFFFFFF`                    |    タイマカウンタの現在値
- 解説**            本関数は、タイマカウンタの現在値を取得します。R\_TMU\_TimerStart()関数をコールした直後と、R\_TMU\_TimerStop()関数をコールした直後に本関数を使用することで、この間のカウント値を求めることが可能であり、経過時間の測定に使用できます。
- 注意**            本関数は、第 1 引数の入力確認を行いません。TMU のチャンネル番号に誤りがあった場合、動作は保障できません。

---

## R\_TMU2\_GetInputCapture

情報取得関数

インプットキャプチャレジスタ値の取得処理

---

|     |                                                                                                                       |
|-----|-----------------------------------------------------------------------------------------------------------------------|
| 書式  | <pre>#include "r_common.h" #include "iodefine.h" #include "r_tmu_if.h" ulong_t R_TMU2_GetInputCapture ( void );</pre> |
| 引数  | なし                                                                                                                    |
| 戻り値 | 0~H'FFFFFFFF                      インプットキャプチャレジスタの値                                                                    |
| 解説  | 本関数は、TMU のチャンネル 2 に搭載されるインプットキャプチャレジスタの値を取得します。インプットキャプチャの発生を確認後、本関数をコールしてください。                                       |
| 注意  | なし                                                                                                                    |

## R\_TMU\_GetFlag

情報取得関数

割り込みフラグの取得処理

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_tmu_if.h"
char_t R_TMU_GetFlag(uchar_t uc_ch_num);
```

引数

|                  |   |                    |
|------------------|---|--------------------|
| char_t uc_ch_num | I | TMU のチャンネル番号 (0~5) |
|------------------|---|--------------------|

戻り値

|                |          |
|----------------|----------|
| H'00~H'03      | ステータス情報  |
| RET_ERR_PARAM1 | 第 1 引数不正 |

解説

本関数は、インプットキャプチャ割り込みフラグ、アンダフローフラグを取得します。

【ステータス情報】各ビットが示す内容

| フラグ               | ビット位置  | 内容                  |
|-------------------|--------|---------------------|
| アンダフローフラグ         | bit[0] | 0 : 発生していない、1 : 発生中 |
| インプットキャプチャ割り込みフラグ | bit[1] | 0 : 発生していない、1 : 発生中 |

注意

なし

## R\_TMU\_ClearFlag

情報取得関数

割り込みフラグのクリア処理

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_tmu_if.h"
char_t R_TMU_ClearFlag(uchar_t uc_ch_num, uchar_t uc_flag);
```

引数

|                   |  |                    |
|-------------------|--|--------------------|
| uchar_t uc_ch_num |  | TMU のチャンネル番号 (0~5) |
| uchar_t uc_flag   |  | ステータスフラグ           |

戻り値

|                |          |
|----------------|----------|
| RET_ERR_PARAM1 | 第 1 引数不正 |
|----------------|----------|

解説

本関数は、引数に指定したステータスフラグに該当するフラグ(アンダフローフラグ、インプットキャプチャ割り込みフラグ)のクリア処理を行う。

【ステータスフラグ】各ビットが示す内容

| フラグ               | ビット位置  | 内容                   |
|-------------------|--------|----------------------|
| アンダフローフラグ         | bit[0] | 0 : クリアしない、1 : クリアする |
| インプットキャプチャ割り込みフラグ | bit[1] | 0 : クリアしない、1 : クリアする |

注意

なし

---

## R\_TMUn\_Interrupt(n=0 to 5)

割り込みハンドラ関数

アンダフロー割り込みハンドラ

---

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_tmu_if.h"
void R_TMUn_Interrupt (void);
```

引数           なし

戻り値        なし

解説

本割り込みハンドラ関数は、TMU 割り込み内での処理を行う。  
TMU の割り込みハンドラから、本割り込みハンドラ関数をコールすることにより、当該チャネルのアンダフロー割り込みを確認できる。  
本割り込みハンドラ関数で、アンダフロー割り込みを確認した場合、該当コールバック関数をコールする。

注意           本割り込みハンドラ関数は、TMU 割り込みのハンドラ内からコールしてください。



## R\_TMU2\_InCapInterrupt

割り込みハンドラ関数

インプットキャプチャ割り込みハンドラ

書式

