

# RH850 Series

R01AN6029EJ0100  
Rev.1.0

## CAN Transmit Procedure (CAN FD Mode)

### Summary

This document describes an example of a procedure for transmitting CAN using the RH850 series. Please refer to the notes in the latest user's manual hardware edition for the settings of each register.

### Operation Confirmed Devices

This document applies to the RH850 series.

The variables described in the text are as follows.

Table 1-1 Target Devices and Variables

	Variable	Target MCU		
		RH850/E2x	RH850/E1M-S2	RH850/P1M-E
RS-CAN FD channel number	m	0~4	0~3	0~2
GAFLIDj, GAFLMj, GAFLP0j, GAFLP1j Register numbers	j	0~15	0~15	0~15
Transmit/Receive FIFO buffer number	k	0~14	0~11	0~8
Receive FIFO buffer number	x	0~7	0~7	0~7
Receive buffer number	q	0~79	0~63	0~47
Transmit buffer number	p	0~79	0~63	0~47
RAM test number	r	0~63	0~63	0~63
GAFLCFGi, GTINTSTSi register number	i	0	0	0
Number of each status register	y	0~2	0,1	0,1

The functions marked with "★" in the text are applicable to cases where 2 or more channels are installed.

In the text, CFD is omitted from the register names.

---

Contents

1. Transmit Function.....	3
2. Transmission Using Transmit Buffers .....	4
3. Transmission Using Transmit/Receive FIFO Buffers .....	18
4. Transmission Using Transmit Queue.....	26
5. Transmit History Buffer Function .....	30
6. CAN-related Interrupt Processing.....	36
7. Transmit Data Padding (only in CAN FD mode) .....	37
8. Precautions for Processing Flow .....	38
9. Appendix .....	39

## 1. Transmit Function

The functions that can be used when transmitting CAN messages are shown below. For details on each process, refer to the following chapters.

- Transmit using the transmit buffer
- Transmit using the transmit / receive FIFO buffer
- Transmit using the transmit queue
- Transmit history buffer function

## 2. Transmission Using Transmit Buffers

The transmit buffer is used to send data frames or remote frames.

There are 16 transmit buffers per channel, which can be used as a transmit buffer, for linking to the transmit/receive FIFO buffer (transmit mode, gateway mode★), or a transmit queue.

When using it as a transmit queue for linking to the transmit/receive FIFO buffer (transmit mode, gateway mode★), set the corresponding TMCp register to "H'00" and the TMIEp bit of the TMIEy register to "0" (interrupt disabled). In this case, the corresponding flags in the corresponding TMSTSp register, TMTRSTSy register, TMTARSTSy register, TMTTCSTSy register, and TMTARSTSy register do not change.

The functions of the transmit buffer are shown below. For the configuration settings for using the transmit buffer, refer to "CAN Configuration Application Note".

- • Message transmit function
- • Transmit abort function
- • One-shot transmit function (retransmit prohibition function)
- • Transmit buffer merge mode

## 2.1 Message Transmit Function

This is the function to send data frames or remote frames.

A message can be sent by setting a transmit request in the transmit buffer (the TMTR bit of the TMCp register is "1").

The transmission result can be confirmed by the TMTRF [1: 0] flag in the corresponding TMSTSp register. If the transmission is successful, the transmission is completed: no transmit abort request (the TMTRF [1: 0] flag is "B'10"), or the transmission is completed: transmit abort required (the TMTRF [1: 0] flag is "B'11"). About the transmission is completed: transmit abort request (the TMTRF [1: 0] flag is "B'11"), refer to "2.2 Transmit abort function")

For each transmit buffer, interrupt enable/disable when transmission is completed can be set with the TMIEp bit of the TMIECy register.

Figure 2-1 shows the operation of the transmit buffer.

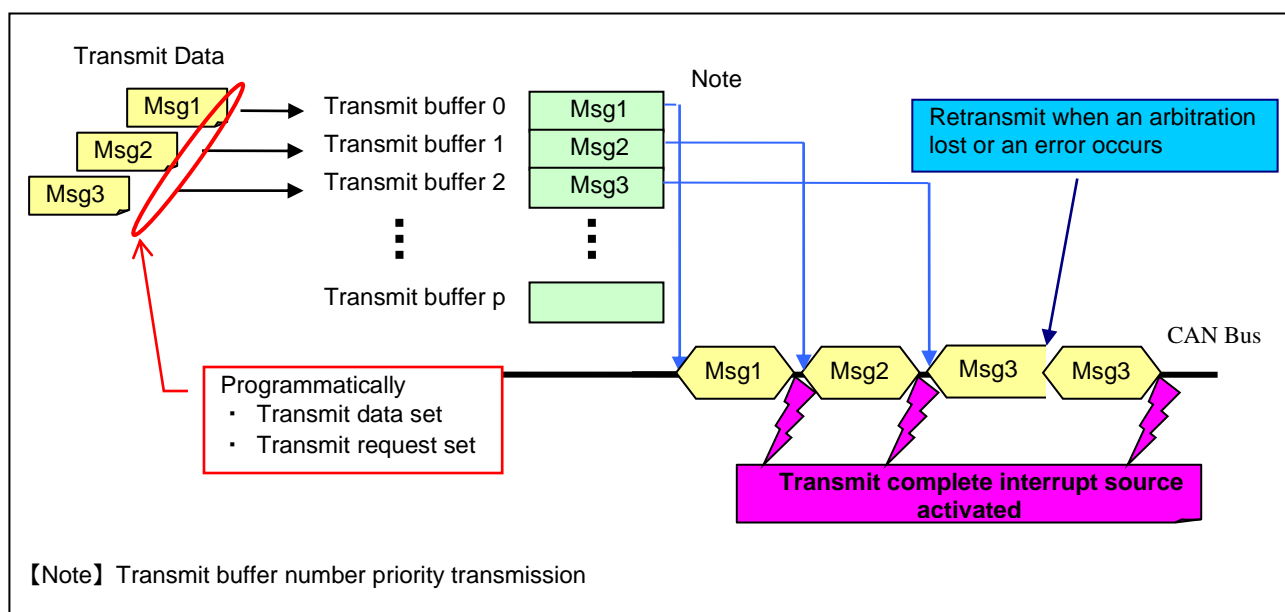
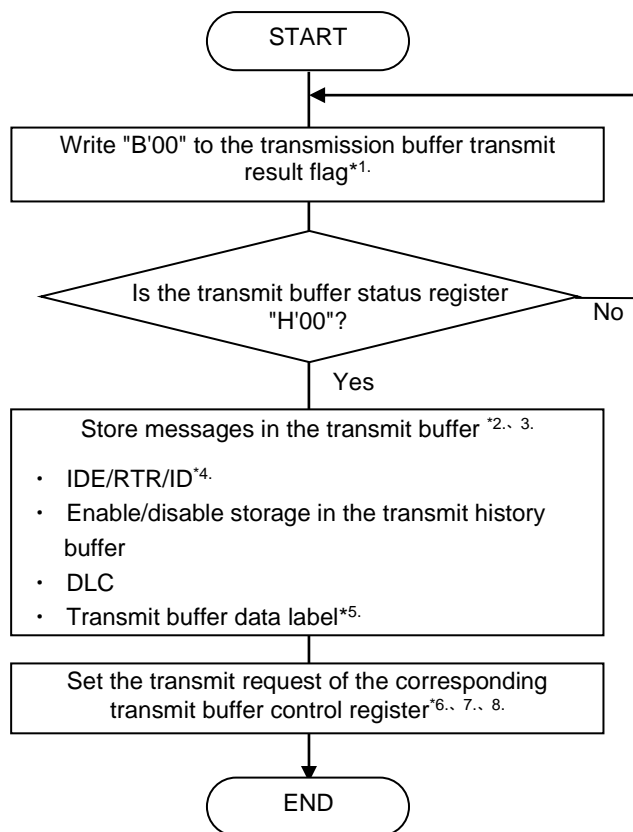


Figure 2-1 Transmit Buffer Operation (Transmit from channel 0)

### 2.1.1 Procedure for Sending Messages from Transmit Buffer

Figure 2-2 shows the procedure for sending messages from the transmit buffer.



- 【Note】**
1. For the TMTRF [1: 0] flag in the TMSTSp register, write "B'00" in channel communication mode or channel standby mode. Do not write any value other than "B'00".
  2. Rewrite the TMIDp register, TMPTRp register, and TMDfB\_p register when there is no transmit request in the corresponding transmit buffer (the TMTRM bit of the TMSTSp register is "0").
  3. Do not write to the TMIDP register, TMPTRp register, or TMDfB\_p register linked to the transmit/receive FIFO buffer.
  4. When setting the standard ID in the transmit ID (the TMID [28: 0] bit of the TMIDp register), set the ID in b10 to b0. Set b28-b11 to "0".
  5. This is valid only when the data is stored in the transmit history buffer (the THLDTE bit of the THLCCm register is "1", the THLE bit is "1", and the THLEN bit of the TMIDp register is "1").
  6. Rewrite the TMCp register in channel communication mode or channel standby mode.
  7. Set the TMCp register that satisfies the following conditions to "H'00".  
 Transmit buffer linked to transmit/receive FIFO buffer  
 Transmit buffer assigned to the transmit queue
  8. Set the transmit request (set the TMTR bit of the TMCp register to "1") when there is no transmit request (the TMTRF [1: 0] flag in the TMSTSp register is "B'00").

Figure 2-2 Procedure for Sending Messages from Transmit Buffer

## 2.2 Transmit Abort Function

If two or more nodes start transmitting at the same time, the node with the lower CAN ID priority will be an

arbitration lost (the message will be aborted in case of one-shot transmissions, and the message will be retained (retransmitted) in case of normal transmissions). A message will not be successfully sent unless it is sent without an arbitration lost or while the CAN bus is idle.

In such cases, there is a transmit abort function to discard the message being retransmitted. The transmit abort function is effective when you want to set a time limit for sending a message, or when you want to send an urgent high-priority message.

Figure 2-3 shows an application example of the transmit abort function.

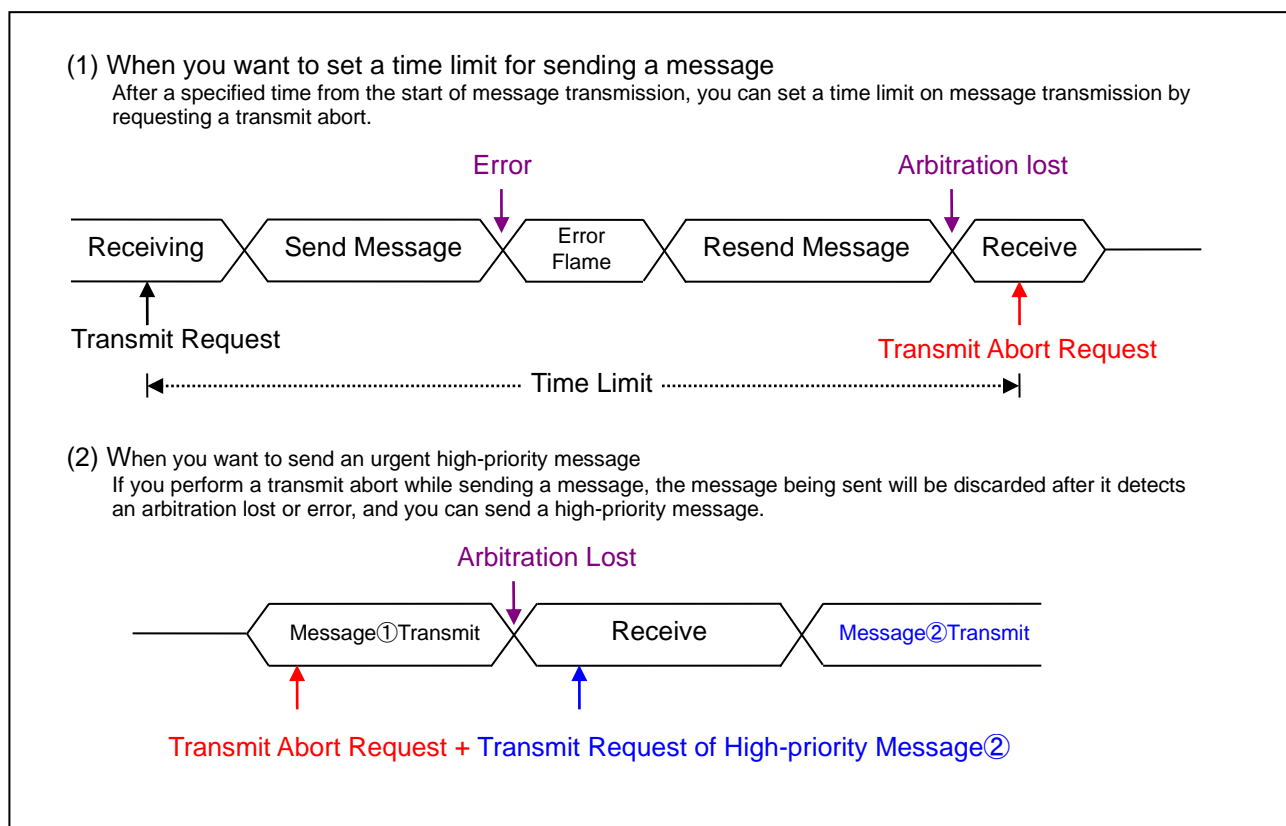


Figure 2-3 Application Example of Transmit Abort Function

When a transmit request is issued (the TMTRM bit of the TMSTSp register is “1”) and a transmit abort request is issued to the transmit buffer (the TMTAR bit of the TMCp register is “1”), the transmit request is cancelled.

The timing of the actual abort after issuing the transmit abort request is shown below.

The message being sent or the message determined to be sent next by the priority determination of transmission

- When an arbitration lost occurs
- When an error occurs

Messages other than the above

- When a transmit abort request is issued

When the transmit abort is completed, the TMTRF [1: 0] flag in the TMSTSp register becomes “B’01” and the transmit request is canceled (the TMTRM bit becomes “0”).

After issuing a transmit abort request to the message being sent or the message determined to be sent next by the priority determination of transmission, if the transmission is completed successfully without an arbitration lost or error, the transmission is completed: There is an abort request (TMTRF [1: 0] flag is “B’11”).

