# M4G Group (1)

# Application Note
# Full Universal Asynchronous Receiver Transmitter Circuit

# (FUART-B)

## Outlines

This application note is a reference material for developing products using Full Universal Asynchronous Receiver Transmitter Circuit (FUART) function of M4G group (1).
This document helps the user check operation of the product and develop its program.

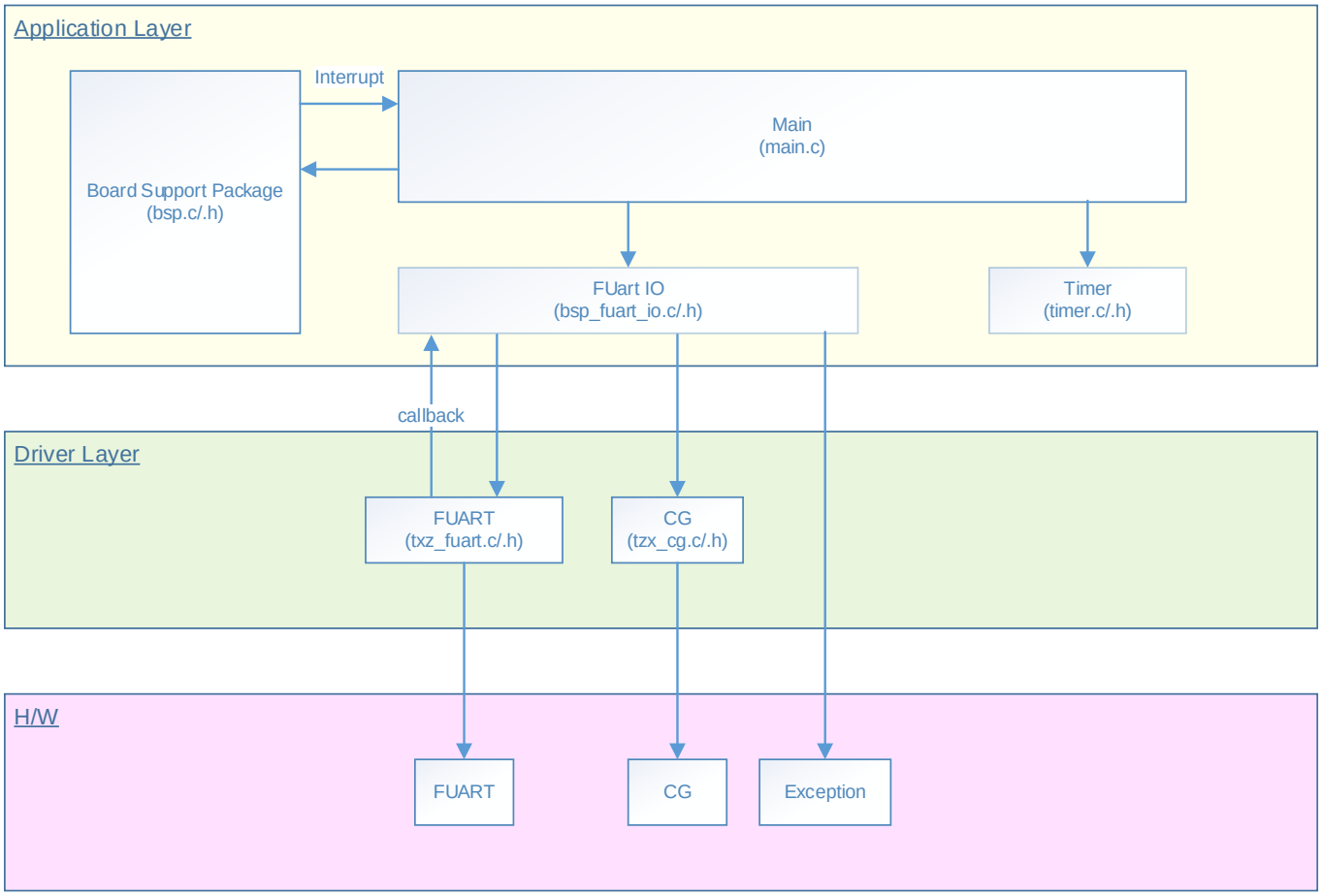
Target sample program: FUART_Echo

# Table of Contents

# 1. Preface

This sample program should be used to check the operation of the FUART.
The data input from the terminal software is echoed back using the FUART. The data is output to the terminal software.

Structure diagram of Sample program



## 2. Reference Document

- Datasheet
  TMPM4G group (1) datasheet Rev1.0 (Japanese edition)
- Reference manual
  Full Universal Asynchronous Receiver Transmitter Circuit (FUART-B) 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 |
|---|---|---|---|
| Full Universal Asynchronous Receiver Transmitter Circuit | ch0 | PG4 (FUT0TXD)<br>PG5 (FUT0RXD) | FUART mode |

## 4. Target Device

The target devices of this application note are as follows.

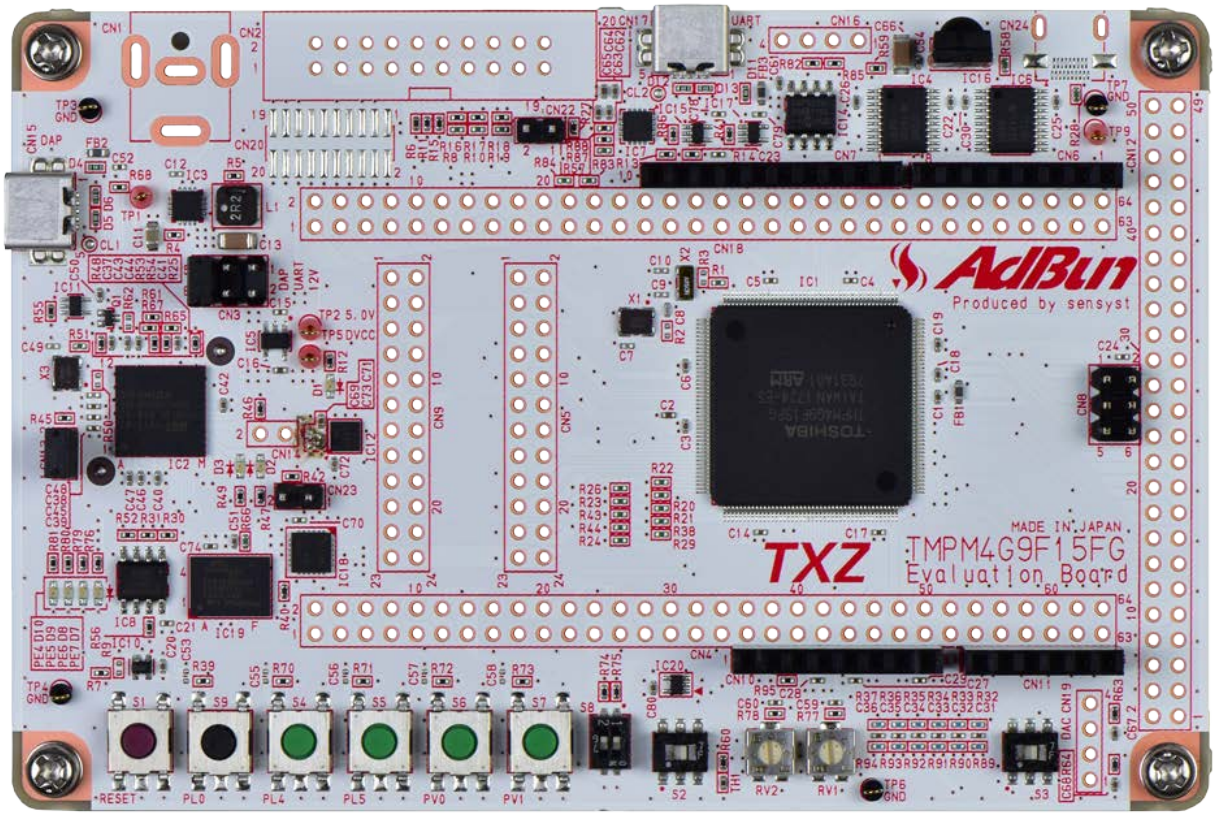
| | | | |
|---|---|---|---|
| TMPM4G9F15FG | TMPM4G9F10FG | TMPM4G9FEFG | TMPM4G9FDFG |
| TMPM4G9F15XBG | TMPM4G9F10XBG | TMPM4G9FEXBG | TMPM4G9FDXBG |
| TMPM4G8F15FG | TMPM4G8F10FG | TMPM4G8FEFG | TMPM4G8FDFG |
| TMPM4G8F15XBG | TMPM4G8F10XBG | TMPM4G8FEXBG | TMPM4G8FDXBG |
| | TMPM4G7F10FG | TMPM4G7FEFG | 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.

