SH7216 Group

R01AN0562EJ0110
Rev.1.10
May 18, 2011

Protocol Conversion Between Ethernet and RCAN Sample Program

Introduction

This application note presents a sample program that sends and receives text data between Ethernet and RCAN networks using a combination of the Ethernet related modules (EtherC and E-DMAC) and the controller area network module (RCAN-ET) included in the SH7216.

Target Device

SH7216

Contents

1. Introduction

1.1 Specifications

This sample program transfers text data between PCs using the Ethernet related modules (EtherC and E-DMAC) and the controller area network (RCAN-ET) included in the SH7216.

It uses two communication terminals (host PC A and host PC B) that provide Telnet applications to perform the following operations.

1. Ethernet to RCAN text data transfer
   Text data input to the Telnet application is converted from the TCP/IP protocol to RCAN communication data by Ethernet/RCAN conversion software on the SH7216 and transmitted over RCAN.
2. RCAN to Ethernet text data transfer
   Text data received over RCAN is converted from RCAN communication data to TCP/IP protocol data by Ethernet/RCAN conversion software on the SH7216 and output by the Telnet application.

Figure 1 shows the system structure for this application note.

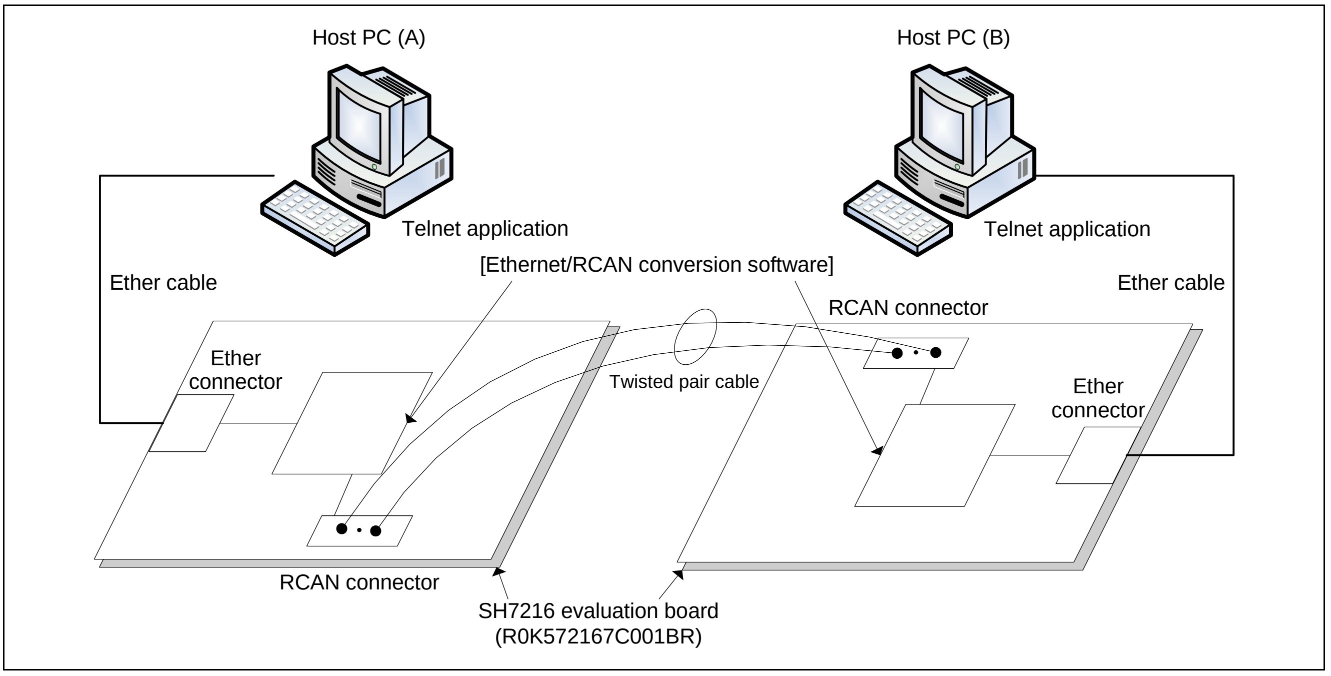

Figure 1 System Structure

1.2 Functions Used

- Interrupt controller (INTC)
- Ethernet controller (EtherC)
- Ethernet controller direct memory access controller (E-DMAC)
- Compare match timer (CMT)
- Control area network (RCAN-ET)

1.3 Applicable Conditions

Microcontroller:	SH7216
Operating frequencies:	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz
Integrated development environment:	Renesas Electronics High-performance Embedded Workshop, Ver. 4.08.00.011
C compiler:	Renesas Electronics SuperH RISC engine Family C/C++ Compiler Package, Ver. 9.03, Release 02
Compiler options:	Other than the options specified in the include file in the High-Performance Embedded Workshop, the default options are used. (-cpu=sh2afpu -pic=1 include="$(WORKSPDIR)\C_Source\common","$(WORKSPDIR)\ C_Source\ether","$(WORKSPDIR)\C_Source\rcan","$(WORKSPDIR)\C_Source\uip\apps\telnetd","$(WORKSPDIR)\C_Source\uip\lib","$(WORKSPDIR)\C_Source\uip\uip","$(WORKSPDIR)\C_Source\uip\unix","$(WORKSPDIR)\C_Source\user-app" -object="$(CONFIGDIR)\$(FILELEAF).obj" -gbr=auto –chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)

1.4 Related Application Notes

- H8S/2472 and SH7216 uIP TCP/IP Protocol Stack Demonstration (Rev. 2.00)
- SH7216 Group RCAN-ET User Program Mode Flash Programming Example
- SH7216 Group Ethernet User Program Mode Flash Programming Example Application Note
- SH7216 Group Ethernet/USB Protocol Conversion Operation Example

2. Operating Environment

This section describes the operating environment necessary to run this SH7216 sample program.

2.1 Hardware Environment

The following hardware environment is required to use this sample program.

- Two SH7216 CPU boards
- One RCAN connection twisted pair cable
- Two Ethernet cables (cross cables)
- Two Telnet terminal host PCs (OS: Windows XP or Windows Vista)

2.2 Evaluation Board Environment Settings

1. SH7216 sample program execution mode settings
   Set the SH7216 CPU boards to single chip mode with the SW5 mode switches before applying power. (See table 1.)

Table 1 SW5 Mode Switch Settings

Switch	Single Chip Mode	Function
SW5-1	On	Internal flash memory write/erase protect
SW5-2	Off	MD1 pin state
SW5-3	Off	MD0 pin state
SW5-4	On	Ethernet functions enabled

2.3 PC Environment Settings

This section describes the environment settings for host PC A and host PC B used as Telnet communication terminals.

2.3.1 Telnet Communication Terminal Host PC A

1. Install a general-purpose communication program such as TeraTerm or Hyper Terminal.
2. Set up the IP address and subnet as shown below.
   IP address: 192.168.1.77
   Subnet mask: 255.255.255.0
3. Set SW6-1 in the user DIP switch (SW6) to the ON position.

2.3.2 Telnet Communication Terminal Host PC B

1. Install a general-purpose communication program such as TeraTerm or Hyper Terminal.
2. Set up the IP address and subnet as shown below.
   IP address: 192.168.1.78
   Subnet mask: 255.255.255.0
3. Set SW6-1 in the user DIP switch (SW6) to the OFF position.

Note: If the IP address used for host PC A or host is changed, the defined data values in the sample program must also be changed to values that allow communication with the change IP addresses. (See figure 2 in section 4.2.)

3. Sample Program Operational Overview

This section describes the use of the sample program.

Before performing the following operations, verify that the SH7216 CPU board switches are set to single chip mode. (See section 1.)

3.1 Usage Procedure

Perform the following connections and turn on the SH7216 CPU board power.

- Connect the E10A to SH7216 CPU board A and E10A to SH7216 CPU board B, respectively.
- Connect RCAN ports of the SH7216 CPU board A and B with the twisted pair cable.
- Connect Telnet communication terminal host PC A to SH7216 CPU board A and Telnet communication terminal host PC B to SH7216 CPU board B with Ethernet cables, respectively.
- Set SW6-1 in the SH7216 CPU board user DIP switch (SW6) on the two SH7216 CPU boards as follows.
  SH7216 CPU board A: ON
  SH7216 CPU board B: OFF
- Turn on the power and start up the SH7216 CPU boards in the order board B and then board A.

(1) Start up the general-purpose communication software on host PC A and host PC B and set the IP addresses as follows. Telnet is used as the application.
Host PC A: 192.168.1.75
Host PC B: 192.168.1.76

(2) Set the general-purpose communication software to CR for reception and CR+LF for transmission, and enable the local echo function.

(3) If text entered on the host PC A Telnet terminal is echoed back on the host PC B Telnet terminal, then the text data transfers from Ethernet to RCAN, from RCAN to RCAN, and from RCAN to Ethernet are operating correctly. Similarly, text data entered at the host PC B Telnet terminal can be verified on the host PC A Telnet terminal.

For text files up to 256 bytes, data can be transferred by specifying the file name in the file selection dialog. A user application must handle the control to send files larger than 256 bytes.

Note: While operation for files up to 256 bytes has been verified, this operation is not guaranteed.

3.2 Disconnecting and Reconnecting Cables

3.2.1 Disconnecting and Reconnecting Ethernet Cables

If an Ethernet cable is disconnected, the Telnet communication terminal on the host PC will exit automatically. Text data entered at the Telnet terminal on the other host PC will be discarded in the program.

If the Ethernet cable is reconnected, perform the following operations on the host PC where the cable was reconnected.

1. Restart the general-purpose communication software on the host PC that was reconnected and perform the operations in steps (1) and (2) in section 3.1. Then open the Telnet terminal.

After the above operations have been performed, text data transfer operations between host PC A and host PC B will be possible again.

4. Sample Program Specifications

This section presents the specifications of the SH7216 sample program used in this application note.

4.1 Functions

1. Ethernet to RCAN text data transfer
   Ethernet reception
   — Ethernet data is received from the host PC using notification by interrupt.
   — Data is copied to an RCAN transmit buffer.

2. RCAN to RCAN text data transfer
   RCAN transmission
   — The RCAN transmit buffer is polled to determine whether or not there is RCAN transmit data.
   — If there is data present in the RCAN transmit buffer, data is transmitted using a transmit mailbox.
   RCAN reception
   — Data is received in a receive mailbox using notification by interrupt.
   — The data is stored in an Ethernet transmit buffer.

3. RCAN to Ethernet text data transfer
   Ethernet transmission
   — The Ethernet transmit buffer is polled to determine whether or not there is Ethernet transmit data.
   — If there is data present in the Ethernet transmit buffer, data is transmitted to the host PC using TCP/IP communication.

4. Ethernet cable disconnection and reconnection
   Disconnection and reconnection
   — Disconnection is detected by interrupt and changes in the Ethernet link signal (link up/down) are reflected in the link signal change flag.
   — The link signal change flag is polled and whether or not there was an Ethernet link signal change, and whether the state is link up or link down, is verified.
   Link down state
   — The uIP timer is stopped and the Ethernet device disabled.
   — After the Ethernet device has been disabled, all RCAN receive data is discarded.
   Link up state
   — The uIP timer, Ethernet driver, and uIP (TCP/IP protocol stack) are initialized.
   — After Telnet connection, the Ethernet device is enabled.
   — After the Ethernet has been enabled, Ethernet/RCAN data transfers are enabled.

4.2 SH7216 Sample Program

This sample SH7216 program consist of the main process, a TCP/IP protocol stack, an Ethernet driver, a timer driver, and an RCAN driver.

This sample program uses 192.168.1.75 and 192.168.1.76 as fixed Telnet connection IP addresses.

Note: If these IP addresses is changed, the defined data (ipAddrData1 and ipAddrData2) values in the sample program (shown in figure 2) must also be changed.

```
(main.c)
        …
/*  network data for Ethernet        */
static struct uip_eth_addr           mac_addr ;
/*       target No1         */
static struct uip_eth_addr           mac_addr1 = {0x00,0x01,0x02,0x03,0x04,0x05} ;
static uchar8_t                      ipAddrData1[4] = {192,168,1,75} ;
/*       target No2         */
static struct uip_eth_addr           mac_addr2 = {0x00,0x01,0x02,0x03,0x04,0x06} ;
static uchar8_t                      ipAddrData2[4] = {192,168,1,76} ;

static uchar8_t                      netMaskData[4] = {255,255,255,0} ;
…
```

Figure 2 IP Address Definitions

Figure 3 shows the structure of this SH7216 sample program. The arrows in the figure show the direction of control.

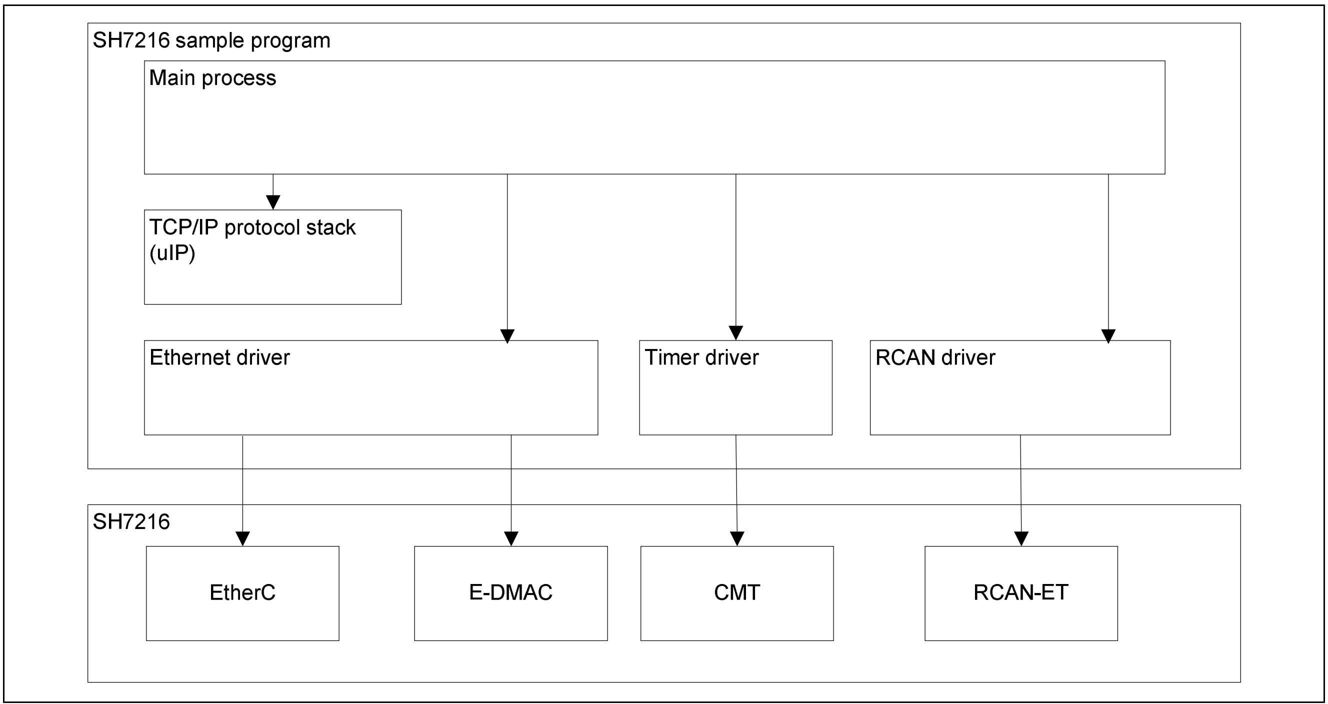

Figure 3 SH7216 Sample Program Structure

4.2.1 Main Process

The main process uses polling to perform the following processing.

- Checking the RCAN connection
- Checking to detect changes in the link signal to insertion/removal of the Ethernet cable.
- Checking for Ethernet frame reception
- Checking for uIP timer timeouts
- Checking for Ethernet data reception

Table 2 lists the flags used for poling control, data send/receive control, and device detection control.

Table 2 Control Flags

No.	Flag	Description
1	EtherDevEnableFlag	Indicates the presence/absence of an Ethernet device
2	EtherInEnableFlag	Controls Ethernet reception
3	EtherLinkChgFlag	Indicates whether or not the link signal changed and the link up/link down information when the link signal is present
4	RcanDevEnableFlag	Indicates the enabled/disabled state of the RCAN device
5	EtherOutEnableFlag	Controls the transfer of text data from RCAN to Ethernet
6	RcanOutEnableFlag	Controls the transfer of text data from Ethernet to RCAN
7	targetSw	Indicates the state of the user DIP switch (SW6-1)

The Telnet IP address and the RCAN send/receive mailboxes (MB0 to MB3) are controlled by the on/off state of the user DIP switch (SW6-1) as listed in table 3.