Figure 2-4 shows the operation during transmit abort.

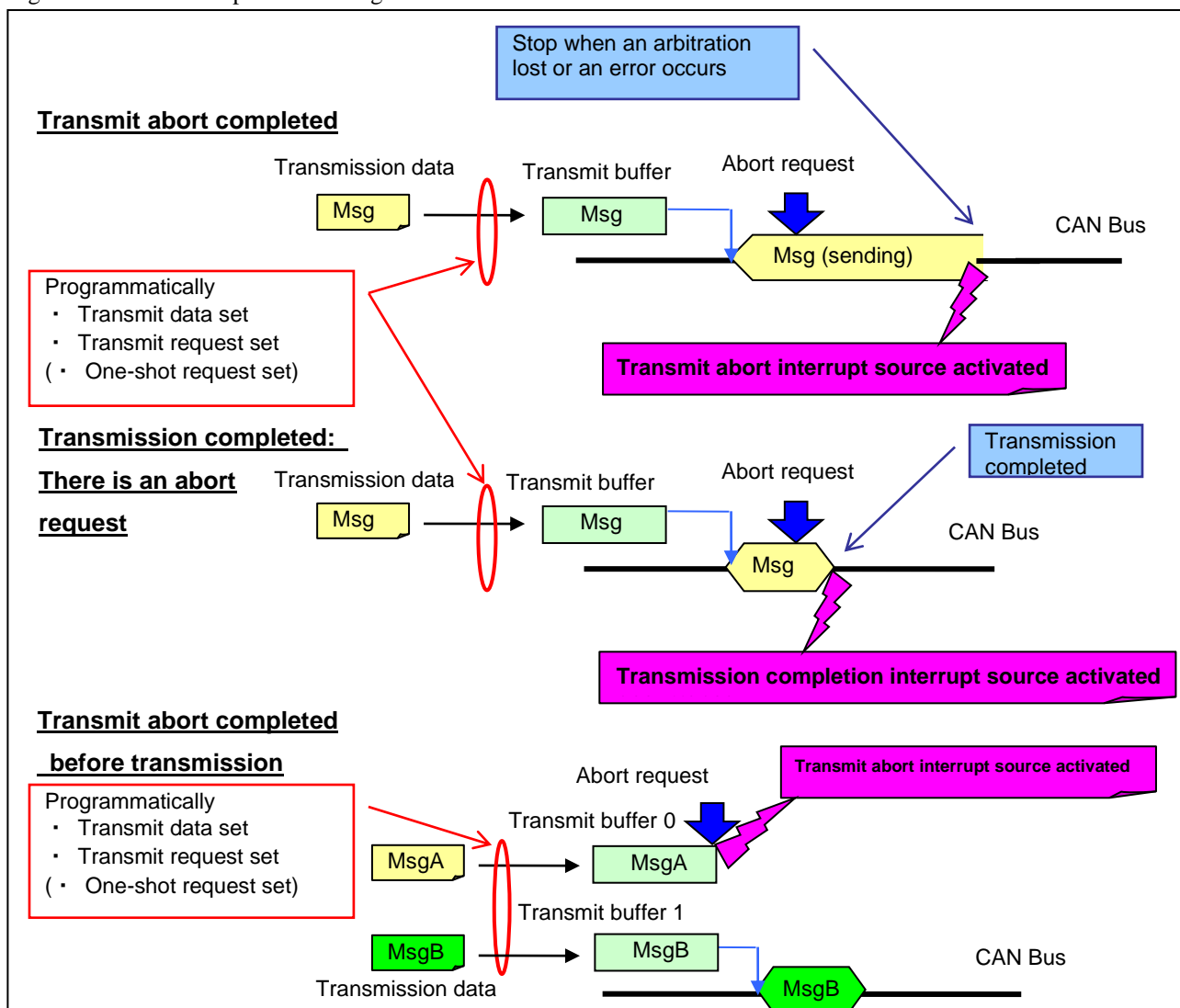
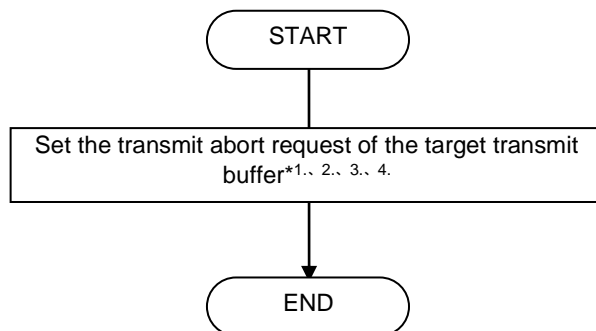


Figure 2-4 Operation during Transmit Abort



## 2.2.1 Transmit Abort Procedure

Figure 2-5 shows the procedure for transmit abort.



- 【Note】
1. Rewrite the TMCp register in channel communication mode or channel standby mode.
  2. Set the transmit TMCp register that satisfies the following conditions to "H'00".  
Transmit buffer linked to the transmit/receive FIFO buffer  
Transmit buffer assigned to the transmit queue
  3. When there is a transmit request in the transmit buffer (the TMTR bit of the TMCp register is "1"), a transmit abort request can be issued (the TMTAR bit of the TMCp register is "1").
  4. Depending on the timing, the transmission result will be transmission is completed: There is a transmit abort (the TMTRF[1:0] flag in the TMSTSp register is "B'11"), so if the transmission result is to be determined, please check not only if the transmission is complete (TMTRF [1:0] flag is "B'01") but also if the transmission is completed: There is a transmit abort. Please refer to "2.4.3 Processing procedure after transmission completion and transmit abort completion" for the procedure for confirming transmission completion and transmit abort completion.

Figure 2-5 Transmit Abort Procedure

### 2.3 One-Shot Transmission Function

When the TMOM bit of the RSCFDn(CFD)TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration lost or an error occurs, retransmission is not performed.

The result of one-shot transmission can be confirmed by the TMTRF [1: 0] flag in the TMSTSp register. If the one-shot transmission is successful, the transmit buffer transmission result status is transmission complete: no transmit abort request (TMTRF [1: 0] flag is "B'10") or transmission complete: There is a transmit abort request (TMTRF [1: 0] flag will be "B'11"). If an arbitration lost or an error occurs, abort is completed (TMTRF [1: 0] flag is "B'01"). (Refer to "2.2 Transmit Abort Function" for transmission completion: There is a transmit abort request (TMTRF [1: 0] flag is "B'11")). Figure 2-6 shows the operation of one-shot transmission.

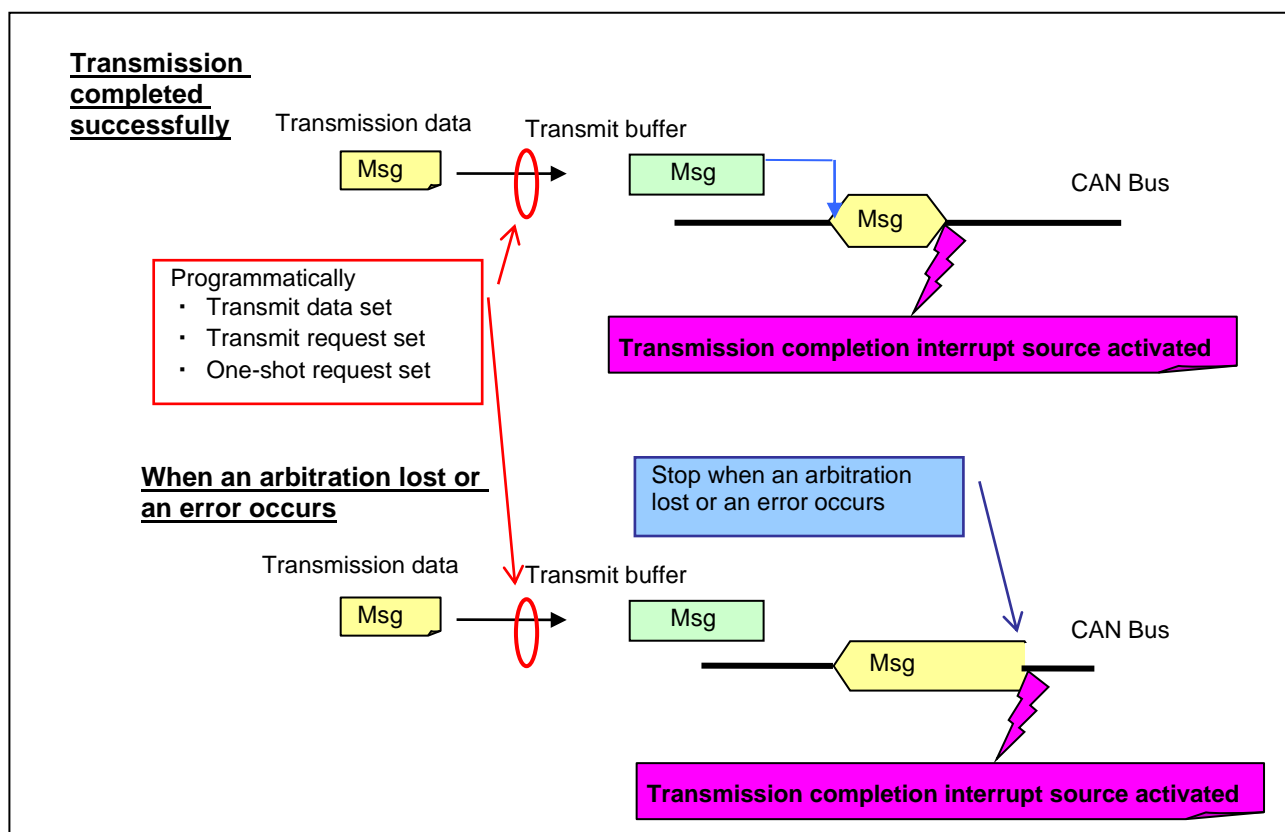
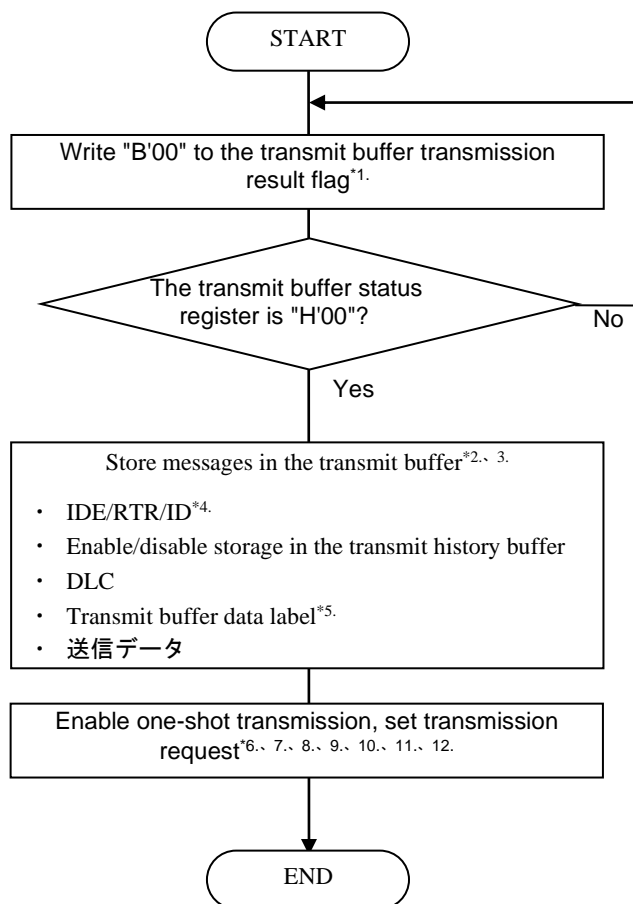


Figure 2-6 Operation of One-shot Transmission

## 2.3.1 One-shot Transmission Procedure

Figure 2-7 shows the one-shot transmission procedure.



- 【Note】
- For the TMTRF [1: 0] flag in the TMSTSp register, write "B'00" in channel communication mode or channel standby mode. Do not write any value other than "B'00".
  - Rewrite the TMIDp register, TMPTRp register, and TMDfB\_p register when there is no transmit request in the corresponding transmit buffer (the TMTRM bit of the TMSTSp register is "0").
  - Do not write the TMIDP register, TMPTRp register, or TMDfB\_p register linked to the transmit/receive FIFO buffer.
  - When setting the standard ID for the transmit ID (the TMID [28: 0] bit of the TMIDp register), set the ID in b10-b0. Set b28-b11 to "0".
  - This is enabled only when the data is stored in the transmit history buffer (the THLDTE bit of the THLCCm register is "1", THLE bit is "1", and the THLEN bit of the TMIDP register is "1").
  - Rewrite the TMCp register in channel communication mode or channel standby mode.
  - Set the TMCp register that satisfies the following conditions to "H'00".  
Transmit buffer linked to the transmit/receive FIFO buffer  
Transmit buffer assigned to the transmit queue
  - A transmit request can be set (the TMTR bit of the TMCp register is set to "1") when there is no transmit request (the TMTRF [1: 0] flag in the TMSTSp register is "B'00").
  - Even if the transmission fails, the retransmission specified in the CAN protocol is not performed.
  - Enable one-shot transmission (the TMOM bit of the TMCp register "1") when there is no transmission request in the transmit buffer (the TMTRM bit of the TMSTSp register is "0").
  - To enable one-shot transmission, set it at the same time as the transmit request (the TMTR bit and the TMOM bit are set to "1" at the same time).

12. Depending on the timing, the transmission result will be transmission is completed: There is a transmit abort (the TMTRF[1:0] flag in the TMSTSp register is "B'11"), so if the transmission result is to be determined, please check not only if the transmission is complete (TMTRF [1:0] flag is "B'01") but also if the transmission is completed: There is a transmit abort. Please refer to "2.4.3 Processing procedure after transmission completion and transmit abort completion" for the procedure for confirming transmission completion and transmit abort completion.

Figure 2-7 One-shot Transmission Procedure

## 2.4 Transmit Buffer Interrupt Processing

### 2.4.1 Transmit Complete Interrupt Processing

If the transmit complete interrupt is enabled, the CANm transmit interrupt is generated when the transmission is completed. Whether to enable or disable the transmit complete interrupt can be set for each transmit buffer by the TMIEp bit of the TMIECy register.