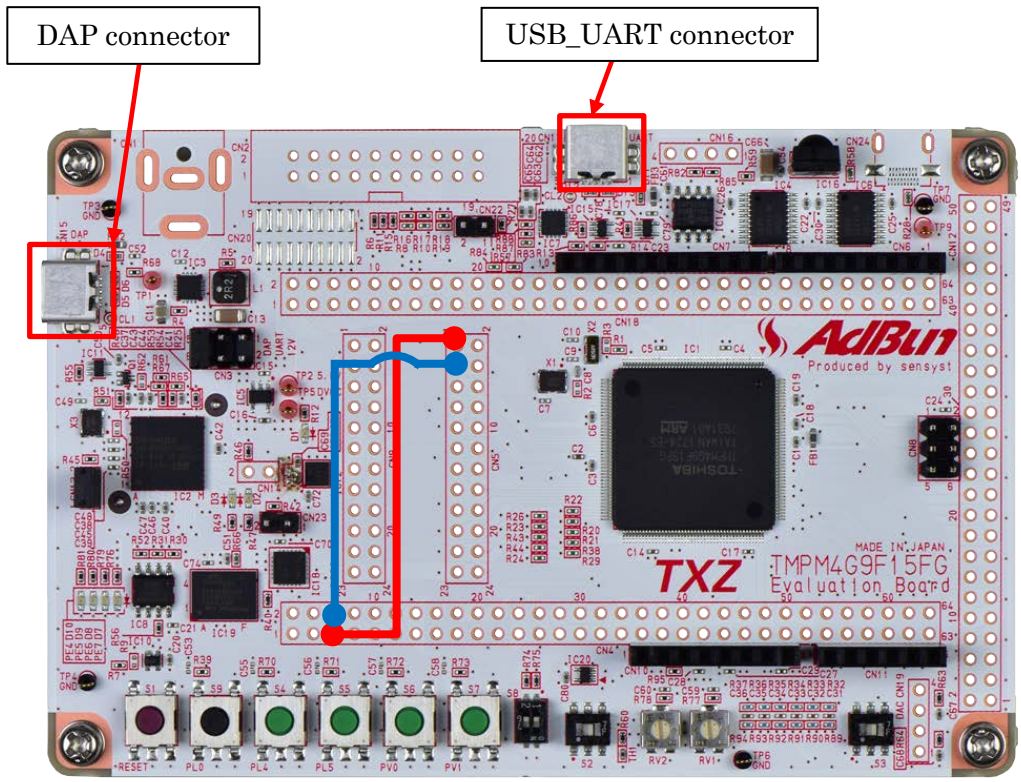
# 5. Operation Confirmation Condition

| | |
|---|---|
| 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




* Connection between through-holes in the evaluation board
[CN5] No.1 and [CN4] No.5 on the evaluation board should be connected.
[CN5] No.3 and [CN4] No.6 on the evaluation board should be connected.

## 7. Operation of Evaluation Board

PC and the USB_UART are connected for communication with the terminal software.
After the sample program starts up, the terminal software receives a user input.
When a user inputs data, the data is echoed back, and is output to the terminal software through the FUART.

# 8. Outline of FUART Function

The Full Universal Asynchronous Receiver Transmitter (FUART) can operate as a transmission/reception circuit of 1 channel (FUTxTXD/FUTxRXD/FUTxCTS_N/FUTxRTS_N) per unit. And reception function using IrDA circuit.The following is a list of functions.

| Function classification | Function | Operation explanation |
|---|---|---|
| Baud rate control | Frequency dividing of prescaler | Selectable from 1/1, 1/2, 1/4 to 1/512 of the ΦT0 frequency. |
| | Baud rate generator | N + (K / 64) (N = 2 to 65535 and K = 0 to 63) dividing of the source clock frequency are possible. |
| | Baud rate (Note1) | Maximum 921.6Kbps |
| Data format | Data length | Selectable 5, 6, 7 or 8 bit. |
| | Parity | Parity control: Enable or disable selection<br>Parity type: Even or odd parity is selectable |
| | Stop bit length | Selectable 1-bit or 2-bit. |
| | Data transfer order | LSB first |
| Transmission/ reception control | FIFO function | FIFO ON and OFF is selectable. |
| | FIFO storage stages | Reception: 32 stages (12-bit width)<br>Transmission: 32 stages(8-bit width) |
| | Error detection | Reception: Parity error, Framing error, Break error, Overrun error |
| | Handshake function | Transmission/reception control by handshake with FUTxCTS_N/FUTxRTS_N signal is possible. |
| IrDA 1.0 circuit | Data rate | Maximum 115.2 kbps (Half duplex) |
| | Mode | Normal IrDA mode and Low power IrDA mode |
| Interlocking control | Interrupt | Interrupt of combination of a transmission/reception completion interrupt, error occurrence interrupt, and a receive timeout interrupt |
| | DMA request (Note2) | Reception DMA request: Burst transfer or Single transfer<br>Transmission DMA request: Burst transfer or Single transfer |

Note1: Regarding Maximum baud rate, please refer datasheet each production.
Note2: Supported DMA requests depend on the product. Please refer to reference manual of “Product Information” for details.

# 9. Sample Program

The data received from the terminal software is echoed back, and the data is output to the terminal software through the FUART.

## 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. Sample Program Main Operation

After the initialization operation, shift to the main function and perform the following initialization.
The initialization of BSP is done.
A variable setting and the data clear are done.
The initialization of driver is done.
The initialization of the application is done.
As the application initialization, the timer and the FUART are initialized.
After the procedure above has been done,"Input =" on the terminal software will be displayed and it will wait for input.

After a character is input, the character is displayed after “Echo =” as the echo-back output.
Then, a character is received again. This procedure can be repeated.

## 9.3. Method of Input Character String Transmission

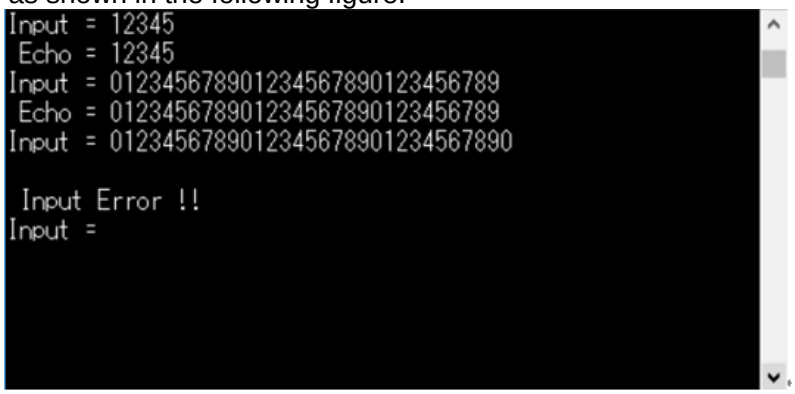
When “proc = PROC_STRING” is enabled in “main.c”, the data transmission is not done until the characters of the specified count including a line-break code are input completely (STRING transmission).
Usually, the data transmission starts when a line-break code is input.

```
#if 0
                    proc = PROC_STRING;
#else
                    proc = PROC_OUTPUT_MSG;
#endif
```

When “proc = PROC_STRING;” is enabled, the data transmission is done after the characters of the specified count are input.
The specified count is 24 as the initial setting value in this sample program.
In this sample program, data are transmitted 3 times by the line-break code, and the STRING transmission is done at the fourth time.
This can be used as a reference when the mixture of both transmissions is done.

