SH7753 グループ
PBI ドライバソフトウェア

R01AN1619JJ0100
Rev.1.00
2013.7.1

要旨

本仕様書は SH7753 グループの PBI ドライバについて説明します。

対象デバイス

SH7753

動作環境

本仕様書に示す PBI 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 4.09.00.007
  : 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

機能概要

本ドライバは SH7753 に搭載されている PBI を使用し、PCIe インタフェースを持つデバイスと通信する為の関数群で構成されています。

1.1 主な機能

PBI ドライバの主な機能をを以下に示す。

・MailBox による PCIe のホストとの通信

・ShardMemory による PCIe のホストとの通信

・VDM による PCIe どうしの送受信

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　ハードウエア編

2. ドライバ仕様

2.1 関数一覧

表 1に、PBI ドライバの関数一覧を示す。

表 1. PBI ドライバ関数一覧

分類	関数名	機能概要	スタック サイズ
MailBox 処理	R_PBI_DmdMbxOsp	オーナシップ取得要求	4
	R_PBI_GetMbxOsp	オーナシップ値取得	4
	R_PBI_GetMbxData	MailBox Data を取得	4
	R_PBI_SetMbxData	MailBox Data を設定	12
	R_PBI_EndMbxData	MailBox Data 設定終了	16
	R_PBI_AbtMbx	Mbox のアボート処理	0
	R_PBI_IniMbx	Mbox 初期化	4
	R_PBI_EnaMbxIntr	Mbox 割込み可能設定	0
	R_PBI_DisMbxIntr	Mbox 割込み不可設定	0
	R_PBI_AstMbxHost	ホストに割込み発生通知	4
	R_PBI_SetRstCb	Reset 通知要求	4
	R_PBI_Interrupt	PBI 割込みハンドラ設定	24
	R_PBI_CancelCb	登録した CallBack のキャンセル	4
	R_PBI_GetErr	エラーの取得	0
	R_PBI_ClrErr	エラーのクリア	4
VDM 処理	R_PBI_RdyVDMrcv	VDM 受信準備	12
	R_PBI_StaVDMrcv	VDM 受信開始	4
	R_PBI_EndVDMrcv	VDM 受信終了処理	4
	R_PBI_RdyVDMsnd	VDM 送信準備	8
	R_PBI_StaVDMsnd	VDM 送信開始	4
	R_PBI_EndVDMsnd	VDM 送信終了処理	4
	R_PBI_VDMInterrupt	VDM 割込み処理	20
	R_PBI_EnaVDMIntr	VDM 割込み可能設定	0
	R_PBI_DisVDMIntr	VDM 割込み不可設定	0
	R_PBI_CancelVDMCb	登録した CallBack のキャンセル	4
	R_PBI_GetVDMErr	VDM エラーの取得	0
	R_PBI_ClrVDMErr	VDM エラーのクリア	20

2.2 コールバック関数仕様一覧

表 2に PBI ドライバのコールバック関数仕様一覧を示す。PBI ドライバのコールバック関数は各ドライバ関数の処理完了時に通知としてコールする。

表 2. PBI 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_PBI_GetMbxData	void (*pv_getmbxdata_end_callback)(ulong_t ul_event_flag)	MailBox Data 取得完了時のコールバック。（NULL の場合、コールバックなし）
R_PBI_SetMbxData	void (*pv_setmbxdata_end_callback)(ulong_t ul_event_flag)	MailBox Data 送信完了時のコールバック。（NULL の場合、コールバックなし）
R_PBI_SetRstCb	void (*pv_reset_end_callback)(ulong_t ul_event_flag)	Reset 要求取得時のコールバック。（NULL の場合、コールバックなし）
R_PBI_StaVDMrcv	void (*pv_rcvVDM_end_callback) (ulong_t ul_event_flag, ulong_t ul_rx_error)	VDM 送信完了時のコールバック。（NULL の場合、コールバックなし）
R_PBI_StaVDMsnd	void (*pv_sndVDM_end_callback)(ulong_t ul_event_flag, ulong_t ul_tx_error)	VDM 送信完了時のコールバック。（NULL の場合、コールバックなし）

2.3 エラーコード一覧

表 3に PBI ドライバのエラーコード一覧を示す。

表 3. PBI ドライバエラーコード一覧

値	エラーコード（マクロ定義）	エラー内容
0	RET_NORMAL	正常終了
-1	RET_ERR_PARAM1	第 1 引数不正
-2	RET_ERR_PARAM2	第 2 引数不正
-21	RET_ERR_OVRFLOW	Data オーバーフロー
-22	RET_ERR_UNDFLOW	Data アンダーフロー

3. 関数

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

関数名 機能概要	分類
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、必ずインクルードする。
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する。
戻り値	関数の戻り値を示す。
解説	関数の仕様について説明する。
注意事項	注意事項があればここに示す。

R_PBI_DmdMbxOsp

PBI 設定関数

PBI オーナシップ取得要求処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
char_t  R_PBI_DmdMbxOsp(  void  );
```

引数

戻り値	RET_NORMAL	正常終了

解説　本関数は、MailBox の OwnerShip を取得するための要求を出す。

注意

R_PBI_GetMbxOsp

PBI 設定関数

PBI オーナシップ値取得処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_GetMbxOsp( uchar_t * puc_wtowner );
```

