アプリケーションノート

EI2C CONTROLLER TRANS

Arm および Keil は、Arm Limited（またはその子会社）の米国およびその他の国における登録商標です。

この資料に記載されている社名・商品名・サービス名などは、それぞれ各社が商標として使用している場合があります。

目次

目次 2
1. はじめに 3
2. 用語 3
3. 関連するドキュメント 3
4. 対象サンプルプログラム 4
5. 構成図 4
6. サンプルプログラム：EI2C_CONTROLLER_TRANS 5
6.1. 動作・操作概要 5
6.2. 使用する機能 5
6.3. 使用する割り込み 5
6.4. コンフィグレーション 5
6.5. ターミナルソフト出力例 5
7. アクティビティ図 6
7.1. main 6
7.2. 割り込み 10
8. 改訂履歴 11
製品取り扱い上のお願い 12

1. はじめに

本書は、I2C インターフェース(EI2C)を用いるサンプルソフト EI2C_CONTROLLER_TRANS について記載しています。製品を開発する際、動作確認用または、プログラム開発の参考としてご利用願います。

2. 用語

用語／略語	定義
BSP	Board Support Package
CG	Clock Control and Operation Mode
I2C	Inter-Integrated Circuit
Timer	T32A：32-bit Timer Event Counter

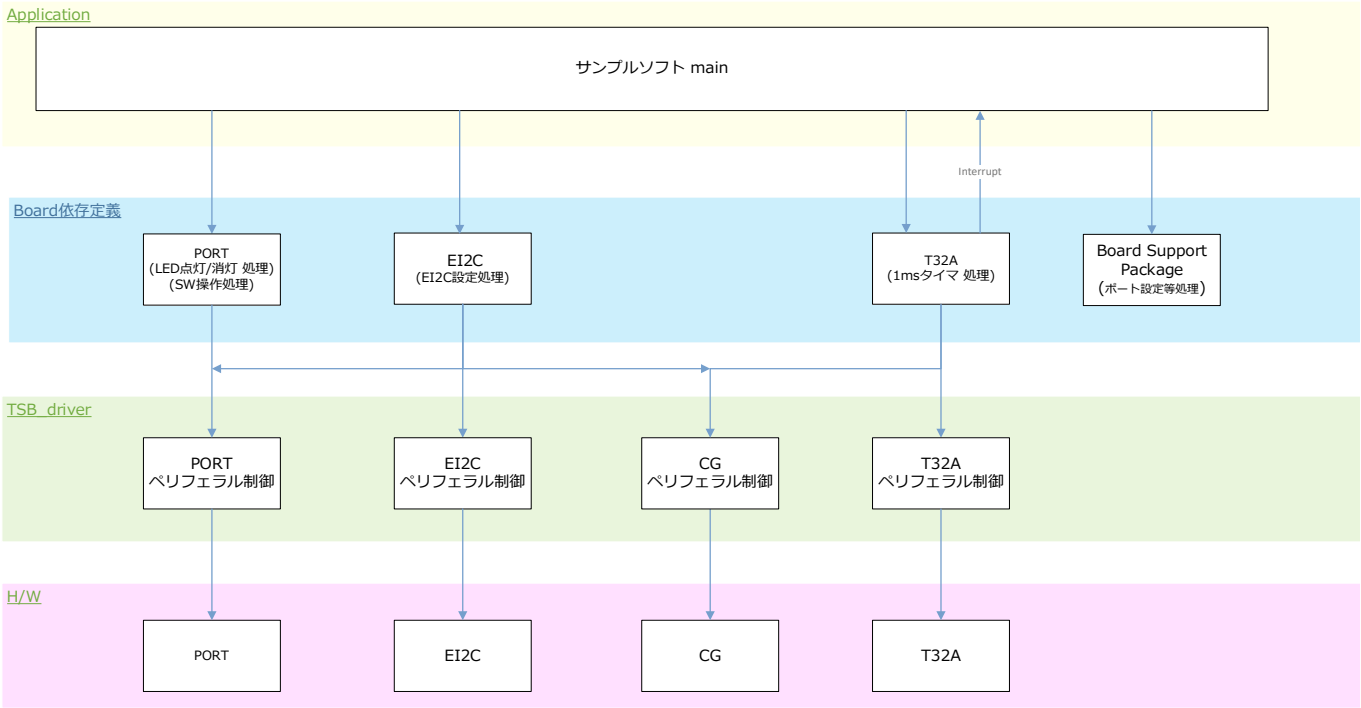
3. 関連するドキュメント

ドキュメント	備考
データシート	利用する MCU のデータシートを参照してください
リファレンスマニュアル	利用する各 IP のリファレンスマニュアルを参照してください
アプリケーションノート MCU 利用説明書	利用する MCU 利用説明書を参照してください

