RH850/U2A EVA Group

CAN Configuration (CAN FD Mode)

Summary

This application note explains the procedure example for CAN configuration in the RH850/U2A series of automotive single-chip microcontrollers from Renesas Electronics (hereafter referred to as U2A).

These documents and programs are intended to understand the RH850/U2A built-in function, and are not intended for mass production design.

Aim of this document and software is to provide supplemental information for the function on RH850/U2A. It is not intended to implement in the design for mass production.

There is no guarantee to update in this document and software to reflect the latest manual, errata, technical update and development environment. You are fully responsible for the incorporation or any other use of the information of this document in the design of your product or system, and please refer to latest manual, errata, technical update and development environment.

Target Device

- RH850/U2A-EVA Group

Target Integrated Development Environment

CS+ (from Renesas Electronics)

Version : V8.07.00

Device File : DR7F702300.DVF

: DR7F702301.DVF

: DR7F702302.DVF

Reference Document

RH850/U2A-EVA User's Manual: Hardware

For function details and electrical characteristics, please refer to "User's Manual: Hardware".

This application note is based on the following manual.

- RH850/U2A-EVA User's Manual (Rev.1.20): R01UH0864EJ0120

The register name in this text omits "RSCFDnCFD".

Contents

1. CAN Configuration 3
2. CAN State (Mode) Transition 10
3. Communication Speed 14
4. Global Function 17
5. Receive Rule Table 23
6. Buffer, FIFO Buffer 32
7. Global Error Interrupt 46
8. Channel Function 48
9. CAN-related Interrupt 56
10. DMA Trigger 59
11. Transmitter Delay Compensation (Only CAN FD Mode) 61
12. Precautions for Processing Flow 62
13. Appendix 64

1. CAN Configuration

CAN configuration means to setup the required function for performing CAN communication. Perform the configuration to start or restart CAN communication after MCU reset, recovery from bus failure detection, wake-up, etc.

CAN configuration can be performed in certain CAN status. Refer to the following for the details.

For CAN State (Mode), refer to ”2. CAN State (Mode) Transition”.

1.1 CAN Configuration after MCU Reset

1.2 CAN Configuration after Global Reset Mode

1.3 CAN Configuration after Channel Reset Mode

1.4 CAN Configuration after Channel Halt Mode

The required functions when setting CAN configuration are shown below. For details of each process, refer to the following chapters.

2. CAN State (Mode) Transition

3. Communication Speed

4. Global Function

5. Receive Rule Table

6. Buffer, FIFO Buffer

7. Global Error Interrupt

8. Channel Function

9. CAN-related Interrupt

10. DMA Trigger

11. Transmitter Delay Compensation (Only CAN FD Mode)

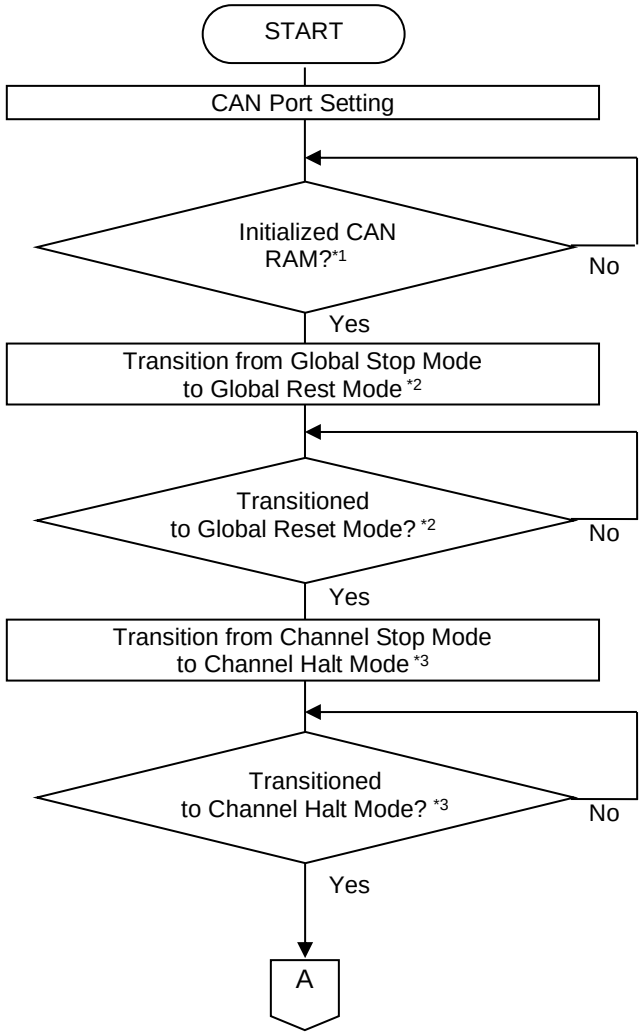
1.1 CAN Configuration after MCU Reset

1.1.1 CAN Configuration after MCU Reset

This section shows the intialization processing procedure of entire RS-CANFD module after MCU reset.

1.1.2 Setting Procedure of CAN Configuration after MCU Reset

Figure 1-1 and 1-2 shows the configuration procedure after MCU reset.

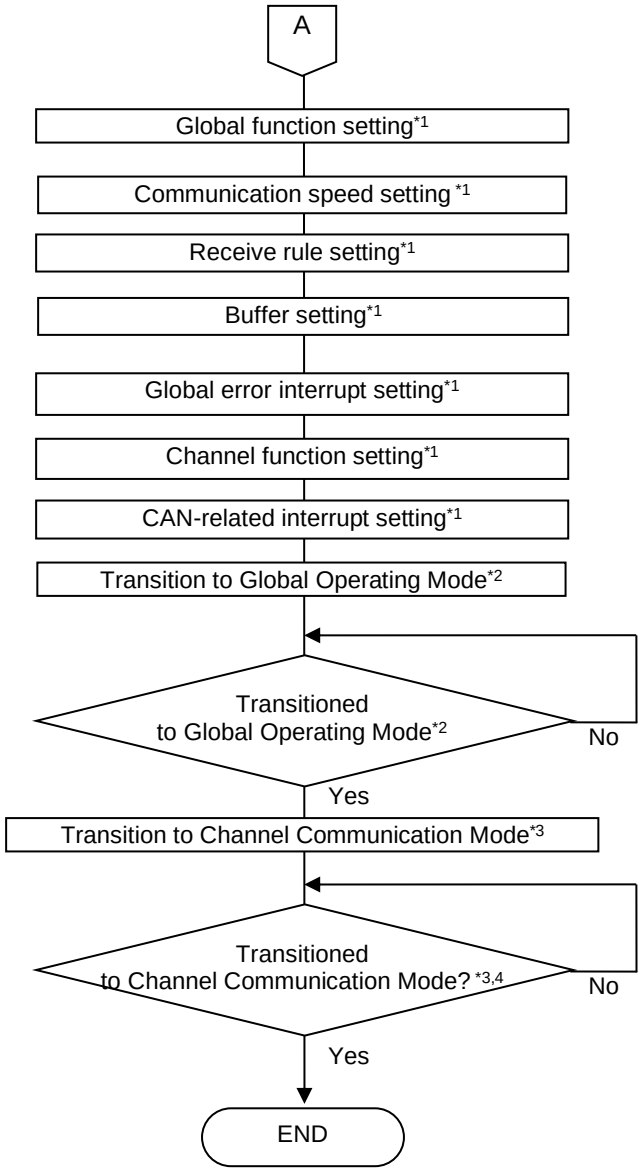


Note 1. After resetting the MCU, do not access to CAN RAM before initializing the CAN RAM (during the GRAMINIT bit of the GSTS register is set to "1").

Note 2. If you change the global mode (the GSLPR bit and the GMDC [1:0] bit of the GCTR register), check that the mode is switched in the GSTS register. Do not change the GMDC[1:0] bit until the mode is switched.

Note 3. If you change the channel mode (the CSLPR bit and the CHMDC [1:0] bit of the CmCTR register), check that the mode is switched in the CmSTS register. Do not change the CHMDC[1:0] bit until the mode is switched.

Figure 1-1 Configuration Procedure after MCU Reset 1/2



Note 1. For processes of each function, refer to the following chapters.

Note 2. If you change the global mode (the GSLPR bit and the GMDC [1:0] bit of the GCTR register), check that the mode is switched in the GSTS register. Do not change the GMDC[1:0] bit until the mode is switched.

Note 3. If you change the channel mode (the CSLPR bit and the CHMDC [1:0] bit of the CmCTR register), check that the mode is switched in the CmSTS register. Do not change the CHMDC[1:0] bit until the mode is switched.

Note 4. After the transition to channel communication mode, when 11 consecutive recessive bits have been detected, communication is ready (the COMSTS flag in the CmSTS register is “1”) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

Figure 1-2 Configuration Procedure after MCU Reset 2/2

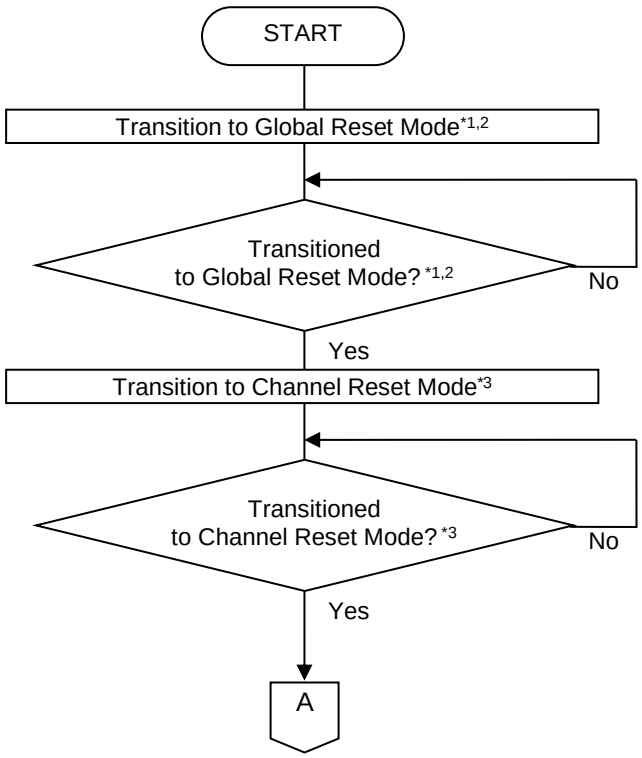
1.2 CAN Configuration after Global Reset Mode

1.2.1 CAN Configuration after Global Reset Mode

This section shows the intialization processing procedure of entire RS-CANFD module after transitioned to global reset mode.

1.2.2 Setting Procedures of CAN Configuration after Global Reset Mode

Figure 1-3 and Figure 1-4 shows the procedure of CAN configuration after global reset mode.

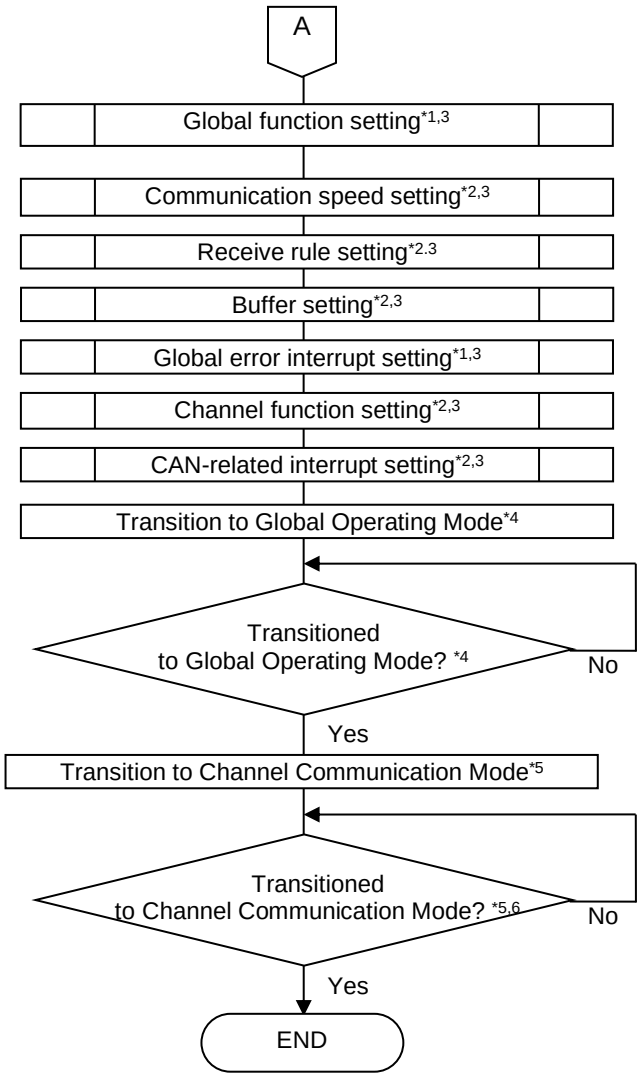


Note 1. When changing the global mode (the GSLPR bit and the GMDC [1:0] bits of the GCTR register), confirm with the GSTS register that the mode has been switched. Do not change the GMDC [1:0] bits until the mode is switched.

Note 2. Channel modes may transition in accordance with transitions of global modes. For details, refer to “2.3 Change of Channel Mode by Transition of Global Mode”.

Note 3. When changing the channel mode (the CSLPR bit and the CHMDC [1:0] bits of the CmCTR register), confirm with the CmSTS register that the mode has been switched. Do not change the CHMDC[1:0] bits until the mode is switched.

Figure 1-3 Procedure of CAN Configuration after Global Reset Mode 1/2



Note 1. These settings do not always have to be executed because the values are not reset in the transition to Global Reset Mode.

Note 2. These settings do not always have to be executed because the values of these settings are not reset in the transition to Channel Reset Mode.

Note 3. For processes of each function, refer to the following chapters.

Note 4. When changing the global mode (the GSLPR bit and the GMDC [1:0] bits of the GCTR register), confirm with the GSTS register that the mode has been switched. Do not change the GMDC [1:0] bits until the mode is switched.

Note 5. When changing the channel mode (the CSLPR bit and the CHMDC [1:0] bits of the CmCTR register), confirm with the CmSTS register that the mode has been switched. Do not change the CHMDC [1:0] bits until the mode is switched.

Note 6. After the transition to channel communication mode, when 11 consecutive recessive bits have been detected, communication is ready (the COMSTS flag in the CmSTS register is “1”) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

Figure 1-4 Procedure of CAN Configuration after Global Reset Mode 2/2

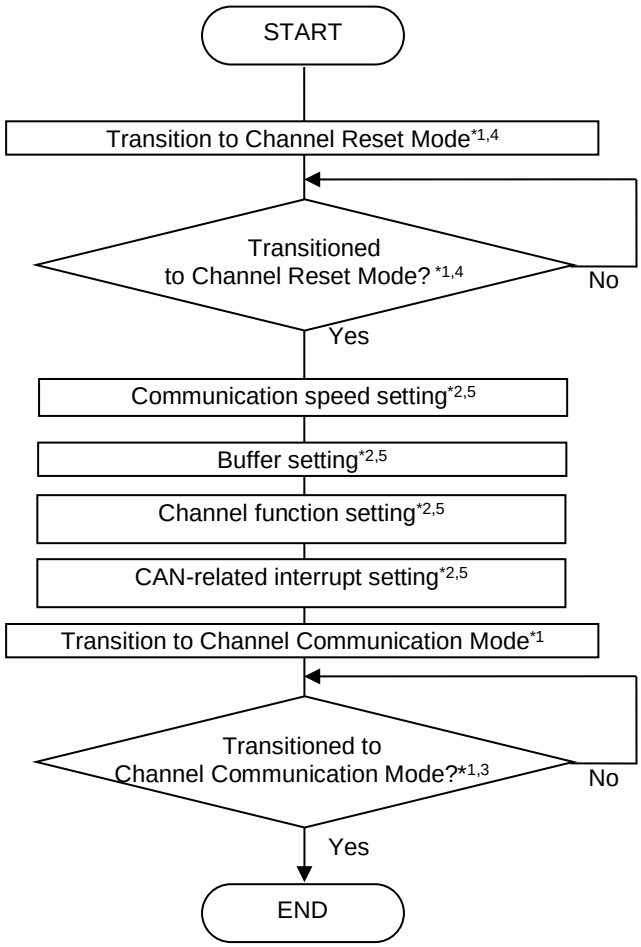
1.3 CAN Configuration after Channel Reset Mode

1.3.1 CAN Configuration after Channel Reset Mode

This section shows the CAN channel intialization processing procedure after transitioned to channel reset mode.

1.3.2 Setting Procedure of CAN Configuration after Channel Reset Mode

Figure 1-5 shows the procedure of CAN configuration after channel reset mode.



Note 1. When changing the channel mode (the CSLPR bit and the CHMDC[1:0] bit of the CmCTR register), confirm with the CmSTS register that the mode has been switched. Do not change the CHMDC[1:0] bits until the mode is switched.

Note 2. These settings do not always have to be executed because the values are not reset in the transition of channel reset mode.

Note 3. After the transition to channel communication mode, when 11 consecutive recessive bits have been detected, communication is ready (the COMSTS flag in the CmSTS register is “1”) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

Note 4. Transition to Channel Reset Mode does not wait for communication to complete. To allow transition to channel reset mode after communication is completed, confirm that communication has been completed and transitioned to Channel Halt Mode, and then transition to channel reset mode.

Note 5. For processes of each function, refer to the following chapters.

