R32C/100 Series

CAN Application note

R01AN1651EJ0101
Rev.1.01
Mar 31, 2013

1. Overview

This document describes the procedures for CAN communication using the R32C/100 Series.

*: For usage notes of the SFRs, refer to the Hardware Manual of the latest version.

2. Application

This document applies to the R32C/100 Series.

The R32C/100 series has the CAN modules in the maximum by four channels (excluding the R32C/145 group).

The variables i, j, and k are used in this document. Those variables represent the following.

i : CAN channel number (i = 0 to 3)

j : Mailbox number (j = 0 to 31)

k :Mask register number (k = 0 to 7)

Contents

3. Initial Settings

The following settings are required for CAN communication:

- Clock setting [Refer to section 3 “Initial Settings”.]
- Bit timing setting [Refer to section 3 “Initial Settings”.]
- Baud rate setting [Refer to section 3 “Initial Settings”.]
- Acceptance filter setting [Refer to section 9 “Using an Acceptance Filter”.]

3.1 CAN Bit Timing

The CAN bit timing settings for the CAN module of the R32C/100 Series affect the three segments that make up each bit of the communication frame.

Figure 1 shows the segment structure and the sampling point.

Of the three segments, time segment 1 (TSEG1) and time segment 2 (TSEG2) specify the sampling point. The sampling timing can be adjusted by changing the values of TSEG1 and TSEG2.

The minimum unit in which timing settings are specified is called a time quanta (Tq). The Tq is determined by the clock frequency input to the CAN module and the baud rate prescaler frequency division value.

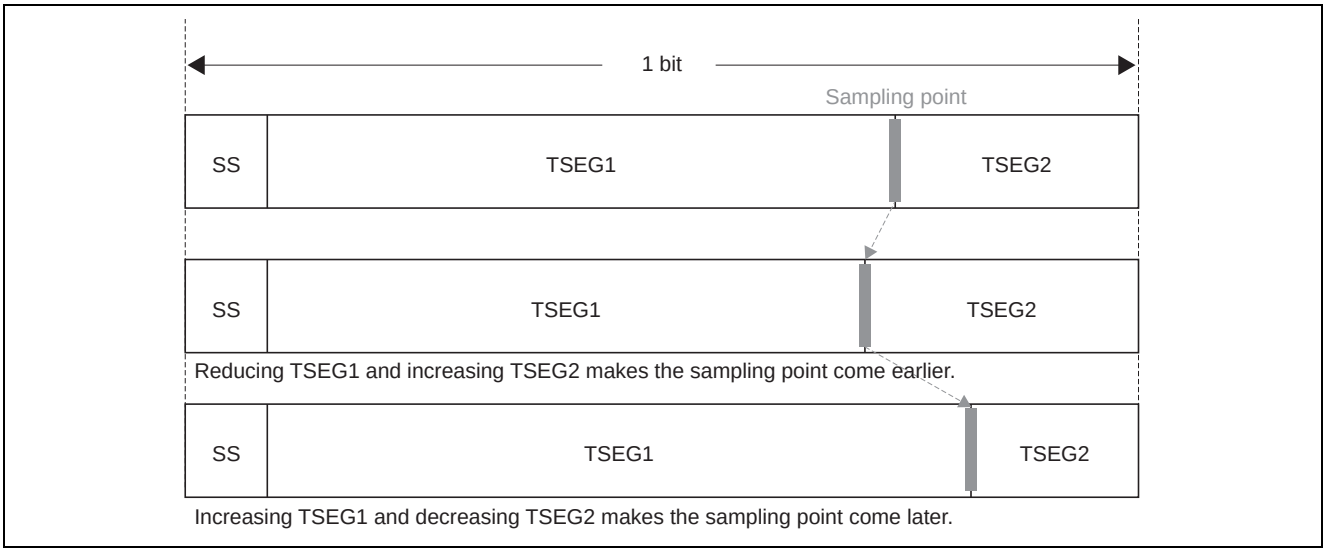


Figure 1 Segment Structure of Bit and Sampling Point

(1) SS: Synchronization Segment
This segment is used for synchronization by monitoring the edge transition from recessive to dominant within the interframe space.*[1]

(2) TSEG1: Time Segment 1
This segment's purpose is to absorb the physical delay inherent in the CAN network and to compensate for phase error*[2] occurring during resynchronization. The physical delay inherent in the network is equal to two times the sum of the bus delay, the input comparator delay, and the output driver delay.

(3) TSEG2: Time Segment 2
This segment's purpose is to compensate for phase error*[2] occurring during resynchronization.

(4) SJW: Resynchronization Jump Width
This is the maximum width of compensation to correct synchronization misalignment due to phase error*[2].

Note 1. The interframe space comprises three conditions: intermission, suspend transmission, and bus idle. All nodes can start transmission during the bus idle condition.
2. Phase error refers to synchronization misalignment between nodes, during message transmission or reception, due to factors such as a shift in the oscillator frequency or delay in the signal transfer path.

3.1.1 Bit Timing Conditions

The settings and restrictions for each segment are as follows:

(1) Setting for Each Segment

- SS = 1 Tq, fixed
- TSEG1 = Setting within range of 4 to 16 Tq
- TSEG2 = Setting within range of 2 to 8 Tq
- SJW = Setting within range of 1 to 4 Tq
- SS + TSEG1 + TSEG2 = 8 to 25 Tq

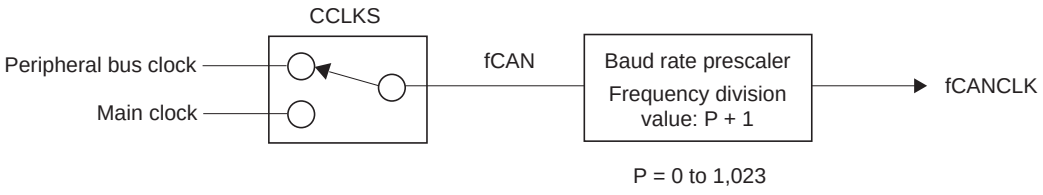
(2) Restrictions for TSEG1 and TSEG2

- TSEG1 > TSEG2 ≥ SJW (however, TSEG2 ≥ 2 when SJW = 1)

3.2 Transfer Speed

The transfer speed is determined by fCAN*, the baud rate prescaler frequency division value, and the Tq count per bit. Figure 2 is a block diagram of the CAN module system clock.

Table 1 shows the main formula for calculating the transfer speed and actual examples, and table 2 shows bit timing setting examples.



fCAN: CAN module system clock
P: Value specified by prescaler division ratio set bits in CANi bit configuration register (P = 0 to 1023)
fCANCLK: CAN communication clock (fCANCLK = fCAN / (P + 1))
CCLKS: Bit in the CiCLKR register

Figure 2 Block Diagram of CAN Module System Clock Generator Circuit

Table 1 Formula for Calculating Transfer Speed and Actual Examples

Formula for calculating transfer speed

$$\frac{\text{fCAN}}{\text{Baud rate prescaler frequency division value}^{*1} \times \text{Tq count per bit}} = \frac{\text{fCANCLK}}{\text{Tq count per bit}}$$

Transfer speed \ fCAN	32 MHz	30 MHz	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz
1 Mbps	8 Tq(4) 16 Tq(2)	10 Tq(3) 15 Tq(2)	8 Tq(3) 12 Tq(2) 24 Tq(1)	10 Tq(2) 20 Tq(1)	8 Tq(2) 16 Tq(1)	10 Tq(1)	8 Tq(1)
500 Kbps	8 Tq(8) 16 Tq(4)	10 Tq(6) 15 Tq(4) 20 Tq(3)	8 Tq(6) 12 Tq(4) 16 Tq(3) 24 Tq(2)	10 Tq(4) 20 Tq(2)	8 Tq(4) 16 Tq(2)	10 Tq(2) 20 Tq(1)	8 Tq(2) 16 Tq(1)
250 Kbps	8 Tq(16) 16 Tq(8)	10 Tq(12) 15 Tq(8) 20 Tq(6)	8 Tq(12) 12 Tq(8) 16 Tq(6) 24 Tq(4)	10 Tq(8) 20 Tq(4)	8 Tq(8) 16 Tq(4)	10 Tq(4) 20 Tq(2)	8 Tq(4) 16 Tq(2)
125 Kbps	8 Tq(32) 16 Tq(16)	10 Tq(24) 15 Tq(16) 20 Tq(12)	8 Tq(24) 12 Tq(16) 16 Tq(12) 24 Tq(8)	10 Tq(16) 20 Tq(8)	8 Tq(16) 16 Tq(8)	10 Tq(8) 20 Tq(4)	8 Tq(8) 16 Tq(4)
83.3 Kbps	8 Tq(48) 12 Tq(32) 16 Tq(24) 24 Tq(16)	8 Tq(45) 10 Tq(36) 12 Tq(30) 15 Tq(24) 20 Tq(18) 24 Tq(15)	8 Tq(36) 12 Tq(24) 16 Tq(18) 24 Tq(12)	8 Tq(30) 10 Tq(24) 12 Tq(20) 15 Tq(16) 16 Tq(15) 20 Tq(12) 24 Tq(10)	8 Tq(24) 12 Tq(16) 16 Tq(12) 24 Tq(8)	8 Tq(15) 10 Tq(12) 12 Tq(10) 15 Tq(8) 20 Tq(6) 24 Tq(5)	8 Tq(12) 12 Tq(8) 16 Tq(6) 24 Tq(4)
33.3 Kbps	8 Tq(120) 10 Tq(96) 12 Tq(80) 15 Tq(64) 16 Tq(60) 20 Tq(48) 24 Tq(40)	10 Tq(90) 12 Tq(75) 15 Tq(60) 20 Tq(45)	8 Tq(90) 10 Tq(72) 12 Tq(60) 15 Tq(48) 16 Tq(45) 20 Tq(36) 24 Tq(30)	8 Tq(75) 10 Tq(60) 12 Tq(50) 15 Tq(40) 20 Tq(30) 24 Tq(25)	8 Tq(60) 10 Tq(48) 12 Tq(40) 15 Tq(32) 16 Tq(30) 20 Tq(24) 24 Tq(20)	10 Tq(30) 12 Tq(25) 15 Tq(20) 20 Tq(15)	8 Tq(30) 10 Tq(24) 12 Tq(20) 15 Tq(16) 16 Tq(15) 20 Tq(12) 24 Tq(10)

Notes: 1. Baud rate prescaler frequency division value = P + 1 (P = 0 to 1023)
P: Value specified by prescaler division ratio set bits in CANi bit configuration register
2. Figures in parentheses ( ) are baud rate prescaler frequency division values.

Table 2 Bit Timing Setting Examples

1 Bit	Setting (Tq)				Sampling Point* (%)
	SS	TSEG1	TSEG2	SJW	
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
12 Tq	1	8	3	1	75.00
	1	9	2	1	83.33
15 Tq	1	10	4	1	73.33
	1	11	3	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
24 Tq	1	15	8	1	66.66

Note: * The point at which the level of a bit is determined.

When the sampling point is set to 75%

3.3 CAN Bit Timing and Transfer Speed Settings

Figure 3 shows the procedure for setting the CAN bit timing and transfer speed.

These settings must be performed during CAN configuration.

See section 4.1 for details on the CAN configuration procedure.

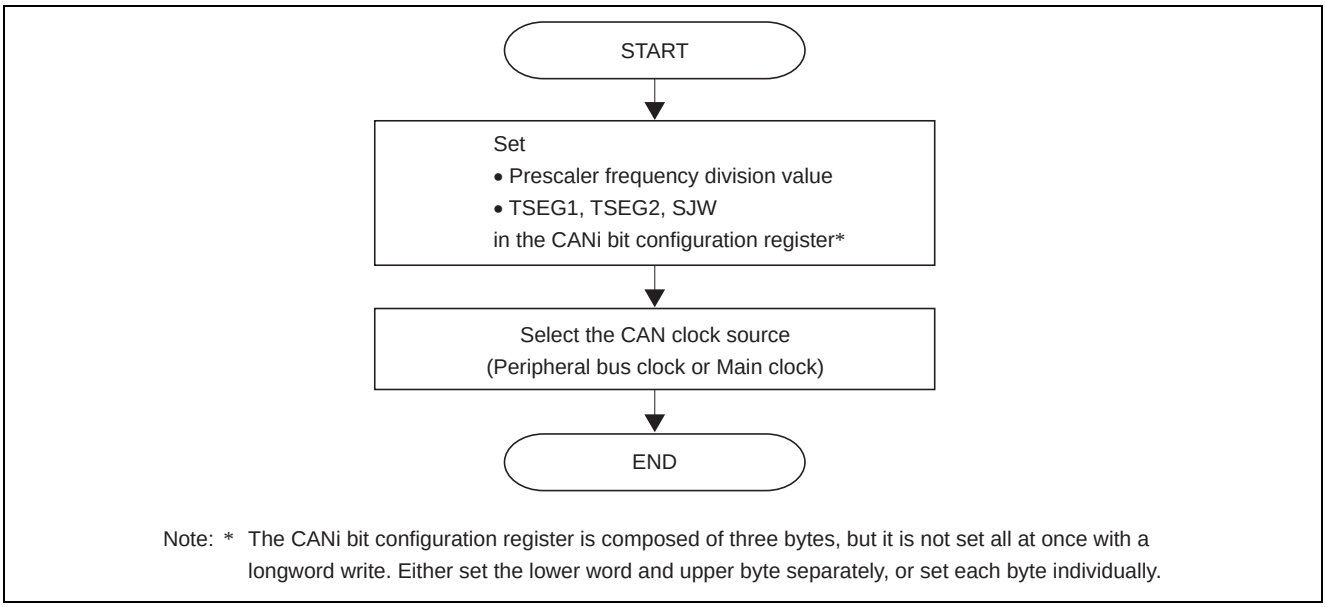


Figure 3 CAN Bit Timing and Transfer Speed Setting Procedure

4. Transmission and Reception of CAN Messages

The three procedures listed below are used to transmit and receive CAN messages.

These procedures are used in normal mailbox mode (CAN mailbox mode select bit (MBM) = 0). For details on FIFO mailbox mode (CAN mailbox mode select bit (MBM) = 1), see section 5.

(1) CAN Configuration Procedure
During CAN configuration, CAN transfer speed, control mode, acceptance filter, and interrupt settings are made.

(2) Mailbox Configuration Procedure
The mailbox settings for the transmission and reception modes are made in the CANi message control register j (CiMCTLj) corresponding to each mailbox.
Table 3 lists the correspondences between the CANi message control register j (CiMCTLj) settings and the transmission/reception modes.

(3) Data Processing Procedure
This is the message processing that takes place during message transmission and reception as well as at normal reception complete.

Table 3 Correspondences of CANi Message Control Register j Settings and Transmission/ Reception Modes

TRMREQ*	RECREQ*	ONESHOT*	Mailbox Transmission/Reception Mode Setting
0	0	0	Mailbox unusable or in transmission-abort state
0	0	1	Setting possible only in one-shot mode when transmission from or reception to the programmed mailbox has been aborted
0	1	0	Mailbox set as receive mailbox for data frame or remote frame
0	1	1	Mailbox set as one-shot receive mailbox data frame or remote frame
1	0	0	Mailbox set as transmit mailbox for data frame or remote frame
1	0	1	Mailbox set as one-shot transmit mailbox for data frame or remote frame
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note: * Bit in CANi message control register j

Keep the following points in mind when setting a mailbox as a receive mailbox or one-shot receive mailbox:

(1) Make sure to set the corresponding CANi message control register j (CiMCTLj) to 00h before setting a mailbox as a receive mailbox or one-shot receive mailbox.
(2) When a message is received, it is stored in the first mailbox that matches the conditions, according to the reception mode setting and acceptance filter processing results. In addition, the mailbox with the lower mailbox number is given priority when storing received messages.
(3) In CAN operation mode, the CAN module does not receive the data transmitted by itself, even if the ID/mask of a receive mailbox matches the transmitted data. In self-test mode, however, the CAN module receives the transmit data. In this case, the CAN module returns an ACK.

Keep the following point in mind when setting a mailbox as a transmit mailbox or one-shot transmit mailbox:

(1) Before setting a mailbox as a transmit mailbox or one-shot transmit mailbox, make sure to set the corresponding CANi message control register j (CiMCTLj) to 00h and confirm that the mailbox is not in the middle of abort processing.

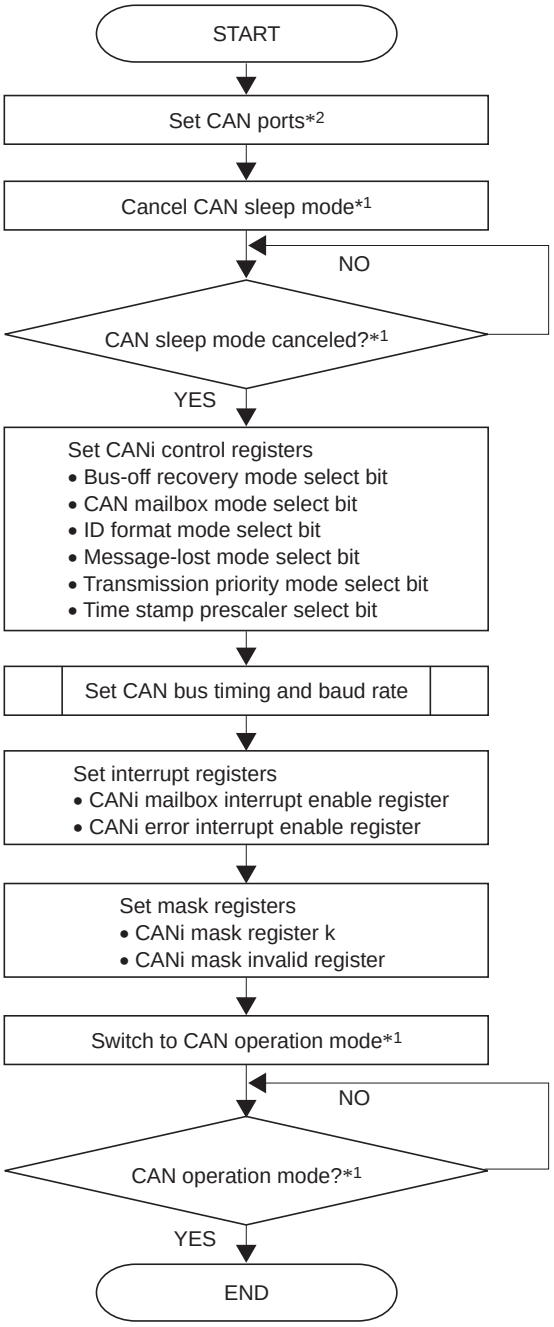
4.1 CAN Configuration

CAN configuration comprises the following three configurations:

(1) Configuration after a Hardware Reset
This configuration mode is used after a hardware reset.
(2) Configuration after CAN Reset Mode
This configuration mode is used after transitioning to CAN reset mode.
The CAN module is reset, so settings must be made again.
This configuration mode must be used if it is necessary to change the transfer speed.
(3) Configuration after CAN Halt Mode
This configuration mode is used after transitioning to CAN halt mode.
The CAN module is not reset, so settings do not need to be made again.
This configuration mode must be used if it is necessary to stop communication temporarily.

4.1.1 Configuration after a Hardware Reset

Figure 4 shows the CAN configuration procedure after a hardware reset.

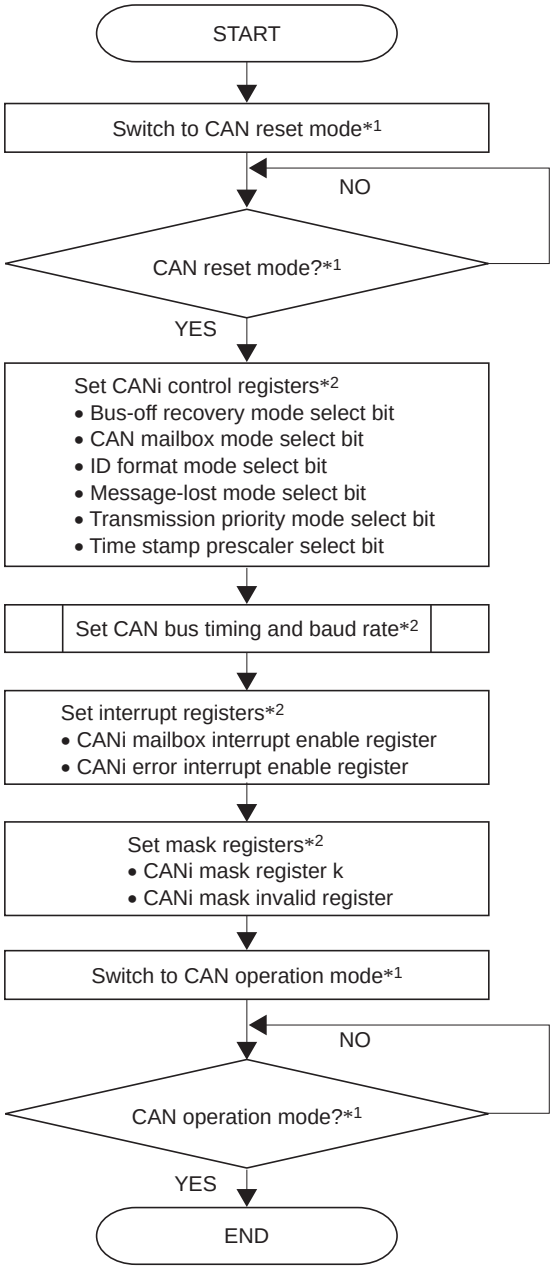


Notes: 1. When the CAN operating mode select bits and the CAN sleep mode bit are changed, check the CANi status register to ensure that the mode has been switched.
Do not change the CAN operating mode select bits and the CAN sleep mode bit until the mode has been switched.
2. Make settings to the necessary output and input function select registers. At this time, confirm that the direction registers for the ports corresponding to CANiOUT and CANiIN are set to output and input, respectively, and that the ports corresponding to CANiOUT are set to high level.