Table 3 User DIP Switch (SW6-1) On/Off Specifications

		SW6-1		
No.	Item	On	Off	Description
1	IP address	192.168.1.75	192.168.1.76	Telnet connection IP address
2	Mailbox 0 ID	0x100	0x101	Control command reception
3	Mailbox 1 ID	0x101	0x100	Control command transmission
4	Mailbox 2 ID	0x120	0x121	Data reception
5	Mailbox 3 ID	0x121	0x120	Data transmission

Note: This SH7216 sample program determines the on/off state of the user DIP switch and sets the Telnet connection UP address and RCAN send/receive mailbox IDs accordingly.

Table 4 lists the control commands used for RCAN transmission and reception.

Table 4 RCAN Transmission and Reception Control Commands

No.	Command	Value	Description
1	ACK	0x06	Data receive acknowledge command
2	NAK	0x15	Data receive negative acknowledge command
3	Start	0x12	RCAN start request command
4	Reply	0x14	RCAN start response command
5	X-ON	0x11	Software flow control (transmit start request)
6	X-OFF	0x13	Software flow control (transmit stop request)

Figures 4 to 8 show the flowcharts for the main process.

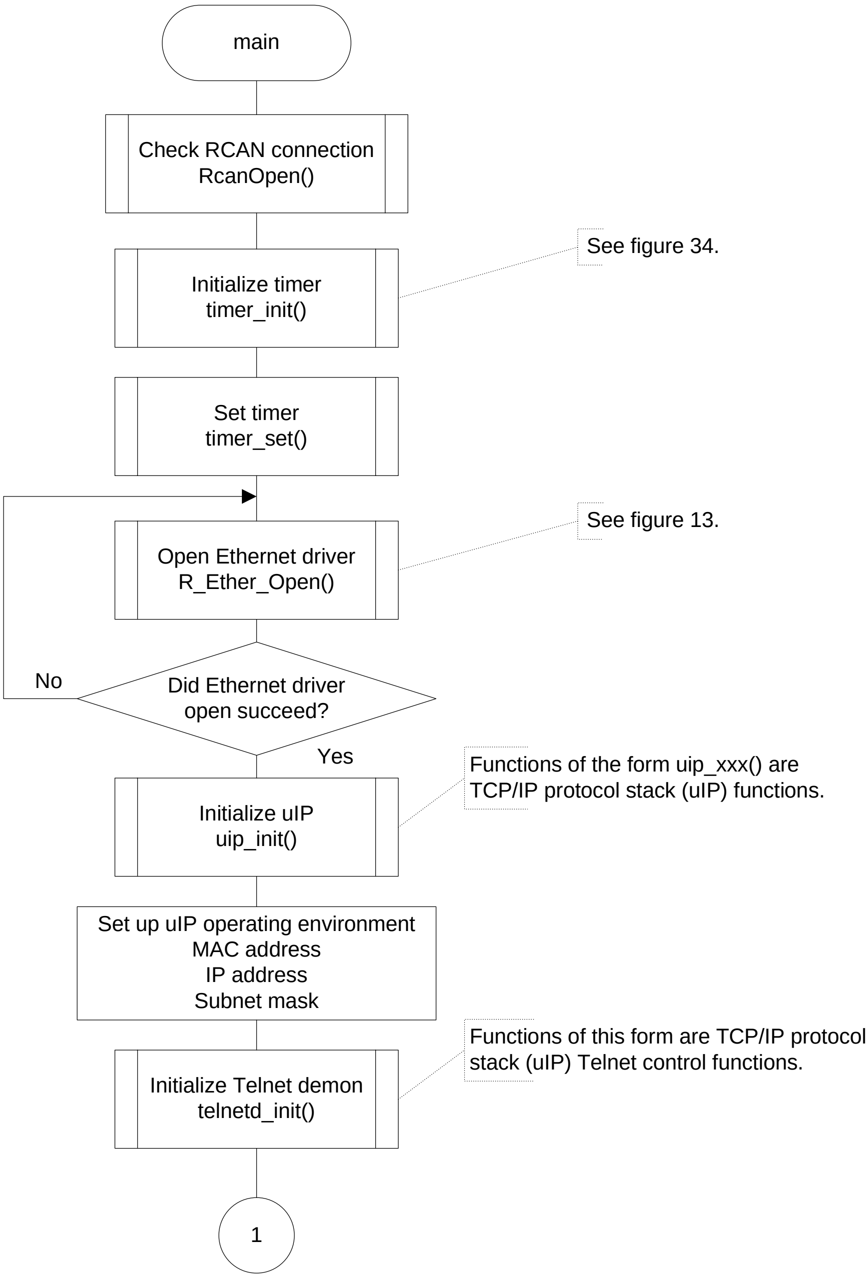

Figure 4 Main Process Flowcharts (1/5)

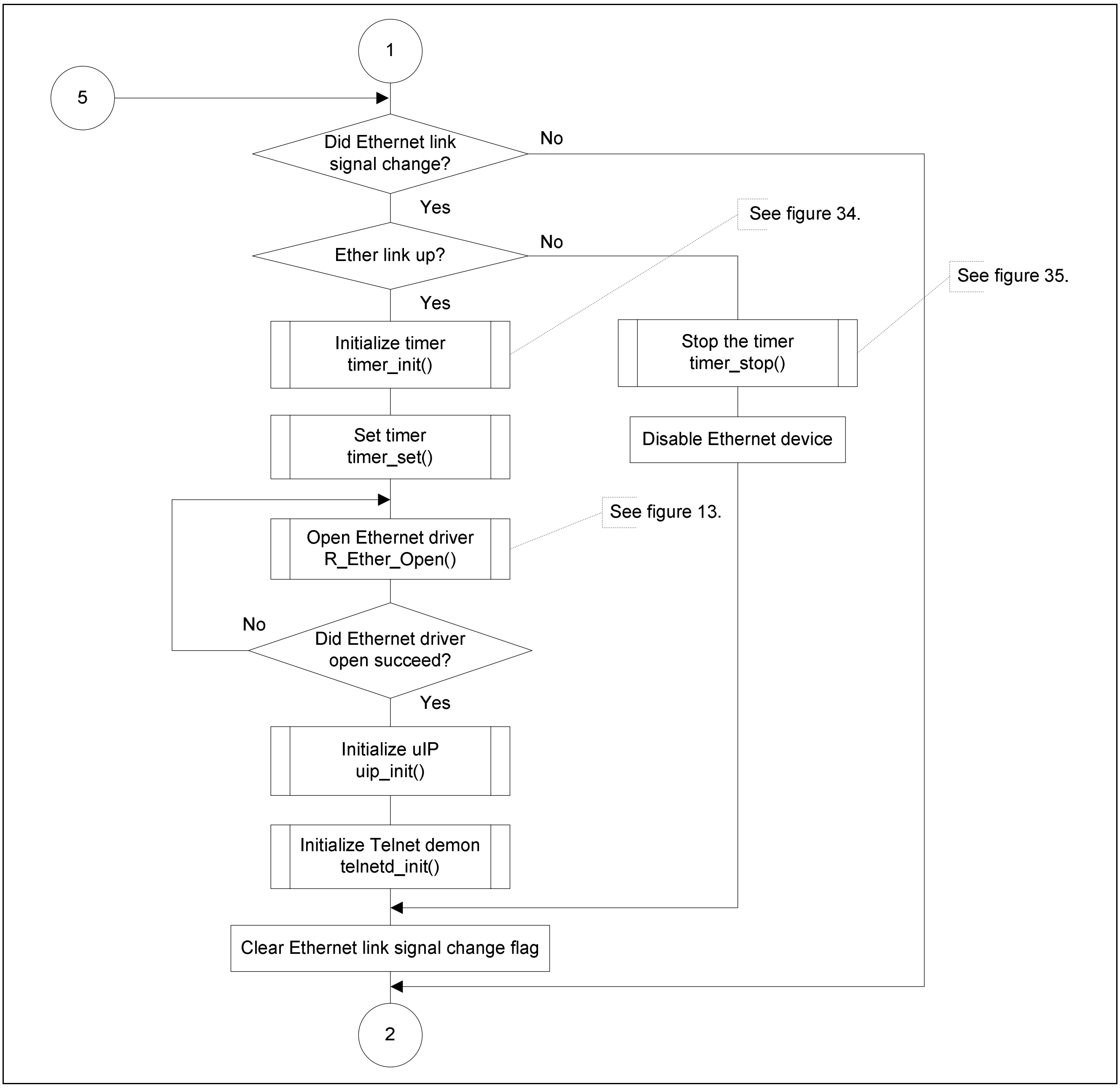

Figure 5 Main Process Flowcharts (2/5)

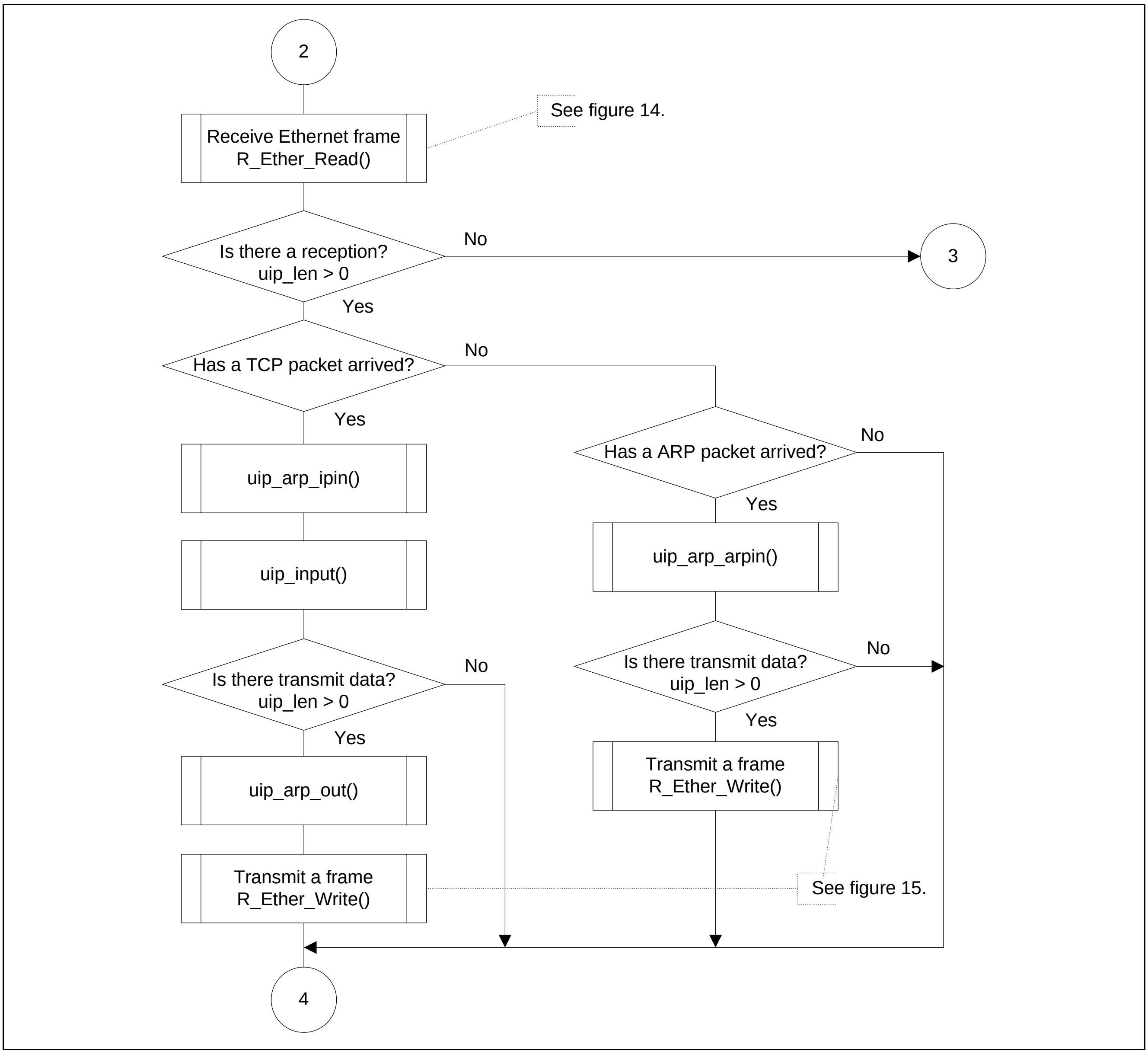

Figure 6 Main Process Flowcharts (3/5)

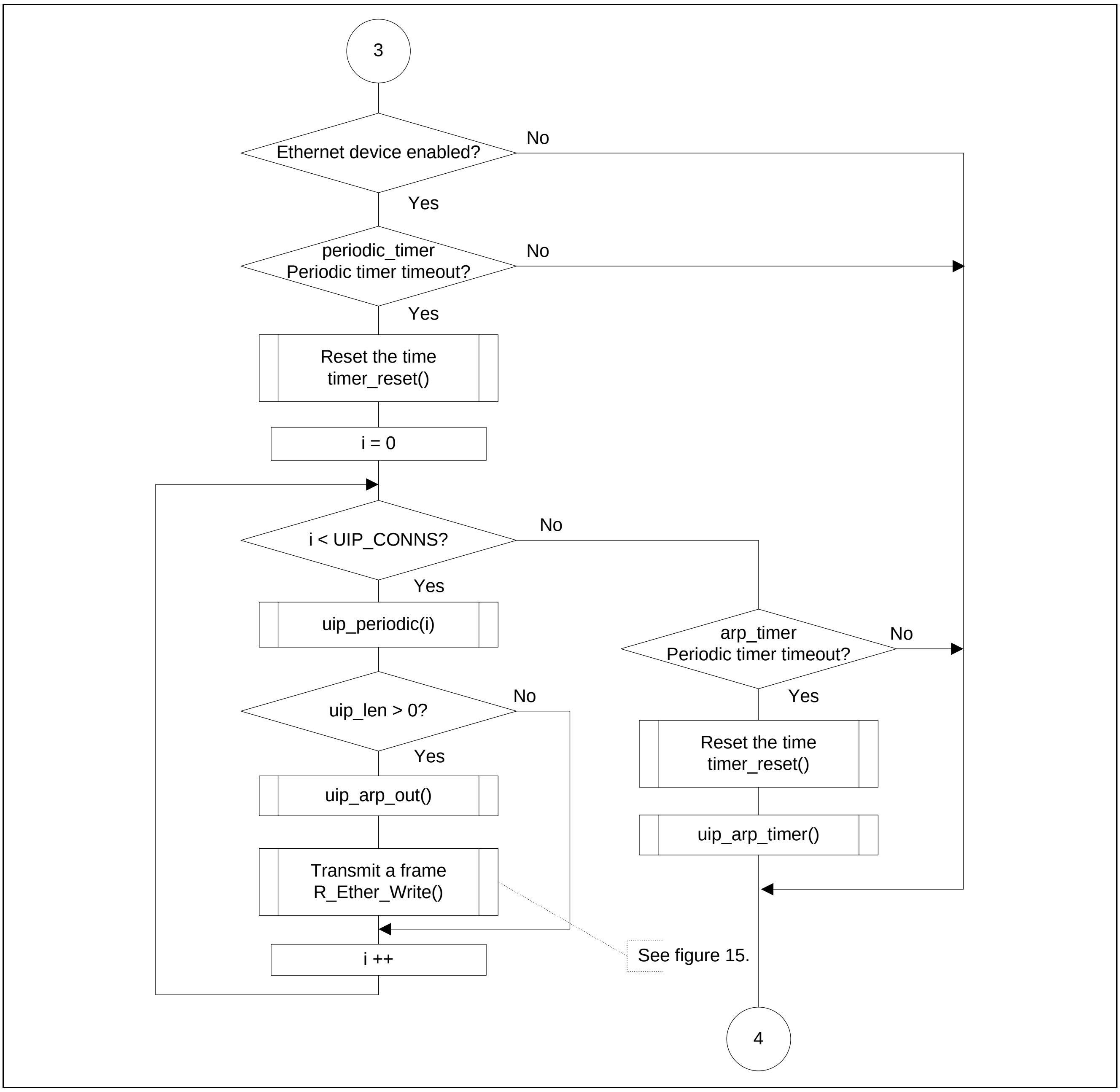

Figure 7 Main Process Flowcharts (4/5)

