R32C/100 Series

I^{2}C-bus Interface Using UARTi Special Mode 1 (Master Transmit/Receive)

REJ05B1395-0100
Rev.1.00
Aug 31, 2010

1. Abstract

This document describes the master transmit/receive processes in I^{2}C-bus interface single master communication using the R32C/100 Series serial interface (UART2) special mode 1 (I^{2}C mode).
Seven channels (UART0 to UART6) can be used in special mode 1 in the R32C/118 Group.
If channels other than UART0 to UART6 are used, refer to the hardware user's manual and modify the registers associated with UARTi (i = 0 to 6).

2. Introduction

The application example described in this document applies to the following microcomputer (MCU) and parameter:

- MCU: R32C/118 Group
- XIN Clock: 16 MHz

This application note can be used with other R32C/100 Series MCUs which have the same special function registers (SFRs) as the above group. Check the user’s manual for any modifications to functions. Careful evaluation is recommended before using the program described in this application note.

3. Application Example

3.1 Program Outline

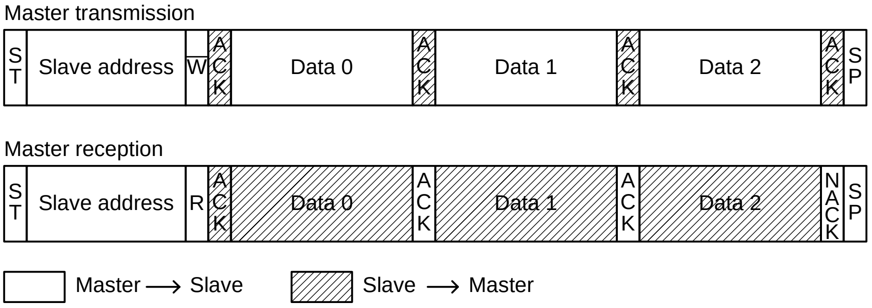
Transmission is performed in 3-byte data both in master transmission and reception. Master transmission and reception are repeated alternately. The transmission and reception procedures conform to the I^{2}C-bus communication protocol when used under the following conditions:

- Slave address: 7 bits
- Transfer rate: Approximately 350 kbps [1]
- Transfer data length: 1 to 255 bytes (not including the slave address)
- Single master communication (multi-master is not supported)
- Restart condition generation is not supported

Note:

1. The setting value is 378 kbps.
   When the clock synchronous function is enabled, there is a sampling delay of the noise filter width plus 1 to 1.5 cycles of the U2BRG count source. As there is also a delay of the SCL clock when high is determined, the SCL clock high width is extended. Therefore, the actual SCL clock becomes slower than SCL clock transfer rate setting.
   In this application example, the actual transfer rate becomes approximately 350 kbps since the clock synchronous function is enabled (reference value: pull up voltage 5 V, pull up resistance 1 kΩ).
   Standard-mode and Fast-mode are supported.

Figure 3.1 shows the Communication Format, Figure 3.2 shows the Block Diagram, Figure 3.3 shows the Outline Flowchart, and Figure 3.4 to Figure 3.6 show Timing Diagrams.



