Application Note

TSPI_TARGET_RECEIVE

Arm and Keil are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Table of Contents

Table of Contents 2
1. Preface 3
2. Technical Term 3
3. Reference Document 3
4. Target Sample Program 4
5. Configuration Diagram 4
6. Sample Program: TSPI_TARGET_RECEIVE 5
6.1. Outlines of Operation 5
6.2. Function to Use 5
6.3. Interrupt to Use 5
6.4. Configuration 6
6.5. Example of Terminal Emulator Output 6
6.6. DMAC Communication Setting 6
7. Activity diagram 7
7.1. main 7
7.2. Interrupt 11
8. Revision History 13
RESTRICTIONS ON PRODUCT USE 14

© 2026
Toshiba Electronic Devices & Storage Corporation

1. Preface

This application note describes the sample software TSPI_TARGET_RECEIVE using Serial Peripheral Interface (TSPI).
This document helps the user check operation of a product under development and develop its program.

2. Technical Term

Term/Abbreviation	Definition
BSP	Board Support Package
CG	Clock Control and Operation Mode
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access Controller
Timer	T32A:32-bit Timer Event Counter
TSPI	Serial Peripheral Interface

3. Reference Document

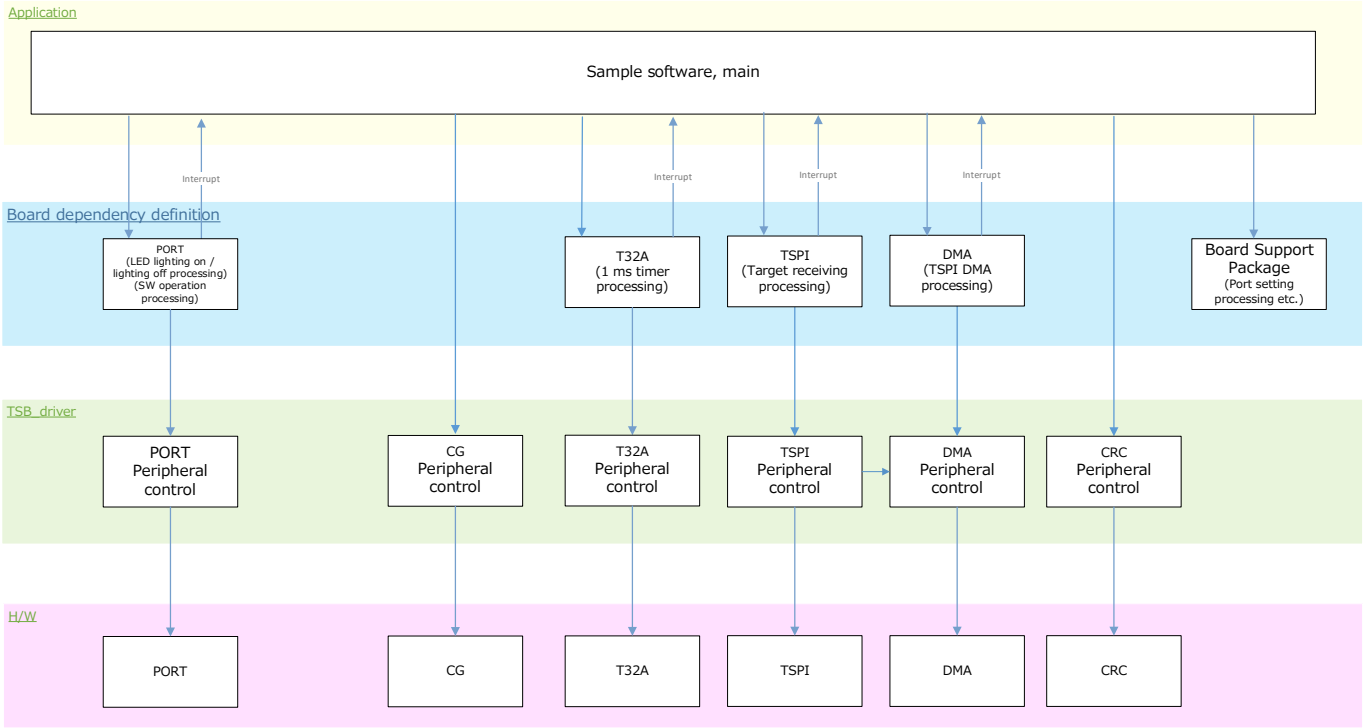
Document	Notes
Data sheet	Refer to the data sheet of MCU to be used.
Reference manual	Refer to the reference manual of each IP to be used.
Application note MCU User Guide	Refer to the MCU user guide to be used.

