M4G Group (1)
Application Note
Flash Memory
(FLASH15MHD32-A)

Outlines

This application note is a reference material for developing products using the Flash memory (FLASH) function of M4G group (1).
This document helps the user check operation of the product and develop its program.

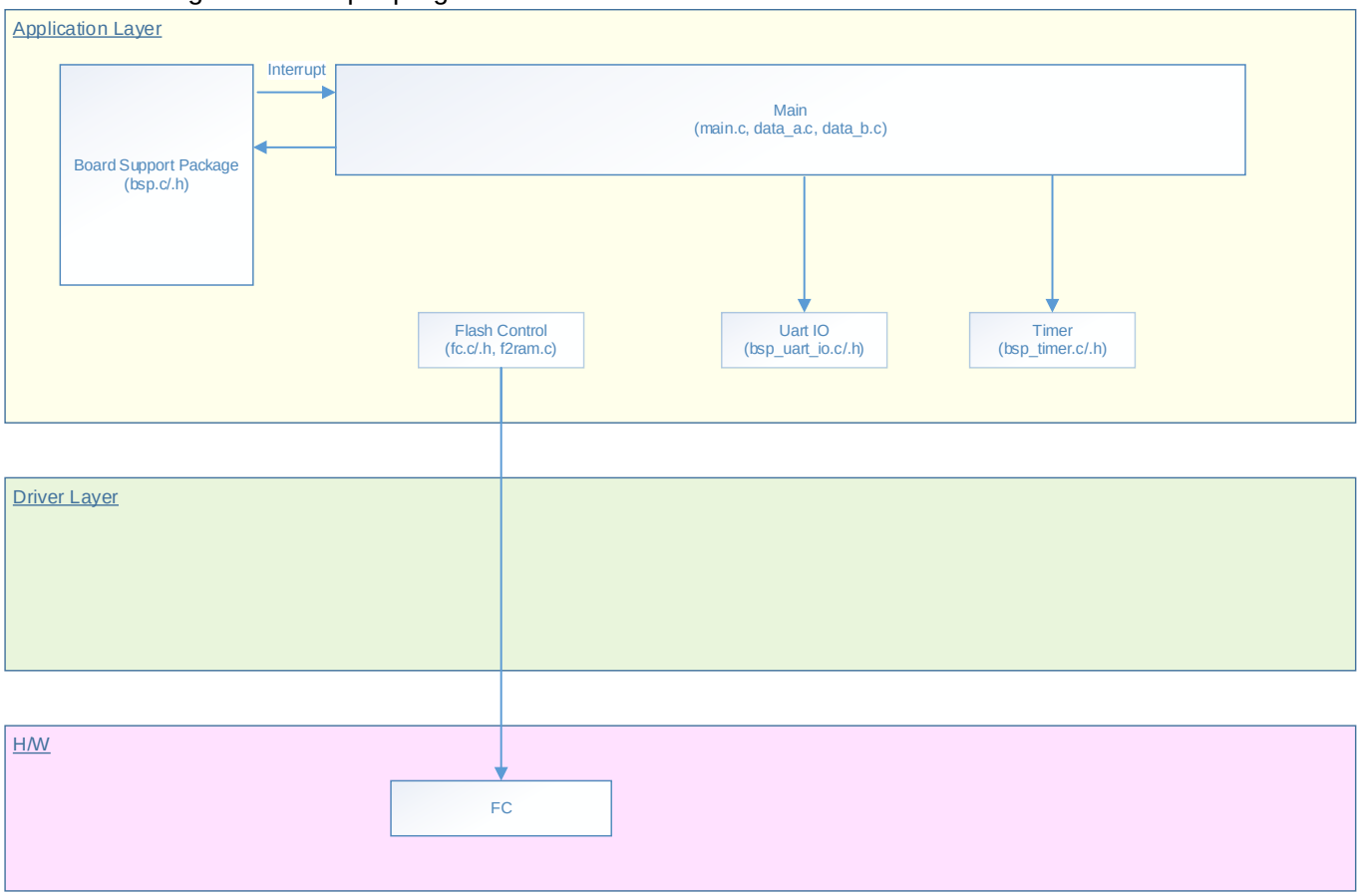
Target sample program: Flash_Userboot

Table of Contents

1. Preface

This sample program lights on and off the LED's using two data stored in the Code Flash.
The data in the Code Flash are replaced each other by a push-down of the push switch.

Structure diagram of Sample program



2. Reference Document

- Datasheet
  TMPM4G group (1) datasheet Rev1.0 (Japanese edition)
- Reference manual
  Flash Memory (FLASH15MHD32-A) Rev1.0 (Japanese edition)
  Asynchronous Serial Communication Circuit (UART-C) Rev3.0 (Japanese edition)
  Input/Output Ports (PORT-M4G (1)) Rev1.0 (Japanese edition)
- Application note
  M4G Group (1) Application Note Startup (CMSIS System & Clock Configuration) Rev1.0
- Other reference document
  TMPM4G (1) Group Peripheral Driver User Manual (Doxygen)

3. Function to Use

IP	Channel	Port	Function/Operation mode
Asynchronous Serial Communication Circuit	ch0	PE2 (UT0RXD) PE3 (UT0TXDA)	UART mode
Flash Memory	-	-	Code Flash
Input/Output Ports	-	PE4 (Output port)	Output
	-	PE5 (Output port)	
	-	PE6 (Output port)	
	-	PE7 (Output port)	
	-	PV1 (Input port)	Input

4. Target Device

The target devices of this application note are as follows;

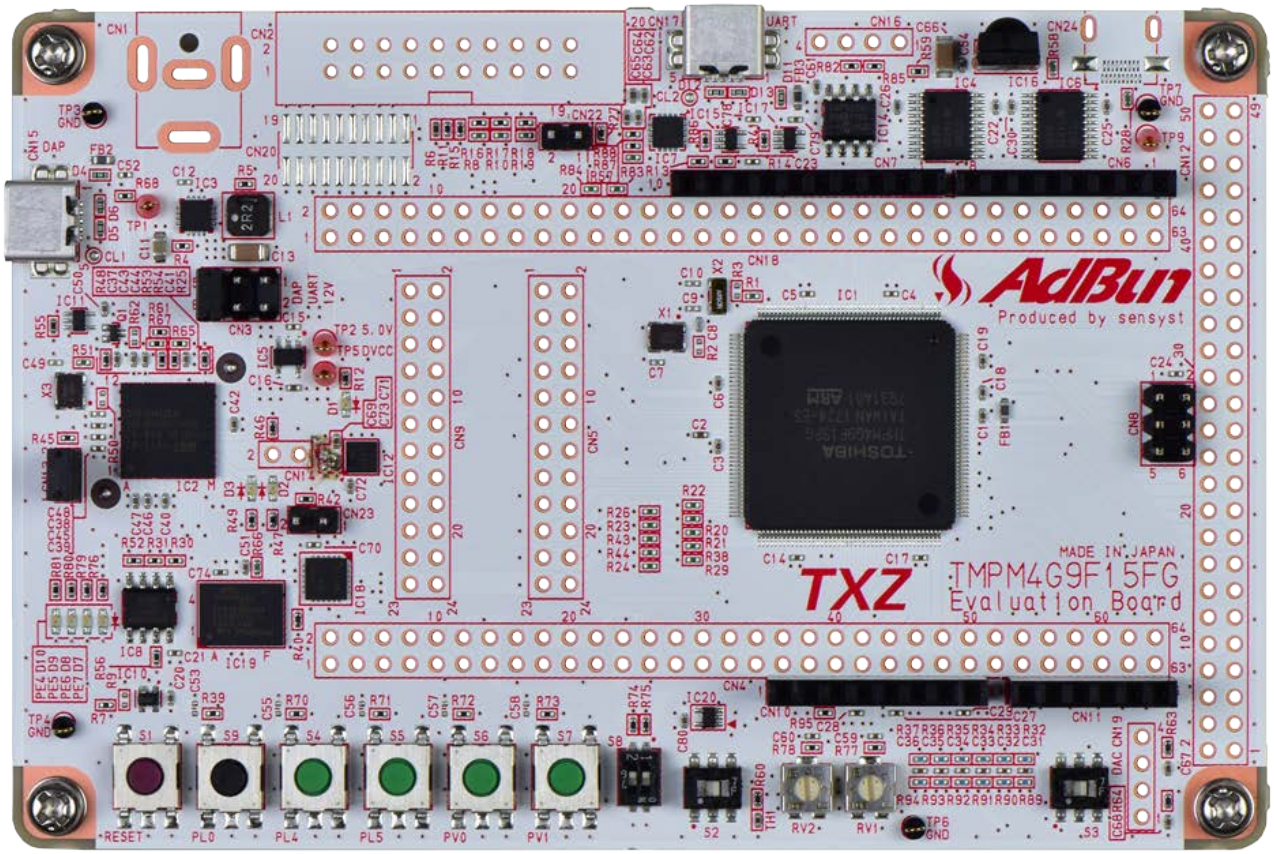
TMPM4G9F15FG	TMPM4G9F10FG	TMPM4G9FEFG	TMPM4G9FDFG
TMPM4G9F15XBG	TMPM4G9F10XBG	TMPM4G9FEXBG	TMPM4G9FDXBG
TMPM4G8F15FG	TMPM4G8F10FG	TMPM4G8FEFG	TMPM4G8FDFG
TMPM4G8F15XBG	TMPM4G8F10XBG	TMPM4G8FEXBG	TMPM4G8FDXBG
	TMPM4G7F10FG	TMPM4G7FEFG	TMPM4G7FDFG
	TMPM4G6F10FG	TMPM4G6FEFG	TMPM4G6FDFG

* This sample program operates on the evaluation board of TMPM4G9F15FG.
If other function than the TMPM4G9F15 one is checked, it is necessary that CMSIS Core related files (C startup file and I/O header file) should be changed properly.
The BSP related file is dedicated to the evaluation board (TMPM4G9F15). If other function than the TMPM4G9F15 one is checked, the BSP related file should be changed properly.