The CANm transmit interrupts share the following sources. When using multiple interrupt sources, determine the sources within the interrupt as necessary.

The source of the CANm transmit interrupt can also be confirmed in the GTINTSTS0 register.

- CANm transmit complete interrupt
- CANm transmit abort interrupt
- CANm transmit/receive FIFO transmit complete interrupt
- CANm transmit queue interrupt
- CANm transmit history interrupt

CANi送信割り込みを発生させるためには、**エラー! 参照元が見つかりません。**にある対応する割り込み許可ビットが"1"である割り込み要求フラグをすべて"0"にする必要があります。

### 2.4.2 Transmit Abort Complete Interrupt Processing

If the transmit abort complete interrupt is enabled, the CANm transmit interrupt is generated when the transmit abort is completed. Whether to enable or disable the transmit abort complete interrupt can be set for each channel with the TAIE bit of the CmCTR register. However, if the transmission is completed: There is an abort request (TMTRF [1: 0] flag is "B'11"), the transmit abort complete interrupt is not generated and the transmit complete interrupt is generated.

The CANm transmit interrupts share the following sources. When using multiple interrupt sources, determine the sources within the interrupt as necessary.

The sources of the CANm transmit interrupt can also be confirmed in the GTINTSTS0 register.

- CANm transmit complete interrupt
- CANm transmit abort interrupt
- CANm transmit/receive FIFO transmit complete interrupt
- CANm transmit queue interrupt
- CANm transmit history interrupt

CANi送信割り込みを発生させるためには、**エラー! 参照元が見つかりません。**にある対応する割り込み許可ビットが"1"である割り込み要求フラグをすべて"0"にする必要があります。

## 2.4.3 Processing Procedure after Transmission Completion and Transmit Abort Completion

Figure 2-8, Figure 2-9, and Figure 2-10 show the processing procedure after the transmission is completed and the transmit abort is completed.

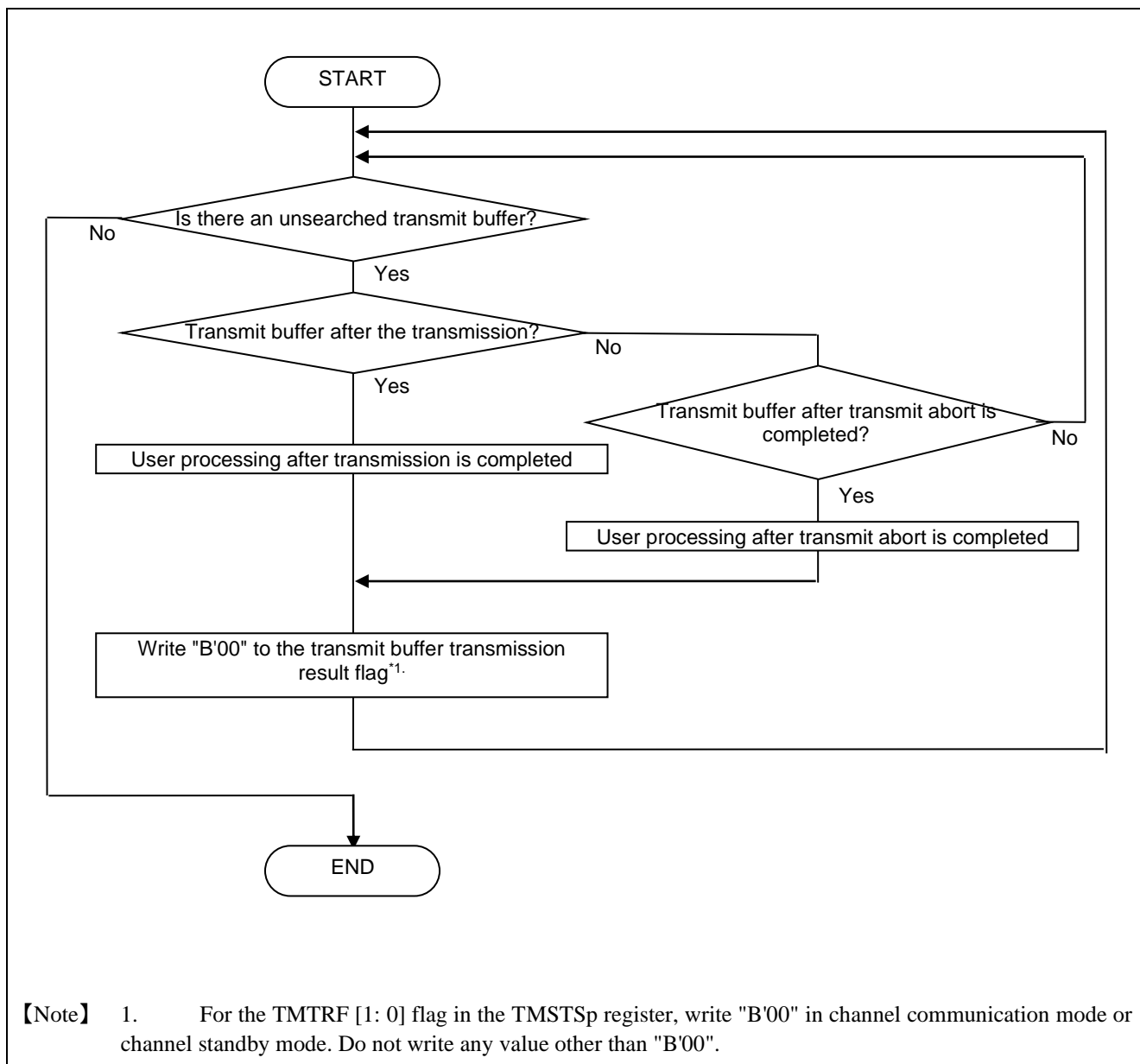
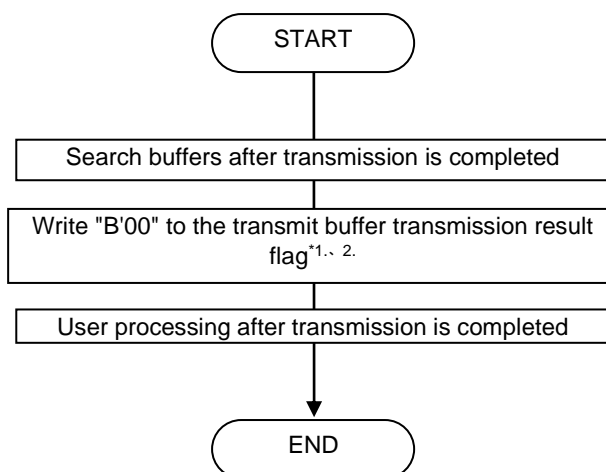
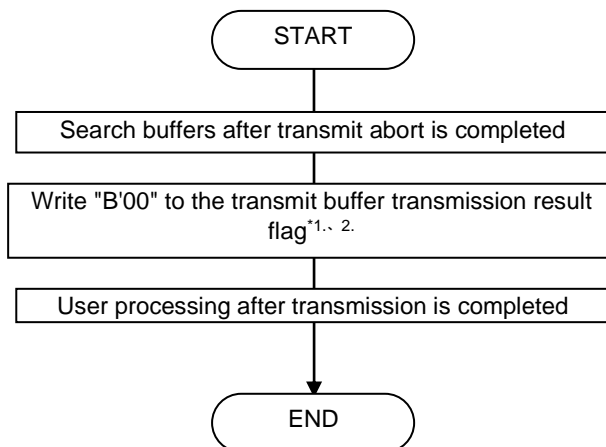


Figure 2-8 Processing Procedure after Transmission Completion and Transmit Abort Completion  
(no interrupt use)



- 【Note】
1. For the TMTRF [1: 0] flag in the TMSTSp register, write "B'00" in channel communication mode or channel standby mode. Do not write any value other than "B'00".
  2. See Figure 6-1 for the interrupt source flag processing procedure when using interrupts.

Figure 2-9 Processing Procedure after Transmission Completion (when using interrupt)



- 【Note】
1. For the TMTRF [1: 0] flag in the TMSTSp register, write "B'00" in channel communication mode or channel standby mode. Do not write any value other than "B'00".
  2. See Figure 6-1 for the interrupt source flag processing procedure when using interrupts.

Figure 2-10 Processing Procedure after Transmit Abort Completion (when using interrupt)

## 2.5 Transmit Buffer Merge Mode (Only in CAN FD Mode)

Transmit buffers can transmit messages with a payload length of 20 bytes, but can transmit messages with a payload length of up to 64 bytes by merging three transmit buffers in transmit buffer merge mode.

Setting the TMME bit to 1 in the CmFDCFG register enables transmit buffer merge mode. In this mode, six buffers per channel become a merge area and two sets of transmit buffers  $(16 \times m) + 0$  to  $(16 \times m) + 2$  and transmit buffers  $(16 \times m) + 3$  to  $(16 \times m) + 5$  are merged. A transmission request is made by the first transmit buffer, and subsequent two buffers are used as a payload storage area.

Set the control register (TMCp register) of the transmit buffer corresponding to the transmit buffer allocated as the payload storage area to "00H". Also, set the enable bit of the corresponding interrupt enable register (TMIECy register) to "0" (interrupt disabled).

While transmit buffer merge mode is enabled, do not link the transmit/receive FIFO buffer to six merged buffers or allocate it to the transmit queue.

Table 2-1 shows an example of data storage when transmitting 64 bytes in transmit buffer merge mode using transmit buffer 0.

Table 2-1 Example of Data Storage Using Transmit Buffer 0

Register Name	Data to Write
TMID0	Transmit buffer 0 ID data, transmit history data store enable bit, RTR bit, and IDE bit
TMPTR0	Transmit buffer 0 label data and DLC data
TMFDCTR0	Transmit buffer 0 ESI bit, BRS bit, and FDF bit
TMDF0_0~TMDF4_0	Transmit buffer 0 data bytes 0, 1, 2, and 3 to transmit buffer 0 data bytes 16, 17, 18, and 19
TMID1	Transmit buffer 0 data bytes 20, 21, 22, and 23
TMPTR1	Transmit buffer 0 data bytes 24, 25, 26, and 27
TMFDCTR1	Transmit buffer 0 data bytes 28, 29, 30, and 31
TMDF0_1~TMDF4_1	Transmit buffer 0 data bytes 32, 33, 34, and 35 to transmit buffer 0 data bytes 48, 49, 50, and 51
TMID2	Transmit buffer 0 data bytes 52, 53, 54, and 55
TMPTR2	Transmit buffer 0 data bytes 56, 57, 58, and 59
TMFDCTR2	Transmit buffer 0 data bytes 60, 61, 62, and 63
TMDF0_2~TMDF4_2	Not used



Figure 2-11 shows the procedure for sending a message from the transmit buffer using the transmit buffer merge mode.

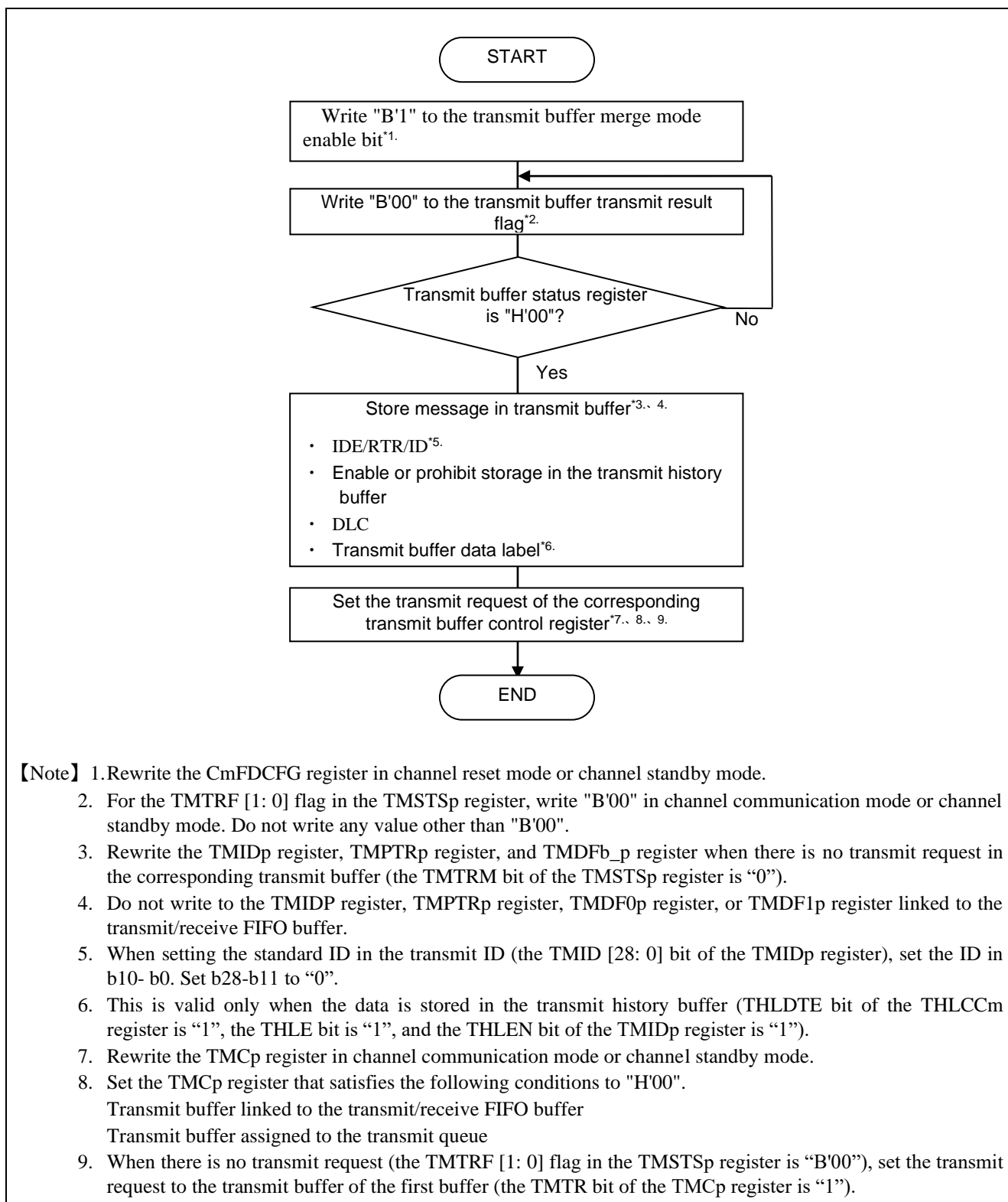


Figure 2-11 Message Transmission Procedure in Transmit Buffer Merge Mode

### 3. Transmission Using Transmit/Receive FIFO Buffers

Data frames or remote frames are transmitted with the transmit/receive FIFO buffer.