ST: Start condition
W̄: Write is 0
ACK: Acknowledge is 0

SP: Stop condition
R: Read is 1
NACK: Not acknowledge is 1

Figure 3.1 Communication Format

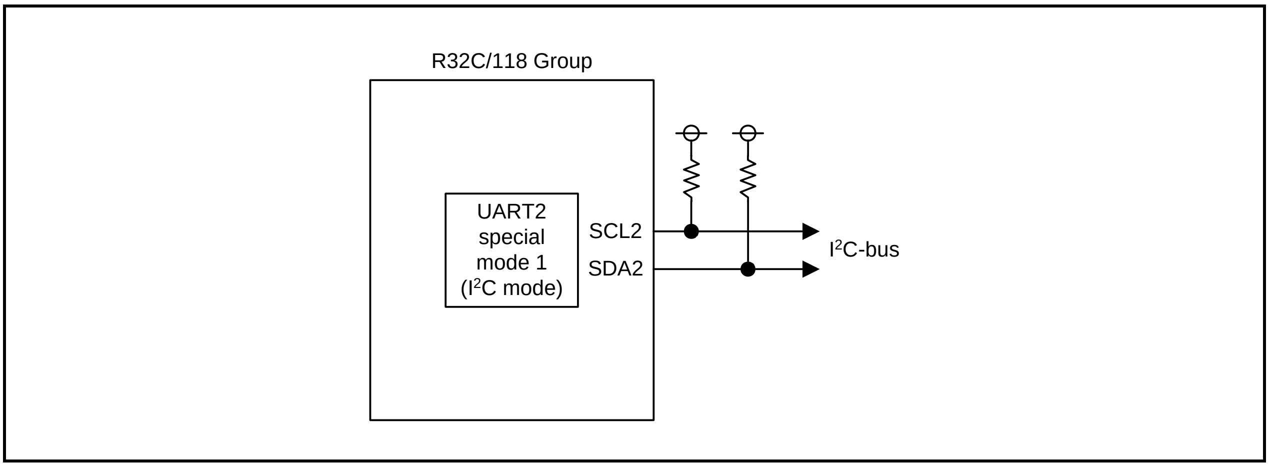


Figure 3.2 Block Diagram

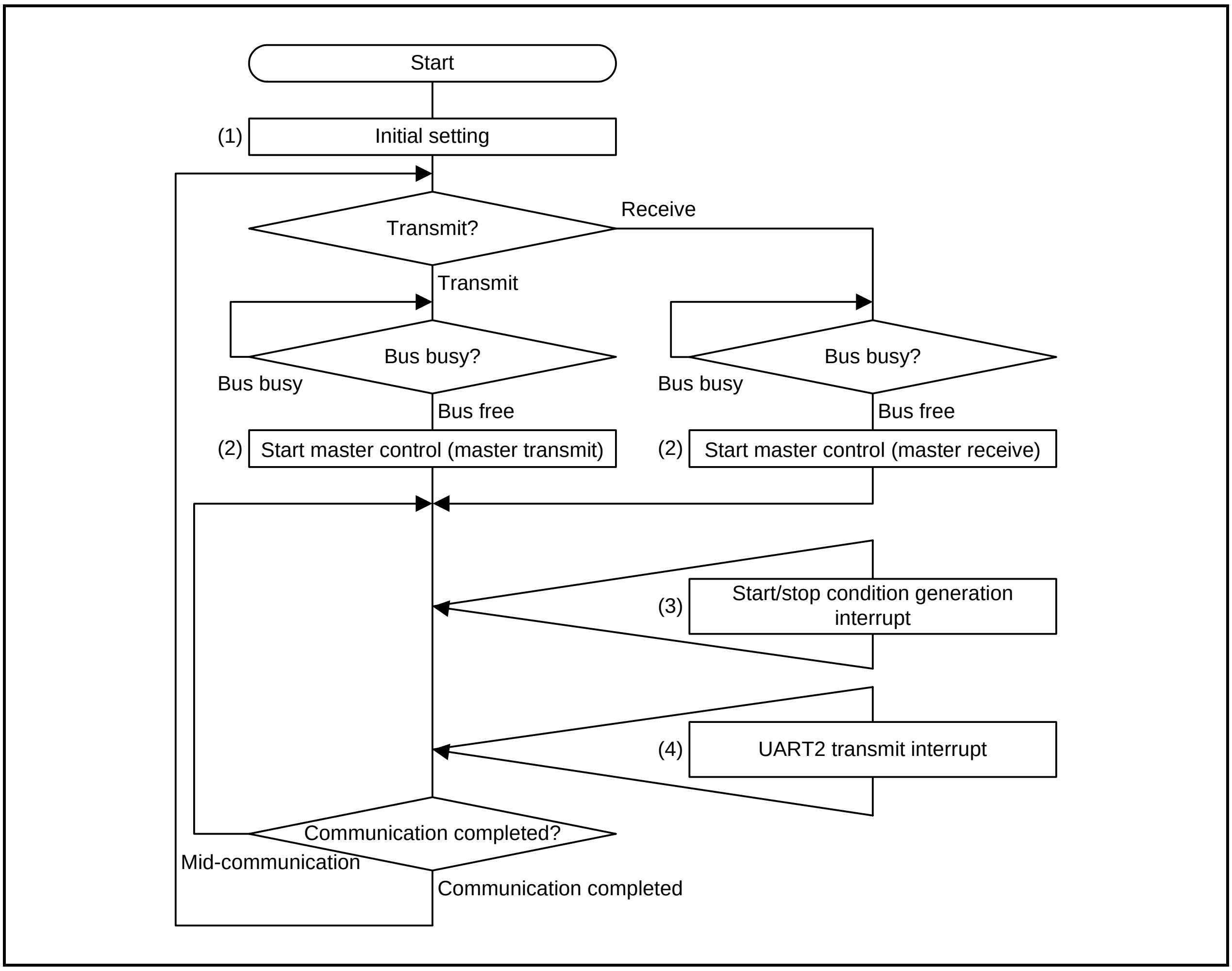


Figure 3.3 Outline Flowchart

The numbers in Figure 3.3 correspond to the numbers indicated in the program processing in the operating timing charts in Figure 3.4 to Figure 3.6.

(1) Initial setting
Initialize the system clock, UART2 associated SFRs, and variables used.

(2) Start master control
Enable the start/stop condition generation interrupt and generate a start condition.

(3) Start/stop condition generation interrupt
An interrupt request is generated when start condition generation is completed and a stop condition is detected. When start condition generation is completed, the UART2 transmit interrupt is enabled and the slave address is transmitted. When a stop condition is detected, SFR values which changed during communication are returned to their initial values.

(4) UART2 transmit interrupt
A UART2 transmit interrupt is generated at the falling edge of the ninth bit of the SCL clock. When transmitting, set the next byte transmit data. When receiving, set ACK/NACK for the next byte. When communication is completed, generate a stop condition.