5. Operation Confirmation Condition

Used microcontroller	TMPM4G9F15FG
Used board	TMPM4G9F15FG Evaluation Board by Sensyst
Unified development environment	IAR Embedded Workbench for ARM 8.11.2.13606
Unified development environment	µVision MDK Version 5.24.2.0
Terminal software	Tera Term V4.96
Sample program	V1000

Evaluation board (TMPM4G9F15FG Evaluation Board) Top view



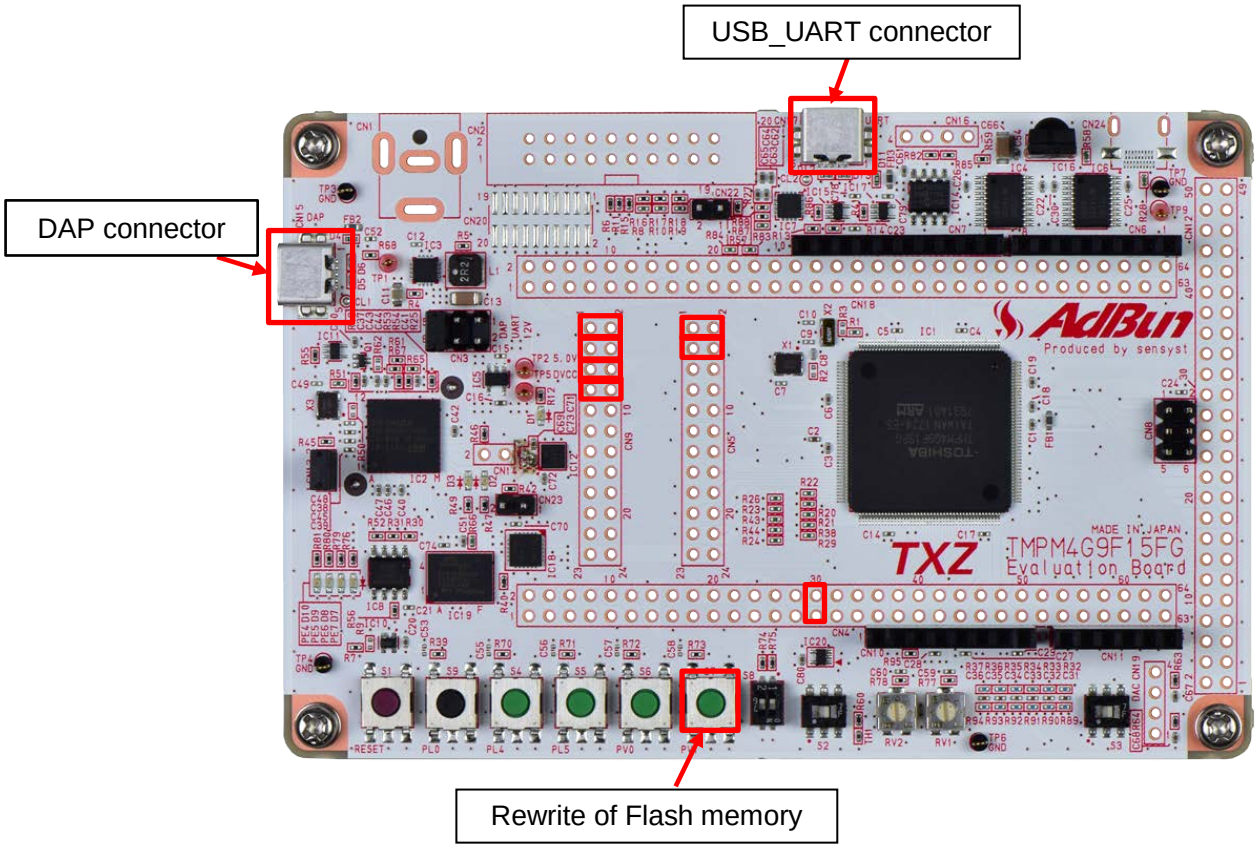
6. Evaluation Board Setting

The following through-holes should be connected.

CN5		
Board function	Through-hole No.	Through-hole No.
USB_UART conversion	1: USB_UT_RX	2: PE2
USB_UART conversion	3: USB_UT_TX	4: PE3

CN9		
Board function	Through-hole No.	Through-hold No.
LED (D10)	1: LED0	2: PE4
LED (D9)	3: LED1	4: PE5
LED (D8)	5: LED2	6: PE6
LED (D7)	7: LED3	8: PE7

CN4		
Board function	Through-hole No.	Through-hole No.
Push switch (S7)	29: ISD_SW1	30: PV1



7. Operation of Evaluation Board

PC and the USB_UART are connected for communication with the terminal software.
This sample program includes LED data “A” (lighting control for LED0 and LED2) and LED data “B” (lighting control for LED1 and LED3).
After the sample program starts up, the data table is read (the initial data is “A”), and the corresponding LED’s light or light off.
When the push switch S7 is pushed down, the data “A” and “B” are copied to the RAM and the corresponding area of the Flash memory is erased. Then, the data “A” is programmed to the area where the data “B” used to be stored in the Flash memory, and the data “B” is programed to the area where the data “A” used to be stored.
After this procedure completes, the data table is read again. The corresponding LED’s light or lights off according to the read data.

8. Outline of Flash memory Function

The code flash which stores a program code, and the data flash which stores data are explained.
A code flash stores an instruction code, and CPU reads and executes it.

There is user information area which can be accessed in a code flash by bank change. Since user information area is not erased by a chip erasing command, for example, it can be written a unique management number etc. for every chip.

A data flash stores data, and even if power supply is intercepted, it keeps data.

Flash memory	Function classification	Function	Functional Description	Comments
Code Flash 1.5MB 1.0MB 768KB 512KB	Programming and Erasing	Automatic Programming	Data programming is performed at 4 words (16 bytes).	
		Automatic chip erasing	Erasing of all the area of a flash memory is performed automatically. Object: Code flash Data flash	Except User information area in code flash.
		Automatic area erasing	Erasing in an area unit is performed automatically.	
		Automatic block erasing	Erasing in a block unit is performed automatically.	
		Automatic page erasing	Erasing in a page unit is performed automatically.	
	Program/erase protection	Protection	Programming and erasing can be prohibited per block.(Note)	
	Security	Security	Prohibition of read-out from the flash memory by a flash writer and using a debugging tools.	
	Memory swap	Automatic memory swap	Swap /swap release /swap size specification of a code flash block is performed automatically.	
	Execute Instruction	Execute Instruction	Instructions can be executed.	
	program/erase to other Flash I/F	program/erase to code/data Flash I/F	Basic operation to a code/data flash can be performed.	Dual mode

Note: First 32KB of each FLASH I/F is protected by page unit.

Flash memory	Function classification	Function	Functional Description	Comments
User information area (Code Flash) 4KB	Programming and Erasing	Automatic Programming	Data programming is performed at 4 words (16 bytes).	
		Automatic page erasing	Erasing all the User information area is performed automatically.	
	Program/erase protection	Protection	Programming and erasing can be prohibited per block.	It is the same as the same address of a code flash.
	Security	Security	Prohibition of read-out of the flash memory by a flash writer and the usage restrictions of a debugging function can be carried out.	It is controlled by the operation on the code flash side.
	Execute Instruction	-	-	Execution of instruction cannot be performed.

Flash memory	Function classification	Function	Functional Description	Comments
Data Flash 32KB	Programming and Erasing	Automatic Programming	Data programming is performed at 1 words (4 bytes).	
		Automatic area erasing	Erasing in an area unit is performed automatically.	
		Automatic block erasing	Erasing in a block unit is performed automatically.	
		Automatic page erasing	Erasing in a page unit is performed automatically.	
	Program/erase protection	Protection	Programming and erasing can be prohibited per block.	
	Security	Security	Prohibition of read-out of the flash memory by a flash writer and the usage restrictions of a debugging function can be carried out.	It becomes effective simultaneously by the operation on the code flash side.
	Execute Instruction	Execute Instruction	Instructions can be executed.	No prefetch buffer
	program/erase to other Flash area	program/erase to code Flash area	Basic operation to a code flash can be performed.	Dual mode

9. Sample Program

When the sample program starts up, the corresponding LED's light or light off according to the content stored in the data table.
When the push switch S7 is pushed down, the corresponding contents in the Flash memory are exchanged, which changes the used LED's.

9.1. Initialization

The following initialization is done after power is supplied.
The initialization of each clock setting and the setting of the watchdog timer are done.

9.2. Main Operation of Sample Program

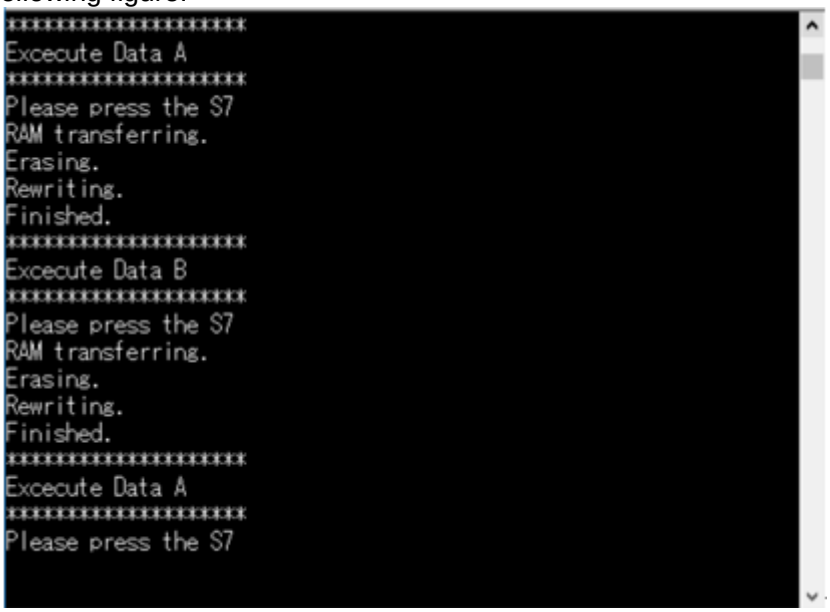
The initialization of BSP is executed.
The initialization of driver is executed.
The initialization of the application is done.
As the initialization of the application, the followings are done; the initialization of the timer, enable of the timer interrupt, the initialization of LED's, the initialization of the push switches, enable of the detect push switch, the initialization of the UART, and enable of the UART interrupt.
The internal oscillator is enabled.

This sample program includes data "A" and "B".
According to the initial data "A" in the data table, the LED's corresponding to the data "A" blink.
The data "A" blinks LED (D10) and LED (D8).
The data "B" blinks LED (D9) and LED (D7).
When the push switch S7 is pushed down, the data "A" and "B" in the Flash memory are copied to the RAM, and the corresponding areas in the Flash memory are erased. After that, the data "B" in the RAM is programmed to the area where the data "A" used to be stored in the Flash memory. And the data "A" is programmed to the area where the data "B" used to be stored.
After the procedure completes, the data table is read again, and the LED's corresponding to the data "B" blink.
The same procedure is done whenever the push switch is pushed down.

The blink of the LED's are controlled by the timer interrupt. In the initial setting, the lighting and lights-off repeat per 1 second.
The procedure of the timer interrupt is done after checking that the data table's content is the data "A" or the data "B".
The reset switch is pushed down to program data to the Flash memory. When the data "B" is stored in the Flash memory, for example, the LED's corresponding to the data "B" blink after the deassertion of the reset.

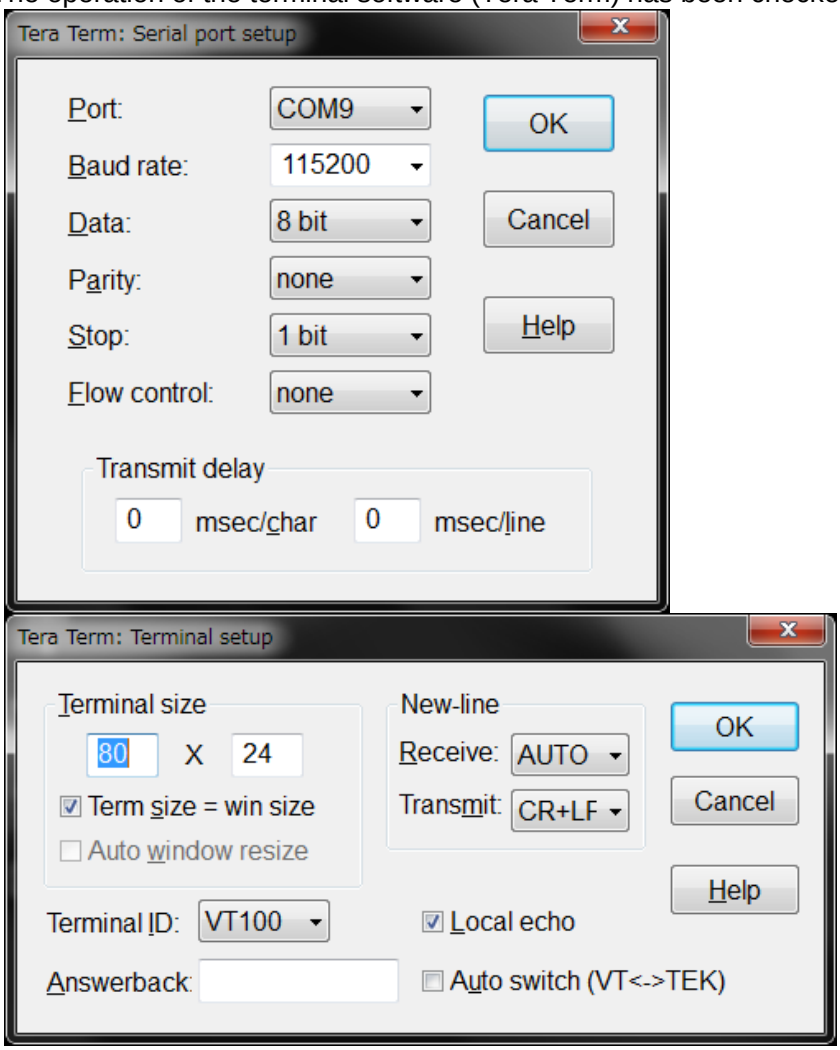
9.3. Output Example of Sample Program

When the sample program executes, the rewrite log of the Flash memory is output as shown in the following figure.