引数			
uchar_t * puc_wtowner	O	BSTATUS.wtowner 値 10:HOST の勝ち、01:SH の勝ち、00:クリア	

戻り値	
RET_NORMAL	正常終了

解説 本関数は、レジスタの wtowner 値を取得する

注意

R_PBI_GetMbxData

PBI 設定関数

PBI MailBox データ取得処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_GetMbxData(uchar_t *puc_data_addr,ulong_t *pul_data_size,
void (*pv_getmbxdata_end_callback)(ulong_t ul_event_flag));
```

引数			
	uchar_t * puc_data_addr	I	受信する data を格納するアドレス 4K 固定サイズの data
	ulong_t *pul_data_size	O	取得した data サイズ
	void (*pv_getmbxosp_end_callback) (ulong_t ul_event_flag)	I	MailBox Data 取得完了時のコールバック。（NULL の場合、コールバックなし）
戻り値	RET_NORMAL		正常終了

解説 本関数は、MailBox からの Data 取得関数である

注意

R_PBI_SetMbxData

PBI 設定関数

PBI MailBox データ設定処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_SetMbxData(uchar_t * puc_data_addr, ulong_t ul_data_size,
void (*pv_setmbxdata_end_callback )( ulong_t ul_event_flag));
```

引数

uchar_t * puc_data_addr	I	送信する data を格納するアドレス
ulong_t ul_data_size	I	data のサイズ
void (*pv_setmbxdata_end_callback) (ulong_t ul_event_flag)	I	MailBox Data 送信完了時のコールバック。（NULL の場合、コールバックなし）

戻り値

RET_NORMAL	正常終了
RET_ERR_OVRFLOW	Data 設定エラー(BSTATUS.OVFLOW が UP している)
RET_ERR_UNDFLOW	Data 設定エラー(BSTATUS.UDFLOW が UP している)
RET_ERR_PARAM2	サイズが 0＜ul_data_size≦4096 の範囲でない

解説

本関数は、MailBox への Data 設定関数である

注意

R_PBI_EndMbxData

PBI 設定関数

PBI MailBox 送信終了処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_EndMbxData( void );
```

引数

戻り値	RET_NORMAL	正常終了

解説

本関数は、Data 設定終了処理関数である
１）HSTATUS.user int に 1 を書き込み
２）HUSRSTSCTL.bit24:16=H‘101 設定
を行う

注意

R_PBI_AbtMbx

PBI 設定関数

PBI MailBox のアボート処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_AbtMbx( void );
```

引数

戻り値	RET_NORMAL	正常終了

解説

本関数は、MailBox の処理をアボートする関数である
BCTL のリセットビットを設定し MailBox をリセットする

注意

R_PBI_IniMbx

PBI 設定関数

PBI 初期化処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_IniMbx( void );
```

引数

戻り値

解説 本関数は、MailBox 初期化処理関数である
BSTATUS の OVFLW/ UDRUN/ USERINT/ RESET フラグをクリアする
BINTEN の DONEE/USERINIE/RESETE を Enable にする

注意

R_PBI_EnaMbxIntr

PBI 設定関数

PBI MailBox 割込み許可処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
void  R_PBI_EnaMbxIntr(  ulong_t  ul_binten_intr  );
```

引数

ulong_t ul_intr_kind	I	許可する割込みビットを ON する	

戻り値

解説

本関数は、割込み設定関数である

【割込みビットマクロ定義】

PBI_BINTEN_DONEE： Done Interrupt Enable
BSTATUSレジスタの
DONEビット設定時SH-4Aに割り込み要求（PBIA）を送信します。

PBI_BINTEN_USERINTE：User Interrupt Enable
BSTATUSレジスタの
USERINTビット設定時SH-4Aに割り込み要求（PBIA）を送信します。

PBI_BINTEN_OVFLWE：Overflow Interrupt Enable
BSTATUSレジスタの
OVFLWビット設定時SH-4Aに割り込み要求（PBIA）を送信します。

PBI_BINTEN_UDRUNE:Underrun Interrupt Enable
BSTATUSレジスタの
UDRUN ビット設定時 SH-4A に割り込み要求（PBIA）を送信します。

PBI_BINTEN_RESETE:Reset Interrupt Enable
BSTATUSレジスタの
RESET ビット設定時 SH-4A に割り込み要求（PBIA）を送信します。

注意

R_PBI_DisMbxIntr

PBI 設定関数

PBI 割込み不可設定処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_DisMbxIntr( ulong_t ul_binten_intr );
```

引数			
ulong_t ul_intr_kind	I	許可しない割込みビットを ON する	

戻り値

解説

本関数は、割込み禁止設定関数である

【割込みビットマクロ定義】

PBI_BINTEN_DONEE： Done Interrupt Enable
BSTATUSレジスタの
DONEビット設定時SH-4Aに割り込み要求（PBIA）を送信しません。

PBI_BINTEN_USERINTE：User Interrupt Enable
BSTATUSレジスタの
USERINTビット設定時SH-4Aに割り込み要求（PBIA）を送信しません。

PBI_BINTEN_OVFLWE：Overflow Interrupt Enable
BSTATUSレジスタの
OVFLWビット設定時SH-4Aに割り込み要求（PBIA）を送信しません。

PBI_BINTEN_UDRUNE:Underrun Interrupt Enable
BSTATUSレジスタの
UDRUN ビット設定時 SH-4A に割り込み要求（PBIA）を送信しません。

PBI_BINTEN_RESETE:Reset Interrupt Enable
BSTATUSレジスタの
RESET ビット設定時 SH-4A に割り込み要求（PBIA）を送信しません。

注意

R_PBI_AstMbxHost

PBI 設定関数

PBI HOST へ割込み発生処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_AstMbxHost( void );
```

引数

戻り値	RET_NORMAL	正常終了

解説 本関数は、Host への割込み発生関数である

注意

R_PBI_SetRstCb

PBI 設定関数

PBI Reset 通知 CallBack 登録処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_SetRstCb(void (*pv_reset_end_callback)(ulong_t ul_event_flag));
```

引数		
	void (*pv_reset_end_callback)(ulong_t ul_event_flag)	Reset 通知のコールバック。（NULL の場合、コールバックなし）

戻り値		
	RET_NORMAL	正常終了

解説 本関数は、Reset 通知 CallBack 登録を行う関数である

注意

R_PBI_Interrupt

PBI 設定関数

PBI 割込みハンドラ処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_Interrupt( void );
```