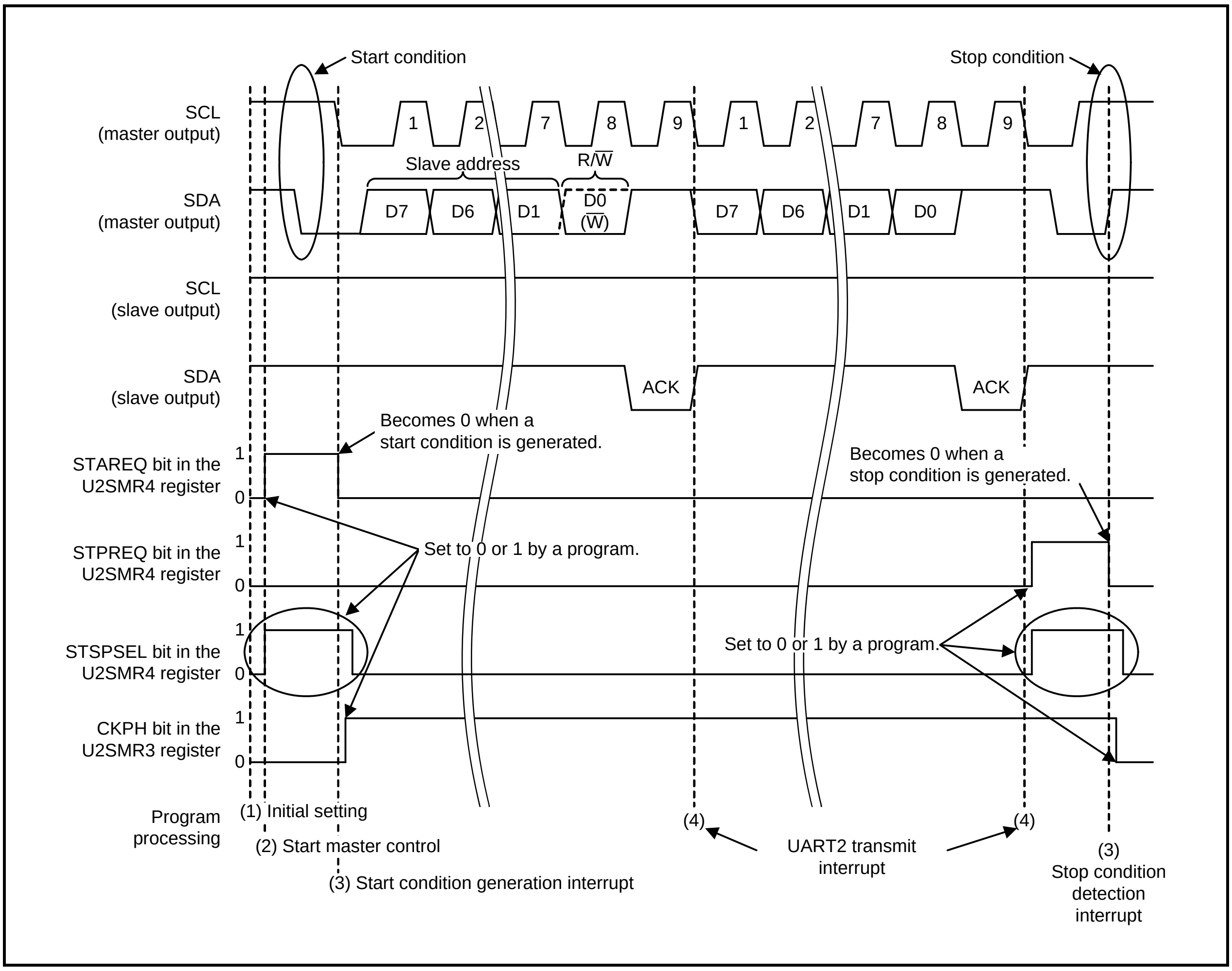


Figure 3.4 Master Transmit Timing

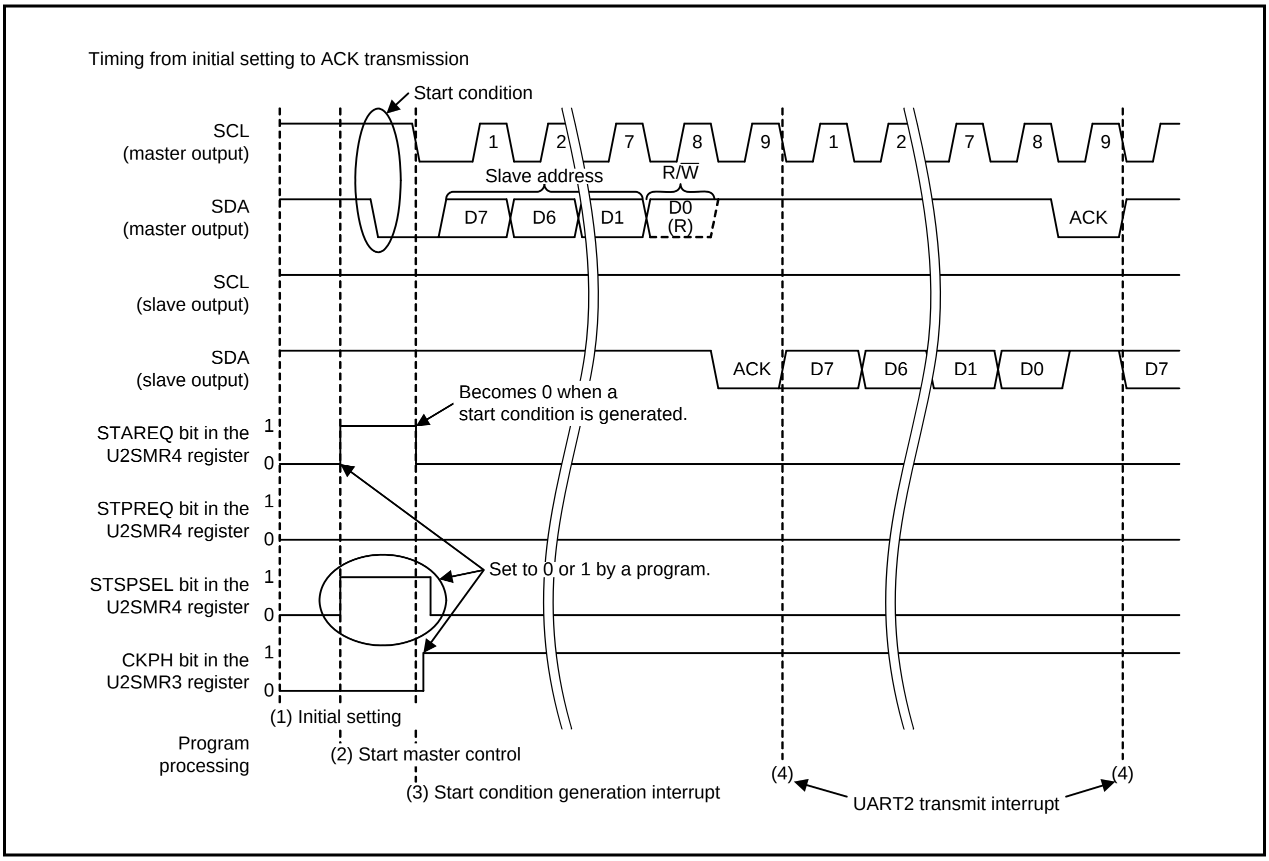

Timing from initial setting to ACK transmission
Start condition
SCL (master output)
1
2
7
8
9
1
2
7
8
9
Slave address
R/W
SDA (master output)
D7
D6
D1
D0 (R)
ACK
SCL (slave output)
SDA (slave output)
ACK
D7
D6
D1
D0
D7
Becomes 0 when a start condition is generated.
STAREQ bit in the U2SMR4 register
STPREQ bit in the U2SMR4 register
STSPSEL bit in the U2SMR4 register
CKPH bit in the U2SMR3 register
1
0
Set to 0 or 1 by a program.
Program processing
(1) Initial setting
(2) Start master control
(3) Start condition generation interrupt
(4)
UART2 transmit interrupt
(4)

Figure 3.5 Master Receive Timing (1)

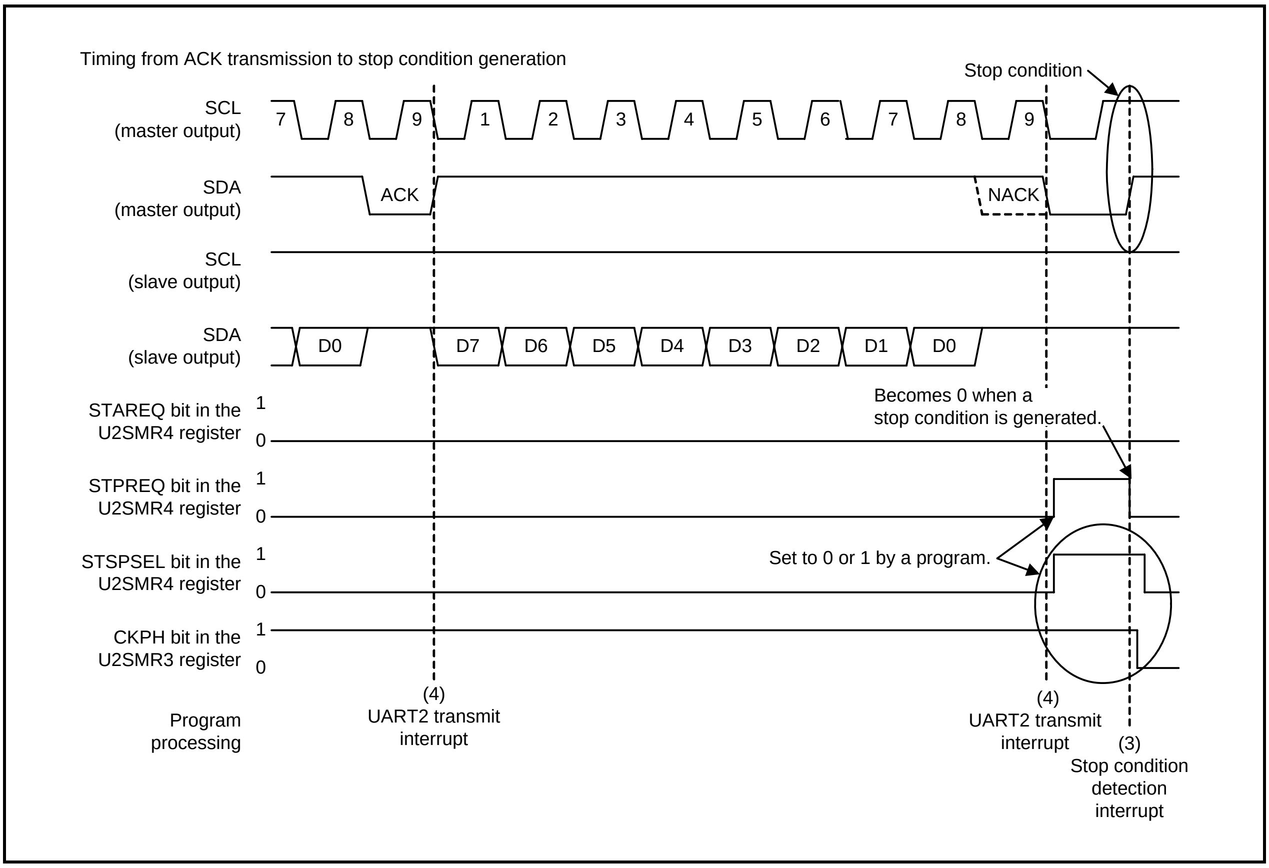

Timing from ACK transmission to stop condition generation
Stop condition
SCL (master output)
7
8
9
1
2
3
4
5
6
7
8
9
SDA (master output)
ACK
NACK
SCL (slave output)
SDA (slave output)
D0
D7
D6
D5
D4
D3
D2
D1
D0
Becomes 0 when a stop condition is generated.
STAREQ bit in the U2SMR4 register
STPREQ bit in the U2SMR4 register
STSPSEL bit in the U2SMR4 register
CKPH bit in the U2SMR3 register
1
0
Set to 0 or 1 by a program.
Program processing
(4) UART2 transmit interrupt
(4) UART2 transmit interrupt
(3) Stop condition detection interrupt