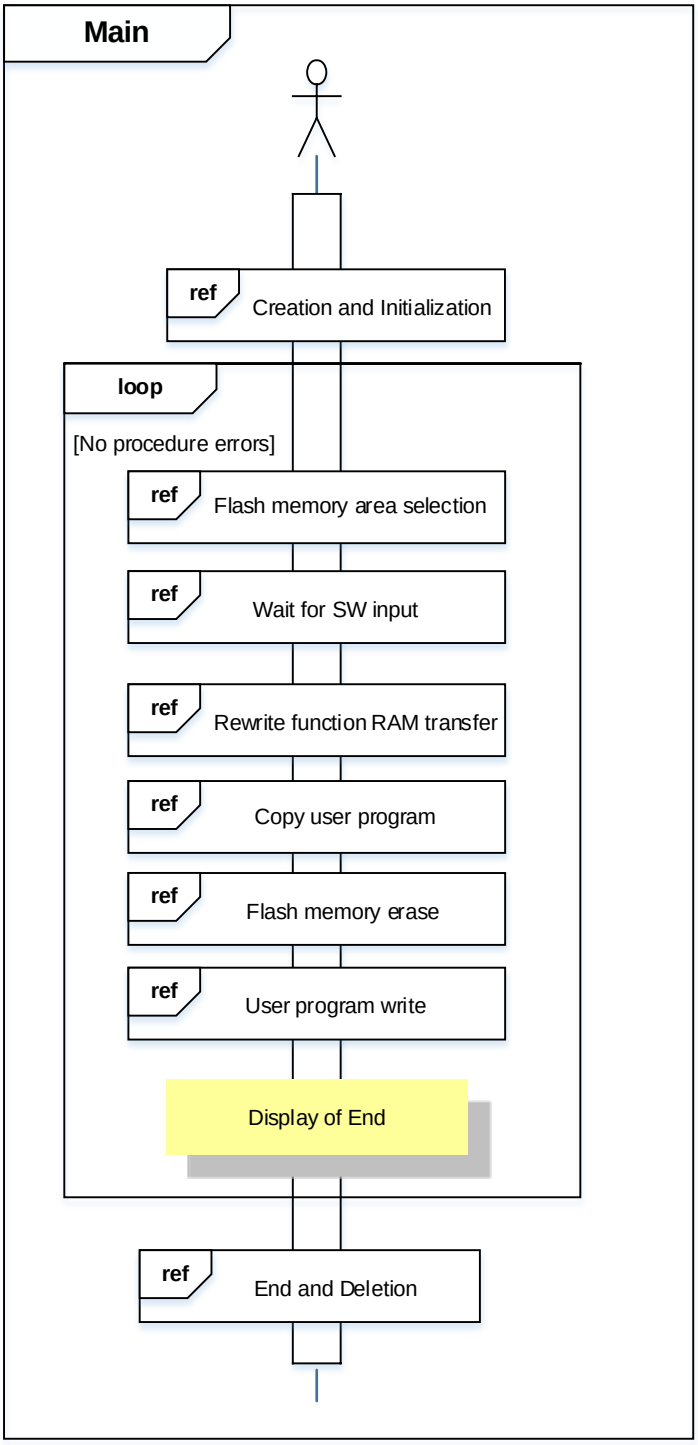
9.3.1. Setting Example of Terminal Software

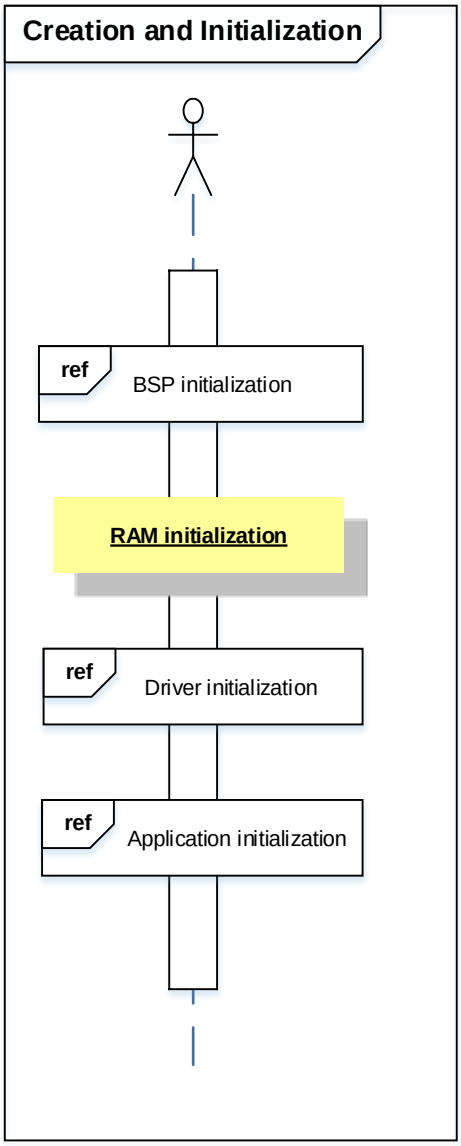
The operation of the terminal software (Tera Term) has been checked with the following settings.

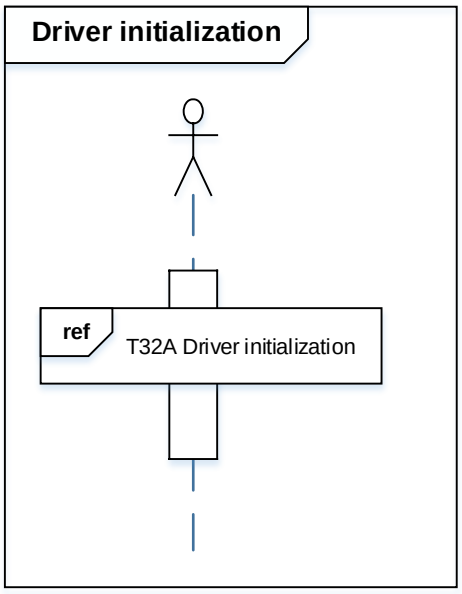


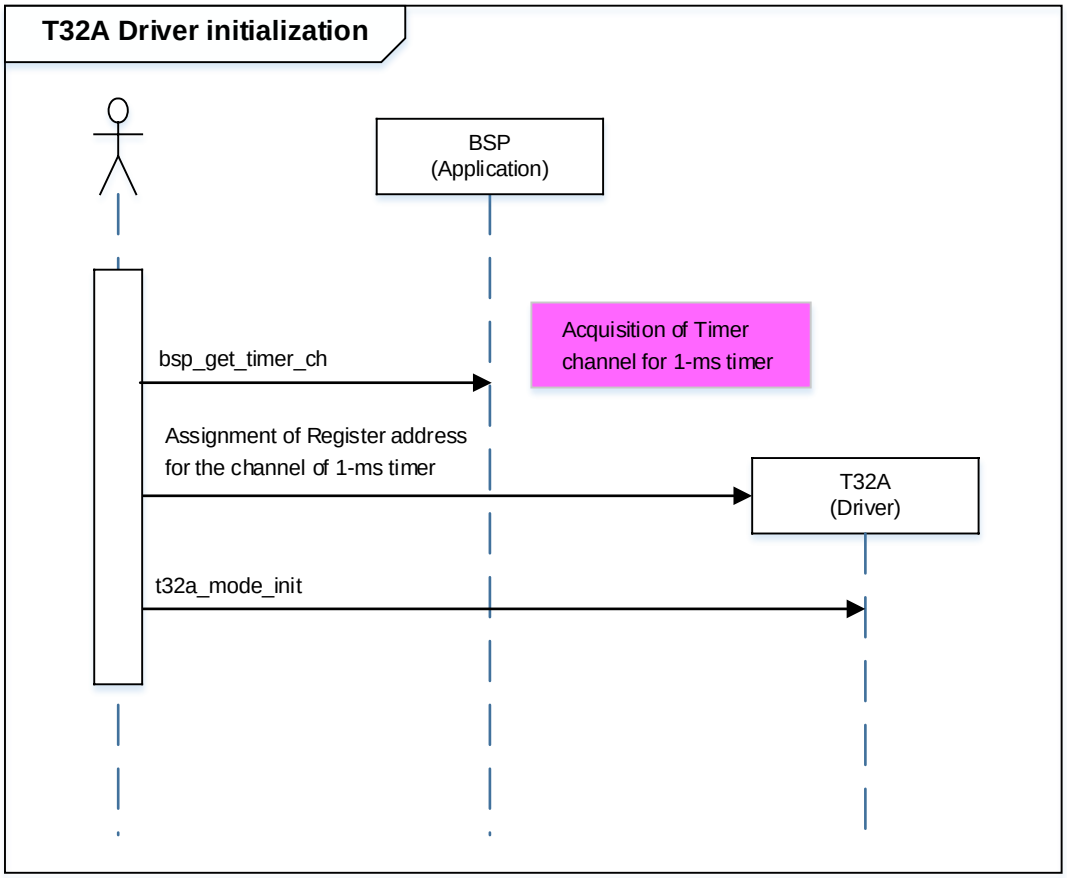
9.4. Operating Flow of Sample Program

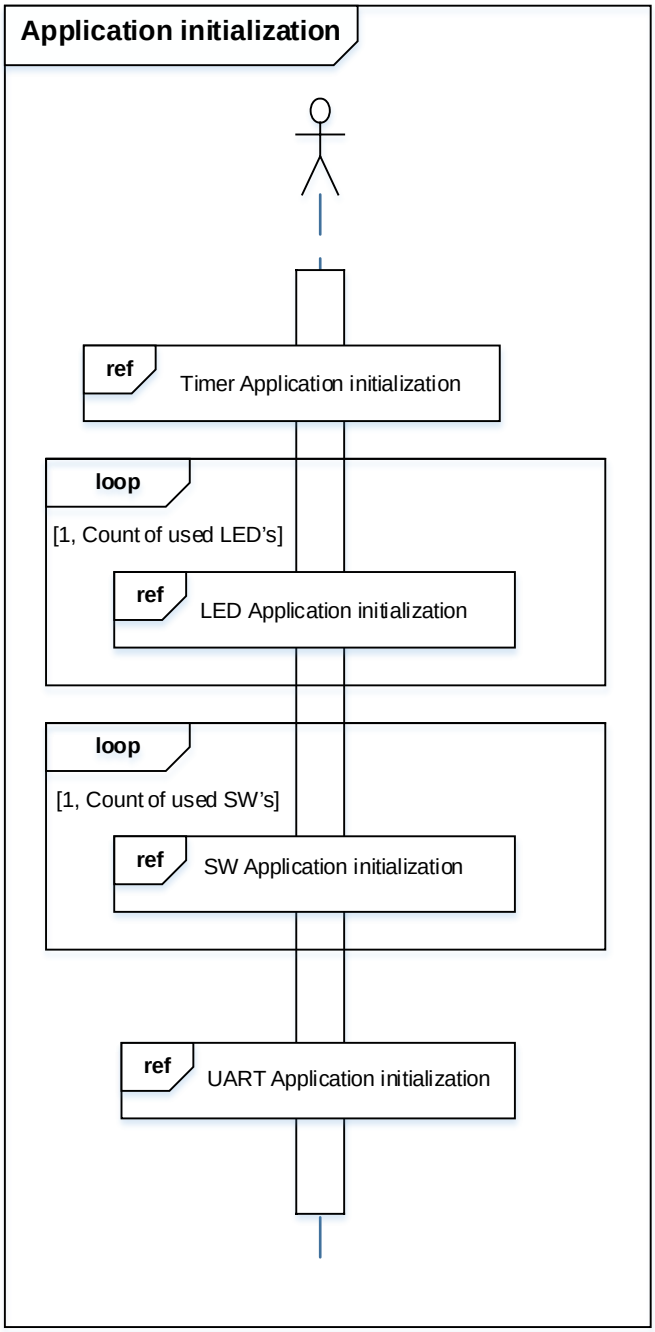
The basic operating flows of the sample program are shown in the following;