Figure 1-5 Procedure of CAN Configuration after Channel Reset Mode

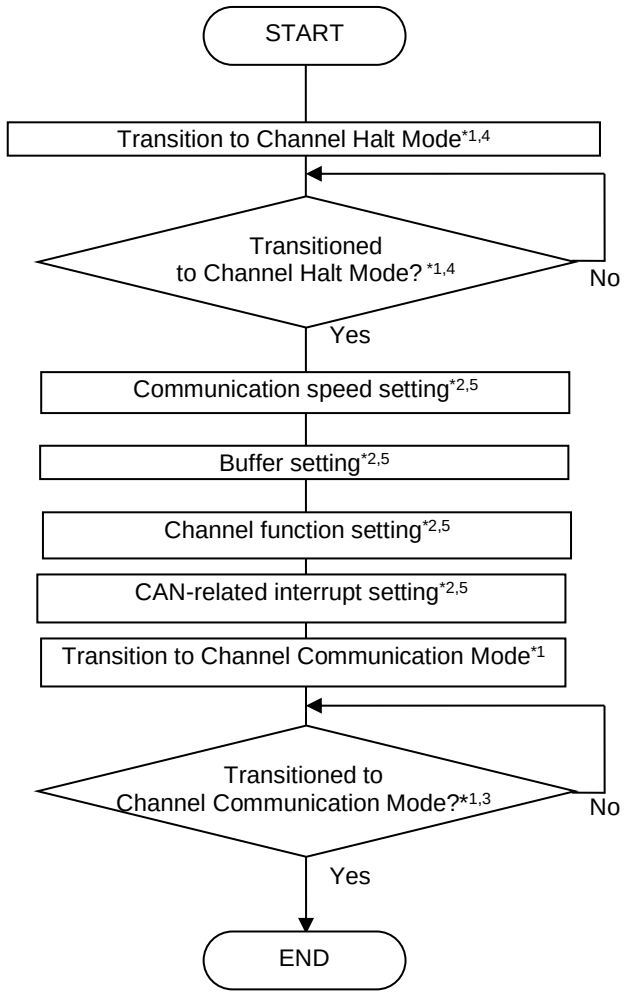
1.4 CAN Configuration after Channel Halt Mode

1.4.1 CAN Configuration after Channel Halt Mode

This section shows the CAN channel intialization processing procedure after transitioned to Chanel Halt Mode.

1.4.2 Setting Procedures of CAN Configuration after Channel Halt Mode

Figure 1-6 shows the procedure of CAN configuration after channel halt mode.



Note 1. When changing the channel mode (the CSLPR bit and the CHMDC[1:0] bits in the CmCTR register), confirm with the CmSTS register that the mode has been switched. Do not change the CHMDC[1:0] bits until the mode is switched.

Note 2. These settings do not always have to be executed because the values are not reset in the transition to Channel Halt Mode.

Note 3. After the transition to channel communication mode, when 11 consecutive recessive bits have been detected, communication is ready (the COMSTS flag in the CmSTS register is “1”) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

Note 4. While the CAN bus is locked to the dominant level (BLF flag in the CmERFLL register is “1”), transition to channel halt mode is not made. In that case, enter channel reset mode.

Note 5. For processes of each function, refer to the following chapters.

Figure 1-6 Procedure of CAN Configuration after Channel Halt Mode

2. CAN State (Mode) Transition

RS-CANFD module has the state of entire channels (hereinafter called Global) and each channel (mode).

The states (modes) of the RS-CANFD module are shown below.

- 2.1 Global Mode
  - 2.1.1 Global Stop Mode
  - 2.1.2 Global Reset Mode
  - 2.1.3 Global Test Mode
  - 2.1.4 Global Operating Mode

- 2.2 Channel Mode
  - 2.2.1 Channel Stop Mode
  - 2.2.2 Channel Reset Mode
  - 2.2.3 Channel Halt Mode
  - 2.2.4 Channel Communication Mode

2.1 Global Mode

This is the mode of entire RS-CANFD mode. Channel modes may transition in accordance with transitions of global modes. Refer to “2.3 Change of Channel Mode by Transition of Global Mode” for the details

2.1.1 Global Stop Mode

This mode stops RS-CANFD module clocks. The CAN clocks do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

2.1.2 Global Reset Mode

This mode is used to configure the entire RS-CANFD module. When the RS-CANFD module transitions to global reset mode, some registers are initialized. Refer to the latest user's manual: hardware for the registers to be initialized.

2.1.3 Global Test Mode

This mode is used to set test-related registers. When the RS-CANFD module transitions to global test mode, all CAN communications are disabled.

2.1.4 Global Operating Mode

This mode operates entire RS-CANFD module. When do a communication using each channel, transition to global operating mode is required.

2.2 Channel Mode

Refer to ”User’s Manual Hardware : Figure 23.11 Channel Mode State Transition Chart” for the transition chart of channel mode.

2.2.1 Channel Stop Mode

This mode stops the clock supply to a channel. CAN clock do not runt and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

2.2.2 Channel Reset Mode

This mode is used to configure channel settings. When a transitions to channel reset mode, some channel-related registers are initialized. Refer to the latest user’s manual: hardware for initialized register.

2.2.3 Channel Halt Mode

This mode is used to set test-related registers of channels. When a channel transitions to Channel Halt Mode, CAN communication of the channel stops.

2.2.4 Channel Communication Mode

This mode performs CAN communication. Each channel is in the following communication states during CAN communication.

- Idle
  Neither reception nor transmission is in progress.
- Reception
  Receiving a message sent from another node.
- Transmission
  Transmitting a message.
- Bus off
  Isolated from CAN communication.

2.3 Change of Channel Mode by Transition of Global Mode

Channel modes may transition in accordance with transitions of global modes. Figure 2-1 shows the transitions of channel modes depending on the global mode setting. Refer to “User’s Manual Hardware: Table 23.135 Possible CAN channel Modes versus Global Module Modes” for the details.

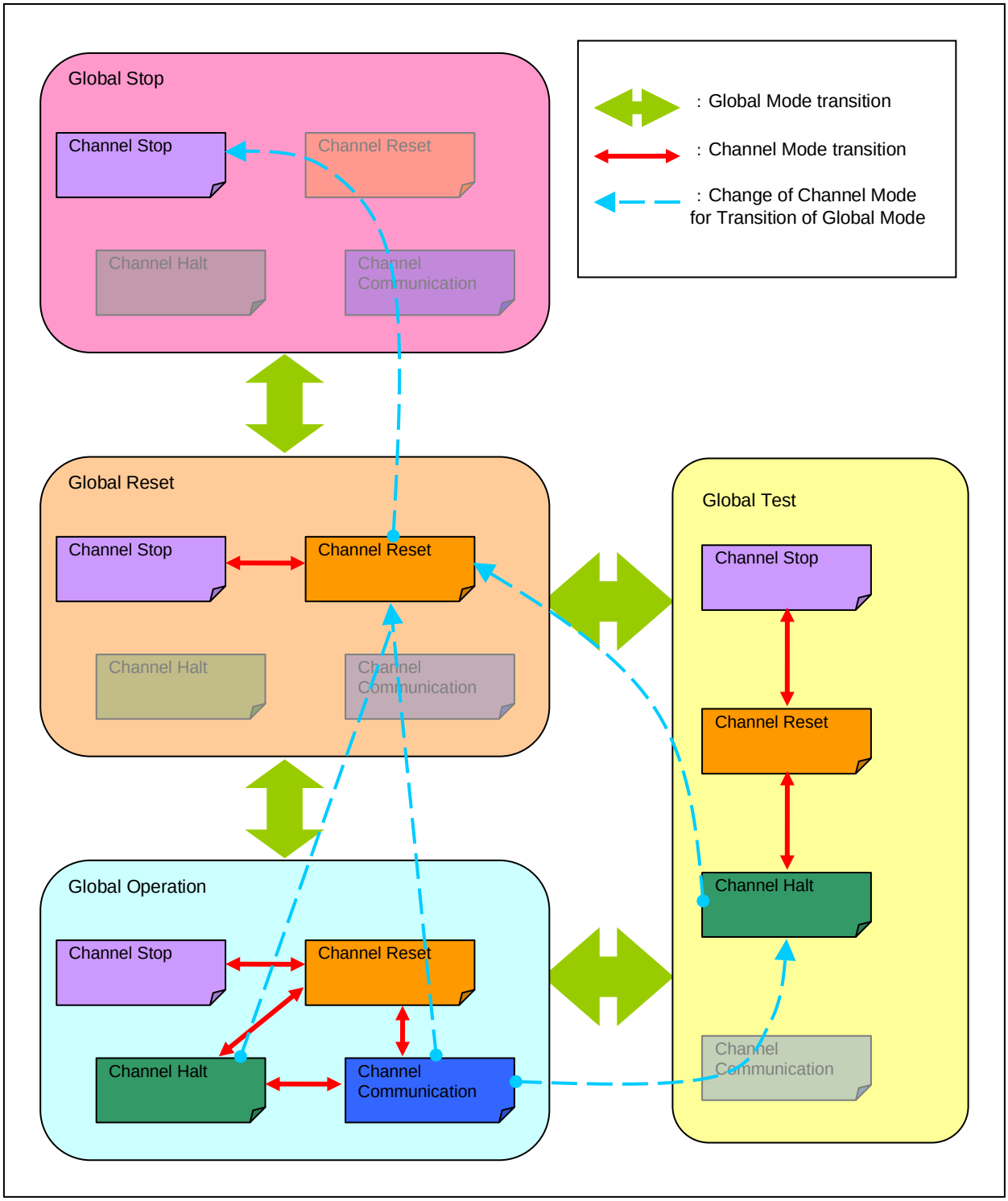


Figure 2-1 Transition of Global Mode and Channel Mode

3. Communication Speed

Set the communication speed for CAN communication. You need to perform the following settings to determine CAN communication speed.

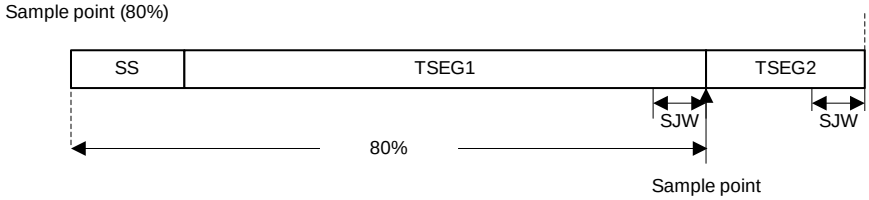
- 3.1 CAN Bit Timing Setting
- 3.2 Setting of Communication Speed
- 3.3 Setting Procedure of CAN Bit Timing and Communication Speed

3.1 CAN Bit Timing Setting

In this CAN bit timing setting of RS-CANFD module, one bit of a communication frame consists of three segments. Figure 3-1 shows segments of bits and sample point.

In these segments, Time Segment 1(hereinafter called TSEG1) and Time Segment 2 (hereinafter called TSEG2) indicate the sample point. Also, it can be changed the timing for sampling by changing the values of segments. CAN FD mode has 2 types of bit rate (nominal bit rate and data bit rate), and each should be set.

The minimum unit for this timing setting is called 1 Time Quantum (hereinafter called Tq) and is determined by the clock frequency supplied to the RS-CANFD module and the baud rate prescaler division value.



Nominal bit rate
SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 128 Tq
Set TSEG2 to a range of 2 Tq to 32 Tq
Set SJW to a range of 1 Tq to 32 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 161 Tq
TSEG1 > TSEG2 ≥ SJW

DATA bit rate
SS = 1 Tq fixed
Set TSEG1 to a range of 2 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 8 Tq
Set SS + TSEG1 + TSEG2 to a range of 5 Tq to 25 Tq
TSEG1 ≥ TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (Time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice of the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (Time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (Resynchronization jump width)
  The SJW is a length to extend or reduce the time segment to compensate for an error in phase due to phase error.

Figure 3-1 Bit Segment Components and Sample Point

3.2 Setting of Communication Speed

Communication speed is determined by CAN clock (fCAN) that is clock source of RS-CANFD module, baud rate prescaler division value, and Tq count per bit. The maximum communication speed that can be set is 1 Mbps at the normal bit rate and 8 Mbps at the data bit rate. Either clkc or clk_xincan can be used as fCAN. For fCAN setting, refer to “4.5 CAN Clock Source Setting”.

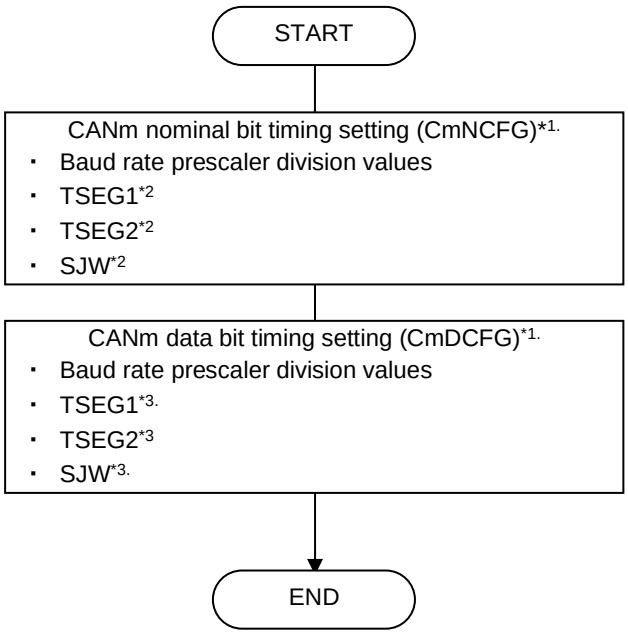
Refer to “User’s Manual Hardware : Table 23.143 Nominal Baud Rate calculation formula and example CAN communication configurations” and “User’s Manual Hardware : Table 23.144 Baud Rate calculation example for nominal and data bit rate CAN communication configurations” for the calculation formula and implementation example. Refer to “User’s Manual Hardware: Table 23.142 Bit timing examples” for the bit timing setting example.

3.3 Setting Procedure of CAN Bit Timing and Communication Speed

Figure 3-2 shows the setting procedure of CAN bit timing and communication speed.

Perform these setting during CAN configuration.

Refer to “1. CAN Configuration” for the procedure of CAN configuration.



Note 1. Rewrite the CmNCFG register and CmDCFG register in channel reset mode or channel halt mode. Also, set the registers before transitioning channel communication mode or channel halt mode. If you use classical CAN frame only, set the same values to CmDCFG register with CmNCFG register.

Note 2. TSEG1, TSEG2, and SJW settings need to meet following conditions.
TSEG1 > TSEG2 >= SJW

Note 3. TSEG1, TSEG2, and SJW settings need to meet following conditions.
TSEG1 >= TSEG2 >= SJW

Figure 3-2 Setting Procedure of CAN Bit Timing and Communication Speed

4. Global Function

Set the following functions that are common to entire RS-CANFD module (all channels).

4.1 Transmit Priority Setting
4.2 DLC Check Setting
4.3 DLC Replacement Function Setting
4.4 Mirror Function Setting
4.5 CAN Clock Source Setting
4.6 Payload Overflow Mode Setting
4.7 Timestamp Clock Setting
4.8 Interval Timer Prescaler Setting

4.1 Transmit Priority Setting

Set the transmit priority when transmit requests are issued from multiple transmit buffers or transmit queues in the same channel.

It is not able to set the transmit priority for each channel because the transmit priority is common to entire channels. You can choose following two of judgment methods.[*1]

- ID Priority
  The messages are transmitted according to the priority of stored message IDs. Priority of IDs conforms to the CAN bus arbitration specification defined in the CAN specification.
  Targets of priority determination are IDs of messages placed in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), and transmit queues.
  When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes a target of priority determination.
  When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the same FIFO buffer becomes a target of priority determination.
  When transmit queues are used, all messages in transmit queues are targets of priority determination.
  When the message that has same ID with the stored message into the transmit queue is stored, the transmission of message that has same stored ID is aborted or cancelled.

- Transmit Buffer Number Priority
  The message in the transmit buffer whose number is the lowest among buffers having transmit requests is transmitted first.
  When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to the linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again whichever transmit priority is selected.

[*1] When using the transmit queue, it is not possible to select the method; please select ID Priority.

4.2 DLC Check Setting

Set to enable or disable DLC check function.

When the DLC check function is enabled, DLC filter processing is performed for messages that pass through the acceptance filter processing.

When the DLC check function is disabled, DLC check is not performed after performing acceptance filter processing.

In DLC check, when the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing. When the DLC value of the received message is smaller than that of the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and a DLC error is present.

Please refer to “5. Receive Rule Table” for the receive rule.

4.3 DLC Replacement Function Setting

Set to enable or disable DLC replacement function.

DLC replacement is effective only when DLC check function is enabled.

When DLC replacement is enabled, the DLC value of the receive rule is stored in the buffer instead of the DLC value of he received message after the DLC value has passed through the DLC filter. In this case, a value of “00_H” is stored in each data byte that exceeds the DLC value of the receive rule.

When DLC replacement is disabled, the DLC value of the received message is stored in the buffer after the DLC value has passed through the DLC filter. In this case, all the data bytes in the received message are stored in the buffer.

Please refer to “5. Receive Rule Table” for the receive rule.

Table 4-1 DLC Filter Processing and DLC Replacement Processing

GCFG Register		Received Message DLC / Receive Rule DLC	Received Message	
DCE Bit	DRE Bit		Processing	Stored DLC
0 (DLC check disabled)	0 (DLC replacement is disabled)	Received message DLC < Receive rule DLC	Stores to buffer *1	Received message DLC
		Received message DLC >= Receive rule DLC		
		Receive rule DLC = 0		
	1 (DLC replacement is enabled)	Received message DLC < Receive rule DLC		
		Received message DLC >= Receive rule DLC		
		Receive rule DLC = 0		
1 (DLC check enabled)	0 (DLC replacement is disabled)	Received message DLC < Receive rule DLC	Discard (DLC error)	—
		Received message DLC >= Receive rule DLC	Stores to buffer	Received message DLC
		Receive rule DLC = 0	Stores to buffer	Received message DLC
	1 (DLC replacement is enabled)	Received message DLC < Receive rule DLC	Discard (DLC error)	—
		Received message DLC >= Receive rule DLC	Stores to buffer	Receive rule DLC *2
		Receive rule DLC = 0	Stores to buffer	Received message DLC

1 DLC check itself is not performed.

2 "00_H" is stored in each data byte that exceeds than the DLC of the receive rule.

4.4 Mirror Function Setting

Set to enable or disable mirror function.

The mirror function allows reception of own transmitted messages.

When the mirror function is in use, receive rules for which the GAFLLB bit in the GAFLIDj register is set to 0 are applied to the data processing for messages received from other CAN nodes. When own transmitted messages are received, receive rules for which the GAFLLB bit in the GAFLIDj register is set to 1.

Please refer to “Section 5, Receive Rule Table” for the receive rule.