© 2026
Toshiba Electronic Devices & Storage Corporation

4. Target Sample Program

Sample Program	Outline
TSPI_TARGET_RECEIVE	Sample program of SPI function (Target Receive)

5. Configuration Diagram



6. Sample Program: TSPI_TARGET_RECEIVE

This sample software uses the Target receive processing function of the SPI communication function to enter the reception waiting state when the switch is pressed, and switches the LED turn on / turn off each time data receive is completed.

6.1. Outlines of Operation

Turns off BSP_LED_2, BSP_LED_3, and BSP_LED_4.
When BSP_PSW_1 is pressed, BSP_LED_3 and BSP_LED_4 are turned off and data for the data size is received. The data that exceeds the data size will be discarded.
Switches the lighting status (turn on / turn off) of BSP_LED_2.
When SPI read error occurs, BSP_LED_3 turns on.
When the CRC value of the received data and the calculated CRC value do not match, BSP_LED_4 is turn on.

6.2. Function to Use

The functions to use are as follows:
For the Port assignment of each BSP channel, refer to the MCU user guide.

IP	Channel	Objective
TSPI	BSP_TSPI_1	SPI communication
T32A	BSP_T32A_TIMER_1	Interval timer
PORT (Push-Switch)	BSP_PSW_1	Event trigger
PORT (LED)	BSP_LED_2	For operation check
	BSP_LED_3	For operation check
	BSP_LED_4	For operation check

6.3. Interrupt to Use

For details, please refer to the “Interrupt List by Sample” in the MCU User’s Manual.

© 2026
Toshiba Electronic Devices & Storage Corporation

6.4. Configuration

Configuration setting.

Configuration	Soft Definition Name	Current Value (Defaults)	Description
Data size	BSP_DATA_LENGTH	12	Data size (unit: byte) Compilation switch (Note1): Be switched by the BSP_CHK_CODE setting.
Check CODE	BSP_CHK_CODE	BSP_CHK_CODE_CRC16	BSP_CHK_CODE_CRC16 and BSP_CHK_CODE_CRC32 can be switched
Communication Control	BSP_REQUEST_MODE	BSP_FIFO_MODE	BSP_FIFO_MODE and BSP_DMA_MODE can be switched.
Send Fill Level Setting	BSP_RX_FILL_LEVEL	4	Set to RIL in TSPIxCR2

Note1: When BSP_CHK_CODE_CRC16, 12byte.
When BSP_CHK_CODE_CRC32, 14byte.

6.5. Example of Terminal Emulator Output

Nothing.

6.6. DMAC Communication Setting

Startup Factors and Transfer Mode

Configuration	Description
Startup Factors	DMA Request from TSPI Driver (Request Based on FIFO level)
Transfer Mode	Singel Transfer/Unit Normal Transfer

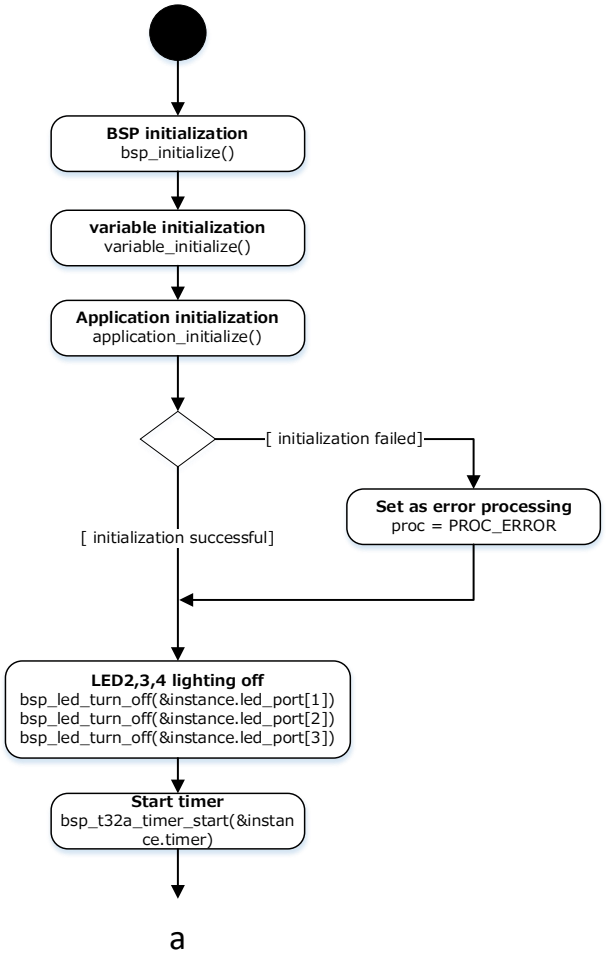
Channel Control Data: Transfer Mode Setting [DMAChnlCfg]