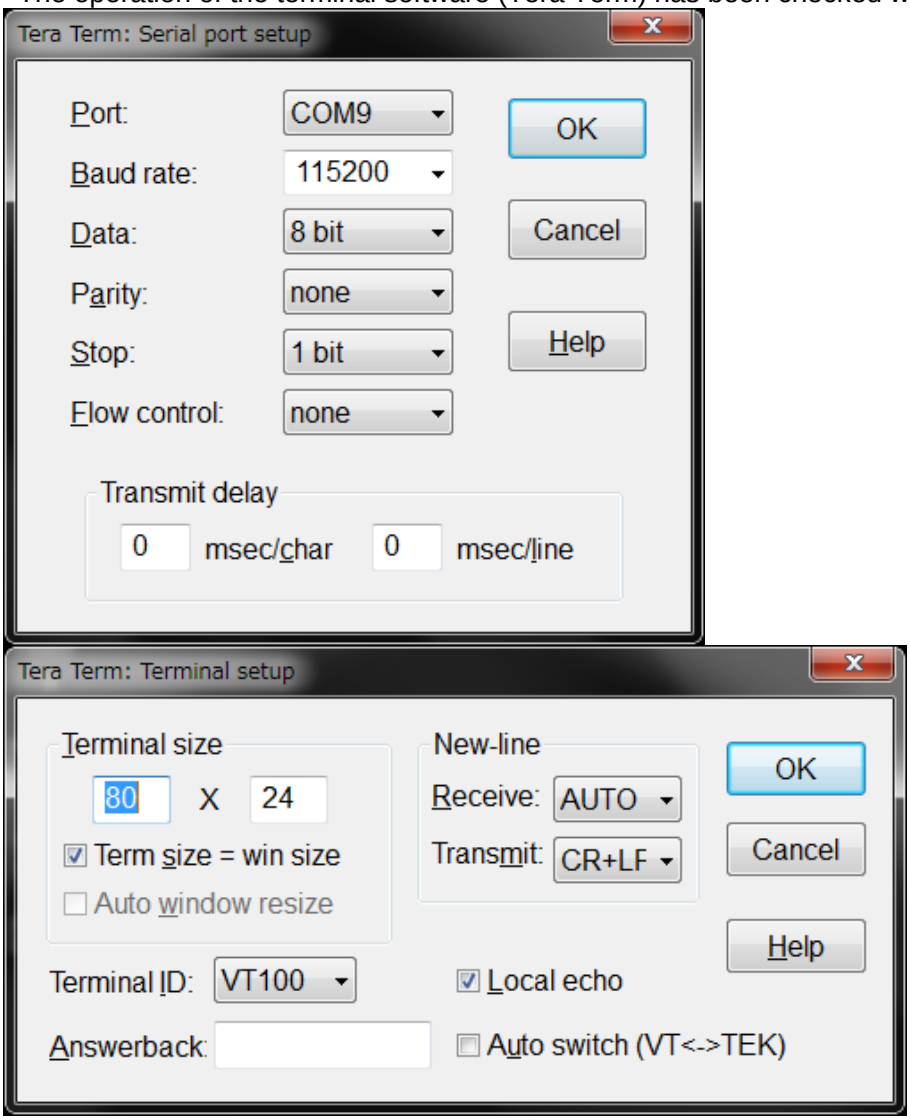
### 9.4. Output Example of Sample Program

When the sample program executes, the input data is echoed back and is output to the terminal software as shown in the following figure.