Table 4-2 Message Targeted for Data Processing by Mirror Function

MME Bit of GCFGL Register	GAFLLB Bit of GAFLIDj Register	Message Targeted for Data Processing of Receive rule
0 (Mirror function is disabled)	0	Message received from other CAN node
	1	No targeted message
1 (Mirror function is enabled)	0	Message transmitted from other CAN node
	1	Own transmitted message transmitted

4.5 CAN Clock Source Setting

Set the CAN clock (fCAN) which is CAN clock source in DCS bit of GCFG register. The clocks available as the CAN clock source are shown below.

- clk_xincan
- clkc

4.6 Payload Overflow Mode Setting

Set the payload overflow mode in the CMPOC bit of the GCFG register. Select the operation when the payload length of the received message exceeds the payload storage size of the storage buffer.

When the bit is set to “0”, the received message which overflows the payload is not stored in the buffer.

When the bit is set to “1”, the received message which overflows the payload is stored in the buffer. Also, the received DLC value or the DLC value of the receive rule table is stored in the buffer depending on the DRE bit. At the time, the payloads exceeding the buffer’s payload storage size are discarded.

Set the payload storage size of the buffer by following bits.

- Receive buffer: RMPLS[2:0] bits in the RMNB register.
- Receive FIFO buffer: RFPLS[2:0] bits in the RFCCx register
- Transmit/receive FIFO buffer: CFPLS[2:0] bits in the CFCCk register

4.7 Timestamp Clock Setting

Set the clock source and the division ratio for using timestamp clock.
The timestamp counter is a 16-bit free-running counter used for recording the message reception time and transmission time. The timestamp counter value is fetched at the timing set with the TSCCFG[1:0] bits in the GFDCFG register and is then stored in a receive buffer or a FIFO buffer together with the message ID and data during data reception.

You can select the following clocks for using timestamp.

- pclk
- CANm nominal bit time clock

When the nominal CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When pclk is used as a clock source, the timestamp function is not affected by channel mode.

Figure 4-1 shows the timestamp function block diagram.

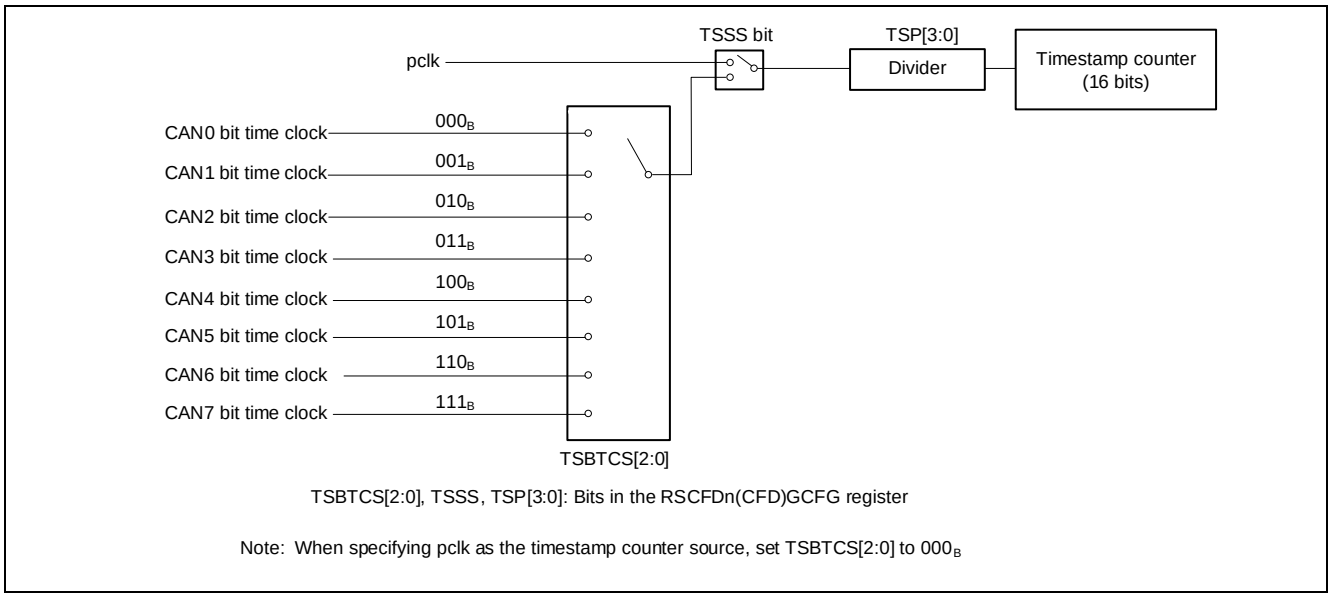


Figure 4-1 Timestamp Function Block Diagram

4.8 Interval Timer Prescaler Setting

Set the prescaler value when pclk is selected as an interval timer count source.

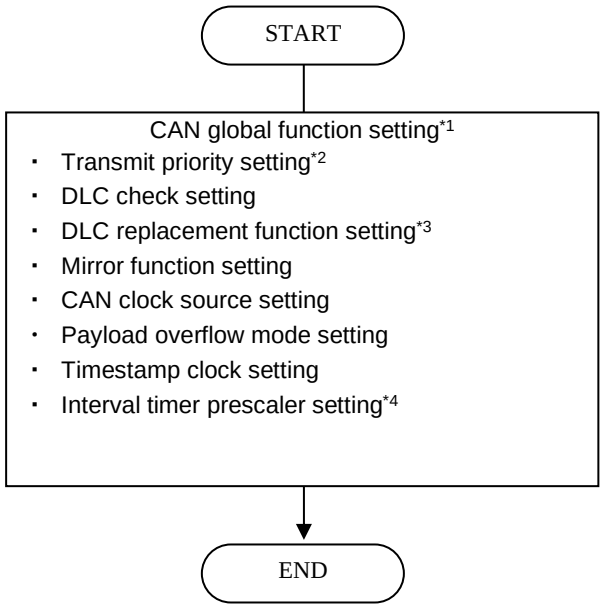
Please refer to “6.3.4 Interval Timer Counter Setting” for interval timer function.

4.9 Global Function Setting

Figure 4-2 shows global function setting procedures.

Perform these settings during CAN configuration.

Please refer to “1. CAN Configuration” for CAN configuration setting procedure.



Note 1. Rewrite the GCFG register in global reset mode.

Note 2. Select the ID priority (TPRI bit of GCFG register is “0”) when using transmit queue.

Note 3. If you disabled the DLC check function (DCE bit in the GCFG register is set to "0"), disable the function after setting “0” to DLC of receive rule table (GAFLDLC bit of GAFLP0j register).

Note 4. If you use pclk for interval timer counter source, do not set 0000_H to interval timer prescaler (ITRCP [15:0] bit of GCFG register).

Figure 4-2 Global Function Setting Procedure

5. Receive Rule Table

Set the receive rule table for filtering received message.

Data processing using the receive rule table stores selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

The following settings are required for receive rule.

5.1 Number of Receive Rules Setting
5.2 IDE/RTR/ID Setting
5.3 Receive Rule Target Message Setting
5.4 IDE Mask/RTR Mask/ID Mask Setting
5.5 DLC Check Value Setting
5.6 Routing Processing
5.7 Receive Rule Label Setting
5.8 Storage Buffer Setting
5.9 Use Example of Receive Rule
5.10 Receive Rule Table Setting Procedures

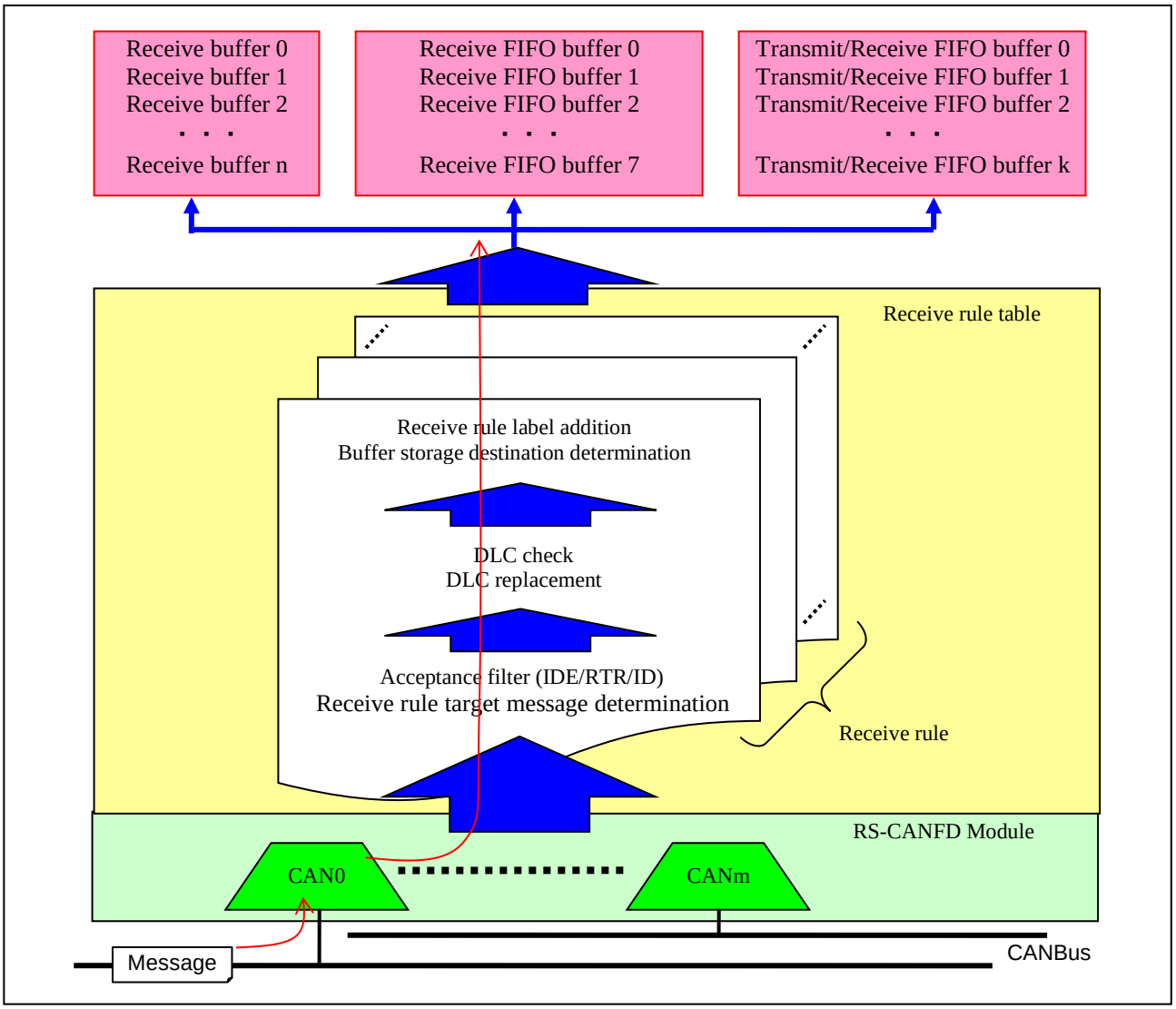


Figure 5-1 Filtering Image by Receive Rule Table

5.1 Number of Receive Rules Setting

Set the number of receive rules for each channel.

The number of receive rules for the entire module is 192 × number of channels for U2A-EVA, U2A16 and U2A8, and 128 × number of channels for U2A6. The maximum number of receive rules that can be registered per channel is 384 for U2A-EVA, U2A16 and U2A8, and 255 for U2A6.

Check processes begin with the lowest-numbered receive rule in ascending order. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. When there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

Table 5-1 shows the limitations on the number of receive rules that can be registered.

Table 5-1 Limitations on the number of receive rules

Limitations	Number of receive rule	
	U2A-EVA U2A16 U2A8	U2A6
Maximum registrations per CANm channel	384	255
Number of receive rules per CANm channel	192	128

Figure 5-2 shows the receive rule example.

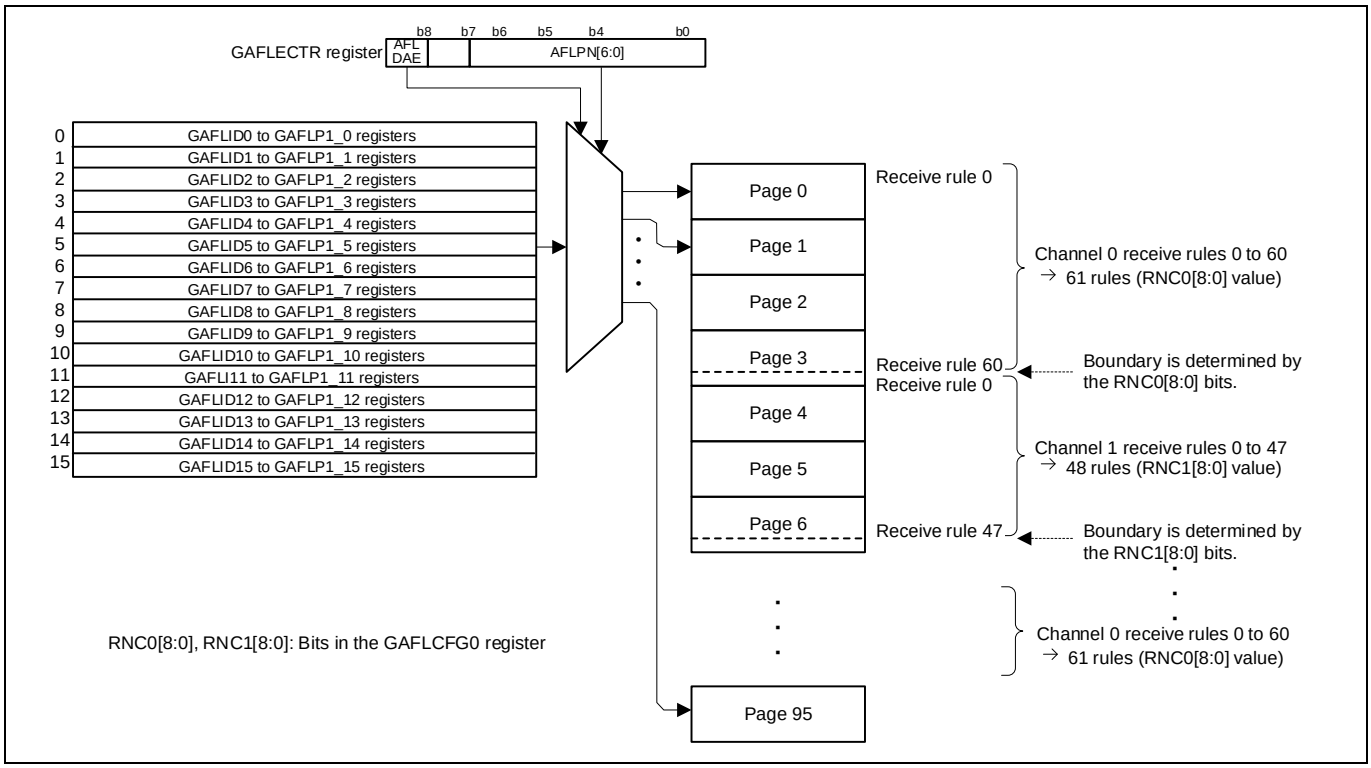


Figure 5-2 Receive Rule Register Example

5.2 IDE/RTR/ID Setting

Set the ID format (standard ID or extended ID), frame format (data frame or remote frame), and receive ID of the received message.

5.3 Receive Rule Target Message Setting

When you set a message transmitted from another CAN node as a target (GAFLLB bit of GAFLIDj register is “0”), data processing using the receive rule is performed when a message transmitted from another CAN node is received.

When you set the own transmitted message as the target when using the mirror function (GAFLLB bit is “1”), data processing using the receive rule is performed when the own transmitted message is received.

Please refer to “4.4 Mirror Function Setting” for mirror function.

5.4 IDE Mask/RTR Mask/ID Mask Setting

Set mask values for IDR/RTR/ID values.

The bits that are not masked in IDE mask/RTR mask/ID mask become effective in the acceptance filter processing.

5.5 DLC Check Value Setting

Set DLC value of the receive rule to be compared with DLC value of received message when the DLC check is enabled.

Please refer to “4.2 DLC Check Setting” for DLC check.

5.6 Routing Processing

When the payload length of the received message exceeds the payload storage size of the storage buffer, the CMPOF flag in the GERFL register is set to 1 (payload overflow) and the processing is handled according to the CMPOC bit in the GCFG register. When the CMPOC bit is “0”, the received message which exceeds the payload storage size is not stored in the buffer. When the CMPOC bit is “1”, the received message is stored in the buffer with payloads exceeding the storage size being discarded.

5.7 Receive Rule Label Setting

Set added 12-bit label information when message that passed through filter processing is stored to buffer.

Any value can be set as a label, and it is able to freely use received message label by program. For example, if you set the receiving channel number to the label, it is possible to check which channel the message with same ID in the receive FIFO buffer was received.

5.8 Storage Buffer Setting

Set the buffer to store message that passed through filter processing.

The buffers that can be selected as a storage destination are shown below.

- Receive buffer q (Only one buffer can be selectable per receive rule.)
- Receive FIFO buffer q
- Transmit/Receive FIFO buffer k (receive mode)
- Transmit Queue

Maximum 8 storage buffers are selectable for 1 receive rule. However, it is only possible to select one receive buffer as storage destination. (For example, it is not possible to store to receive buffer 0 and 1.)

Setting example of maximum storage destination)

Example 1)

○: Receive buffer 1 + Receive FIFO buffer 0,2,4 + Transmit/Receive FIFO buffer 0,3,6,9 (Total 8 buffers)

Example 2)

○: Transmit FIFO buffer 0,1,2,3 + Transmit/Receive FIFO buffer 0,3,6,9(Total 8 buffers)

Impossible setting example)

Example 3)

×: Store to the receive buffer 0, receive buffer 1, Receive FIFO buffer 2 *Storing in two receive buffers is impossible.

5.9 Use Example of Receive Rule

Use example of receive rule are shown below.

- Example 1

  Example of each register is shown when the following messages are received.