4. 対象サンプルプログラム

サンプルプログラム	概要
EI2C_CONTROLLER_TRANS	EI2C 通信機能(Controller 送信処理)のサンプル

5. 構成図



6. サンプルプログラム：EI2C_CONTROLLER_TRANS

EI2C 通信機能の Controller 送信処理機能を使用して、データ送信完了ごとに、LED の点灯・消灯を切り替えるサンプルソフトです。

6.1. 動作・操作概要

BSP_LED_1 と BSP_LED_3 を消灯します。BSP_PSW_1 が押下した場合、文字列を送信します。
送信する文字列は 6.4 コンフィグレーションで設定されます。
送信後 BSP_LED_1 の点灯状態(点灯/消灯)を切り替えます。
エラーが発生した場合は、BSP_LED_3 を点灯します。

6.2. 使用する機能

使用する機能は下記のとおりです。
各 Channel の Port 割り当ては、MCU 利用説明書を参照願います。

IP	チャネル	目的
EI2C	BSP_EI2C_1	EI2C 通信
T32A	BSP_T32A_TIMER_1	インターバルタイマー用
PORT (Push-Switch)	BSP_PSW_1	イベントトリガー用
PORT (LED)	BSP_LED_2	動作確認用
	BSP_LED_3	動作確認用

6.3. 使用する割り込み

MCU 利用説明書“サンプル毎の割り込み一覧”を参照願います。

6.4. コンフィグレーション

コンフィグレーション設定

コンフィグレーション	ソフト定義名	設定値 (デフォルト)	説明
EI2Cクロック	CFG_EI2C_FREQUENCY	800000	Controller 動作時周波数（単位： Hz）
文字列	BSP_MCU_NAME	（注 1）	本サンプルでは BSP_MCU_NAME で定義