Figure 3.6 Master Receive Timing (2)

3.1.1 Peripheral Functions

Serial interface (UART2) special mode 1 (I^2C mode) is used under the following setting conditions:

- I^2C mode is used.
- Transfer clock is internal clock source.
- f1 is used as U2BRG count source.
- SDA2 and SCL2 pins are N-channel open drain.
- Transfer format uses MSB first.
- Transmission completed (TXEPT is 1) is selected as the UART2 transmit interrupt source.
- Clock phase setting is clock delay.
- Seven to eight cycles of U2BRG count source is selected as SDA2 digital delay value.
- Clock synchronization is enabled.
- SCL2 wait output function 2 is not used.
- SDA2 output disable function is not used.
- Start/stop condition generation interrupt is used.
- UART2 transmit interrupt is used.
- UART2 receive interrupt is not used.
- PLL clock is 100 MHz.
- Base clock is 50 MHz.
- CPU clock is 50 MHz.
- Peripheral bus clock is 25 MHz.
- Peripheral clock is 25 MHz.
- Transfer rate is approximately 378 kbps.

Calculating the transfer rate:
Transfer rate = U2BRG count source / (2 × (U2BRG register setting value + 1))
= 25 MHz (f1) / (2 × (32 + 1))
≈ 378.788 kbps

Table 3.1 Pins Used and Their Function

Pin	I/O	Function
P7_1/SCL2	I/O	I^2C mode clock I/O pin
P7_0/SDA2	I/O	I^2C mode data I/O pin

3.1.2 Notes on Using the Attached Sample Program

Note the following when using the program included with this application note:

- Do not use multiple interrupts.
- When setting the system clock to anything other than the XIN clock (16 MHz), change the setting value of the U2BRG count source and the U2BRG register according to the transfer rate calculation shown in **3.1.1 Peripheral Functions**.

3.2 Memory

Table 3.2 Memory

Memory	Size	Remarks
ROM	496 bytes	In the iic.c module
RAM	8 bytes	In the iic.c module
Maximum user stack	24 bytes	
Maximum interrupt stack	64 bytes	

Usage memory size varies depending on C compiler version and compile options. The above applies under the following conditions:

C compiler: R32C/100 Series C Compiler V.1.02 Release 01

Compile option: -c -finfo -dir “$(CONFIGDIR)”

4. Software

This chapter shows the program example to set the example described in chapter 3. Application Example. Refer to the latest hardware user’s manual for details on individual registers.

4.1 Variables

Definition file name: rej05b1395.c

Variable Name	Size	Description
unsigned char iic_tx[BUFSIZE]	255 bytes	Transmit buffer
unsigned char iic_rx[BUFSIZE]	255 bytes	Receive buffer
unsigned char retry_counter	1 byte	Count number of communication retries

Definition file name: iic.c

Variable Name		Size /Bit Number	Description
static byte_dt iic_str1		-	Structure to store slave address
Structure member	iic_slave_addr	1 byte	Slave address
	iic_rw	b0	R/$\overline{\text{W}}$ flag 0: Write ($\overline{\text{W}}$) 1: Read (R)
	-	b7 to b1	7-bit address
static byte_dt iic_str2		-	Structure to store status
Structure member	iic_status	1 byte	All statuses
	iic_start	b0	Mid-communication flag 0: Communication completed 1: Mid-communication
	iic_err_par	b1	Parameter error flag 0: No error 1: Parameter error
	iic_err_nack	b2	NACK detection error flag 0: No error 1: NACK detection error
	iic_err_addr	b3	No address match error flag 0: No error 1: No address match error
	-	b7 to b4	Not used (undefined)
unsigned char iic_length		1 byte	Transfer data length
unsigned char iic_index		1 byte	Number of transmit/receive bytes
unsigned char far *iic_pointer		4 bytes	Transmit/receive buffer pointer

4.2 Function Tables

Declaration	void main (void)		
Outline	Main processing		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	unsigned char iic_tx[BUFSIZE]		Transmit buffer
	unsigned char iic_rx[BUFSIZE]		Receive buffer
	unsigned char retry_counter		Count number of communication retries
Returned value	Type	Value	Meaning
	None	-	-
Function	After initializing the system clock and UART2, master transmission and reception are repeated alternately. After calling the iic_master_start function to start master control, call the iic_master_end function and wait for master control to be completed. When the iic_master_end function returns ADD_ERR (communication stop because of address not matched error), communication is retried.		

Declaration	void SetPLLClock (void)		
Outline	PLL mode setting		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	None		-
Returned value	Type	Value	Meaning
	None	-	-
Function	Call this function from the main processing. Process for transition to PLL mode. Set the peripheral clock source to 25 MHz.		

Declaration	void uart2_init (unsigned char ini)		
Outline	UART2 initial setting		
Argument	Argument name	Meaning	
	unsigned char ini	0: I^{2}C mode disabled 1: I^{2}C mode enabled	
Variable (global)	Variable name		Contents
	None		-
Returned value	Type	Value	Meaning
	None	-	-
Function	Call this function from the main processing. Initialize the SFRs used for UART2 in special mode 1 (I^{2}C mode).		

Declaration	unsigned char iic_master_start (unsigned char addr, unsigned char rw, unsigned char far *buf, unsigned char len)		
Outline	Master control start processing		
Argument	Argument name	Meaning	
	unsigned char addr	0x00 to 0x7F: Specify slave address	
	unsigned char rw	0x00: Master transmit 0x01: Master receive	
	unsigned char far *buf	Transmit or receive buffer pointer	
	unsigned char len	0x01 to 0xFF: Transfer data length	
Variable (global)	Variable name	Contents	
	(structure member) iic_status	All statuses	
	(structure member) iic_start	Mid-communication flag	
	(structure member) iic_err_par	Parameter error flag	
	(structure member) iic_slave_addr	Slave address	
	unsigned char iic_length	Transfer data length	
	unsigned char far *iic_pointer	Transmit/receive buffer pointer	
Returned value	Type	Value	Meaning
	unsigned char	BUSY	Bus busy
		RDY	Bus free
		PAR_ERR	Parameter error
Function	This function is called by the main function to perform master control start processing. Before executing this function, execute the uart2_init function to enable I^{2}C mode. In the function header, all statuses are initialized and argument parameters are checked. If any parameter value is invalid, the parameter error flag is set to 1 and PAR_ERR is returned. Master control start processing is not performed when a parameter error is detected. Next, the bus status is checked. • When the bus is busy, the returned value is BUSY and master control start processing is not performed. • When the bus is free, the returned value is RDY and master control start processing is performed. Set the mid-communication flag to 1 and a start condition is generated.		