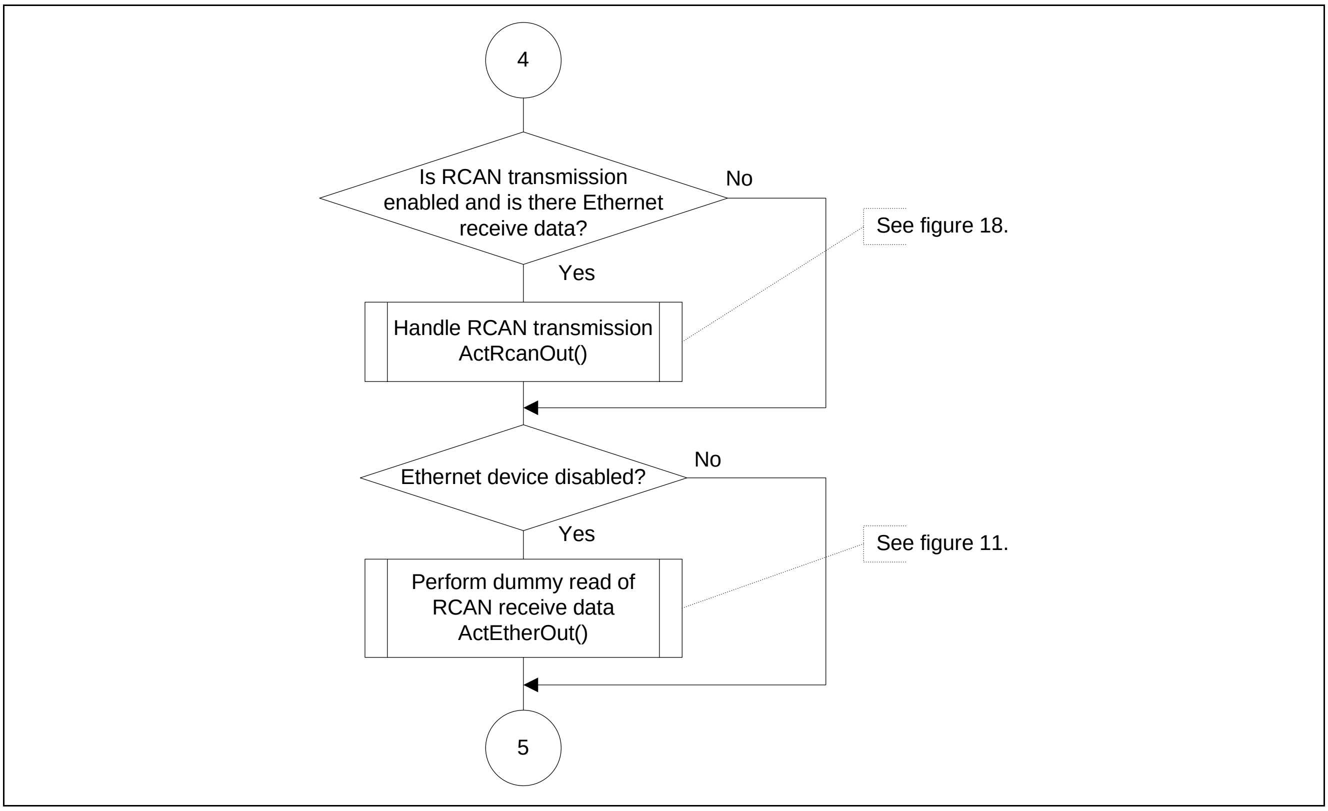

Figure 8 Main Process Flowcharts (5/5)

4.2.2 TCP/IP Protocol Stack

This application note uses uIP 1.0, which is open-source software, as the TCP/IP protocol stack.

The uIP program is a TCP/IP protocol stack that was developed by Adam Dunkels of SICS for 8 and 16-bit microcontrollers.

This application note uses Telnet as a TCP/IP application running on uIP.

Note: SICS: **S**wedish **I**nstitute of **C**omputer **S**cience

Figures 9 and 10 show the flowcharts for the Telnet application.

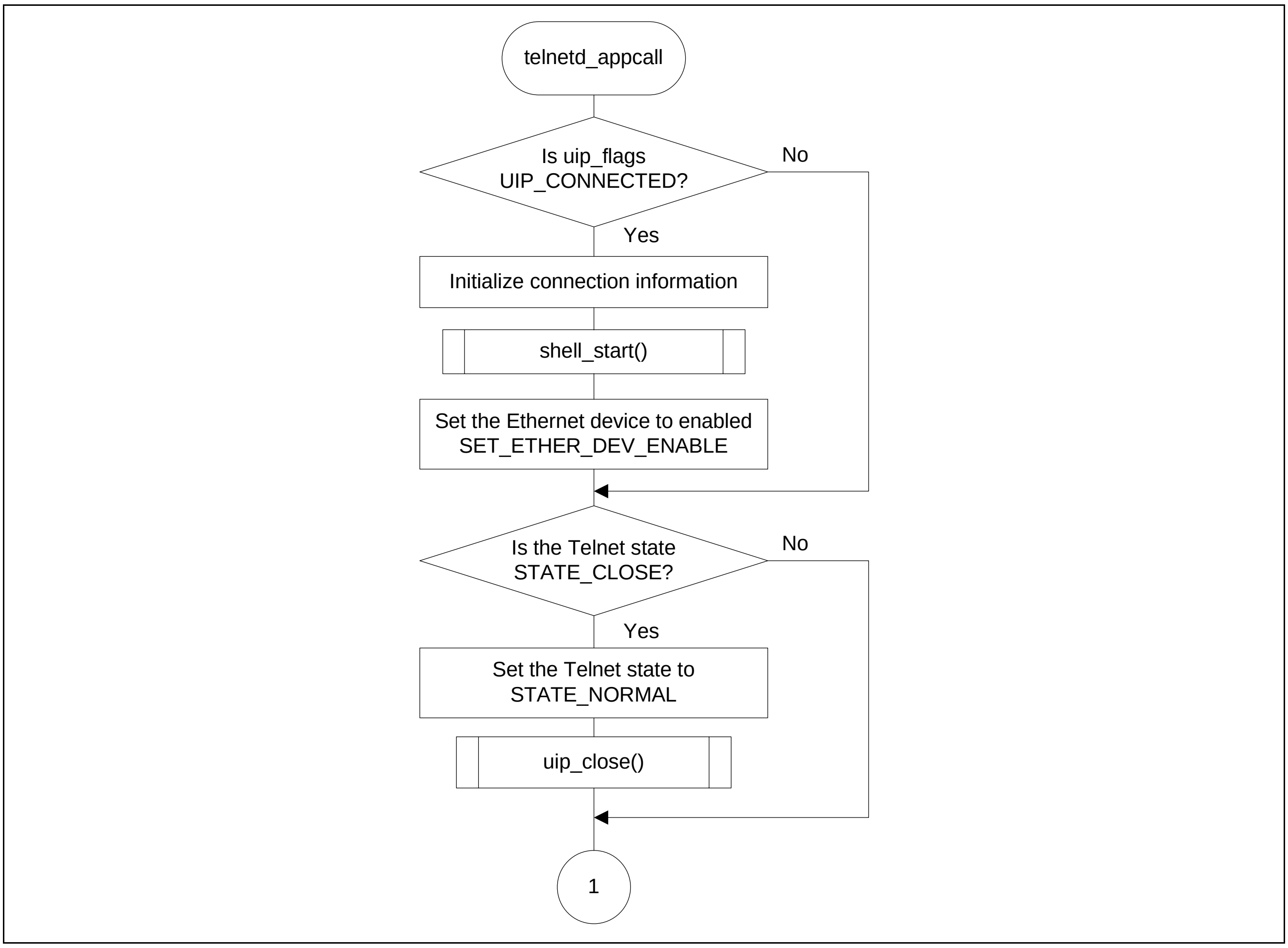

Figure 9 Telnet Application (telnetd_appcal) Flowchart (1/2)

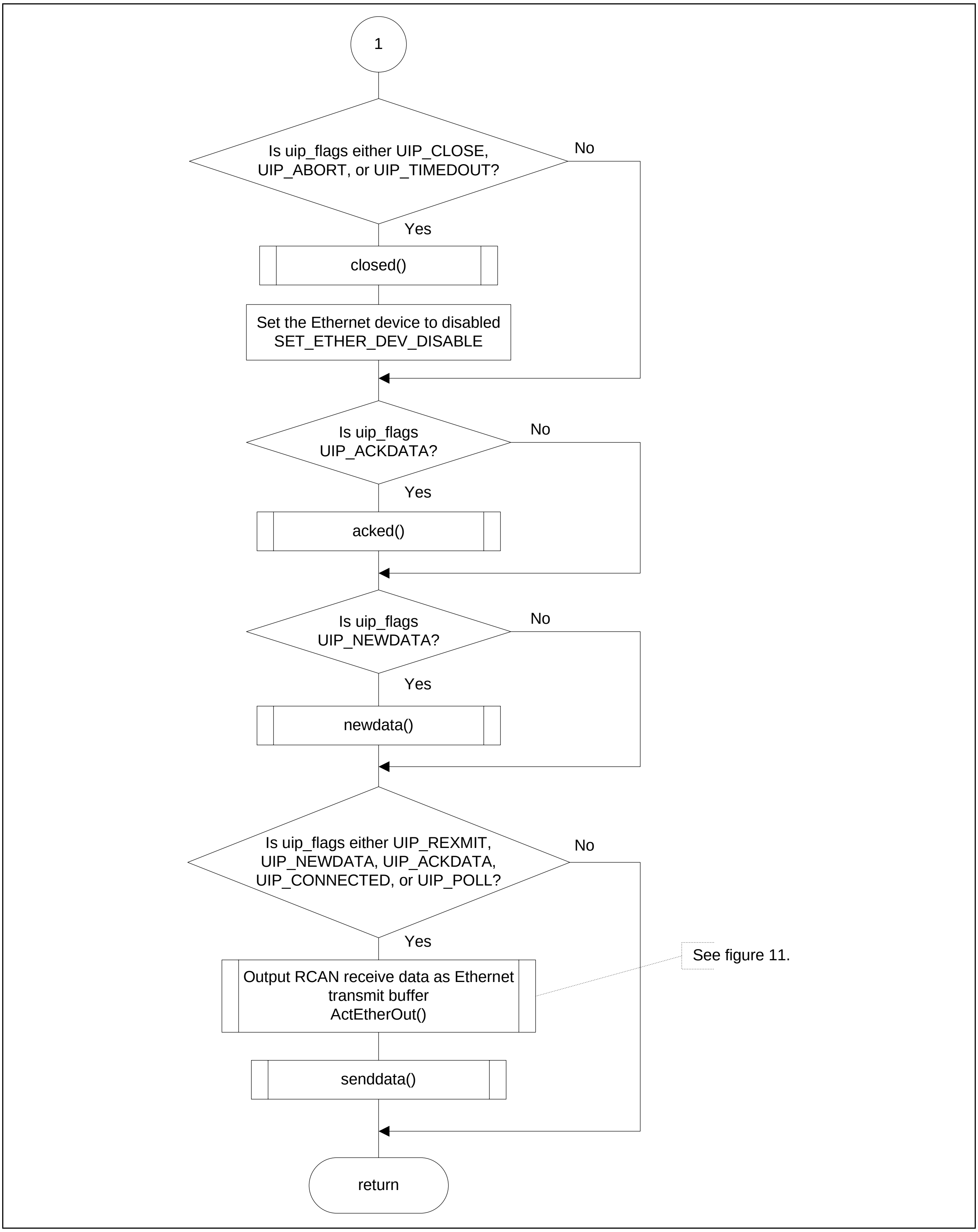

Figure 10 Telnet Application (telnetd_appcal) Flowchart (2/2)

4.2.3 Ethernet Driver

The Ethernet driver provides the Ethernet frame send/receive functionality that uses the SH7216 Ethernet controller (EtherC) and Ethernet controller direct memory access controller (E-DMAC).

The SH7216 provides a dedicated DMAC (E-DMAC) for use by the Ethernet controller so that Ethernet frames can be transmitted and received efficiently. The E-DMAC can be controlled easily using descriptors generated in memory.

Table 5 lists the Ethernet driver related functions.

Table 5 Ethernet Driver Functions

No.	Function	Description
1	ActEtherOut	RCAN receive buffer to Ethernet transmission processing
2	ActEtherIn	Ethernet to RCAN transmit buffer storage
3	R_Ether_Open	EtherC, E-DMAC, and PHY initialization
4	R_Ether_Read	Ethernet frame reception
5	R_Ether_Write	Ethernet frame transmission
6	lan_isr	EtherC and E-DMAC interrupt handling

Note: Functions of the form R_xxx_xxx are Ethernet driver public functions.

Figures 11 to 16 show the flowcharts for these functions.