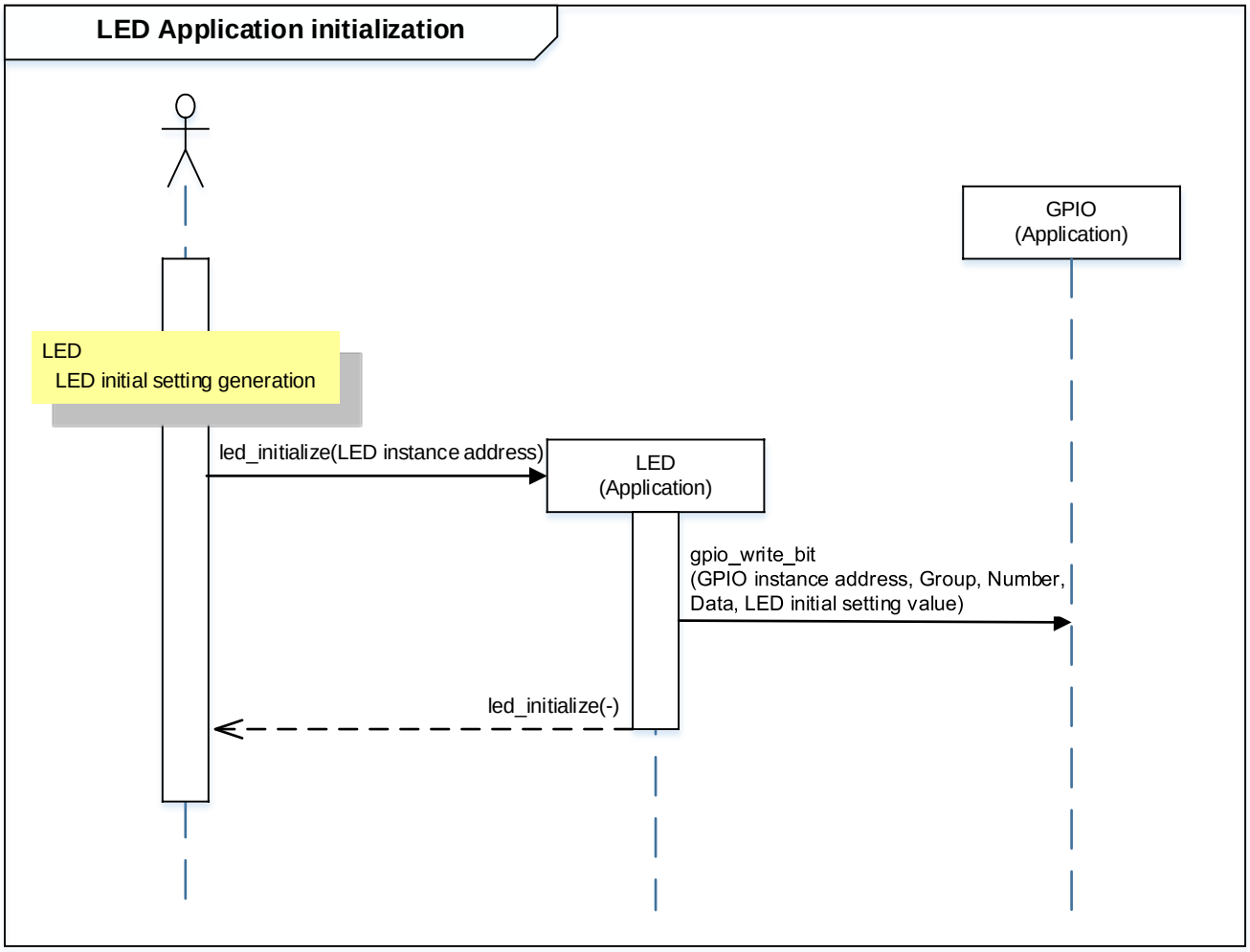


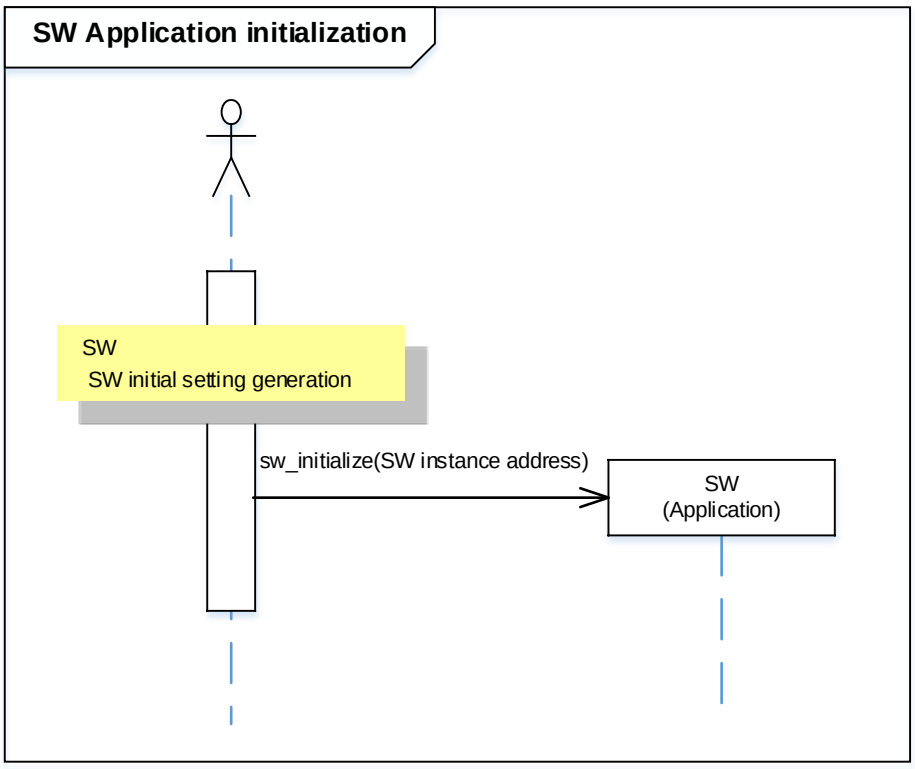

Creation and Initialization
ref
BSP initialization
RAM initialization
ref
Driver initialization
ref
Application initialization

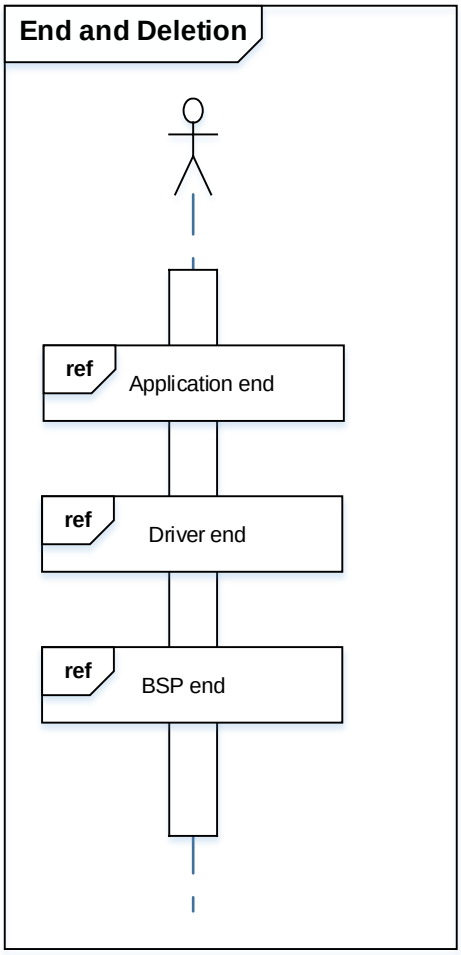

Driver initialization
ref
T32A Driver initialization

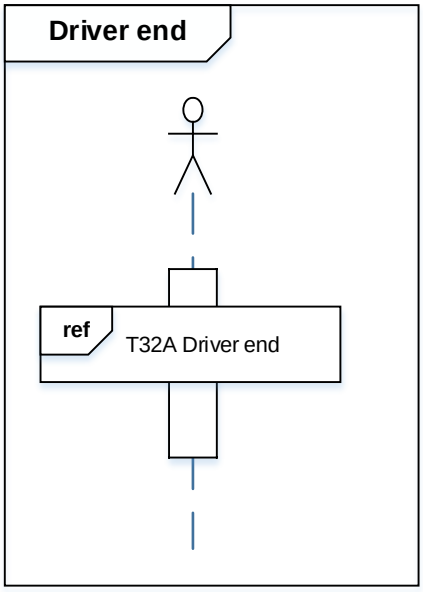

T32A Driver initialization
BSP
(Application)
Acquisition of Timer
channel for 1-ms timer
bsp_get_timer_ch
Assignment of Register address
for the channel of 1-ms timer
T32A
(Driver)
t32a_mode_init

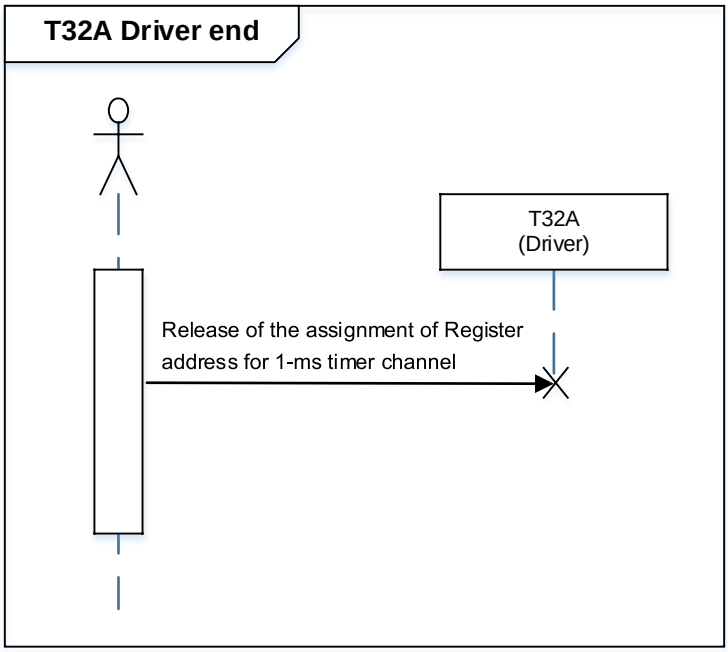

Application initialization
ref
Timer Application initialization
loop
[1, Count of used LED's]
ref
LED Application initialization
loop
[1, Count of used SW's]
ref
SW Application initialization
ref
UART Application initialization