There are 3 transmit / receive FIFO buffers per channel and can store up to 128 messages. It will be sent in order from the first stored message.

The transmit/receive FIFO buffer can be used in either receive mode, transmit mode, or gateway mode★ (only transmit mode is described in this chapter).

The transmit/receive FIFO buffer is linked to the transmit buffer (selected by the CFTML [3: 0] bit of the CFCCk register). When the transmit/receive FIFO buffer is used (the CFE bit of the CFCCk register is set to “1”), the priority determination of transmission is performed, and the priority determination is performed only for the next message to be sent in the transmit/receive FIFO buffer.

The transmission function of the transmit/receive FIFO buffer is shown below. For the configuration settings for using the transmit/receive FIFO buffer, refer to "CAN Configuration Application Note".

- • Message transmit function
- • Transmit abort function
- • Interval transmission function

### 3.1 Message Transmit Function

This is the function to transmit data frames or remote frames. The messages stored in the transmit/receive FIFO buffer are transmitted in the order in which they were stored.

Figure 3-1 shows the transmit mode operation of the transmit/receive FIFO buffer.

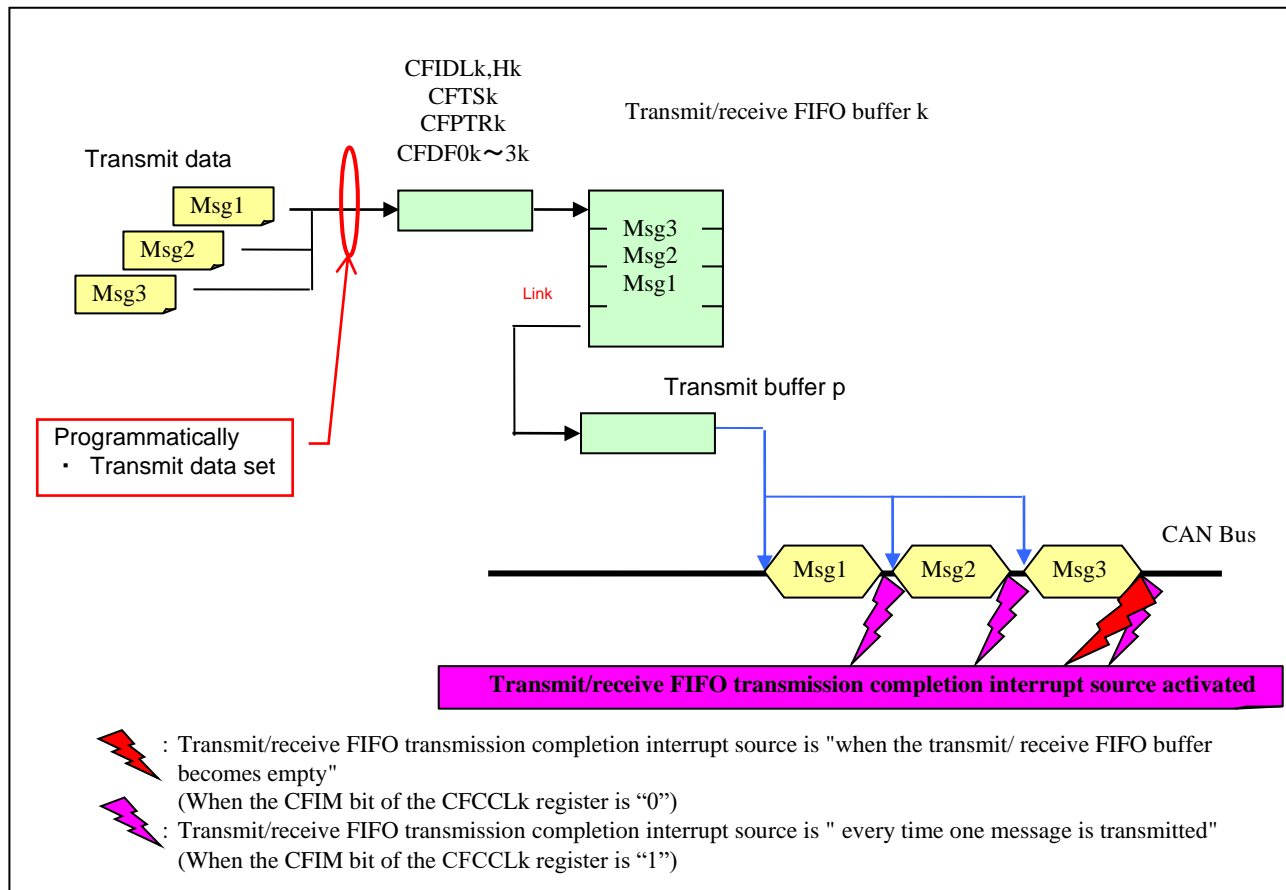
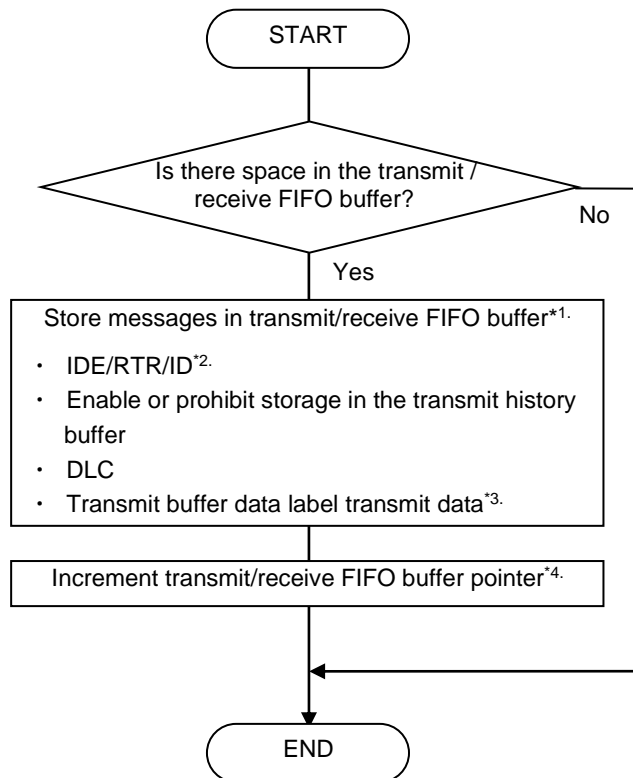


Figure 3-1 Operation of Transmit/receive FIFO Buffer (Transmit mode)

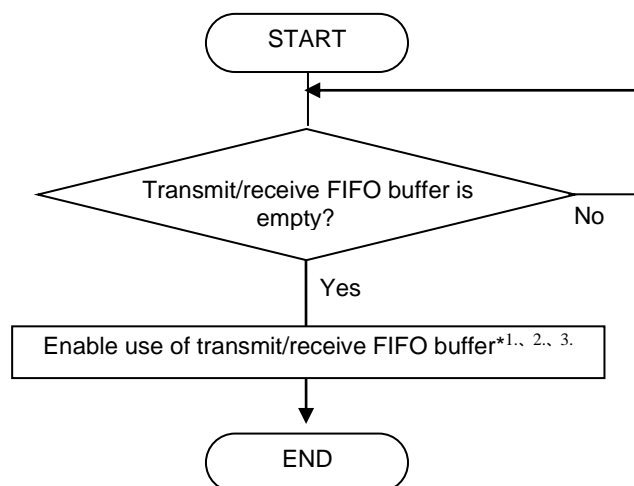
## 3.1.1 Message Transmission Procedure from Transmit/receive FIFO

Figure 3-2 shows the procedure for sending a message from the transmit/receive FIFO buffer, and Figure 3-3 and Figure 3-4 show the procedure for enabling/prohibiting the transmit/receive FIFO buffer.



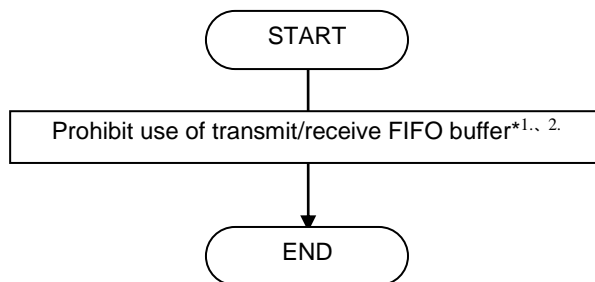
- 【Note】
- Only when the transmit/receive FIFO buffer is in transmit mode (the CFM [1: 0] bit of the CFCCk register is "B'01"), the CFIDk register, CFPTRk register, and CFDFd\_k register can be written.
  - When setting the standard ID in the transmission ID (the CFID [28: 0] bit of the CFIDk register) in the transmit mode, set the ID in b10-b0. Set b28-b11 to "0".
  - This is valid only when the transmit mode is set and the data is stored in the transmit history buffer (THLE bit of the THLCCm register is "1" and the THLEN bit of the CFIDk register is "1").
  - Increment the transmit/receive FIFO buffer pointer if the following conditions are satisfied (write "H'FF" in the CFPC bit of the CFPCTRk register).
    - In transmit mode
    - When using transmit/receive FIFO (the CFE bit of the CFCCk register is "1")
    - After writing a transmit message to the transmit/receive FIFO buffer
    - The transmit/receive FIFO is not full (the CFFLL flag in the CFSTSk register is "0")

Figure 3-2 Message Transmission Procedure from Transmit/receive FIFO Buffer



- 【Note】
1. When using the transmit mode, if the transmit/receive FIFO buffer is disabled, the transmit/receive FIFO buffer will be empty after transmission completion, CAN bus error detection, or arbitration lost if the message is being transmitted or is determined to be the next transmission.
  2. When using the transmit mode, rewrite the transmit/receive FIFO buffer enable/disable (the CFE bit of the CFCCk register) in channel communication mode or channel standby mode.
  3. Make the configuration settings for using the transmit/receive FIFO buffer, and then enable the use of the transmit/receive FIFO buffer (CFE bit is "1").

Figure 3-3 Procedure for Enabling Use of Transmit/receive FIFO Buffer



- 【Note】
1. When using the transmit mode, rewrite the enable/disable (the CFE bit of the CFCCk register) of the transmit / receive FIFO buffer in the channel communication mode or channel standby mode.
  2. Even if the use of the transmit/receive FIFO buffer is prohibited (the CFE bit is "0") while an interrupt request is generated (the CFTXIF flag in the CFSTSk register is "1"), the CFTXIF flag is not automatically set to "0". Set the interrupt request flag to "0" with the program.

Figure 3-4 Procedure for Prohibiting Use of Transmit/receive FIFO Buffer

### 3.2 Transmit Abort Function

By disabling the use of the transmit/receive FIFO buffer, the messages in the transmit/receive FIFO buffer can be aborted. Aborting the transmit/receive FIFO buffer aborts all messages in the transmit/receive FIFO buffer, not just the messages being sent (the transmit/receive FIFO buffer becomes empty (the CFEMP flag in the CFSTSk register is “1”). You can check the completion of aborting the transmit/receive FIFO buffer by checking that the transmit/receive FIFO buffer is empty.

No interrupt is generated when the transmit abort of the transmit/receive FIFO buffer is completed. However, if aborted during transmission, a transmit/receive FIFO transmission completion interrupt may occur due to transmission completion. For details, refer to "Figure 2-3 Application example of transmission abort function".

For the transmit abort procedure of the transmit/receive FIFO buffer, refer to "Figure 3-4 Procedure for prohibiting use of transmit/receive FIFO buffer ".

### 3.3 Interval Transmission Function

To transmit messages continuously from the same transmit/receive FIFO buffer that is set to transmit mode or gateway mode<sup>★</sup>, message transmission interval time can be set.

When the use of the transmit/receive FIFO buffer is enabled (the CFE bit of the CFCCk register is “1”), the interval timer starts counting after the first message is successfully transmitted from the transmit/receive FIFO buffer ((After the 7th bit of EOF in the CAN protocol). After that, when the interval time has elapsed, the next message will be sent and the interval timer will be reset.

The timing for the interval timer to stop is shown below.

- When the use of the transmit/receive FIFO buffer is prohibited (the CFE bit is “0”)
- When transitioning to channel reset mode

Table 3-1 shows the count source of the interval timer and the calculation formula of the interval timer, Figure 3-5 shows the block diagram of the interval timer, and Figure 3-6 shows the operation example of the interval timer.