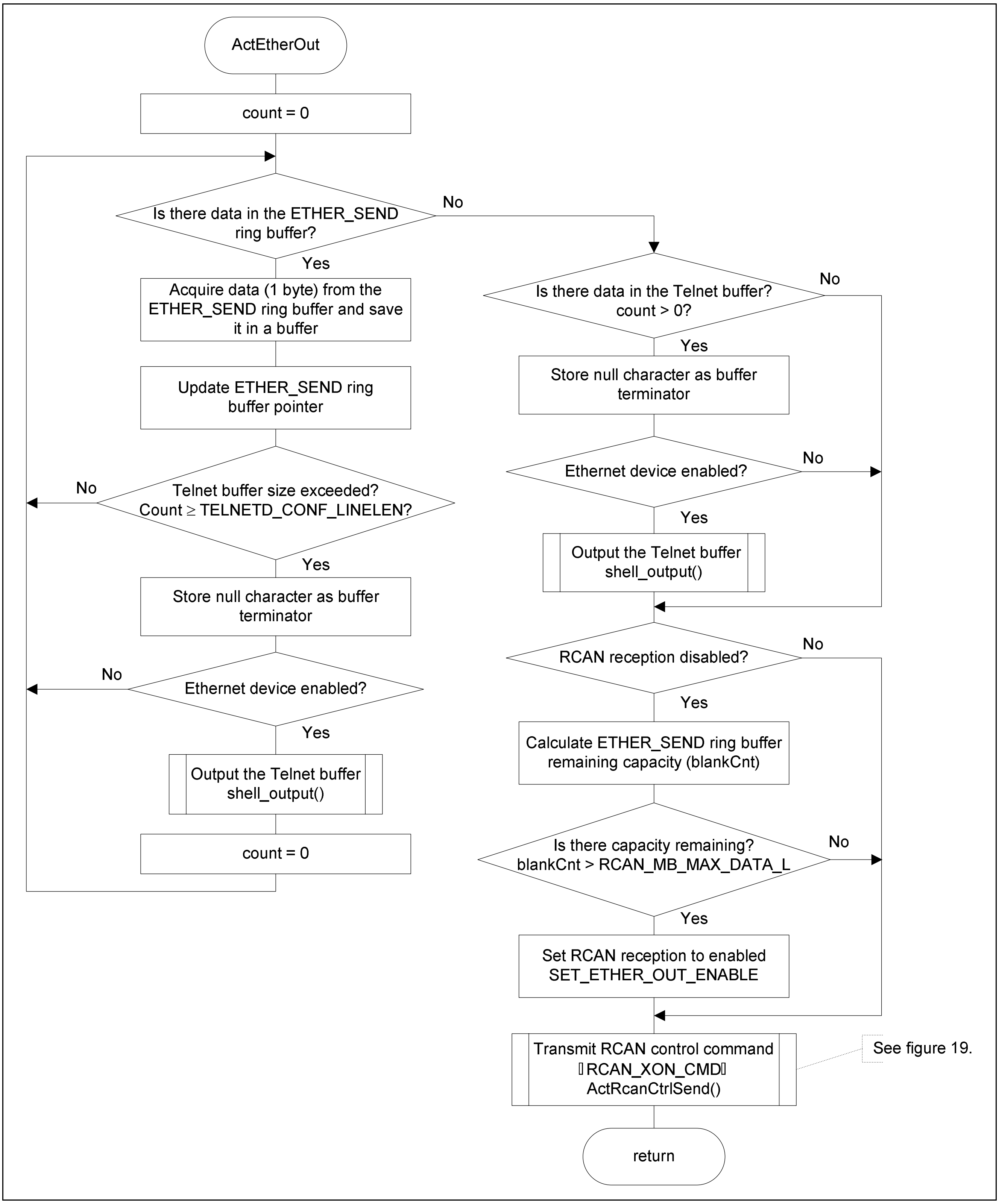

Figure 11 ActEtherOut() Flowchart

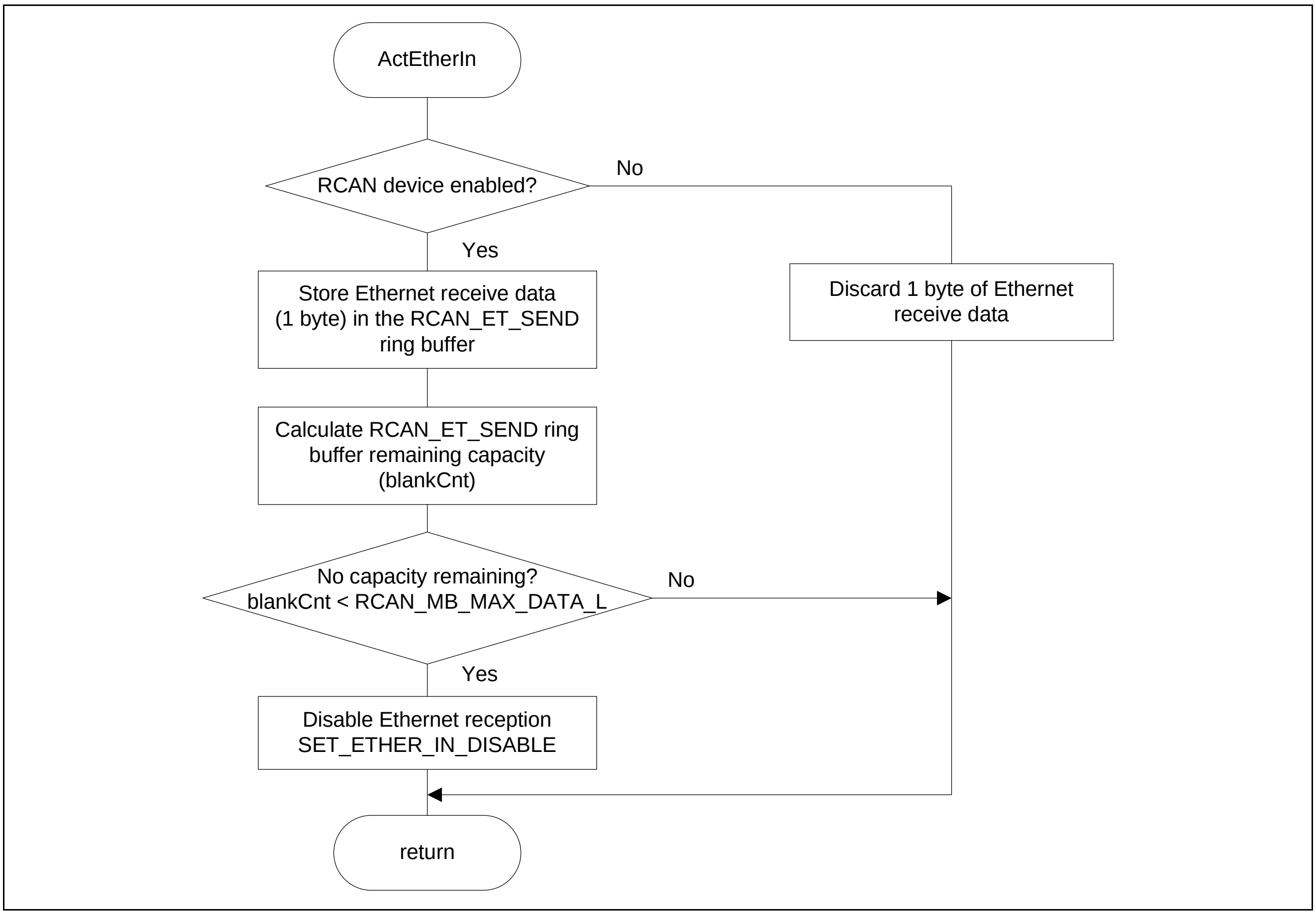

Figure 12 ActEtherIn() Flowchart

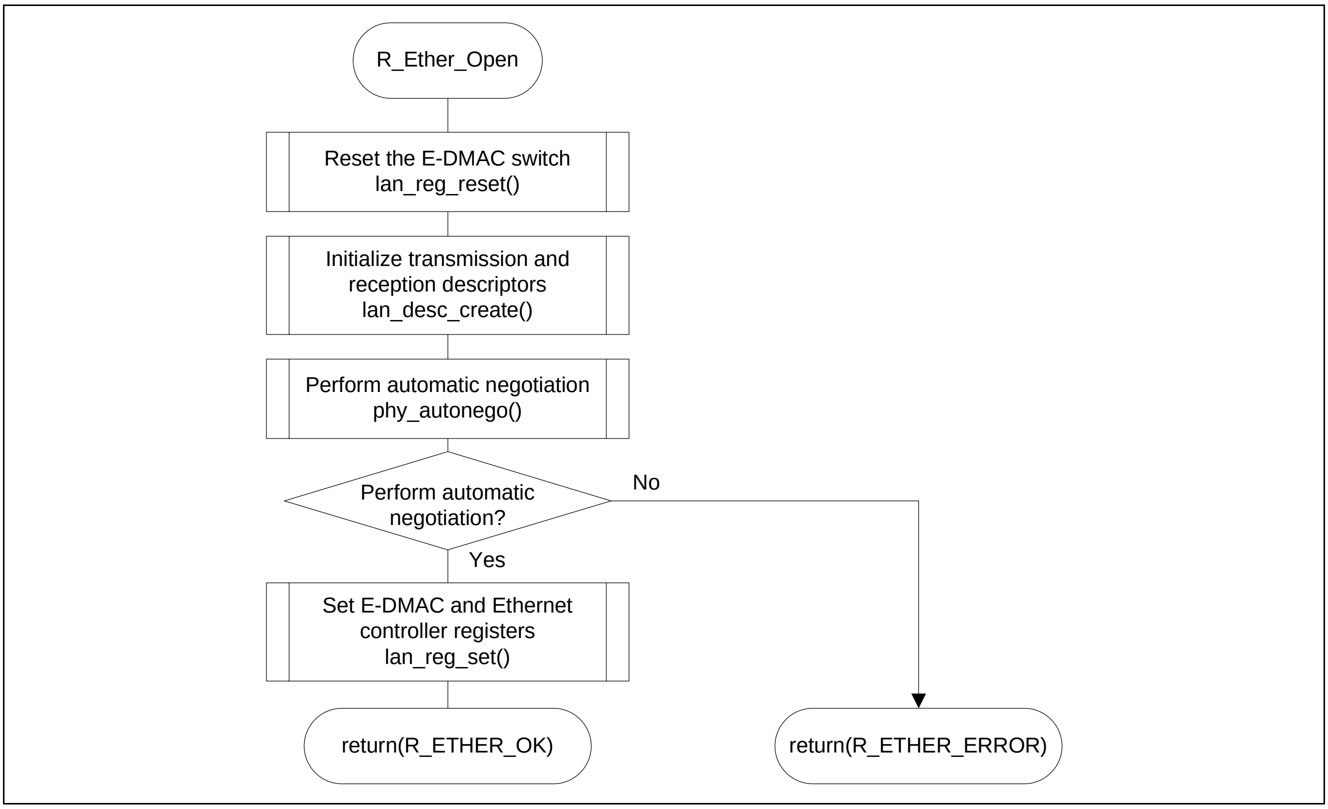

Figure 13 R_Ether_Open() Flowchart

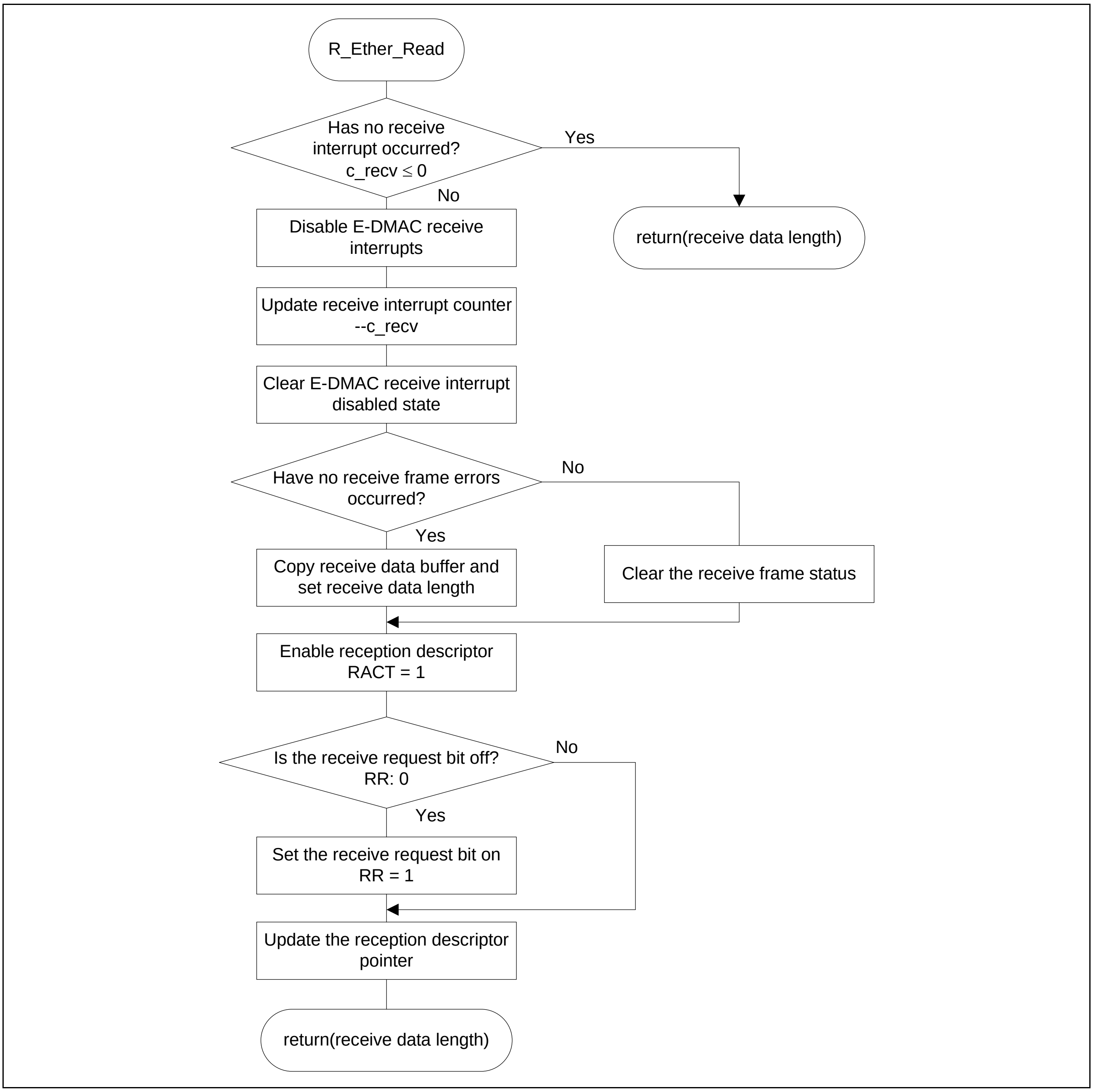

Figure 14 R_Ether_Read() Flowchart

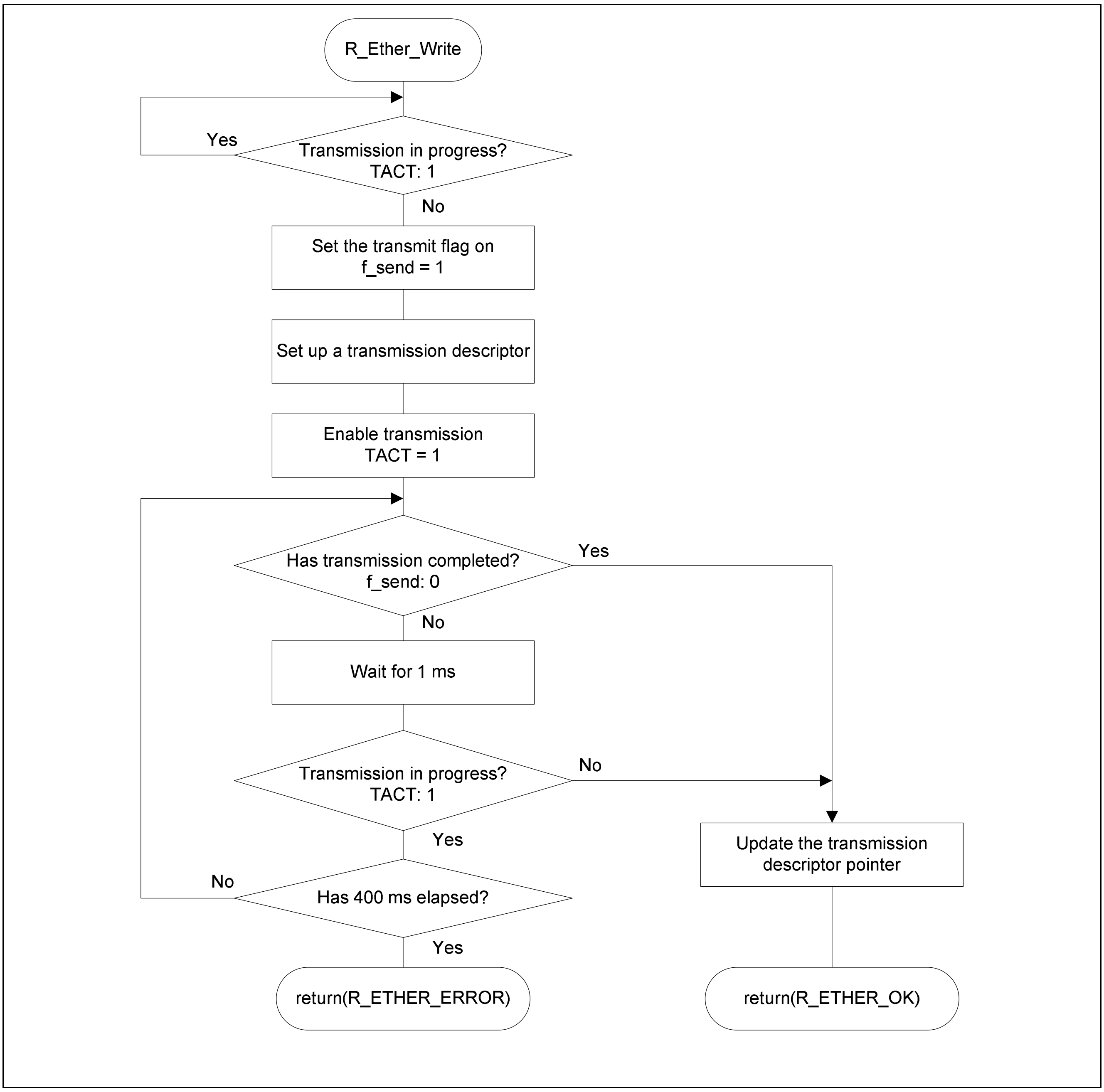

Figure 15 R_Ether_Write() Flowchart

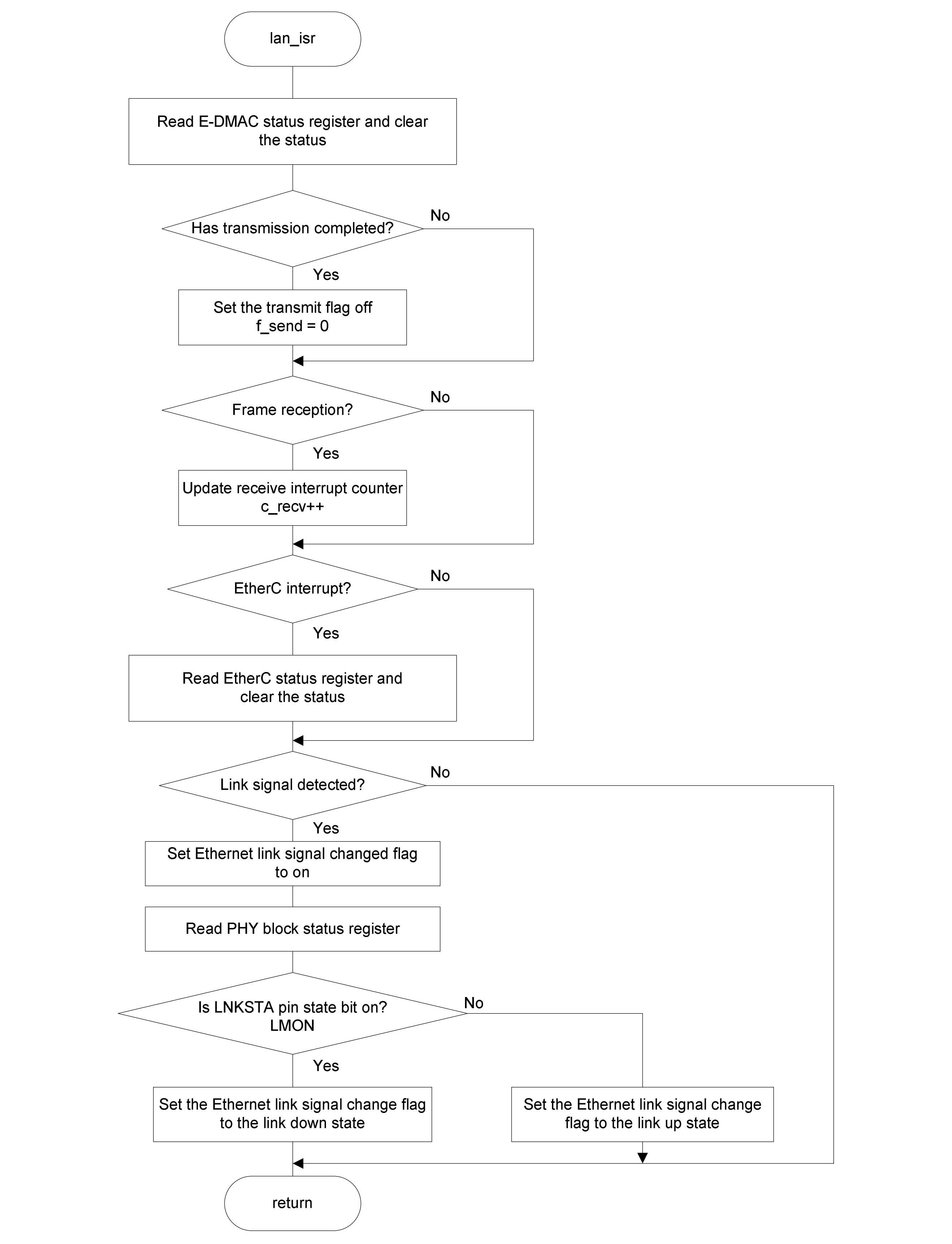

Figure 16 lan_isr() Flowchart

4.2.4 RCAN Driver

The RCAN driver is a group of functions that provide mailbox transfer processing using the SH7216 control area network (RCAN-ET) module.