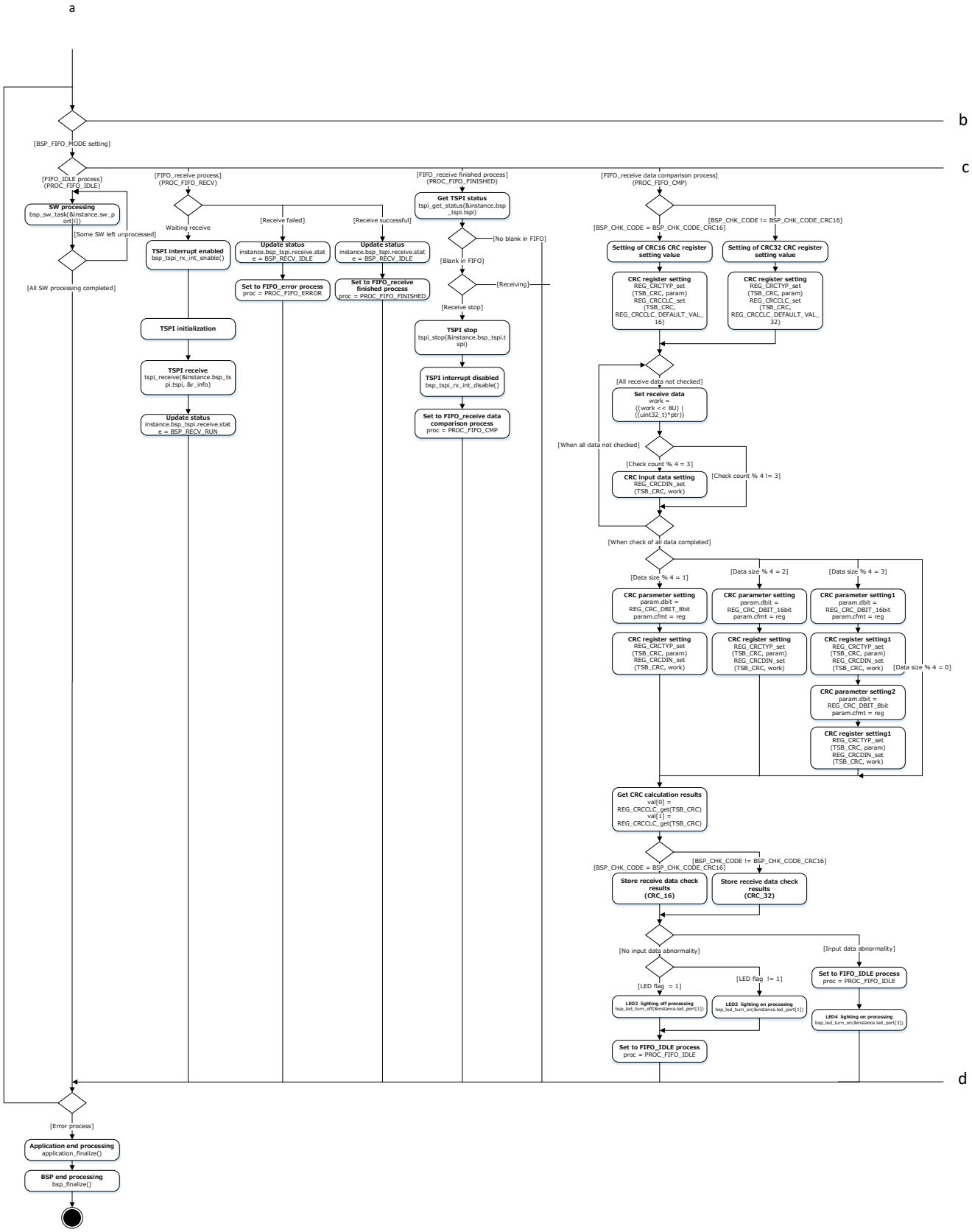
Configuration	Description
Destination Address Increment	00: [1byte]
Destination Data Size	00: [1byte]
Source Address Increment	11: [No increment]
Source Data Size	00: [1byte]
Arbitration	0000: [After 1 Rotation Transfer]
Transfer Count	0x00B: [12times (BSP_DATA_LENGTH)]
Singel Transfer Setting	0: [Not used because it is not a chain transfer]
Transfer Mode	001: [Unit Normal Transfer]