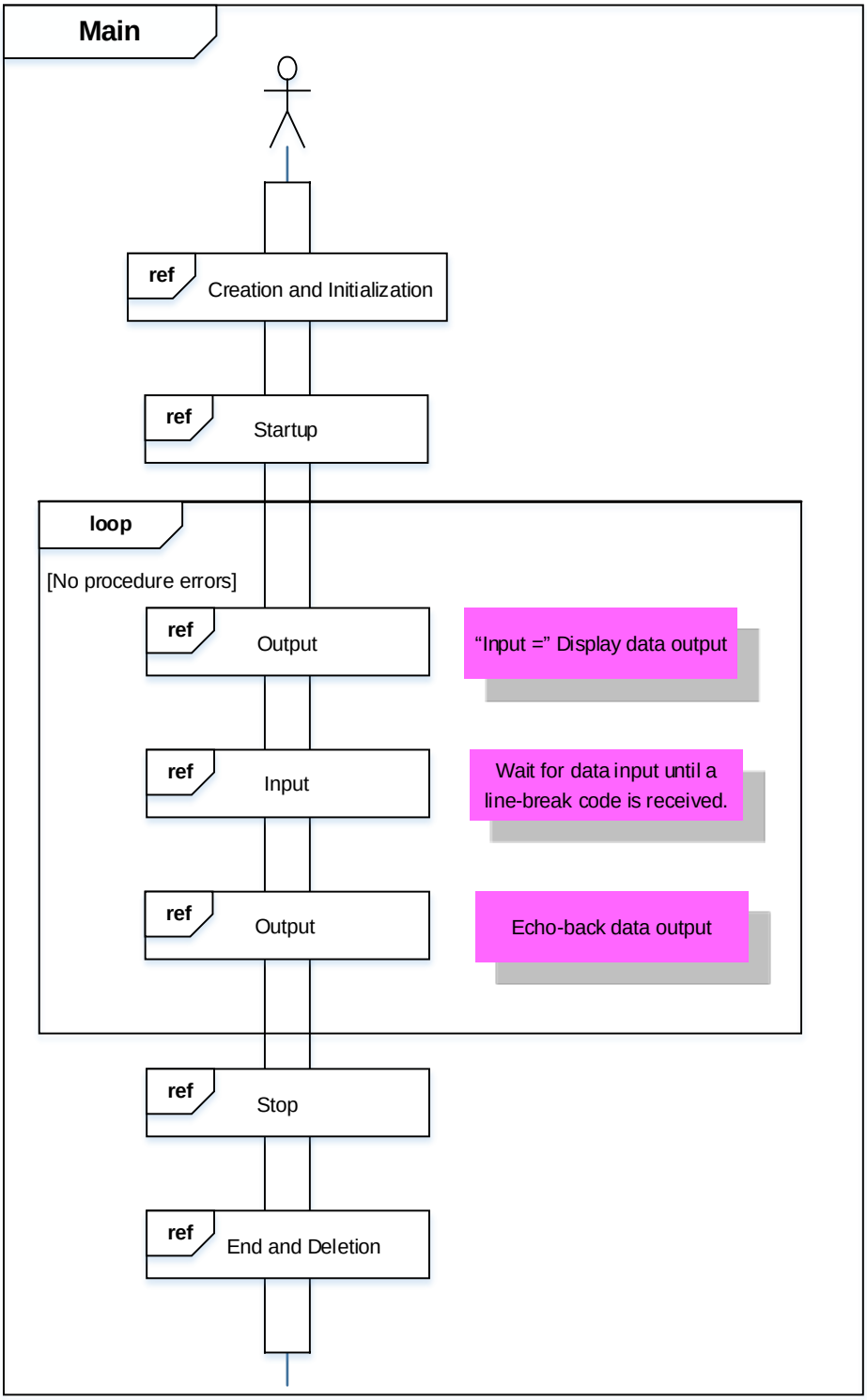

#### 9.4.1. Setting Example of Terminal Software

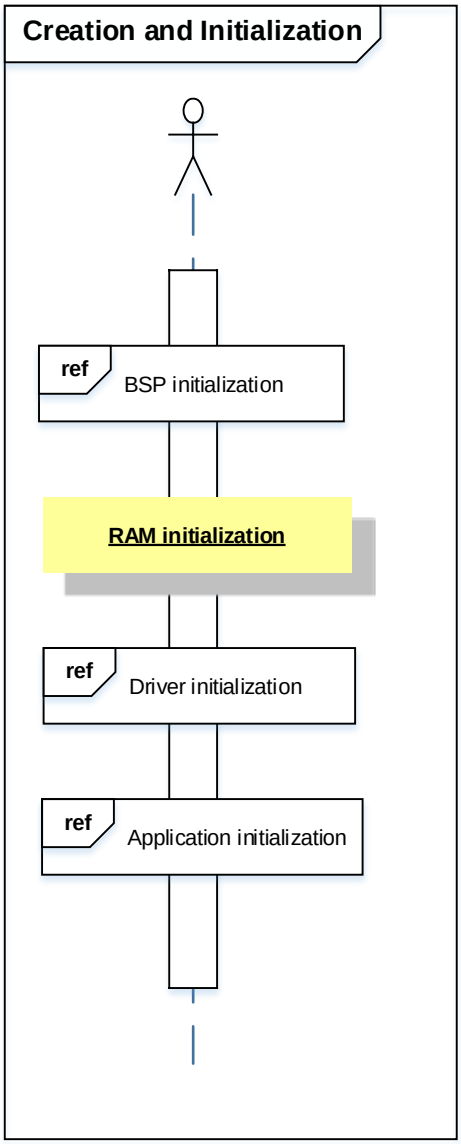
The operation of the terminal software (Tera Term) has been checked with the following settings.

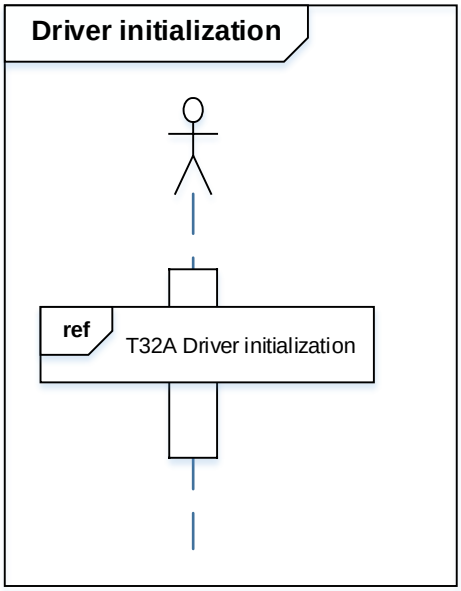


## 9.5. Operating Flow of Sample Program

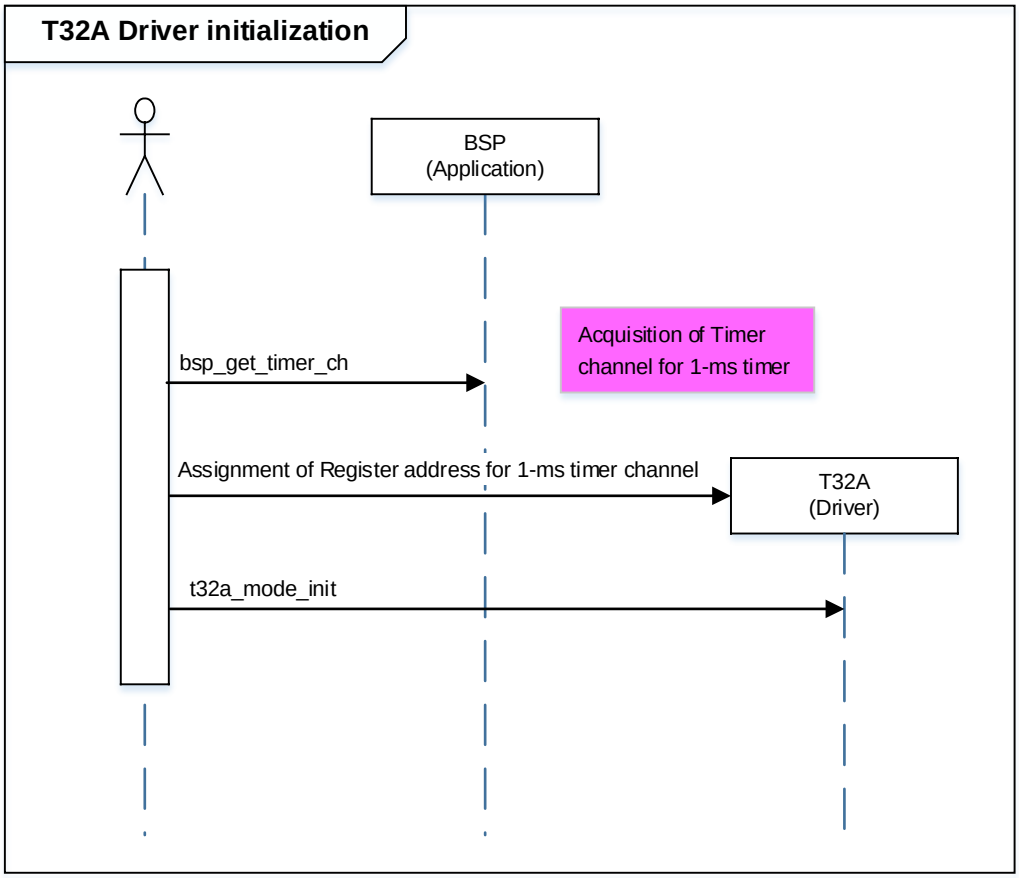
The basic operating flows of the sample program are shown in the following;