Table 6 lists the RCAN driver functions.

Table 6 RCAN Driver Functions

No.	Function	Description
1	RcanOpen	RCAN initial connection processing
2	ActRcanOut	RCAN transmission processing
3	ActRcanCtrlSend	RCAN control command transmission processing
4	ActRcanCtrlRecv	RCAN control command reception processing
5	ActRcanRecv	RCAN reception processing
6	R_CAN_Initial*	RCAN initialization
7	R_CAN_SetBitrate*	RCAN communication speed settings
8	R_CAN_SetMask*	RCAN mailbox mask settings
9	R_CAN_EnablePort*	RCAN port enable settings
10	R_CAN_SetTxStdId	RCAN transmit mailbox ID setting
11	R_CAN_SetTxStdData*	RCAN transmission processing
12	R_CAN_CheckTxStdData*	RCAN transmission checking
13	R_CAN_CheckTxComplete	RCAN transmit compete checking
14	R_CAN_SetRxStdData*	RCAN receive mailbox ID setting
15	INT_RCANET0_RM01_0	RCAN interrupt handling
16	CanConfigInterrupts	RCAN interrupt setup

Note: The R_xxx_xxx marked with an asterisk are RCAN driver public functions.

Figures 17 to 33 show the flowcharts for these functions.

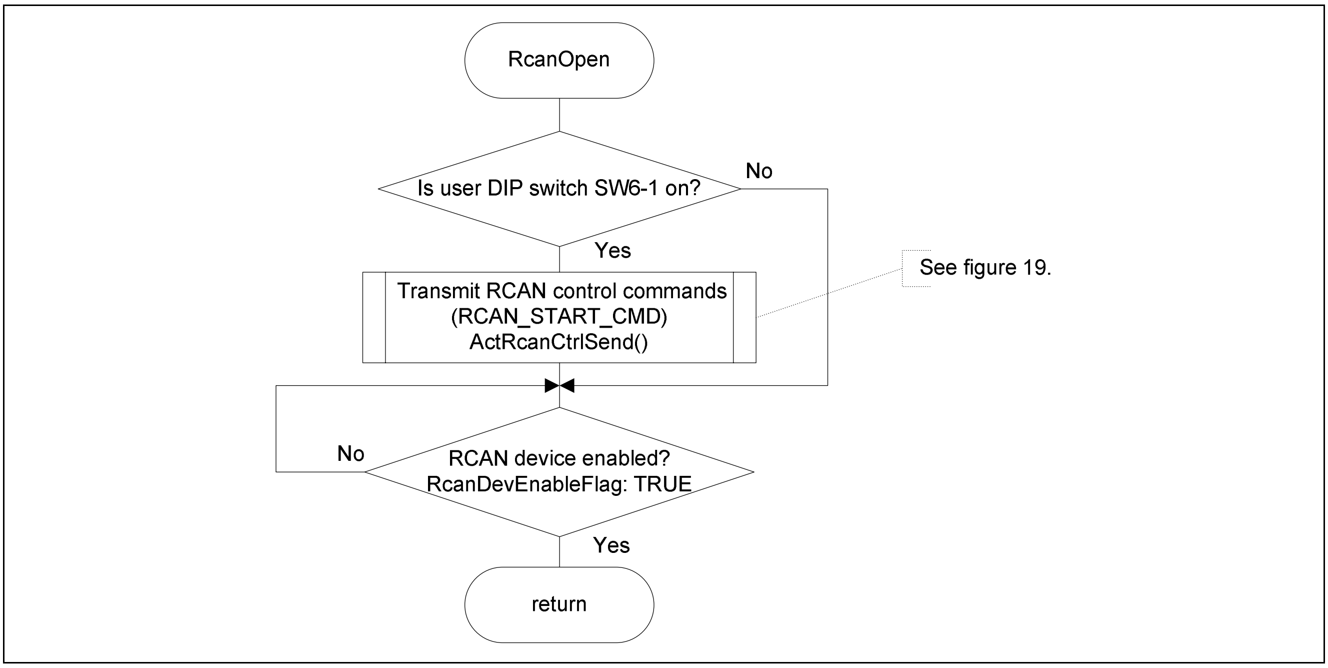

Figure 17 RcanOpen() Flowchart

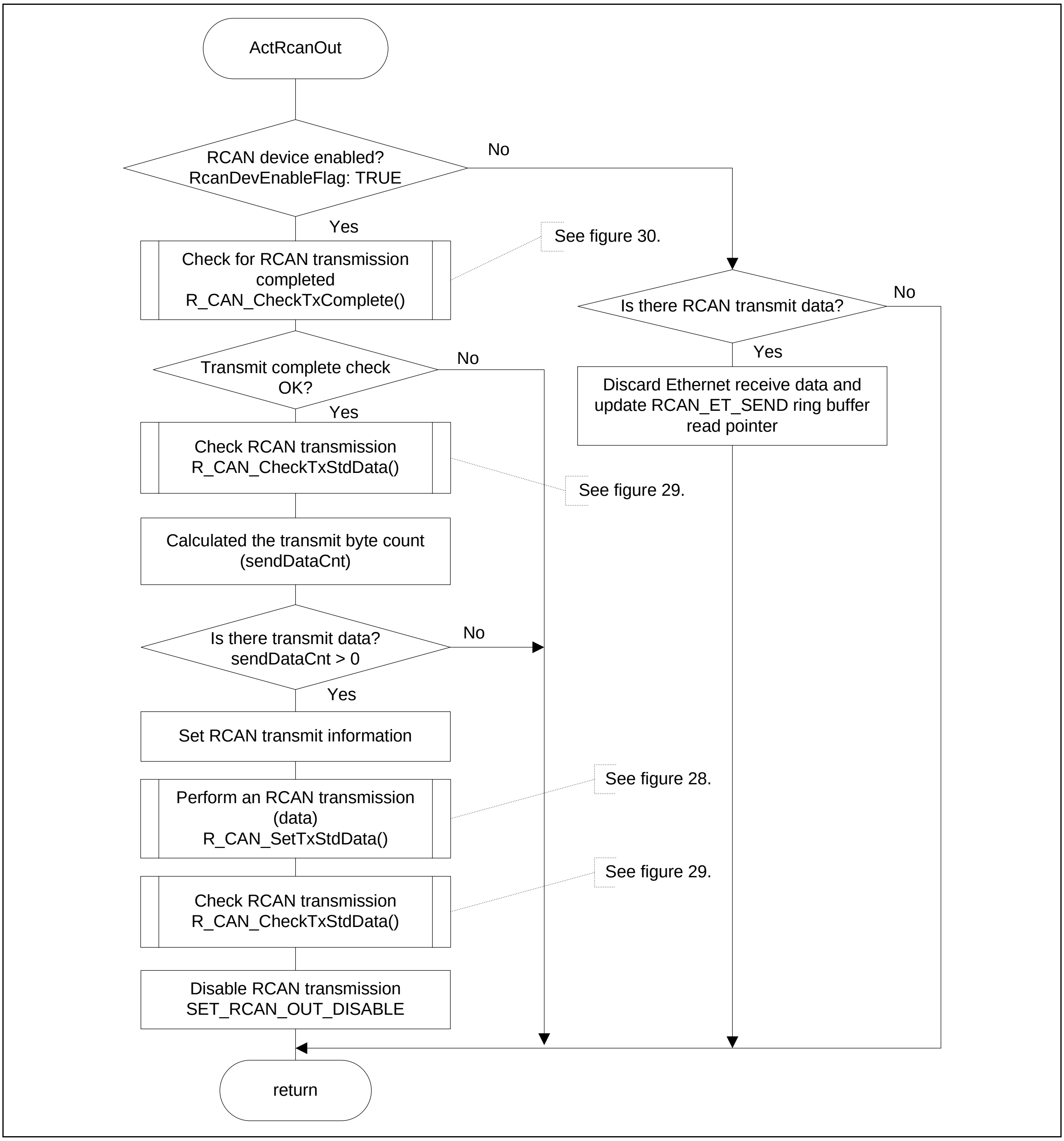

Figure 18 ActRcanOut() Flowchart

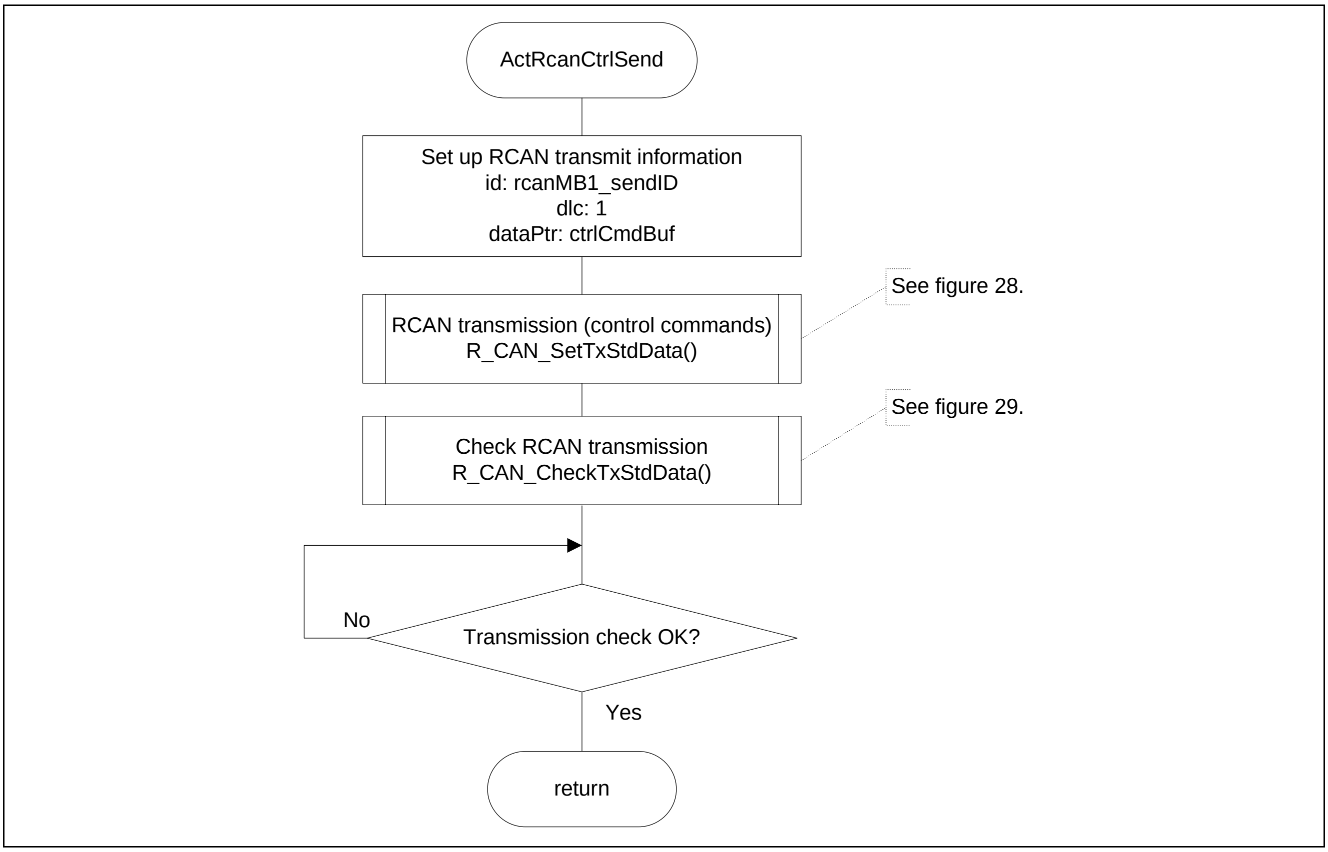

Figure 19 ActRcanCtrlSend() Flowchart

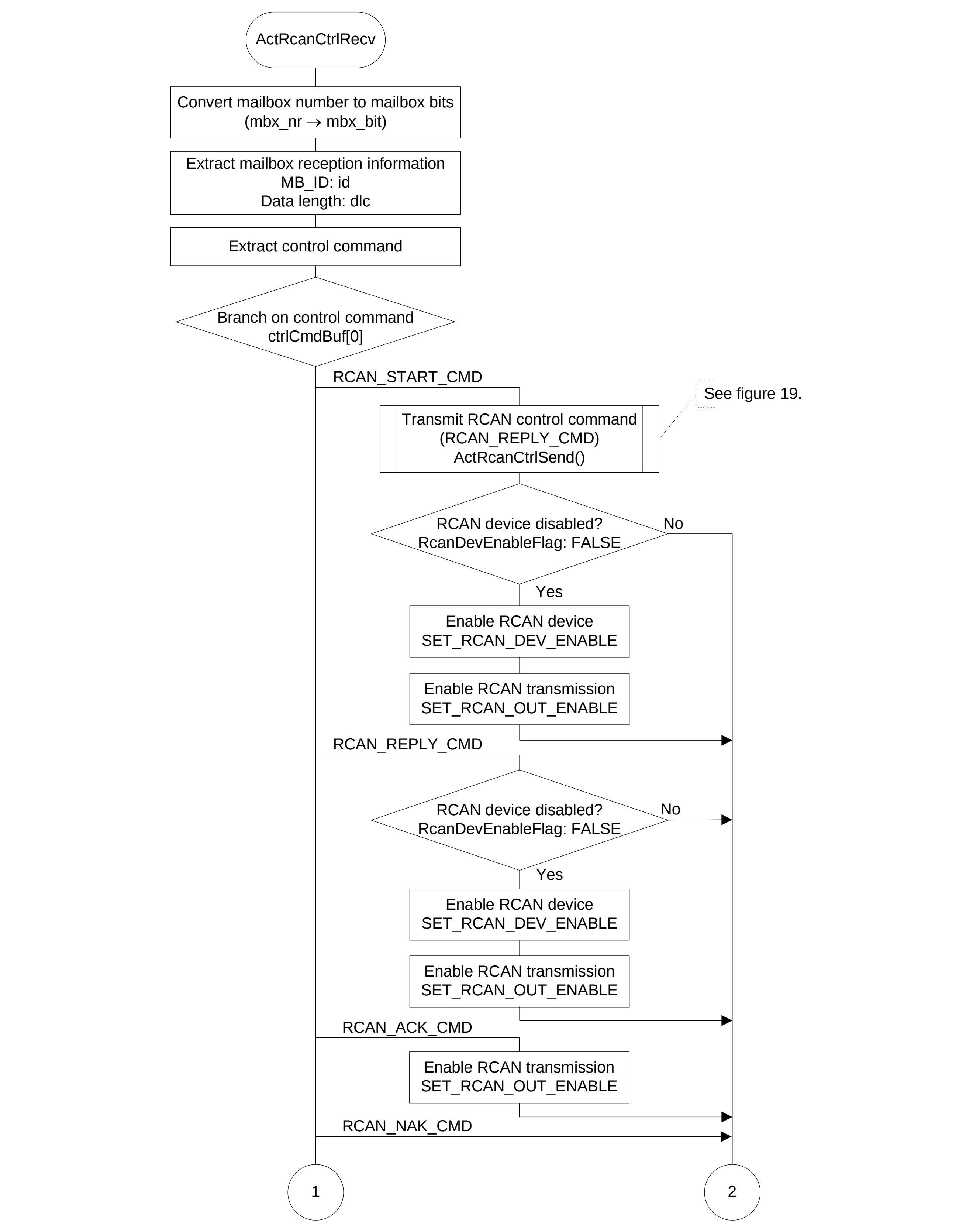

Figure 20 ActRcanCtrlRecv() Flowchart (1/2)