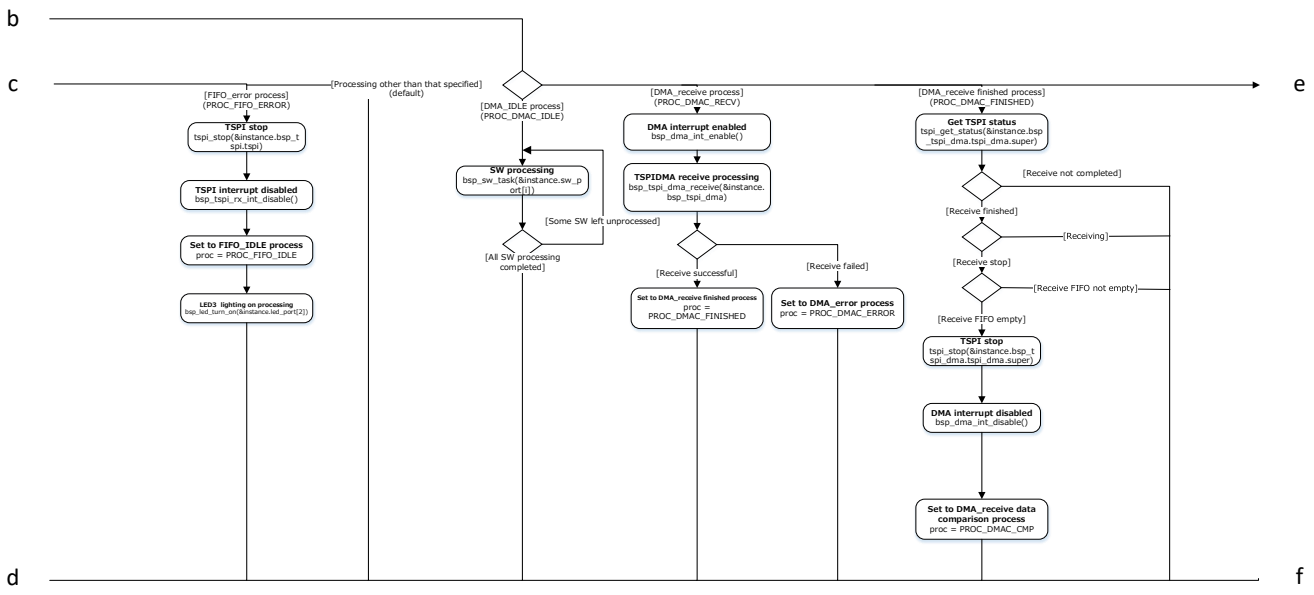
© 2026
Toshiba Electronic Devices & Storage Corporation