```
#include "r_common.h"
#include "iodefine.h"
#include "r_tmu_if.h"
void R_TMU2_InCapInterrupt (void);
```

引数

なし

戻り値

なし

解説

本割り込みハンドラ関数は、TMU 割り込み内での処理を行う。  
TMU の割り込みハンドラから、本割り込みハンドラ関数をコールすることにより、チャンネル 2 のインプットキャプチャ割り込みを確認できる。  
本割り込みハンドラ関数で、インプットキャプチャ割り込みを確認した場合、該当コールバック関数をコールする。

注意

本割り込みハンドラ関数は、TMU 割り込みのハンドラ内からコールしてください。

## 4. 動作条件

### 4.1 割り込みについて

SH7753 グループの TMU は、チャンネル毎にアンダフロー割り込みを使用できます。また、チャンネル 2 ではインプットキャプチャ割り込みを使用できます。

表 3 に登録が必要な割り込みハンドラ関数を示します。

表 3. 割り込みベクタテーブル設定関数一覧

| チャンネル     | 割り込み要因番号 | 割り込み要因 | 割り込みハンドラ関数            |
|-----------|----------|--------|-----------------------|
| channel 0 | H'580    | TUNI0  | R_TMU0_Interrupt      |
| channel 1 | H'5A0    | TUNI1  | R_TMU1_Interrupt      |
| channel 2 | H'5C0    | TUNI2  | R_TMU2_Interrupt      |