Declaration	void _start_stop_condition_detection (void)		
Outline	Start/stop condition generation interrupt handling		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	None		-
Returned value	Type	Value	Meaning
	None	-	-
Function	An interrupt is generated when the start condition generation is completed and a stop condition is detected. The sta_int function is called when the start condition generation is completed. The stp_int function is called when a stop condition is detected.		

Declaration	static void sta_int (void)		
Outline	Start condition detection processing		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	(structure member) iic_slave_addr		Slave address
	unsigned char iic_index		Number of transmit/receive bytes
Returned value	Type	Value	Meaning
	None	-	-
Function	Called from the start/stop condition generation interrupt handling. UART2 transmit/receive interrupt is enabled. Transmit the slave address.		

Declaration	static void stp_int (void)		
Outline	Stop condition detection processing		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	unsigned char iic_index		Number of transmit/receive bytes
	(structure member) iic_start		Mid-communication flag
Returned value	Type	Value	Meaning
	None	-	-
Function	Called from the start/stop condition generation interrupt handling. SFR associated with UART2 values which changed during communication are returned to their values, and the mid-communication flag is set to 0.		

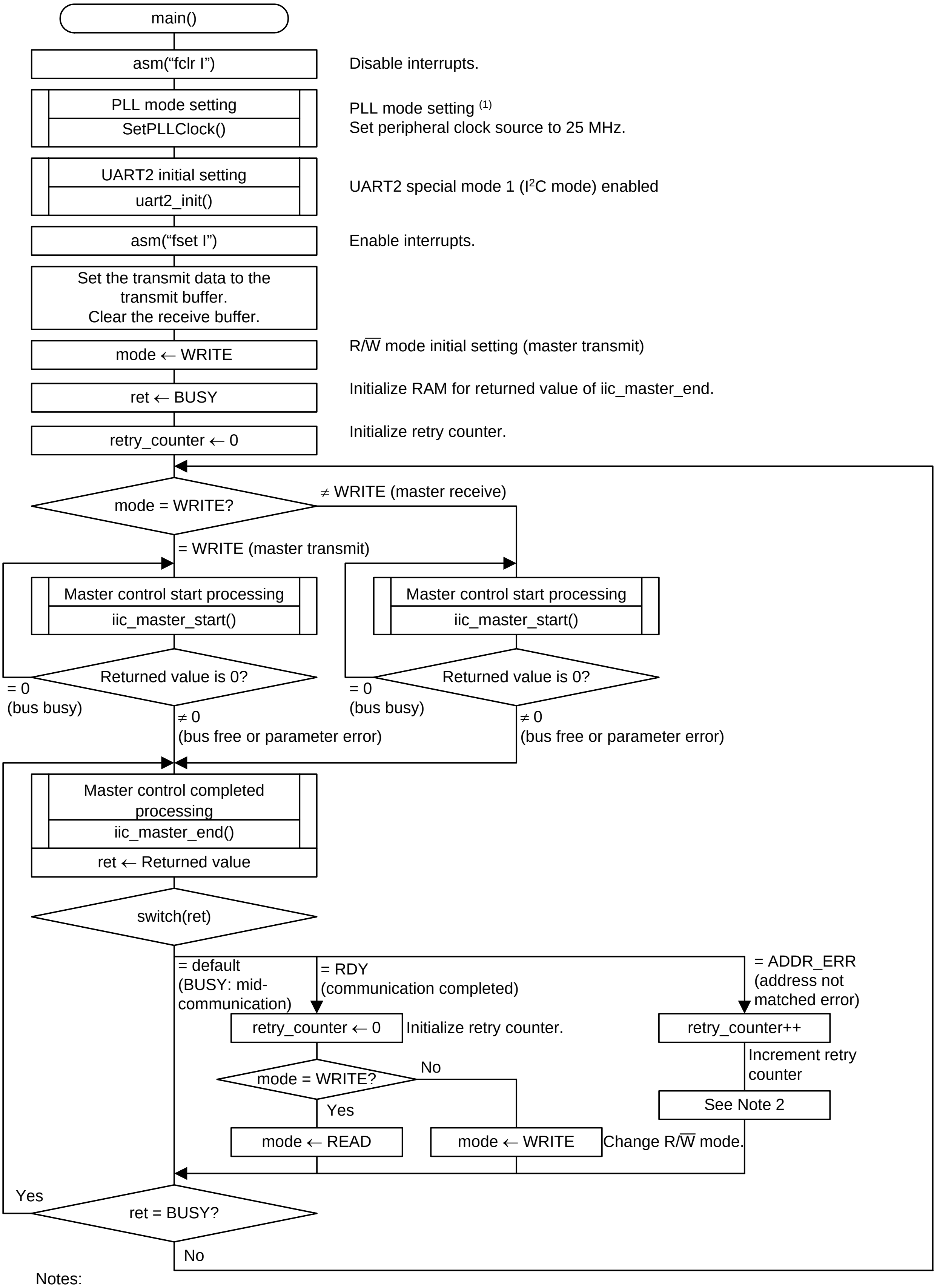
Declaration	void _uart2_trans (void)		
Outline	UART2 transmit interrupt handling		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	unsigned char iic_index		Number of transmit/receive bytes
	(structure member) iic_err_addr		No address match error flag
	(structure member) iic_rw		$R/\overline{W}$ flag
Returned value	Type	Value	Meaning
	None	-	-
Function	An interrupt is generated at the falling edge of the ninth bit of the SCL clock. Read the U2RB register in the function header. When a NACK is detected during slave address transmission, set the no address match error flag to 1. At all other times, the master_trn_int function is called during master transmission and the master_rcv_int function is called during master reception. When communication is completed, generate a stop condition.		

Declaration	static unsigned char master_trn_int (unsigned short rb_data)		
Outline	Master transmit processing		
Argument	Argument name		Meaning
	unsigned short rb_data		Data read from the U2RB register
Variable (global)	Variable name		Contents
	(structure member) iic_err_nack		NACK detection error flag
	unsigned char iic_index		Number of transmit/receive bytes
	unsigned char iic_length		Transfer data length
	unsigned char far *iic_pointer		Transmit/receive buffer pointer
Returned value	Type	Value	Meaning
	unsigned char	IIC_SP_ON	0: Stop condition generated
		IIC_SP_OFF	1: Stop condition not generated
Function	Called from the UART2 transmit interrupt handling. IIC_SP_OFF is returned in the following case: • ACK is detected and not the last byte (starts the next transmission). IIC_SP_ON is returned in the following cases: • NACK is detected (NACK detect error flag is set to 1). • When the last byte transmission is completed.		

Declaration	static unsigned char master_rcv_int (unsigned short rb_data)		
Outline	Master receive processing		
Argument	Argument name		Meaning
	unsigned short rb_data		Data read from the U2RB register
Variable (global)	Variable name		Contents
	unsigned char iic_index		Number of transmit/receive bytes
	unsigned char iic_length		Transfer data length
	unsigned char far *iic_pointer		Transmit/receive buffer pointer
Returned value	Type	Value	Meaning
	unsigned char	IIC_SP_ON	0: Stop condition generated
		IIC_SP_OFF	1: Stop condition not generated
Function	Called from the UART2 transmit interrupt handling. The argument value is stored in the receive buffer (except for the slave address data). NACK is set to the transmit register when the following data is the last byte. ACK is set to the transmit register when the following data is not the last byte. After setting ACK or NACK to the transmit register, the next receive operation starts. IIC_SP_OFF is returned in the following case: • The following data is not the last byte data. IIC_SP_ON is returned in the following case: • The last byte receive operation is completed.		