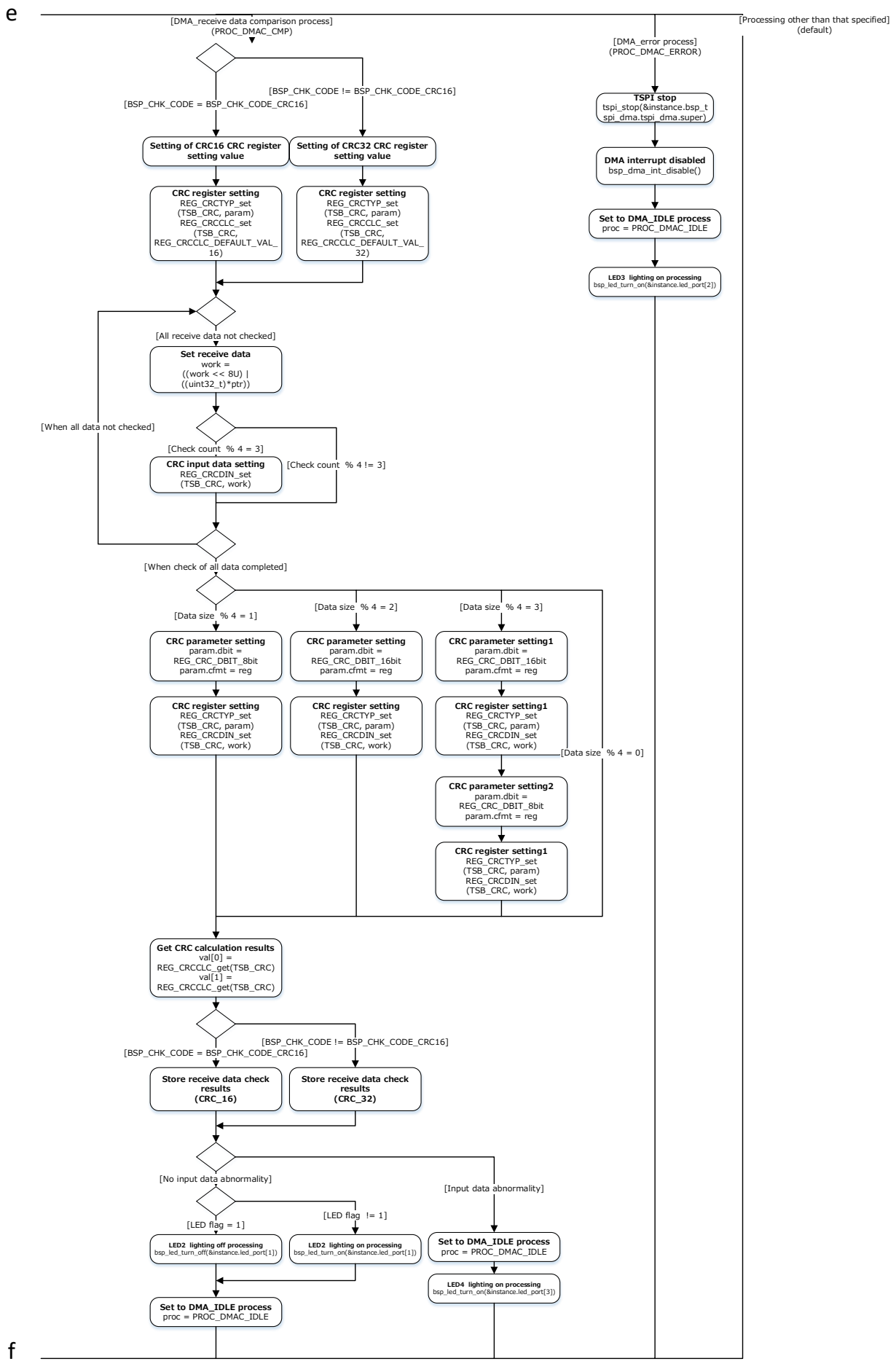
7. Activity diagram

7.1. main

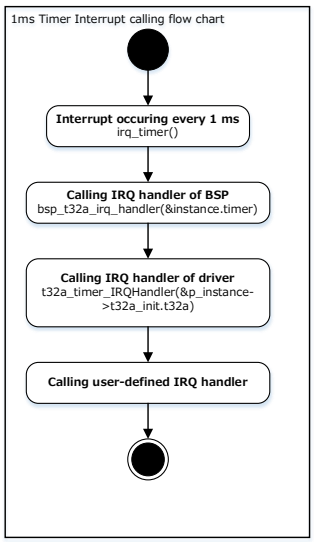


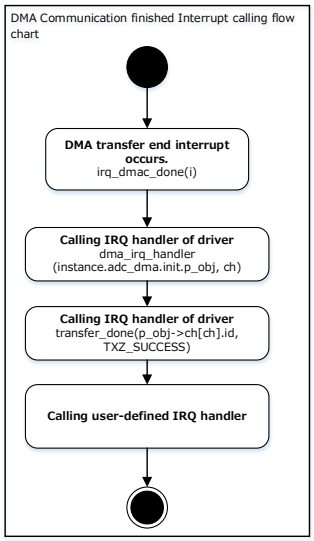

a
b
c
d
[BSP_FIFO_MODE setting]
[FIFO_IDLE process]
(PROC_FIFO_IDLE)
SW processing
bsp_sw_task(&instance.sw_port[i])
[Some SW left unprocessed]
[All SW processing completed]
[FIFO_receive process]
(PROC_FIFO_RECV)
Waiting receive
[Receive failed]
[Receive successful]
TSPI interrupt enabled
bsp_tspi_rx_int_enable()
Update status
instance.bsp_tspi.receive.state = BSP_RECV_IDLE
Update status
instance.bsp_tspi.receive.state = BSP_RECV_IDLE
Set to FIFO_error process
proc = PROC_FIFO_ERROR
Set to FIFO_receive finished process
proc = PROC_FIFO_FINISHED
TSPI initialization
TSPI receive
tspi_receive(&instance.bsp_tspi.tspi, &r_info)
Update status
instance.bsp_tspi.receive.state = BSP_RECV_RUN
[FIFO_receive finished process]
(PROC_FIFO_FINISHED)
Get TSPI status
tspi_get_status(&instance.bsp_tspi.tspi)
[No blank in FIFO]
[Blank in FIFO]
[Receiving]
[Receive stop]
TSPI stop
tspi_stop(&instance.bsp_tspi.tspi)
TSPI interrupt disabled
bsp_tspi_rx_int_disable()
Set to FIFO_receive data comparison process
proc = PROC_FIFO_CMP
[FIFO_receive data comparison process]
(PROC_FIFO_CMP)
[BSP_CHK_CODE = BSP_CHK_CODE_CRC16]
[BSP_CHK_CODE != BSP_CHK_CODE_CRC16]
Setting of CRC16 CRC register setting value
Setting of CRC32 CRC register setting value
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCCLC_set
(TSB_CRC,
REG_CRCCLC_DEFAULT_VAL_16)
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCCLC_set
(TSB_CRC,
REG_CRCCLC_DEFAULT_VAL_32)
[All receive data not checked]
Set receive data
work =
((work << 8U) |
((uint32_t)*ptr))
[When all data not checked]
[Check count % 4 = 3]
[Check count % 4 != 3]
CRC input data setting
REG_CRCDIN_set
(TSB_CRC, work)
[When check of all data completed]
[Data size % 4 = 1]
[Data size % 4 = 2]
[Data size % 4 = 3]
[Data size % 4 = 0]
CRC parameter setting
param.dbit =
REG_CRC_DBIT_8bit
param.cfmt = reg
CRC parameter setting
param.dbit =
REG_CRC_DBIT_16bit
param.cfmt = reg
CRC parameter setting1
param.dbit =
REG_CRC_DBIT_16bit
param.cfmt = reg
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
CRC register setting1
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
CRC parameter setting2
param.dbit =
REG_CRC_DBIT_8bit
param.cfmt = reg
CRC register setting1
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
Get CRC calculation results
val[0] =
REG_CRCCLC_get(TSB_CRC)
val[1] =
REG_CRCCLC_get(TSB_CRC)
[BSP_CHK_CODE = BSP_CHK_CODE_CRC16]
[BSP_CHK_CODE != BSP_CHK_CODE_CRC16]
Store receive data check results
(CRC_16)
Store receive data check results
(CRC_32)
[No input data abnormality]
[Input data abnormality]
[LED flag = 1]
[LED flag != 1]
LED2 lighting off processing
bsp_led_turn_off(&instance.led_port[1])
LED2 lighting on processing
bsp_led_turn_on(&instance.led_port[1])
Set to FIFO_IDLE process
proc = PROC_FIFO_IDLE
LED4 lighting on processing
bsp_led_turn_on(&instance.led_port[3])
Set to FIFO_IDLE process
proc = PROC_FIFO_IDLE
[Error process]
Application end processing
application_finalize()
BSP end processing
bsp_finalize()