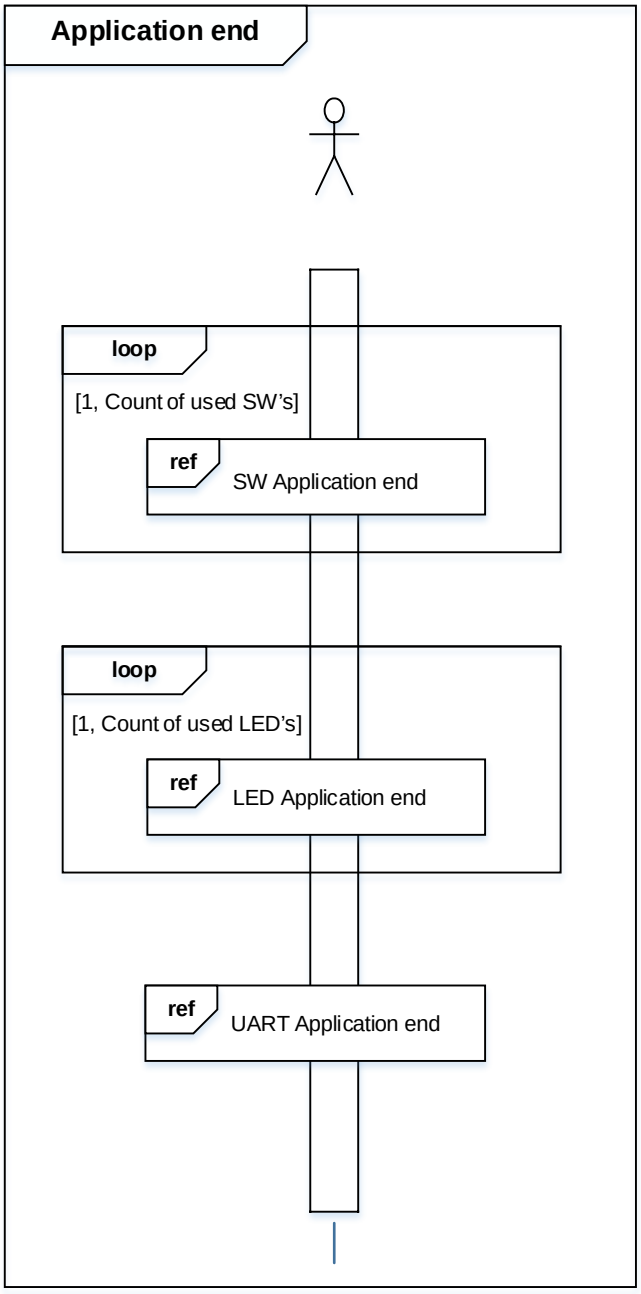

LED Application initialization
GPIO
(Application)
LED
LED initial setting generation
led_initialize(LED instance address)
LED
(Application)
gpio_write_bit
(GPIO instance address, Group, Number,
Data, LED initial setting value)
led_initialize(-)

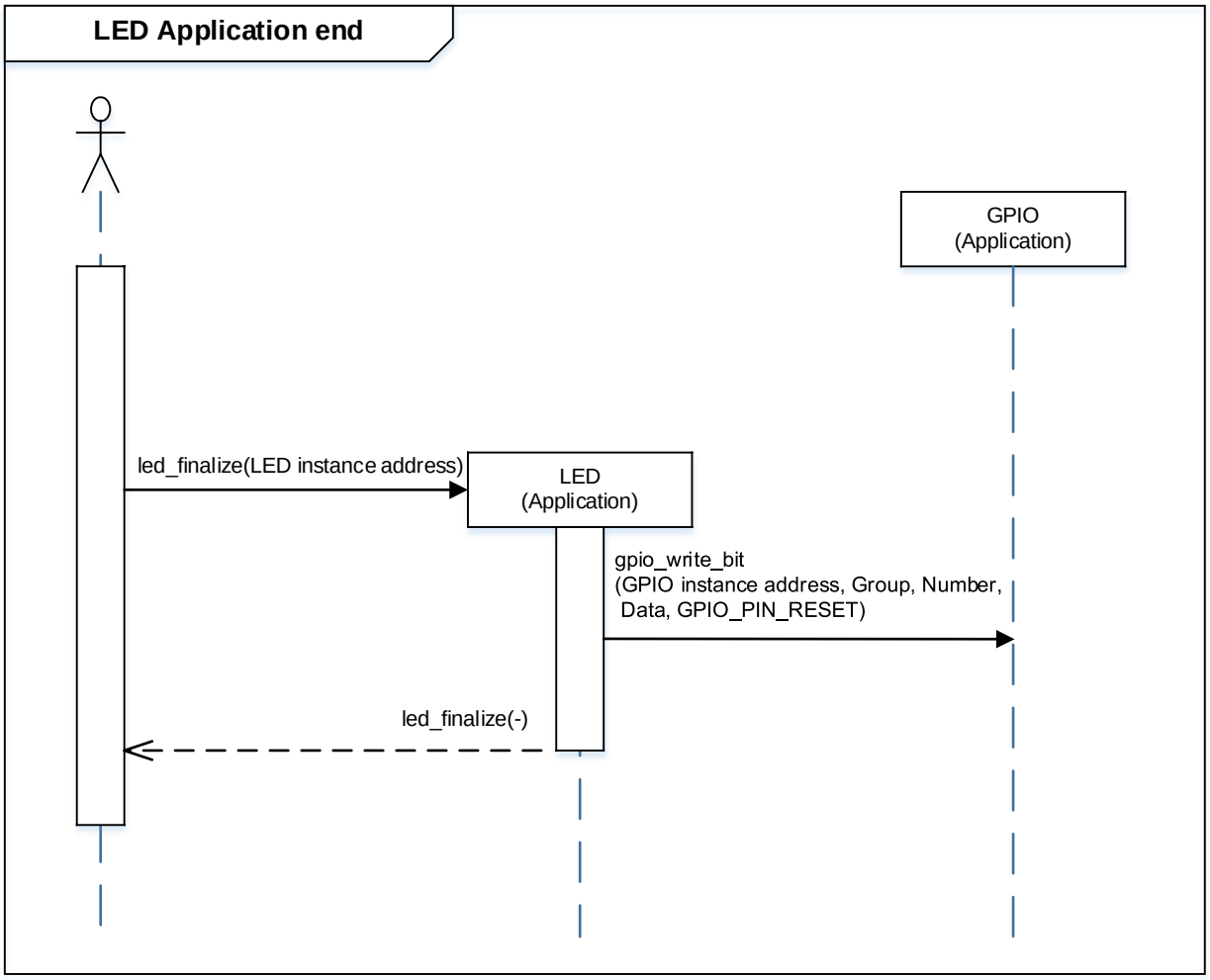

SW Application initialization
SW
SW initial setting generation
sw_initialize(SW instance address)
SW
(Application)