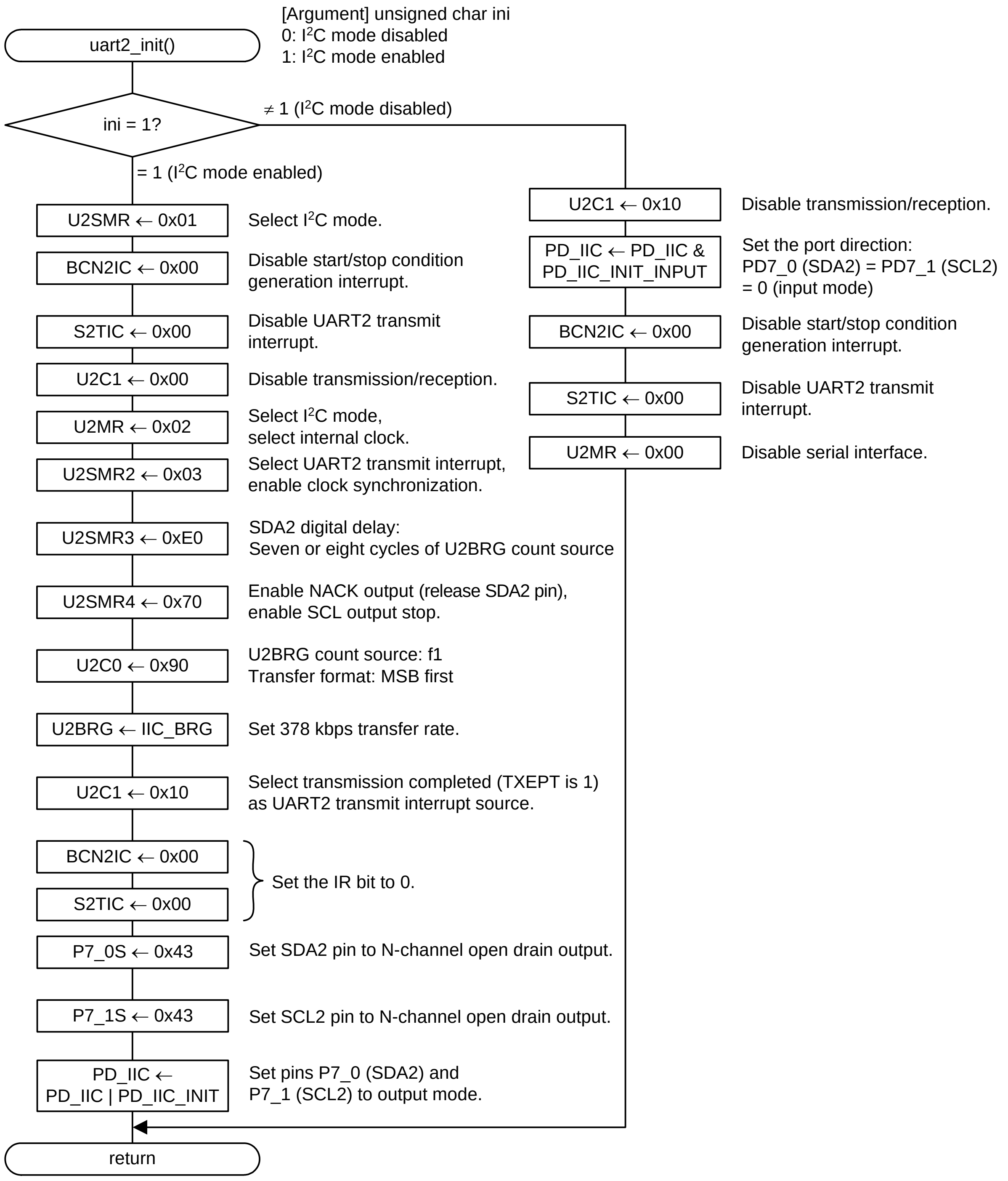
Declaration	unsigned char iic_master_end (void)		
Outline	Master control completed processing		
Argument	Argument name		Meaning
	None		-
Variable (global)	Variable name		Contents
	(structure member) iic_status		All statuses
	(structure member) iic_start		Mid-communication flag
	(structure member) iic_err_par		Parameter error flag
	(structure member) iic_err_nack		NACK detection error flag
	(structure member) iic_err_addr		No address match error flag
Returned value	Type	Value	Meaning
	unsigned char	BUSY	Mid-communication
		RDY	Communication completed
		ADDR_ERR	Address not matched
Function	Called from the main function. Informs the user of the master control state completed. During communication, this function returns BUSY. When communication is completed, this function returns RDY. Additional processing after communication is completed can be added as needed.		