Figure 4 CAN Configuration Procedure after a Hardware Reset

4.1.2 Configuration after CAN Reset Mode

Figure 5 shows the CAN configuration procedure after transitioning to CAN reset mode.

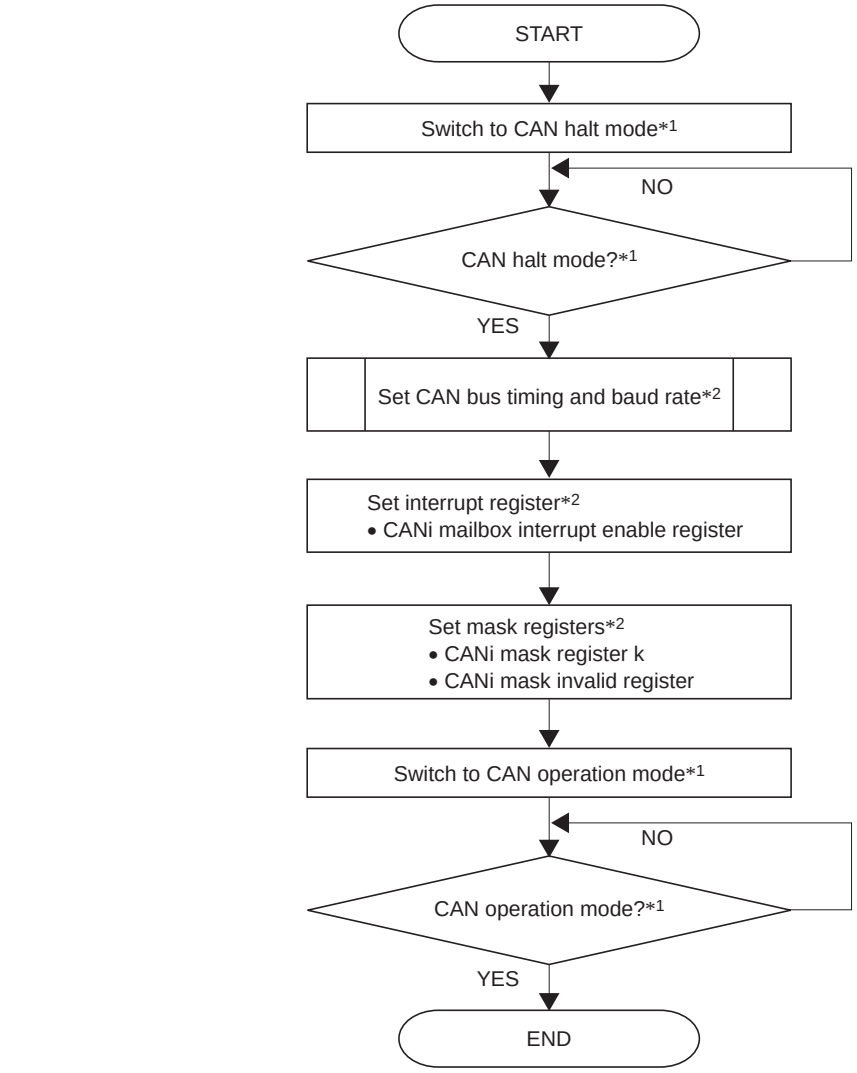


Notes: 1. When the CAN operating mode select bits and the CAN sleep mode bit are changed, check the CANi status register to ensure that the mode has been switched. Do not change the CAN operating mode select bits and the CAN sleep mode bit until the mode has been switched.
2. This step is not always necessary because these registers are not reset.

Figure 5 CAN Configuration Procedure after Transitioning to CAN Reset Mode

4.1.3 Configuration after CAN Halt Mode

Figure 6 shows the CAN configuration procedure after transitioning to CAN halt mode.



Notes: 1. When the CAN operating mode select bits are changed, check the CANi status register to ensure that the mode has been switched.
Do not change the CAN operating mode select bits until the mode has been switched.
2. This step is not always necessary because these registers are not reset after entering CAN halt mode.

Figure 6 CAN Configuration Procedure after Transitioning to CAN Halt Mode

4.2 Message Transmission

The CAN module supports 32 mailboxes for each CAN channel.

The following two transmit modes are supported. The all mailboxes can be used for transmission in transmit modes.

- Normal transmission mode
- One-shot transmission mode

(1) Normal Transmission Mode

When a mailbox is set to normal transmission mode, data frames or remote frames set to that mailbox can be transmitted.

It is possible to confirm whether or not a normal transmission completed successfully by checking the transmission complete flag (SENTDATA) of the corresponding mailbox. The transmission complete flag (SENTDATA) is set to 1 when a normal transmission completes successfully. When the mailbox loses an arbitration conflict or an error occurs during transmission, the message is retained (the CAN module transmits the message again).

(2) One-Shot Transmission Mode

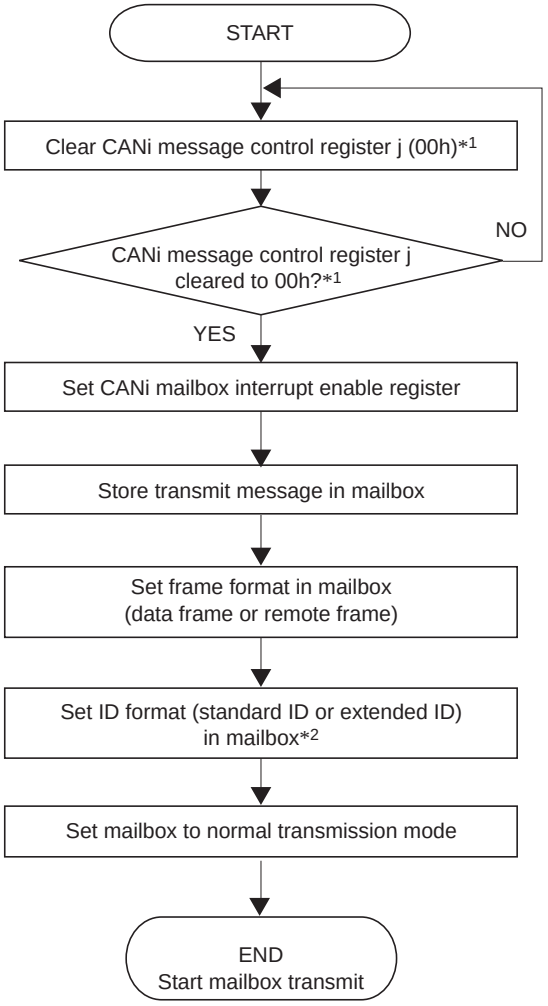
When a mailbox is set to one-shot transmission mode, data frames or remote frames set to that mailbox can be transmitted. When the one-shot enable bit (ONESHOT) is set to 1, the mailbox attempts to transmit a message one time only (The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration lost occurs) .

It is possible to confirm whether or not a one-shot transmission completed successfully by checking the transmission complete flag (SENTDATA) or the transmission abort complete flag (TRMABT) of the corresponding mailbox. The transmission complete flag (SENTDATA) is set to 1 when a one-shot transmission completes successfully. The transmission abort complete flag (TRMABT) is set to 1 when the corresponding mailbox loses an arbitration conflict or an error occurs during transmission.

4.2.1 Normal Transmission Request

Figure 7 shows the normal transmission request procedure.

This process should be carried out when there is no outstanding transmission or reception request for the corresponding mailbox (CANi message control register j (CiMCTLj) = 00h and abort processing is not underway).

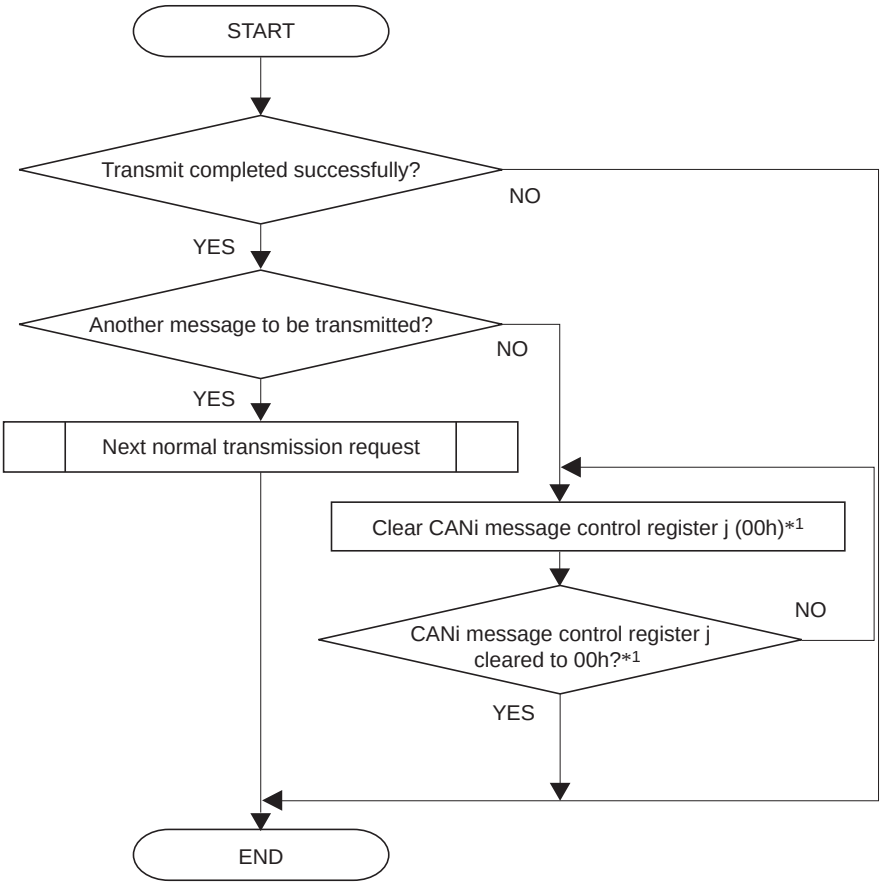


Notes: 1. To make new transmit settings after a transmission completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.
2. Set the ID extension bit (IDE) as appropriate only when the ID format mode select bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.

Figure 7 Normal Transmission Request Procedure

4.2.2 Normal Transmission Complete Processing

Figure 8 shows the necessary procedure following normal transmission complete. The required procedure is the same regardless of whether an interrupt or polling is used. For details on performing the succeeding normal transmission request, see section 4.2.1.



Notes: 1. To make new transmit settings after a transmission completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.

Figure 8 Normal Transmission Complete Procedure

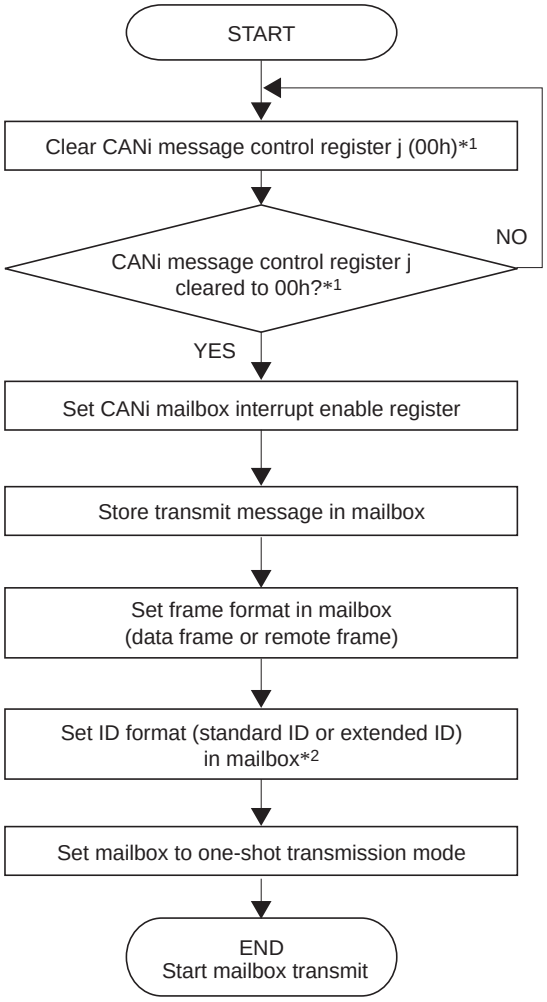
4.2.3 One-Shot Transmission Request

When the one-shot enable bit (ONESHOT) is set to 1 in transmission mode, the CAN module transmits once only from the corresponding mailbox.

Figure 9 shows the one-shot transmission request procedure.

This process should be carried out when there is no outstanding transmission or reception request for the corresponding mailbox (CANi message control register j (CiMCTLj) = 00h and abort processing is not underway).

It is possible to confirm whether or not a one-shot transmission completed successfully by checking the transmission complete flag (SENTDATA) or the transmission abort complete flag (TRMABT) of the corresponding mailbox. The transmission complete flag (SENTDATA) is set to 1 when a one-shot transmission completes successfully. The transmission abort complete flag (TRMABT) is set to 1 when the corresponding mailbox loses an arbitration conflict or an error occurs during transmission.

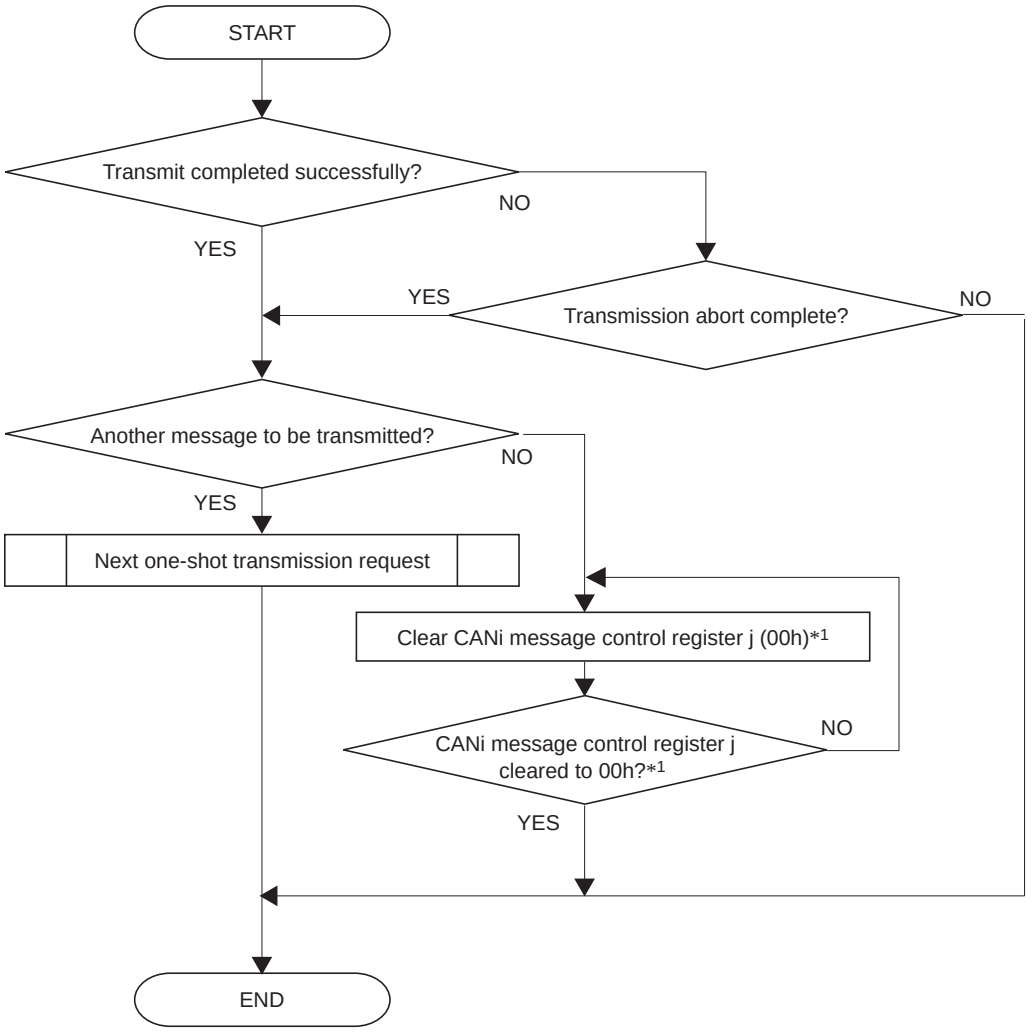


Notes: 1. To make new transmit settings after a transmission completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.
2. Set the ID extension bit (IDE) as appropriate only when the ID format mode select bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.

Figure 9 One-Shot Transmission Request Procedure

4.2.4 One-Shot Transmission Complete Processing

Figure 10 shows the necessary procedure following one-shot transmission complete. This procedure requires that polling be performed. An interrupt cannot be used because no CANi transmission complete interrupt is generated when transmission terminates due to loss of an arbitration conflict or an error. For details on performing the succeeding one-shot transmission request for a mailbox set to one-shot transmission mode, see section 4.2.3.



Notes: 1. To make new transmit settings after a transmission completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.

Figure 10 One-Shot Transmission Complete Procedure

4.2.5 Transmission-Abort

When two or more nodes start transmission simultaneously, the node(s) whose message(s) have lower CAN ID priority lose the arbitration conflict. (The message is aborted in the case of a one-shot transmission and retained in the case of a normal transmission (retransmission).) Transmission of a message cannot complete successfully unless the message wins the arbitration conflict or is transmitted when the CAN bus is idle. The transmission-abort function enables discarding of messages that are being retransmitted in this type of circumstance. It is possible to confirm whether or not transmission-abort completed successfully by checking the transmission complete flag (SENTDATA) or the transmission abort complete flag (TRMABT) of the corresponding mailbox.

The transmission complete flag (SENTDATA) is set to 1 when a transmission completes successfully.

The transmission abort complete flag (TRMABT) is set to 1 in the following case.

- Following a transmission abort request, when the transmission abort is completed before starting transmission.
- Following a transmission abort request, when the CAN module detects CAN bus arbitration lost or a CAN bus error.
- In one-shot transmission mode (Receive mailbox set bit (RECREQ) = 0, Transmit mailbox set bit (TRMREQ) = 1, and One-shot enable bit (ONESHOT) = 1), when the CAN module detects CAN bus arbitration lost or a CAN bus error.

The transmission abort complete flag (TRMABT) is not set to 1 when data transmission is completed. In this case, the transmission complete flag (SENTDATA) is set to 1.Using the transmission-abort function is effective in cases such as when only limited time may be allocated for the transmission of a single message or when it is necessary to transmit an urgent, high-priority message.

Figure 11 shows application examples using the transmission-abort function, and figure 12 shows the transmission-abort procedure.

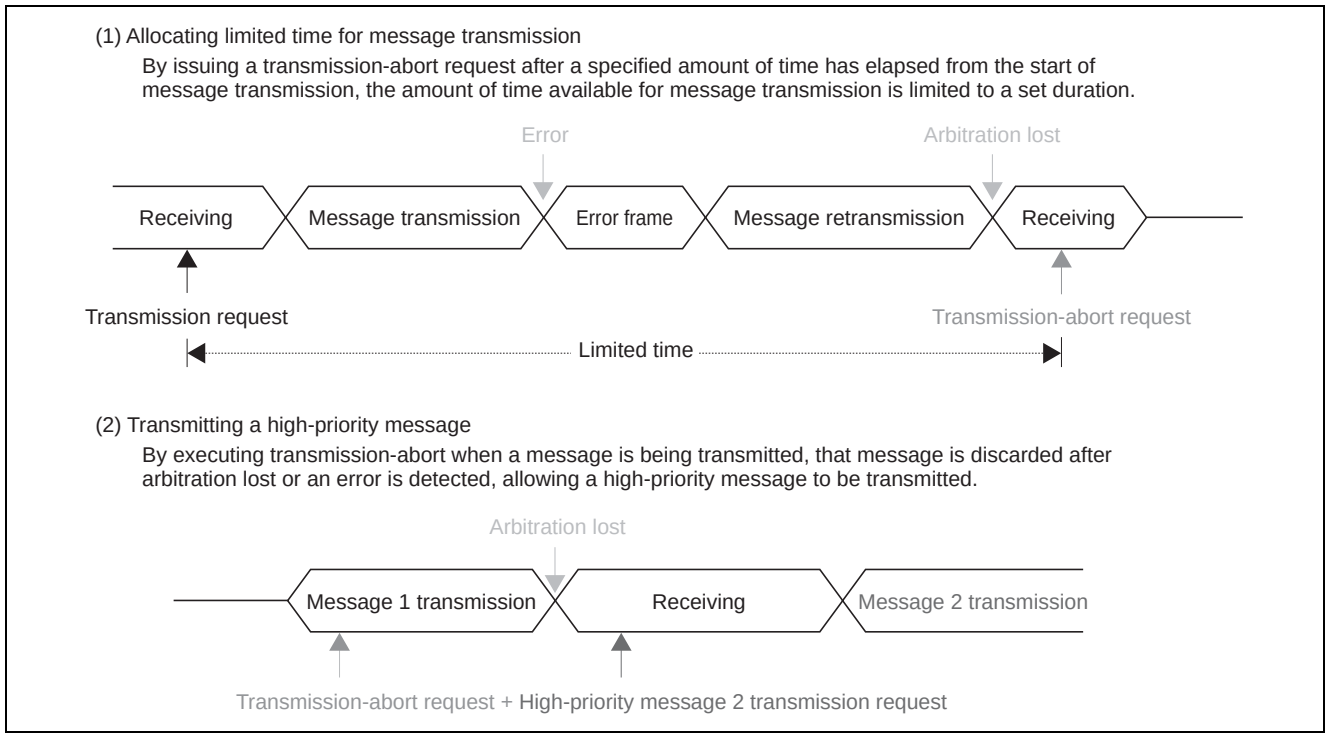
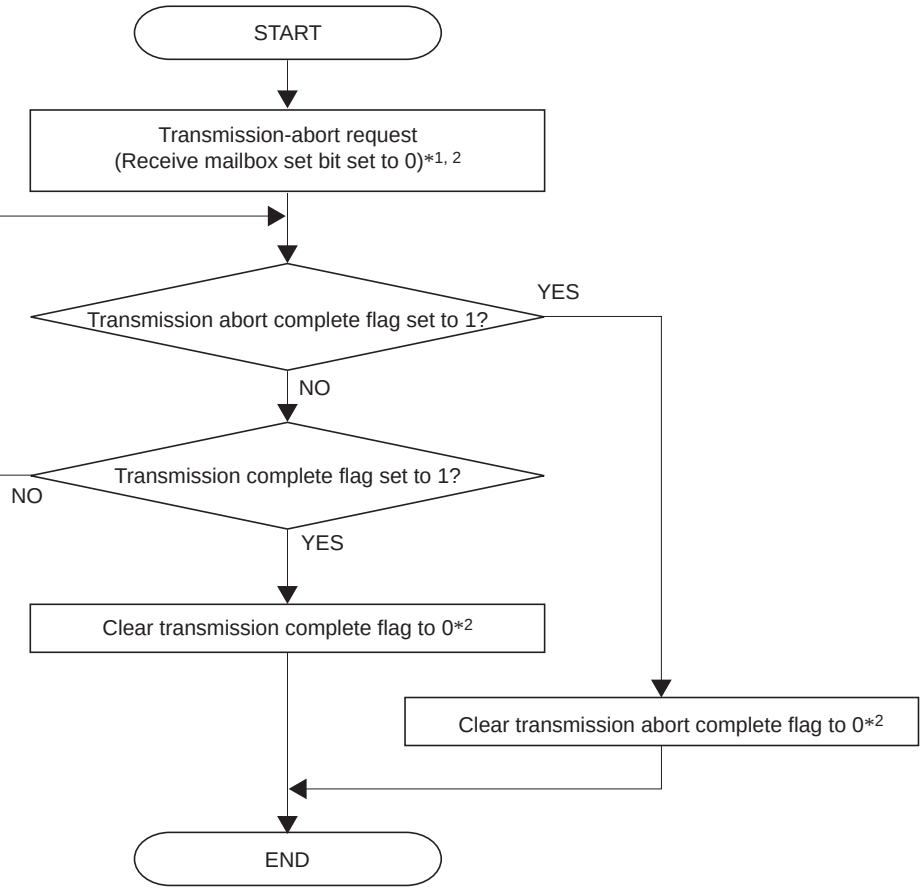


Figure 11 Transmission-Abort Function Application Examples



Notes:1. Do not change the one-shot setting during this process. (If the one-shot enable bit is set to 1, do not clear it to 0.)
2. When writing 0 to the reception complete flag, the transmission complete flag, the message lost flag, the transmission abort complete flag, the receive mailbox set bit, and the transmit mailbox set bit by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.