Table 3-1 Interval Timer Count Source and Interval Timer Calculation Formula

CFCCk		Count Source	Interval Timer Calculation Formula*
CFITR	CFITSS		
0	0	Clock obtained by dividing pclk/2 by the value of the ITRCP [15: 0] bit of the GCFG register	$1/f_{PBA} \times 2 \times M \times N$
1	0	Clock obtained by dividing pclk/2 by the value of the ITRCP [15: 0] bit of the GCFG register x10.	$1/f_{PBA} \times 2 \times M \times 10 \times N$
-	1	CANm normal bit time clock	$1/f_{CANBIT} \times N$

【Note】 M: Divided value of the clock source of the interval timer for FIFO (set value of the GCFG.ITRCP [15: 0])

N : Message transmission interval (set value of the CFCCk.CFITT [7: 0])

f<sub>PBA</sub> : pclk frequency

f<sub>CANBIT</sub> : Normal CANm bit time clock frequency

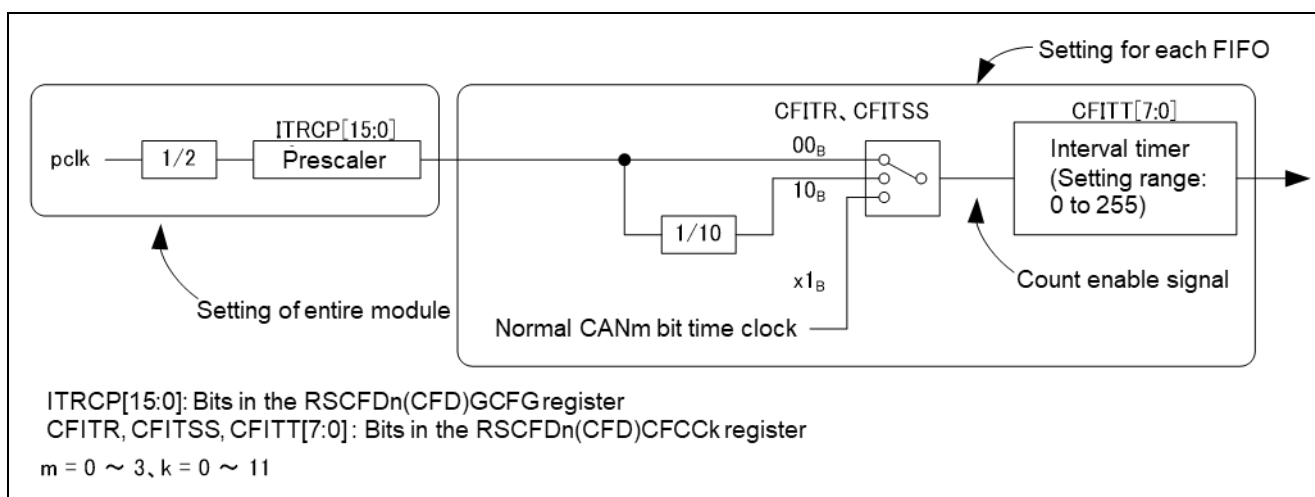


Figure 3-5 Block Diagram of the Interval Timer

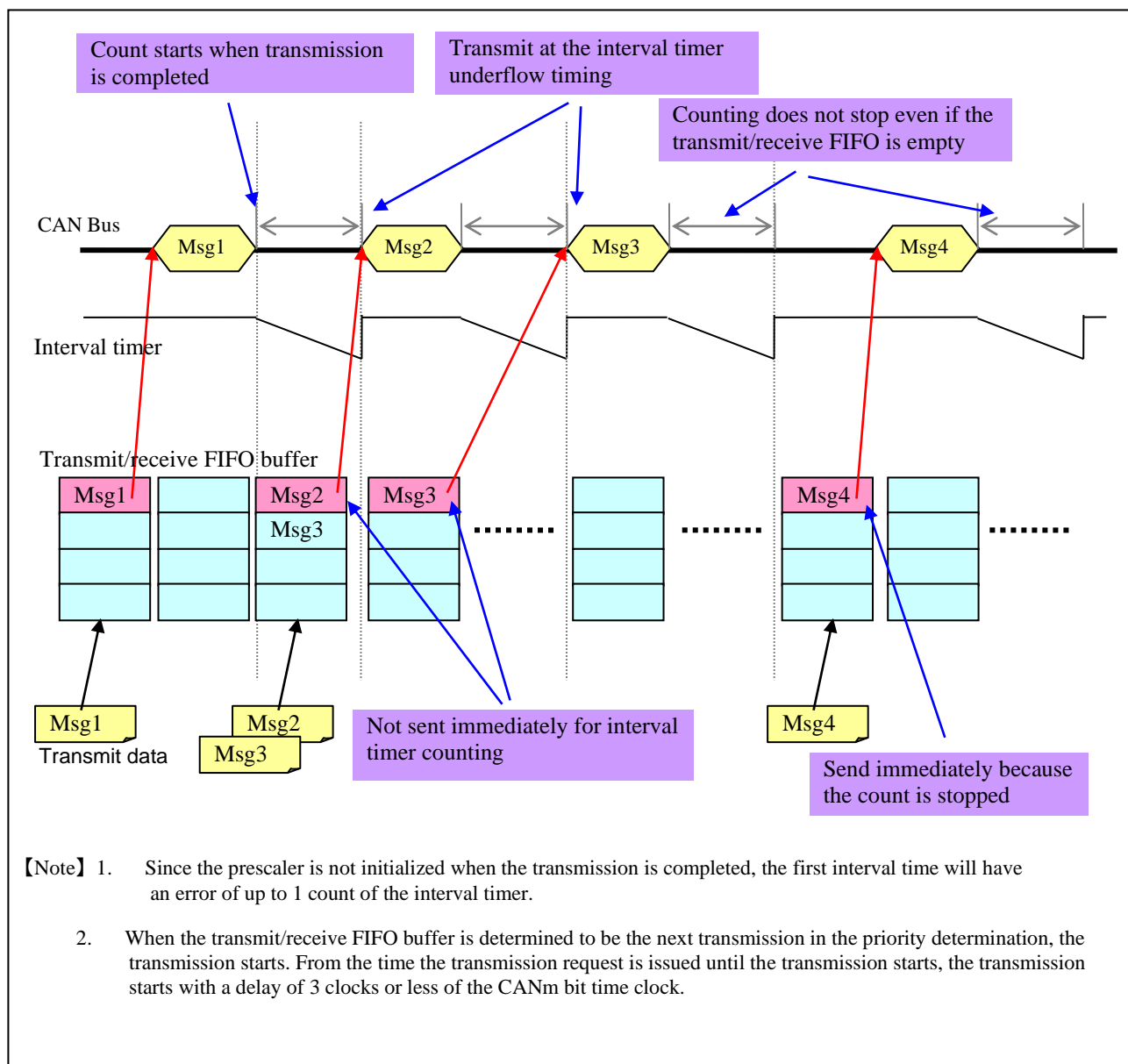


Figure 3-6 Interval Transmission Operation Example (Transmit Mode)



### 3.4 Interrupt Processing of Transmit/receive FIFO Buffer (Transmit Mode)

#### 3.4.1 Transmit/receive FIFO Transmit Interrupt Processing

If the transmit/receive FIFO transmission completion interrupt is enabled, the CANm transmit interrupt is generated when the condition selected in the CFIM bit setting of the CFCCk register is satisfied.

The CANm transmission interrupts share the following sources. When using multiple interrupt sources, determine the sources within the interrupt as necessary.

The sources of the CANm transmit interrupt can also be confirmed in the GTINTSTS0 register.

- CANm transmit completion interrupt / CANm transmit abort interrupt
- CANm transmit/receive FIFO transmission completion interrupt
- CANm transmit queue interrupt
- CANm transmit history interrupt

Even if the use of the transmit/receive FIFO buffer is prohibited (the CFE bit is "0") while an interrupt request is being generated (the CFTXIF flag in the CFSTSk register is "1"), the CFTXIF flag is not automatically set to "0". Set the interrupt request flag to "0" with the program.

You can enable or disable the transmit/receive FIFO transmit interrupt for each transmit/receive FIFO buffer using the CFTXIE bit of the CFCCk register.

The sources of the transmit/receive FIFO transmission completion interrupt in the transmit mode are shown below.

- Transmit/receive FIFO transmission completion interrupt request occurs when the buffer becomes empty due to the message transmission completion.

- Transmit/receive FIFO transmission completion interrupt request is generated each time one message transmission is completed.

送信割り込みを発生させるためには、**エラー! 参照元が見つかりません。**にある対応する割り込み許可ビットが"1"である割り込み要求フラグをすべて"0"にする必要があります。

## 4. Transmission Using Transmit Queue

The transmit queue is used to transmit data frames or remote frames.

The transmit queue is allocated up to 16 buffers per channel, and the transmit buffer  $[16 \times m + 15]$  is the common access window.

The functions of the transmit queue are shown below. For the configuration settings for using the transmit queue, refer to "CAN Configuration Application Note".

- Message transmission function
- Transmit abort function

### 4.1 Message Transmission Function

This is the function to transmit data frames or remote frames.

When using the transmit queue, set the message priority to ID priority.

All messages in the transmit queue are subject to transmit priority determination and are sent in order of ID priority, regardless of the order in which they were stored. If two messages with the same ID are stored in the transmit queue, the order in which these messages are sent may differ from the order in which they were stored in the transmit queue.

Figure 4-1 shows the operation of the transmit queue.

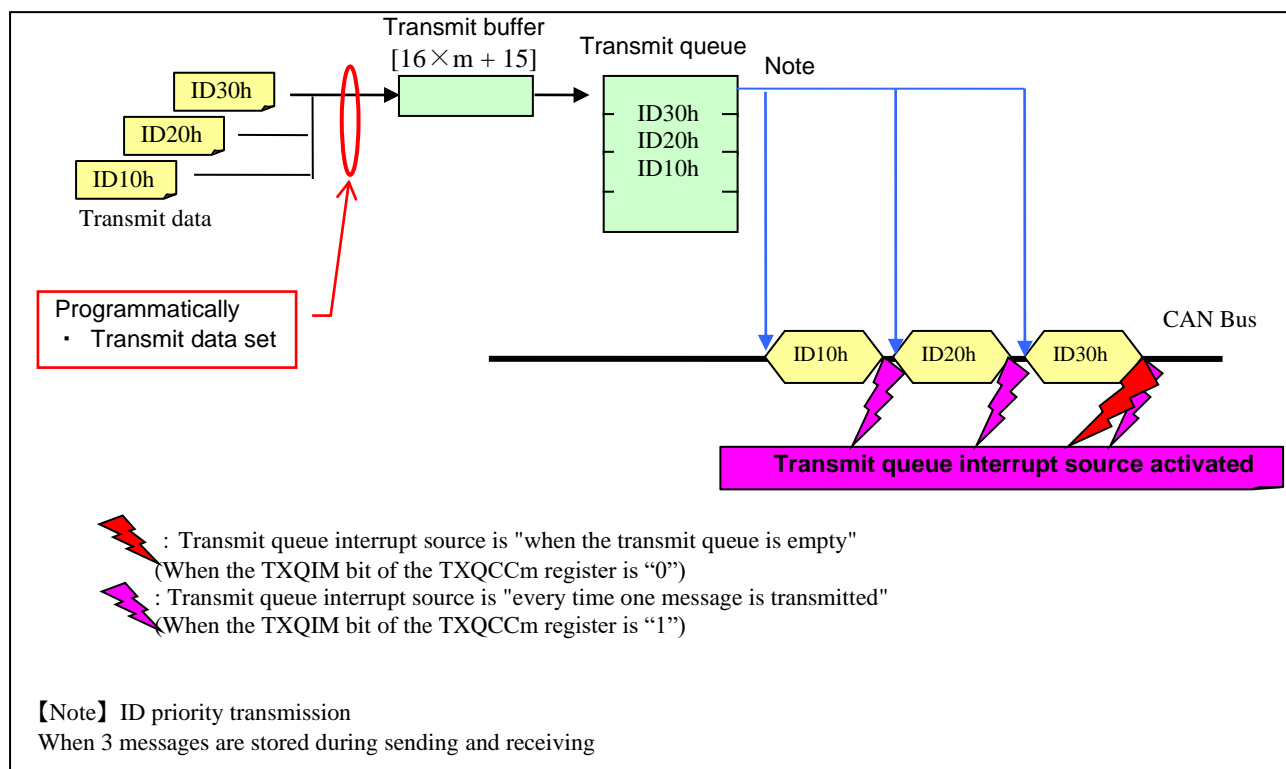
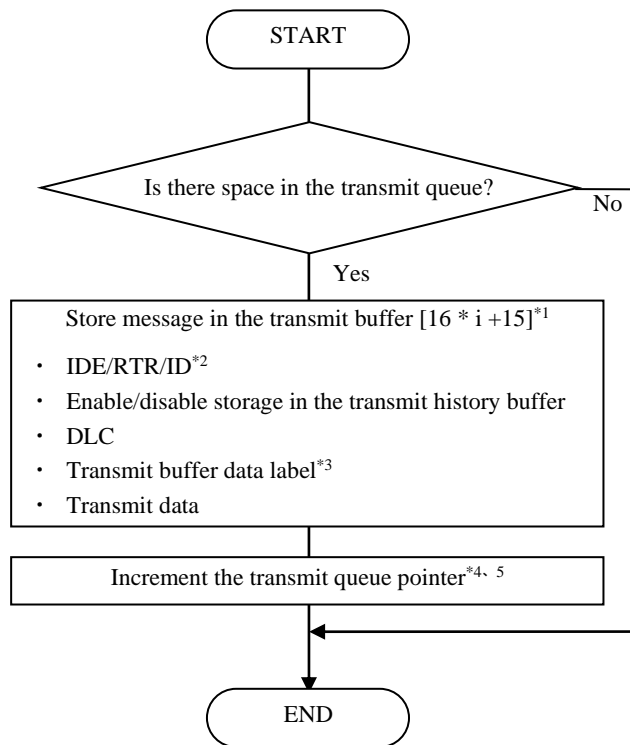


Figure 4-1 Transmit Queue Operation

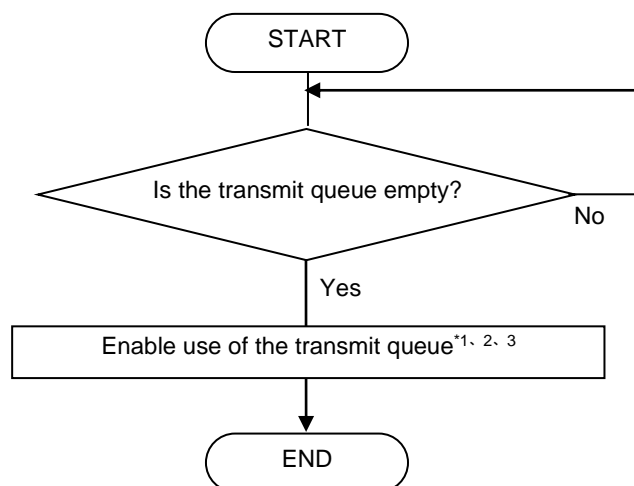
## 4.1.1 Message Transmission Procedure from Transmit Queue

Figure 4-2 shows the procedure for transmitting messages from the transmit queue, and Figure 4-3 and Figure 4-4 show the procedure for enabling or prohibiting the transmit queue.