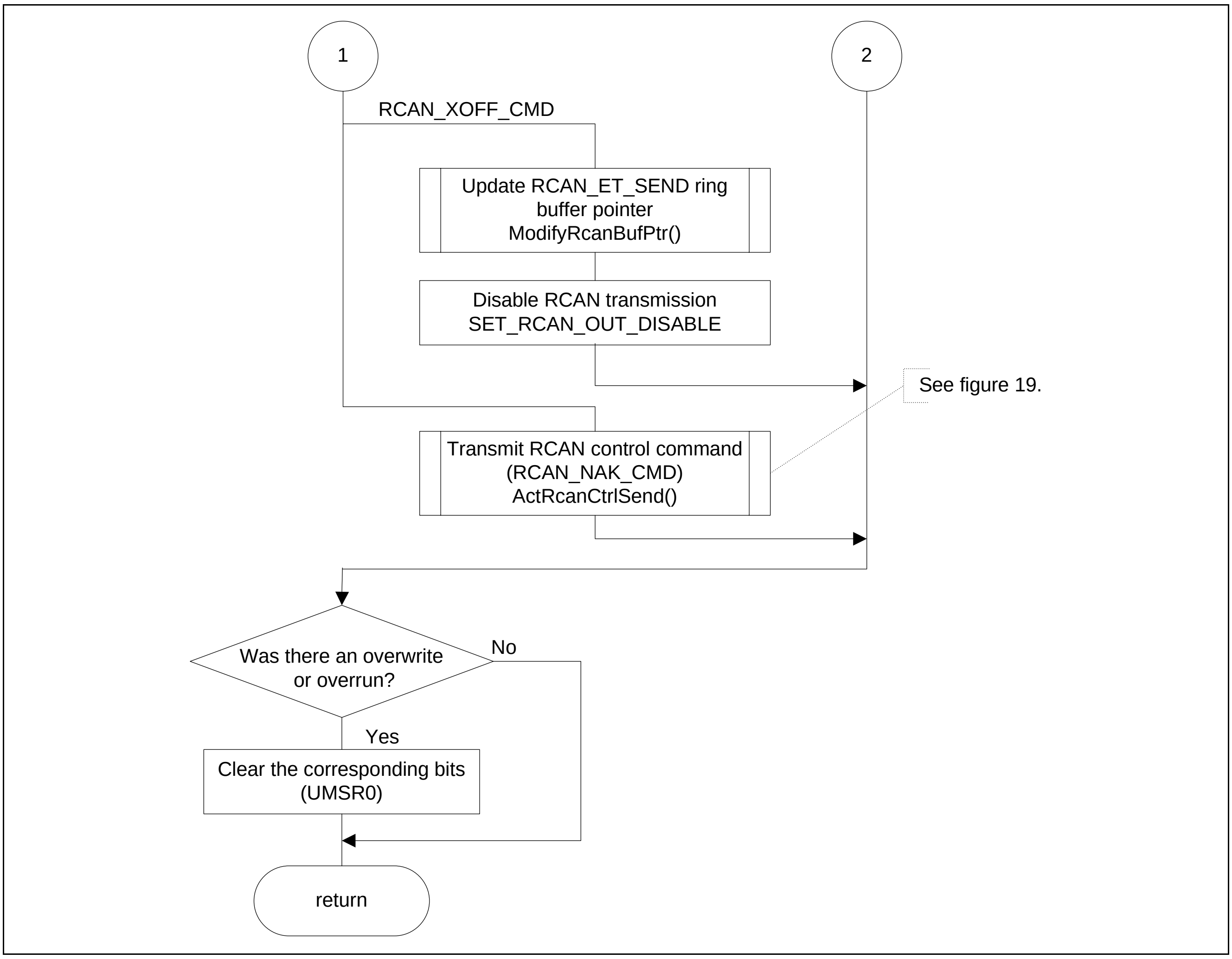

Figure 21 ActRcanCtrlRecv() Flowchart (2/2)