引数

戻り値

解説 本関数は、割込みハンドラ関数である、割込み要因により処理分岐する
割込み要因に応じて、それぞれのメッセージを PBI に送り、CallBack を実行させる

注意

R_PBI_CancelCb

PBI 設定関数

PBI　CallBack 取得要求キャンセル処理

書式

```
#include　“r_common.h”
#include　“iodefine.h”
#include　“r_pbi_if.h”
char_t　R_PBI_CancelCb(　uchar_t　uc_type　);
```

引数			
	uchar_t　uc_type	I	要求をキャンセルする TYPE

戻り値		
	RET_NORMAL	正常終了
	RET_ERR_PARAM1	登録可能な CallBack 関数 TYPE 以外が入力

解説　本関数は、登録した CallBack の登録キャンセル関数である

注意

R_PBI_GetErr

PBI 設定関数

PBI エラー取得処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_GetErr( pbi_err_t *pst_err );
```

引数			
pbi_err_t *pst_err	O	PBI エラー構造体へのポインタ	

戻り値

解説 本関数は、BSTATUS レジスタ状態取得関数である

注意

R_PBI_ClrErr

PBI 設定関数

PBI エラークリア処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_ClrErr( void );
```

引数

戻り値

解説 本関数は、エラークリア関数である(R_PBI_GetErr 関数で得られる、エラーBIT をクリア)

注意

R_PBI_RdyVDMrcv

PBI VDM 処理関数

PBI VDM 受信準備処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
char_t R_PBI_RdyVDMrcv( void );
```

引数

戻り値	RET_NORMAL	正常終了

解説

本関数は、VDM 受信時のレジスタを設定する

・VDMINTFR の RER/ RDFUL/ REMP/ REOM ビットが立っているなら落とす

・VDMRXERFR のエラー要因ビットをすべてクリア

注意

R_PBI_StaVDMrcv

PBI VDM 処理関数

PBI VDM 受信開始処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
char_t  R_PBI_StaVDMrcv( uchar_t  *puc_distbl_addr,
  void (*pv_rcvVDM_end_callback)(ulong_t ul_event_flag, ulong_t ul_tx_error));
```

引数			
	uchar_t *puc_distbl_addr	I	ディスクリプタの開始アドレス
	void (*pv_rcvVDM_end_callback)(ulong_t ul_event_flag, ulong_t ul_tx_error)	I	受信完了時実行される callback 関数
戻り値	RET_NORMAL		正常終了
	RET_ERR_PARAM1		puc_distbl_addr が 128Byte 境界にない

解説

本関数は、VDM 受信を開始する
・VDMRXTADRにディスクリプタの開始アドレスを設定
・VDMRXCMD.STRビットを設定

注意

受信ディスクリプタ・テーブルは付録参照

R_PBI_EndVDMrcv

PBI VDM 処理関数

PBI VDM 受信終了処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
char_t  R_PBI_EndVDMrcv(void);
```

引数

戻り値		
	RET_NORMAL	正常終了

解説

本関数は、VDM 受信の終了処理を行う
・VDMRXCMD.STR ビット 0 クリア
受信中断（アボート）する場合も本関数を使用する

注意

R_PBI_RdyVDMsnd

PBI VDM 処理関数

PBI VDM 送信準備処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
char_t  R_PBI_IniVDMsnd(void);
```

引数

戻り値	RET_NORMAL	正常終了

解説

本関数は、VDM 送信時のレジスタを初期化する
・VDMINTFR の TER/TINT ビットが立っていたら落とす
・VDMTXERFR のエラー要因ビットをすべてクリア

注意

R_PBI_StaVDMsnd

PBI VDM 処理関数

PBI VDM 送信開始処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
char_t  R_PBI_StaVDMsnd( uchar_t  *puc_distbl_addr,
void (*pv_sndVDM_end_callback)(ulong_t ul_event_flag, ulong_t ul_tx_error));
```

引数	uchar_t *puc_distbl_addr	I	ディスクリプタの開始アドレス
	void (*pv_sndVDM_end_callback)(ulong_t ul_event_flag, ulong_t ul_tx_error)	I	送信完了時実行される Callback 関数
戻り値	RET_NORMAL		正常終了
	RET_ERR_PARAM1		puc_distbl_addr が 128Byte 境界にない

解説

本関数は、VDM 送信開始処理をする
・VDMTXTADRにディスクリプタテーブルのアドレスを設定
・VDMTXCMD.STR ビットを設定

注意

R_PBI_EndVDMsnd

PBI VDM 処理関数

PBI VDM 送信終了処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
char_t  R_PBI_EndVDMsnd(void);
```

引数

戻り値		
	RET_NORMAL	正常終了

解説

本関数は、VDM 送信終了時の処理を行う
送信中断（アボート）する場合も本関数を使用する
・VDMTXCMD.STR ビット 0 クリア

注意

R_PBI_VDMInterrupt

PBI VDM 処理関数

PBI VDM 割込みハンドラ

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
void  R_PBI_VDMInterrupt(void);
```

引数

戻り値

解説　本関数は、VDM 送信受信時の割込み処理を行う

PBIB 割込み（H’2A20）ハンドラ
ハンドラ内処理
・VDMINTFR.REOM=1 かつ VDMINTEN.REOME=1 の場合
  write 1 in VDMINTFR.REOM bit
・VDMINTFR.RDFUL=1 かつ VDMINTEN.RDFULE=1 の場合
  write 1 in VDMINTFR.RDFUL bit
・VDMINTFR.REMP=1 かつ VDMINTEN.REMPE=1 の場合
  write 1 in VDMINTFR.REMP bit
・VDMINTFR.TINT=1 かつ VDMINTEN.TINTE=1 の場合
  write 1 in VDMINTFR.TINT bit

注意

R_PBI_EnaVDMIntr

PBI 設定関数

PBI MailBox 割込み許可処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
void  R_PBI_EnaVDIntr(ulong_t  ul_vdminten_intr,
                      ulong_t  ul_vdmtxeren_intr, ulong_t  ul_vdmrxeren_intr);
```