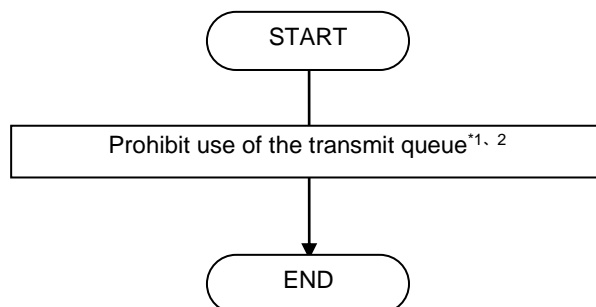
- 【Note】**
1. If assigned to the transmit queue, write only in the transmit buffer  $[16 \times m + 15]$ .
  2. When setting the standard ID in the transmit ID (the TMID [28: 0] bit of the TMIDp register), set the ID in b10-b0. Set b28-b11 to "0".
  3. This is valid only when the data is stored in the transmit history buffer (the THLE bit of the THLCCm register is "1" and the THLEN bit of the TMIDP register is "1").
  4. After writing the transmit message to the transmit queue, increment the transmit queue pointer (write "H'FF" in the TXQPC [7: 0] bit of the TXQPCTRM register).
  5. Increment the transmit queue pointer when the transmit queue is used (the TQE bit of the TXQCCm register is "1") and the transmit queue is not full (the TXQFLL flag in the TXQSTSm register is "0").

Figure 4-2 Message Transmission Procedure from Transmit Queue



- 【Note】
1. Disabling the transmit queue immediately empties the message in the transmit queue if it is not being transmitted and is not determined for the next transmission. If the message in the transmit queue is already being transmitted or is determined to be the next transmission, it will be empty after transmission completion, CAN bus error detection, or arbitration lost.
  2. Rewrite the transmit queue enable/disable (the TXQE bit of the TXQCCm register) in channel communication mode or channel standby mode.
  3. Set the number of buffers to be allocated to the transmit queue (set the TXQPC [3: 0] bit of the TXQCCm register to "B'0010" or higher), and then enable the transmit queue (set the TXQE bit to "1").

Figure 4-3 Procedure for Enabling the Transmit Queue



- 【Note】
1. Rewrite the transmit queue enable/disable (the TXQE bit of the TXQCCm register) in channel communication mode or channel standby mode.
  2. Even if the use of the transmit queue is prohibited (TXQE bit is "0") while an interrupt request is generated (the TXQIF flag in the TXQSTSm register is "1"), the TXQIF flag is not automatically set to "0". Set the interrupt request flag to "0" with the program.

Figure 4-4 Procedure for Prohibiting the Transmit Queue

## 4.2 Transmit Abort Function

By prohibiting the use of the transmit queue, messages in the transmit queue can be aborted. Transmit queue abort will abort all messages in the transmit queue, not just the message being transmitted (the transmit queue will be empty (the TXQEMP flag in the TXQSTSm register will be "1")). You can complete the abort of the transmit queue by checking the availability of the transmit queue.

No interrupt is generated when the transmit abort of the transmit queue is completed. However, if aborted during transmission, a transmit queue completion interrupt may occur due to transmission completion. For details, refer to "Figure 2-3 Application Example of Transmit Abort Function".

## 4.3 Transmit Queue Interrupt Processing

### 4.3.1 Transmit Queue Interrupt Processing

If the transmit queue interrupt is enabled, a CANm transmit-related interrupt is generated when the condition selected in the TXQIM bit setting of the TXQCCm register is satisfied. The transmit queue interrupt can be enabled or disabled for each transmit queue using the TXQIE bit of the TXQCCm register.

CANm transmission-related interrupts share the following sources. When using multiple interrupt sources, determine the sources within the interrupt as necessary.

The sources of CANm transmission-related interrupts can also be confirmed in the GTINTSTS0 register.

- CANm transmission completion interrupt
- CANm transmit abort interrupt
- CANm transmit/receive FIFO transmission completion interrupt
- CANm transmit queue interrupt
- CANm transmit history interrupt

Even if the use of the transmit queue is prohibited (the TXQE bit is "0") while an interrupt request is generated (the TXQIF flag of the TXQSTSm register is "1"), the TXQIF flag is not automatically set to "0". Set the interrupt request flag to "0" with the program.

The sources of the transmit queue interrupt are shown below.

- Transmit queue interrupt request occurs when the transmit queue becomes empty due to transmission completion
- A transmit queue interrupt request is generated each time one message is sent.

## 5. Transmit History Buffer Function

Information on messages that have been transmitted (transmit history data) can be stored in the transmit history buffer. Each channel has one transmit history buffer, and the transmit history buffer can store 16 transmit history data.

### 5.1 Transmit History Data Storage Function

You can set the type of buffer from which the message is transmitted and whether to store transmit history data for each message. The type of buffer from which the message is transmitted can be set during configuration. For the configuration settings for using the transmit history buffer, refer to "CAN Configuration Application Note".

You can set whether to store transmit history data and label data when sending each message.

For the setting procedure at the time of transmission, refer to "Figure 2-2 Procedure for Sending Messages from Transmit Buffer ", "Figure 3-2 Message Transmission Procedure from Transmit/receive FIFO Buffer ", and "Figure 4-2 Message Transmission Procedure from Transmit Queue ".

After successful transmission, the following information is stored in the transmit history buffer as transmit history data.

Buffer type: The buffer where the stored message was sent

Type of (transmit buffer, transmit queue, or transmit/receive FIFO buffer).

Buffer Number: The number of the source transmit buffer, transmit queue, or transmit/receive FIFO buffer (see Table 5-1).

Label data: Label information of transmit messages. Label information can be set when storing transmit messages.

Timestamp: Timestamp value of the transmit message.

Table 5-1 Transmit History Data Buffer Number

Buffer Type	B'001	B'010	B'100
Buffer Number	Transmit buffer	Transmit/receive FIFO buffer	Transmit queue
B'0000	Transmit buffer $m \times 16 + 0$	The number of the transmit buffer linked to the transmit/receive FIFO buffer with the CFTML [3: 0] bit of the CFCCk register	The number of the transmit buffer assigned to the transmit queue that made transmission
B'0001	Transmit buffer $m \times 16 + 1$		
B'0010	Transmit buffer $m \times 16 + 2$		
B'0011	Transmit buffer $m \times 16 + 3$		
B'0100	Transmit buffer $m \times 16 + 4$		
B'0101	Transmit buffer $m \times 16 + 5$		
B'0110	Transmit buffer $m \times 16 + 6$		
B'0111	Transmit buffer $m \times 16 + 7$		
B'1000	Transmit buffer $m \times 16 + 8$		
B'1001	Transmit buffer $m \times 16 + 9$		
B'1010	Transmit buffer $m \times 16 + 10$		
B'1011	Transmit buffer $m \times 16 + 11$		
B'1100	Transmit buffer $m \times 16 + 12$		
B'1101	Transmit buffer $m \times 16 + 13$		
B'1110	Transmit buffer $m \times 16 + 14$		
B'1111	Transmit buffer $m \times 16 + 15$		

バッファタイプ バッファ番号	B'01	B'10
	送信バッファ	送受信FIFOバッファ
B'00	送信バッファ $i \times 4 + 0$	CFCCHkレジスタの CFTML[1:0]ビットで送受信 FIFOバッファにリンクさせ た送信バッファの番号
B'01	送信バッファ $i \times 4 + 1$	
B'10	送信バッファ $i \times 4 + 2$	
B'11	送信バッファ $i \times 4 + 3$	

Figure 5-1 shows the operation of the transmit history buffer.

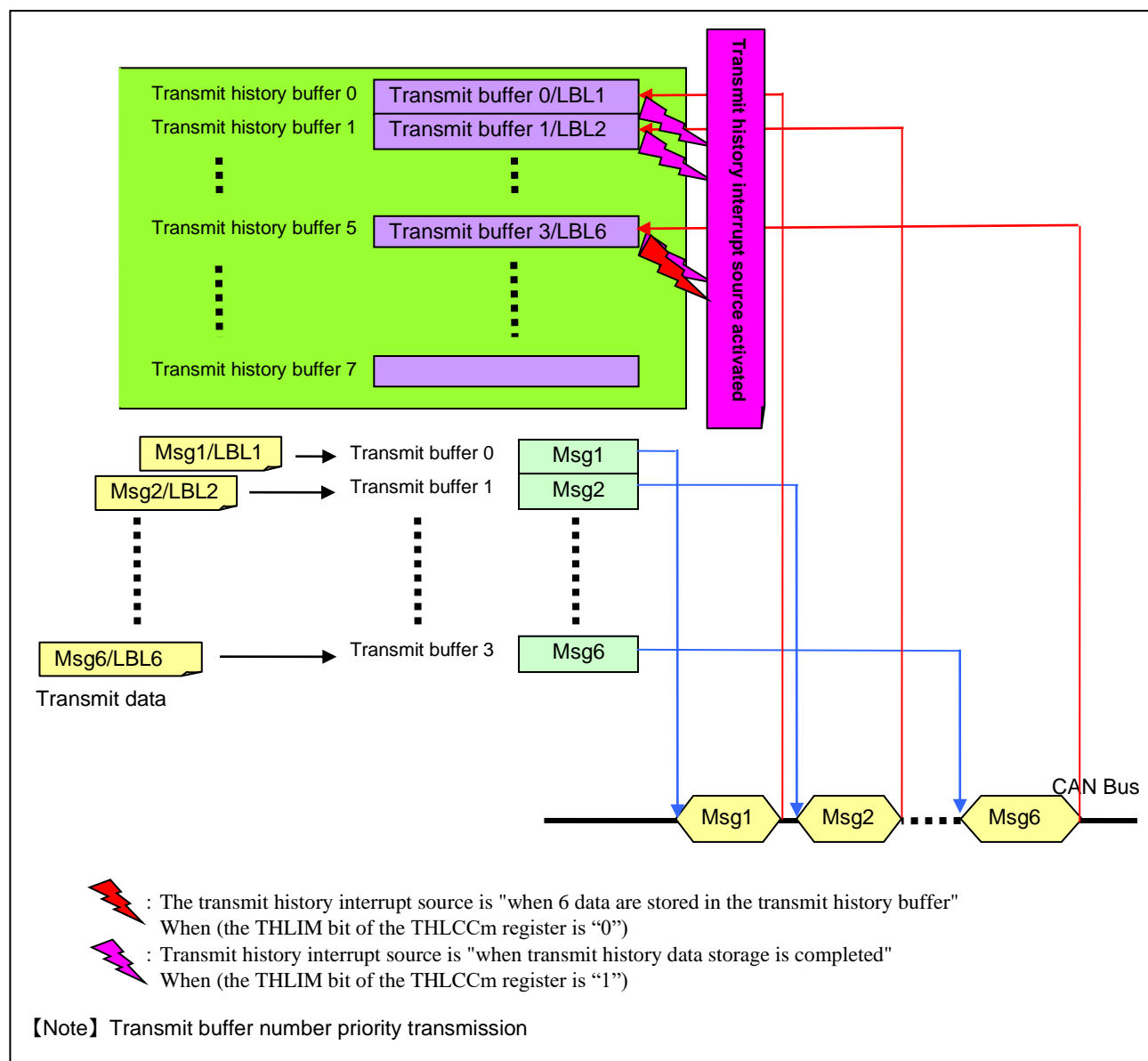
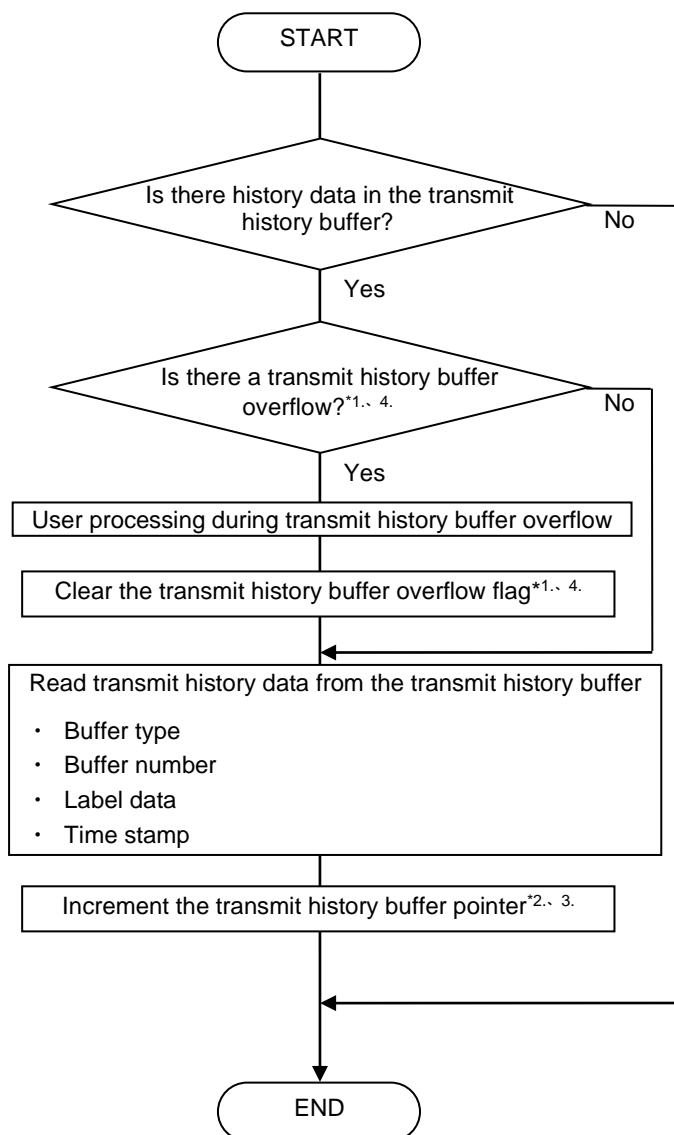


Figure 5-1 Operation of Transmit History Buffer



## 5.1.1 Transmit History Buffer Read Procedure

Figure 5-2 shows the procedure for reading transmit history data from the transmit history buffer, and Figure 5-3 and Figure 5-4 show the procedure for enabling and prohibiting the use of the transmit history buffer.