— ID format : Standard ID
— Message format : Data frame
— Mirror Function : Message receive from another CAN node
— Receive ID : 120_H, 121_H, 122_H, 123_H
— DLC : Receive message DLC≧6
— Label : 010_H
— Storage destination buffer : Receive buffer 3, Receive FIFO buffer 0, 1, 2

		GAFLIDE	GAFLIDEM	GAFLRTR	GAFLRTRM	GAFLLB	GAFLID/GAFLIDM			
							Bit28-24	Bit23-16	Bit15-8	Bit7-0
GAFLIDj		0	-	0	-	0	00000_B	00000000_B	00000001_B	00100000_B
GAFLMj		-	1	-	1	—	00000_B	00000000_B	00000111_B	11111100_B
Receivable message	120_H	0	0	0	0	0	-----$_B$	--------$_B$	-----001_B	00100000_B
	121_H						----- $_B$	-------- $_B$	-----001_B	00100001_B
	122_H						----- $_B$	-------- $_B$	-----001_B	00100010_B
	123_H						----- $_B$	-------- $_B$	-----001_B	00100011_B

	GAFLDLC	GAFLPTR	GAFLRMV	GAFLRMDP	GAFLSRD2	GAFLSRD1	GAFLSRD0	GAFLFDP
GAFLP0j	6	010_H	1	3	0	0	0	—
GAFLP1j	—	—	—	—	—	—	—	00000007_H

- Example 2
  Example of each register is shown when below messages are received.

— ID format : Extension ID
— Message format : Data flame
— Mirror function : Receive message from another CAN node.
— Receive ID : 130_H, 131_H, 2130_H, 2131_H
— DLC : Unused DLC check
— Label : 130_H
— Storage destination buffer : Receive FIFO buffer 4, 6, Transmit FIFO buffer 1, 2 (ch0)

		GAFLIDE	GAFLIDEM	GAFLRTR	GAFLRTRM	GAFLLB	GAFLID/GAFLIDM			
							Bit28-24	Bit23-16	Bit15-8	Bit7-0
GAFLIDj		1	-	0	-	0	00000_B	00000000_B	00000001_B	00110000_B
GAFLMj		-	1	-	1	—	11111_B	11111111_B	11011111_B	11111110_B
Receivable Message	130_H	1	0	0	0	0	00000_B	00000000_B	00000001_B	00110000_B
	131_H						00000_B	00000000_B	00000001_B	00110001_B
	2130_H						00000_B	00000000_B	00100001_B	00110000_B
	2131_H						00000_B	00000000_B	00100001_B	00110001_B

	GAFLDLC	GAFLPTR	GAFLRMV	GAFLRMDP	GAFLSRD2	GAFLSRD1	GAFLSRD0	GAFLFDP
GAFLP0j	0	130_H	1	3	0	0	0	—
GAFLP1j	—	—	—	—	—	—	—	00000650_H

Example 3

- Example of each register is shown when below messages are received.

— ID format : Unused ID check
— Message format : Data frame
— Mirror function : Receive message from another CAN node.
— Receive ID : All ID (Standard ID/Extension ID)
— DLC : DLC≧0xF (64 bytes) of receive message
— Label : 010_H
— Storage destination buffer : CAN0 transmit queue 0

		GAFLIDE	GAFLIDEM	GAFLRTR	GAFLRTRM	GAFLLB	GAFLID/GAFLIDM			
							Bit28-24	Bit23-16	Bit15-8	Bit7-0
GAFLIDj		0	-	0	-	0	-----$_B$	--------$_B$	--------$_B$	--------$_B$
GAFLMj		-	0	-	1	—	00000_B	00000000_B	00000000_B	00000000_B
Receivable Message	xxx_B	0	0	0	0	0	-----$_B$	--------$_B$	-----xxx_B	$xxxxxxxx_B$
	$xxxxxxxx_B$	1					$xxxxx_B$	$xxxxxxxx_B$	$xxxxxxxx_B$	$xxxxxxxx_B$

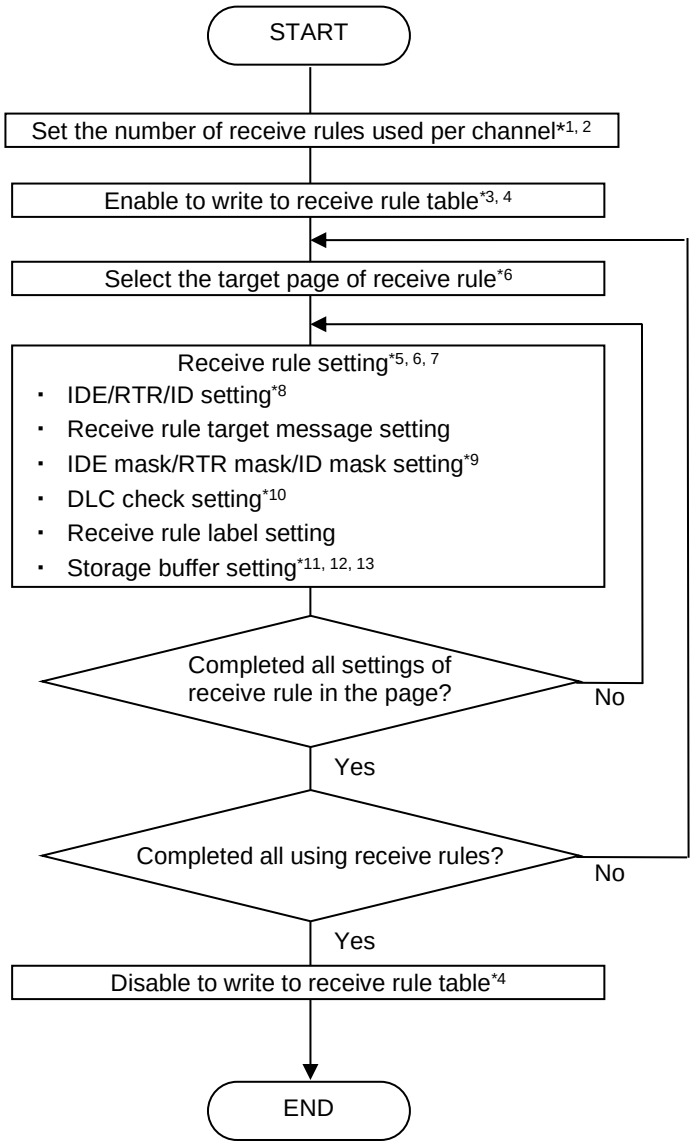
	GAFLDLC	GAFLPTR	GAFLRMV	GAFLRMDP	GAFLSRD2	GAFLSRD1	GAFLSRD0	GAFLFDP
GAFLP0j	F_H	010_H	0	0	0	0	1	—
GAFLP1j	—	—	—	—	—	—	—	00000100_H

5.10 Receive Rule Table Setting Procedures

Figure 5-3 shows the receive rule table setting procedure.

Set these settings during CAN configuration.

Please refer to “1. CAN Configuration” for CAN configuration processing.



Note 1. Rewrite number of receive rule setting (RNC(0 + v * 2)[8:0] bits and RNC(1 + v * 2)[8:0] bits in GAFLCFGv registers) in global reset mode.

Note 2. Meet following conditions for the number of receive rules used per channel(RNC(0 + v * 2)[8:0] bits and RNC(1 + v * 2)[8:0] bits).

・Set the number of rules per channel to 384 or less.

・The total of the number of rules allocated to each channel is not larger than the number of rules permitted to the entire module.

Note 3. After completion of writing to the receive rule table, set write disabled (AFLDAE bit in the GAFLECTR register is set to "0").

Note 4. Do not set the page more than number of pages that can be set in module to setting target page (AFLPN[6:0] in the GAFLECTR register).
Note 5. Rewrites receive rule(GAFLIDj register, GAFLMj register, GAFLP0j register, GAFLP1j register) under the condition of receive rule table write enable (AFLDAE bit is "1") and global reset mode.
Note 6. Set the receive rules for each channel consecutively. It is not possible to share or alternately set them with other channels.
Note 7. In standard ID, set standard ID values to bits 10 to 0 of ID (AFLID [28:0] bits in GAFLIDj register), and set “0” to bits 28 to 11.
Note 8. If IDE bit is not compared (GAFLIDEM bit in GAFLMj register is “0”), set all ID bits to not compare (set all GAFLIDM[28:0] bits in the GAFLMj register to 0).
Note 9. It is valid only when the DLC check function is enabled (DCE bit in the GCFG register is set to 1).
Note 10. Up to eight FIFO buffers can be selected. However, when the message is stored in the receive buffer (GAFLRMV bit in the GAFLP0j register is set to 1), up to seven FIFO buffers can be selected.
Note 11. Select only receive FIFO buffers or transmit/receive FIFO buffers set to receive mode or gateway mode.
Note 12. When selecting a receive buffer as the storage destination, enable storing to the receive buffer (GAFLRMV bit is “1”), and set the number smaller than the number of receive buffers to be used (setting value to NRXMBm[7:0] bit of RMNBy register).

Figure 5-3 Receive Rule Table Setting Procedure

6. Buffer, FIFO Buffer

Set buffers and FIFO buffers used for transmission and reception. Following buffer and FIFO buffer setting are required.

6.1 Receive Buffer Setting
6.2 Receive FIFO Buffer Setting
6.3 Transmit/Receive FIFO Buffer Setting
6.4 Transmit Buffer Setting
6.5 Transmit Queue Setting
6.6 Transmit History Buffer Settings
6.7 Buffer Setting Procedures
6.8 Flexible CAN Mode

Table 6-1 shows the limitations on the number of buffers that can be set for receive buffers, receive FIFO buffers, and transmit/receive FIFO buffers. Refer to “User’s Manual Hardware: Figure 23.27 Message Buffer Configuration” for each buffers configuration.

Table 6-1 Limitations on the number of buffers

		U2A-EVA U2A16 U2A8	U2A6 (BGA292)	U2A6 (BGA176)	U2A6 (BGA156)	U2A6 (QFP144)
Channels		16ch	12ch	11ch	8ch	7ch
Buffers		5120 buffers	1152 buffers	1056 buffers	768 buffers	672 buffers
Individual buffer for each channel		1024 (64×16ch)	384 buffers	352 buffers	256 buffers	224 buffers
			32×(chs)			
	Transmit buffer	Each channel 64 buffers	Each channel 32 buffers			
	Transmit queue	4 queues for each channel				
Shared buffer between channels		4096 buffers (256×16ch)	768 buffers	704 buffers	512 buffers	448 buffers
			64×(number of channels)			
	Receive buffer	256 buffers (16×16ch)	64 buffers			
	Receive FIFO buffer	8 FIFO (Up to 128 buffers can be allocated per FIFO.)				
	Transmit/receive FIFO buffer	3 FIFO for each channel (Up to 128 buffers can be allocated per FIFO.)				

U2A6 has a limited number of transmit buffers compared to other U2A devices.

Also, note that the index values pointing to the transmit buffers for U2A6 are discontinuous.

Refer to “User's Manual Hardware: Table 23.4 Indices for Individual Products” for details.

6.1 Receive Buffer Setting

Set number of buffers assigned to receive buffers and payload size that can be stored per buffer. It is possible to assign buffers of 0~16×number of channels to receive buffer. Receive buffer is not able to use if “0” is set to number of receive buffer.

There are no interrupt-related settings because there are no receive buffer-related interrupts.

6.2 Receive FIFO Buffer Setting

The required settings to use the receive FIFO buffer are shown below.

- Number of buffers and payload size setting
- Interrupt enable/disable setting, and interrupt source setting

6.2.1 Number of Buffers setting

Set number of buffers assigned to the receive FIFO buffer and payload size.

There are 8 receive buffers, and maximum 128 buffers can be assigned.

Number of buffers assigned to receive FIFO buffer can be selected from 0[*3], 4, 8, 16, 32, 48, 64, 128.

6.2.2 Interrupt enable/disable Setting, and Interrupt Source Setting

- Receive FIFO Interrupt

  Enable/Disable the receive FIFO interrupt and set the interrupt source. When using the receive FIFO interrupt, the interrupt source can be selected from the following.

  - Receive FIFO interrupt occurs when the following condition selected by RFIGCV[2:0] bits in the RFCCx register is met. (RFIM bit in the RFCCx register is set to "0".)
    - When a message is stored up to 1/8 of the receive FIFO buffer.[*4]
    - When a message is stored up to 2/8 of the receive FIFO buffer.
    - When a message is stored up to 3/8 of the receive FIFO buffer.[*2]
    - When a message is stored up to 4/8 of the receive FIFO buffer.
    - When a message is stored up to 5/8 of the receive FIFO buffer. [*2]
    - When a message is stored up to 6/8 of the receive FIFO buffer.
    - When a message is stored up to 7/8 of the receive FIFO buffer. [*2]
    - When receive FIFO buffer is full.
  - Receive FIFO interrupt occurs every time when a message reception is completed (RFIM bit of RFCCx register is “1”).

---

3 If you do not use receive FIFO buffer, set 0 message (RFDC[2:0] bit of RFCCx register is “000_B”) to the number of buffers in the receive FIFO buffer.

4 When setting 4 messages to the number of buffers in the receive FIFO buffer (setting 001B to the RFDC[2:0] bits), do not select this condition.

- Receive FIFO Full Interrupt Processing

  If enables the receive FIFO buffer (RFFIE bit of RFCCx register is “1”.), the receive FIFO full interrupt is occurred when the receive FIFO buffer is full.

  Even if the use of receive FIFO buffer is disabled (RFE bit is “0”) while interrupt request is occurred (RFFIF flag of RFSTSx register is “1”), the interrupt request flag (RFFIF flag) does not become “0” automatically. Set “0” to the interrupt request flag by the program.

6.3 Transmit/Receive FIFO Buffer Setting

Required settings for using transmit/receive FIFO are shown below.

- Number of buffers setting
- Interrupt enable/disable setting, and interrupt source setting
- Transmit/Receive FIFO mode setting
- Interval timer counter setting (Transmit mode, gateway mode)
- Transmit buffer link setting (Transmit mode, gateway mode)

6.3.1 Number of buffers setting

Set the number of buffers for the transmit/receive FIFO buffer.

There are three transmit/receive FIFO buffers for each channel, and maximum 128 buffers can be assigned. Number of buffers assigned to transmit/receive FIFO buffer can be selected from 0[*5], 4, 8, 16, 32, 48, 64, 128.

[5] If the transmit/receive FIFO buffer is not used, set the number of buffers in the transmit/receive FIFO buffer to 0 message (CFDC[2:0] bits in the CFCCLk register to “000_B”).

6.3.2 Interrupt Enable, Disable, Interrupt Source Setting

Set interrupt enable/disable setting, and interrupt source setting of each transmit/receive FIFO buffer. Table 6-2 shows settable interrupt source for each transmit/receive FIFO mode. Transmit/Receive FIFO transmission completion interrupt is the occurrence fact of CANm transmit interrupt. Refer to “9. CAN-related Interrupt” for CANm transmit interrupt occurrence factor.

Table 6-2 Transmit/Receive FIFO Buffer Interrupt Source

Transmit/Receive FIFO Mode	Interrupt Source			Interrupt Factor
Receive mode	Transmit/Receive FIFO receive interrupt	CFCCk.CFIM	0	When the number of received messages reaches the condition set in CFIGCV[2:0] bits in the CFCCk register, a transmit/receive FIFO receive complete interrupt occurs. CFIGCV[2:0] bit setting 000_B: When a message is stored up to 1/8 of the transmit/receive FIFO buffer.[*6] 001_B: When a message is stored up to 2/8 of the transmit/receive FIFO buffer. 010_B: When a message is stored up to 3/8 of the transmit/receive FIFO buffer.[*1] 011_B: When a message is stored up to 4/8 of the transmit/receive FIFO buffer. 100_B: When a message is stored up to 5/8 of the transmit/receive FIFO buffer.[*1] 101_B: When a message is stored up to 6/8 of the transmit/receive FIFO buffer. 110_B: When a message is stored up to 7/8 of the transmit/receive FIFO buffer. 111_B: When the transmit/receive FIFO buffer is full.
			1	Receive complete interrupt of transmit/receive FIFO is occurred when a message reception is completed.
	Transmit/Receive FIFO one-frame receive interrupt			Trasnmit/Receive FIFO receives 1 frame message.
	Transmit/Receive FIFO Full interrupt			Transmit/Receive FIFO is full.
Transmit mode	Transmit/Receive FIFO transmission completion interrupt	CFCCk.CFIM	0	Receive complete interrupt of transmit/receive FIFO is occurred when buffer becomes empty by completed message transmit.
			1	Receive complete interrupt of transmit/receive FIFO is occurred when a message transmit is completed.
	Transmit/Receive FIFO one-frame transmit interrupt			Trasnmit/Receive FIFO transmits 1 frame message.

[6] Do not set if you set 4 messages (RFDC[2:0] bit is “001_B”) to the number of buffers in the transmit/receive FIFO buffer.

6.3.3 Transmit/Receive FIFO Mode Setting

Set transmit/receive FIFO buffer mode. Settable from receive mode, transmit mode, and gateway mode.

— Receive mode
It operates as receive FIFO buffer.
— Transmit mode
It operates as transmit FIFO buffer

6.3.4 Interval Timer Counter Setting

Set counter source of interval timer counter and transmission interval. The interval timer counter is effective in transmit mode and gateway mode.

Table 6-3 shows the count source of interval timer counter and formula of interval time.

Table 6-3 Count Source of Interval Timer Counter and Formula of Interval Time

CFITR bit and CFITSS bit of CFCCk register	Count Source	Formula*
00_B	Clock obtained by dividing pclk by the value of ITRCP[15:0] bits in GCFGH register.	$1/f_{CLK} \times 2 \times a \times b$
10_B	Clock obtained by dividing pclk by 10 times the value of ITRCP[15:0] bits in GCFGH register.	$1/f_{CLK} \times 2 \times a \times 10 \times b$
$x1_B$	CANm nominal bit time clock	$1/f_{CANBIT} \times b$

a : pclk prescaler value (setting value of ITRCP[15:0] bits)
b : Setting value of message transmit interval (CFITT[7:0] bits in CFCCk register)
f_{CLK} : pclk frequency
f_{CANBIT} : CANm nominal bit time clock frequency

6.3.5 Transmit Buffer Link Setting

Link transmit/receive FIFO buffer to transmit buffer. Linking to the transmit buffer is valid only in transmit mode and gateway mode.

Do not assigin transmit buffer linked to transmit/receive FIFO buffer to transmit qeue. Only one transmit/receive FIFO buffer can be linked to one transmit buffer. Do not link multiple transmit/receive FIFO buffers to the same numbered transmit buffer.