|           | H'5E0    | TICPI2 | R_TMU2_InCapInterrupt |
| channel 3 | H'E00    | TUNI3  | R_TMU3_Interrupt      |
| channel 4 | H'E20    | TUNI4  | R_TMU4_Interrupt      |
| channel 5 | H'E40    | TUNI5  | R_TMU5_Interrupt      |

## 4.2 使用メモリ領域について

本デバイスドライバで使用するメモリ領域を表 4に示します。

表 4. メモリ使用量一覧

| 内容       | セクション | 属性            | バイト数     |
|----------|-------|---------------|----------|
| プログラムコード | P     | code, align=4 | 1656 バイト |
| 定数データ    | C     | data, align=4 | 0 バイト    |
| 初期値ありデータ | D     | data, align=4 | 0 バイト    |
| 初期値なしデータ | B     | data, align=4 | 24 バイト   |

5. 付録

5.1 ドライバ関数の使用例

TMU ドライバ関数を使用して、周期割り込みを発生させる手順を以下に示します。

5.1.1 TMU 周期割り込み使用例(単一チャンネル)

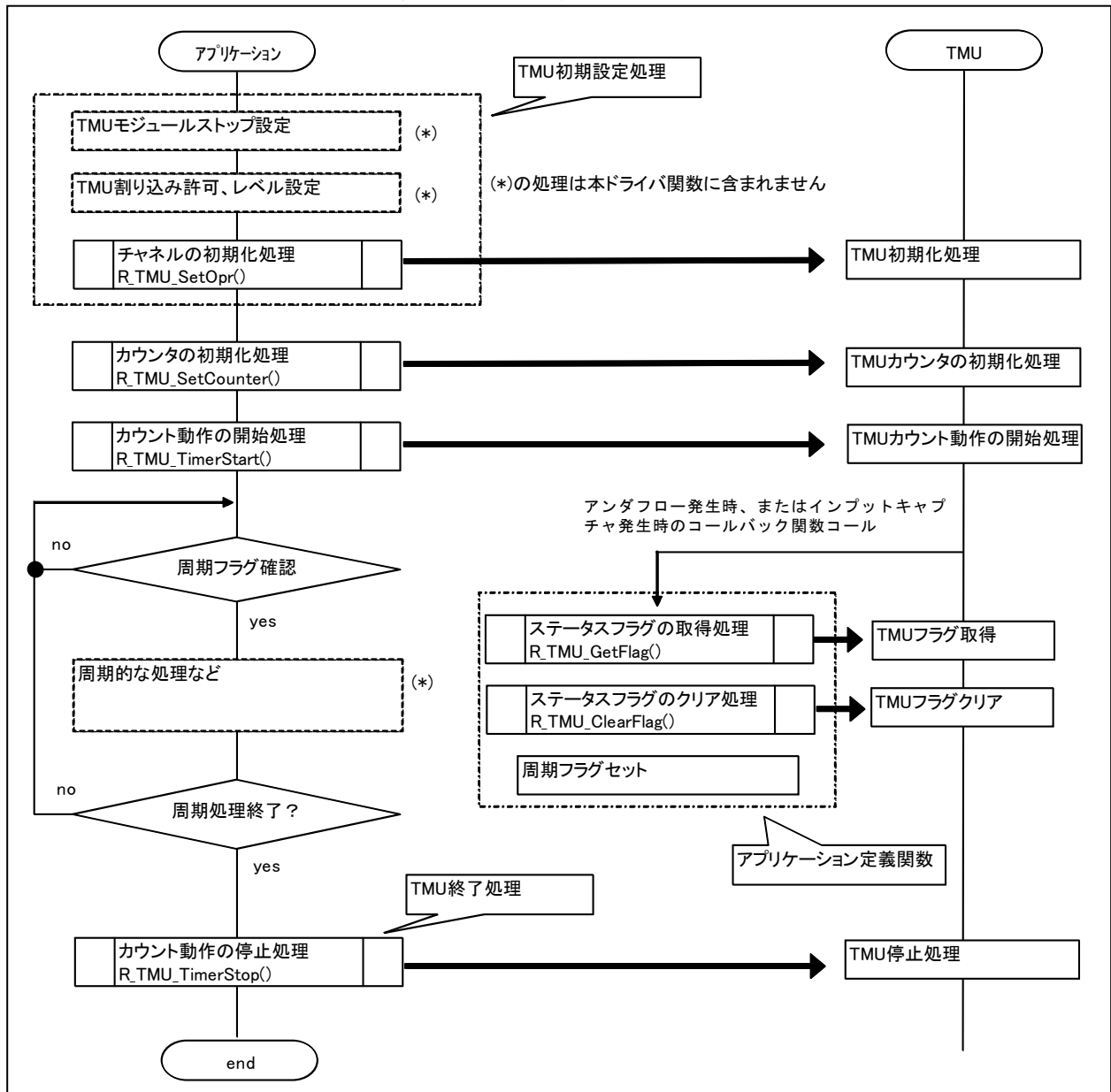