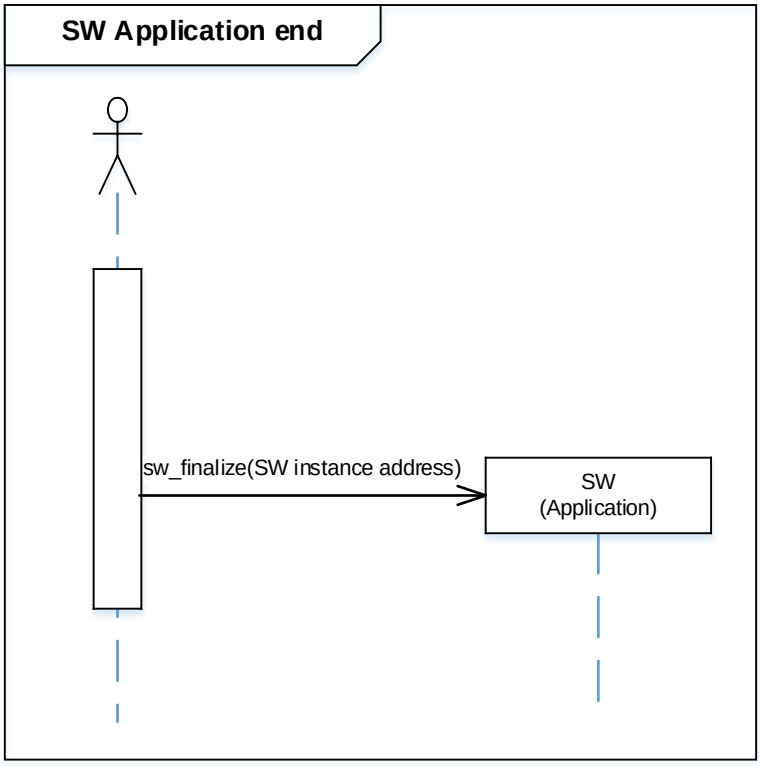

End and Deletion
ref
Application end
ref
Driver end
ref
BSP end

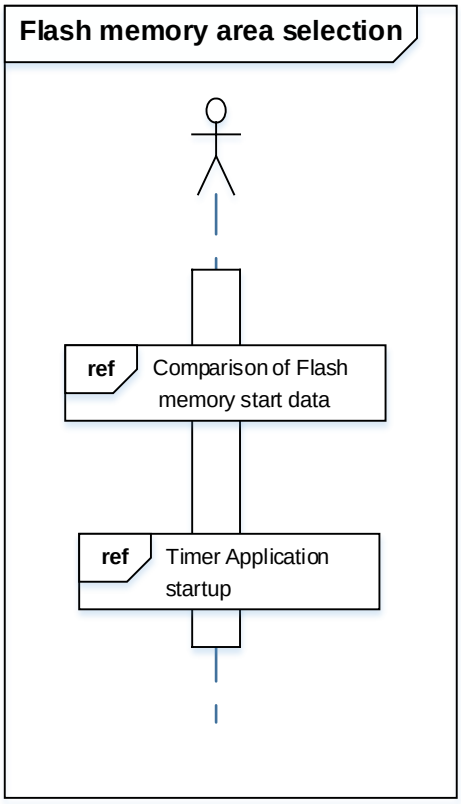

Driver end
ref
T32A Driver end

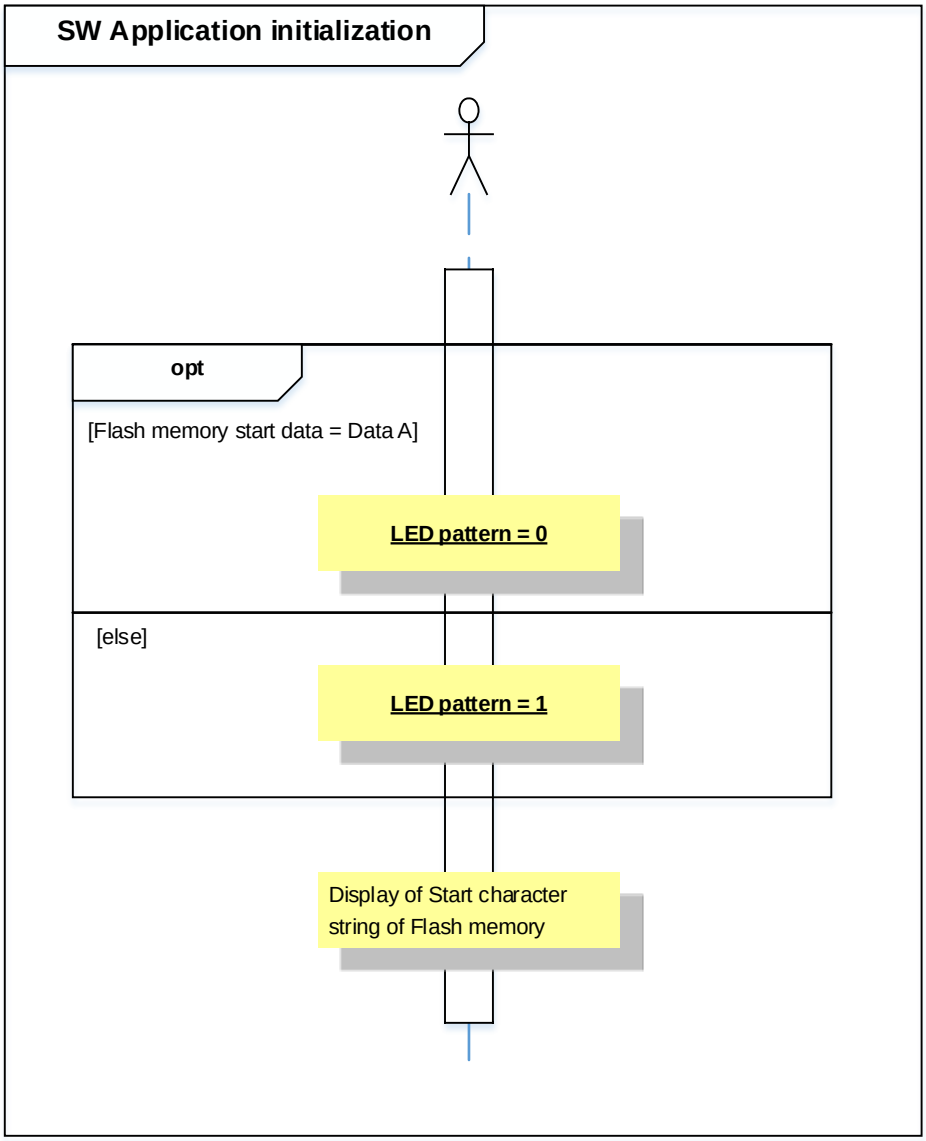

T32A Driver end
T32A
(Driver)
Release of the assignment of Register
address for 1-ms timer channel

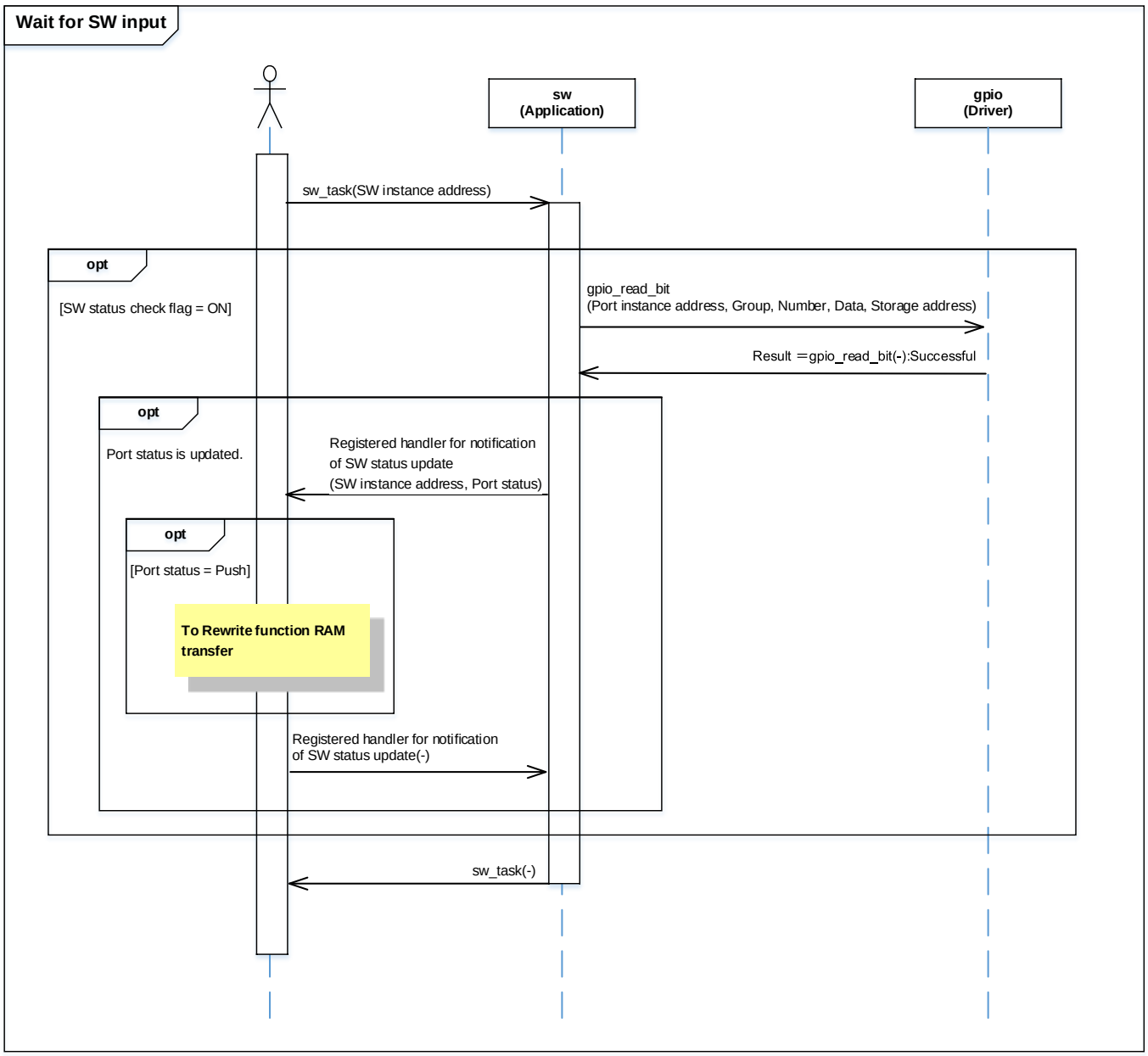

Application end
loop
[1, Count of used SW's]
ref
SW Application end
loop
[1, Count of used LED's]
ref
LED Application end
ref
UART Application end

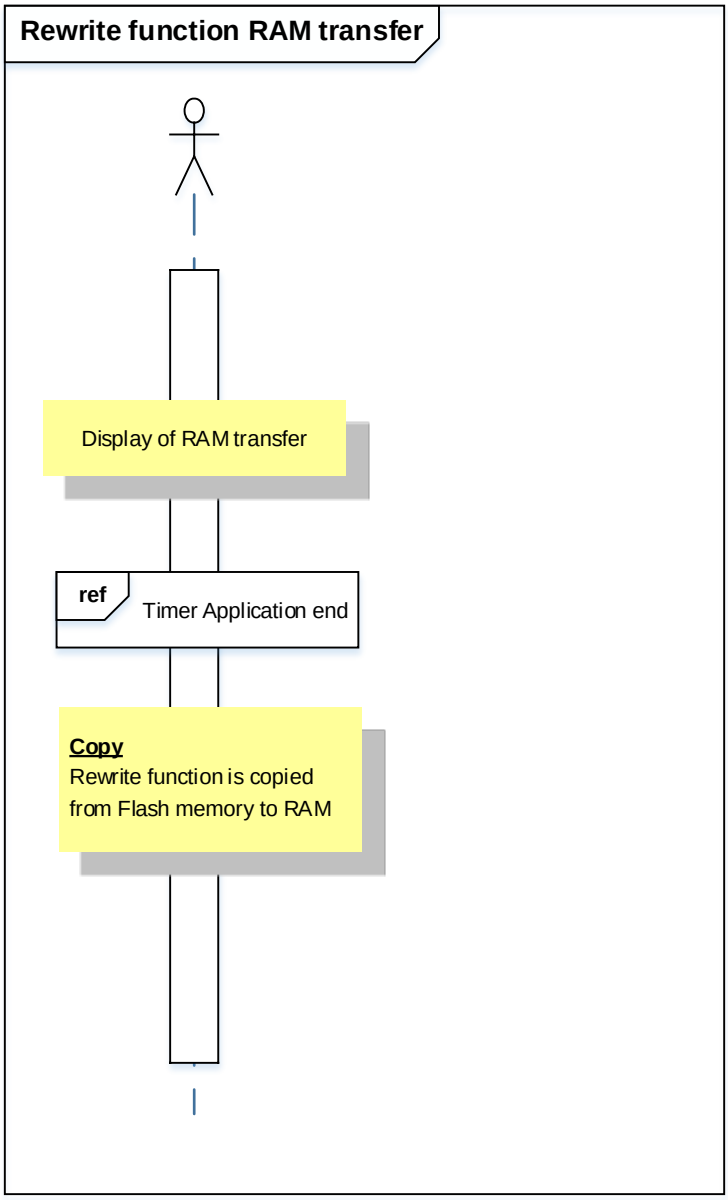

LED Application end
GPIO
(Application)
led_finalize(LED instance address)
LED
(Application)
gpio_write_bit
(GPIO instance address, Group, Number,
Data, GPIO_PIN_RESET)
led_finalize(-)

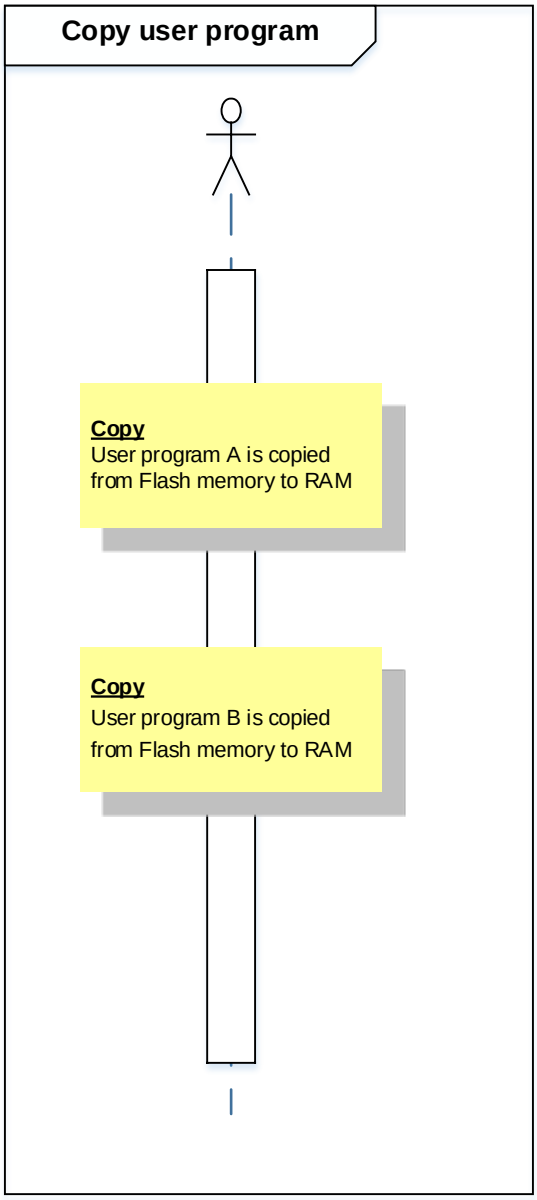

SW Application end
sw_finalize(SW instance address)
SW
(Application)