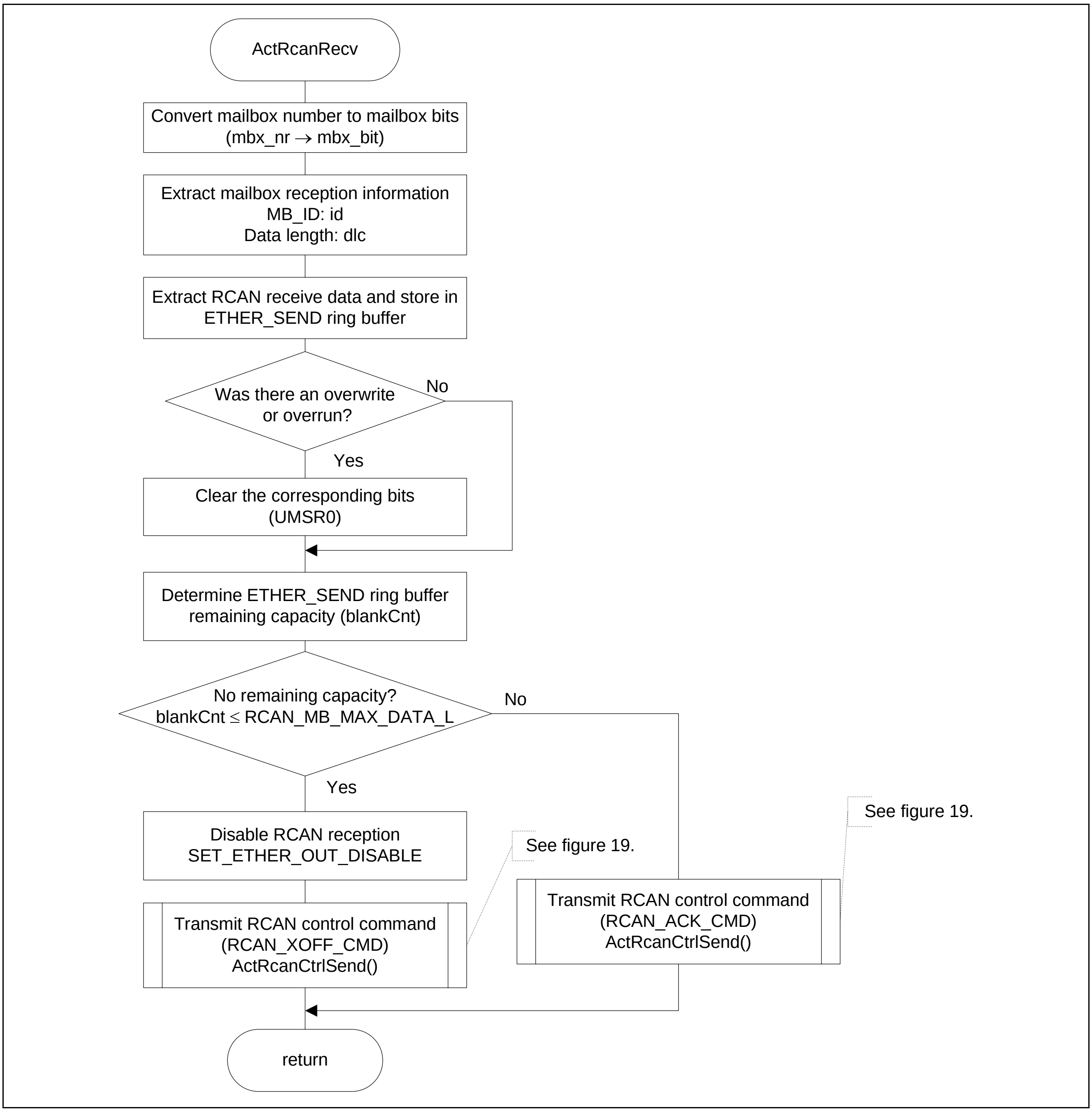

Figure 22 ActRcanRecv() Flowchart

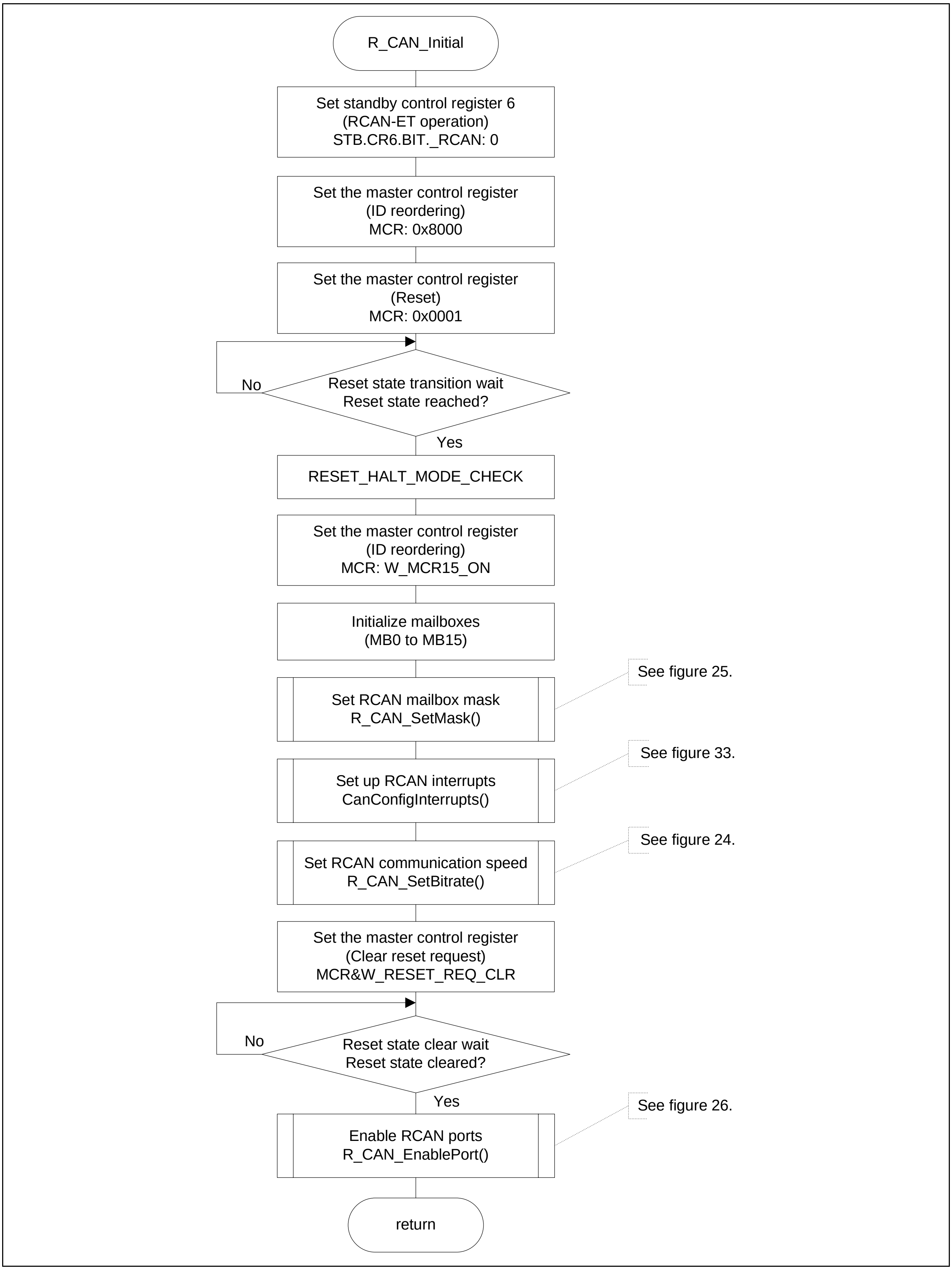

Figure 23 R_CAN_Initial() Flowchart

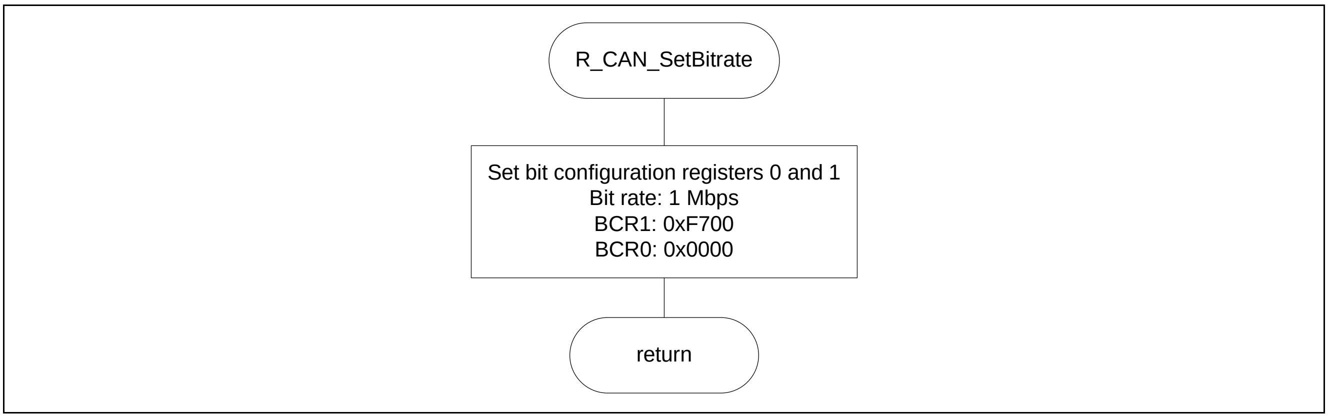

Figure 24 R_CAN_SetBitrate() Flowchart

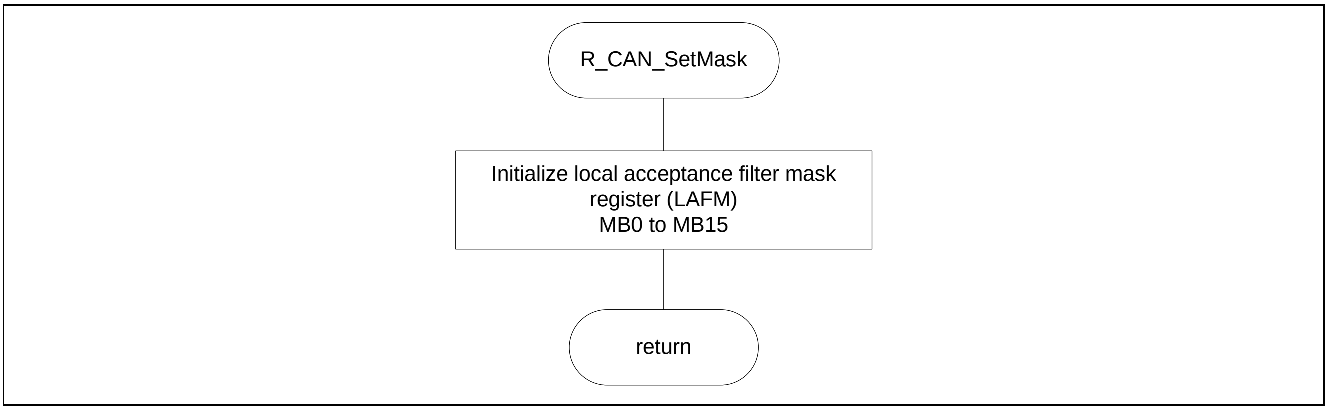

Figure 25 R_CAN_SetMask() Flowchart

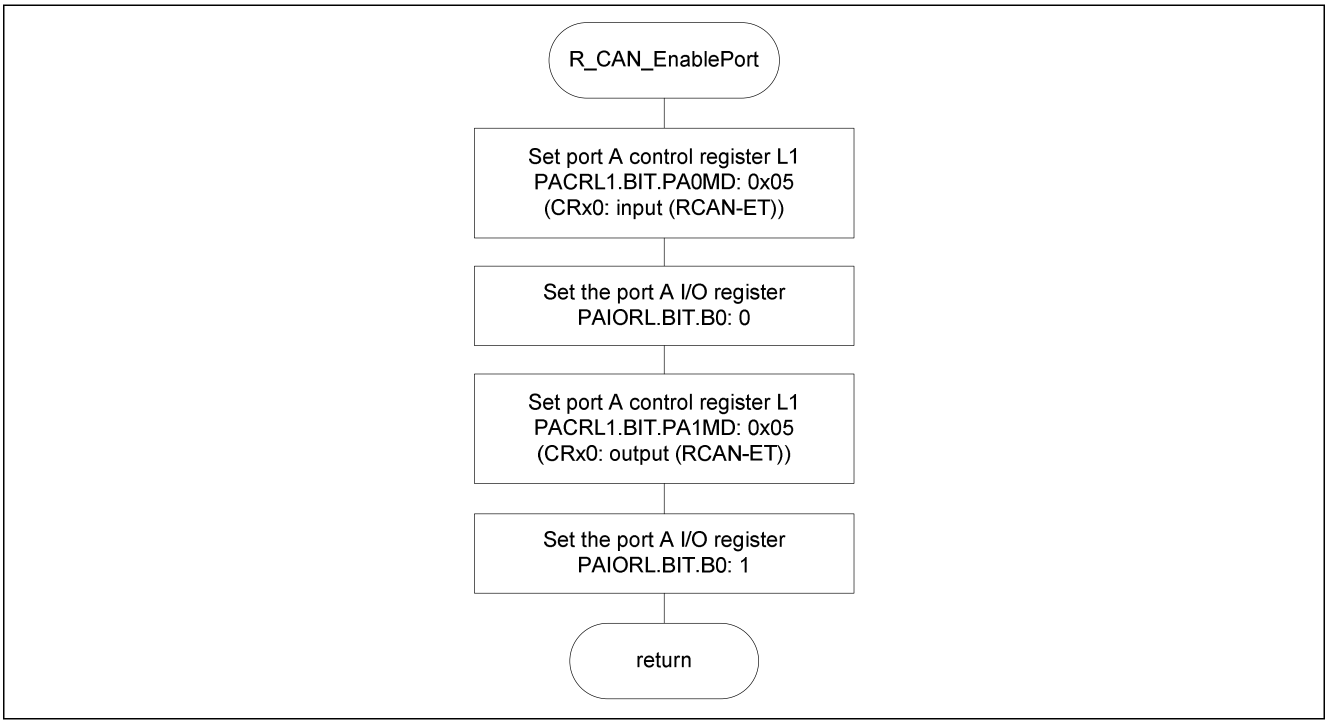

Figure 26 R_CAN_EnablePort() Flowchart

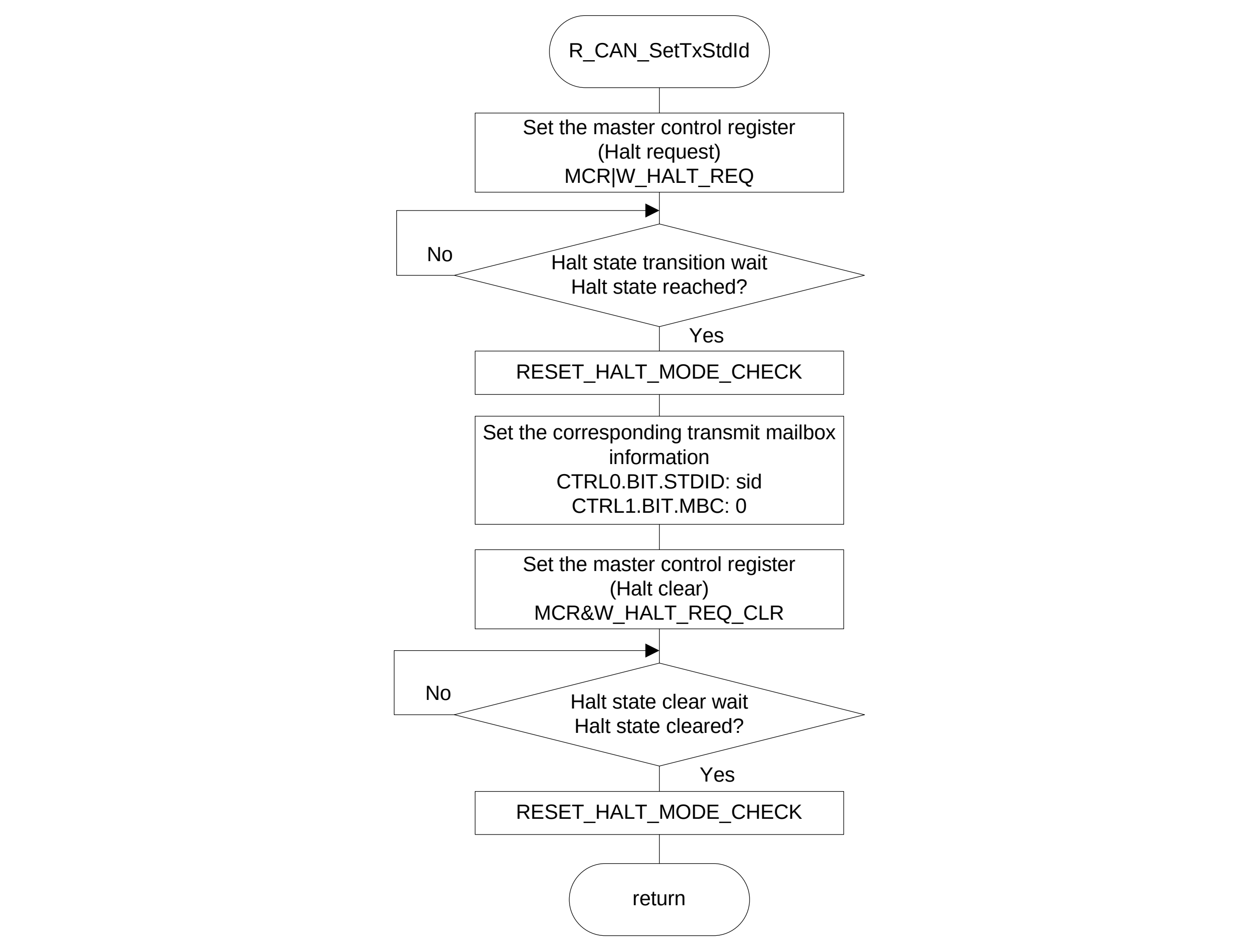

Figure 27 R_CAN_SetTxStdId() Flowchart

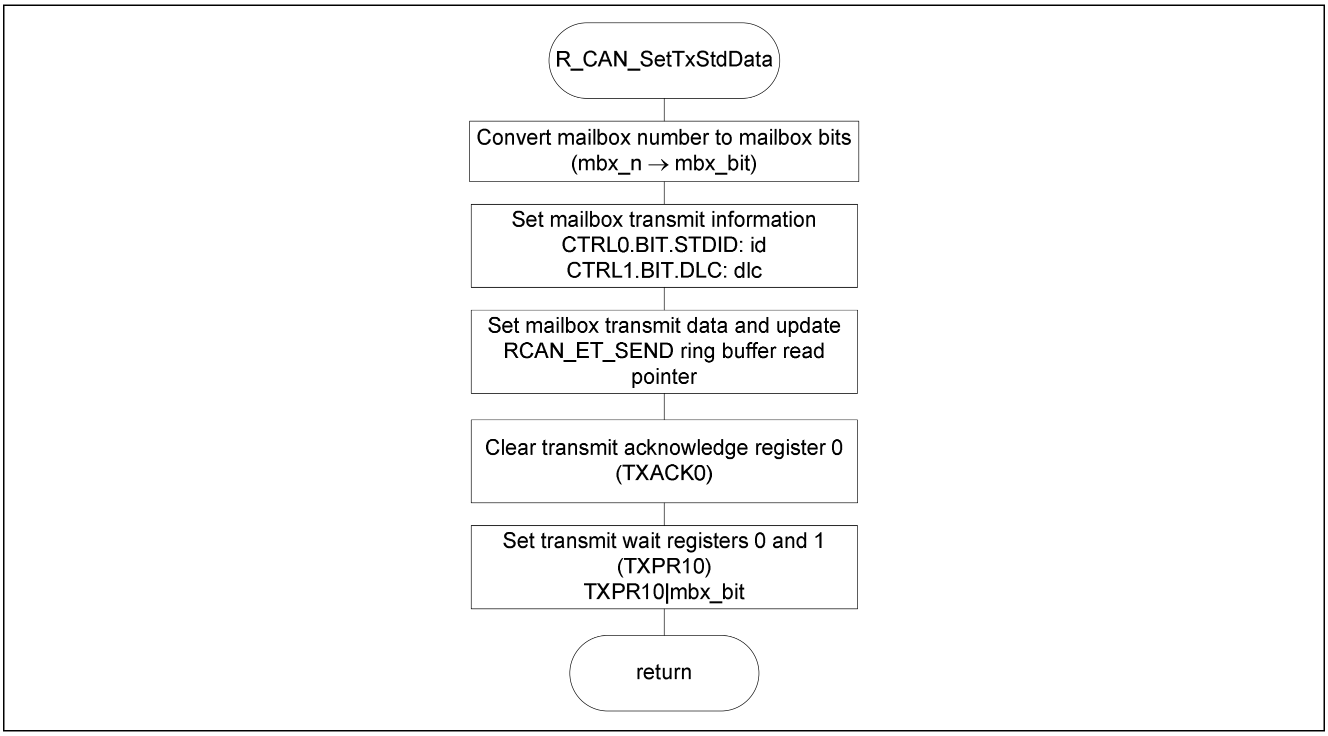

Figure 28 R_CAN_SetTxStdData() Flowchart

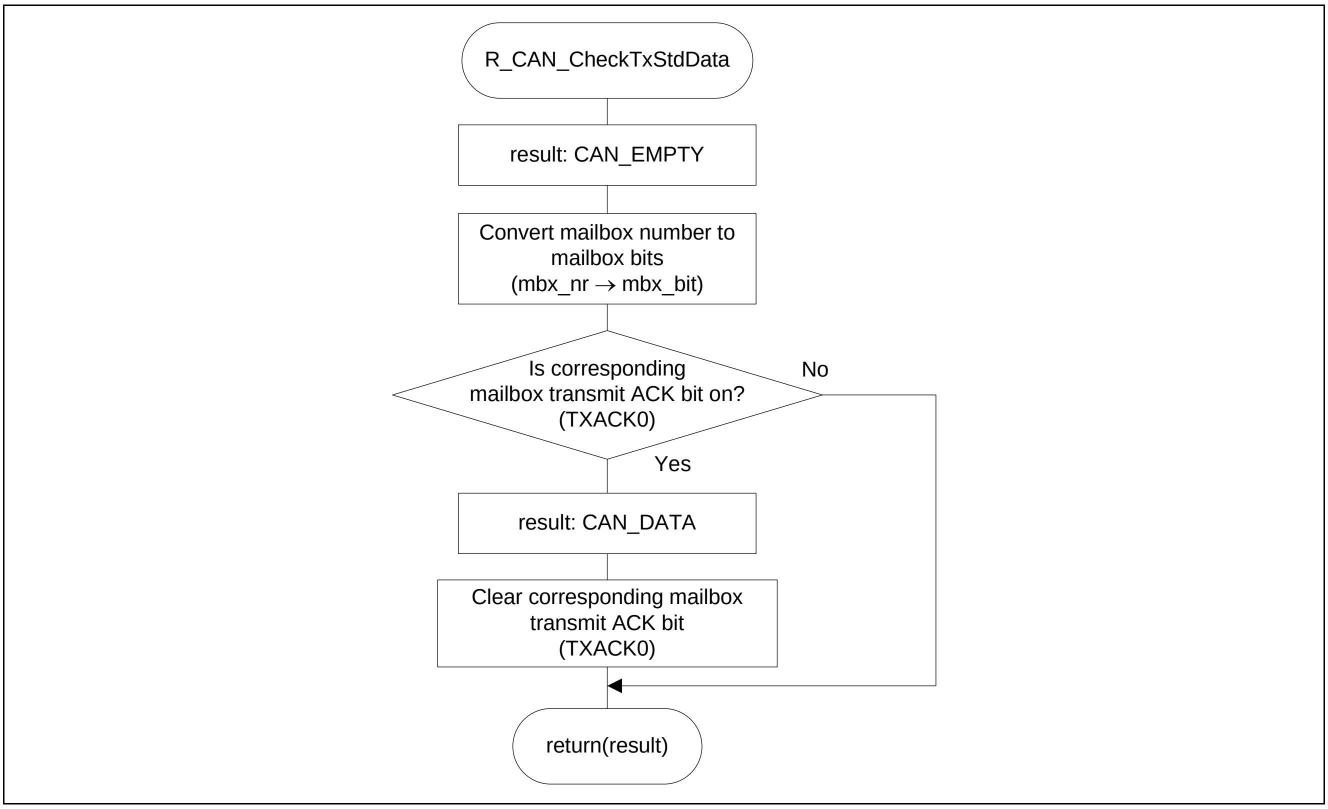

Figure 29 R_CAN_CheckTxStdData() Flowchart

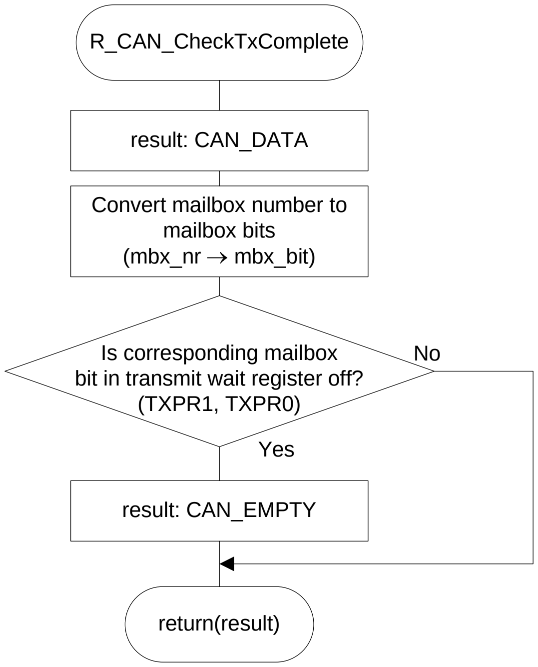

Figure 30 R_CAN_CheckTxComplete() Flowchart

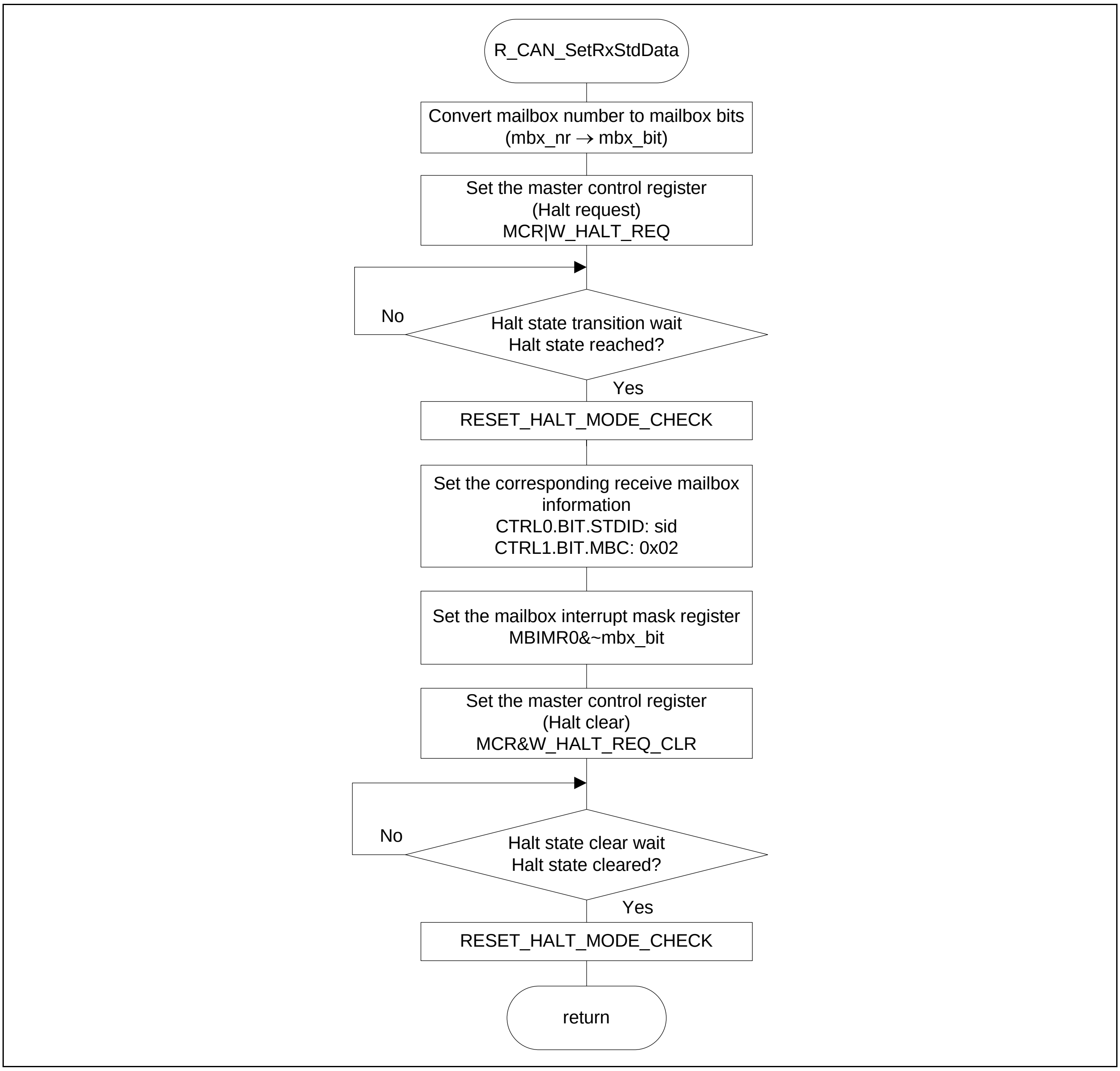

Figure 31 R_CAN_SetRxStdData() Flowchart

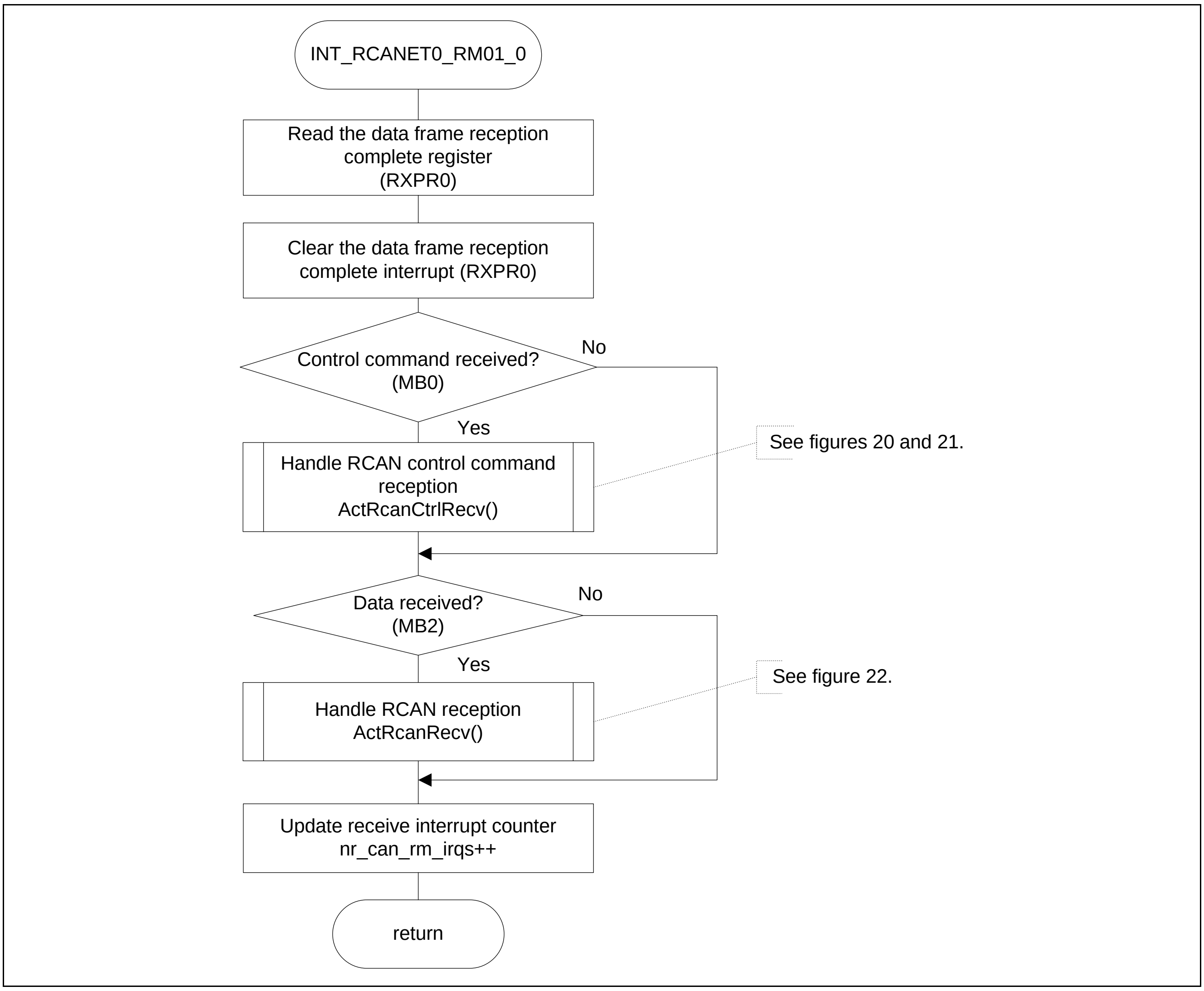

Figure 32 INT_RCANET0_RM01_0() Flowchart

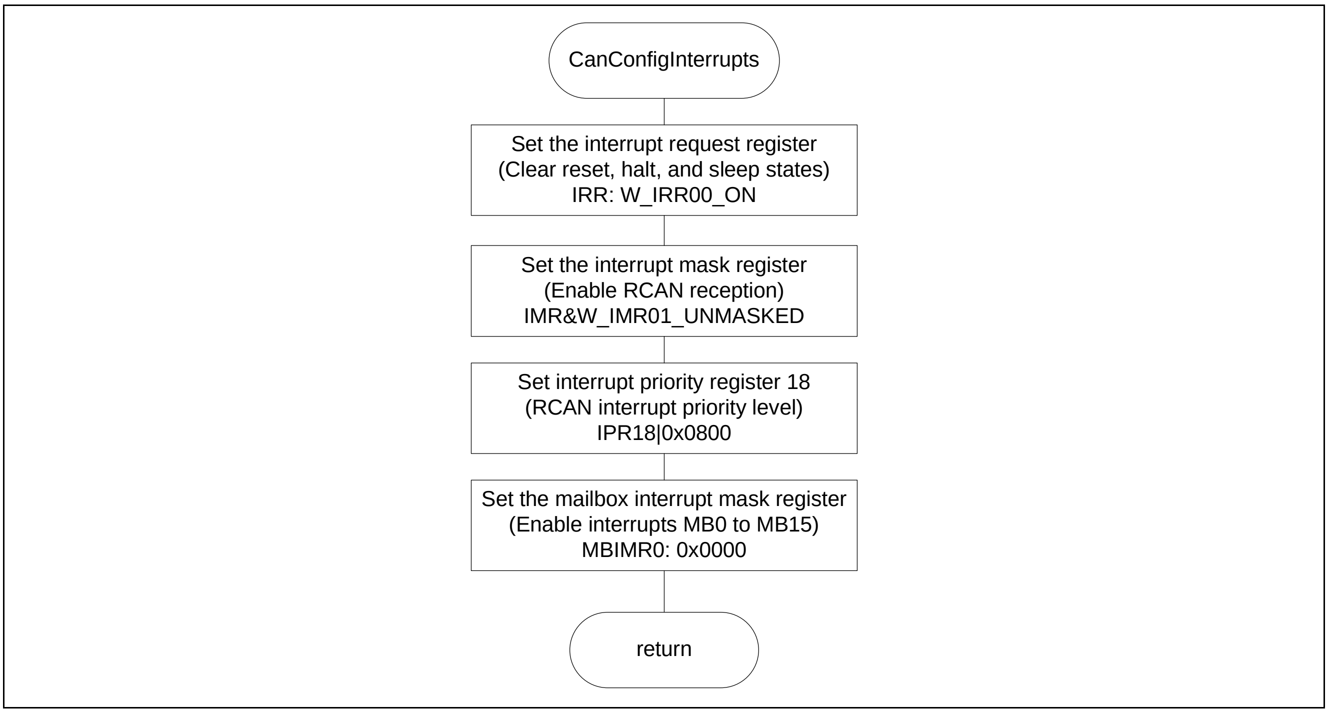

Figure 33 CanConfigInterrupts() Flowchart

4.2.5 Timer Driver

The timer driver is a group of functions that provide periodic control processing using the SH7216 compare match timer (CMT).

The timer driver is used as the uIP timer.

Table 7 lists the timer driver functions.

Table 7 Tmer Driver Functions

No.	Function	Description
1	timer_init	CMT0 initialization and startup processing
2	timer_stop	CMT0 stop processing
3	clock_time	CMT0 counter value acquisition
4	int_cmt0_isr	CMT0 interrupt handling

Figures 34 to 37 show the flowcharts for these functions.

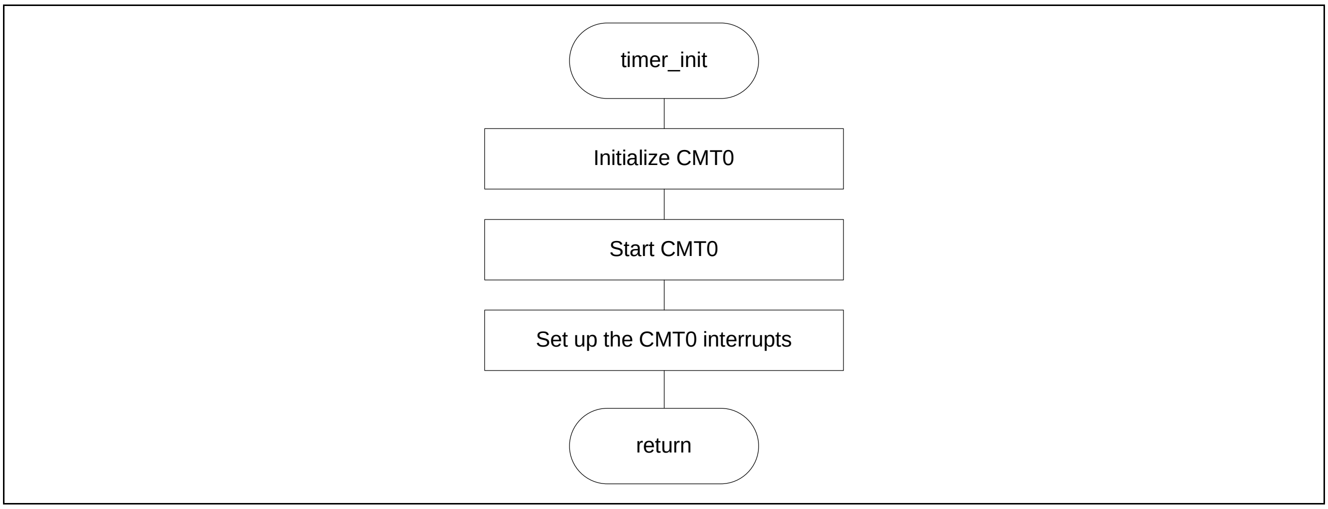

Figure 34 timer_init() Flowchart

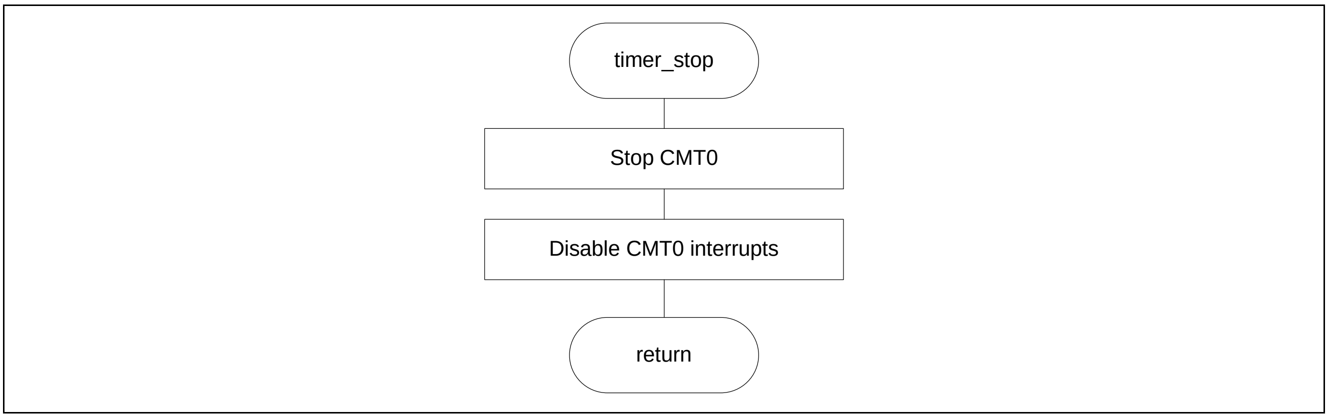

Figure 35 timer_stop() Flowchart

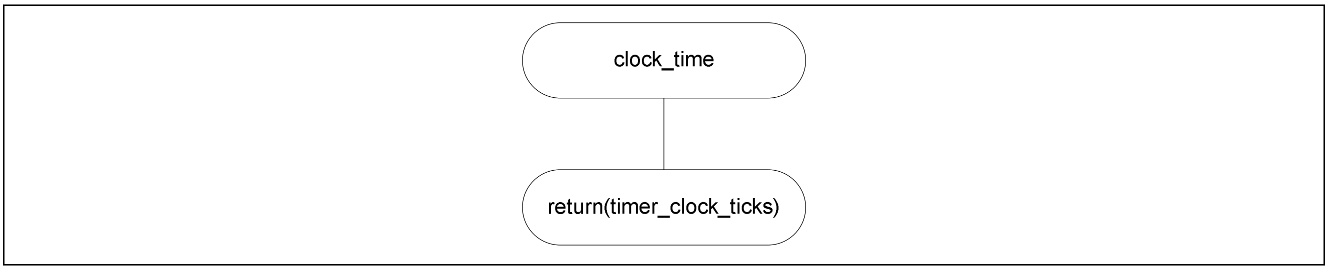

Figure 36 clock_time() Flowchart

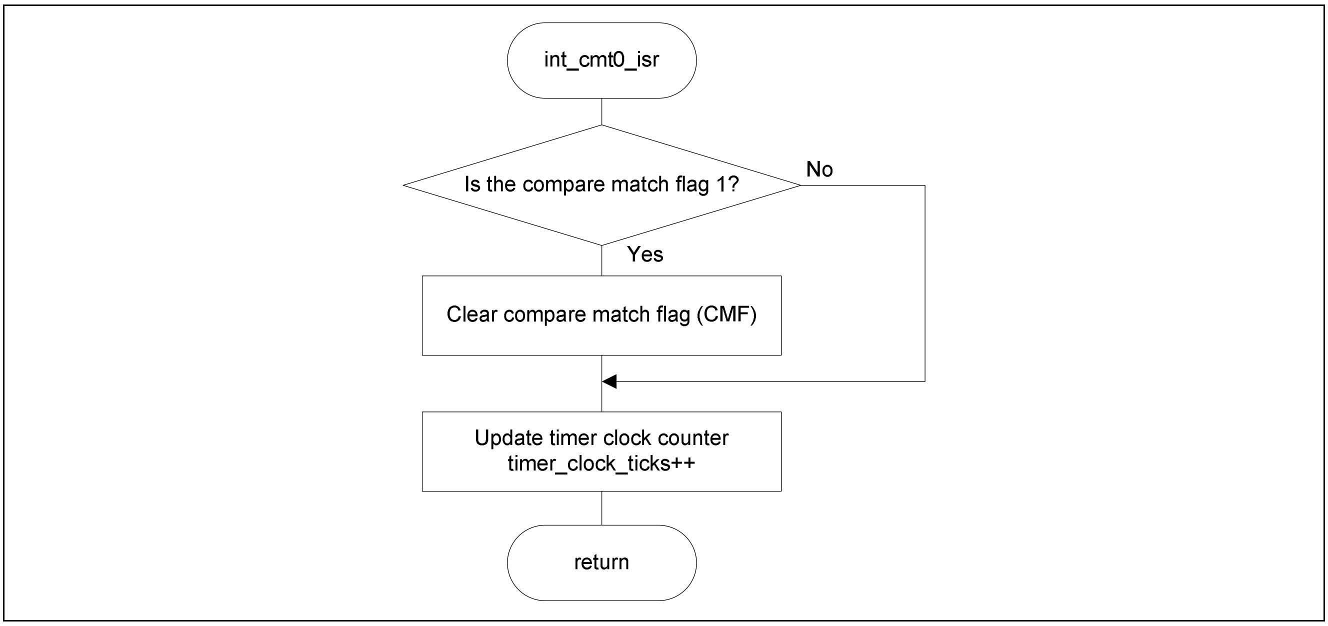

Figure 37 int_cmt0_isr() Flowchart

4.2.6 Section Settings

Table 8 lists information on the sections of the SH7216 sample program.