図 1. TMU 周期割り込みプログラム例(単一チャンネル)

5.1.2 TMU 周期割り込み使用例(マルチスタート)

チャンネル0 とチャンネル1 をマルチスタートする場合の使用例を以下に示します。

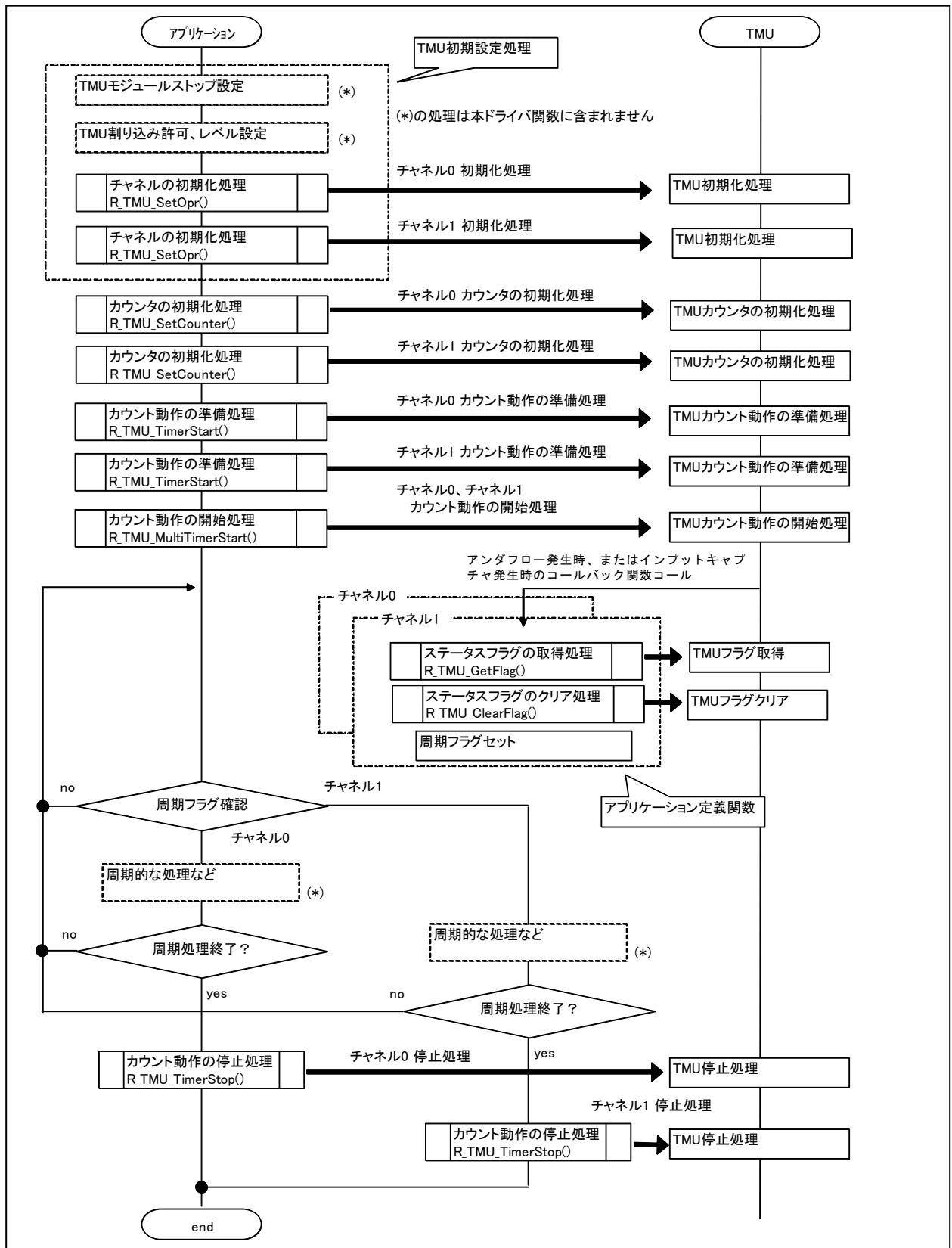


図 2. TMU 周期割り込みプログラム例(マルチスタート)

5.1.3 TMU 経過時間測定の使用例

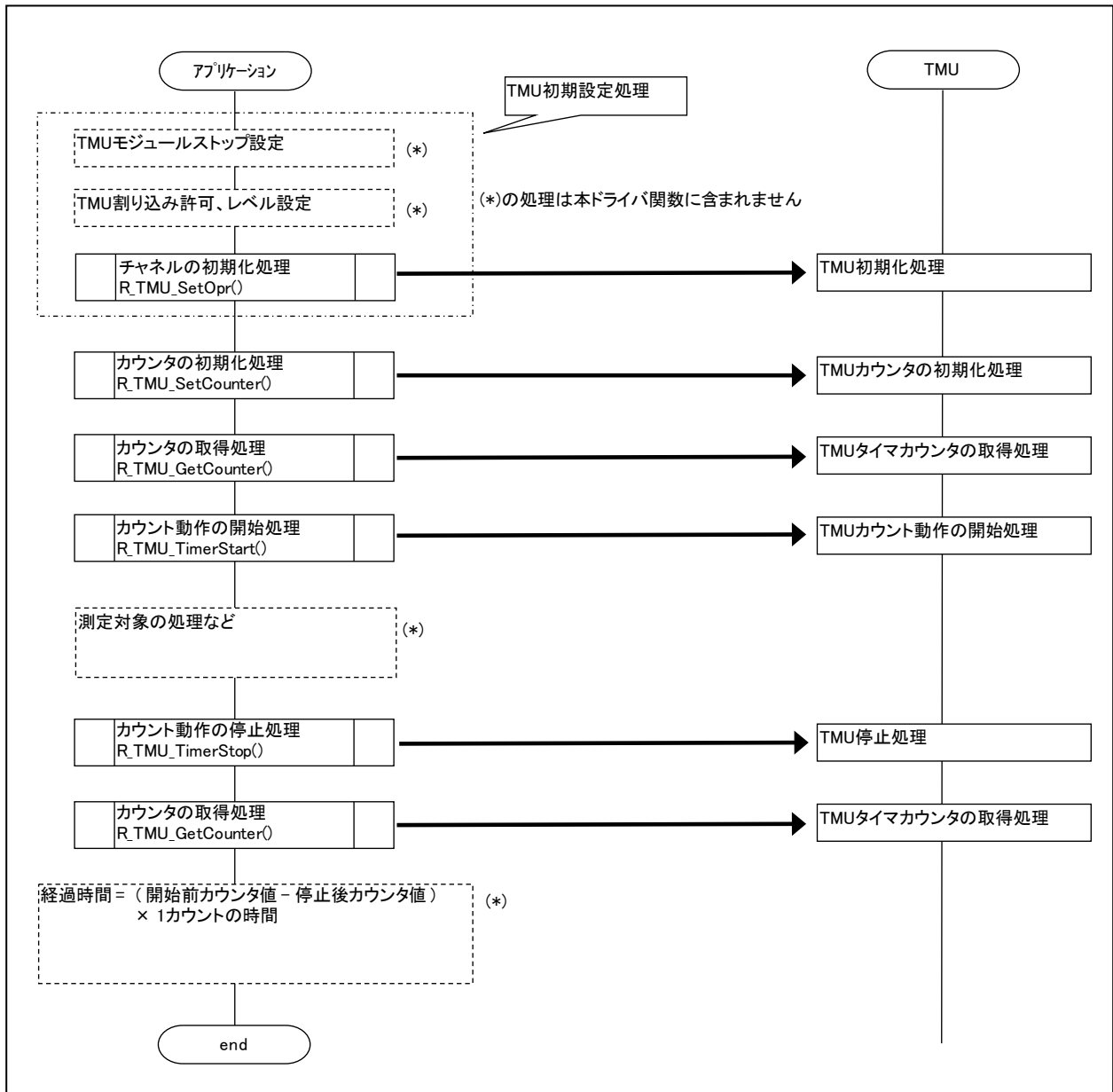


図 3. TMU 経過時間測定プログラム例

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/contact/>





## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、  
防災・防犯装置、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っていません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

- 注1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



営業お問合せ窓口

<http://www.renesas.com>

営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町 2-6-2 (日本ビル)

技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口： <http://japan.renesas.com/contact/>