Figure 12 Transmission-Abort Procedure

4.3 Message Reception

The CAN module supports 32 mailboxes for each CAN channel. Received message are always stored in the lowest-numbered mailbox among the mailboxes set to the matching ID. The receiving mailbox can be selected by using an acceptance filter. See section 9 for details on the acceptance filter function.

The following two receive modes are supported: The all mailboxes can be used for reception in receive modes.

- Normal reception mode
- One-shot reception mode

(1) Normal Reception Mode

When a mailbox is set to normal reception mode, data frames or remote frames with the same ID as the mailbox's ID setting (combined with the results of the applicable acceptance filter) can be received. If two or more mailboxes set to normal reception mode have the same ID, received messages are always stored in the lowest-numbered mailbox among the mailboxes set to the matching ID. This means it is possible that an overwrite or overrun may occur. (The overwrite mode or overrun mode can be selected by the message lost mode select bit (MLM).)

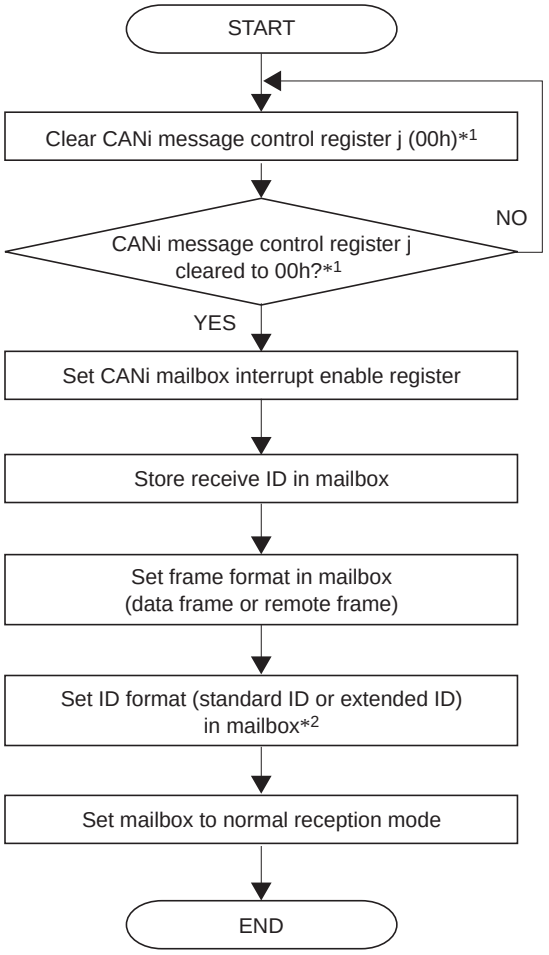
(2) One-Shot Reception Mode

When a mailbox is set to one-shot reception mode, data frames or remote frames with the same ID as the mailbox's ID setting (combined with the results of the applicable acceptance filter) can be received. When the one-shot enable bit (ONESHOT) is set to 1, the CAN module transmits a message only one time. If two or more mailboxes set to one-shot reception mode have the same ID, received messages are stored starting from the lowest-numbered mailbox and then working upward. In other words, the first received message is stored in the lowest-numbered mailbox. Then, if the first message has not yet been processed, the next received message is stored in the second-lowest-numbered mailbox.

4.3.1 Normal Reception Request

Figure 13 shows the normal reception request procedure.

This process should be carried out when there is no outstanding transmission or reception request for the corresponding mailbox (CANi message control register j (CiMCTLj) = 00h and abort processing is not underway).



Notes: 1. To make new transmit settings after reception completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.
2. Set the ID extension bit (IDE) as appropriate only when the ID format mode bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.

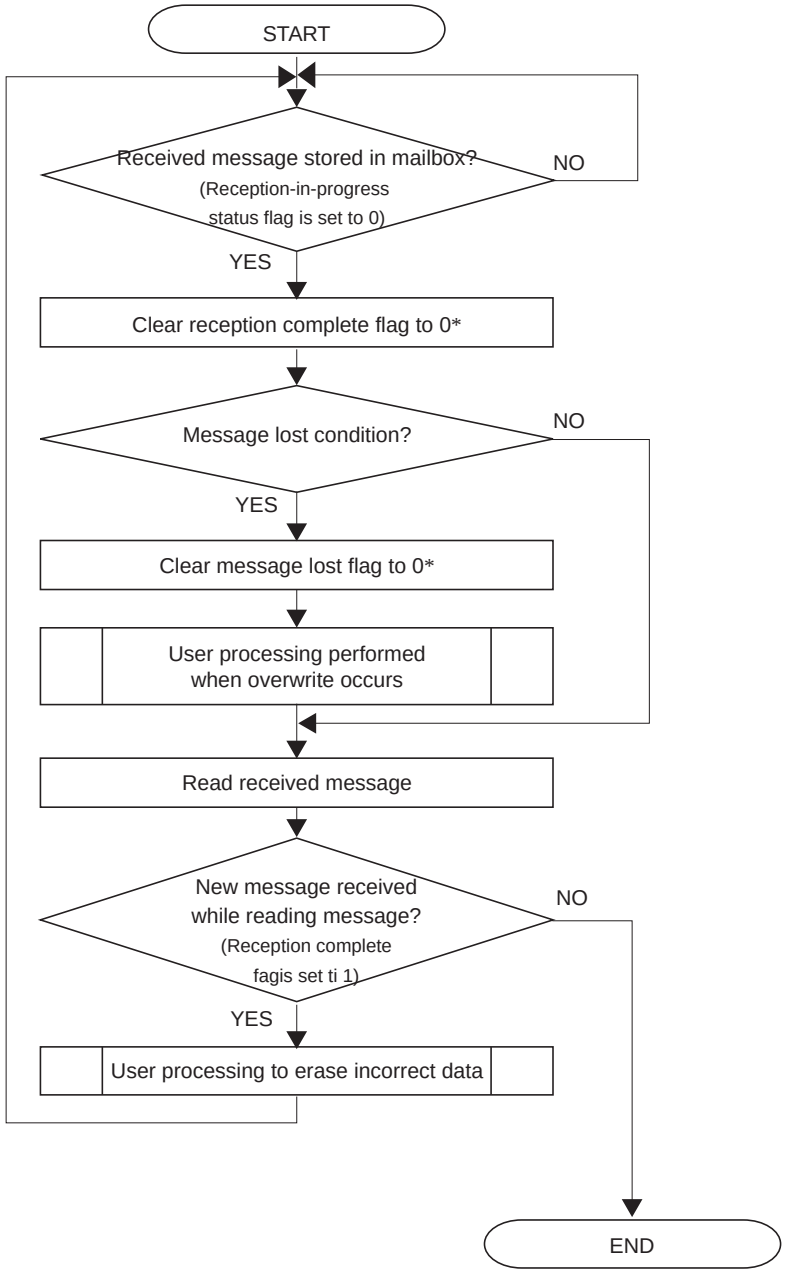
Figure 13 Normal Reception Request Procedure

4.3.2 Reception Complete (Overwrite Mode) with Mailbox Set to Normal Reception Mode

Figure 14 shows the received message processing procedure when a mailbox is set to overwrite mode (message lost mode select bit (MLM) = 0) and normal reception mode.

In this mode, when a mailbox receives a new message before processing of the previously received message by software has completed, the old message in the mailbox is overwritten by the new one. Therefore, it is necessary for the software, after reading the received message from the mailbox, to confirm that the mailbox was not overwritten while the read operation was in progress. If the mailbox was overwritten, the reception complete flag (NEWDATA) is set to 1. When a mailbox is overwritten while the reception complete flag (NEWDATA) is 1, the corresponding message lost flag (MSGLOST) is set to 1.

When using the receive mailbox search function, perform the above process after checking to determine the numbers of mailboxes with unprocessed received messages. For details on the receive mailbox search function, see section 6.1.1.



Note: * When writing 0 to the reception complete flag, the transmission complete flag, the message lost flag, the transmission abort complete flag, the receive mailbox set bit, and the transmit mailbox set bit by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.

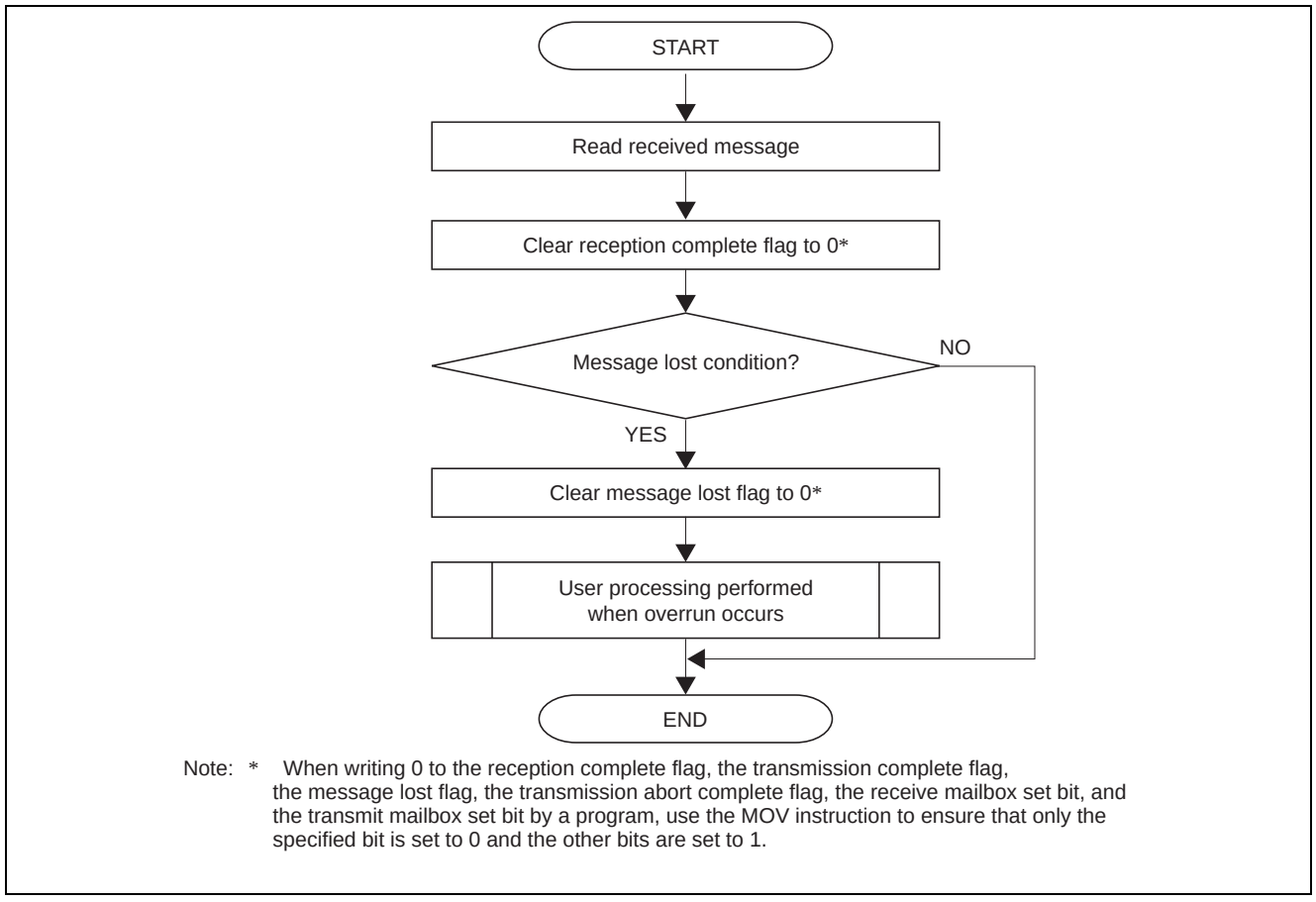
Figure 14 Message Reception Processing with Mailbox Set to Normal Reception Mode (Overwrite Mode)

4.3.3 Reception Complete (Overrun Mode) with Mailbox Set to Normal Reception Mode

Figure 15 shows the received message processing procedure when a mailbox is set to overrun mode (message lost mode select bit (MLM) = 1) and normal reception mode.

In this mode, when a mailbox receives a new message before processing of the previously received message by software has completed, the new message is discarded (not stored in the mailbox). In this case, the message lost flag (MSGLOST) corresponding to the mailbox is set to 1 and an overrun interrupt is generated (if the overrun interrupt is enabled).

When using the receive mailbox search function, perform the above process after checking to determine the numbers of mailboxes with unprocessed received messages. For details on the receive mailbox search function, see section 6.1.1.



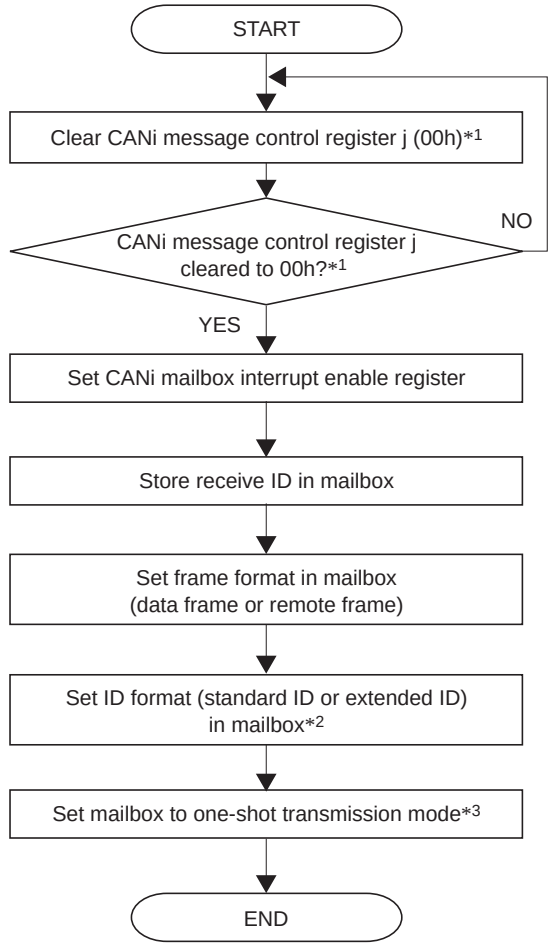
Note: * When writing 0 to the reception complete flag, the transmission complete flag, the message lost flag, the transmission abort complete flag, the receive mailbox set bit, and the transmit mailbox set bit by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.

Figure 15 Message Reception Processing with Mailbox Set to Normal Reception Mode (Overrun Mode)

4.3.4 One-Shot Reception Request

Figure 16 shows the one-shot reception request procedure.

This process should be carried out when there is no outstanding transmission or reception request for the corresponding mailbox (CANi message control register j (CiMCTLj) = 00h and abort processing is not underway).



Notes: 1. To make new transmit settings after reception completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.
2. Set the ID extension bit (IDE) as appropriate only when the ID format mode bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.
3. Set the One-shot enable bit to 1 at the same time as setting the receive mailbox setting bit to 1.

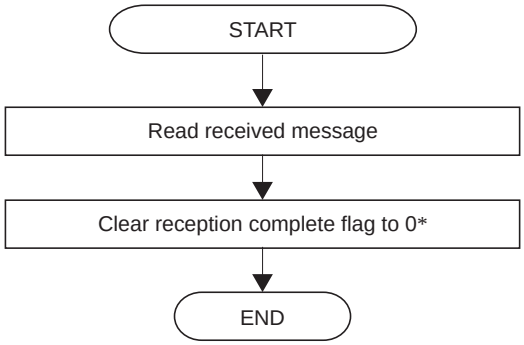
Figure 16 One-Shot Reception Request Procedure

4.3.5 Reception Complete with Mailbox Set to One-Shot Reception Mode

Figure 17 shows the received message processing procedure when a mailbox is set to one-shot reception mode.

In one-shot reception mode, the setting of the message lost mode select bit (MLM) is irrelevant and a message lost condition is never generated. This is because a mailbox that has received a message receives no further messages until the reception complete flag (NEWDATA) is cleared to 0.

When using the receive mailbox search function, perform the above process after checking to determine the numbers of mailboxes with unprocessed received messages. For details on the receive mailbox search function, see section 6.1.1.



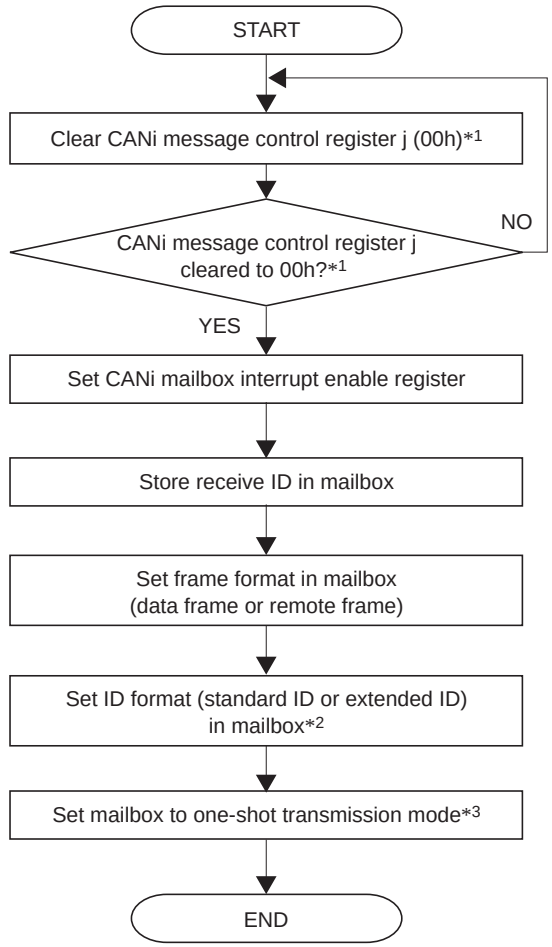
Note: * When writing 0 to the reception complete flag, the transmission complete flag, the message lost flag, the transmission abort complete flag, the receive mailbox set bit, and the transmit mailbox set bit by a program, use the MOV instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.

Figure 17 Message Reception Processing with Mailbox Set to One-Shot Reception Mode

4.3.6 Reception-Abort

A reception-abort is executed when the receive mailbox set bit (RECREQ), reception complete flag (NEWDATA), and message lost flag (MSGLOST) in the CANi message control register j (CiMCTLj) are cleared to 0 simultaneously. (Wait for the abort operation to complete before clearing the one-shot enable bit (ONESHOT) to 0.)

Figure 18 shows the reception-abort procedure.



Notes: 1. To make new transmit settings after reception completes successfully, it is necessary first to clear the CANi message control register j to 00h. Perform this step without fail.
2. Set the ID extension bit (IDE) as appropriate only when the ID format mode bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.
3. Set the One-shot enable bit to 1 at the same time as setting the receive mailbox setting bit to 1.