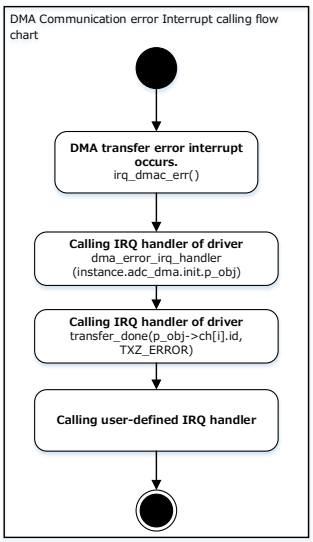

b
c
e
d
f
[Processing other than that specified]
(default)
[FIFO_error process]
(PROC_FIFO_ERROR)
TSPI stop
tspi_stop(&instance.bsp_tspi.tspi)
TSPI interrupt disabled
bsp_tspi_rx_int_disable()
Set to FIFO_IDLE process
proc = PROC_FIFO_IDLE
LED3 lighting on processing
bsp_led_turn_on(&instance.led_port[2])
[DMA_IDLE process]
(PROC_DMAC_IDLE)
SW processing
bsp_sw_task(&instance.sw_port[1])
[Some SW left unprocessed]
[All SW processing completed]
[DMA_receive process]
(PROC_DMAC_RECV)
DMA interrupt enabled
bsp_dma_int_enable()
TSPIDMA receive processing
bsp_tspi_dma_receive(&instance.bsp_tspi_dma)
[Receive successful]
[Receive failed]
Set to DMA_receive finished process
proc = PROC_DMAC_FINISHED
Set to DMA_error process
proc = PROC_DMAC_ERROR
[DMA_receive finished process]
(PROC_DMAC_FINISHED)
Get TSPI status
tspi_get_status(&instance.bsp_tspi_dma.tspi_dma.super)
[Receive not completed]
[Receive finished]
[Receiving]
[Receive stop]
[Receive FIFO not empty]
[Receive FIFO empty]
TSPI stop
tspi_stop(&instance.bsp_tspi_dma.tspi_dma.super)
DMA interrupt disabled
bsp_dma_int_disable()
Set to DMA_receive data comparison process
proc = PROC_DMAC_CMP