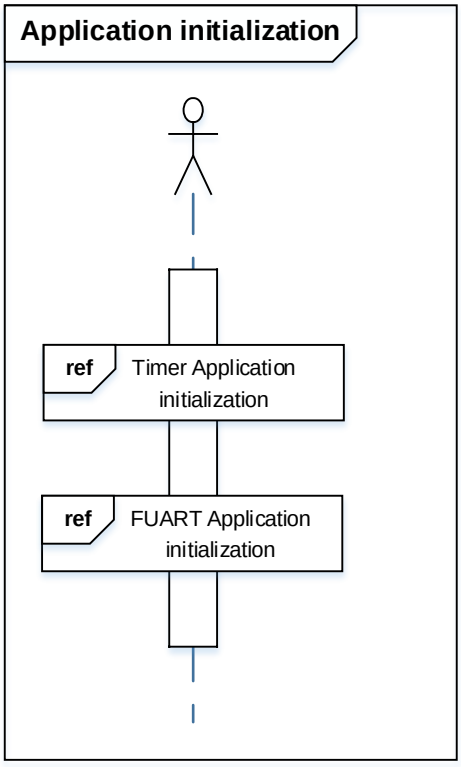


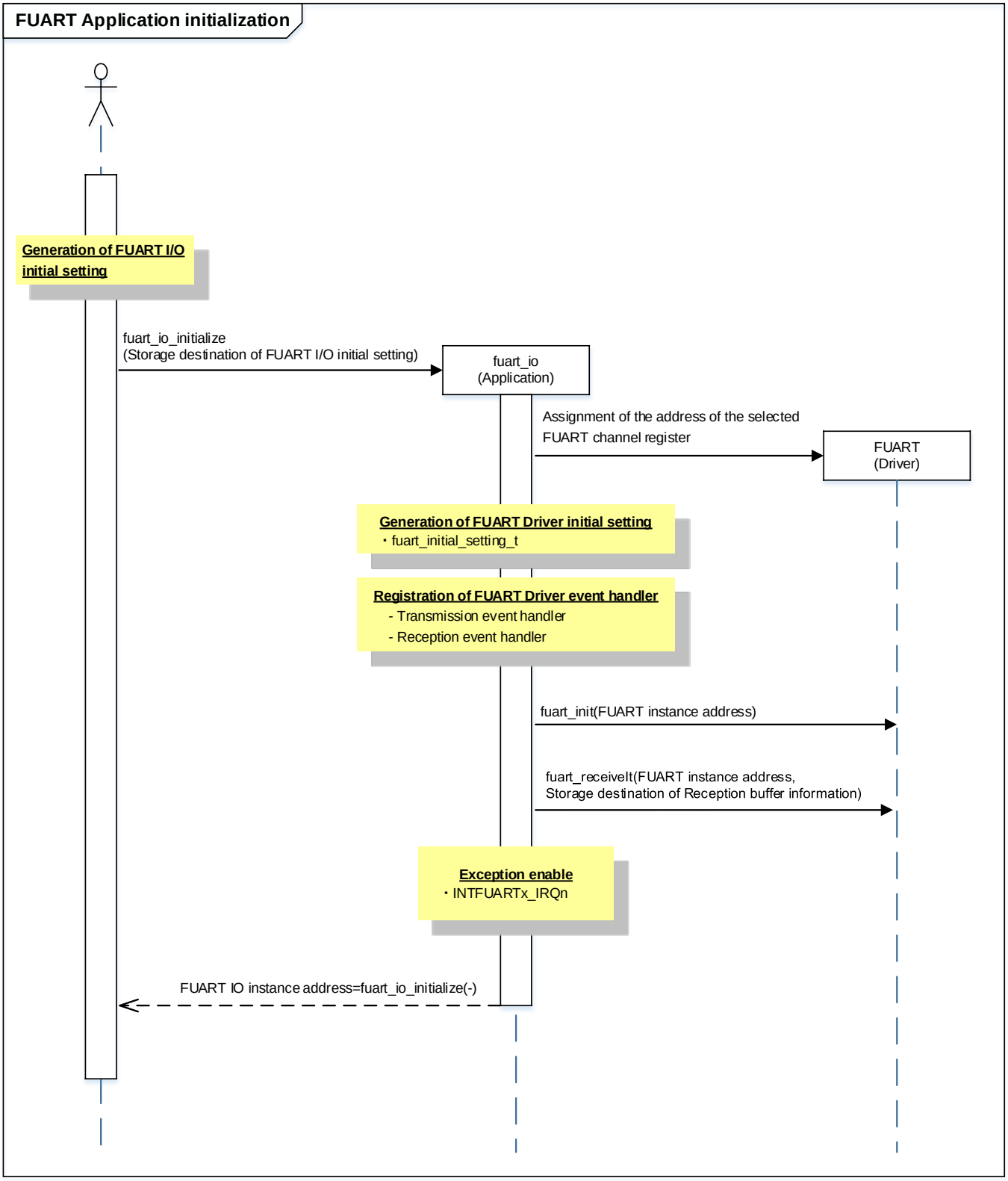

Creation and Initialization
ref
BSP initialization
RAM initialization
ref
Driver initialization
ref
Application initialization

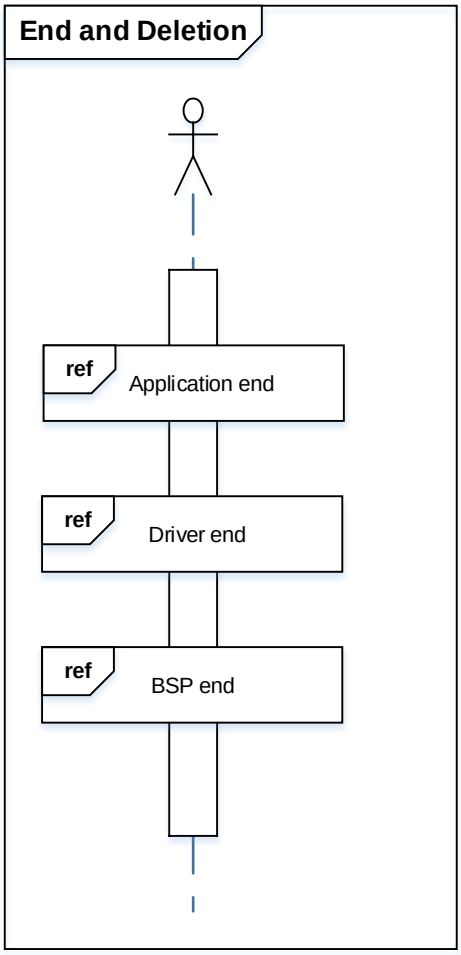

Driver initialization
ref
T32A Driver initialization

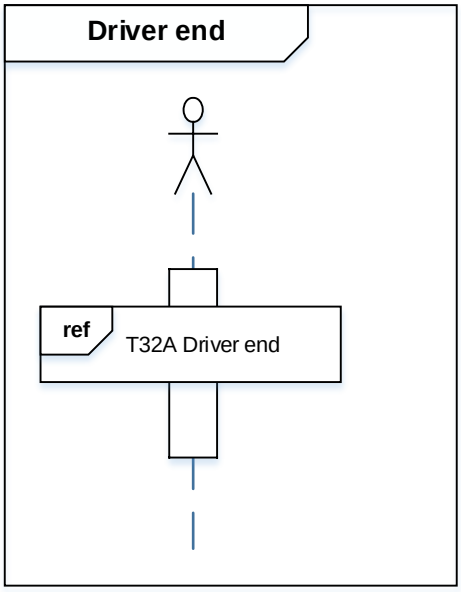
The 32-bit timer event counter of TMPM4G9 is running, but processing using timer count is not performed.