Figure 18 Reception Abort Procedure

5. Mailbox Modes

During CAN configuration, either of the following two modes can be selected by setting the CAN mailbox mode select bit (MBM) in the CANi control register (CiCTLR):

- Normal mailbox mode
- FIFO mailbox mode

(1) Normal Mailbox Mode
All mailboxes are set to normal transmit or receive.

(2) FIFO Mailbox Mode
Mailboxes [0] to [23] are set to normal transmit or receive, mailboxes [24] to [27] are set as a transmit FIFO, and mailboxes [28] to [31] are set as a receive FIFO.

5.1 Normal Mailbox Mode

All mailboxes are set to normal transmit or receive. Figure 19 shows the mailbox configuration in normal mailbox mode.

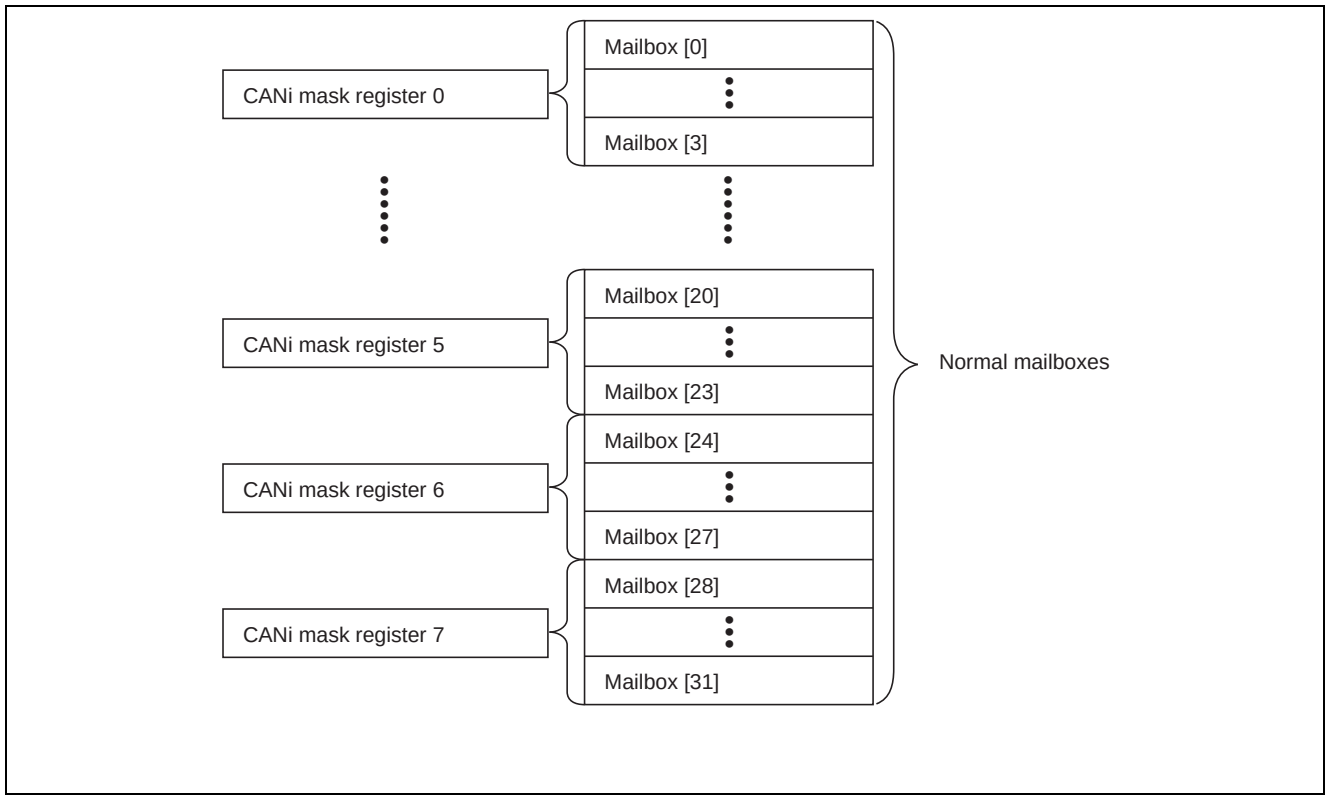


Figure 19 Mailbox Configuration in Normal Mailbox Mode

5.2 FIFO Mailbox Mode

Figure 20 shows the mailbox configuration in FIFO mailbox mode.

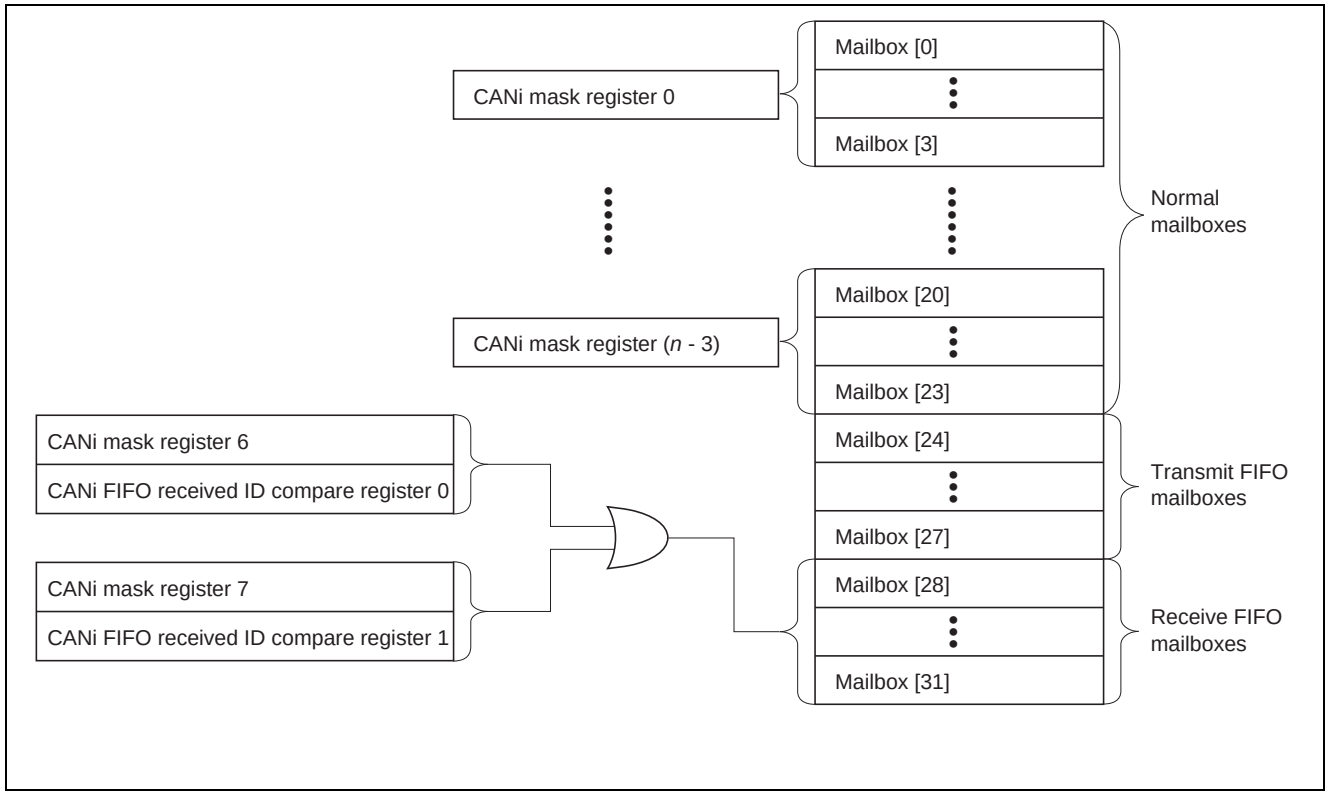


Figure 20 Mailbox Configuration in FIFO Mailbox Mode

Mailboxes [24] to [27] are set as a transmit FIFO, and mailboxes [28] to [31] are set as a receive FIFO. Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO).Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO). Updating of the CPU's pointers is accomplished by writing FFh to the CANi transmit/receive FIFO pointer control registers (CiTFPCR and CiRFPCR). Mailboxes [0] to [23] are set as normal transmit or receive mailboxes.

The following options can be selected by setting the corresponding bits in the CANi mailbox interrupt enable register (CiMIER).

- Transmit FIFO and receive FIFO interrupt disable/enable
- Transmit FIFO and receive FIFO interrupt generation source

The bits as assigned as follows:

- b24: Transmit FIFO interrupt enable bit (0: disable/1: enable)
- b25: Transmit FIFO interrupt generation timing control bit (0: every time transmission is completed /1: when the transmit FIFO becomes empty upon completion of transmission)
- b28: Receive FIFO interrupt enable bit (0: disable/1: enable)
- b29: Receive FIFO interrupt generation timing control bit (0: every time reception is completed /1: when the receive FIFO becomes buffer warning upon completion of reception*)

Note: * No interrupt is generated when the receive FIFO state changes from full to buffer warning.

The message control register j (CiMCTLj) corresponding to mailboxes [24] to [31] are not used in FIFO mailbox mode. Instead, the CANi transmit FIFO control register (CiTFCR) and CANi receive FIFO control register (CiRFCR) are used.

Setting the transmit FIFO enable bit (TFE) in the CANi transmit FIFO control register (CiTFCR) to 1 causes mailboxes [24] to [27] to function as a transmit FIFO. When the transmit FIFO enable bit (TFE) is cleared to 0, mailboxes [24] to [27] do not function as a transmit FIFO. (The transmit FIFO is halted.)

When the transmit FIFO enable bit (TFE) is cleared to 0 during transmission from a transmit FIFO mailbox, the transmit FIFO is emptied and any unsent messages it contained are lost after the next transmission complete, error, arbitration lost, or transition to CAN halt mode occurs.

The transmit FIFO mailboxes compose a four-stage FIFO. When there are no messages in the transmit FIFO, the transmit FIFO empty status bit (TFEST) in the CANi transmit FIFO control register (CiTFCR) is set to 1. When all four of the mailboxes composing the transmit FIFO contain messages (in other words, if there are four unsent messages), the transmit FIFO full status bit (TFFST) in the CANi transmit FIFO control register (CiTFCR) is set to 1.

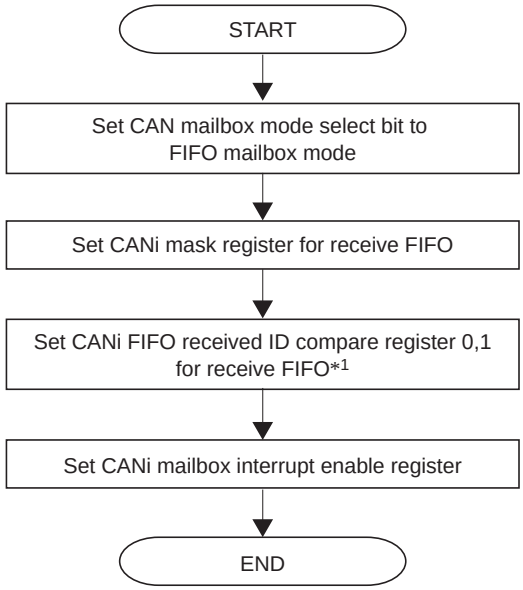
Setting the receive FIFO enable bit (RFE) in the CANi receive FIFO control register (CiRFCR) to 1 causes mailboxes [28] to [31] to function as a receive FIFO. When the receive FIFO enable bit (RFE) is cleared to 0, mailboxes [28] to [31] do not function as a receive FIFO. (The receive FIFO is halted.)

The receive FIFO mailboxes compose a four-stage FIFO. When there are no messages in the receive FIFO, the receive FIFO empty status bit (RFEST) in the CANi receive FIFO control register (CiRFCR) is set to 1. When three of the four mailboxes composing the transmit FIFO contain messages, the receive FIFO buffer warning status bit (RFWST) in the CANi receive FIFO control register (CiRFCR) is set to 1. When all four of the mailboxes composing the receive FIFO contain messages, the receive FIFO full status bit (RFFST) in the CANi receive FIFO control register (CiRFCR) is set to 1. Furthermore, when a new message is received while the receive FIFO is full, the receive FIFO message lost flag (RFMLF) in the CANi receive FIFO control register (CiRFCR) is set to 1. In this case, the new message is discarded (not stored in the mailbox) if the message lost mode select bit (MLM) in the CANi control register (CiCTLR) is set to overrun mode. If the message lost mode select bit (MLM) is set to overwrite mode, the first message received among those stored in the receive FIFO is overwritten by the new message. (The receive pointer are incremented automatically.)

FIFO mailbox mode uses two mask registers(CiMKR6 and CiMKR7) and two FIFO received ID compare registers (CiFIDCR0 and CiFIDCR1). . If the ID of a message matches the combined results of one of the two mask registers and one of the two FIFO received ID compare registers, it is stored in the receive FIFO.

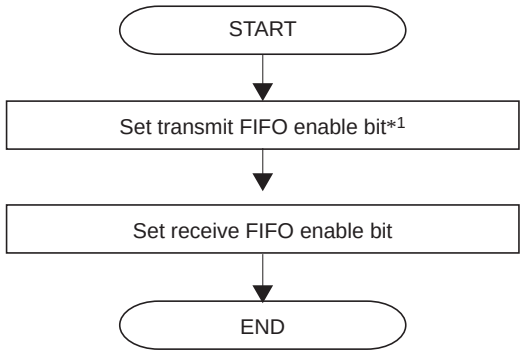
5.2.1 FIFO Mailbox Mode Settings

Settings for FIFO mailbox mode are made by carrying out procedures in CAN reset mode and CAN operation mode. Figure 21 shows the procedure in CAN reset mode, and figure 22 shows the procedure in CAN operation mode.



Notes: 1. Set the ID extension bit (IDE) as appropriate only when the ID format mode select bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.

Figure 21 FIFO Mailbox Mode Setting Procedure in CAN Reset Mode

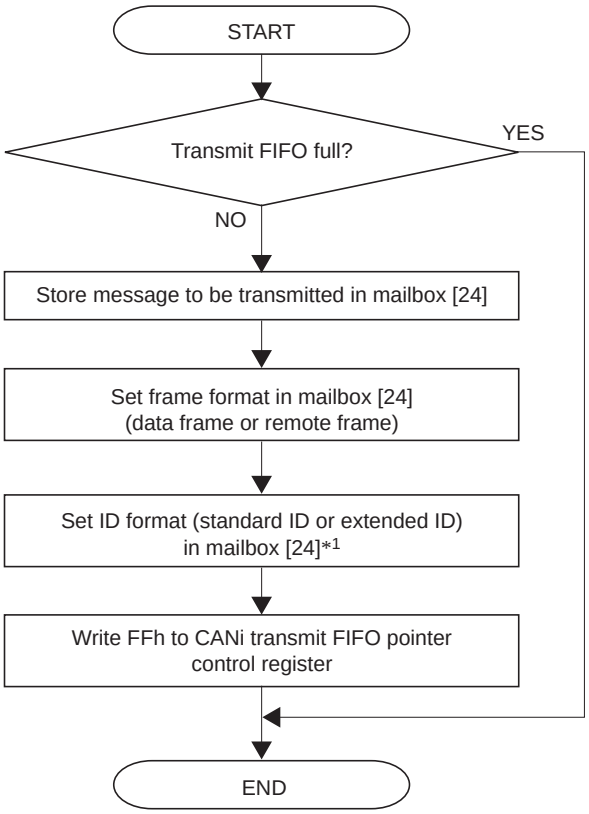


Notes: 1. When carrying out this process after disabling the transmit FIFO by clearing the transmit FIFO enable bit to 0, confirm that the transmit FIFO empty status bit is set to 1 before resetting the transmit FIFO enable bit to 1.

Figure 22 FIFO Mailbox Mode Setting Procedure in CAN Operation Mode

5.2.2 FIFO Transmission

Figure 23 shows the processing procedure for transmitting a message from the FIFO.



Notes: 1. Set the ID extension bit (IDE) as appropriate only when the ID format mode bit in the CANi control register is set to mixed ID mode. Otherwise, set the ID extension bit (IDE) to 0.

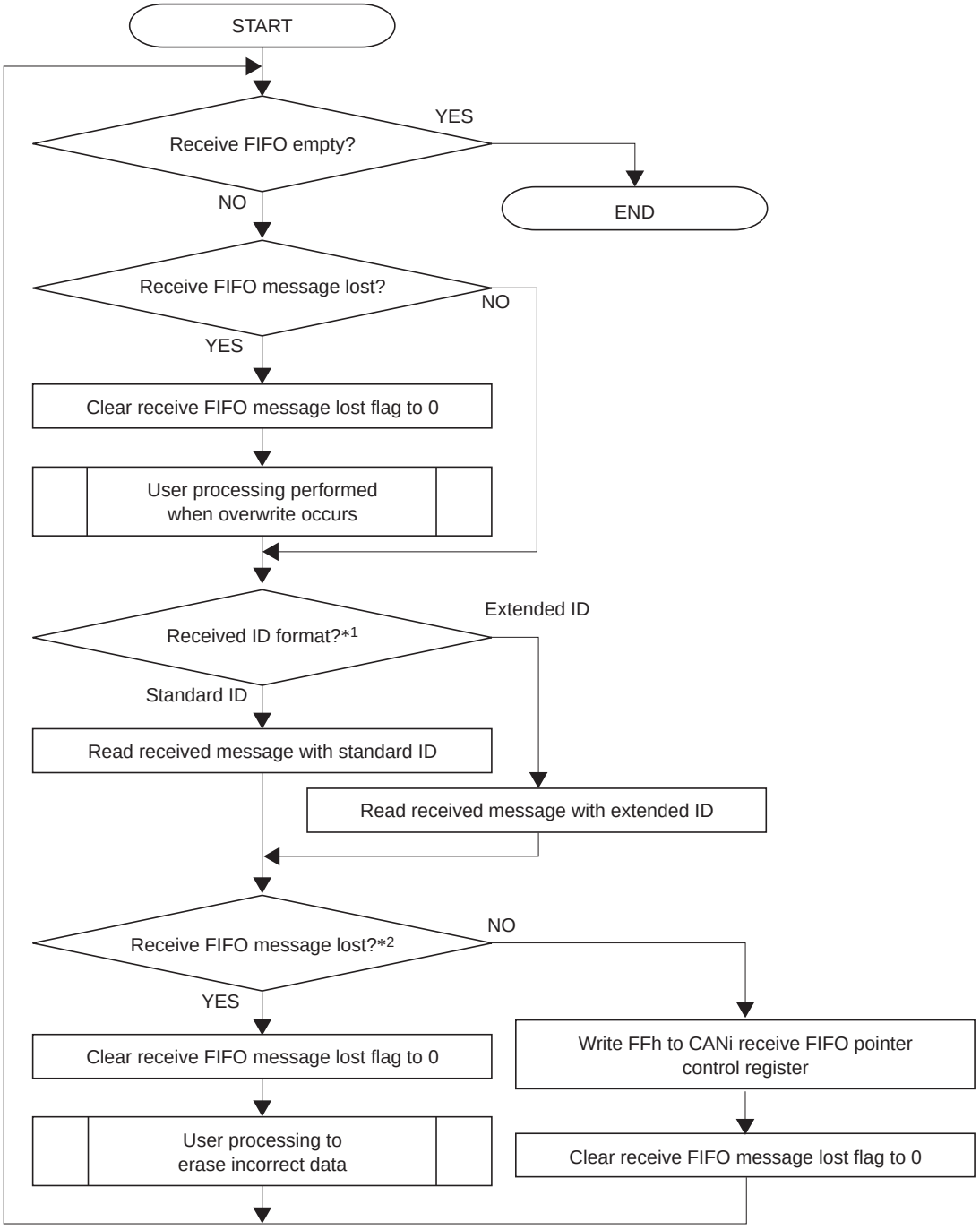
Figure 23 Transmission Processing Procedure in FIFO Mailbox Mode

The procedure for aborting transmission from a FIFO is similar to that for aborting transmission from a normal mailbox described in section 4.2.5. To abort FIFO transmission, it is necessary to clear the transmit FIFO enable bit (TFE) to 0 instead of the transmit mailbox set bit (TRMREQ). In addition, the transmit FIFO empty status bit (TFEST) is set to 1 instead of the transmission abort complete flag (TRMABT).

5.2.3 FIFO Reception (Overwrite Mode)

Figure 24 shows the processing procedure for receiving a message in the FIFO using overwrite mode.

In overwrite mode it is necessary to consider the possibility of a message being overwritten while it is being read. If a message is overwritten while it is being read, it cannot be used as a normal message.



Notes: 1. This determination is necessary only when the ID format mode bit in the CANi control register is set to mixed ID mode.
2. This determines whether or not the message was overwritten while the data was being read. If a message lost condition occurs, the message cannot be used as a normal message.

Figure 24 Reception Processing Procedure (Overwrite Mode) in FIFO Mailbox Mode

5.2.4 FIFO Reception (Overrun Mode)

Figure 25 shows the processing procedure for receiving a message in the FIFO using overrun mode. In overrun mode it is not necessary to consider the possibility of a message being overwritten by the CAN while it is being read by the CPU. The read value has never been overwritten, even if an overrun occurs.