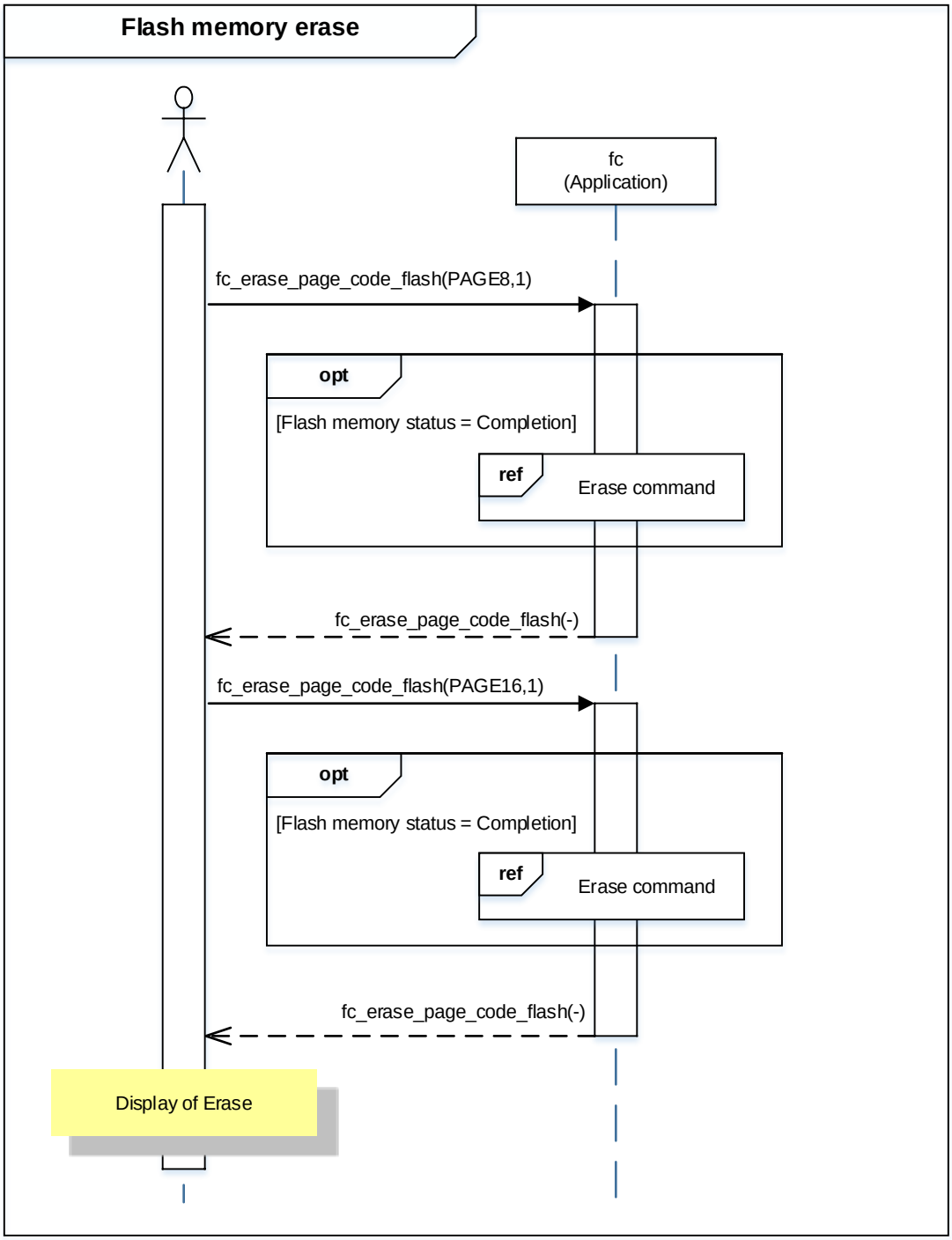

Flash memory area selection
ref
Comparison of Flash memory start data
ref
Timer Application startup

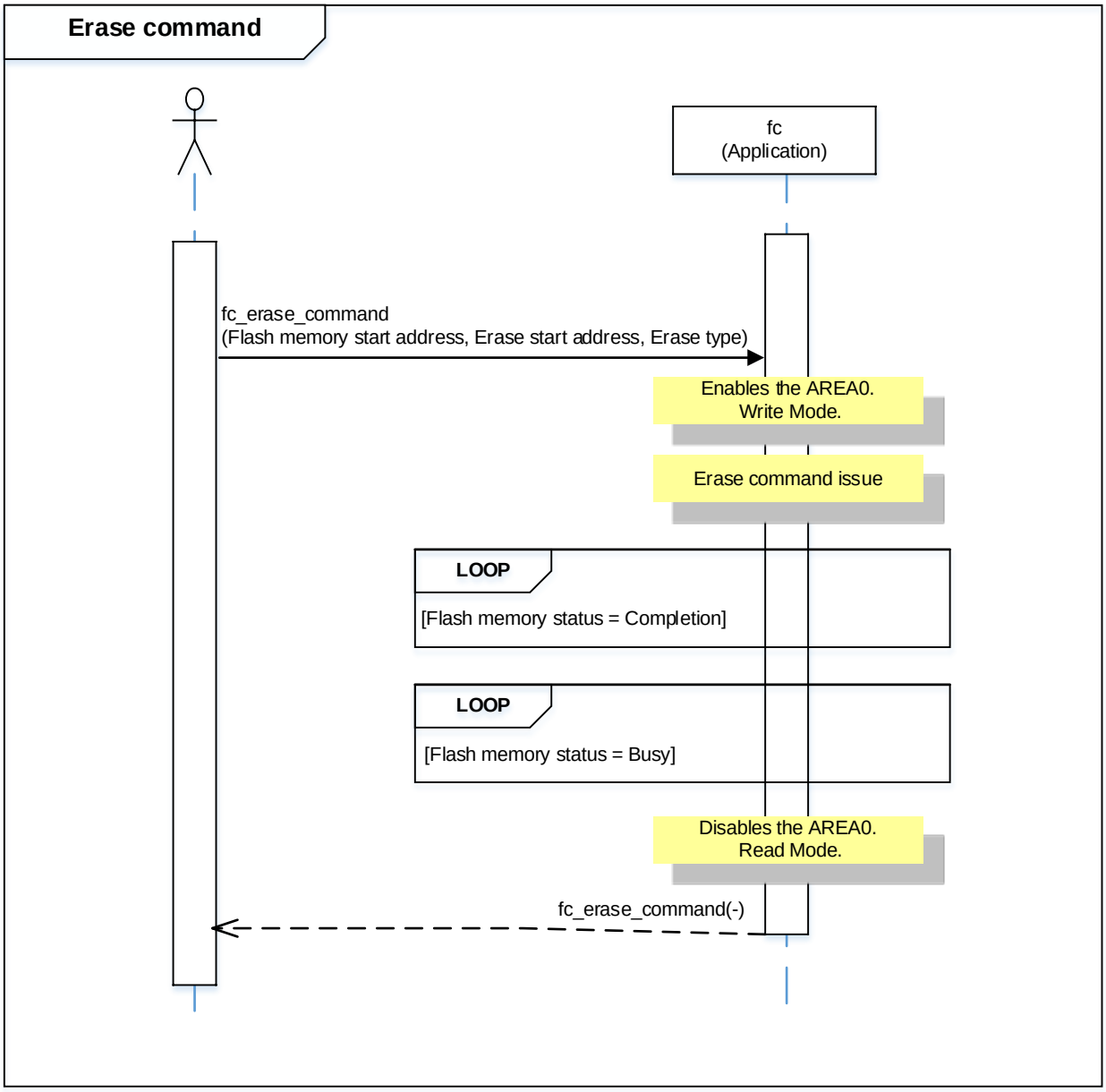

SW Application initialization
opt
[Flash memory start data = Data A]
LED pattern = 0
[else]
LED pattern = 1
Display of Start character string of Flash memory

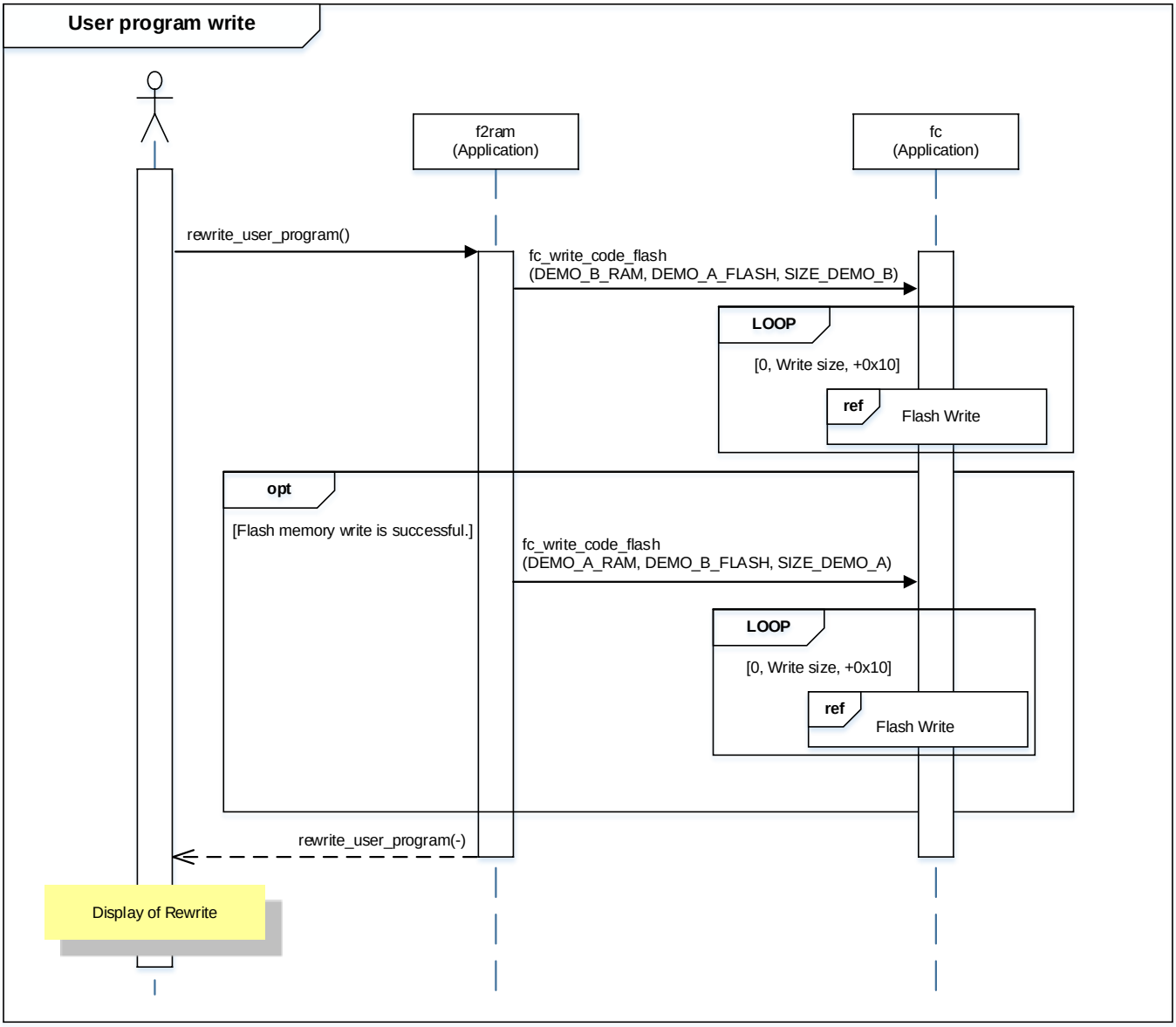

Wait for SW input
sw
(Application)
gpio
(Driver)
sw_task(SW instance address)
opt
[SW status check flag = ON]
gpio_read_bit
(Port instance address, Group, Number, Data, Storage address)
Result =gpio_read_bit(-):Successful
opt
Port status is updated.
Registered handler for notification
of SW status update
(SW instance address, Port status)
opt
[Port status = Push]
To Rewrite function RAM
transfer
Registered handler for notification
of SW status update(-)
sw_task(-)