引数

ulong_t ul_vdminten_intr	I	許可する VDMINTEN 割込みビット
ulong_t ul_vdmtxeren_intr	I	許可する VDMTXEREN 割込みビット
ulong_t ul_vdmrxeren_intr	I	許可する VDMRXEREN 割込みビット

戻り値

解説

本関数は、割込み許可設定関数である

【割込みビットマクロ定義】

```
VDM_VDMINTEN_TINTE      : VDM TX Interrupt Enable
VDM_VDMINTEN_TERE       : VDM TX Error Interrupt Enable
VDM_VDMINTEN_REOME      : VDM RX EOM Interrupt Enable
VDM_VDMINTEN_REMPE      : VDM RX Buffer Empty Interrupt Enable
VDM_VDMINTEN_RDFULE     : VDM RX DDR Area Full Interrupt Enable
VDM_VDMINTEN_RERE       : VDM RX Error Interrupt Enable

VDM_VDMTXEREN_SBE       : System Bus Error Interrupt Enable
VDM_VDMTXEREN_MPSE      : MPS Error Interrupt Enable
VDM_VDMTXEREN_RSTE      : Reset Error Interrupt Enable
VDM_VDMTXEREN_REGIONE   : Region Error Interrupt Enable
VDM_VDMTXEREN_HDE       : Header Error Interrupt Enable
VDM_VDMTXEREN_LENE      : Length Error Interrupt Enable
VDM_VDMTXEREN_ TLPCHKE  : TLP Check Error Interrupt Enable

VDM_VDMRXEREN_SBE       : System Bus Error Interrupt Enable
VDM_VDMRXEREN_EOTE      : End of Table Interrupt Enable
VDM_VDMRXEREN_REGIONE   : Region Error Interrupt Enable
VDM_VDMRXEREN_LENE      : Length Error Interrupt Enable
```

注意

R_PBI_DisVDMIntr

PBI 設定関数

PBI 割込み不可設定処理

書式

```
#include  “r_common.h”
#include  “iodefine.h”
#include  “r_pbi_if.h”
void  R_PBI_DisVDMIntr(ulong_t  ul_vdminten_intr,
                       ulong_t  ul_vdmtxeren_intr, ulong_t  ul_vdmrxeren_intr,);
```

引数			
ulong_t ul_vdminten_intr	I	不許可する VDMINTEN 割込みビット
ulong_t ul_vdmtxeren_intr	I	不許可する VDMTXEREN 割込みビット
ulong_t ul_vdmrxeren_intr	I	不許可する VDMRXEREN 割込みビット

戻り値

解説

本関数は、割込み禁止設定関数である

【割込みビットマクロ定義】

```
VDM_VDMINTEN_TINTE      : VDM TX Interrupt Enable
VDM_VDMINTEN_TERE       : VDM TX Error Interrupt Enable
VDM_VDMINTEN_REOME      : VDM RX EOM Interrupt Enable
VDM_VDMINTEN_REMPE      : VDM RX Buffer Empty Interrupt Enable
VDM_VDMINTEN_RDFULE     : VDM RX DDR Area Full Interrupt Enable
VDM_VDMINTEN_RERE       : VDM RX Error Interrupt Enable

VDM_VDMTXEREN_SBE        : System Bus Error Interrupt Enable
VDM_VDMTXEREN_MPSE       : MPS Error Interrupt Enable
VDM_VDMTXEREN_RSTE       : Reset Error Interrupt Enable
VDM_VDMTXEREN_REGIONE    : Region Error Interrupt Enable
VDM_VDMTXEREN_HDE        : Header Error Interrupt Enable
VDM_VDMTXEREN_LENE       : Length Error Interrupt Enable
VDM_VDMTXEREN_ TLPCHKE   : TLP Check Error Interrupt Enable

VDM_VDMRXEREN_SBE       : System Bus Error Interrupt Enable
VDM_VDMRXEREN_EOTE      : End of Table Interrupt Enable
VDM_VDMRXEREN_REGIONE   : Region Error Interrupt Enable
VDM_VDMRXEREN_LENE      : Length Error Interrupt Enable
```

注意

R_PBI_CancelVDMCb

PBI 設定関数

PBI　VDM CallBack 取得要求キャンセル処理

書式

```
#include　“r_common.h”
#include　“iodefine.h”
#include　“r_pbi_if.h”
char_t　R_PBI_CancelVDMCb(　uchar_t　uc_type　);
```