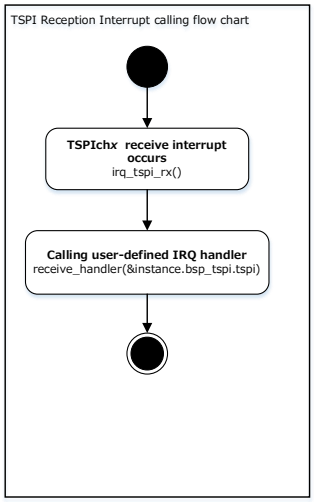

e
[DMA_receive data comparison process]
(PROC_DMAC_CMP)
[BSP_CHK_CODE = BSP_CHK_CODE_CRC16]
[BSP_CHK_CODE != BSP_CHK_CODE_CRC16]
Setting of CRC16 CRC register setting value
Setting of CRC32 CRC register setting value
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCCLC_set
(TSB_CRC,
REG_CRCCLC_DEFAULT_VAL_16)
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCCLC_set
(TSB_CRC,
REG_CRCCLC_DEFAULT_VAL_32)
[All receive data not checked]
Set receive data
work =
((work << 8U) |
((uint32_t)*ptr))
[When all data not checked]
[Check count % 4 = 3]
CRC input data setting
REG_CRCDIN_set
(TSB_CRC, work)
[Check count % 4 != 3]
[When check of all data completed]
[Data size % 4 = 1]
[Data size % 4 = 2]
[Data size % 4 = 3]
CRC parameter setting
param.dbit =
REG_CRC_DBIT_8bit
param.cfmt = reg
CRC parameter setting
param.dbit =
REG_CRC_DBIT_16bit
param.cfmt = reg
CRC parameter setting1
param.dbit =
REG_CRC_DBIT_16bit
param.cfmt = reg
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
CRC register setting
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
CRC register setting1
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
[Data size % 4 = 0]
CRC parameter setting2
param.dbit =
REG_CRC_DBIT_8bit
param.cfmt = reg
CRC register setting1
REG_CRCTYP_set
(TSB_CRC, param)
REG_CRCDIN_set
(TSB_CRC, work)
Get CRC calculation results
val[0] =
REG_CRCCLC_get(TSB_CRC)
val[1] =
REG_CRCCLC_get(TSB_CRC)
[BSP_CHK_CODE = BSP_CHK_CODE_CRC16]
[BSP_CHK_CODE != BSP_CHK_CODE_CRC16]
Store receive data check results
(CRC_16)
Store receive data check results
(CRC_32)
[No input data abnormality]
[Input data abnormality]
[LED flag = 1]
[LED flag != 1]
LED2 lighting off processing
bsp_led_turn_off(&instance.led_port[1])
LED2 lighting on processing
bsp_led_turn_on(&instance.led_port[1])
Set to DMA_IDLE process
proc = PROC_DMAC_IDLE
LED4 lighting on processing
bsp_led_turn_on(&instance.led_port[3])
Set to DMA_IDLE process
proc = PROC_DMAC_IDLE
f
[DMA_error process]
(PROC_DMAC_ERROR)
TSPI stop
tspi_stop(&instance.bsp_tspi_dma.tspi_dma.super)
DMA interrupt disabled
bsp_dma_int_disable()
Set to DMA_IDLE process
proc = PROC_DMAC_IDLE
LED3 lighting on processing
bsp_led_turn_on(&instance.led_port[2])
[Processing other than that specified]
(default)