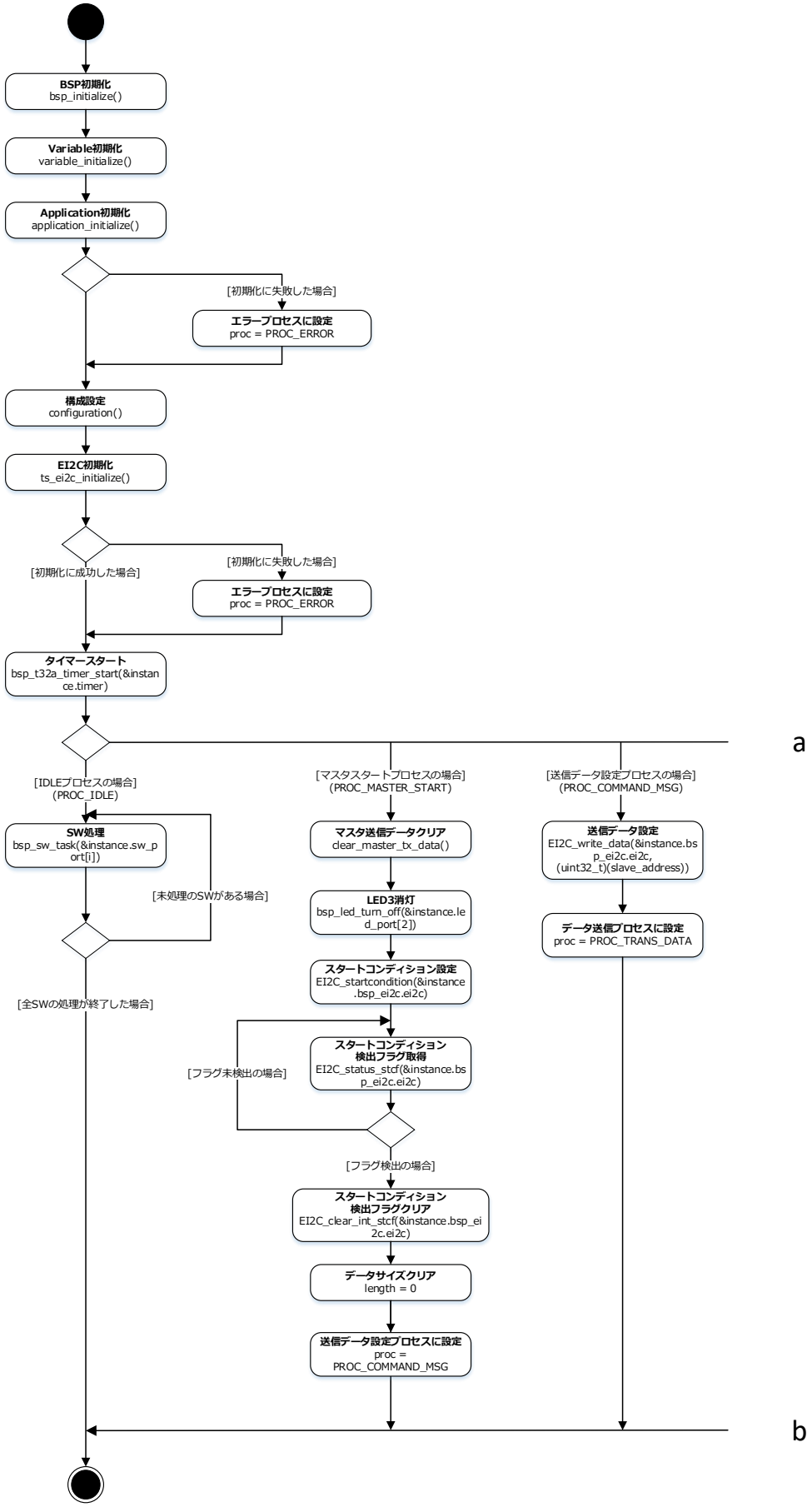
注 1： MCU 利用説明書”Log 制御その他”章 参照

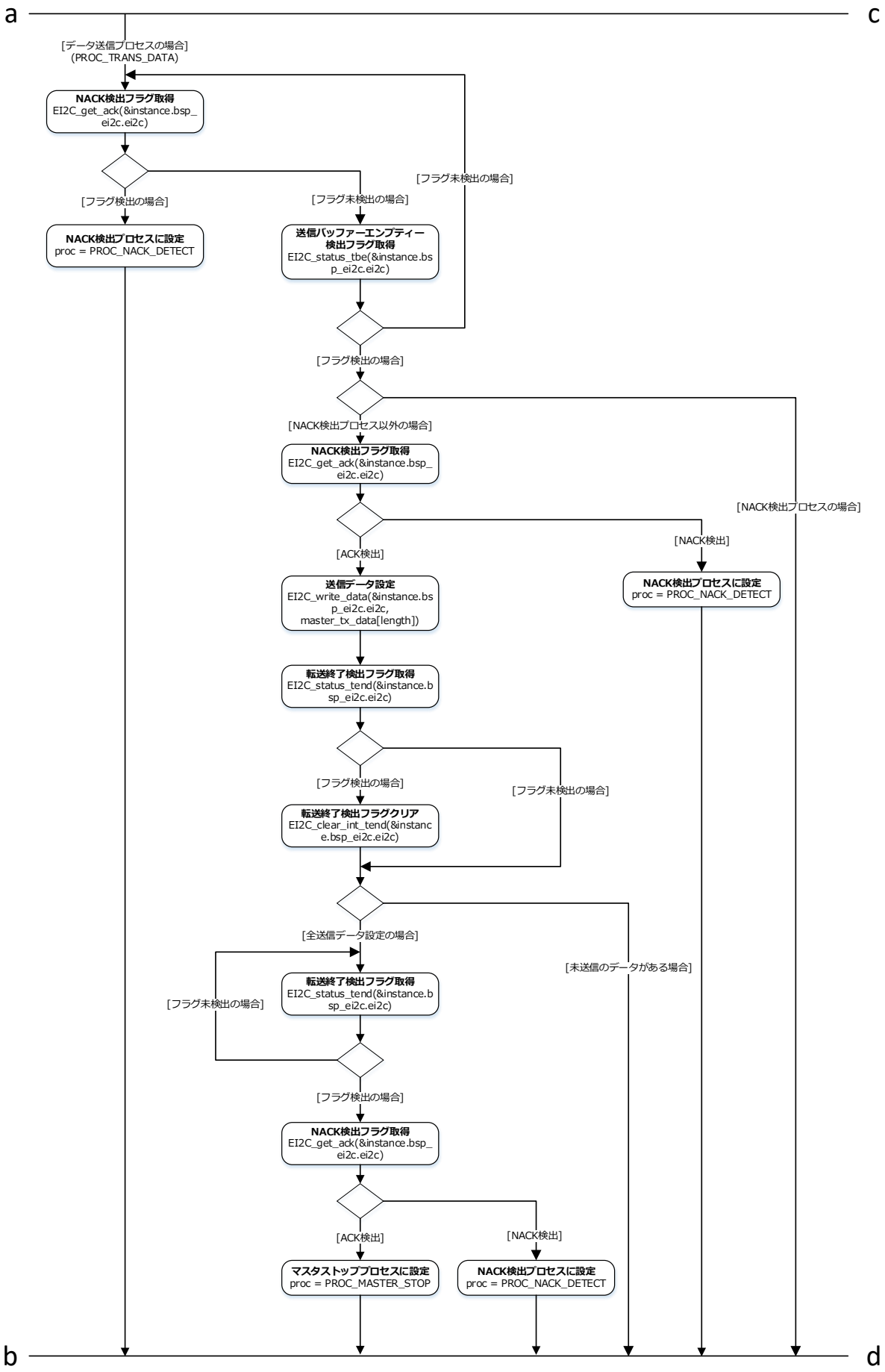
6.5. ターミナルソフト出力例

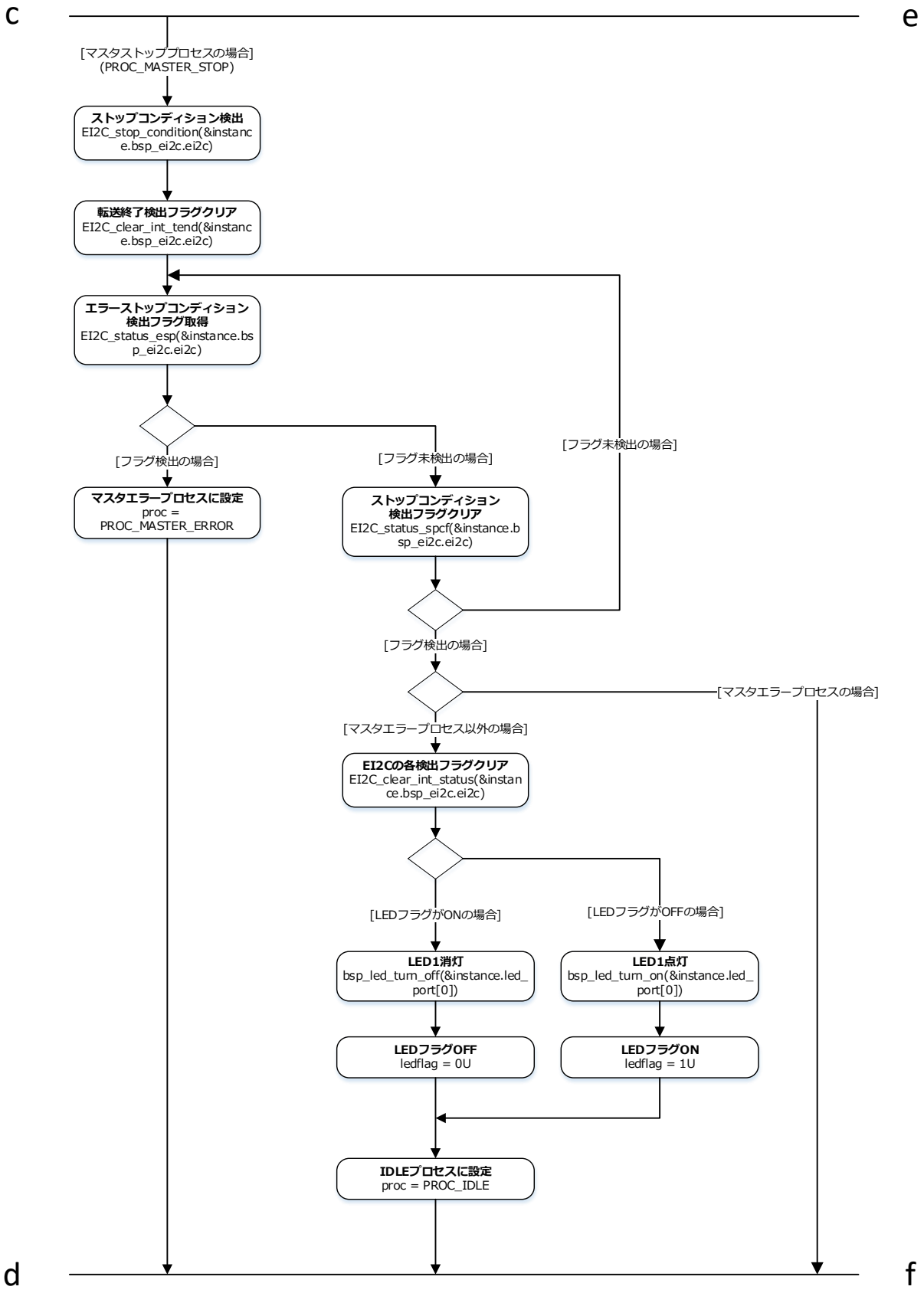
特に無し。

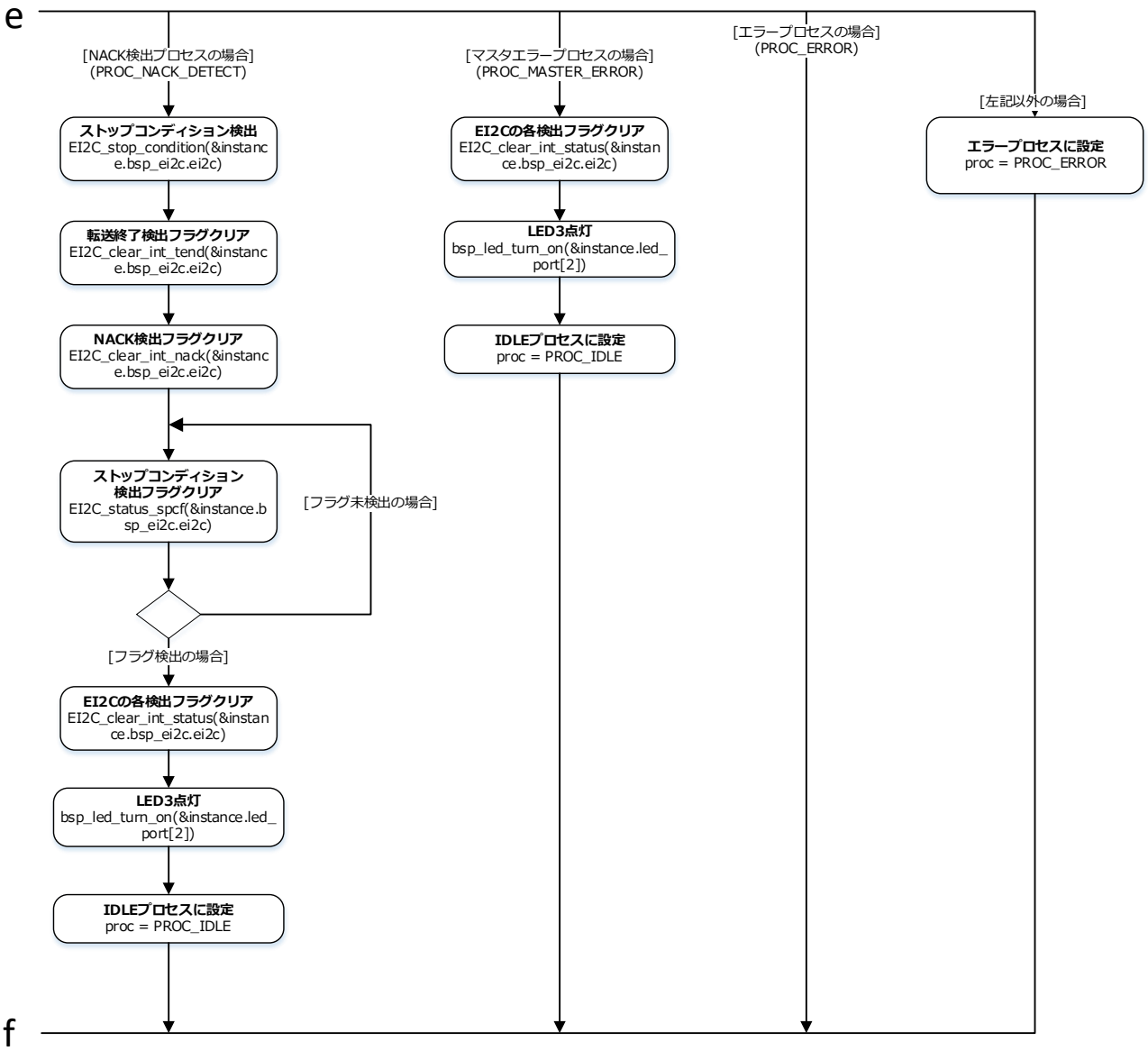
7. アクティビティ図

7.1. main

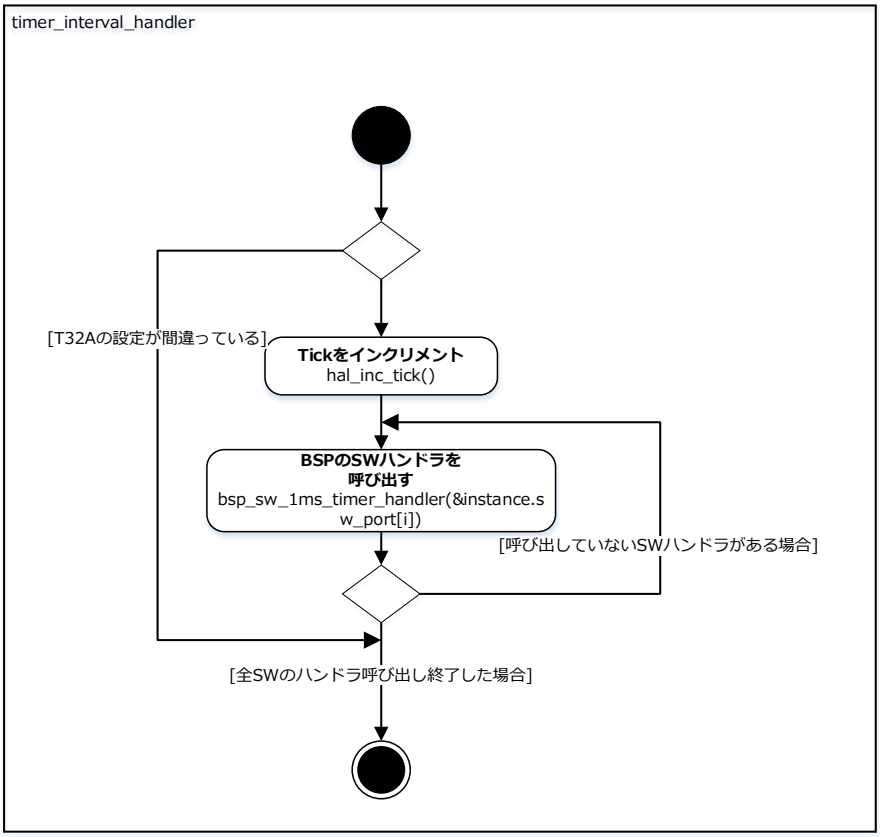


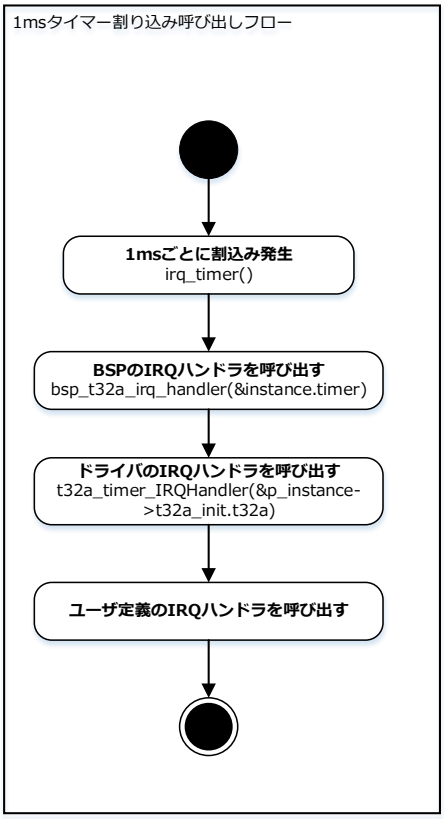

a
c
[データ送信プロセスの場合]
(PROC_TRANS_DATA)
NACK検出フラグ取得
EI2C_get_ack(&instance.bsp_ei2c.ei2c)
[フラグ検出の場合]
[フラグ未検出の場合]
[フラグ未検出の場合]
NACK検出プロセスに設定
proc = PROC_NACK_DETECT
送信バッファーエンプティー検出フラグ取得