4.3 Main Processing

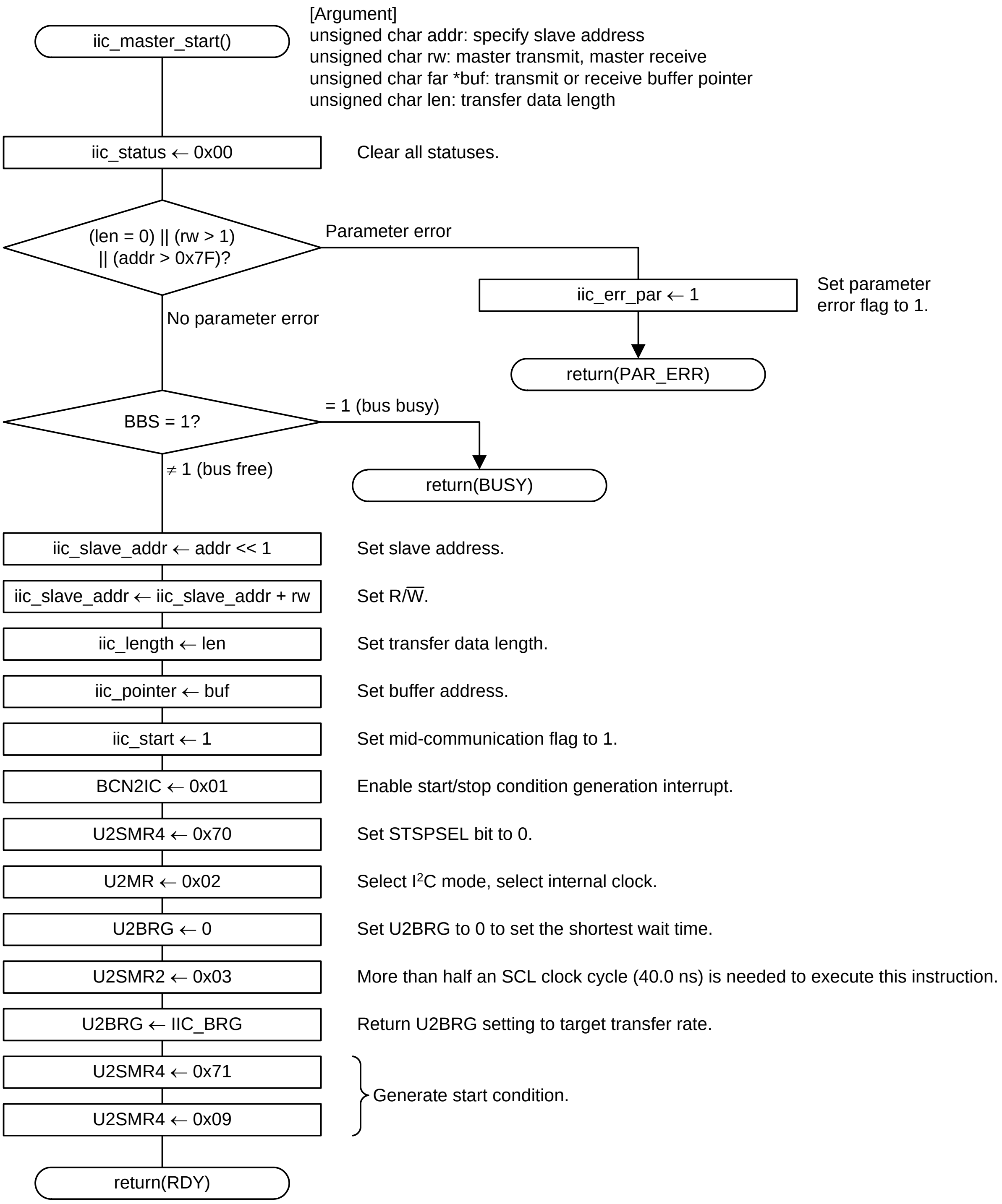

main()
asm("fclr I")
Disable interrupts.
PLL mode setting
SetPLLClock()
PLL mode setting (1)
Set peripheral clock source to 25 MHz.
UART2 initial setting
uart2_init()
UART2 special mode 1 (I²C mode) enabled
asm("fset I")
Enable interrupts.
Set the transmit data to the transmit buffer.
Clear the receive buffer.
mode ← WRITE
R/W mode initial setting (master transmit)
ret ← BUSY
Initialize RAM for returned value of iic_master_end.
retry_counter ← 0
Initialize retry counter.
mode = WRITE?
≠ WRITE (master receive)
= WRITE (master transmit)
Master control start processing
iic_master_start()
Returned value is 0?
= 0
(bus busy)
≠ 0
(bus free or parameter error)
Master control start processing
iic_master_start()
Returned value is 0?
= 0
(bus busy)
≠ 0
(bus free or parameter error)
Master control completed processing
iic_master_end()
ret ← Returned value
switch(ret)
= default
(BUSY: mid-communication)
= RDY
(communication completed)
= ADDR_ERR
(address not matched error)
retry_counter ← 0
Initialize retry counter.
retry_counter++
Increment retry counter
mode = WRITE?
No
Yes
See Note 2
mode ← READ
mode ← WRITE
Change R/W mode.
Yes
ret = BUSY?
No

Notes:
1. Refer to the hardware user's manual.
2. Additional processing can be added as needed.

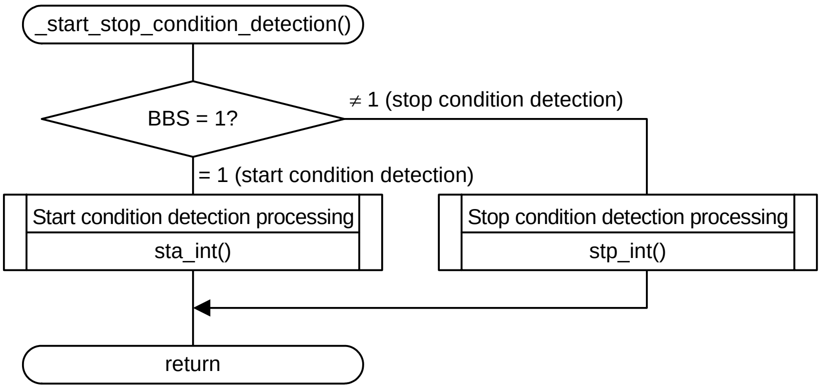
4.4 UART2 Initial Setting


uart2_init()
[Argument] unsigned char ini
0: I2C mode disabled
1: I2C mode enabled
ini = 1?
≠ 1 (I2C mode disabled)
= 1 (I2C mode enabled)
U2SMR ← 0x01
Select I2C mode.
BCN2IC ← 0x00
Disable start/stop condition generation interrupt.
S2TIC ← 0x00
Disable UART2 transmit interrupt.
U2C1 ← 0x00
Disable transmission/reception.
U2MR ← 0x02
Select I2C mode, select internal clock.
U2SMR2 ← 0x03
Select UART2 transmit interrupt, enable clock synchronization.
U2SMR3 ← 0xE0
SDA2 digital delay: Seven or eight cycles of U2BRG count source
U2SMR4 ← 0x70
Enable NACK output (release SDA2 pin), enable SCL output stop.
U2C0 ← 0x90
U2BRG count source: f1
Transfer format: MSB first
U2BRG ← IIC_BRG
Set 378 kbps transfer rate.
U2C1 ← 0x10
Select transmission completed (TXEPT is 1) as UART2 transmit interrupt source.
BCN2IC ← 0x00
S2TIC ← 0x00
Set the IR bit to 0.
P7_0S ← 0x43
Set SDA2 pin to N-channel open drain output.
P7_1S ← 0x43
Set SCL2 pin to N-channel open drain output.
PD_IIC ← PD_IIC | PD_IIC_INIT
Set pins P7_0 (SDA2) and P7_1 (SCL2) to output mode.
U2C1 ← 0x10
Disable transmission/reception.
PD_IIC ← PD_IIC & PD_IIC_INIT_INPUT
Set the port direction: PD7_0 (SDA2) = PD7_1 (SCL2) = 0 (input mode)
BCN2IC ← 0x00
Disable start/stop condition generation interrupt.
S2TIC ← 0x00
Disable UART2 transmit interrupt.
U2MR ← 0x00
Disable serial interface.
return

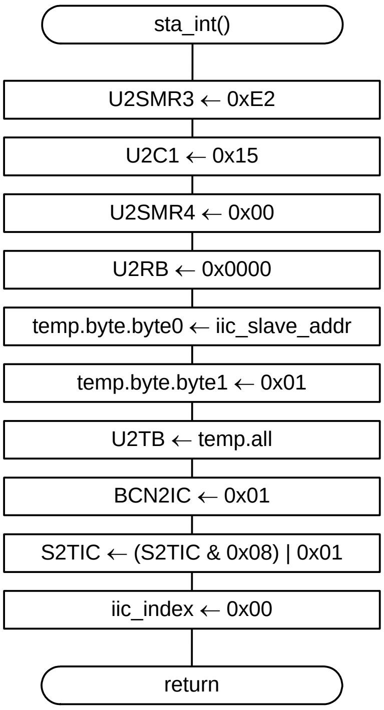
4.5 Master Control Start Processing


iic_master_start()
[Argument]
unsigned char addr: specify slave address
unsigned char rw: master transmit, master receive
unsigned char far *buf: transmit or receive buffer pointer
unsigned char len: transfer data length
iic_status ← 0x00
Clear all statuses.
(len = 0) || (rw > 1) || (addr > 0x7F)?
Parameter error
iic_err_par ← 1
Set parameter error flag to 1.
return(PAR_ERR)
No parameter error
BBS = 1?
= 1 (bus busy)
return(BUSY)
≠ 1 (bus free)
iic_slave_addr ← addr << 1
Set slave address.
iic_slave_addr ← iic_slave_addr + rw
Set R/W̄.
iic_length ← len
Set transfer data length.
iic_pointer ← buf
Set buffer address.
iic_start ← 1
Set mid-communication flag to 1.
BCN2IC ← 0x01
Enable start/stop condition generation interrupt.
U2SMR4 ← 0x70
Set STSPSEL bit to 0.
U2MR ← 0x02
Select I2C mode, select internal clock.
U2BRG ← 0
Set U2BRG to 0 to set the shortest wait time.
U2SMR2 ← 0x03
More than half an SCL clock cycle (40.0 ns) is needed to execute this instruction.
U2BRG ← IIC_BRG
Return U2BRG setting to target transfer rate.
U2SMR4 ← 0x71
U2SMR4 ← 0x09
Generate start condition.
return(RDY)