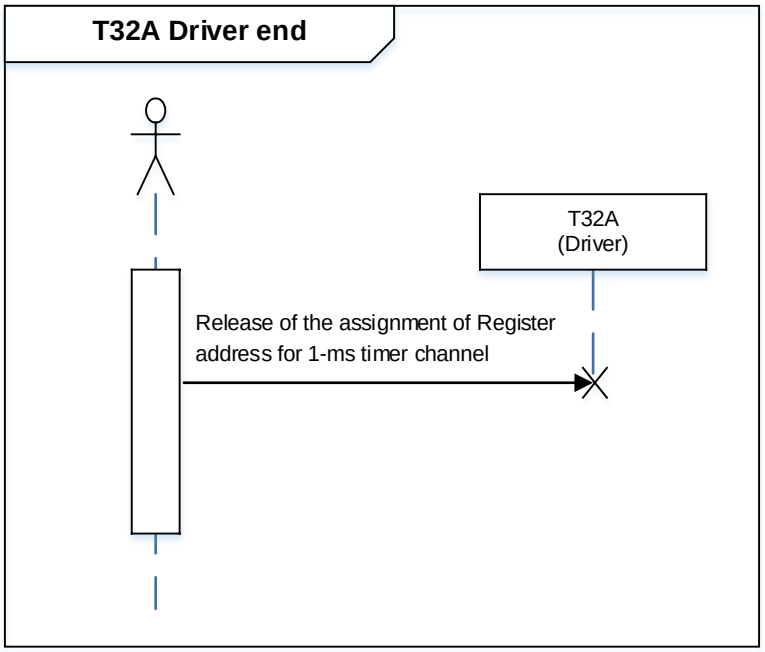


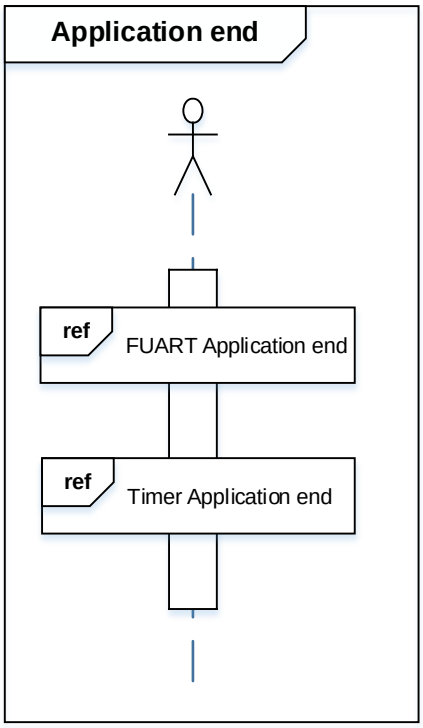



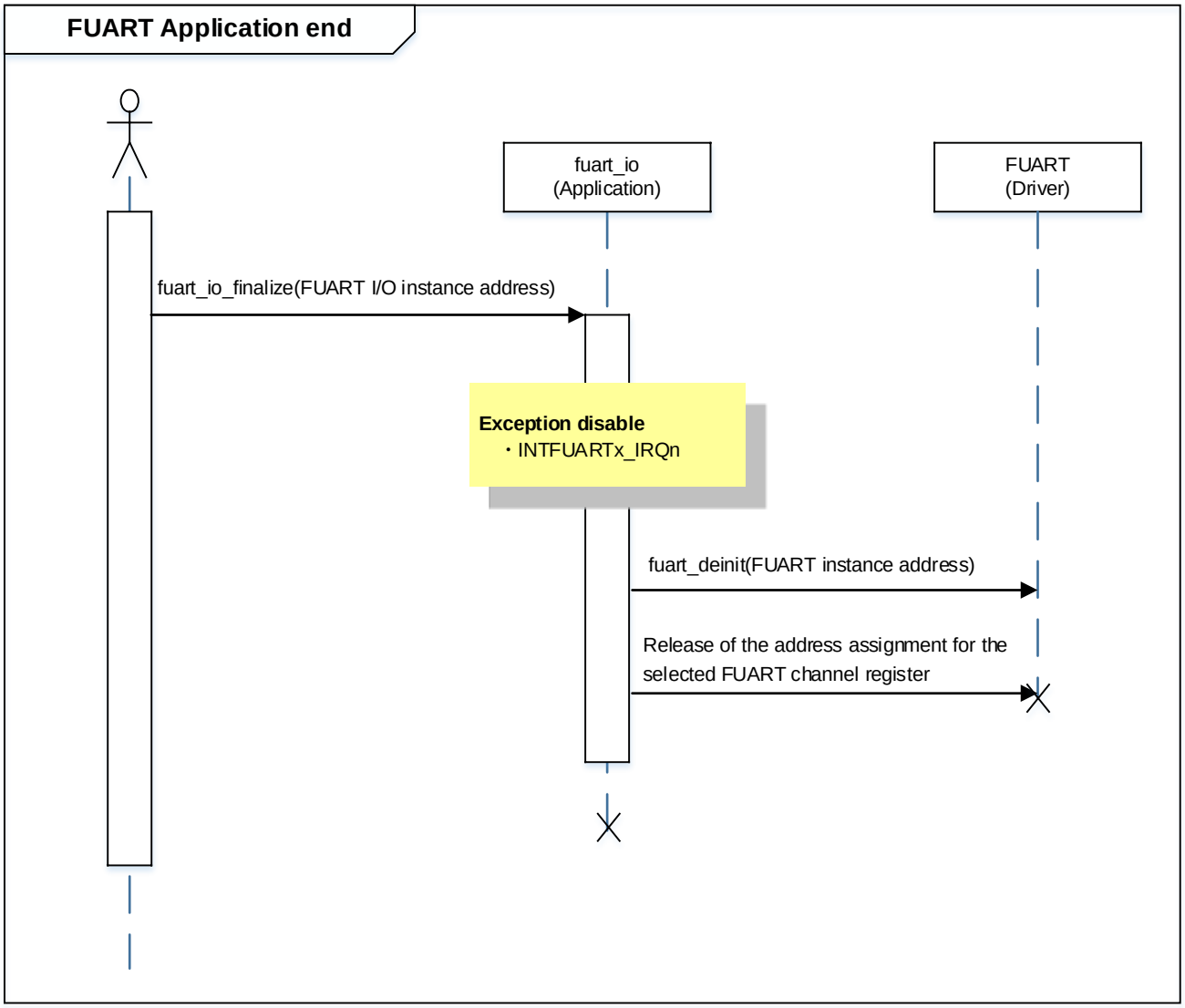

FUART Application initialization
Generation of FUART I/O initial setting
fuart_io_initialize
(Storage destination of FUART I/O initial setting)
fuart_io
(Application)
Assignment of the address of the selected FUART channel register
FUART
(Driver)
Generation of FUART Driver initial setting
・fuart_initial_setting_t
Registration of FUART Driver event handler
- Transmission event handler
- Reception event handler
fuart_init(FUART instance address)
fuart_receiveIt(FUART instance address,
Storage destination of Reception buffer information)
Exception enable
・INTFUARTx_IRQn
FUART IO instance address=fuart_io_initialize(-)