- 【Note】
1. Write “0” to set the THLELT bit of the THLSTSm register to “0” with the program.
  2. After reading the transmit history buffer (the THLACCm register), increment the transmit history buffer pointer (the THLPC [7: 0] bit of the THLPCTRm register is “H'FF”).
  3. Increment the transmit history buffer pointer when the transmit history buffer is used (the THLE bit of the THLCCm register is “1”) and the transmit history buffer is not empty (the THLEMP bit of the THLSTSm register is “0”).
  4. If the transmit history buffer overflow interrupt is enabled, execute it within the global error interrupt processing.

Figure 5-2 Transmit History Buffer Read Procedure

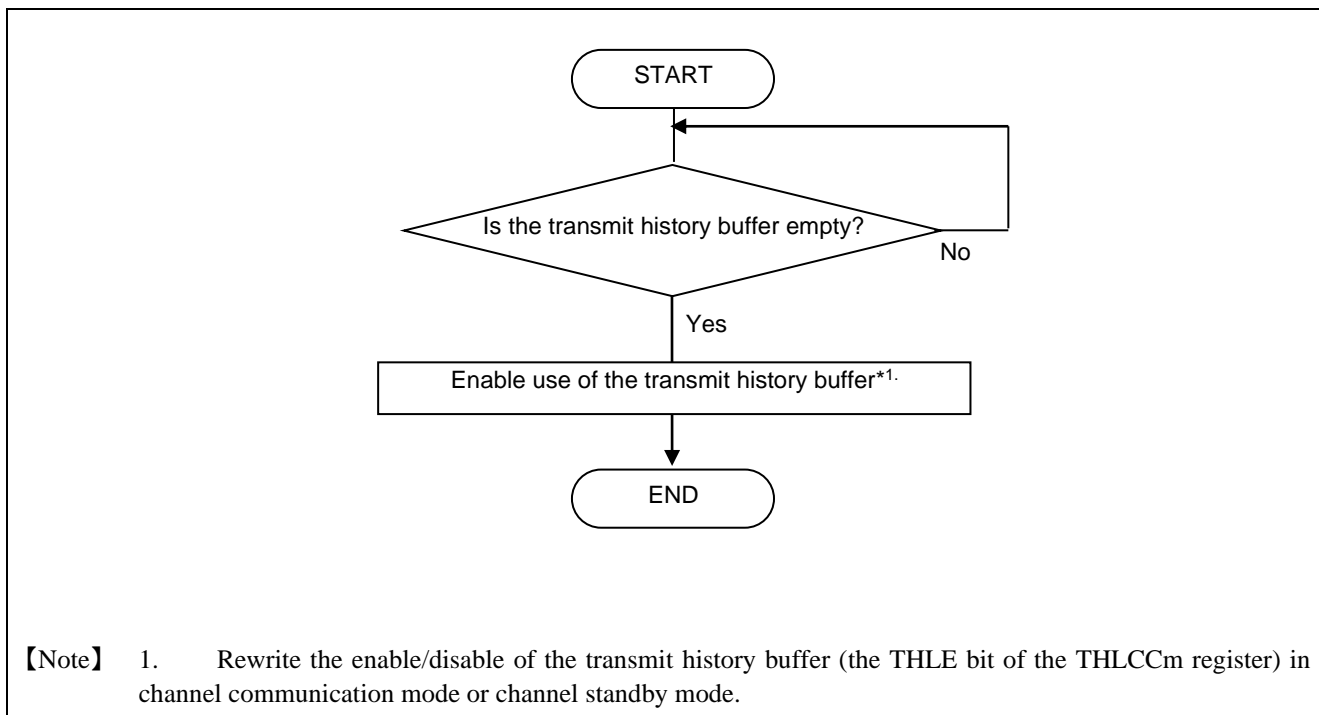


Figure 5-3 Procedure for Enabling Use of Transmit History Buffer

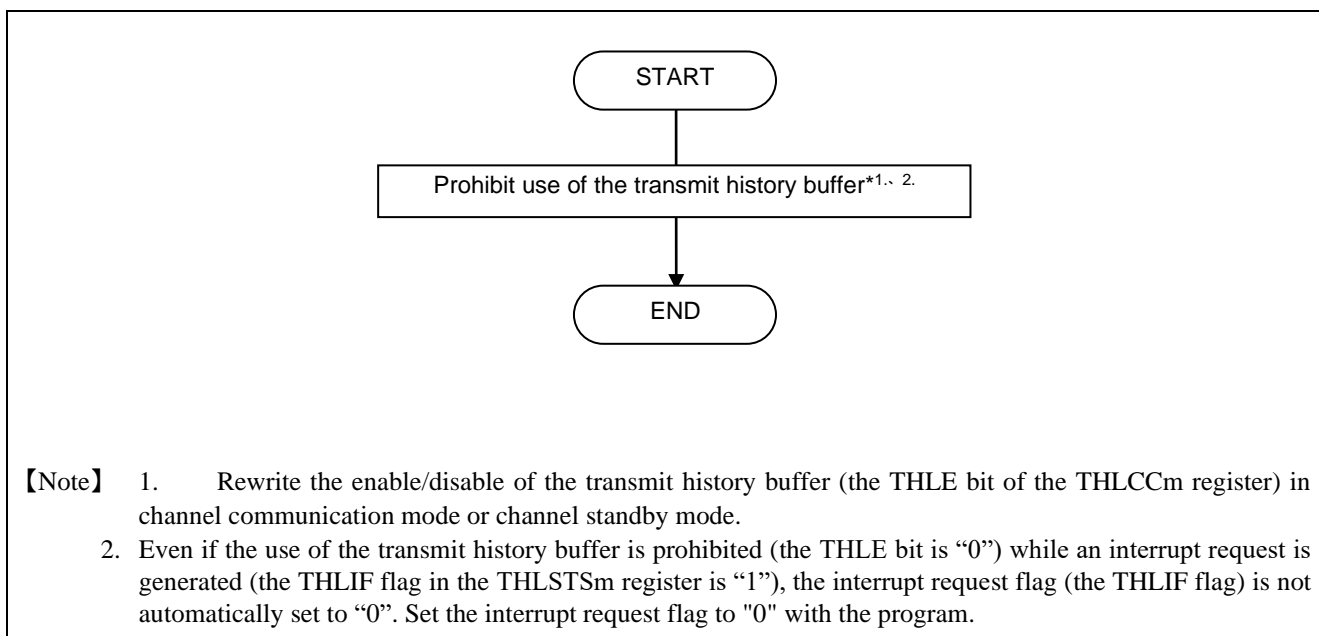


Figure 5-4 Procedure for Prohibiting Use of Transmit History Buffer

## 5.2 Transmit History Buffer Interrupt Processing

### 5.2.1 Transmit History Buffer Interrupt Processing

If the transmit history interrupt is enabled, the CANm transmit interrupt is generated when the condition selected in the THLIM bit setting of the THLCCm register is satisfied.

The CANm transmit interrupts share the following sources. When using multiple interrupt sources, determine the sources within the interrupt as necessary.

The sources of the CANm transmit interrupt can also be confirmed in the GTINTSTS0 register.

- CANm transmit completion interrupt
- CANm transmit abort interrupt
- CANm transmit/receive FIFO transmit completion interrupt
- CANm transmit queue interrupt
- CANm transmit history interrupt

Even if the use of transmit history is prohibited (the THLE bit is "0") while an interrupt request is generated (the THLIF flag in the THLSTSm register is "1"), the THLIF flag is not automatically set to "0". Set the interrupt request flag to "0" with the program.

Whether to enable or disable transmit history interrupts can be set for each transmit history buffer using the THLIE bit of the THLCCm register.

The sources of transmit history interrupts are shown below.

- Transmit history interrupt request occurs when 6 data are stored in the transmit history buffer
- A transmit history interrupt request is generated each time the storage of one transmit history data is completed.

### 5.2.2 Global Error Interrupt Processing

If the transmit history buffer overflow interrupt is enabled, a global error interrupt will occur when the transmit history buffer overflow is detected. The transmit history buffer overflow interrupt enable/disable can be set in common for the entire module with the THLEIE bit of the GCTR register.

## 6. CAN-related Interrupt Processing

When using interrupts, the interrupt source flag must be cleared to “0”. For the CAN-related flags related to each interrupt source flag on the interrupt control side, see "8.2 CAN-related interrupt sources".

Figure 6-1 shows how to clear the interrupt source flag to “0” in interrupt processing.

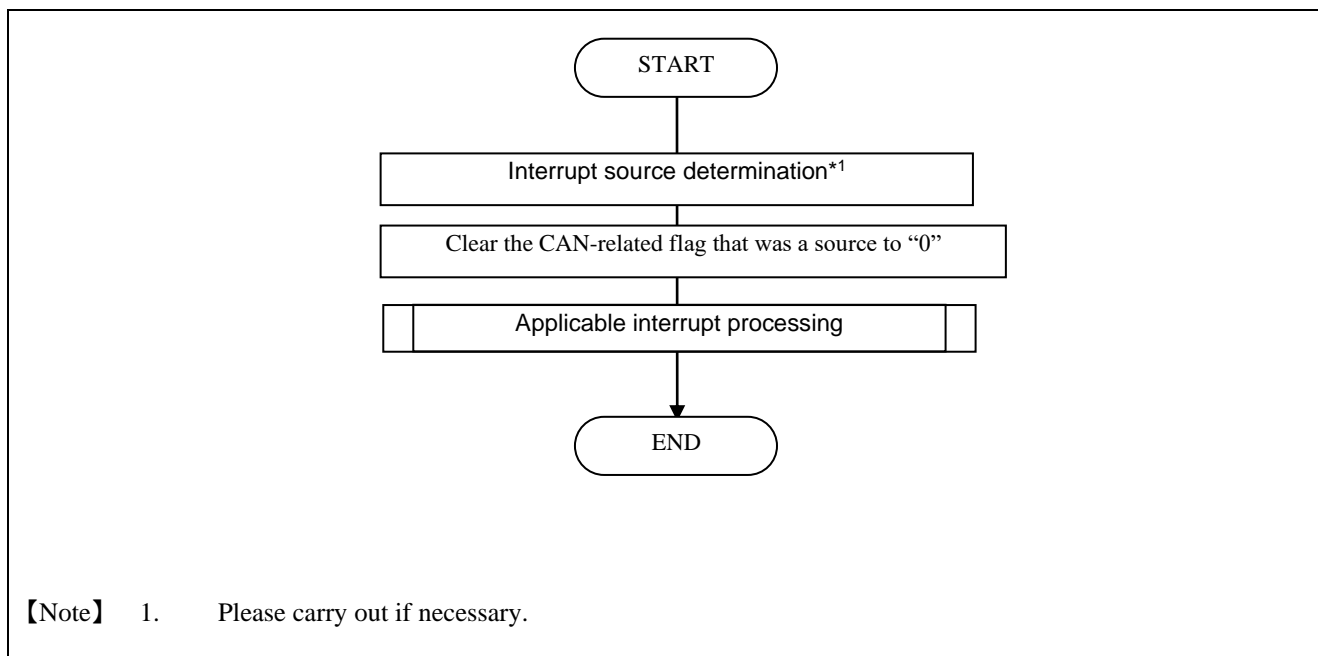


Figure 6-1 CAN-related Interrupt Processing Procedure

## 7. Transmit Data Padding (only in CAN FD mode)

If the payload length indicated by the DLC value of the set transmit message exceeds the payload storage area size of the buffer used for transmission, the excess payload is padded with “CCH”.

This processing is performed when the transmit buffer merge mode is not enabled (the TMME bit of the CFG register is “0”) and when the following cases occur.

- Transmit/receive FIFO set to transmit or gateway mode:

When the payload length of the transmit DLC exceeds the payload storage area size of the transmit/receive FIFO set in the CFPLS [2: 0] bit of the CFCCk register.

- Transmit buffer (including the transmit queue):

When the payload length of the transmit DLC exceeds 20 bytes

When transmit buffer merge mode is enabled, transmit data padding is not performed for transmissions using the transmit buffer, the transmit/receive FIFO buffer, or the transmit queue. At this time, do not set the DLC value of the transmit message to a payload length that exceeds the payload storage size of the buffer used for transmission.

## 8. Precautions for Processing Flow

### 8.1 About Functions

In this application note, there is a part that is functionalized even in the processing of single line, but this is only described as a function to clarify the processing for each function. When you actually create a program, you don't necessarily have to make it functional.

### 8.2 Settings for Each Channel, FIFO, and Buffer

In this application note, even if processing is required for each channel, FIFO, or buffer, only one processing is described. When actually creating a program, perform multiple processes as necessary.

### 8.3 Infinite Loop

To simplify the notation, there are some infinite loops in the processing flow. When actually creating a program, give each loop a time limit so that it can be exited during overtime. Figure 7-1 shows an example of processing with a loop time limit.