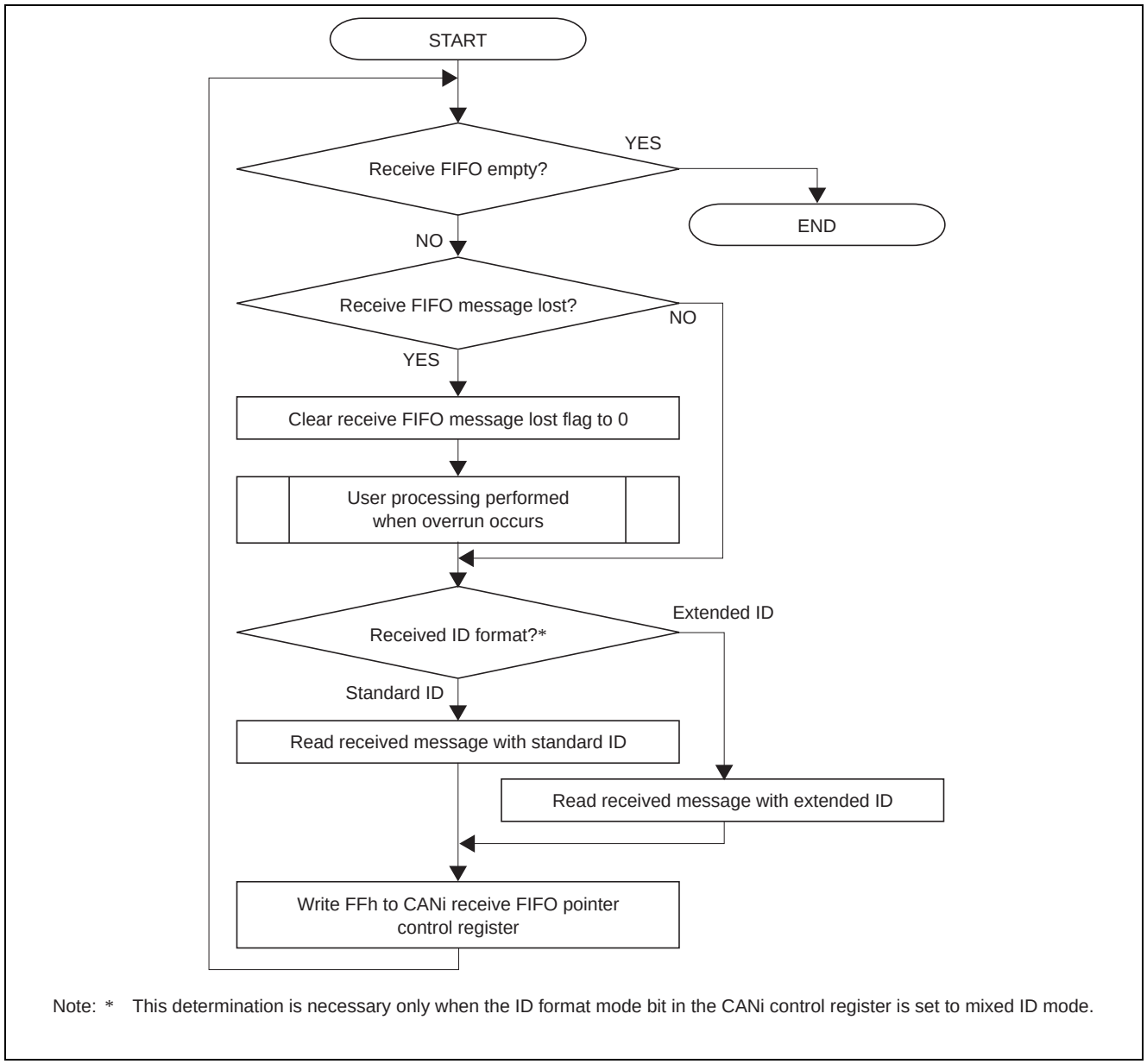


Figure 25 Reception Processing Procedure (Overrun Mode) in FIFO Mailbox Mode

6. Mailbox Search Function

Usually it is necessary to search for the mailbox number after each transmission or reception operation completes when two or more transmit or receive mailboxes have been set up. This means that the load on the software grows as the number of mailboxes increases.

The mailbox search function can be used to reduce the load on the software.

The mailbox search mode provides an easy way to search for the mailbox number that triggered a reception complete or transmission complete message.

The following four modes are used for mailbox searches:

- Receive mailbox search mode
- Transmit mailbox search mode
- Message lost search mode
- Channel search mode

These modes can be used in both normal mailbox mode and FIFO mailbox mode.

For polling processing, checking the CANi status register (CiSTR) is recommended before using the mailbox search function.

Figure 26 shows an example.

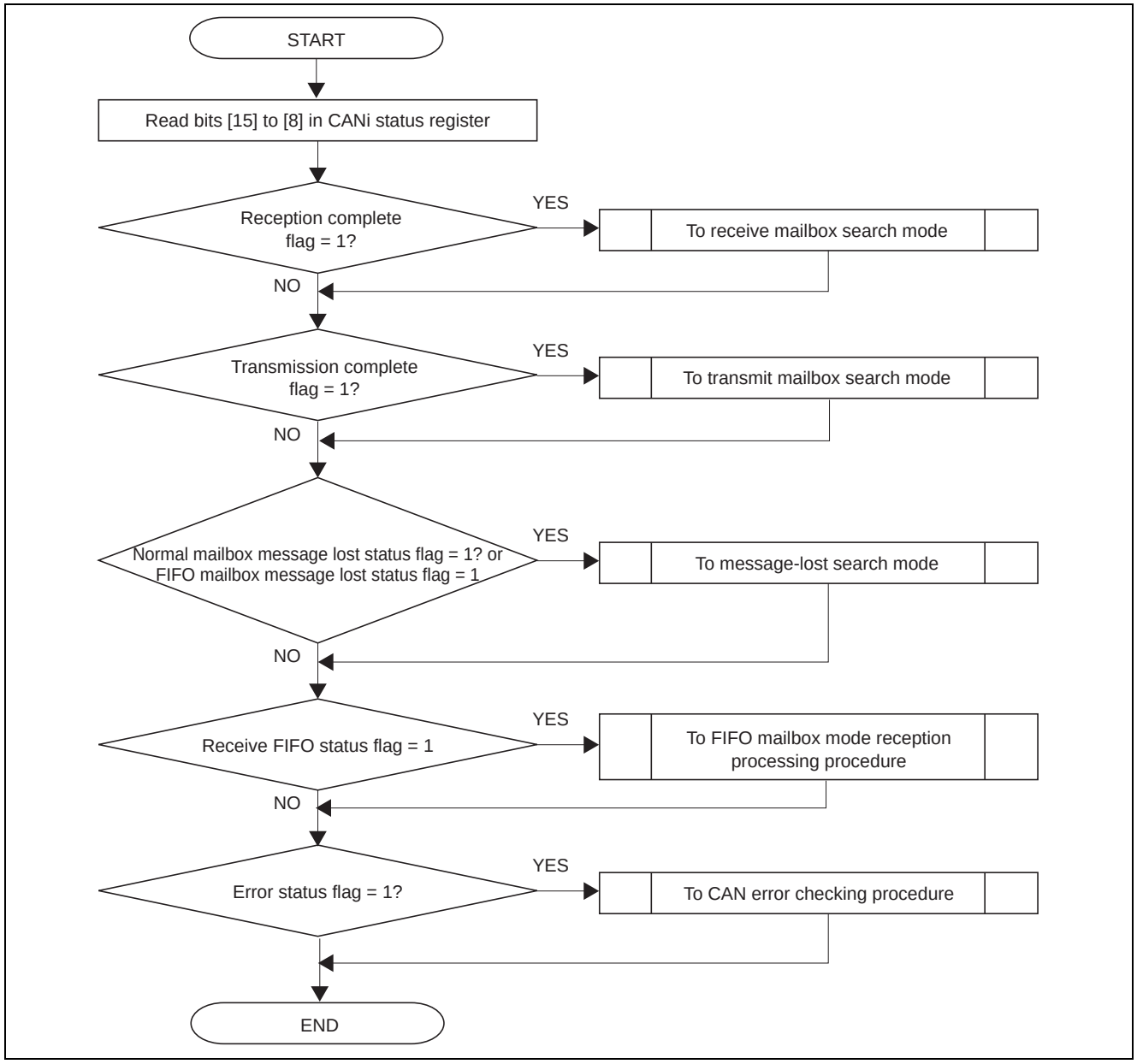


Figure 26 CANi Status Register Checking Example (Polling Operation)

6.1 Using the Mailbox Search Function

Table 4 lists the setting values of the mailbox search mode select bits (MBSM).

Table 4 Setting Values of Mailbox Search Mode Bits

CANi Mailbox Search Mode Register(CiMSMR)		
b1	b0	**Search Mode**
0	0	Receive mailbox search mode (Search for reception complete flag)
0	1	Transmit mailbox search mode (Search transmission complete flag)
1	0	Message lost search mode (Search message lost flag)
1	1	Channel search mode

Note: i indicates the CAN channel.

6.1.1 Receive Mailbox Search Mode

This mode searches the lowest mailbox number in the receive-end state.

To use this mode, set the mailbox search mode select bits (MBSM) to 00b. The number of the mailbox in the reception complete state can then be read from the CANi mailbox search status register (CiMSSR). When two or more mailboxes are in the reception complete state (that is, when two or more mailboxes have their reception complete flag (NEWDATA) set to 1), the lowest mailbox number is read.

The reception complete flag (NEWDATA) is cleared to 0 by software as part of the reception complete processing. After this occurs, the mailbox search function can be used again to read the next mailbox number if there are other mailboxes in the reception complete state. When no other mailboxes are in the reception complete state, the search result status bit (SEST) is set to 1.

In FIFO mailbox mode, mailbox number 28 is read for the receive FIFO mailbox when the receive FIFO empty status bit (RFEST) is cleared to 0 (message present in receive FIFO mailbox).

Figure 27 shows the procedure for using receive mailbox search.

For details on reception complete processing, see section 4.3.2, 4.3.3, 4.3.5, 5.2.3, or 5.2.4.

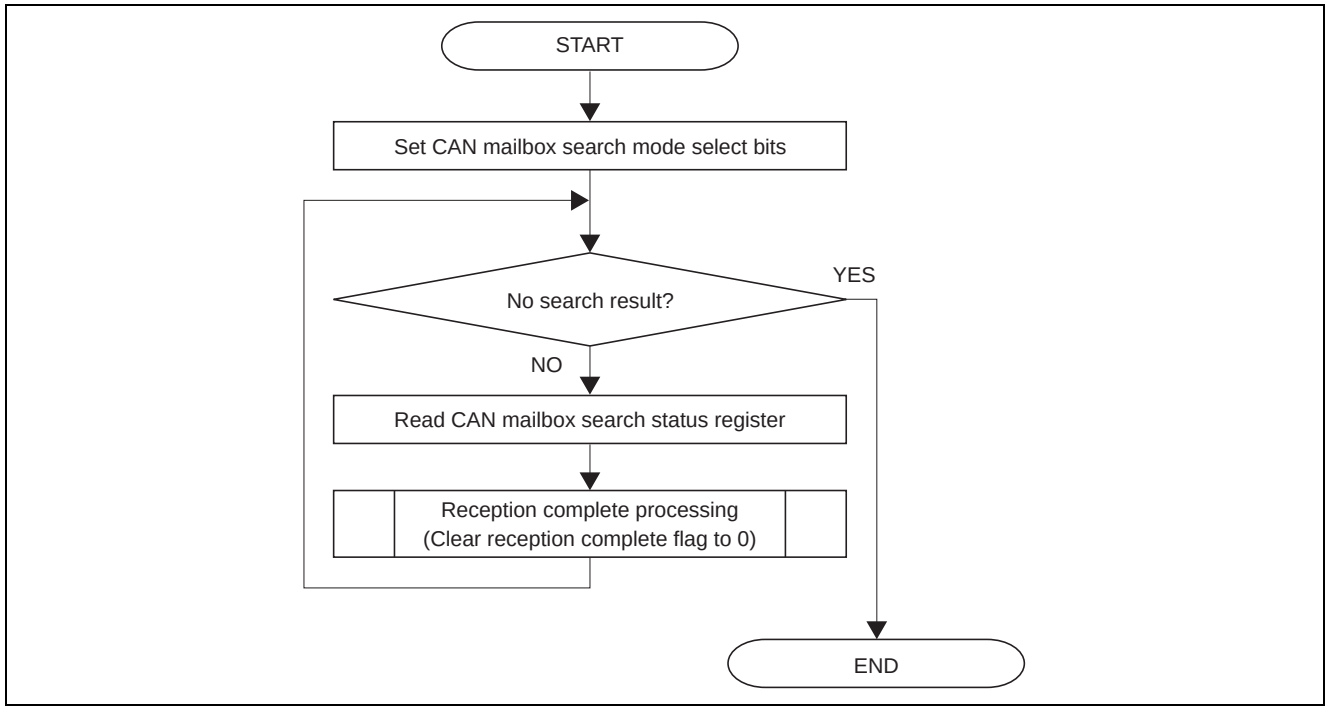


Figure 27 Receive Mailbox Search Procedure

6.1.2 Transmit Mailbox Search Mode

This mode searches the number of the mailbox that successfully completed a transmission.

To use this mode, set the mailbox search mode select bits (MBSM) to 01b. The search result can be read from the CANi mailbox search status register (CiMSSR). When two or more mailboxes have successfully entered the transmission complete state (that is, when two or more mailboxes have their transmission complete flag (SENTDATA) set to 1), the lowest mailbox number is read.

The transmission complete flag (SENTDATA) is cleared to 0 by software as part of the transmission complete processing. After this occurs, the mailbox search function can be used again to read the next mailbox number if there are other mailboxes in the transmission complete state. When no other mailboxes are in the transmission complete state, the search result status bit (SEST) is set to 1.

In FIFO mailbox mode, transmit FIFO mailboxes are not covered by the mailbox search function.

Figure 28 shows the procedure for using transmit mailbox search.

For details on continuing with a normal transmission request in normal mailbox mode, see section 4.2.1.

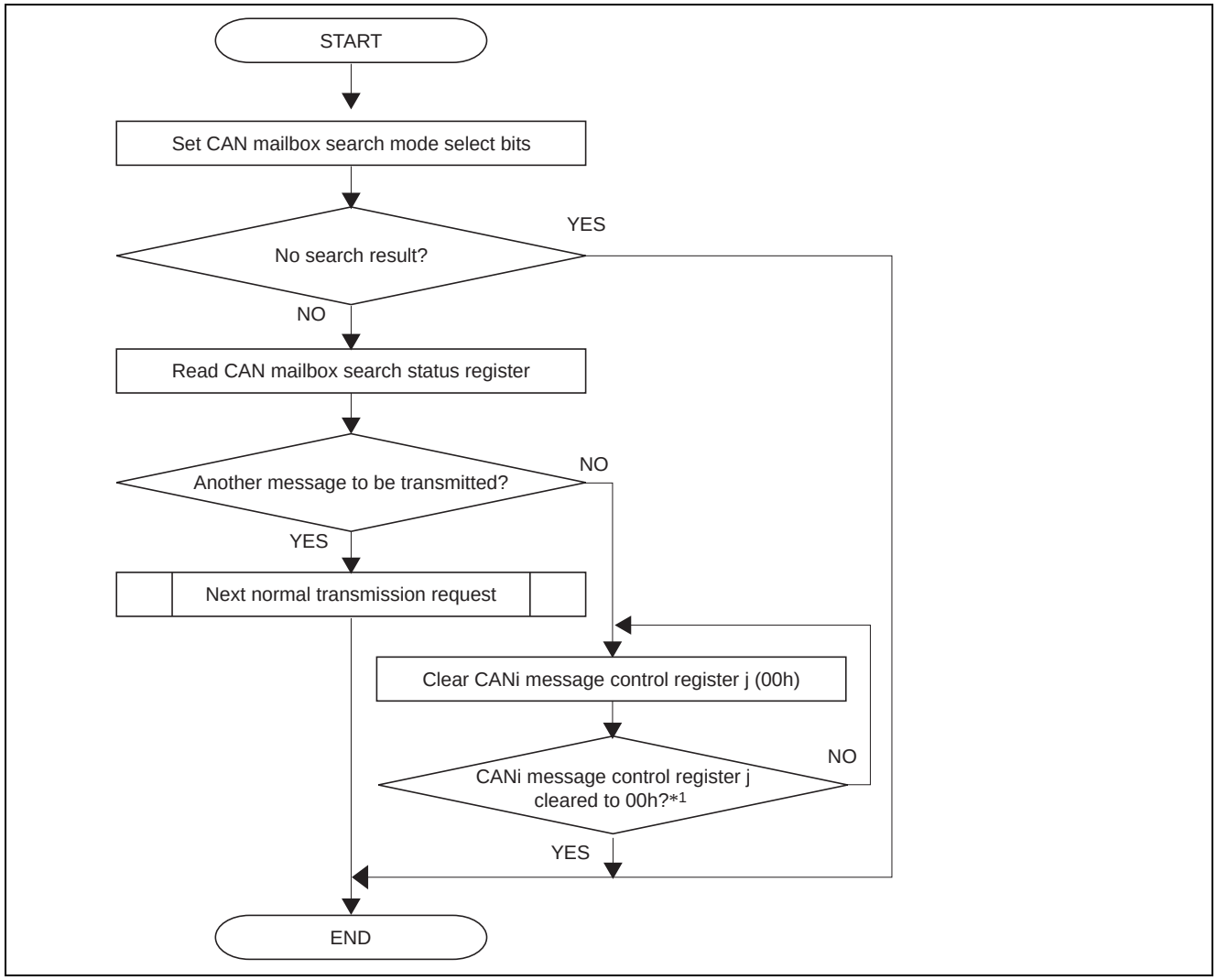


Figure 28 Transmit Mailbox Search Procedure

6.1.3 Message Lost Search Mode

This mode searches the number of the mailbox that triggered a message lost condition.

To use this mode, set the mailbox search mode select bits (MBSM) to 10b. The search result can be read from the CANi mailbox search status register (CiMSSR). When two or more mailboxes are in the message lost state (that is, when two or more mailboxes have their message lost flag (MSGLOST) or receive FIFO message lost flag (RFMLF) set to 1), the lowest mailbox number is read.

The message lost flag (MSGLOST) is cleared to 0 by software as part of the message lost processing. After this occurs, the mailbox search function can be used again to read the next mailbox number if there are other mailboxes in the message lost state. When no other mailboxes are in the message lost state, the search result status bit (SEST) is set to 1.

In FIFO mailbox mode, mailbox number 28 is read when the receive FIFO message lost flag (RFMLF) is set to 1 (message lost condition occurred in receive FIFO mailbox).

Figure 29 shows the procedure for using message lost search.

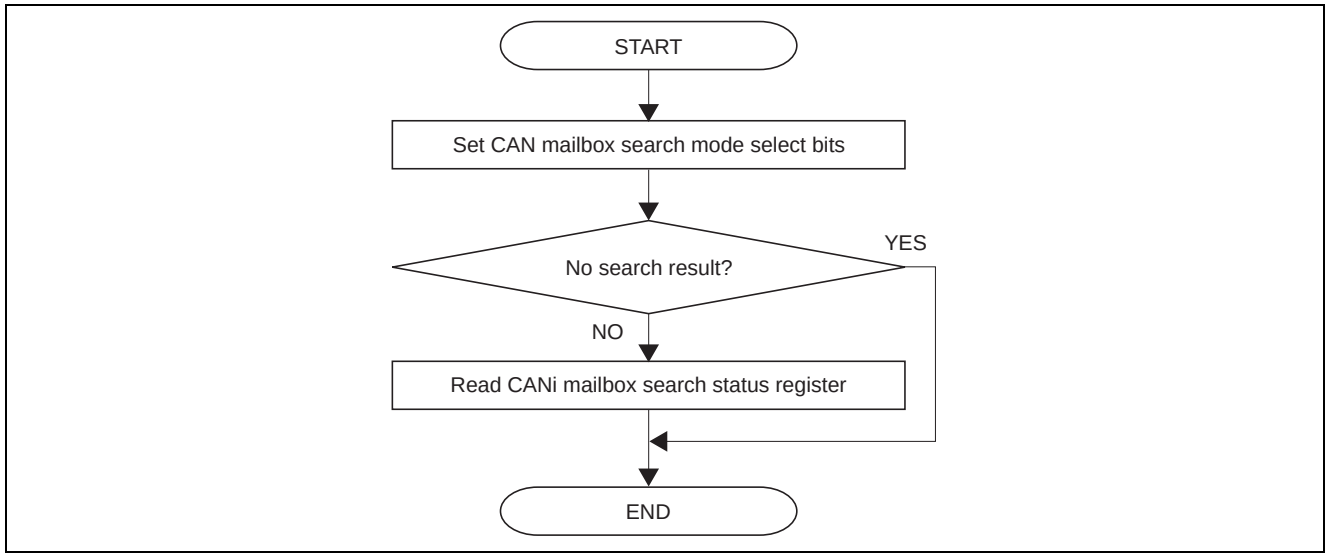


Figure 29 Message lost Search Procedure

6.1.4 Channel Search Mode

The purpose and procedure for using the channel search mode differ from those of the other three modes. This mode does not search for a mailbox number. To use this mode, set the mailbox search mode select bits (MBSM) to 11b.

Set the channel search value (table value) in the CANi channel search support register (CiCSSR). The encoded value can then be read from the CANi mailbox search status register (CiMSSR). When there are two or more channels, the channel numbers are read in order, starting from the lowest.

When the CANi mailbox search status register (CiMSSR) is read, the search result is updated automatically. The next channel number can then be read if there are other channels. When there are no other channels, the search result status bit (SEST) is set to 1. Figures 30 and 31 show the procedure for using channel search.