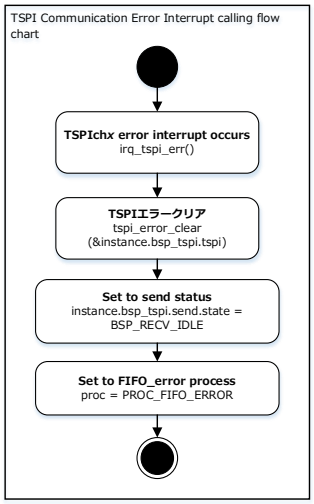
7.2. Interrupt

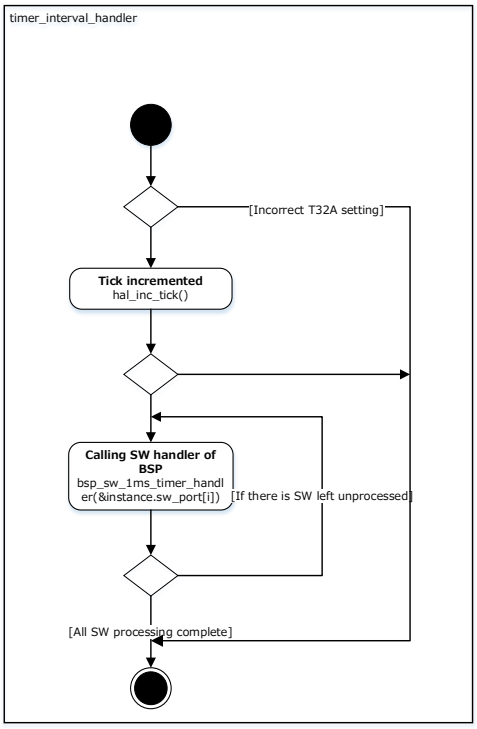


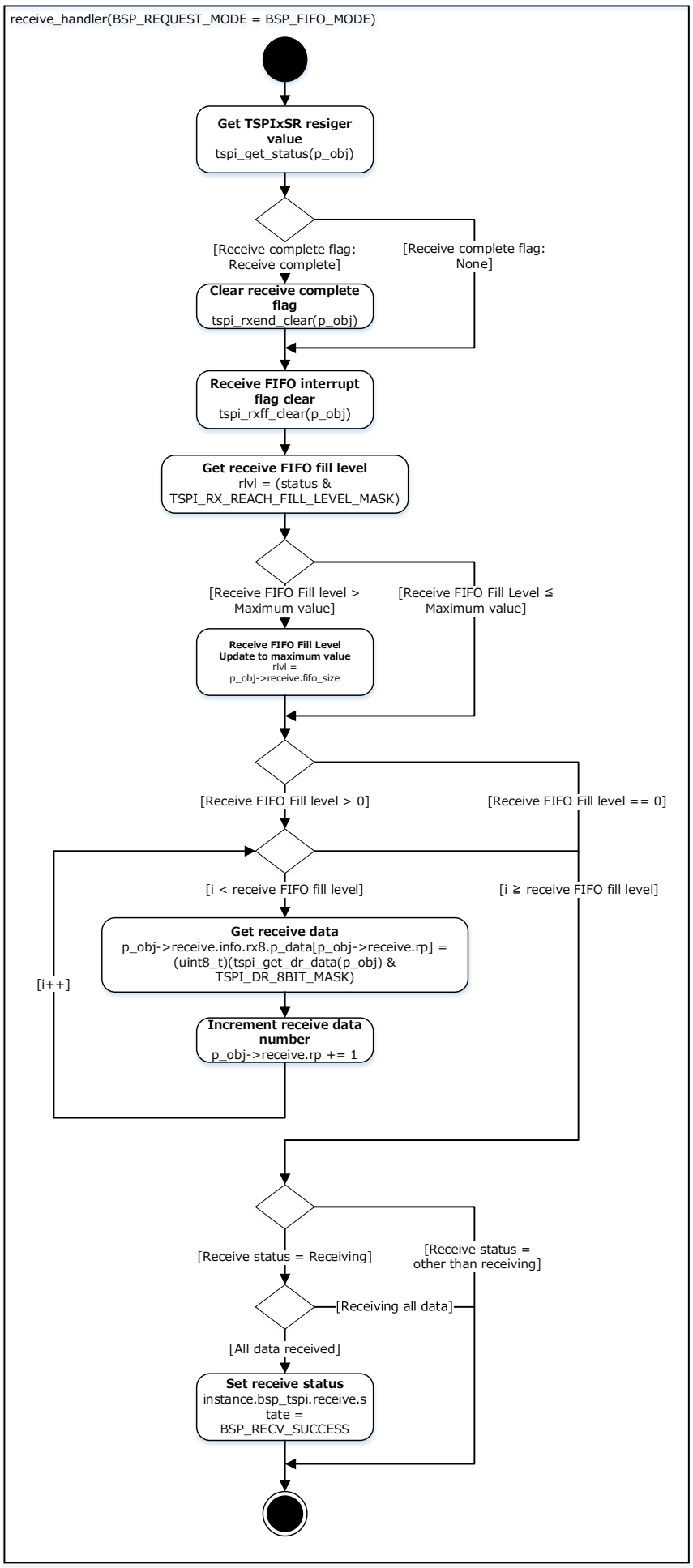











receive_handler(BSP_REQUEST_MODE = BSP_FIFO_MODE)
Get TSPIxSR resiger value
tspi_get_status(p_obj)
[Receive complete flag: Receive complete]
[Receive complete flag: None]
Clear receive complete flag
tspi_rxend_clear(p_obj)
Receive FIFO interrupt flag clear
tspi_rxff_clear(p_obj)
Get receive FIFO fill level
rlvl = (status & TSPI_RX_REACH_FILL_LEVEL_MASK)
[Receive FIFO Fill level > Maximum value]
[Receive FIFO Fill Level ≦ Maximum value]
Receive FIFO Fill Level Update to maximum value
rlvl = p_obj->receive.fifo_size
[Receive FIFO Fill level > 0]
[Receive FIFO Fill level == 0]
[i < receive FIFO fill level]
[i ≧ receive FIFO fill level]
Get receive data
p_obj->receive.info.rx8.p_data[p_obj->receive.rp] = (uint8_t)(tspi_get_dr_data(p_obj) & TSPI_DR_8BIT_MASK)
[i++]
Increment receive data number
p_obj->receive.rp += 1
[Receive status = Receiving]
[Receive status = other than receiving]
[Receiving all data]
[All data received]
Set receive status
instance.bsp_tspi.receive.state = BSP_RECV_SUCCESS

8. Revision History

Revision	Date	Description
1.0	2025-01-20	First release
1.1	2025-10-30	6.3 Interrupt to Use Added M3H Interrupt Name.
1.2	2026-07-30	6.3 Interrupt to Use Detailed information is transcribed into the User's Manual
1.3	2026-09-30	7.2 Interrupts Removal of MCU-specific descriptions

© 2026
Toshiba Electronic Devices & Storage Corporation

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, Class 3 medical devices, equipment used for automobiles, and military vehicles and munitions. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

Toshiba Electronic Devices & Storage Corporation
https://toshiba.semicon-storage.com/

© 2026
Toshiba Electronic Devices & Storage Corporation