引数			
	uchar_t　uc_type	I	要求をキャンセルする TYPE

戻り値		
	RET_NORMAL	正常終了
	RET_ERR_PARAM1	登録可能な VDM の CallBack 関数 TYPE 以外が入力

解説　本関数は、登録された VDM の CallBack 関数の登録キャンセル関数である

注意

R_PBI_GetVDMErr

PBI 設定関数

PBI エラー取得処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_GetVDMErr(vdm_err_t *pst_vdm_err );
```

引数			
vdm_err_t *pst_vdm_err	**O**	VDM エラー構造体へのポインタ	

戻り値

解説 本関数は、VDMRXSTS/VDMTXESTS/VDMRXERFR/VDMTXERFR/VDMINTFR
レジスタ状態取得関数である

・VDMRXSTS レジスタ値
ACT(Active)、ERR (ERROR) 、INT (Interrupt) 、BFUL (Buffer Full)

・VDMTXSTS レジスタ値
ACT(Active)、ERR (ERROR) 、INT (Interrupt)

・VDMRXERFRレジスタ値
SB(System Bus Error),EOT(End of Table),REGION(Region Error),LEN(Length Error)

・VDMTXERFRレジスタ値
SB(System Bus Error),MPS(MPS Error),RST(Reset Error),REGION(Region Error),
HD (Header Error) , LEN (Length Error) , TLPCHK (TLP Check Error)

・VDMINTFRレジスタ値
REA(VDM RX Error),RDFUL(VDM RX DDR Area Full), REMP(VDM RX Buffer Empty),
REOM(VDM RX EOM), TER(VDM TX Error), TINT(VDM TX Interrupt)

注意

R_PBI_ClrVDMErr

PBI 設定関数

PBI エラークリア処理

書式