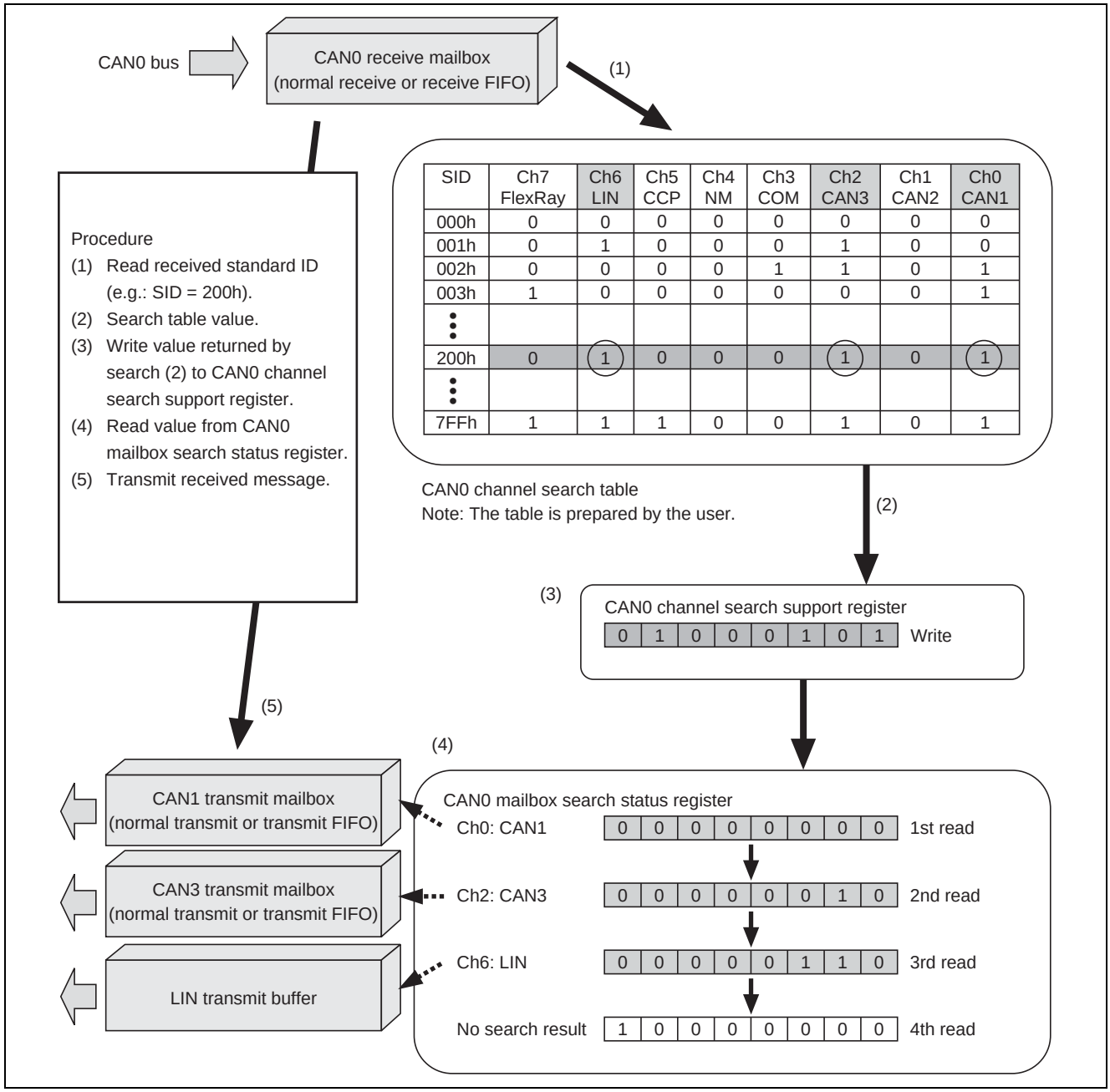


Figure 30 Outline of Channel Search Mode (When CAN0 is used)

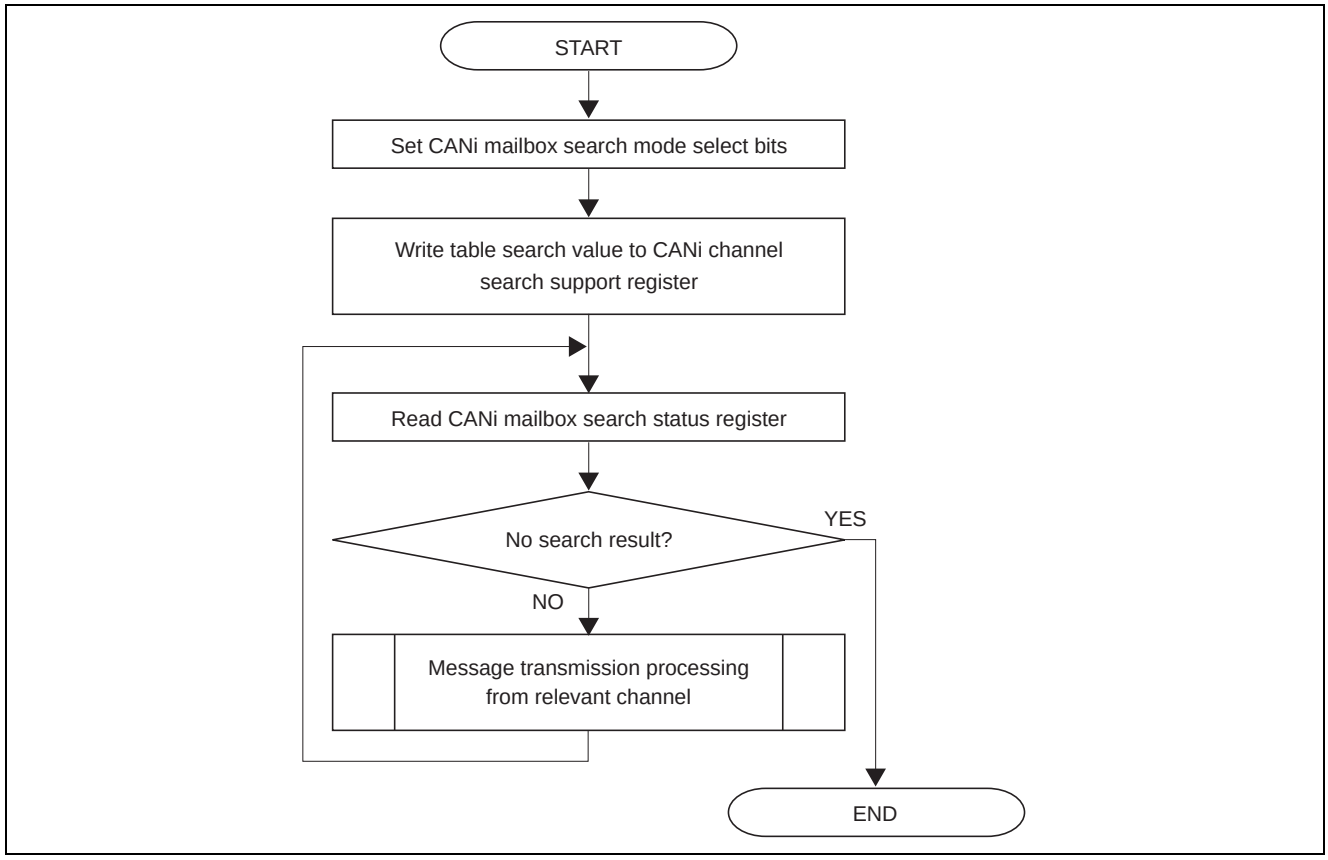


Figure 31 Channel Search Procedure

7. CAN Errors

When an error is detected due to a communication frame irregularity during transmission or reception by a mailbox, the transmit error counter or receive error counter value is incremented, depending on whether the error occurred during transmission or reception. When the transmit error counter or receive error counter value reaches 96 or greater, an error warning is detected and the error warning detect flag (EWIF) is set to 1. When the transmit error counter or receive error counter value reaches 128 or greater, the CAN status changes from error-active to error-passive, the error-passive state is detected, and the error-passive detect flag (EPIF) is set to 1. When the transmit error counter value reaches 256 or greater, the CAN module enters the bus-off state, the bus-off state is detected, and the bus-off state detect flag (BOEIF) is set to 1.

To use CANi error interrupts, set to 1 the bits in the CANi error interrupt enable register (CiEIER) corresponding to the error interrupts to be used. Whether or not a particular interrupt has occurred can then be confirmed by reading the CANi error interrupt factor judge register (CiEIFR). Only make settings in the CANi error interrupt enable register (CiEIER) when in CAN reset mode. To use CANi error interrupts, it is necessary to make settings in the CANi error interrupt control register (CiEIC) beforehand.

7.1 CAN Error Checking

(1) Using Polling of the error state

The error state of the CAN module can be checked by polling the error-passive status flag (EPST) and bus-off status flag (BOST) in the CANi status register (CiSTR).

Figure 32 shows the CANi error state checking procedure using polling.

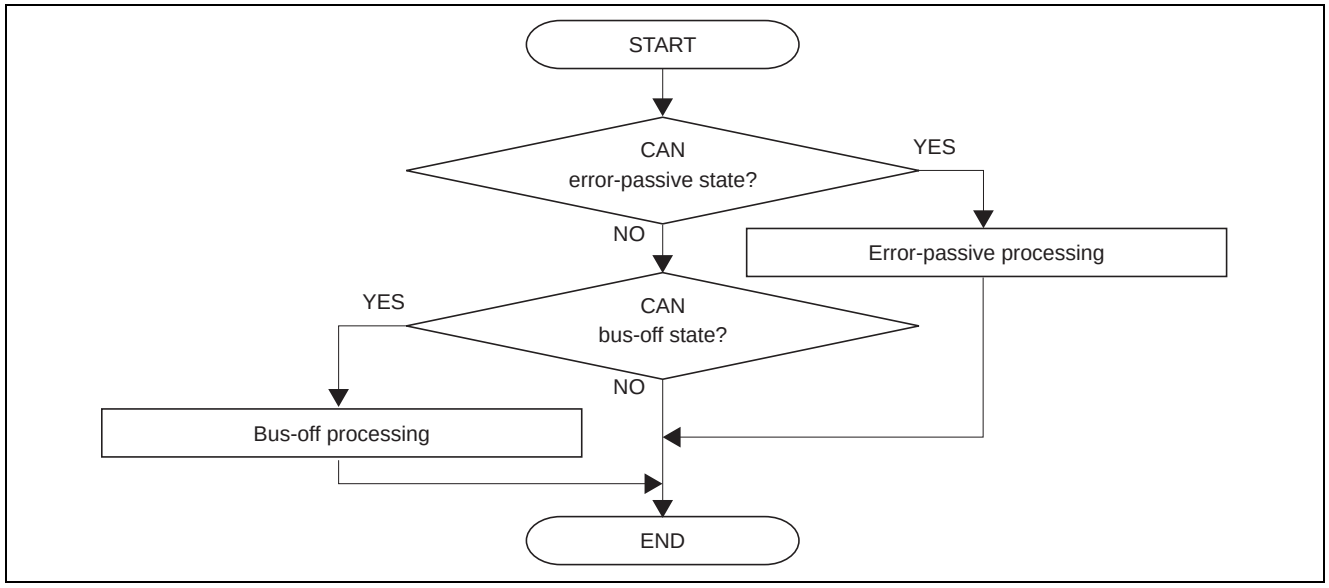
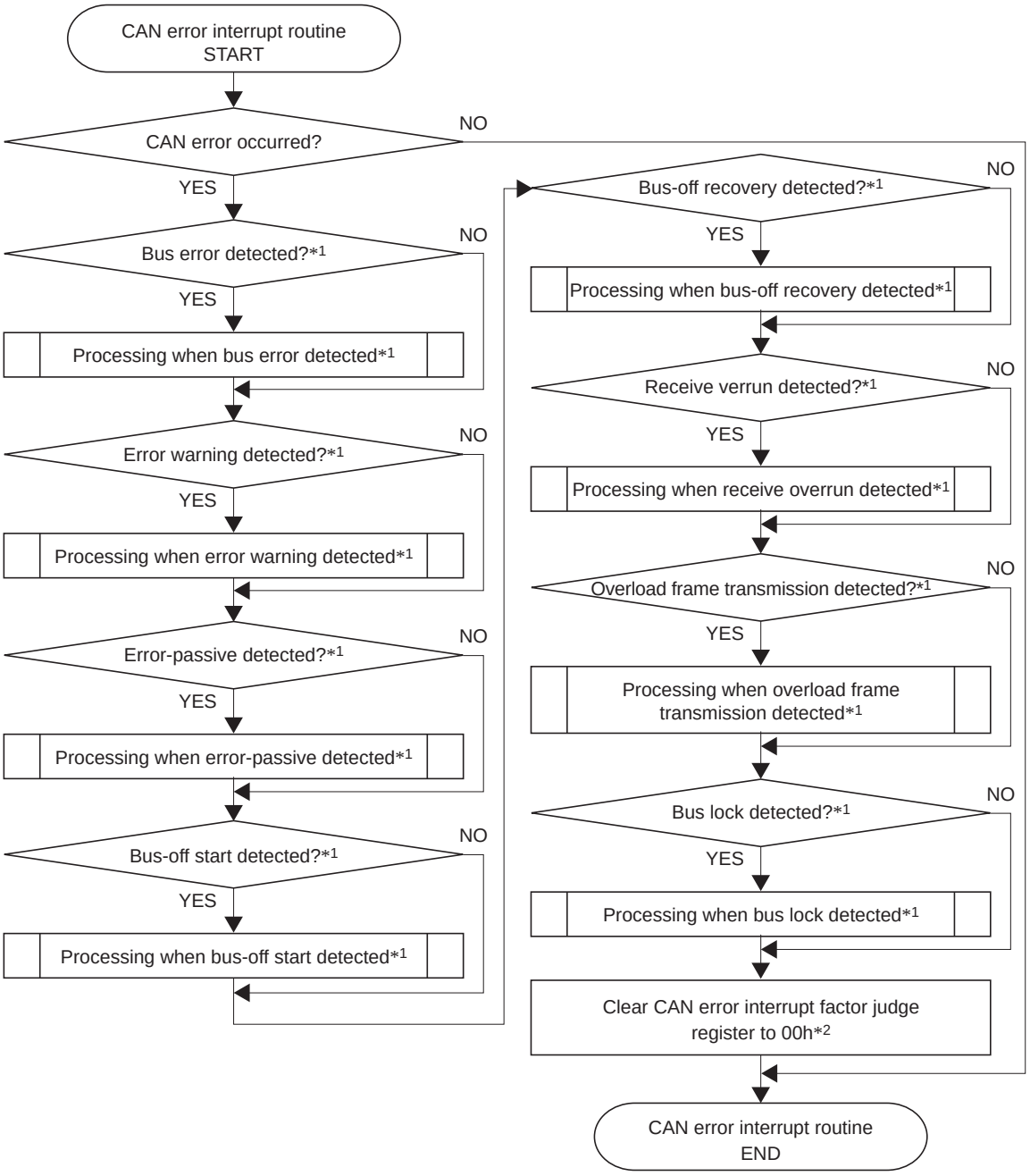


Figure 32 CAN Error State Checking Procedure (Using Polling)

(2) Using CAN Error Interrupts

CANi error interrupts can be used to check the error state by enabling interrupts in the CANi error interrupt control register (CiEIC). The CAN error state is determined by reading the CANi error interrupt factor judge register (CiEIFR) as part of the CANi error interrupt routine. Figure 33 shows the CAN error state checking procedure using CANi error interrupts. For details on returning from the bus-off state, see section 8.



Notes: 1. These processes are not required when the corresponding error interrupt is disabled.
2. Do not use the bit-clear instruction in software to clear to 0 a single bit in the CAN error interrupt source determination register. Use the MOV instruction to make sure the designated bit is cleared to 0. Writing 1 to this bit does not change its value.

Figure 33 CANi Error State Checking Procedure (Using CAN Error Interrupts)

8. Bus-Off Recovery Modes

When CAN communication errors occur repeatedly, the CAN module transitions to the bus-off state, according to the fault restriction rules in the CAN specification.

The CAN module has five modes for returning from the bus-off state. Table 5 lists these modes and their associated bits (in the CANi control register) and settings. Figure 34 shows transition to and from the bus-off state.

Table 5 Bus-Off-Return Modes

	Name	Description	Bit(s) Used	Bit Setting
(1)	Normal mode	After returning from the bus-off state, the CAN module immediately transitions to the error-active state so that CAN communication can begin without delay.*[1,2]	BOM*[5]	00b*[6]
(2)	Forcible return from bus-off	The CAN module immediately transitions to the error-active state, and CAN communication is possible.*[3]	BOM*[5]	00b*[6]
			RBOC*[5]	1*[7]
(3)	Entry to CAN halt mode automatically at bus-off start	The CAN module switches immediately to CAN halt mode when the bus-off state is reached.*[3]	BOM*[5]	01b*[6]
(4)	Entry to CAN halt mode automatically at bus-off end	The CAN module switches to CAN halt mode after returning from the bus-off state.*[1,2]	BOM*[5]	10b*[6]
(5)	Entry to CAN halt mode by a program request	When in the bus-off state, the CAN module switches immediately to CAN halt mode when the CAN operating mode select bits are set to 10b (CAN halt mode).*[3,4]	BOM*[5]	11b*[6]
			CANM*[5]	10b*[8]

Notes: 1. The CAN module returns from the bus-off state when 11 successive recessive bits are detected 128 times.
2. In this case the bus-off recovery detect flag in the CANi error interrupt source determination register is set to 1 (bus-off recovery detected).
3. In this case the bus-off recovery detect flag is not set to 1.
4. When the CAN mode bits are not set to 10b (CAN halt mode) while in the bus-off state, operation is identical to (1).
5. Bits in CANi control register.
6. Only make this setting when in CAN reset mode.
7. Only make this setting when in the bus-off state. After the forcible return from bus-off bit is set to 1 by a program, it is automatically cleared to 0.
8. After changing the CAN mode bits, make sure to check the CANi status register.

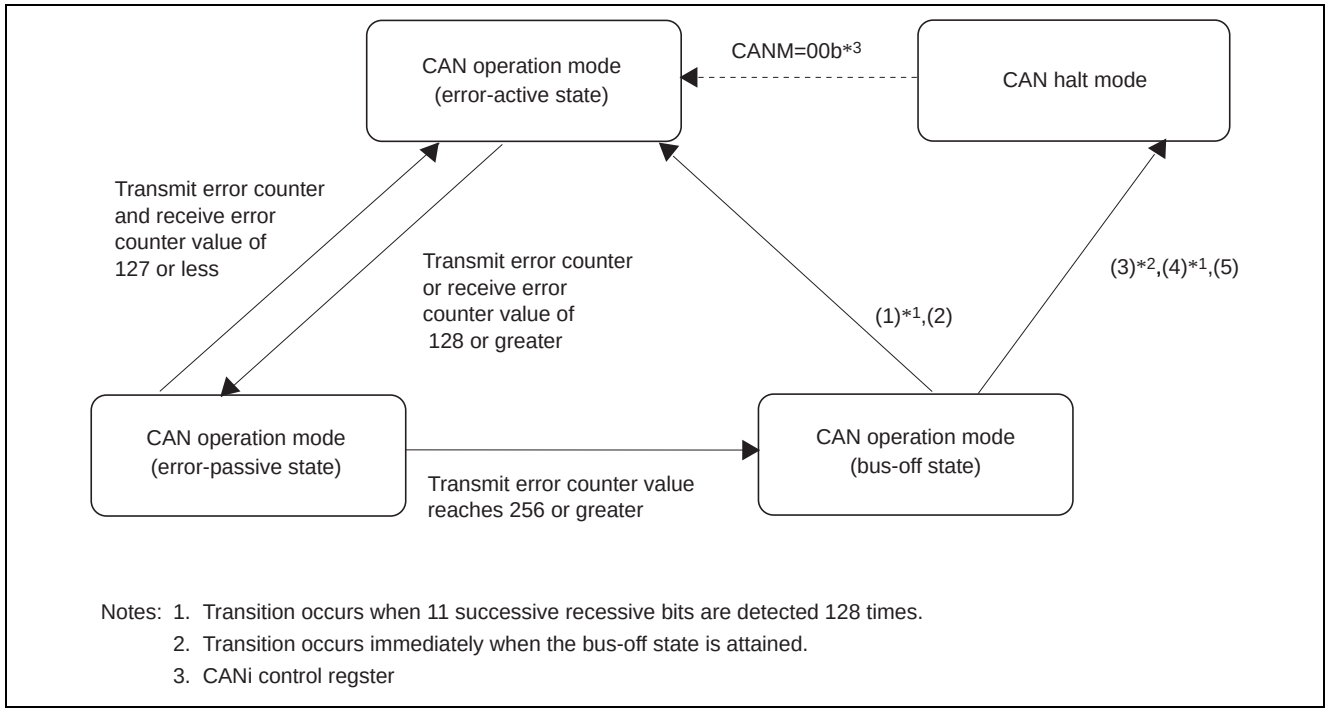


Figure 34 Transition to and from the Bus-Off State

9. Using an Acceptance Filter

An acceptance filter determines in hardware whether messages are received or discarded.

9.1 Standard ID and Extended ID

There are two CAN message ID formats: standard ID and extended ID. The former consists of 11 bits and the later consists of 29 bits. Figure 35 shows bit maps of the standard ID and extended ID.