Table 8 SH7216 Sample Program Section Information

No.	Address	Section Name	Description
1	0x00000000	DVECTTBL	Vector table
2	0x00000400	P	Program area
3		P_CH_CPG	CPG setting program area
4		C	Constant area
5		D	Initialized data area
6	0xFFF80000	B	Uninitialized data area
7		R	Initialized data area (for RAM allocation)
8	0xFFF85000	R_CH_CPG	CPG setting program area (for RAM allocation)
9	0xFFF88000	BETH_BUFF	Ethernet buffer area
10	0xFFF8C000	BETH_DESC	Ethernet descriptor area
11	0xFFF8FC00	S	Stack area

Note: The item 7 and item 8 Ethernet driver areas must be aligned with 32-byte boundaries.

5. Reference Documents

- Software Manual
  SH2A/SH2A-FPU Software Manual (REJ09B0051)
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Hardware Manual
  SH7216 Group Hardware Manual (REJ09B0543)
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- uIP 1.0 Reference Manual
  The uIP Embedded TCP/IP Stack: The uIP 1.0 Reference Manual (June 2006)

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Mar.28.11	—	First edition issued
1.10	May.18.11	—	Source project revised

General Precautions in the Handling of MPU/MCU Products

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

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

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

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

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

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

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

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

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

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

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

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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

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

Colophon 1.1