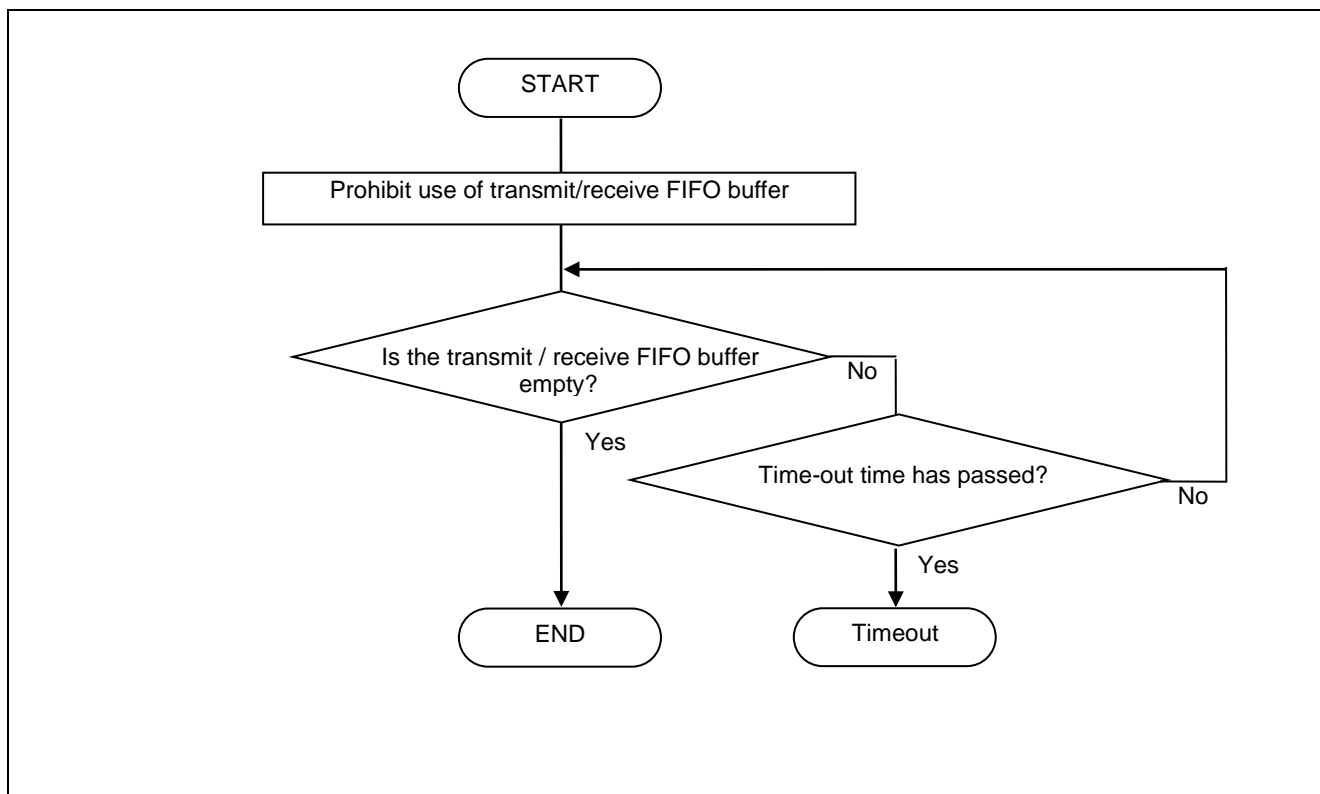


Figure 8-1 Example of Processing with Loop Time Limit

## 9. Appendix

### 9.1 Request to Transmit Buffer

The interrupt source that activates depends on the request to be issued to the transmit buffer and the conditions under which transmission is stopped.

Table 8-1 lists the requests to the transmit buffer and the interrupt sources.

Table 9-1 Requests to Transmit Buffer and Interrupt Sources

TMCp register			Event	Transmission result (TMTRF [1: 0] flag in the TMSTSp register)	Interrupt Source
Transmit request (TMTR)	Transmit abort request (TMTAR)	One-shot transmit request (TMOM)			
"1"	"0"	"0"	Transmission completion	"B'10" Transmission completion: No abort request	Transmission completion interrupt
			Arbitration lost or error occurs	"B'00" Transmitting	None
"1"	"1"	"0"	Transmission completion	"B'11" Transmission completion: Abort request	Transmission completion interrupt
			Arbitration lost or error occurs	"B'01" Transmission abort completion	Transmit abort interrupt
"1"	"0"	"1"	Transmission completion	"B'10" Transmission completion: No abort request	Transmission completion interrupt
			Arbitration lost or error occurs	"B'01" Transmission abort completion	Transmit abort interrupt
"1"	"1"	"1"	Transmission completion	"B'11" Transmission completion: Abort request	Transmission completion interrupt
			Arbitration lost or error occurs	"B'01" Transmission abort completion	Transmit abort interrupt
"0"	"x"	"x"	Cannot be set		

### 9.2 CAN-related Interrupt Sources

Table 8-2 shows the CAN-related interrupt sources.

Interrupt	Occurrence Source	Configuration (Enable) Unit * <sup>1</sup>	Interrupt Source	Request Clear Method * <sup>1</sup>
Global receive FIFO interrupt	Receive FIFOx interrupt request	RFIE bit of the RFCCx register	When the condition set by the RFIGCV[2:0] bit of the RFCCx register is met* <sup>2</sup>	RFIF flag in the RFSTSx register = "0"
			Every time one message is received	
Global error interrupt	DLC check error	DEF flag in the GERFL register	When the DLC check detects an error	DEF flag in the GERFL register = "0"
	FIFO message lost	MES bit of the GERFL register	When a message lost in the transmit/receive FIFO buffer is detected	<ul style="list-style-type: none"> <li>CFMLT flag in the CFSTSx register of all channels = "0".</li> <li>RFMLT flag in the CRFSRx register of all channels = "0"</li> </ul>
			When a message lost in the receive FIFO buffer is detected	
	Transmit history buffer overflow	THLES flag in the GERFL register	When the transmission history buffer is full and you try to store more new transmission history data	THMLT flag in the THLSTSm register of all channels = "0"
CANm transmit interrupt	CANm transmit complete interrupt request	TMIEp bit of the TMIECy register	When the buffer becomes empty due to the completion of message transmission	TMTRF[1:0] flag in the TMSTSp register = "B'00".
	CANm transmit abort interrupt request	TAIE bit of the CmCTR register	When the buffer becomes empty due to the completion of sending a message abort	
	CANm transmit queue interrupt request	TQIE bit of the TXQCCm register	When the send queue becomes empty due to the completion of transmission	TQIF bit of the TXQSTSm register = "0"
			Every time one message is sent	
	CANm transmit /receive FIFO transmission complete interrupt request	CFTXIE bit of the CFCCk register	When the buffer becomes empty due to the completion of message transmission	CFTXIF flag in the CFSTSx register = "0"
			Every time one message is sent	
	CANm transmit history interrupt request	THIE bit of the THLCCm register	When 12 data are stored in the transmission history buffer	THIF flag in the THLSTSm register = "0"
			Every time the transmission history data is stored	
CANm transmit/receive FIFO reception completion interrupt	Channel m transmit/receive FIFO reception complete interrupt request	CFRXIE bit of the CFCCk register	When the condition set by the TRFRIT bit of the CFCCk register is met* <sup>3</sup>	CFRXIF flag in the CFSTSx register = "0"
			Every time one message is received	
CANm error interrupt	Bus error	BEIE bit of the CmCTR register	When any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags of the CmERFL register becomes "1".* <sup>4</sup>	BEF flag in the CmERFL register = "0"
	Error warning	EWIE bit of the CmCTR register	When the value of the REC [7: 0] or TEC [7: 0] bit of the CmERFL register exceeds 95	EWf flag in the CmERFL register = "0"
	Error passive	EPIE bit of the CmCTR register	In case of the error passive state (REC [7: 0] or TEC [7: 0] bits > 127).	EPF flag in the CmERFL register = "0"
	Bus off start	BOEIE bit of the CmCTR register	When the bus is off (TEC [7: 0] bits > 255)	BOEF flag in the CmERFL register = "0"
	Bus off return	BORIE bit of the CmCTR register	When 11-bit consecutive recessive is detected 128 times and the bus is restored from the off state.* <sup>5</sup>	BORF flag in the CmERFL register = "0"
	Overload frame transmission	OLEE bit of the CmCTR register	When the transmission condition of the overloaded frame is detected when receiving or transmitting	OVLf flag in the CmERFL register = "0"
	Bus lock	BLIE bit of the CmCTR register	When a 32-bit consecutive dominant is detected on the CAN bus in channel communication mode	BLF flag in the CmERFL register = "0"
	Arbitration lost	ALIE bit of the CmCTR register	When arbitration lost is detected	ALF flag in the CmERFL register = "0"
CANm wakeup interrupt	CAN bus falling edge detection	--	When a falling edge is detected at the CRXmD pin	--



Table 9-2 CAN-related Interrupt Sources

- 【Note】 1. The interrupt request flag and interrupt enable bit in the interrupt function are not described. For details, refer to the interrupt chapter in the hardware chapter of each user's manual.
2. Settings of the RFIGCV[2:0] bit of the RFCCx register
- ▪ When a message is stored up to 1/8 in the receive FIFO buffer\*
  - ▪ When a message is stored up to 2/8 in the receive FIFO buffer
  - ▪ When a message is stored up to 3/8 in the receive FIFO buffer\*
  - ▪ When a message is stored up to 4/8 in the receive FIFO buffer
  - ▪ When a message is stored up to 5/8 in the receive FIFO buffer\*
  - ▪ When a message is stored up to 6/8 in the receive FIFO buffer
  - ▪ When a message is stored up to 7/8 in the receive FIFO buffer\*
  - ▪ When the receive FIFO buffer is full
- \* Do not set if the number of buffers in the receive FIFO buffer is set to 4 messages (the RFDC [2:0] bit of the RFCCx register is "B'001").
3. Settings of the RFIGCV[2:0] bit of the CFCCk register
- ▪ When a message is stored up to 1/8 in the transmit/receive FIFO buffer\*
  - ▪ When a message is stored up to 2/8 in the transmit/receive FIFO buffer
  - ▪ When a message is stored up to 3/8 in the transmit/receive FIFO buffer\*
  - ▪ When a message is stored up to 4/8 in the transmit/receive FIFO buffer
  - ▪ When a message is stored up to 5/8 in the transmit/receive FIFO buffer\*
  - ▪ When a message is stored up to 6/8 in the transmit/receive FIFO buffer
  - ▪ When a message is stored up to 7/8 in the transmit/receive FIFO buffer\*
  - ▪ When the transmit/receive FIFO buffer is full
- ※Do not set if the number of buffers in the transmit/receive FIFO buffer is set to 4 messages (the CFDC [2:0] bit of the CFCCk register is "B' 001" ).
4. An interrupt is generated when any one of the following is detected.
- ▪ The ADERR flag in the CmERFL register is "1", and a form error is detected by ACK delimiter.
  - ▪ The B0ERR flag in the CmERFLL register is "1", and a recessive is detected despite sending a dominant.
  - ▪ The B1DRR flag in the CmERFL register is "1", and a dominant is detected despite sending a recessive.
  - ▪ The CERR flag in the CmERFL register is "1", and a CRC error is detected.
  - ▪ The AERR flag in the CmERFLL register is "1", and an ACK error is detected.
  - ▪ The FERR flag in the CmERFLL register is "1", and a form error is detected.
  - ▪ The SERR flag in the CmERFLL register is "1", and a stuff error is detected.
5. If you return from the bus-off state by the following methods before 11 consecutive recessive bits 128 times are detected, no interrupt will be generated (the BORF flag will not be "1").
- ▪ When the CHMDC [1: 0] bit of the CmCTR register is set to "B'01" (channel reset mode)
  - ▪ When the RTBO bit of the CmCTR register is set to "1" (forced recovery from bus off)
  - ▪ When the BOM [1: 0] bit of the CmCTR register is set to "B'01" (transition to channel standby mode when bus off starts)
  - ▪ When the BOM [1: 0] bit is "B'11" (transition to channel standby mode at the request of the program during bus off) and the CHMDC [1: 0] bit is set to "B'10" (channel standby mode) before detecting 11 consecutive recessive bits 128 times.

## Our Company's Website and Inquiry

### Website

<http://japan.renesas.com/>

### Inquiry

<http://japan.renesas.com/contact/>

All trademarks and registered trademarks are the property of their respective owners.

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	2017.5.10	—	Initial edition

## Precautions for use of the product

This section describes the "Precautions" that apply to all microcontroller products. Please refer to this document and the Technical Update for precautions on individual products.

### 1. Treatment of unused pins

[Caution] Please dispose of unused pins according to "Handling of unused pins" in the text.

The impedance of the input pins of CMOS products is generally high impedance. If the unused pins are operated in an open state, noise around the LSI may be applied due to the induction phenomenon, a through current may flow inside the LSI, or it may be recognized as an input signal and malfunction may occur. Dispose of unused pins according to the instructions given in "Disposal of unused pins" in the text.

### 2. Treatment at power-on

[Caution] The state of the product is undefined when the power is turned on.

When the power is turned on, the state of the internal circuits of the LSI is indeterminate and the state of register settings and pins is undefined.

For products that are reset using the external reset pin, the pin state cannot be guaranteed from the time the power is supplied until the reset becomes valid.

Similarly, in the case of products that are reset using the built-in power-on reset function, the pin states cannot be guaranteed from the time the power is turned on until the voltage reaches a certain level.

### 3. Prohibition of Access to Reserved Addresses

[Caution] Access to reserved addresses is prohibited.

The address area has a reserved address allocated for future function expansion. The operation when these addresses are accessed cannot be guaranteed, so do not access them.

### 4. About clock

[Caution] When resetting, release the reset after the clock has stabilized.

When switching the clock during program execution, switch the clock after the switching destination clock is stable.

In a system that starts operating with a clock that uses an external oscillator (or external oscillator circuit) at reset, release the reset after the clock is sufficiently stable. Also, when switching to a clock that uses an external oscillator (or external oscillator circuit) in the middle of a program, make sure that the clock to be switched to is sufficiently stable before switching.

### 5. Differences between products

[Caution] When changing to a product with a different model name, perform a system evaluation test for each product model name.

Even if the MCUs in the same group have different model numbers, the characteristic values, operating margins, noise immunity, noise radiation, etc. may differ within the range of electrical characteristics due to differences in internal ROM and layout patterns. When changing to a product with a different model name, perform a system evaluation test for each individual product.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.  
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space systems; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.  
(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.  
(Note 2) "Renesas Electronics products" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4-0-1 November 2017)



### SALES OFFICES

### Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "http://www.renesas.com" for the latest and detailed information.

#### Renesas Electronics America Inc.

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.  
Tel: +1-408-432-8888, Fax: +1-408-434-5351

#### Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3  
Tel: +1-905-237-2004

#### Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-651-700, Fax: +44-1628-651-804

#### Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

#### Renesas Electronics (China) Co., Ltd.

Room 1709 Quantum Plaza, No. 27 Zhichun Lu, Haidian District, Beijing, 100191 P. R. China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

#### Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852-2886-9022

#### Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9500, Fax: +886-2-8175-9670

#### Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #05-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300

#### Renesas Electronics Malaysia Sdn.Bhd.

Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jin Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

#### Renesas Electronics India Pvt. Ltd.

No. 777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India  
Tel: +91-80-67208700, Fax: +91-80-67208777

#### Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 252, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5338