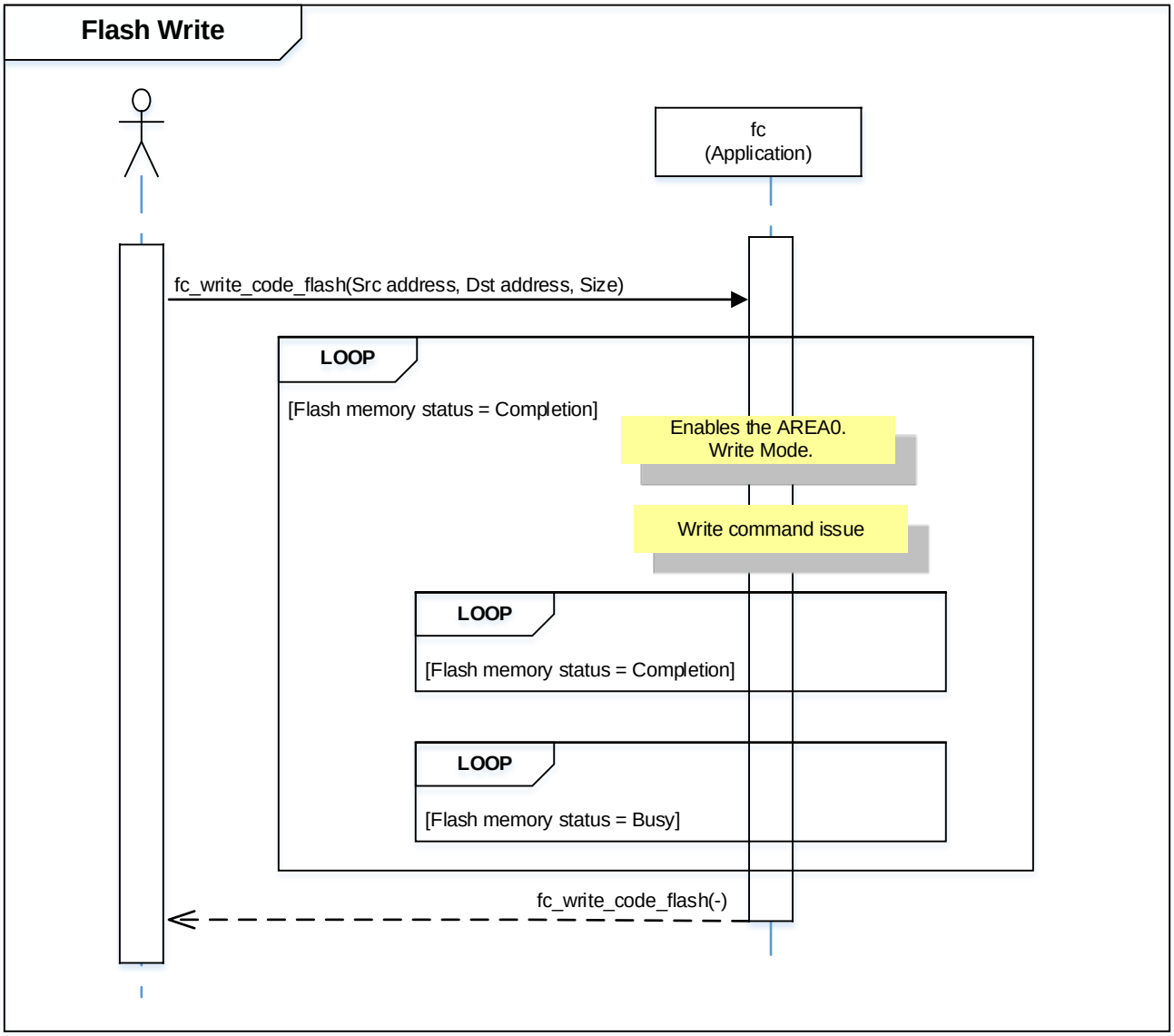

Rewrite function RAM transfer
Display of RAM transfer
ref
Timer Application end
Copy
Rewrite function is copied
from Flash memory to RAM

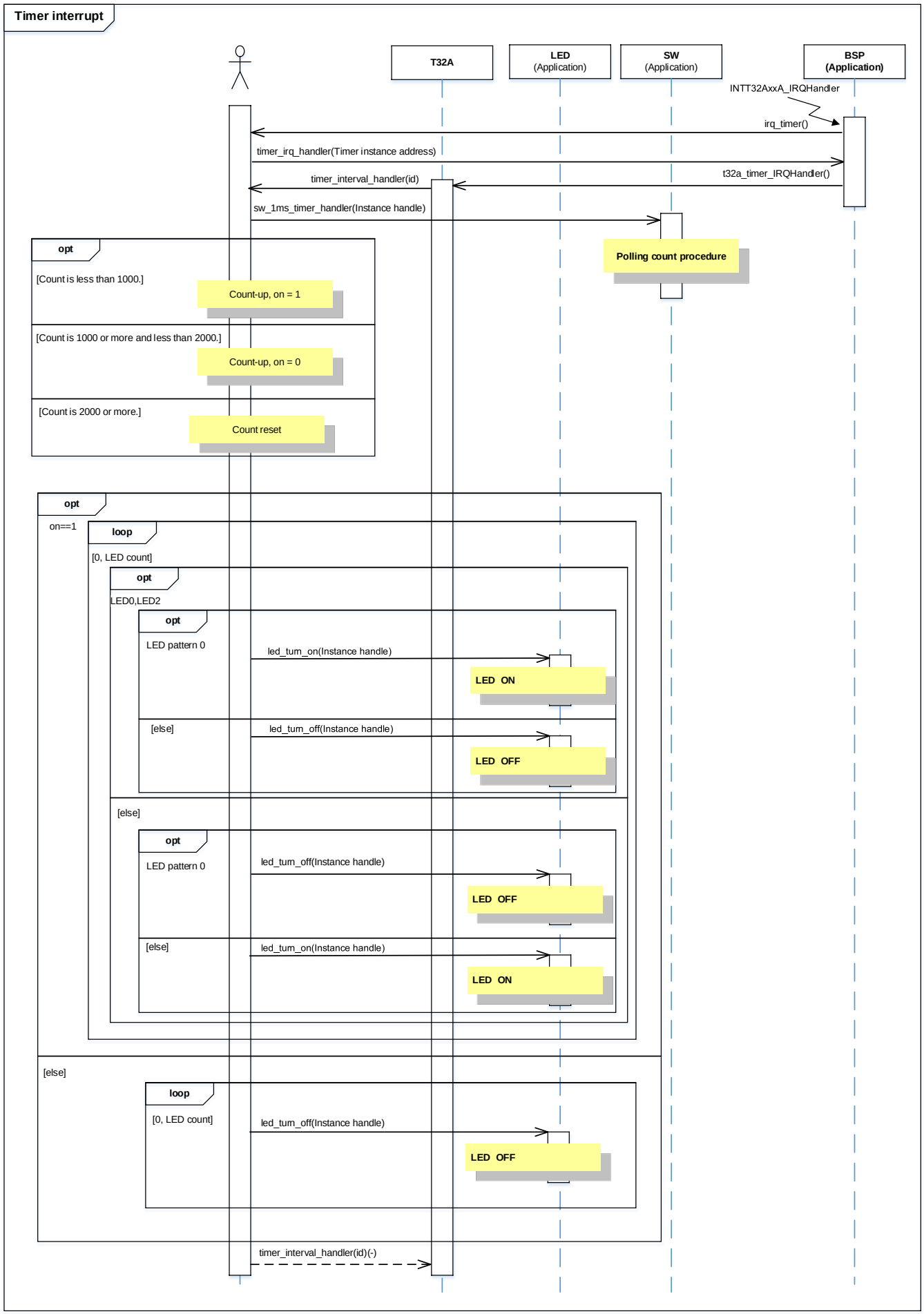

Copy user program
Copy
User program A is copied
from Flash memory to RAM
Copy
User program B is copied
from Flash memory to RAM


Flash memory erase
fc
(Application)
fc_erase_page_code_flash(PAGE8,1)
opt
[Flash memory status = Completion]
ref
Erase command
fc_erase_page_code_flash(-)
fc_erase_page_code_flash(PAGE16,1)
opt
[Flash memory status = Completion]
ref
Erase command
fc_erase_page_code_flash(-)
Display of Erase


Erase command
fc
(Application)
fc_erase_command
(Flash memory start address, Erase start address, Erase type)
Enables the AREA0.
Write Mode.
Erase command issue
LOOP
[Flash memory status = Completion]
LOOP
[Flash memory status = Busy]
Disables the AREA0.
Read Mode.
fc_erase_command(-)


User program write
f2ram
(Application)
fc
(Application)
rewrite_user_program()
fc_write_code_flash
(DEMO_B_RAM, DEMO_A_FLASH, SIZE_DEMO_B)
LOOP
[0, Write size, +0x10]
ref
Flash Write
opt
[Flash memory write is successful.]
fc_write_code_flash
(DEMO_A_RAM, DEMO_B_FLASH, SIZE_DEMO_A)
LOOP
[0, Write size, +0x10]
ref
Flash Write
rewrite_user_program(-)
Display of Rewrite


Flash Write
fc
(Application)
fc_write_code_flash(Src address, Dst address, Size)
LOOP
[Flash memory status = Completion]
Enables the AREA0.
Write Mode.
Write command issue
LOOP
[Flash memory status = Completion]
LOOP
[Flash memory status = Busy]
fc_write_code_flash(-)


Timer interrupt
T32A
LED (Application)
SW (Application)
BSP (Application)
INTT32AxxA_IRQHandler
irq_timer()
timer_irq_handler(Timer instance address)
t32a_timer_IRQHandler()
timer_interval_handler(id)
sw_1ms_timer_handler(Instance handle)
Polling count procedure
opt
[Count is less than 1000.]
Count-up, on = 1
[Count is 1000 or more and less than 2000.]
Count-up, on = 0
[Count is 2000 or more.]
Count reset
opt
on==1
loop
[0, LED count]
opt
LED0,LED2
opt
LED pattern 0
led_tum_on(Instance handle)
LED ON
[else]
led_tum_off(Instance handle)
LED OFF
[else]
opt
LED pattern 0
led_tum_off(Instance handle)
LED OFF
[else]
led_tum_on(Instance handle)
LED ON
[else]
loop
[0, LED count]
led_tum_off(Instance handle)
LED OFF
timer_interval_handler(id)(-)

10. Precaution

When using the sample program with CPU other than TMPM4G9F15, please check operation sufficiently.

11. Revision History

Rev	Date	Description
1.0	2018-11-28	First release

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, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **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.**