6.3.6 Transmit/Receive FIFO Buffer Overwrite Mode

In the gateway mode, if a transmit/receive FIFO buffer attempts to receive a new message while the transmit/receive FIFO buffer is full, the oldest data buffer is overwritten with the received message or the message is discarded. This operation is determined by the CFMOWN in the CFCCEk register.

CFMOWM bit in CFCCEk register	Transmit/Receive FIFO Buffer Operation
0_B	If the transmit/receive FIFO buffer is full, the received message is discarded.
1_B	If the transmit/receive FIFO buffer is full, the buffer with the oldest data is overwritten with the received message.

6.4 Transmit Buffer Setting

Set to enable or disable the transmit complete interrupt for each transmit buffer.

There are 64 transmit buffers per channel for U2A-EVA, U2A16, and U2A8, and 32 buffers per channel for U2A6, which can be used for either transmit buffers, for linking to transmit/receive FIFO buffers (in transmit mode or gateway mode), or for transmit queues.

When using the transmit buffer as a link to the transmit/receive FIFO buffer (in transmit mode or gateway mode) or as a transmit queue, the corresponding TMCp register should be set to "00". The TMIEp bit of the corresponding TMIECy register should be set to "0" (interrupt disabled).

In addition, the transmit complete interrupt becomes a source of CANm transmit interrupt. Refer to “9. CAN-related Interrupt” for the occurrence factor of CANm.

6.5 Transmit Queue Setting

The settings required to use the transmit queue are shown below.

- Number of buffers setting
- Interrupt enable/disable setting, and interrupt source setting

6.5.1 Number of Buffers Settings

Set the number of buffers in the transmit queue.

There are four transmit queues per channel, and up to 32 buffers can be allocated to each transmit queue for U2A-EVA, U2A16, and U2A8, and up to 16 buffers for U2A6.

Table 6-4 shows the access window, number of stages, buffer allocation direction, routing, CPU access, and DMA access for TXQ0 to TXQ3.

Table 6-4 TXQ0 to 3 Setting

Queue	Access Window	Satges		Buffer Allocation		HW routing access*	CPU access	DMA access
		U2A-EVA U2A16 U2A8	U2A6	U2A-EVA U2A16 U2A8	U2A6			
TXQ0	TXMB0	0,3-32	0,3-16	TXMB0 -> TXMB31	TXMB0 -> TXMB15	Possible	Possible	Possible
TXQ1	TXMB31	0,3-32	0,3-16	TXMB31 -> TXMB0	TXMB15 -> TXMB0	Possible	Possible	Impossible
TXQ2	TXMB32	0,3-32	0,3-16	TXMB32 -> TXMB63	TXMB32 -> TXMB47	Possible	Possible	Impossible
TXQ3	TXMB63	0,3-32	0,3-16	TXMB63 -> TXMB32	TXMB47 -> TXMB32	Impossible	Possible	Possible

HW routing access and CPU access/DMA access can not be used at the same time.

When using transmit queue 0 and transmit queue 1, set the total number of stages to 32 or less for U2A-EVA, U2A16, and U2A8, and 16 or less for U2A6.

When using transmit queue 2 and transmit queue 3, set the total number of stages to 32 or less for U2A-EVA, U2A16, and U2A8, and 16 or less for U2A6.

When using a transmit queues, the transmit priority should be set to ID Priority.

* Access by gateway mode.

6.5.2 Interrupt Enable/Disable Setting, and Interrupt Source Settings

Set interrupt enable/disable settings and interrupt sources for the transmit queue. The table 6-5 shows the configurable insterrupt sources when using the transmit queue interrupt.

Table 6-5 Transmit Queue Interrupt Source

Queue Status	Interrupt Source			Interrupt Source
In Transmit	Transmit queue transmit completion interrupt	TXQCC.TXQIM	0	When the buffer becomes empty due to the completion of message transmission, the transmit queue transmit completion interrupt request occurs.
			1	Transmit queue transmit completion interrupt request occurs for each message transmission completed.
	Transmits queue one-frame transmit interrupt			When the transmit queue transmitted a message for one frame.
GW Mode[*1]	Transmits queue one-frame receive inetrrupt			When the transmit queue received a message for one frame.
	Transmit queue full interrupt			When the transmit queue is full.

The transmit queue interrupt is also a source of CANm transmit-related interrupt. Refer to “9. CAN-related Interrupts” for the source of CANm transmit interrupts.

6.5.3 Transmit Queue Overwrite Mode

The transmission queue overwrite mode setting allows the user to select message overwrite/discard when a message with the same ID is stored in the transmission queue.

- When the transmit queue overwrite mode is selected[*2]

  When a message with the same ID as the message stored in the transmit queue is stored, the stored message with the same ID is overwritten.

[*1] The transmit queue 3 cannot be set to GW mode.

[*2] The transmit queue overwrite mode can be set for each transmit queue, set by the TXQOWE bit of TXQCC0 to 3m register.

6.6 Transmit History Buffer Settings

The settings required to use the transmit history buffer are shown below.

The transmit history buffer can store 64 transmit history data per channel.

- Storage target buffer setting
- Interrupt enable/disable and interrupt source settings

6.6.1 Storage Target Buffer Setting

Set target buffer for storing transmit history data to transmit history buffer. The target buffer to be stored can be selected from the following.

Also, you can set whether or not to store the transmit history data of the message at the time of storing the transmit message.

- Entries from transmit/receive FIFO buffers and transmit queues.
- Entries from transmit buffers, transmit/receive FIFO buffers, and transmit queues.

6.6.2 Interrupt Enable/Disable and Interrupt Source Settings

Set transmit history interrupt enable/disable setting and interrupt source. Transmit history buffer interrupt sources are shown below.

- Transmit history interrupt is occurred when data is stored up to 3/4 of the number of transmission history buffer stages.
- Transmit history interrupt is occurred each time a transmit history data storage is completed.

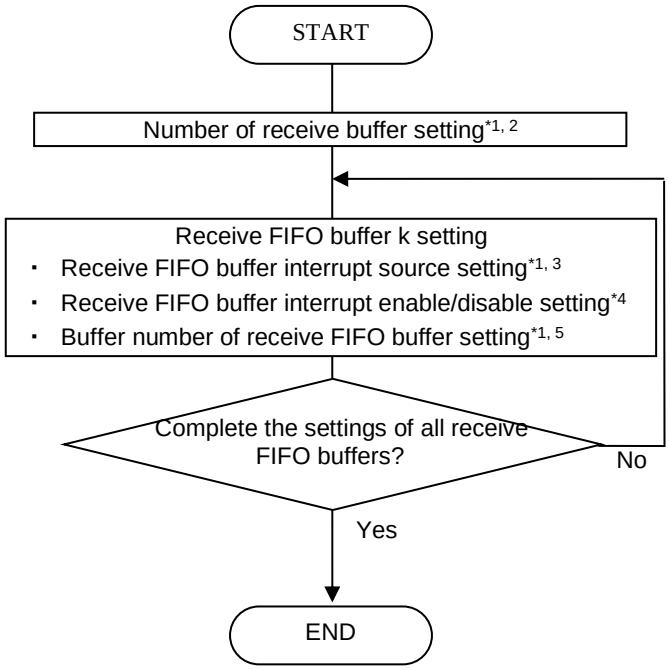
In addition, transmit history interrupt becomes a source of CANm transmit interrupt. Refer to “9. CAN-related Interrupts” for the source of CANm transmit interrupts.

6.7 Buffer Setting Procedures

Figure 6-1 shows the procedure for setting the receive buffer and receive FIFO buffer, and Figure 6-2 shows the procedure for setting the transmit FIFO buffer, transmit buffer, and transmit history buffer.

These settings should be performed during CAN configuration.

Please refer to “1 CAN Configuration” for CAN configuration procedure.



Note 1. Rewrite the RMNB register, the RFIGCV[2:0] bits, the RFIM bit, and the RFDC[2:0] bits in the RFCCx register in the global reset mode.

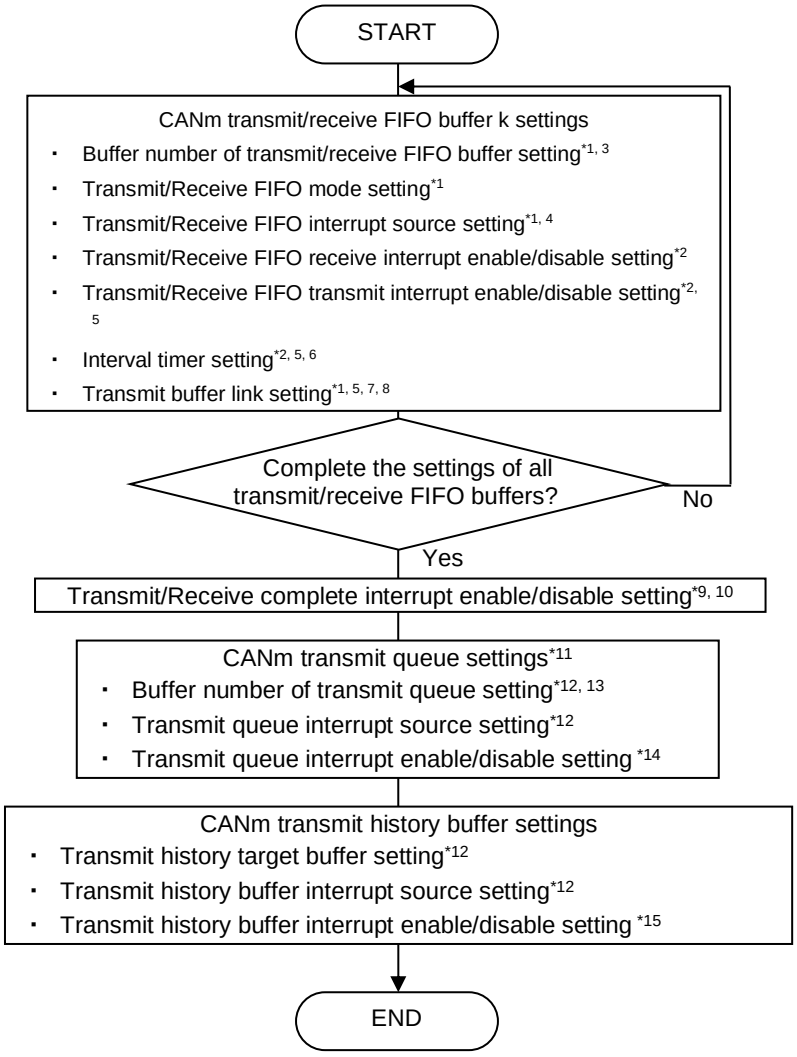
Note 2. The receive buffer cannot be used if the number of receive buffer (NRXMB[7:0] bits in RMNB register) is set to “0”. Also, set the number in the range of 0 to 16 x number of channels.

Note 3. If you set the number of buffers in the receive FIFO buffer to 4 messages (RFDC [2:0] bit is “001_B”), please set the RFIGCV[2:0] bits to “001_B”, “011_B”, “101_B”, or “111_B”.

Note 4. Perform the receive FIFO buffer enable/disable (RFIE bit in RFCCx register) settings in the state the receive FIFO buffer is not used (RFE bit of RFCCx register is “0”).

Note 5. Do not use receive FIFO buffer if buffer number of receive FIFO buffer is set to “0” (RFDC[2:0] bits are “000_B”).

Figure 6-1 Receive Buffer and Receive FIFO Buffer Setting Procedures



Note 1. Rewrite the CFDC [2:0] bits, the CFIM bit, the CFIGCV [2:0] bits, the CFM[1:0] bits, and the CFTML[4:0] bits in the CFCCk register in the global reset mode.

Note 2. Rewrite the CFRXIE bit, the CFTXIE bit, the CFITR bit, the CFITSS bit, and the CFITT [7:0] bits in the CFCCk register in the state the transmit/receive FIFO buffer is not used (CFE bit in CFCCk register is “0”).

Note 3. Set it to zero message (CFDC [2:0] bit to “000_B”) if the transmit/receive FIFO buffer is not used.

Note 4. If you set four messages (CFDC [2:0] bit is “001_B”) to the number of buffers in the transmit/receive FIFO buffer, please set “001_B”, “011_B”, “101_B”, or “111_B” to CFIGCV [2:0] bit.

Note 5. Valid only for transmit/receive FIFO buffers set to transmit mode or gateway mode.

Note 6. Set “0” to the message transmit interval (CFITT [7:] bits in CFCCk register) if the interval timer is not used.

Note 7. Set the different values to the link destination (CFTML [4:0] bit) of the transmit/receive FIFO buffers (transmit mode, gateway mode) in the same channel.

Note 8. Do not set the transmit buffer assigned to the transmit queue to the link destination (CFTML[4:0] bits) of the transmit/receive FIFO buffer (transmit mode and gateway mode).

Note 9. Perform the transmit buffer interrupt enable/disable (TMIEp bit of TMIECy register) setting when the corresponded transmit buffer has no transmit request (TMTR flag in TMSTSp register is “0”).

Note 10. Set the TMIEp bit corresponding to the transmit buffer linked to the transmit FIFO buffer or the transmit buffer assigned to the transmit queue to "0".

Note 11. When using the transmit queue, set the transmit priority to ID priority (TPRI bit in the GCFG register to "0").

Note 12. Rewrite the TXQDC[4:0] bits and TXQIM bit in the TXQCC0 to 3m register and the THLDTE bit and THLIM bit in the THLCCm register in the channel reset mode.

Note 13. The transmit queue cannot be used if the number of buffers (TQDC[4:0] bits) of the transmit queue is set to "0". Also, do not set “1”.
Note 14. Perform the transmit queue interrupt enable/disable (TXQFIE bit, TXQOFRXIE bit and TXQOFTXIE bit in TXQCC0 to 3m register) settnig when the transmit queue is not in use (TQE bit is “0”).
Note 15. Rewrite when the transmit history buffer is not in use (THLE bit in THLCCm register is “0”).

Figure 6-2 Transmit Buffer, Transmit/Receive FIFO Buffer, and Transmit History Buffer Setting Procedures

6.8 Flexible CAN Mode

In Flexible CAN mode, two channels can be connected and treated like a single CAN channel.

Refer to “User’s Manual Hardware: Figure 23.58 Diagram of the Flexible CAN” for flexible CAN mode channel connection.

The pairs of CAN channels that can be connected in flexible CAN mode are shown below.

- Channel 0 and channel 1 (Set FLXC0 bit in GFCMC register)
- Channel 2 and channel 3 (Set FLXC1 bit in GFCMC register)
- Channel 4 and channel 5 (Set FLXC2 bit in GFCMC register)
- Channel 6 and channel 7 (Set FLXC3 bit in GFCMC register)

In Flexible CAN mode, odd-numbered channels (channels 1, 3, 5, 7) use the input/output pins of even-numbered channels (channels 0, 2, 4, 6). The input/output pins of odd-numbered channels (channels 1, 3, 5, 7) set to Flexible CAN mode cannot be used.

In Flexible CAN mode, each channel performs communication processing independently, but if one channel is performing transmission, the other channel does not return the ACK bit.

6.9 Transmit Buffer Allocation

There are 64 transmission buffers per channel, but by using the flexible transmission buffer assignment function, up to 64 + 32 (U2A-EVA, U2A16, U2A8) or 16 (U2A6) transmission buffers can be allocated. Refer to “Figure 23.60 Flexible transmission buffer assignment” for the buffer allocation example in the flexible CAN mode.

The channels that can lend and borrow buffers between each channel are shown below.

- Channel 0 and channel 1 (FLXMB0 bit in GFTBAC register)
- Channel 2 and channel 3 (FLXMB1 bit in GFTBAC register)
- Channel 4 and channel 5 (FLXMB2 bit in GFTBAC register)
- Channel 6 and channel 7 (FLXMB3 bit in GFTBAC register)

The number of allocated buffers can be set for each pair with the FLXMBv [3: 0] bits in the GFTBAC register. Table 6-6 shows the number of buffers that can be set and the setting value of FLXMBv.

Table 6-6 Setting the number of allocated buffers

Number of allocated buffers	FLXMBv setting value[*1]
0	0000_B
4	0001_B
8	0010_B
12	0011_B
16	0100_B
20	0101_B
24	0110_B
28	0111_B
32	1000_B

It is prohibited to set GFTBAC (buffer allocation) and GFCMC (configuration) at the same time.
The lending buffer interrupt occurs on the lending channel.
In the case of using transmit queues, allocate a buffer for each channel to the transmit queue for each channel.
The state of the buffer changes depending on the mode of the lending channel.
Example) If the flexible transmission buffer assignment function is used on channels 0-1, channel 0 cannot lend a buffer on channel 1 if channel 1 is reset.

[*1] Setting values for FLXMBv other than those shown in Table 6-6 are prohibited.

7. Global Error Interrupt

Set global error interrupt. When the corresponding interrupt enable bits are enabled, an interrupt request is output from the CAN module. The occurrence of interrupts also depends on the interrupt control register settings of the interrupt controller.

7.1 Global Error Interrupt Setting

The following is a list of sources of the global error interrupt.

7.1.1 DLC Check Error
7.1.2 FIFO Message Lost
7.1.3 Transmit History Buffer Entry Lost Error
7.1.4 CAN-FD Message Payload Overflow
7.1.5 Transmit Queue Message Overwrite
7.1.6 Transmit Queue Message Lost
7.1.7 GW FIFO Message Overwrite

7.1.1 DLC Check Error

When DLC check is enabled, this is detected in case DLC of message in DLC check after passing acceptance filtering process is smaller than DLC of receive rule.

7.1.2 FIFO Message Lost

This is detected when the receive FIFO buffer or send/receive FIFO buffer is at FIFO full and a further attempt is made to store a new received message in the FIFO.

7.1.3 Transmit History Buffer Entry Lost Error

This is detected when the transmission history buffer is full and a new transmission history data is about to be stored in the transmission history buffer.

7.1.4 CAN-FD Message Payload Overflow

This is detected when the payload length of received message exceeds payload storage size of the destination buffer.

7.1.5 Transmit Queue Message Overwrite

This is detected when the message is overwritten in the transmit queue.