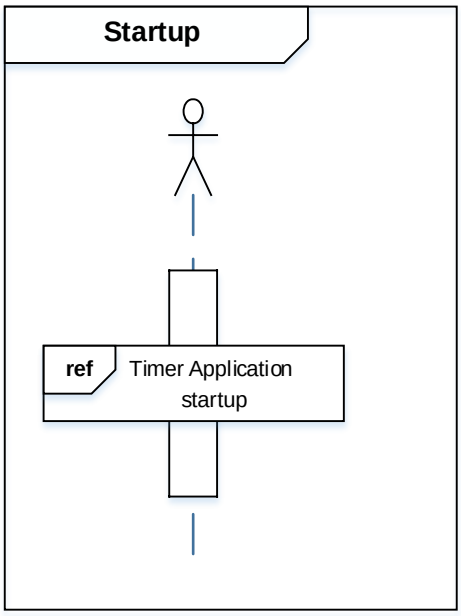

End and Deletion
ref
Application end
ref
Driver end
ref
BSP end

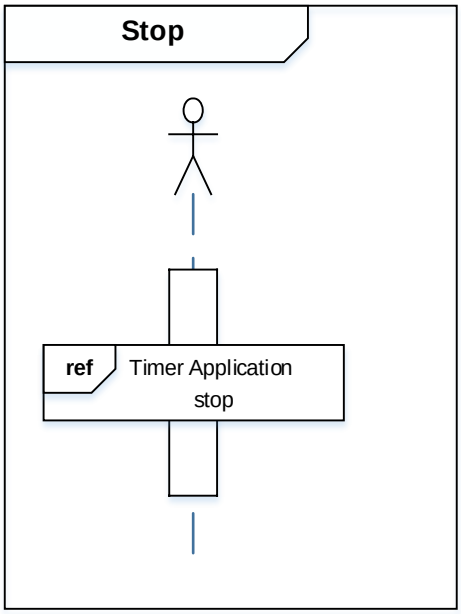

Driver end
ref
T32A Driver end

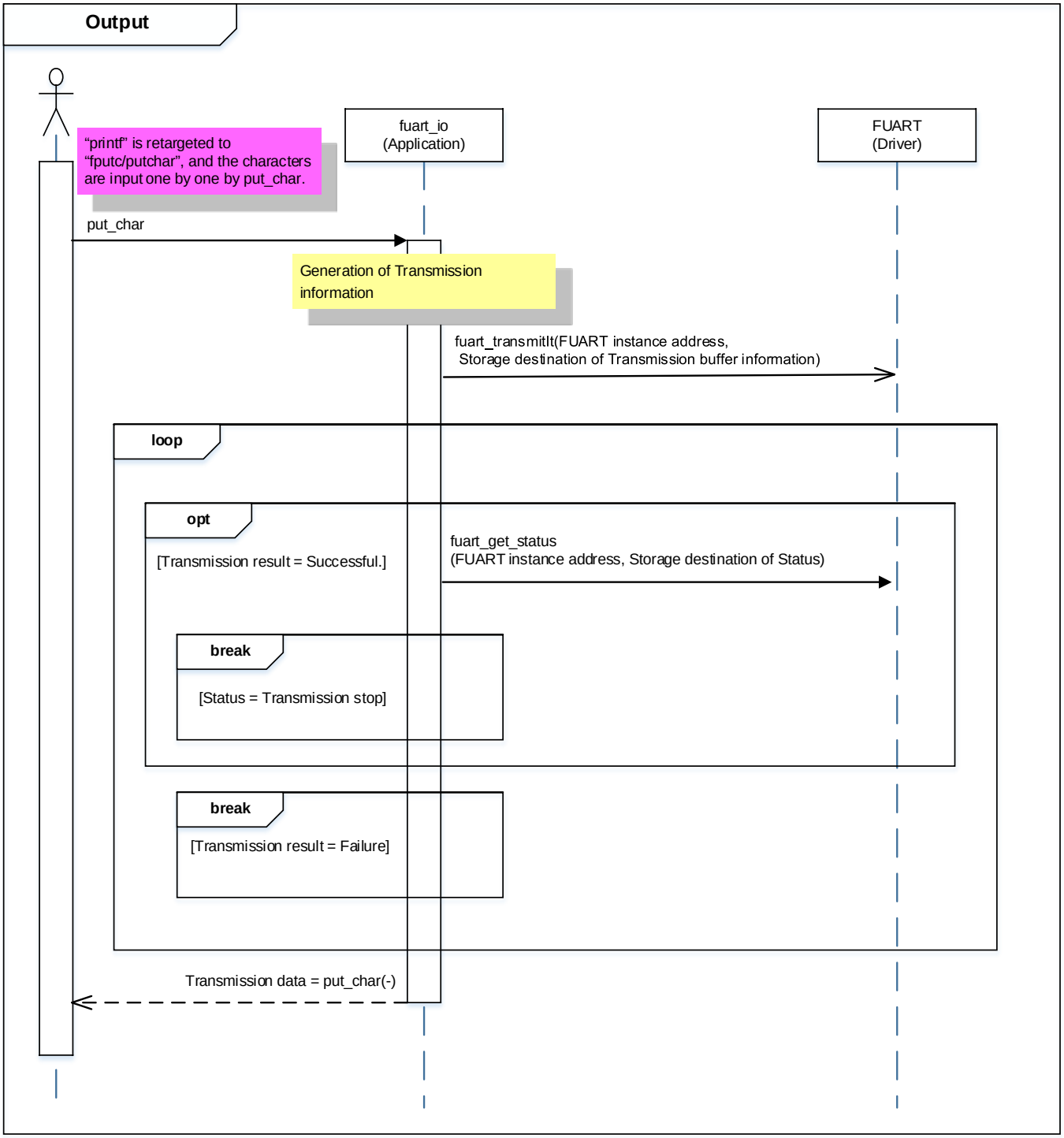

T32A Driver end
T32A
(Driver)
Release of the assignment of Register
address for 1-ms timer channel