EI2C_status_tbe(&instance.bsp_ei2c.ei2c)
[フラグ検出の場合]
[NACK検出プロセス以外の場合]
NACK検出フラグ取得
EI2C_get_ack(&instance.bsp_ei2c.ei2c)
[NACK検出プロセスの場合]
[NACK検出]
[ACK検出]
NACK検出プロセスに設定
proc = PROC_NACK_DETECT
送信データ設定
EI2C_write_data(&instance.bsp_ei2c.ei2c, master_tx_data[length])
転送終了検出フラグ取得
EI2C_status_tend(&instance.bsp_ei2c.ei2c)
[フラグ検出の場合]
[フラグ未検出の場合]
転送終了検出フラグクリア
EI2C_clear_int_tend(&instance.bsp_ei2c.ei2c)
[全送信データ設定の場合]
[未送信のデータがある場合]
[フラグ未検出の場合]
転送終了検出フラグ取得
EI2C_status_tend(&instance.bsp_ei2c.ei2c)
[フラグ検出の場合]
NACK検出フラグ取得
EI2C_get_ack(&instance.bsp_ei2c.ei2c)
[ACK検出]
[NACK検出]
マスタストッププロセスに設定
proc = PROC_MASTER_STOP
NACK検出プロセスに設定
proc = PROC_NACK_DETECT
b
d


c
e
[マスタストッププロセスの場合]
(PROC_MASTER_STOP)
ストップコンディション検出
EI2C_stop_condition(&instance.bsp_ei2c.ei2c)
転送終了検出フラグクリア
EI2C_clear_int_tend(&instance.bsp_ei2c.ei2c)
エラーストップコンディション検出フラグ取得
EI2C_status_esp(&instance.bsp_ei2c.ei2c)
[フラグ検出の場合]
[フラグ未検出の場合]
[フラグ未検出の場合]
マスタエラープロセスに設定
proc = PROC_MASTER_ERROR
ストップコンディション検出フラグクリア
EI2C_status_spcf(&instance.bsp_ei2c.ei2c)