```
#include “r_common.h”
#include “iodefine.h”
#include “r_pbi_if.h”
void R_PBI_ClrVDMErr(void);
```

引数

戻り値

解説

本関数は、VDMRXERFR/VDMTXERFR/VDMINTFR レジスタエラークリア関数である
・VDMRXERFRレジスタ値
SB(System Bus Error),EOT(End of Table),REGION(Region Error),LEN(Length Error)
・VDMTXERFRレジスタ値
SB(System Bus Error),MPS(MPS Error),RST(Reset Error),REGION(Region Error),
HD (Header Error) , LEN (Length Error) , TLPCHK (TLP Check Error)
・VDMINTFRレジスタ値
REA(VDM RX Error),RDFUL(VDM RX DDR Area Full), REMP(VDM RX Buffer Empty),
REOM(VDM RX EOM), TER(VDM TX Error), TINT(VDM TX Interrupt)
エラーをクリア

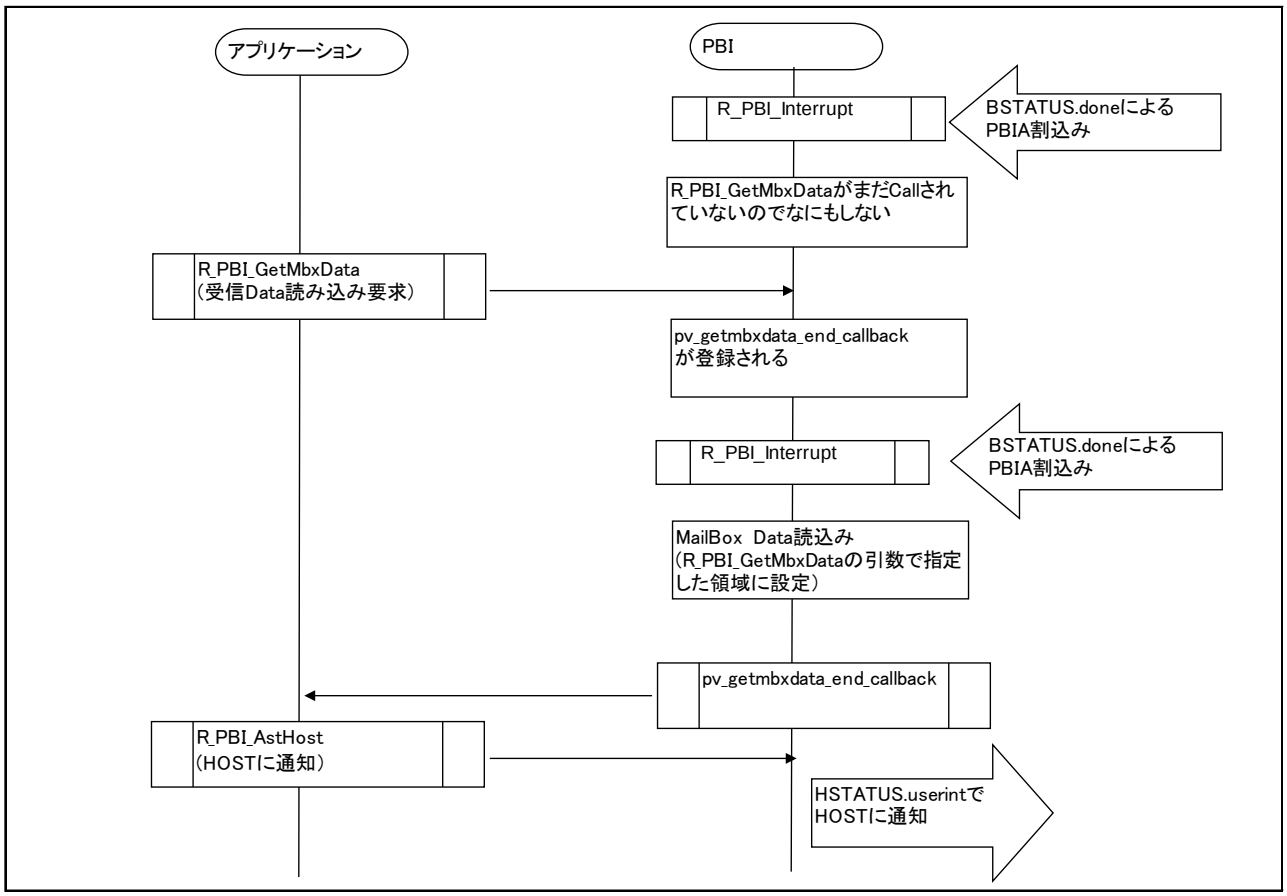
注意

4. 付録

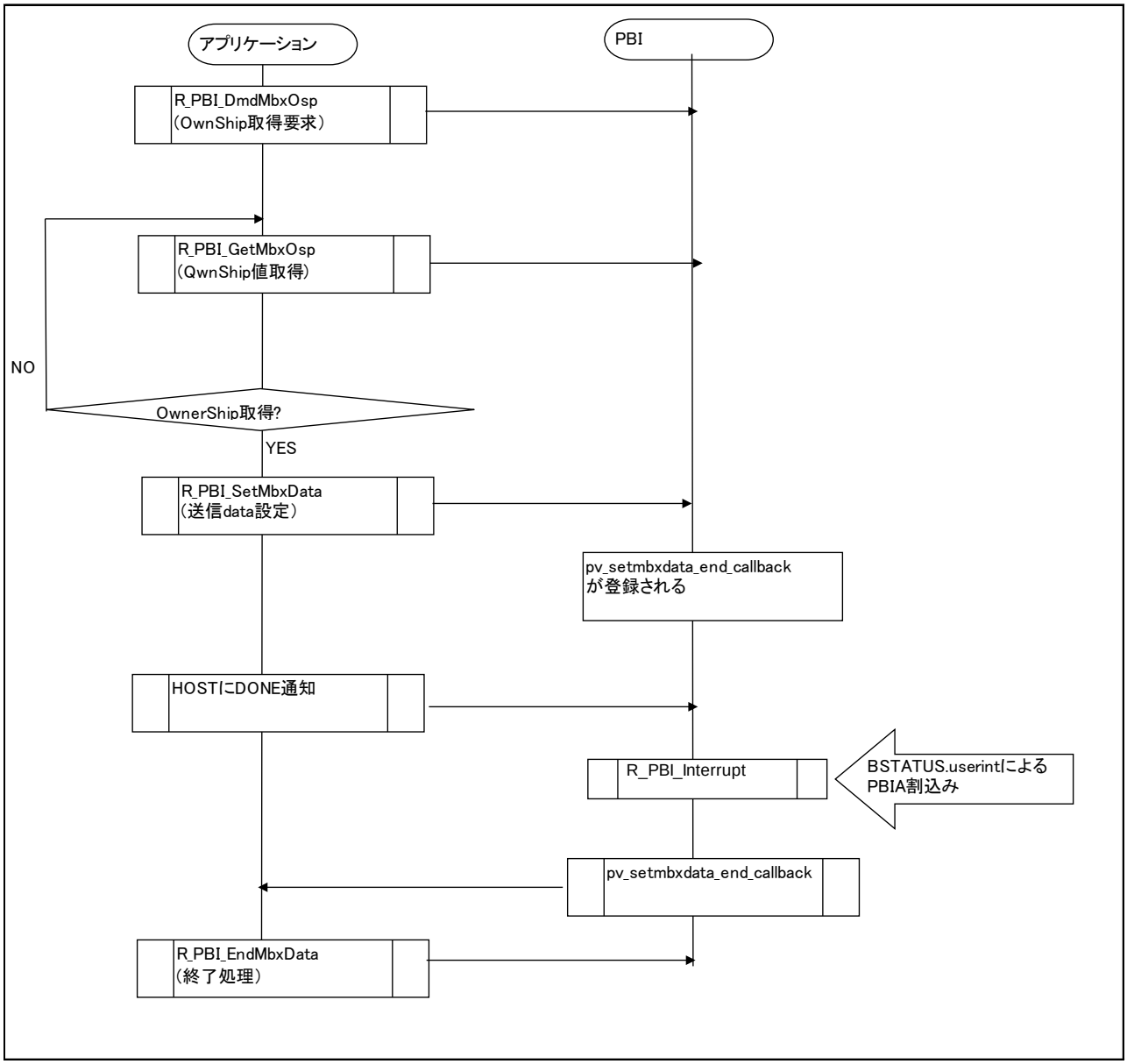
4.1 ドライバ関数の使用例

PBI ドライバの通信時の使用例を 1～5 に示す。

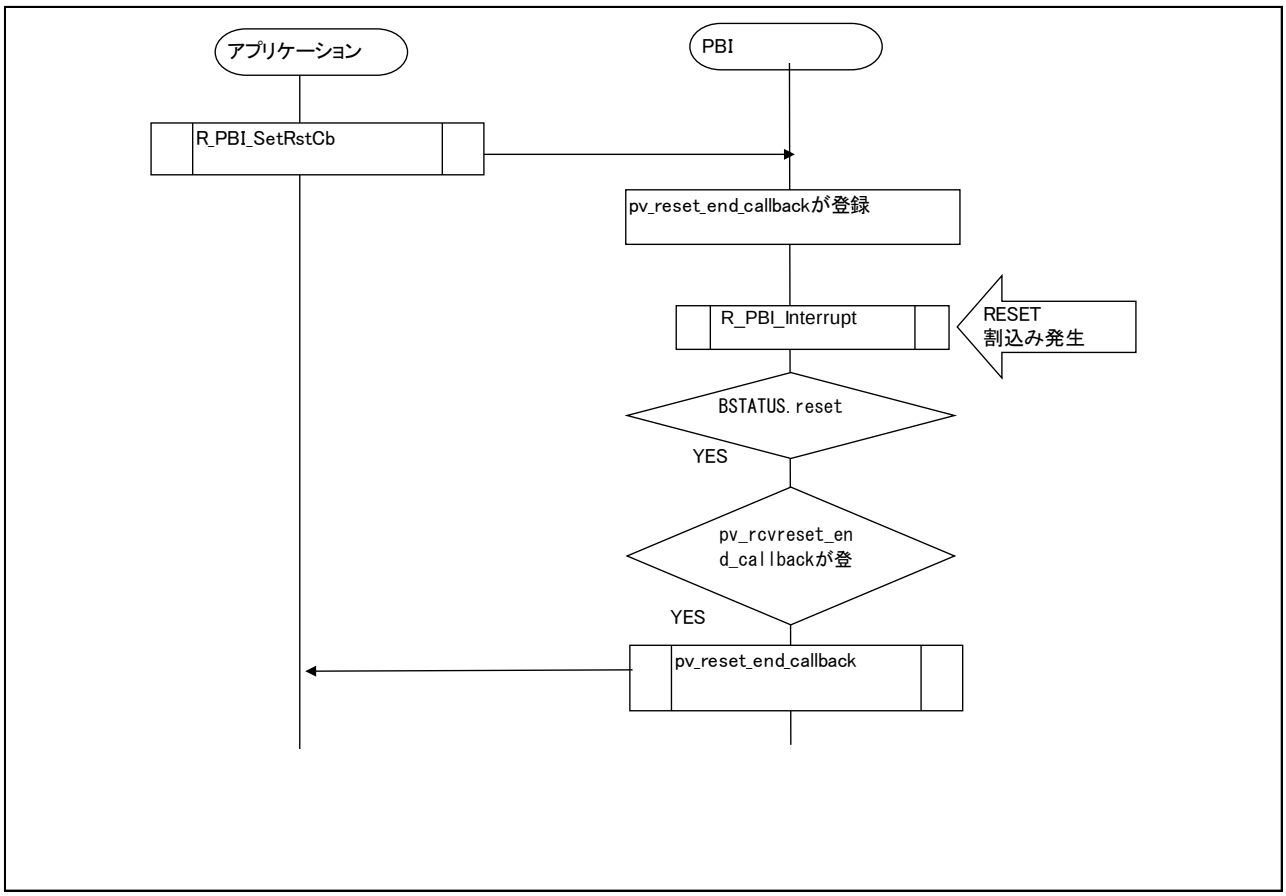