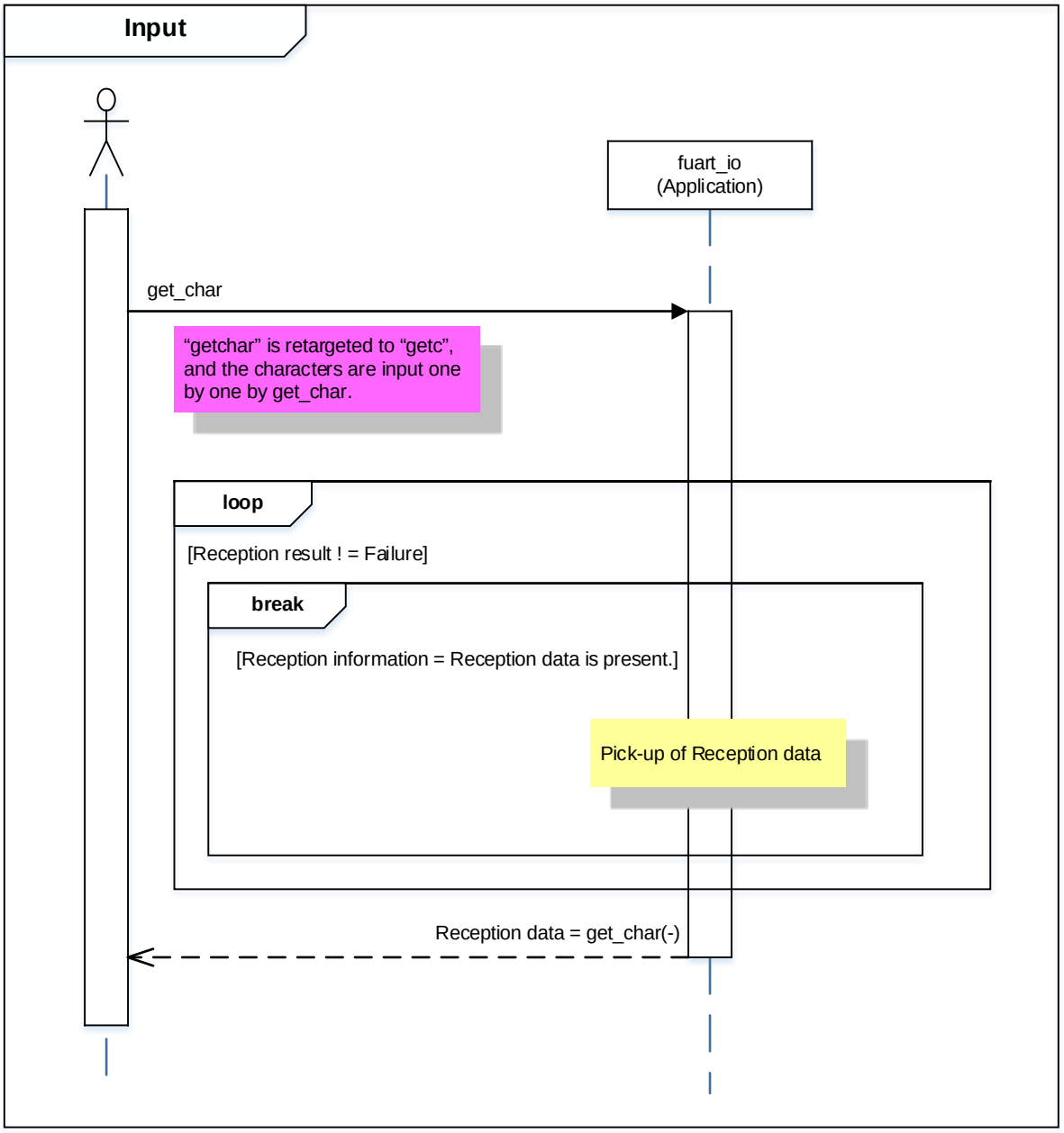

Application end
ref
FUART Application end
ref
Timer Application end

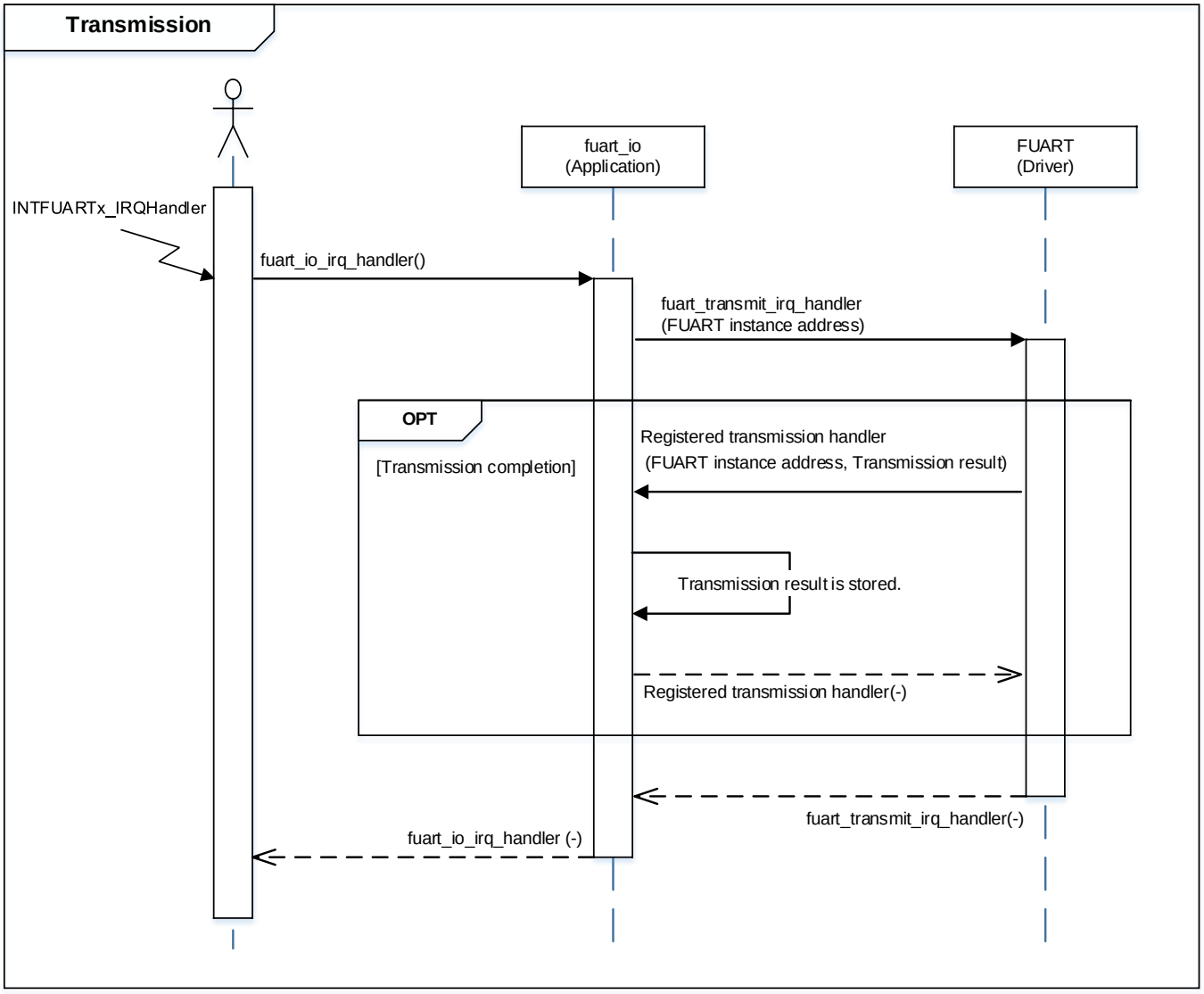

FUART Application end
fuart_io
(Application)
FUART
(Driver)
fuart_io_finalize(FUART I/O instance address)
Exception disable
・INTFUARTx_IRQn
fuart_deinit(FUART instance address)
Release of the address assignment for the
selected FUART channel register

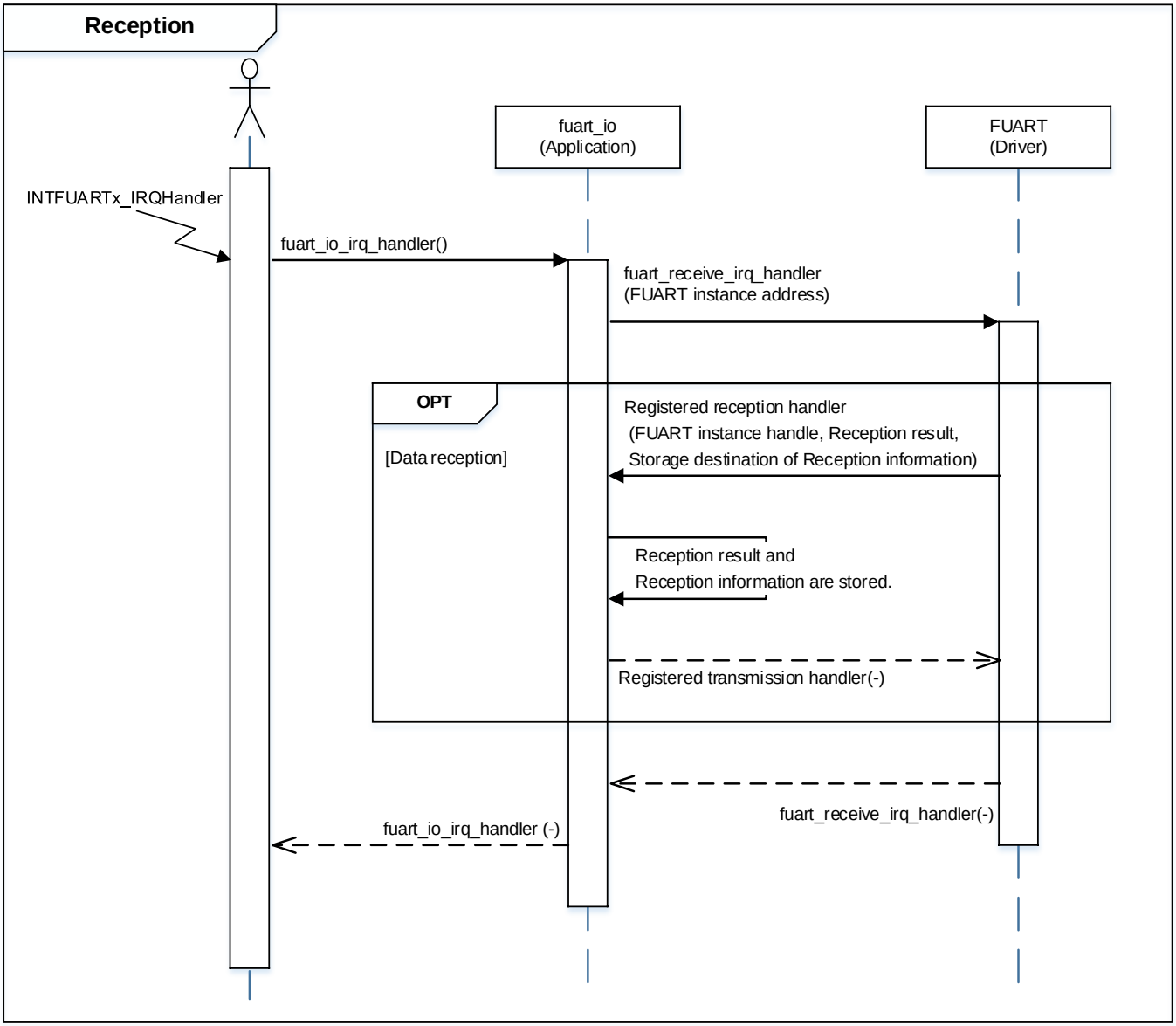

Startup
ref
Timer Application
startup


Stop
ref
Timer Application
stop


Output
"printf" is retargeted to "fputc/putchar", and the characters are input one by one by put_char.
fuart_io (Application)
FUART (Driver)
put_char
Generation of Transmission information
fuart_transmitIt(FUART instance address, Storage destination of Transmission buffer information)
loop
opt
[Transmission result = Successful.]
fuart_get_status (FUART instance address, Storage destination of Status)
break
[Status = Transmission stop]
break
[Transmission result = Failure]
Transmission data = put_char(-)


Input
fuart_io
(Application)
get_char
“getchar” is retargeted to “getc”, and the characters are input one by one by get_char.
loop
[Reception result ! = Failure]
break
[Reception information = Reception data is present.]
Pick-up of Reception data
Reception data = get_char(-)


Transmission
fuart_io
(Application)
FUART
(Driver)
INTFUARTx_IRQHandler
fuart_io_irq_handler()
fuart_transmit_irq_handler
(FUART instance address)
OPT
[Transmission completion]
Registered transmission handler
(FUART instance address, Transmission result)
Transmission result is stored.
Registered transmission handler(-)
fuart_transmit_irq_handler(-)
fuart_io_irq_handler (-)


Reception
fuart_io
(Application)
FUART
(Driver)
INTFUARTx_IRQHandler
fuart_io_irq_handler()
fuart_receive_irq_handler
(FUART instance address)
OPT
[Data reception]
Registered reception handler
(FUART instance handle, Reception result,
Storage destination of Reception information)
Reception result and
Reception information are stored.
Registered transmission handler(-)
fuart_receive_irq_handler(-)
fuart_io_irq_handler (-)

## 10. Precaution

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

## 11. Revision History

| Rev | Date | Description |
|---|---|---|
| 1.0 | 2018-11-30 | 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.**