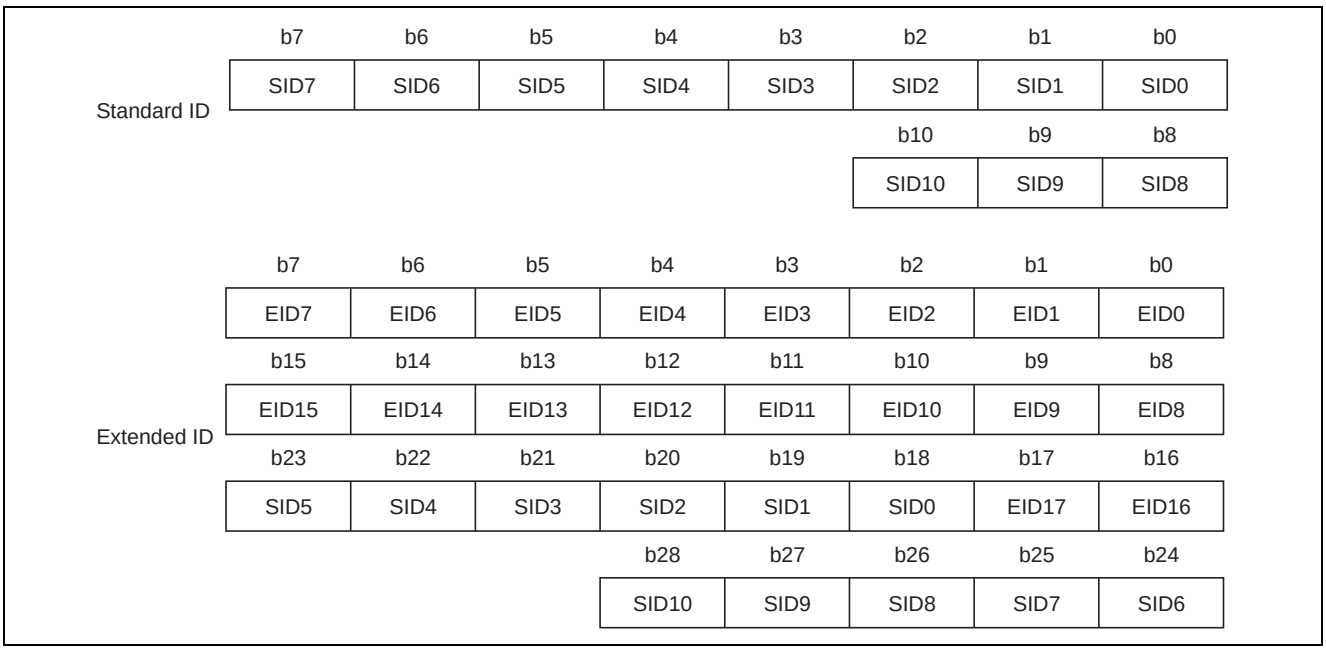


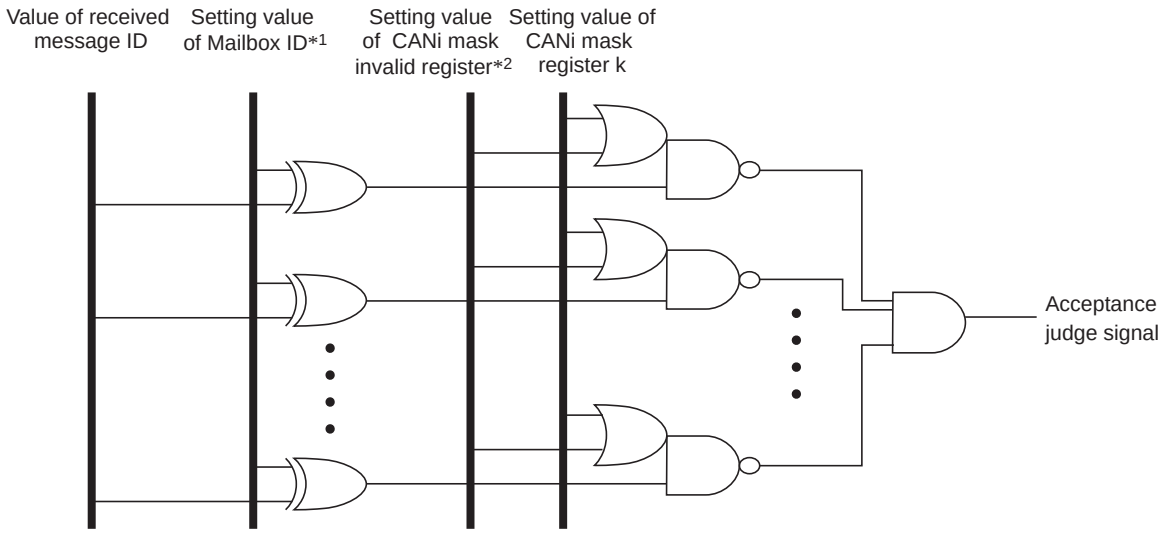
Figure 35 Bit Maps of Standard ID and Extended ID

9.2 Applying an Acceptance Filter to the Normal Receive Mailbox

An acceptance filter enables filtering by using CANi mask registers (CiMKRk). For details on using an acceptance filter in FIFO mailbox mode, see section 9.3.

(1) Acceptance Filter Register Configuration

Figure 36 shows the ID and mask register configuration, and figure 37 shows a bit map.

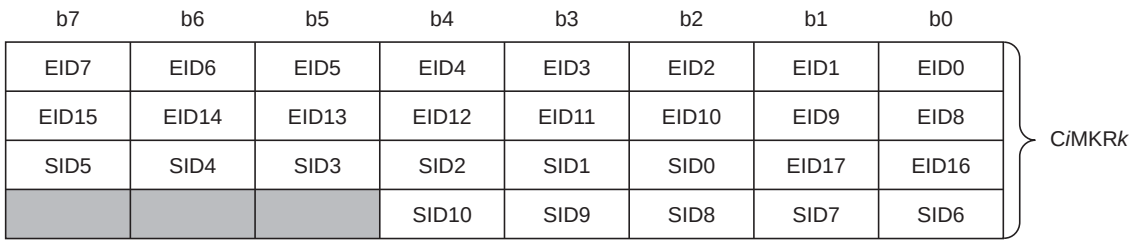


Notes: 1. The values set in the CANi FIFO Received ID Compare Register 0/1 are used in FIFO mailbox mode.
2. Invalid in FIFO mailboxes.

Figure 36 ID and Mask Register Configuration

Mailbox ID	Received Message ID Value Setting
CANi mask invalid register	0: Mask valid 1: Mask invalid
CANi mask register k	0: Corresponding ID bit is not compared 1: Corresponding ID bit is compared*
Acceptance judge signal	0: Discard message 1: Receive message

Note: * The compared IDs include the ID extension bit (IDE) and remote frame request bit (RTR). Note that the ID extension bit (IDE) is compared only when the ID format mode select bit in the CANi control register is set to mixed ID mode.



b7	b6	b5	b4	b3	b2	b1	b0	
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	C*i*MKR*k*
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	
SID5	SID4	SID3	SID2	SID1	SID0	EID17	EID16	
			SID10	SID9	SID8	SID7	SID6	

i: CAN channel
k: Mask register number

Figure 37 Bit Map

(2) Acceptance Filter Usage Examples

(a) **Usage Example 1**

Table 6 shows the register settings for receiving a standard data frame with ID 123h in mailbox [0].

Table 6 Acceptance Filter Usage Example 1

		IDE*[1], RTR, SID10 to SID6	SID5 to SID0, EID17 to EID16	EID15 to EID8	EID7 to EID0
Mailbox [0]		0000100	100011XX	XXXXXXXX	XXXXXXXX
Mask register	CiMKR0	--11111	111111XX	XXXXXXXX	XXXXXXXX
Received message	ID 123h	0000100	100011--	--------	--------

Note 1. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, the IDE bit should be written with 0.

(b) **Usage Example 2**

Table 7 shows the register settings for receiving a standard remote frame with ID 123h in mailbox [0].

Table 7 Acceptance Filter Usage Example 2

		IDE*[1], RTR, SID10 to SID6	SID5 to SID0, EID17 to EID16	EID15 to EID8	EID7 to EID0
Mailbox [0]		0100100	100011XX	XXXXXXXX	XXXXXXXX
Mask register	CiMKR0	--11111	111111XX	XXXXXXXX	XXXXXXXX
Received message	ID 123h	0100100	100011--	--------	--------

Note 1. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, the IDE bit should be written with 0.

(c) **Usage Example 3**

Table 8 shows the register settings for receiving two standard data frames with IDs 122h and 123h in mailbox [0].

Table 8 Acceptance Filter Usage Example 3

		IDE*[1], RTR, SID10 to SID6	SID5 to SID0, EID17 to EID16	EID15 to EID8	EID7 to EID0
Mailbox [0]		0000100	10001XXX	XXXXXXXX	XXXXXXXX
Mask register	CiMKR0	--11111	111110XX	XXXXXXXX	XXXXXXXX
Received message	ID 122h	0000100	100010--	--------	--------
	ID 123h	0000100	100011--	--------	--------

Note 1. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, the IDE bit should be written with 0.

(d) **Usage Example 4**

Table 9 shows the register settings for receiving an extended data frame with ID 12345678h in mailbox [0].

Table 9 Acceptance Filter Usage Example 4

		IDE*[1], RTR, SID10 to SID6	SID5 to SID0, EID17 to EID16	EID15 to EID8	EID7 to EID0
Mailbox [0]		1010010	00110100	01010110	01111000
Mask register	CiMKR0	--11111	11111111	11111111	11111111
Received message	ID 12345678h	1010010	00110100	01010110	01111000

Note 1. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, the IDE bit should be written with 0.

(e) **Usage Example 5**

Table 10 shows the register settings for receiving an extended remote frame with ID 12345678h in mailbox [0].

Table 10 Acceptance Filter Usage Example 5

		IDE*[1], RTR, SID10 to SID6	SID5 to SID0, EID17 to EID16	EID15 to EID8	EID7 to EID0
Mailbox[0]		1110010	00110100	01010110	01111000
Mask register	CiMKR0	--11111	11111111	11111111	11111111
Received message	ID 12345678h	1110010	00110100	01010110	01111000

Note 1. The IDE bit is enabled when the IDFM bit in the CiCTLR register is 10b (mixed ID mode). When the IDFM bit is not 10b, the IDE bit should be written with 0.

9.3 Applying an Acceptance Filter to the Receive FIFO

This acceptance filter mode is used with FIFO mailbox mode. Two filters can be applied to the receive FIFO. This extends the range of IDs that can be received by the receive FIFO. If there are 32 mailboxes for example, the two mask registers CiMKR6 and CiMKR7, and the two FIFO receive ID compare registers CiFIDCR0 and CiFIDCR1, would be used. In this acceptance filter mode the IDs of received messages are compared with the CANi FIFO receive ID compare registers (CiFIDCR0 and CiFIDCR1) instead of the mailbox ID.

Figure 38 shows the ID and mask register configuration.

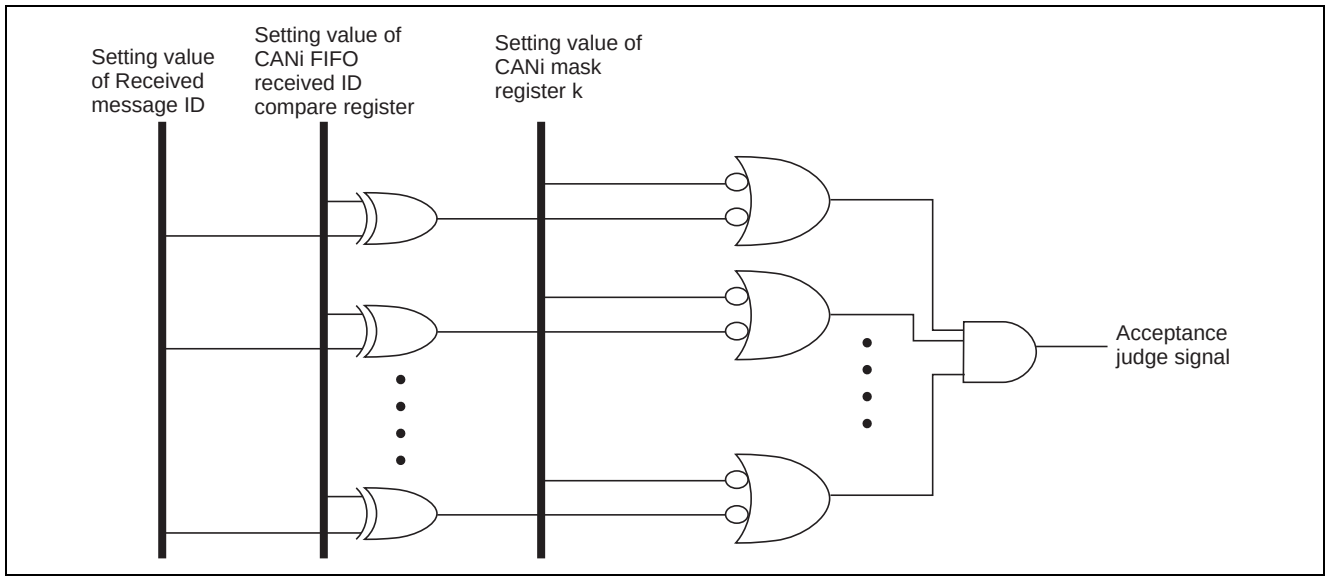


Figure 38 FIFO Receive ID Compare Register and Mask Register Configuration

CANi FIFO Received ID Compare Register	Received Message ID Value Setting
CANi mask register k	0: Corresponding ID bit is not compared 1: Corresponding ID bit is compared*
Acceptance judge signal	0: Discard message 1: Receive message

Note: * The compared IDs include the ID extension bit (IDE) and remote frame request bit (RTR). Note that the ID extension bit (IDE) is compared only when the ID format mode select bit in the CANi control register is set to mixed ID mode.

9.4 Acceptance Filter Support Unit

The acceptance filter support unit determines whether a receive ID is valid or invalid by means of a table lookup (8 bit * 256) . First, the IDs to be received are registered in a data table. Next, the ID of each received message is stored in the CANi acceptance filter support register (CiAFSR), the decoded receive ID is read from the CANi acceptance filter support register (CiAFSR), and a table lookup is performed. The acceptance filter support unit can only be used with standard frame IDs.

Using the acceptance filter support unit is effective in the following situations:

- When it is not possible to mask the IDs to be received by using an acceptance filter (e.g.: receive IDs 078h, 087h, and 111h)
- When the number of IDs to be received is extremely large and software filtering would take an excessive amount of time

9.4.1 Using the Acceptance Filter Support Unit

This example shows how to use the acceptance filter support unit to allow reception of IDs 000h, 00Dh, 6F3h, 6F4h, and 6FFh.

(1) Data Table Settings
Create a data table in ROM or RAM for registering the IDs to be received. The data table may be mapped to any range of addresses. In the data table, each vertical column contains the value of the upper eight bits (SID10 to 3) of a valid receive ID, and each horizontal row the value of the lower three bits (SID2 to 0) decoded into eight bits. Corresponding bits are set to 1 and the other bits are cleared to 0.

(2) Writing to the CANi Acceptance Filter Support Register (CiAFSR)
When CANi receives a message, the received ID is written to the CANi acceptance filter support register (CiAFSR).

(3) Reading from the CANi Acceptance Filter Support Register (CiAFSR)
The value of the upper eight bits (SID10 to 3) of the receive ID, and the value of the lower three bits (SID2 to 0) decoded into eight bits, are read from the CANi acceptance filter support register (CiAFSR).

(4) Determining Validity of Received IDs
Using the values read from the CANi acceptance filter support register (CiAFSR) in step (3), a lookup is performed of the data table created in step (1) to determine whether the message is valid or invalid.

Figure 39 shows the data table configuration, and figure 40 illustrates the states when writing to and reading from the CANi acceptance filter support register (CiAFSR).

(Example) ID 6F4h

Upper 8 bits								Lower 3 bits			
1	1	0	1	1	1	1	0	1	0	0	(6F4h)
D				E				3 to 8 conversion			
Data table vertical column								Horizontal row			

DEh

XXDEh (vertical column) bit 4 (horizontal row) set to 1

Lower 3 bits		
b2	b1	b0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

8 bits							
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Values 000h, 00Dh, 6F3h, and 6FFh are converted in like manner, and registered in the data table.

b7	b6	b5	b4	b3	b2	b1	b0	Address
007h 0	006h 0	005h 0	004h 0	003h 0	002h 0	001h 0	000h 1	Top
00Fh 0	00Eh 0	00Dh 1	00Ch 0	00Bh 0	00Ah 0	009h 0	008h 0	Top + 01h
6F7h 0	6F6h 0	6F5h 0	6F4h 1	6F3h 1	6F2h 0	6F1h 0	6F0h 0	Top + DEh
6FFh 1	6FEh 0	6FDh 0	6FCh 0	6FBh 0	6FAh 0	6F9h 0	6F8h 0	Top + DFh
7FFh 0	7FEh 0	7FDh 0	7FCh 0	7FBh 0	7FAh 0	7F9h 0	7F8h 0	Top + FFh

1: Valid 0: Invalid

Figure 39 Data Table Configuration

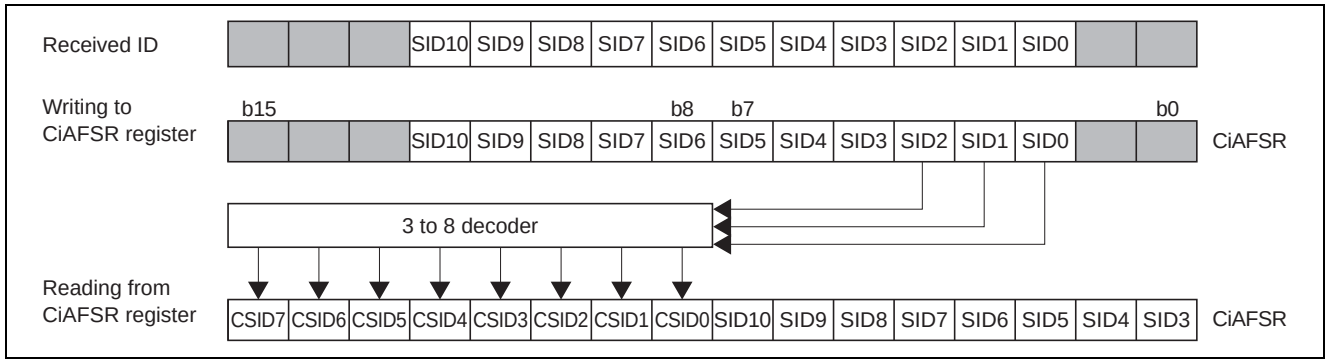
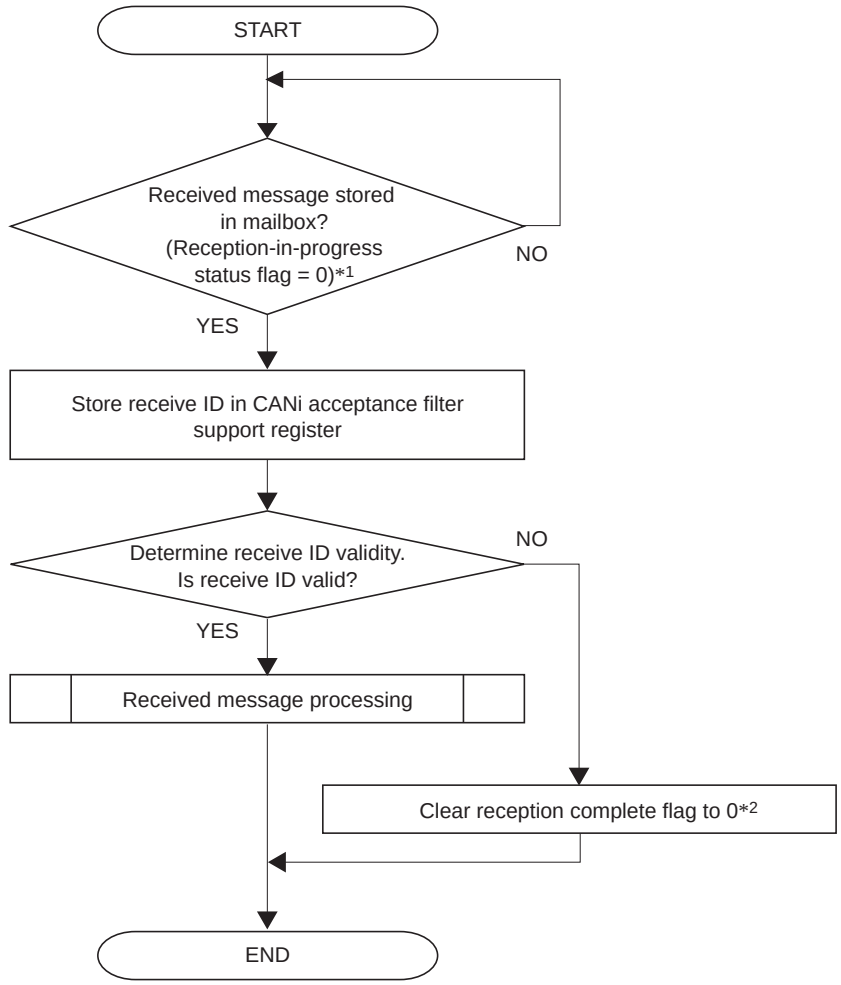


Figure 40 States when Writing to and Reading from CANi Acceptance Filter Support Register

Figure 41 shows the usage procedure for the acceptance filter support unit.



Notes: 1. This process is not necessary in overrun mode or one-shot receive mode.
2. Do not use the bit-clear instruction to clear this bit to 0. Use the MOV instruction to make sure the designated bit is cleared to 0. Writing 1 to this bit does not change its value.

Figure 41 Acceptance Filter Support Unit Usage Procedure

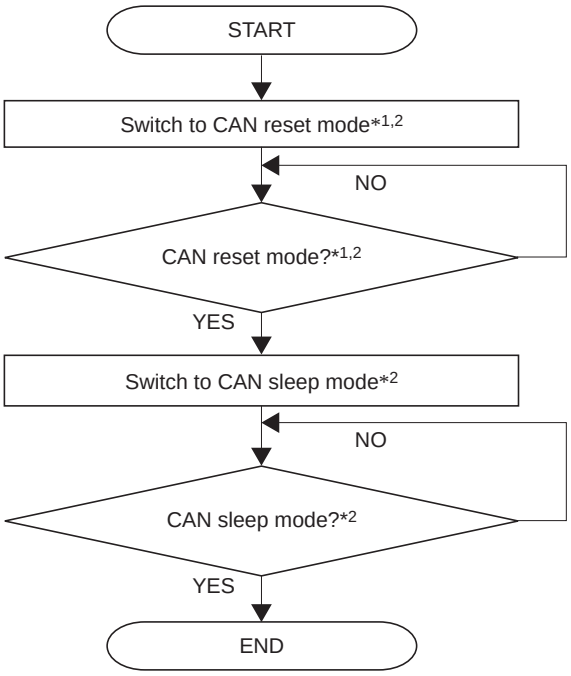
10. CAN Sleep Operation and CAN Wakeup Operation

10.1 CAN Sleep Operation

No clock is supplied to the CAN module when it is in CAN sleep mode, so the CAN module does not operate at all. Putting the CAN module into CAN sleep mode is recommended when the CAN module is not in use or when it is necessary to reduce the current consumption.