4.1.1 MailBox 受信処理



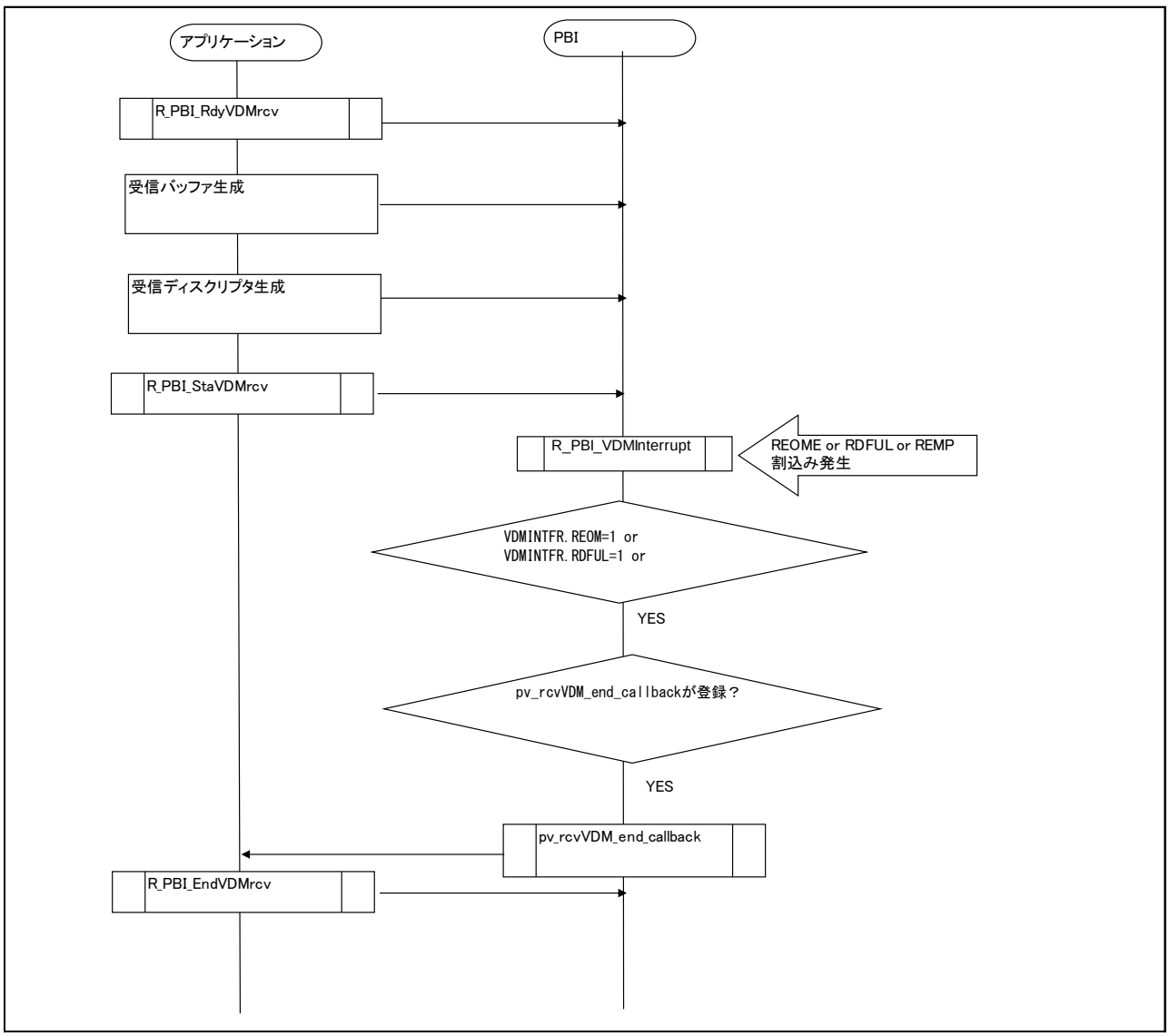
4.1.2 MailBox 送信処理



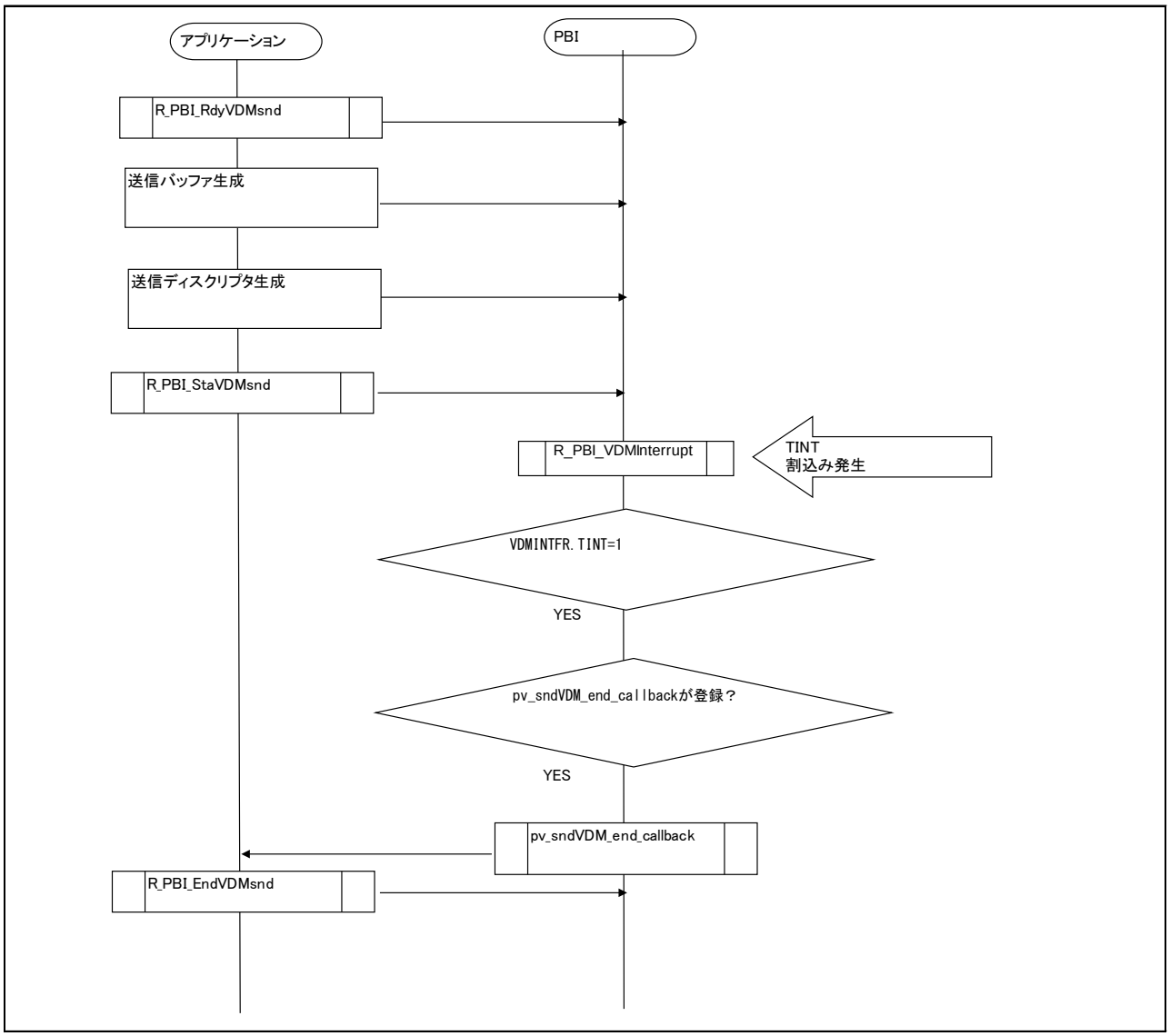
4.1.3 RESET 処理


アプリケーション
PBI
R_PBI_SetRstCb
pv_reset_end_callbackが登録
R_PBI_Interrupt
RESET
割込み発生
BSTATUS.reset
YES
pv_rcvreset_end_callbackが登
YES
pv_reset_end_callback

4.1.4 VDM 受信


アプリケーション
PBI
R_PBI_RdyVDMrcv
受信バッファ生成
受信ディスクリプタ生成
R_PBI_StaVDMrcv
R_PBI_VDMInterrupt
REOME or RDFUL or REMP
割込み発生
VDMINTFR.REOM=1 or
VDMINTFR.RDFUL=1 or
YES
pv_rcvVDM_end_callbackが登録？
YES
pv_rcvVDM_end_callback
R_PBI_EndVDMrcv

4.1.5 VDM 送信


アプリケーション
PBI
R_PBI_RdyVDMsnd
送信バッファ生成
送信ディスクリプタ生成
R_PBI_StaVDMsnd
R_PBI_VDMInterrupt
TINT
割込み発生
VDMINTFR.TINT=1
YES
pv_sndVDM_end_callbackが登録？
YES
pv_sndVDM_end_callback
R_PBI_EndVDMsnd

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/inquiry

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.7.01	—	初版発行

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）があります。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うと、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。型名が違う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ものですが、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も、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おりません。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用途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、お客様にて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。

ルネサス エレクトロニクス株式会社

■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/

Colophon 2.1