[フラグ検出の場合]
[マスタエラープロセスの場合]
[マスタエラープロセス以外の場合]
EI2Cの各検出フラグクリア
EI2C_clear_int_status(&instance.bsp_ei2c.ei2c)
[LEDフラグがONの場合]
[LEDフラグがOFFの場合]
LED1消灯
bsp_led_turn_off(&instance.led_port[0])
LED1点灯
bsp_led_turn_on(&instance.led_port[0])
LEDフラグOFF
ledflag = 0U
LEDフラグON
ledflag = 1U
IDLEプロセスに設定
proc = PROC_IDLE
d
f


e
[NACK検出プロセスの場合]
(PROC_NACK_DETECT)
ストップコンディション検出
EI2C_stop_condition(&instance.bsp_ei2c.ei2c)
転送終了検出フラグクリア
EI2C_clear_int_tend(&instance.bsp_ei2c.ei2c)
NACK検出フラグクリア
EI2C_clear_int_nack(&instance.bsp_ei2c.ei2c)
ストップコンディション検出フラグクリア
EI2C_status_spcf(&instance.bsp_ei2c.ei2c)
[フラグ未検出の場合]
[フラグ検出の場合]
EI2Cの各検出フラグクリア
EI2C_clear_int_status(&instance.bsp_ei2c.ei2c)
LED3点灯
bsp_led_turn_on(&instance.led_port[2])
IDLEプロセスに設定
proc = PROC_IDLE
[マスタエラープロセスの場合]
(PROC_MASTER_ERROR)
EI2Cの各検出フラグクリア
EI2C_clear_int_status(&instance.bsp_ei2c.ei2c)
LED3点灯
bsp_led_turn_on(&instance.led_port[2])
IDLEプロセスに設定
proc = PROC_IDLE
[エラープロセスの場合]
(PROC_ERROR)
[左記以外の場合]
エラープロセスに設定
proc = PROC_ERROR
f