7.1.6 Transmit Queue Message Lost

This is detected when the transmit queue is full and a new received message is about to be stored in the transmit queue.

7.1.7 GW FIFO Message Overwrite

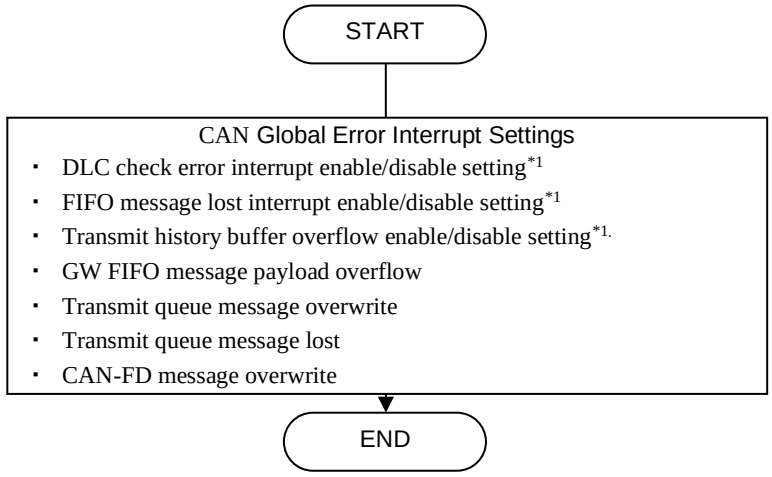
This is detected when the message is overwritten to the transmit/receive FIFO.

7.2 Global Error Interrupt Setting Procedure

Figure 7-1 shows global error interrupt setting procedure.

Please perform these settings during CAN configuration.

Please refer to “1 CAN Configuration” for CAN configuration procedure.



Note 1. Rewrite the THLEIE bit, the MEIE bit, the DEIE bit, the CMPOFIE bit, the QOWEIE bit, QMEIE bit, and MOWEIE bit of the GCTR register in the global reset mode.

Figure 7-1 Global Error Interrupts Setting Procedure

8. Channel Function

Set the following functions for each channel.

- 8.1 Channel Error Interrupt
- 8.2 CANm Transmit Abort Interrupt
- 8.3 Bus Off Recovery Mode Setting
- 8.4 Error Display Mode Setting
- 8.5 Communication Test Mode Setting
- 8.6 Gateway Mode setting
- 8.7 Channel Function Setting Procedure

8.1 Channel Error Interrupt

Set to enable or disable channel error interrupts. The following is a list of sources of the channel error interrupt.

- Bus error
- Error warning
- Error passive
- Bus off entry
- Bus off recovery
- Overload flame transmit
- Bus lock
- Arbitration lost

8.1.1 Bus Error

The interrupt is occured when any one of the following is detected.

— A form error is detected in the ACK delimiter (ADERR flag in CmERFLL register is “1”).
— A recessive is detected even though a dominant is transmitted (B0ERR flag in CmERFL register is “1”).
— A dominant is detected even though a recessive is transmitted (B1ERR flag in CmERFL register is “1”).
— A CRC error is detected (CERR flag in CmERFL register is “1”).
— An ACK error is detected (AERR flag in CmERFL register is “1”).
— A form error is detected (FERR flag in CmERFL register is “1”).
— A stuff error is detected (SERR flag in CmERFL register is “1”).

8.1.2 Error Warning

The interrupt is occurred when an error warning condition (receive error counter or transmit error counter > 95) is detected. The interrupt is occurred only when the receive error counter or transmit error counter exceeds 95 for the first time.

8.1.3 Error Passive

The interrupt is occurred when an error passive condition (receive error counter or transmit error counter > 127) is detected. The interrupt is occurred only wheny the receive error counter or transmit error counter exceeds 127 for the first time.

8.1.4 Bus Off Entry

The interrupt is occurred when a bus off state (transmit error counter > 225) is detected.

The interrupt is also occurred when the bus off recovery mode setting is set to transition to channel halt mode at bus-off entry (BOM [1:0] bit in CmCTR register is "01_B") and a bus-off state is detected.

8.1.5 Bus Off Recovery

The interrupt is occurred when a return from the bus-off state is detected by detecting 11 consecutive recessive bits 128 times. Please refer to "8.3 Bus Off Recovery Mode Setting" for details.

8.1.6 Overload flame transmit

The interrupt is occurred in case transmit condition of overload frame is detected when receiving or transmitting.

8.1.7 Bus Lock

The interrupt is occurred in case a bus lock is detected.

It is judged as a bus lock when a 32-bit consecutive dominant is detected on the CAN bus in channel communication mode.

8.1.8 Arbitration Lost

The interrupt is occurred in case an arbitration lost is detected.

8.2 CANm Transmit Abort Interrupt

Set to enable or disable the transmit abort interrupt. The interrupt is occurred when transmit abort complete has been detected in case the transmit abort interrupt is enabled.

The transmit abort interrupt becomes a source of CANm transmit interrupt. Refer to "9 CAN-related Interrupt" for the source of the CANm transmit interrupt.

8.3 Bus Off Recovery Mode Setting

Set the operation when recovering bus off. Table 8-1, Figure 8-1 to Figure 8-4 show the operation of each bus off recovery mode.

Table 8-1 Operation When Recovering Bus Off

CmCTR BOM[1:0] Bit	Function	Bus Off Entry Interrupt	Bus Off Recovery Interrupt[*1]
00_B	ISO11898-1 compliant	Occur	Occur[*2]
01_B	Transitions to channel halt mode automatically at bus-off entry[*3, 4]	Occur	Not occur
10_B	Transitions to channel halt mode automatically at bus-off end[*3, 4]	Occur	Occur
11_B	Transitions to channel halt mode (in bus-off state) by program request	Occur	Occur[*5]

1. When transitioning to channel reset mode before detecting an 11 bit recessive 128 times (Set "01_B" to CHMDC[1:0] bit in CmCTR register), the interrupt is not occurred in case.
2. When transitioned to channel halt mode (CHMDC[1:0] is "01_B") before detecting a 11-bit consecutive recessive 128 times, it is not transitioned to channel halt mode until a 11 bit consecutive recessive is detected 128 times. The interrupt is not occurred in case of recovering from bus off forcibly (setting RTBO bit in CmCTR register to "1").
3. If a transition to the channel halt mode by the CAN module and a writing to the CHMDC[1:0] bits by the program occur simultaneously, the writing by the program has priority.
4. An automatic transition to channel halt mode is made only in channel communication mode (CHMDC[1:0] bit is "00_B").
5. When transitioned to channel halt mode by program request before detecting a 11-bit consecutive recessive 128 times during bus off, the interrupt is not occurred.

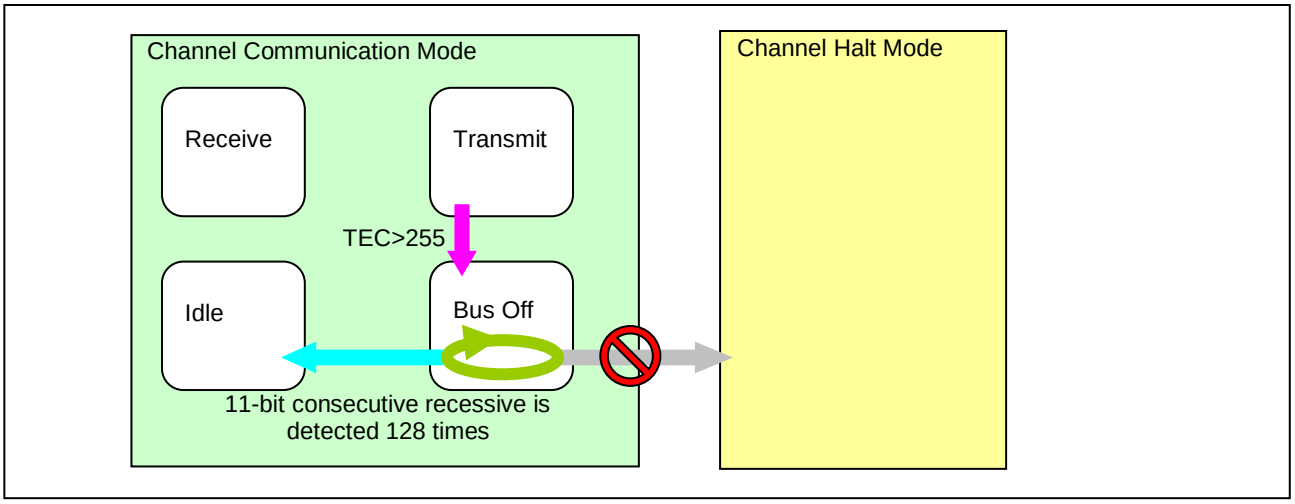


Figure 8-1 The operation when compliant with ISO11898-1 (BOM[1:0] bits is 00_B)

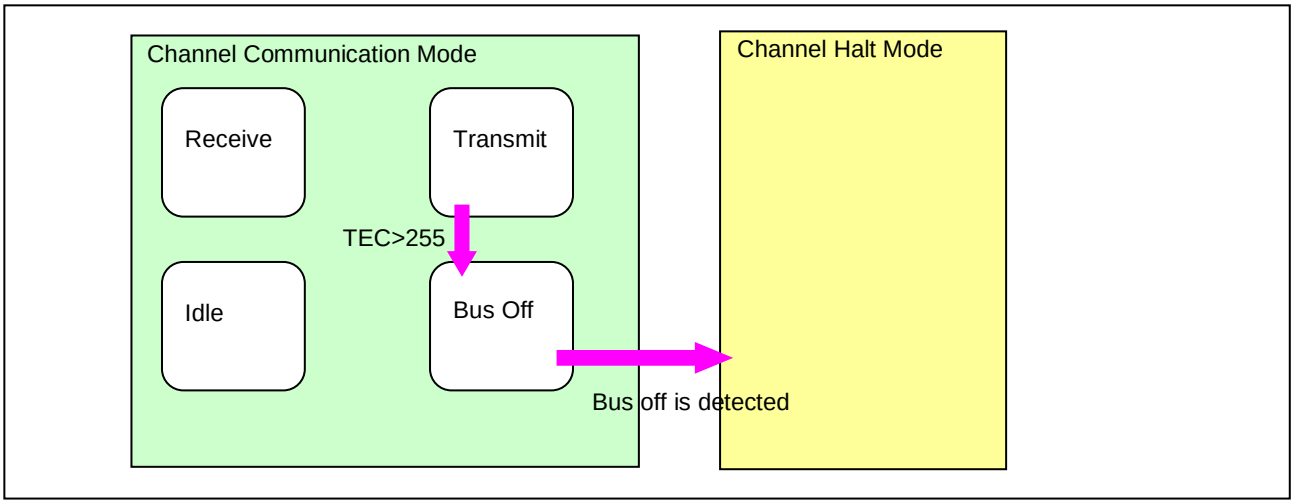


Figure 8-2 Operation at transition to channel halt mode at bus off entry (BOM[1:0] bits is 01_B)

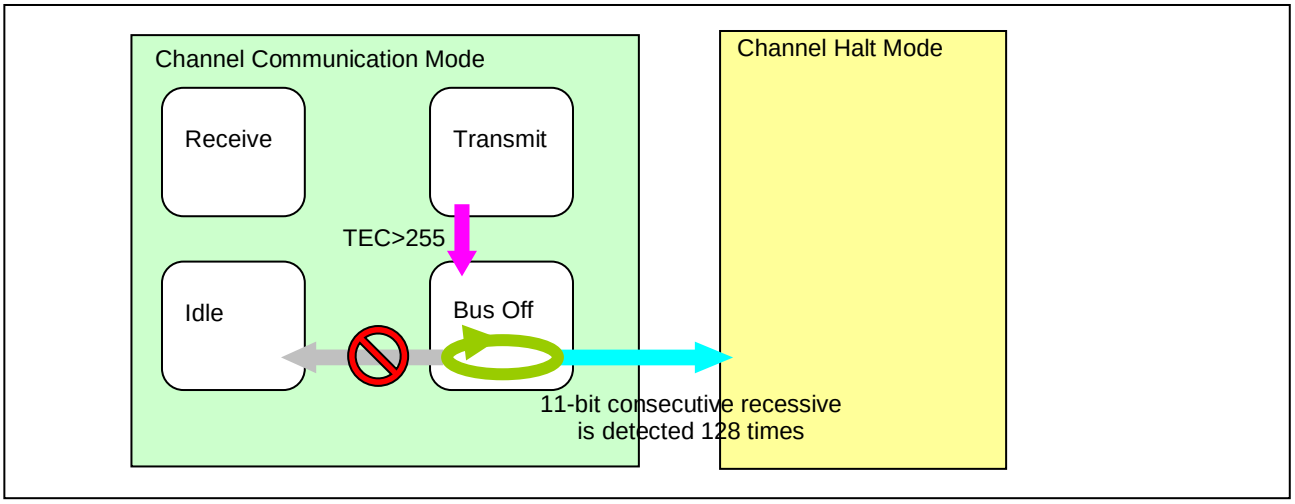


Figure 8-3 Operation at transition to channel halt mode at bus off end (BOM[1:0] bit is 10_B)

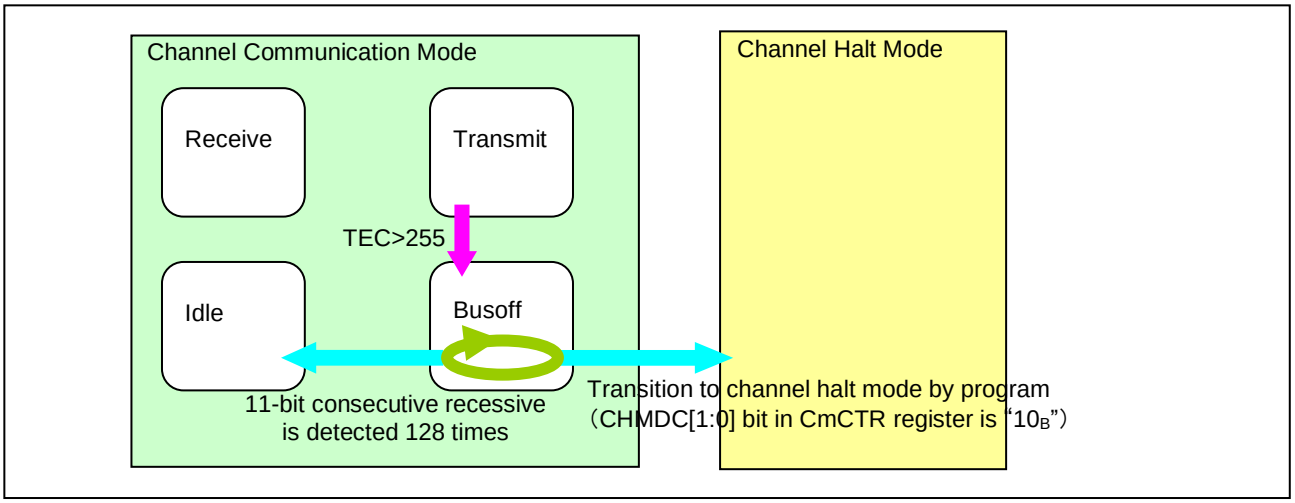


Figure 8-4 Operation when transitioning to channel halt mode by program request while bus off (BOM[1:0] bit is 11_B)

8.4 Error Display Mode Setting

Set the display mode for bits 14 to 8 of the CmERFL register when a CAN buss error occurs. The display mode available for setting are shown below.

- Display only the first error information (setting ERRD bit in CmCTR register to “0”)
  Only the first error flag is set to "1". If multiple errors occur at the same time, the flags for all detected errors are set to "1".
- Display all error information that occurred (setting ERRD bit to “1”)
  Regardless of the order of occurrence, all the flags of the errors that occur will be "1".

Figure 8-5 shows the operation example of CmERFL register in each error display mode.

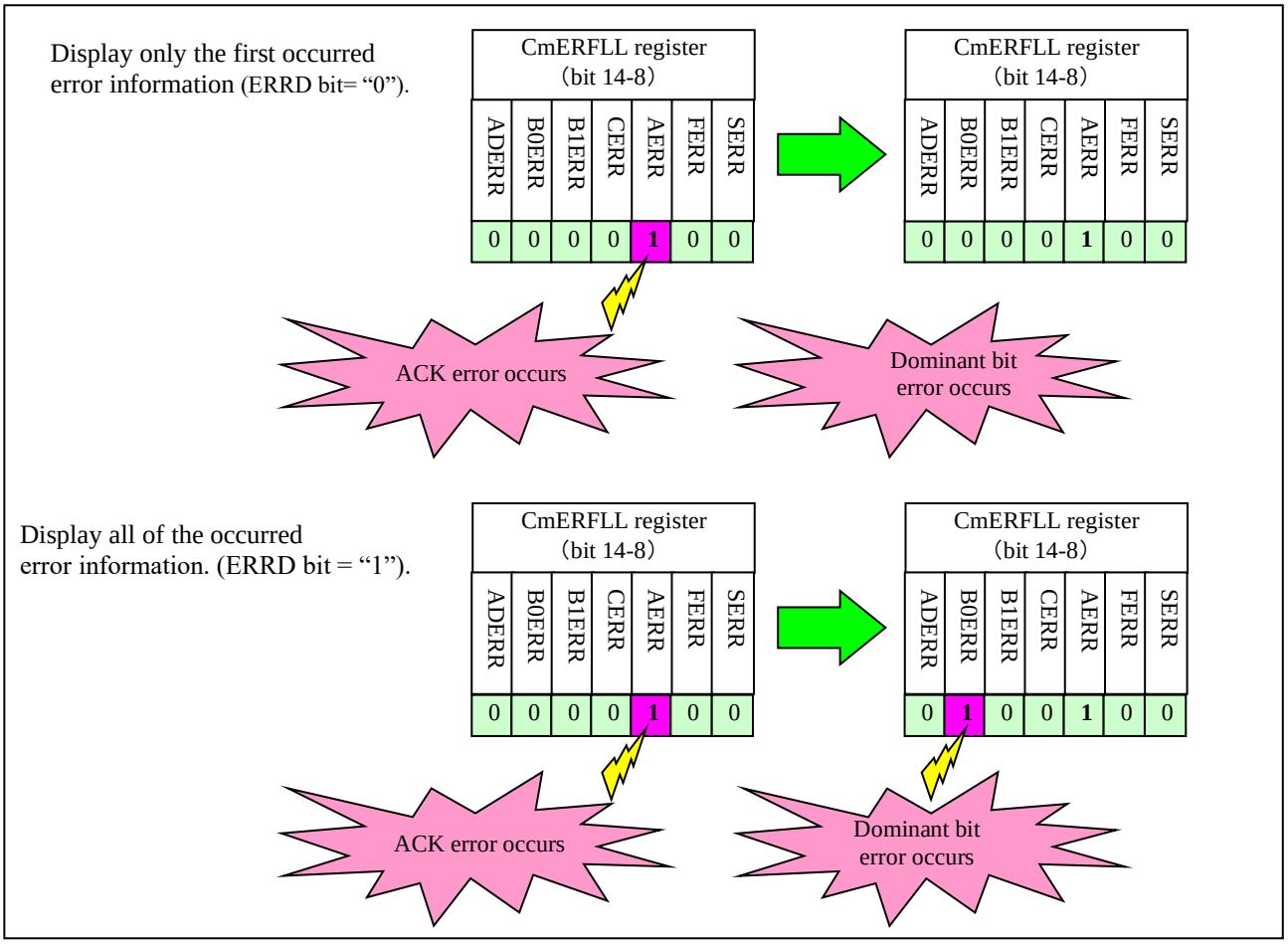


Figure 8-5 Operation Example of Error Display Mode

8.5 Communication Test Mode Setting

Set communication test mode. Please refer to “Test Mode Procedure Application Note” for the communication test mode.