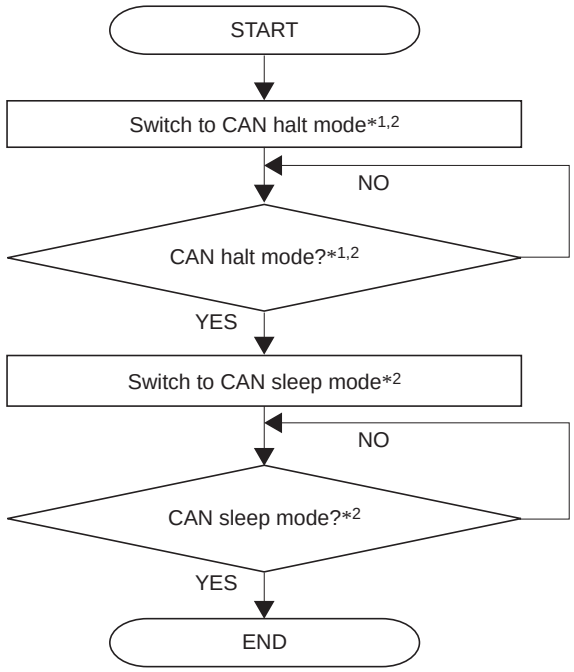
Always switch the CAN module to CAN reset mode or CAN halt mode before transitioning to CAN sleep mode. Figure 42 shows the procedure for switching the CAN module to CAN reset mode and then to CAN sleep mode, and figure 43 shows the procedure for switching the CAN module to CAN halt mode and then to CAN sleep mode.

When the CAN module is in CAN sleep mode, a further reduction in current consumption can be achieved by switching the MCU to wait mode or stop mode.



Notes: 1. If a transition to CAN reset mode is triggered while the CAN module is transmitting, the reset does not occur until the transmit operation completes. Therefore, always check the CAN reset status flag.
2. Always check the CAN sleep status flag when switching the CAN module to CAN sleep mode.

Figure 42 Procedure for Switching CAN Module to CAN Reset Mode and then CAN Sleep Mode



Notes: 1. If a transition to CAN halt mode is triggered while the CAN module is transmitting or receiving, the switch to CAN halt mode does not occur until the transmit or receive operation completes. Therefore, always check the CAN halt status flag.
2. Always check the CAN sleep status flag when switching the CAN module to CAN sleep mode.

Figure 43 Procedure for Switching CAN Module to CAN Halt Mode and then CAN Sleep Mode

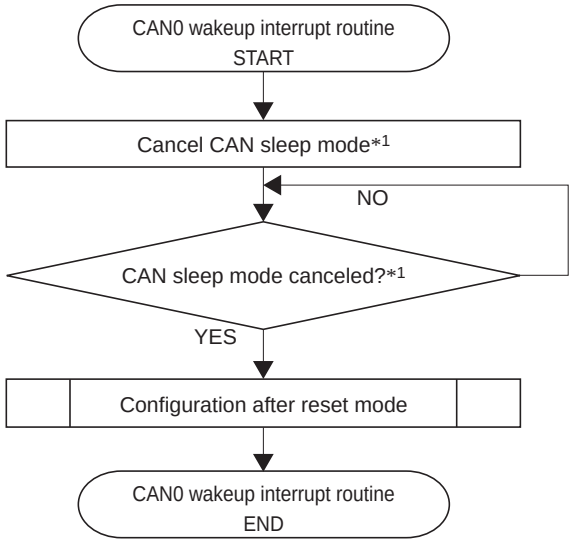
10.2 CAN Wakeup Operation

The CAN module can be restored from CAN sleep mode by means of a CANi wakeup interrupt issued when the CAN receive line signal level falls. To use CAN wakeup, enable the CANi wakeup interrupt in the CANi wakeup interrupt control register (CiWIC).

When CAN sleep mode is canceled by software, the CAN module reenters the mode (CAN reset mode or CAN halt mode) it was in before the transition to CAN sleep mode.

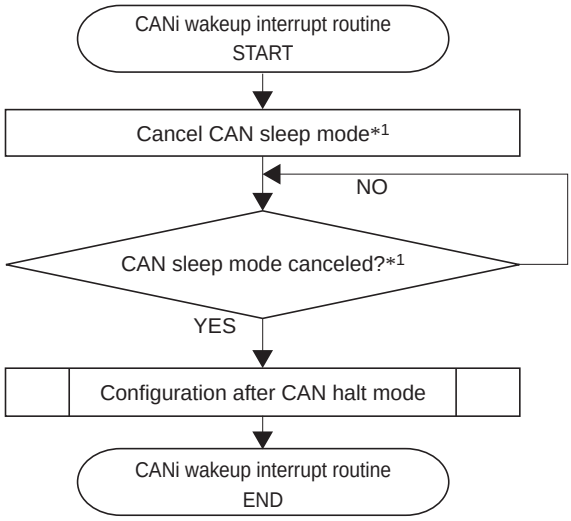
Figure 44 shows the CAN wakeup procedure when the CAN module was switched to CAN sleep mode from CAN reset mode, and figure 45 shows the CAN wakeup procedure when the CAN module was switched to CAN sleep mode from CAN halt mode.

In these examples, the CAN module is returned to CAN operation mode within the CAN wakeup interrupt routine.



Note:1. Always check the CAN sleep status flag when canceling CAN sleep mode.

Figure 44 CAN Wakeup Procedure in CAN Reset Mode



Note:1. Always check the CAN sleep status flag when canceling CAN sleep mode.

Figure 45 CANi Wakeup Procedure in CAN Halt Mode

11. Test Modes

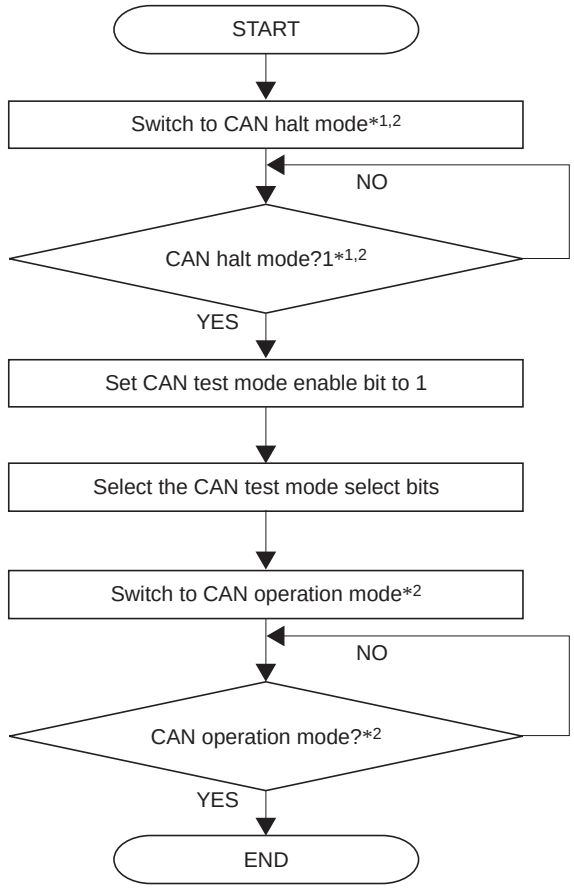
The following three test modes are provided for evaluation by the user:

- Listen-only mode
- Self-test mode 0 (external loop-back)
- Self-test mode 1 (internal loop-back)

Only select test modes when the CAN module is in CAN halt mode.

11.1 Test Mode Selection

Figure 46 shows the procedure for selecting a test mode.



Notes: 1. If a transition to CAN halt mode is triggered while the CAN module is transmitting or receiving, the switch to CAN halt mode does not occur until the transmit or receive operation completes. Therefore, always check the CAN halt status flag.
2. Always check the CAN halt status flag when switching the CAN module to CAN operation mode.

Figure 46 Test Mode Selection Procedure

11.2 Listen-Only Mode

The CAN specification (ISO 11898-1) recommends the implementation of an optional bus monitor mode. In listen-only mode, the CAN node can receive valid data frames and valid remote frames, but only recessive bits are sent via the CAN bus and transmit-start is not enabled. When the CAN module receives a request to transmit a dominant bit (ACK bit, overload flag, or active error flag), the bit is rerouted internally so that the CAN module can treat it as dominant even though the CAN bus remains in the recessive state.

Listen-only mode can be used for baud rate detection.

Do not initiate transmit requests from any mailbox when in listen-only mode.

To select listen-only mode, set the CAN test mode select bits (TSTM) to 01b.

Figure 47 illustrates the operation of listen-only mode.

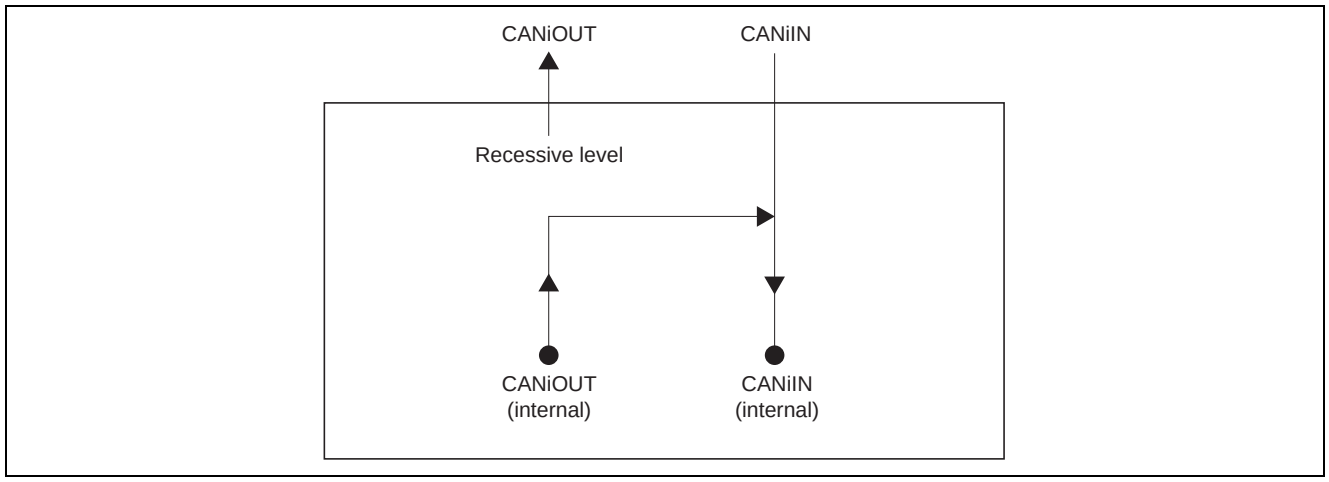


Figure 47 Listen-Only Mode Operation Illustration

11.3 Self-Test Mode 0 (External Loop-Back)

Self-test mode 0 is for CAN transceiver testing.

In self-test mode 0, the CAN module handles messages it has transmitted itself as messages received via a CAN transceiver. Each transmitted message is stored in the receive buffer. Since this function is performed independently of the external device, the CAN module generates its own ACK bits.

To use self-test mode 0, connect the CANiOUT and CANiIN pins to a CAN transceiver.

To select self-test mode 0, set the CAN test mode select bits (TSTM) to 10b.

Figure 48 illustrates the operation of self-test mode 0.

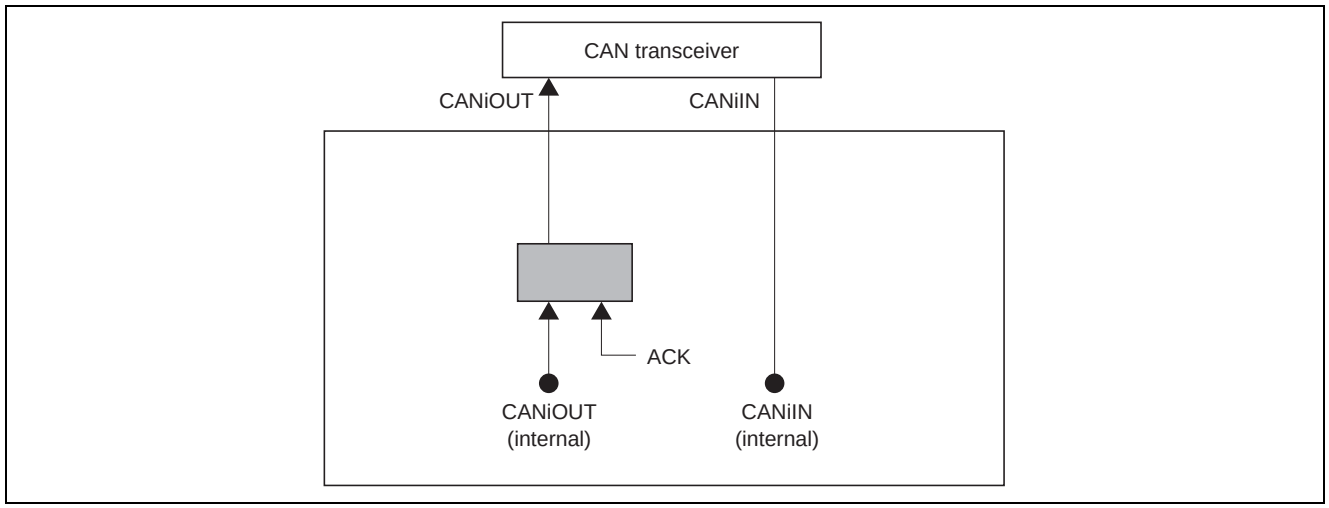


Figure 48 Self-Test Mode 0 Operation Illustration

11.4 Self-Test Mode 1 (Internal Loop-Back)

Self-test mode 1 is for self-testing.

In self-test mode 1, the CAN module handles messages it has transmitted itself as received messages. Each transmitted message is stored in the receive buffer. Since this function is performed independently of any external device, the CAN module generates its own ACK bits.

In self-test mode 1, the CAN module performs internal feedback from the internal CANiOUT pin to the internal CANiIN pin. The CAN module ignores the actual value of input to the external CANiIN pin. The external CANiOUT pin outputs only recessive bits. It is not necessary to connect the CANiOUT or CANiIN pin to the CAN bus or to any device to use self-test mode 1.

To select self-test mode 1, set the CAN test mode select bits (TSTM) to 11b. Figure 49 illustrates the operation of self-test mode 1.

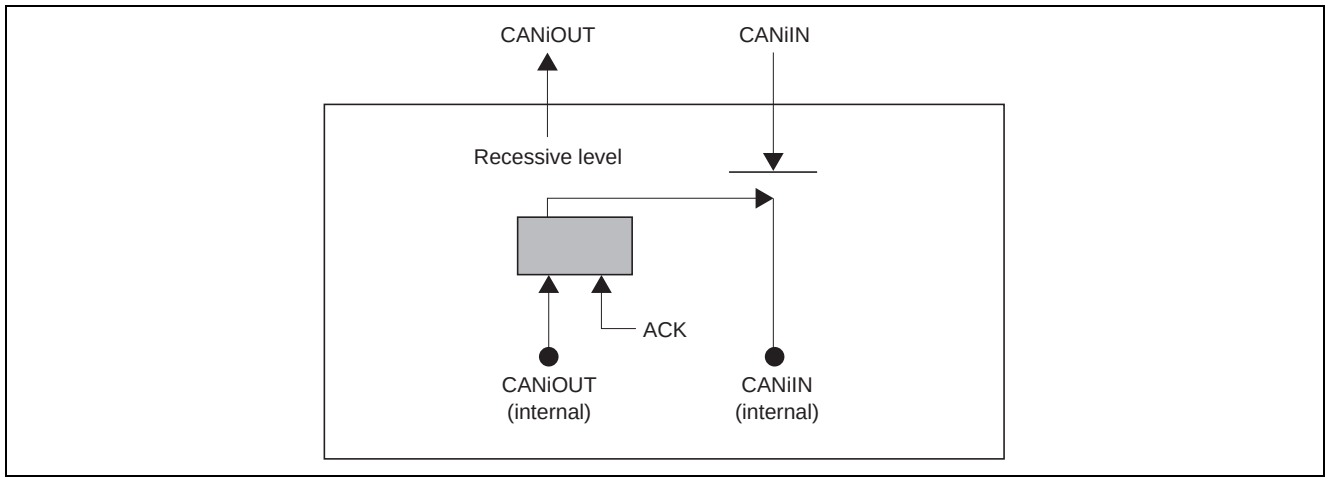


Figure 49 Self-Test Mode 1 Operation Illustration

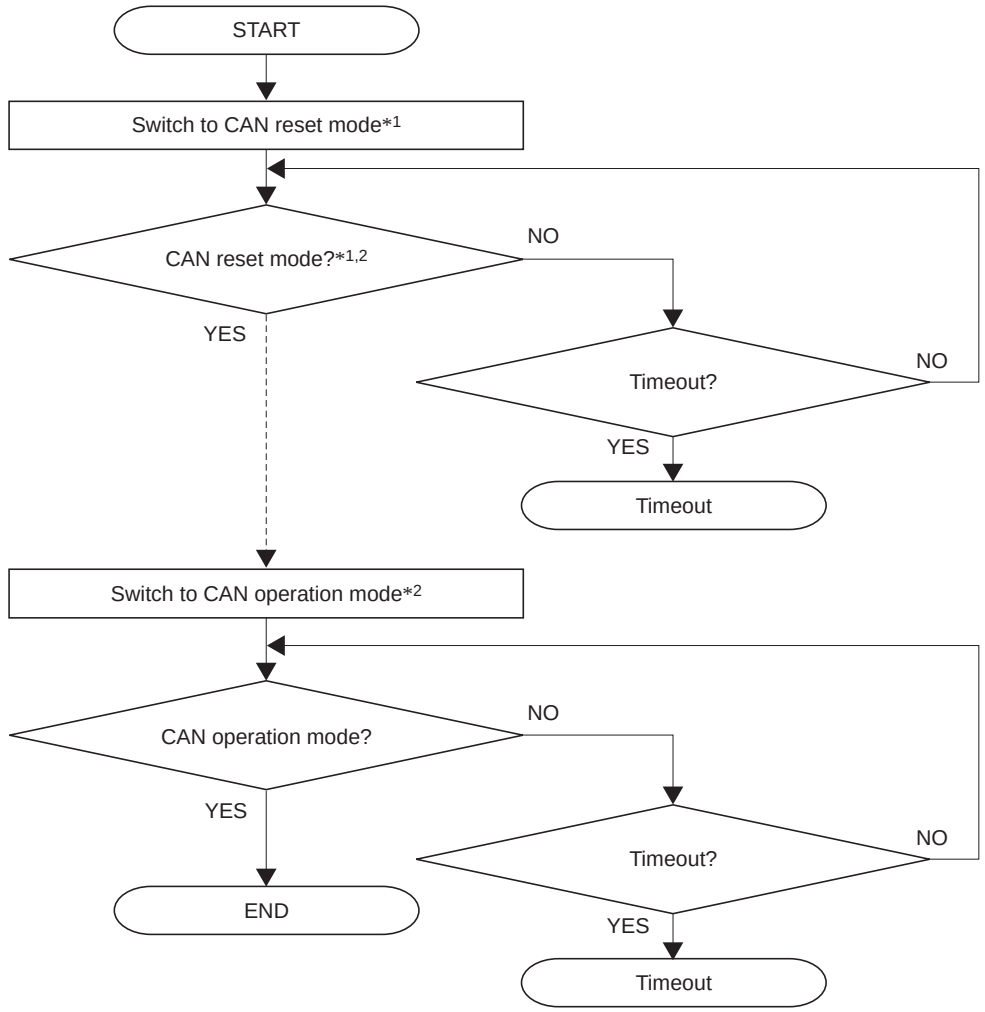
12. Notes on the Processing Flow

12.1 Infinite Loops

Since the notations in this document are abbreviated, they may contain instances where an infinite loop can occur in the processing flow. When creating the actual program code, make sure to insert loop time limits to prevent overruns from occurring.

Figure 50 shows a processing example using a loop time limit.

(Configuration after CAN reset mode)



Note: 1. When the CAN module is reset during transmission of a CAN message, it waits for the transmission operation to complete and then transitions to CAN reset mode. The time limit for the transition to CAN reset mode is the maximum time required to transmit one frame of approximately 160 bits (maximum number of bits in extended data frame (approximately 130 bits) + maximum number of stuff bits (approximately 30 bits)).

(Example)
Time required to transmit one frame (160 bits) at 500 Kbps: 320 μsec.
The time limit is set to 320 μsec., so processing breaks the loop after more than 320 μsec. elapse.

Figure 50 Processing Example with Loop Time Limit

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	December 20, 2009	—	First edition issued
1.01	February 15, 2010	1	(j = 0 to 32)-> (j = 0 to 32)
		7	Add formula for calculation transfer speed
		15	mask disable register -> mask invalid register
		27	Change Fig15
		28	Change Note3 of fig 16
		29	Change Fig17
		34	CiMCTLj ->CiCTLR
		43	Change fig 27.
	March 31, 2013	—	Document number revised : REJ05B1354 ⇒ R01AN1651EJ

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2