7.2. 割り込み


timer_interval_handler
[T32Aの設定が間違っている]
Tickをインクリメント
hal_inc_tick()
BSPのSWハンドラを
呼び出す
bsp_sw_1ms_timer_handler(&instance.sw_port[i])
[呼び出していないSWハンドラがある場合]
[全SWのハンドラ呼び出し終了した場合]


1msタイマー割り込み呼び出しフロー
1msごとに割込み発生
irq_timer()
BSPのIRQハンドラを呼び出す
bsp_t32a_irq_handler(&instance.timer)
ドライバのIRQハンドラを呼び出す
t32a_timer_IRQHandler(&p_instance->t32a_init.t32a)
ユーザ定義のIRQハンドラを呼び出す

8. 改訂履歴

Revision	日付	変更項目
1.0	2025-01-20	初版
1.1	2025-10-30	6.3 使用する割り込み M3H 使用割り込み名追記
1.2	2026-07-30	6.3 使用する割り込み 詳細は利用説明書に転載

© 2026
Toshiba Electronic Devices & Storage Corporation

製品取り扱い上のお願い

株式会社東芝およびその子会社ならびに関係会社を以下「当社」といいます。
本資料に掲載されているハードウエア、ソフトウエアおよびシステムを以下「本製品」といいます。

- 本製品に関する情報等、本資料の掲載内容は、技術の進歩などにより予告なしに変更されることがあります。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。また、文書による当社の事前の承諾を得て本資料を転載複製する場合でも、記載内容に一切変更を加えたり、削除したりしないでください。
- 当社は品質、信頼性の向上に努めていますが、半導体・ストレージ製品は一般に誤作動または故障する場合があります。本製品をご使用頂く場合は、本製品の誤作動や故障により生命・身体・財産が侵害されることのないように、お客様の責任において、お客様のハードウエア・ソフトウエア・システムに必要な安全設計を行うことをお願いします。なお、設計および使用に際しては、本製品に関する最新の情報（本資料、仕様書、データシート、アプリケーションノート、半導体信頼性ハンドブックなど）および本製品が使用される機器の取扱説明書、操作説明書などをご確認の上、これに従ってください。また、上記資料などに記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を使用する場合は、お客様の製品単独およびシステム全体で十分に評価し、お客様の責任において適用可否を判断してください。
- 本製品は、特別に高い品質・信頼性が要求され、またはその故障や誤作動が生命・身体に危害を及ぼす恐れ、膨大な財産損害を引き起こす恐れ、もしくは社会に深刻な影響を及ぼす恐れのある機器（以下“特定用途”という）に使用されることは意図されていませんし、保証もされていません。特定用途には原子力関連機器、航空・宇宙機器、医療機器（生命直結機器）、車載・輸送機器、防衛関連機器などが含まれますが、本資料に個別に記載する用途は除きます。特定用途に使用された場合には、当社は一切の責任を負いません。なお、詳細は当社営業窓口まで、または当社 Web サイトのお問い合わせフォームからお問い合わせください。
- 本製品を、国内外の法令、規則及び命令により、製造、使用、販売を禁止されている製品に使用することはできません。
- 本資料に掲載してある技術情報は、製品の代表的動作・応用を説明するためのもので、その使用に際して当社及び第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。
- 別途、書面による契約またはお客様と当社が合意した仕様書がない限り、当社は、本製品および技術情報に関して、明示的にも黙示的にも一切の保証（機能動作の保証、商品性の保証、特定目的への合致の保証、情報の正確性の保証、第三者の権利の非侵害保証を含むがこれに限らない。）をしておりません。
- 本製品、または本資料に掲載されている技術情報を、大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」、「米国輸出管理規則」等、適用ある輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
- 本製品の RoHS 適合性など、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い合わせください。本製品のご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用ある環境関連法令を十分調査の上、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、当社は一切の責任を負いかねます。

東芝デバイス&ストレージ株式会社
https://toshiba.semicon-storage.com/jp/

© 2026
Toshiba Electronic Devices & Storage Corporation