8.6 Gateway Mode setting

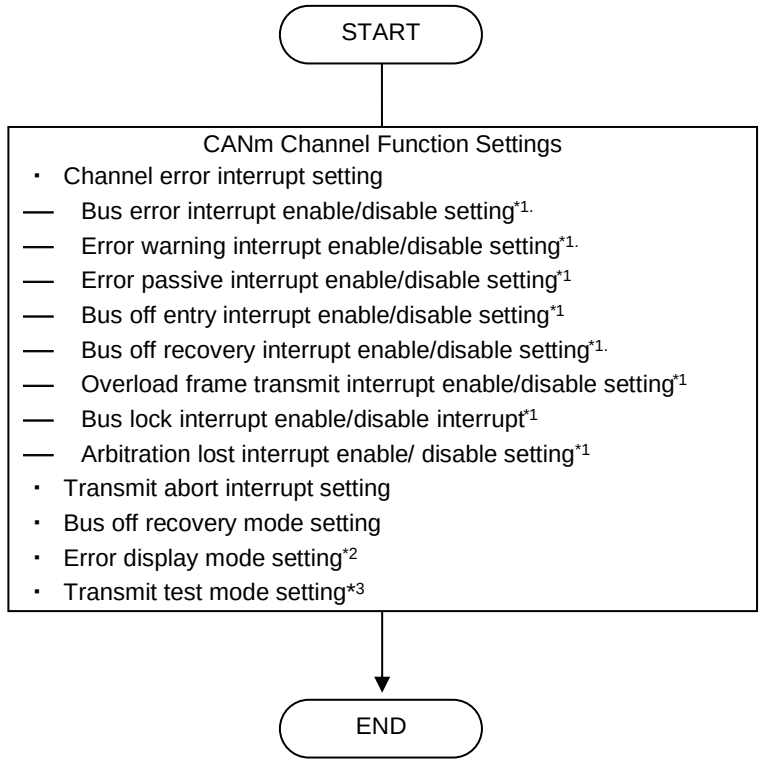
Set gateway mode. Please refer to “Gateway Mode Procedure Application Note” for the gateway mode.

8.7 Channel Function Setting Procedure

Figure 8-6 shows the channel function procedure.

Please perform these settings during CAN configuration.

Refer to “1 CAN Configuration” for procedure of CAN configuration.



Note 1. Rewrite the BEIE bit, the EWIE bit, the EPIE bit, the BOEIE bit, the BORIE bit, the OLIE bit, the BLIE bit, the ALIE bit, and the TAIE bit in the CmCTR register in the channel reset mode.

Note 2. Rewrite the ERRD bit in the CmCTR register in the channel reset mode or the channel halt mode.

Note 3. Rewrite the CTMS bit and the CTME bit in the CmCTR register in the channel halt mode. It becomes “0” in the channel reset mode.

Figure 8-6 Setting Procedure of Channel Function

9. CAN-related Interrupt

Set the corresponding EI level interrupt control register (EIC register) to enable/disable the CAN-related interrupt.

The available CAN-related interrupts are shown below.

- Global receive FIFO interrupt
- Global error interrupt
- CANm transmit interrupt
- CANm transmit/receive FIFO receive complete interrupt
- CANm error interrupt

Table 9-1 CAN-related Interrupts and Occurrence Sources (1/2)

Interrupts	Occurrence Sources	Details	Interrupt Enables		Request Clears	
			Register	Bit	Register	Bit
Global receive FIFO interrupt	Receive FIFO interrupt request	When the condition set in RFIGCV[2:0] bits of the RFCC register is met. For each message received	RFCCx	RFIE	RFSTSx	RFIF
	Receive FIFO full interrupt	When receive FIFO is full.	RFCCx	RFFIE	RFSTSx	RFFIF
Global error interrupt	DLC check error	When an error is detected in the DLC check.	GCTR	DEIE	GERFL	DEF
	FIFO message lost	When a message lost in the transmit/receive FIFO buffer is detected. When a message lost in the receive FIFO buffer is detected.	GCTR	MEIE	GERR	MES
	Transmit history buffer overflow	When attempting to store new transmission history data while the transmission history buffer is full	GCTR	THLEIE	GERFL	THLES
	CAN-FD message payload overflow	When payload overflow occurs	GCTR	CMPOFIE	GERR	THLMLT
	Transmit queue overwrite	When message overwrite is detected in the transmit queue	GCTR	QOWEIE	CERR	CMPOF
	Transmit queue message lost	When a message lost is detected in the transmit queue	GCTR	QMEIE	CERR	QOWES
	Transmit/Receive FIFO message overwrite	When message overwrite is detected in the transmit/receive FIFO in Gateway mode	GCTR	MOWEIE	CERR	QWES
Channels transmit interrupt	Channel m transmit completion interrupt request	When buffer is empty due to the completion of message transmission	TMIECy	TMIEp	TMSTSp	TMTRF
	Channel m transmit abort interrupt request	When buffer is empty due to the completion of a message transmission abort	CmCTR	TAIE	TMSTSp	TMTRF
	Channel m transmit queue interrupt request	When transmit queue is empty due to transmission completion Each message transmission is completed	TXQCC0~3m	TXQTXIE	TXQSTS0~3m	TXQFIF
	Channel m transmit history buffer interrupt request	When stored 3/4 data of transmit history buffer depth. Each 1 message transmits history stores completion.	THLCCm	THLIE	THLSTSm	THLIF
	Channel m transmit/receive FIFO transmit completion interrupt request	Transmits/Receives FIFO transmit completion interrupt request occurs when buffer is empty due to message transmission completion Transmits/Receives FIFO receives completion interrupts request is occurred for each message transmission completed	CFCCk	CFTXIE	CFSTSk	CFTXIF
	Channel m transmit/receive FIFO transmit completion interrupt request	When one frame of message is transmitted from the transmit/receive FIFO	CFCCEk	CFOFTXIE	CFSTSk	CFOFTXIF
	Channel m transmit queue one-frame transmit interrupt	When one frame of message is transmitted from transmit queue.	TXQCC0~3m	TXQOFTXIE	TXQSTS0~3m	TXQOFTXIF

Table 9-2 CAN-related Interrupts and Occurrence Sources (2/2)

Interrupts	Occurrence Sources	Details	Interrupt Enables		Request Clears	
			Register	Bit	Register	Bit
Channel transmit/receive FIFO receive interrupt	Channel m transmit/receive FIFO receive completion interrupt request	When reached the setting condition in CFIGCV bit of CFCCk register each completion of 1 message receives.	CFCCk	CFRXIE	CFSTSk	CFRXIF
	Channel m transmit/receive FIFO one-frame receive completion interrupt request	When Transmits/Receives FIFO 1 frame-received the message.	CFCCEk	CFOFRXIE	CFSTSk	CFOFRXIF
	Channel m transmit/receive FIFO buffer interrupt request	When transmits/receive FIFO is full.	CFCCEk	CFFIE	CFSTSk	CFFIF
	Channel m transmit queue one-frame routing interrupt request	When received 1 frame of message with TXQ selected as routing destination in GW mode	TXQCC 0~2m[*2]	TXQOFRXIE	TXQSTS 0~2m	TXQOFRXIF
	Channel m transmit queue full interrupt request	When TXQ of routing destination is full in GW mode.	TXQCC 0~2m	TXQFIE	TXQSTS 0~2m	TXQFIF
Channel Error Interrupt[*3]	Bus error	When any one of ADERR, B0ERR, B1ERR CERR, AERR, FERR and SERR flag of CmERFA register is "1".	CmCTR	BEIE	CmERFL	BEF
	Error warning	When CnERFL of REC [7:0] or TEC [70] bit value of CnEREL register exceed “95”.	CmCTR	EWIE	CmERFL	EWF
	Error passive	When becomes error passive status (REC[7:0] or TEC[7:0]bit > 127)	CmCTR	EPIE	CmERFL	EPF
	Bus off enrty	When becomes bus off status (TEC[7:0] bits > 255)	CmCTR	BOEIE	CmERFL	BOEF
	Bus off recovery	When recovered from bus off status to detect 11bit continuous recessives 128 times.[*4]	CmCTR	BORIE	CmERFL	BORF
	Overload frame transmit	When detected transmits condition of overload frame if performs the transmits or receives.	CmCTR	OLIE	CmERFL	OVLF
	Bus lock	When detected 31bit continuous dominants on CAN bus in channel communication mode.	CmCTR	BLIE	CmERFL	BLF
	Arbitration lost	When detects arbitration lost.	CmCTR	ALIE	CmERFL	ALF
	Error Counter overflow	When the error occurrence counter has overflowed	CmCTR	EOCOIE	CmFDSTS	EOCO
	Successful Occurrence Counter Over Flow	When successful occurrence counter has overflowed	CmCTR	SOCOIE	CmFDSTS	SOCO
	Transmitter Delay Compensation Vioration	When a transmitter delay compensation violation is present (TDCVF = 1)	CmCTR	TDCVFIE	CmFDSTS	TDCVF

[*2] TXQ3 is unelectable as the routing destination.

[*3] An interrupt occurs when any one of the following is detected:

- ADERR flag of CmERFL register is “1”, and a form error in ACK delimiter is detected.L
- B0ERR flag of CmERFL register is “1”, and a recessive is detected even though a dominant is transmitted.
- B1DRR flag of CmERFL register is “1”, and a dominant is detected even though a recessive is transmitted .
- CERR flag of CmERFL register is “1”, and a CRC error is detected.L
- AERR flag of CmERFL register is “1”, and an ACK error is detected.
- FERR flag of CmERFL register is “1”, and a form error is detected.
- SERR flag of CmERFL register is “1”, and a stuff error is detected.

[*4] Before detecting the 11-bit consecutive recessives 128 times, the interrupt is not occurred when recovered from bus off state by the following method. (BORF flag is not “1”.)

- When set “01_B” (channel reset mode) to CHMDC[1:0] bits of CmCTR register.L
- When set “1” (forced -recovery from bus off) to RTBO bit of CmCTR register.
- When set “01_B” (transition to channel halt mode by bus off entry) to BOM[1:0] bits of CmCTR register.
- When BOM[1:0] bit is “11_B”, and set “10_B” to CHMDC[1:0] bit before detecting the 11-bit consecutive recessives 128 times.

9.1 CAN-related Interrupt Setting Procedure

Figure 9-1 shows the interrupt setting procedure.

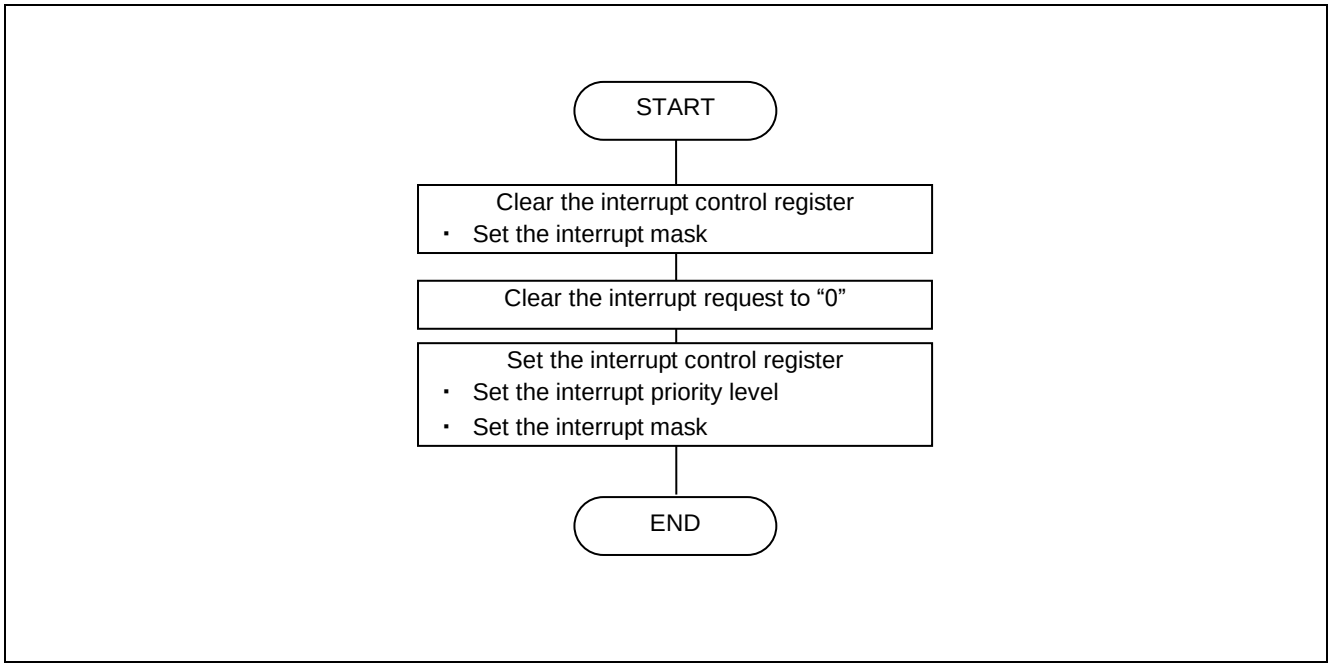


Figure 9-1 Interrupt Setting Procedure

10. DMA Trigger

The receive FIFO buffer, transmit/receive FIFO buffer, or transmit queue can be associated with the DMA channel.

The receive FIFO buffer, transmit/receive FIFO buffer, and transmit queue that have the DMA trigger function are shown below.

- Receive FIFO buffer (x=0~7)
- Transmit/receive FIFO buffer (Transmit/receive FIFO buffer (m+1)*3 - 3 of each channel)
- Transmit queue (TXQ0, TXQ3 of each channel)
- Transmit/receive FIFO buffer (Transmit/receive FIFO buffer (m+1)*3 - 1 of each channel)

Refer to “Figure 23.31 Message Buffer connectable to a DMA channel” for the receive FIFO buffer, transmit/receive FIFO buffer, and transmit queue that have the DMA trigger function.

10.1 DMA Transfers

10.1.1 DMA Transfers on Reception

The buffer and queues that can request transfers to the DMA channel on reception are shown below.

- Receive FIFO buffer (x=0~7)
- Transmit/receive FIFO buffer (Transmit/receive FIFO buffer (m + 1) * 3 - 3 of each channel)

A DMA transfer request trigger is generated when there is an unread message in the FIFO buffer with the DMA transfer enable bit (CDTCT.RFDMAEx or CDTCT.CFDMAEm) set. Specify the address of the FIFO access register[*1] as the transfer source address, and adjust the transfer size so that the end of the payload storage area is read in one trigger. This end depends on the payload storage size set by RFPLS[2:0] bits in the RFCCx register or CFPLS[2:0] bits in the CFCCk register.

10.1.2 DMA Transfers on Transmission

The buffer and queues that can request transfers to the DMA channel on transmission are shown below.

- Transmit queue (TXQ0, TXQ3 of each cahnnel)
- Transmit/receive FIFO buffer (Transmit/receive FIFO buffer (m+1) * 3 - 1 of each channel)

When the DMA enable bit (CDTTCT.TQ0DMAEm or CDTTCT.TQ3DMAEm or CDTTCT.CFDMAEm) is set, emssages in the corresponding transmit queue or transmit/receive FIFO are processed by the DMA controller. To process a transmit queue or transmit/receive FIFO buffer with the DMA controller, follow the procedure below.

1. Check the transmit queue or transmit/receive FIFO buffer is empty.
2. When data to be transmitted is available for transmission, the CPU enables DMA to store the data to be transmitted in the transmit queue or transmit/receive FIFO buffer. Then, the DMA transfer request trigger is generated.[*2]

[*1] RFID, RFPTR, RFFDSTS, and RFDFd register on using the receive FIFO buffer.

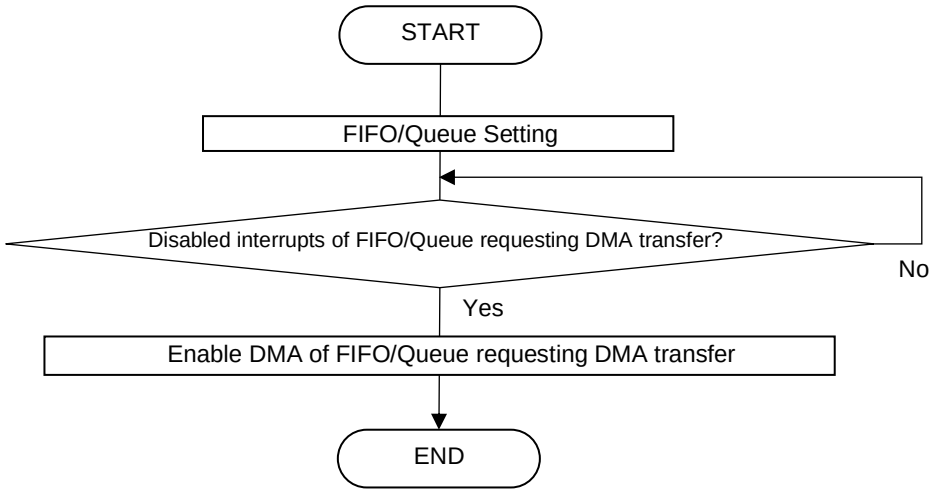
CFID, CFPTR, CFFDCSTS, and CFDFd register on using the transmit/receive FIFO buffer.

[*2] TMID, TMPTR, TMFDCTR, and TMDFd register on using the transmit queue.
CFID, CFPTR, CFFDCSTS, and CFDFd register register on using the transmit/receive FIFO buffer.

3. For transmit/receive FIFO buffers, the transmit/receive FIFO pointer is automatically incremented when all data of payload length (set by the CFPLS bit of the CFCCk register) is written during a DMA transfer.
   For transmit queues, the pointer is automatically incremented when data with a payload length of 64 bytes is written. If the payload data is less than 64 bytes, dummy data needs to be written and set the data payload size to 64 bytes.
   Only 32-bit write accesses are possible during DMA transfers.

10.1.3 DMA Function Setting Procedures

Figure 10-1 shows the setting procedure of the DMA function.



1. Do not write to the FIFO control register (RFCCx register or CFCCk register) when the DMA transfer request is enabled.
2. If the RFDMAEx or CFDMAEm bit is set to "0" (DMA transfer request disabled) during DMA transfer, wait until the DMA transfer status (RFDMASTSx or CFDMASTSm bit in the CDTSTS register) becomes "0" (not in DMA transfer) before moving to the next process (such as allowing DMA transfer again).
3. If DMA transfer is disabled, consider how to handle messages remaining in the FIFO buffer and newly arriving messages. If the FIFO buffer is enabled, the FIFO buffer will continue to receive.

Figure 10-1 DMA Function Setting Procedure

11. Transmitter Delay Compensation (Only CAN FD Mode)

A high baud rate is used in the data phase of CAN FD mode. Transmitter delay compensation is provided as a function to accept propagation delay in this case.

To use this function, set the TDCE bit in the CmFDCFG register to “1”. Also set the secondary sample point (SSP) timing used in the data phase by the TDCOC bit and TDCO[7:0] bits in the CmFDCFG register.

When the TDCOC bit is “0”, the SSP timing equals the total value of the delay measured by the RS-CANFD module and the TDCO[7:0] value. (This value is rounded off to the nearest integer of Tq.) Usually, the TDCO[7:0] value must be equal to SS + TSEG1, the sample point timing.

Figure 11-1 shows the SSP timing.

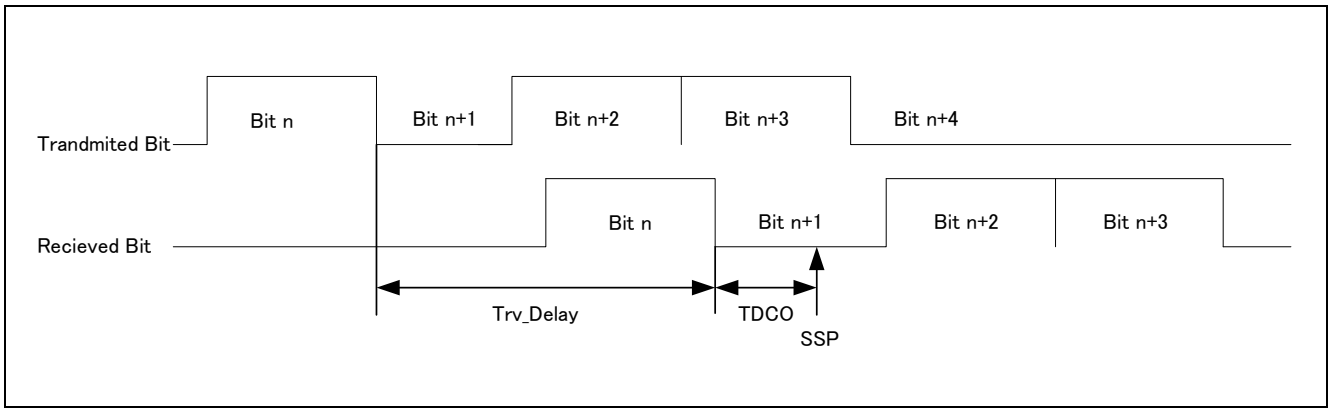


Figure 11-1 SSP Timing

When the TDCOC bit is “1”, the SSP timing is determined only by the TDCO[7:0] value. (When the DBRP[7:0] value in the CmDCFG register is larger than “0”, the TDCO[7:0] value is also rounded off to the nearest integer of Tq.) The SSP offset value becomes the set values of the TDCO[7:0] bit +1.

The RS-CANFD module compensates a delay up to 6 CANm bit time (2 fCAN). (CANm bit time is the value of data bit rate.)

The TDCR[7:0] flag in the CmFDSTS register is the bit that indicates the transmitter delay compensation result as a multiple of CAN clock frequency (fCAN). This result depends on the settings of the TDCOC bit and TDCO[7:0] bits in the CmFDCFG register. These flags are updated at the timing of the falling edge between the FDF bit and res bits when the TDCE bit is set to “1” (transmitter delay compensation enabled) and the TDCOC bit is set to “0” (measurement and offset) in the CmFDCFG register.

The TDCVF bit in the CmFDSTS register indicates violation of transmitter delay compensation. The transmit data is compared to the receive CAN bus level delayed by the transceiver loop delay. This delay varies due to physical factors such as temperature, therefore the tempolary maximum delay cannot be confirmed. This bit is set to “1” when the transmitter delay compensation exceeds the maximum compensation 6 CANm bit times (2 fCAN). (CANm bit time is the value of data bit rate).

12. Precautions for Processing Flow

12.1 Functions

In this application note, there are cases where even a single line of processing is described as a function, but this is only to clarify the processing for each function. When actually creating a program, it is not always necessary to make it a function. This is just to clarify the processing for each function.

12.2 Setting for Each Channel

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

12.3 Infinite Loop

In order to simplify the explanation, some parts of the processing flow are shown as an infinite loop. When actually creating the program, please make sure that each loop has a time limit and that it exits when the loop overtimes. **Figure 12-1** shows an example of processing when the loop time limit is set. Table 12-1 and Table 12-2 shows the maximum transition time to each mode.

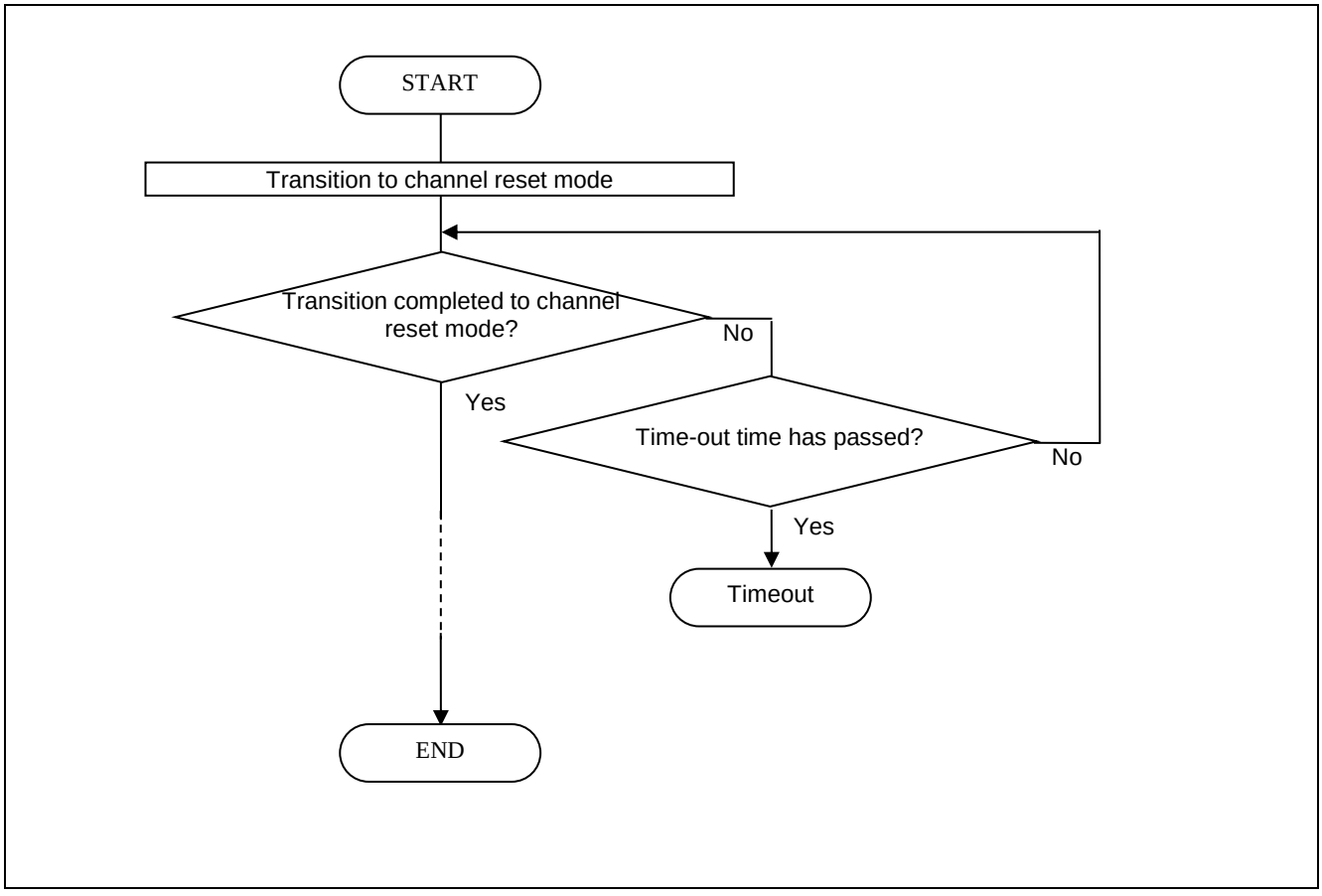


Figure 12-1 Example of Processing with Loop Time Limit

Table 12-1 Transition Time in Global Mode

Mode before transition	Mode after transition	Maximum transition time
Global stop	Global reset	3 clocks of pclk
Global reset	Global stop	3 clocks of pclk
Global reset	Global test	10 clocks of pclk
Global reset	Global operation	10 clocks of pclk
Global test	Global reset	2 CAN bit time*1, 2
Global test	Global operation	3 clocks of pclk
Global operation	Global reset	2 CAN bit time*1, 2
Global operation	Global test	3 CAN frames*1

Note 1. It is the CAN bit time and CAN frame time of the slowest communication speed of the channels used.

Note 2. In CAN FD mode, this is the nominal bit rate of CANm bit time.

Table 12-2 Transition Time in Channel Mode

Mode before transition	Mode after transition	Maximum transition time
Channel Stop	Channel reset	3 clocks of pclk
Channel reset	Channel Stop	3 clocks of pclk
Channel reset	Channel halt	3 CANm bit time*1
Channel reset	Channel communication	4 CANm bit time*1
Channel halt	Channel reset	2 CANm bit time*1
Channel halt	Channel communication	4 CANm bit time*1
Channel communication	Channel reset	3 CANm bit time*1
Channel communication	Channel halt	2 CANm frames

Note1. In CAN FD mode, this is the nominal bit rate of CANm bit time.

13. Appendix

13.1 CAN Configuration Processing to be performed in Each State

Table 13-1 shows CAN configuration processing to be performed in each state.

Table 13-1 CAN Configuration Processing to be performed in Each State

Processing		CAN Configuration[*1]			
		After MCU reset	After global reset mode	After channel reset mode	After channel halt mode
CAN state (mode) transition	Global mode transition	○	○	—	—
	Channel mode transition	○	○	○	○
Global function setting	Transmit priority setting	○	△	—	—
	DLC check setting				
	DLC replacemt function setting				
	Mirror function setting				
	Clock setting				
	Timestamp clock setting				
	Interval timer prescaler setting				
Communication speed setting	Bit timing setting	○	△	△	△
	Communication speed setting				
Receive rule table setting			△	—	—
Buffer setting	Receive buffer setting	○	△	—	—
	Receive FIFO buffer setting				
	Transmit/Receive FIFO buffer setting			△[*2]	△[*2]
	Transmit buffer setting			△	△
	Transmit queue setting				
	Transmit history buffer setting				
Global error interrupt setting			△	—	—
Channel function setting			△	△	△

Note 1. ○ : Setting required, — : Cannot be set, △ : No setting required

Note 2. Rewrite the following bits in global reset mode.

CFTML[4:0] bits, CFM[1:0] bits, CFIGCV[2:0] bits, CFIM bit, CFE bit in CFCCk register.

13.1.1 Software Explanation

Module Explanation

The following is a list of modules for an example configuration setting in the case of a sample program for sending a 64-byte message.

Table 13-2 Module List

Module Name	Label Name	Function
Main routine	main_pm0	Performs various setting and appreciation starts.
CAN initialize setting routine	R_CAN_Init	Performs CAN controller initial setting.
GSTS register access processing	RSCFDnCFDGSTS	Performs GSTS register reading.
GCFG register access processing	RSCFDnCFDGCFG	Performs GCFG register reading/writing.
CmNCFG register access processing	RSCFDnCFDCmNCFG	Performs CmNCFG register reading/writing.
CmDCFG register access processing	RSCFDnCFDCmDCFG	Performs CmDCFG register reading/writing.
GAFLCFGv register access processing	RSCFDnCFDGAFLCFGv	Performs GAFLCFGv register reading/writing.
GAFLECTR register access processing	RSCFDnCFDGAFLECTR	Performs GAFLECTR register reading/writing.
GAFLIDj register access processing	RSCFDnCFDGAFLIDj	Performs GAFLIDj register reading/writing.
RMNB register access processing	RSCFDnCFDRMNB	Performs RMNB register reading/writing.
RFCCx register access processing	RSCFDnCFDRFCCx	Performs RFCCx register reading/writing.
CFCCk register access processing	RSCFDnCFDCFCCk	Performs CFCCk register reading/writing.
TMIECy register access processing	RSCFDnCFDTMIECy	Performs CFCCk register reading/writing.
TXQCC0m register access processing	RSCFDnCFDTXQCC0m	Performs TXQCC0m register reading/writing.

Register Setting

The following shows the register settings for each function in the case of the 64-byte message transmission sample program.

Note that since no message reception is performed in this program, the register settings related to reception are not performed.

Table 13-3 CAN FD Register Setting (Part 1)

Register Name	Setting Value	Function
GCFG	0x00001006	・Unused interval timer presccaler ・Set channel 0 bit time clock to time stamp clock source selection. ・Set bit time clock to time stamp source selection. ・No time stamp source division. ・Reject the message in message payload overflow. ・Set internal clock (clkc[80MHz]) to CAN clock source selection. ・Mirror mode disables ・DLC exchanges enables ・DLC check enables ・Set ID priority to transmit priority selection.
CmNCFG (m=0)	0x061C0C03	Set 1 Mbps to communication speed. ・NBRP : 3 (4BRP) ・NTSEG1: 14 (15TQ) ・NTSEG2: 3 (4TQ) ・NSJW : 3 (4TQ)
CmNCFG (m=1 to 7)	0x00000000	Not set
CmDCFG (m=0)	0x03030E00	Set 4 Mbps to communication speed. ・DBRP : 0 (1BRP) ・DTSEG1: 14 (15TQ) ・DTSEG2: 3 (4TQ) ・DSJW : 3 (4TQ)
CmDCFG (m=1 to 7)	0x00000000	Not set
CmFDCFG (m=0 to 7)	0x00000000	Not set
GAFLCFGv (v=0 to 3)	0x00000000	Not set
GAFLECTR	0x00000000	Not set
GAFLIDj (j=1 to 16)	0x00000000	Not set
GAFLMj (j=1 to 16)	0x00000000	Not set
GAFLP0j (j=1 to 16)	0x00000000	Not set
GAFLP1j (j=1 to 16)	0x00000000	Not set
RMNB	0x00000000	Not set
RFCCx (x=0 to 7)	0x00000000	Not set

Table 13-4 CANFD register setting (Part 2)

Register Name	Setting Value	Function
CFCCk (k=0 to 23)	0x00000000	Not set
TMIECy (y=0)	0x00000001	Transmit interrupt enables
TMIECy (y=1 to 15)	0x00000000	Not set
TXQCm (m=0 to 3)	0x00000000	Not set
GCTR	0x00000007	・No time stamp counter reset ・GW FIFO message overwrite interrupt disables ・TXQ message lost interrupt disables ・TXQ message overwrite interrupt disables ・Payload overflow interrupt disables ・Transmit history buffer overflow interrupt disables ・FIFO message lost interrupt disables ・DLC error interrupt disables ・Global sleep request enables ・Keep the present value of global mode controls.
CmCTR (m=0)	0x00000001	・Restricted Operation mode disabled. ・No leading bit inversion in the received ID field of CRC error test. ・Set standard test mode to communication test mode. ・Communication test mode prohibited ・Set the error flag display of first issued error information only after cleared the bit 14 to 8 of CmERFL register in error display mode selection. ・Set ISO11898-1 compliant to bus off recovery mode selection. ・Transmitter delay compensation violation interrupt disabled ・Successful Occurrence Counter Overflow interrupt disabled ・Error occurrence Counter Overflow interrupt disabled ・Transmit abort interrupt disabled ・Arbitration lost interrupt disabled ・Bus lock interrupt disabled ・Overload flame transmit interrupt disabled ・Bus off recovery interrupt disabled ・Bus off entry interrupt disabled ・Error passive interrupt disabled ・Error warning interrupt disabled ・Bus error interrupt disabled ・No bus off forced-recovery ・Not channel stop mode ・Channel reset mode
CmCTR (m=1 to 7)	0x00000005	Not set

Revision History

Rev.	Date	Description	
		Page	Summary
1.11	2023.01.23	-	Released English version of R01AN4891JJ0111.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

   Handle unused pins in accordance 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 the 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.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.

6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given 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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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 system; undersea repeaters; 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.