4.6 Start/Stop Condition Generation Interrupt Handling


_start_stop_condition_detection()
BBS = 1?
≠ 1 (stop condition detection)
= 1 (start condition detection)
Start condition detection processing
sta_int()
Stop condition detection processing
stp_int()
return

4.7 Start Condition Detection Processing



Clock phase: With clock delay

Enable transmission/reception.

Do not generate start/stop conditions.

Clear ABT bit.

Set slave address.

Set data to release SDA2 pin at the ninth bit.

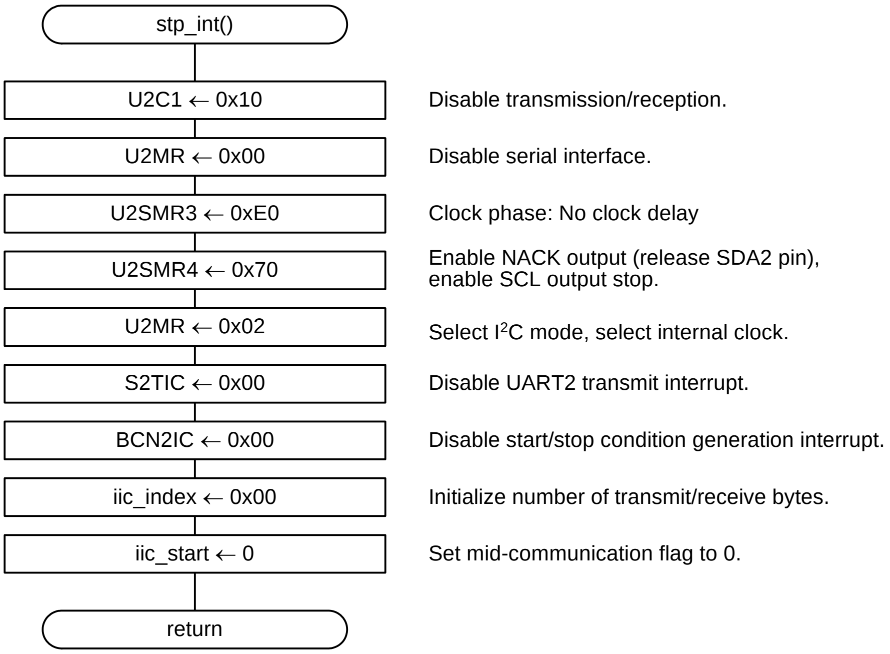
Start first byte (slave address) transmission.

Set IR bit to 0 after changing CKPH bit.

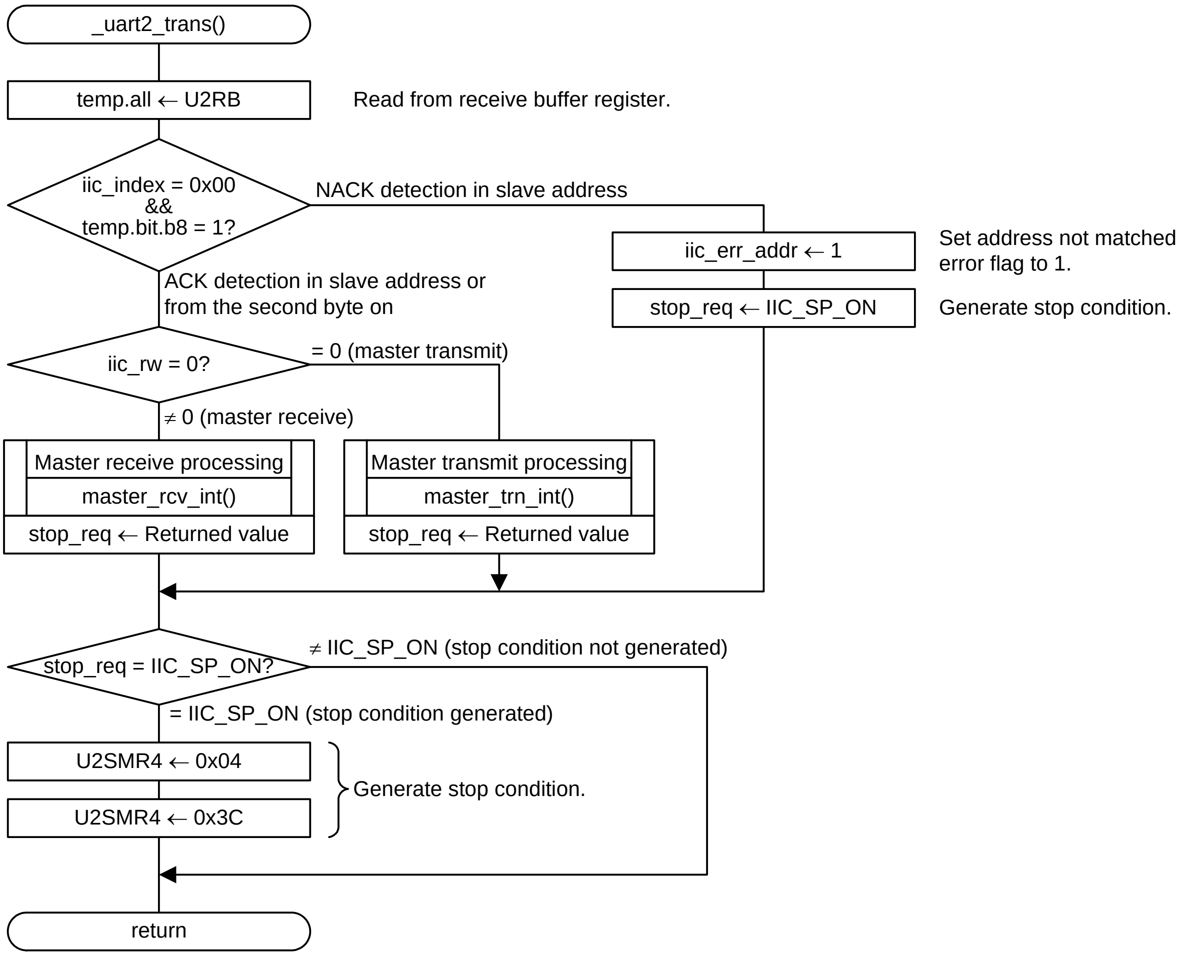
Enable UART2 transmit interrupt.

Initialize number of transmit/receive bytes.

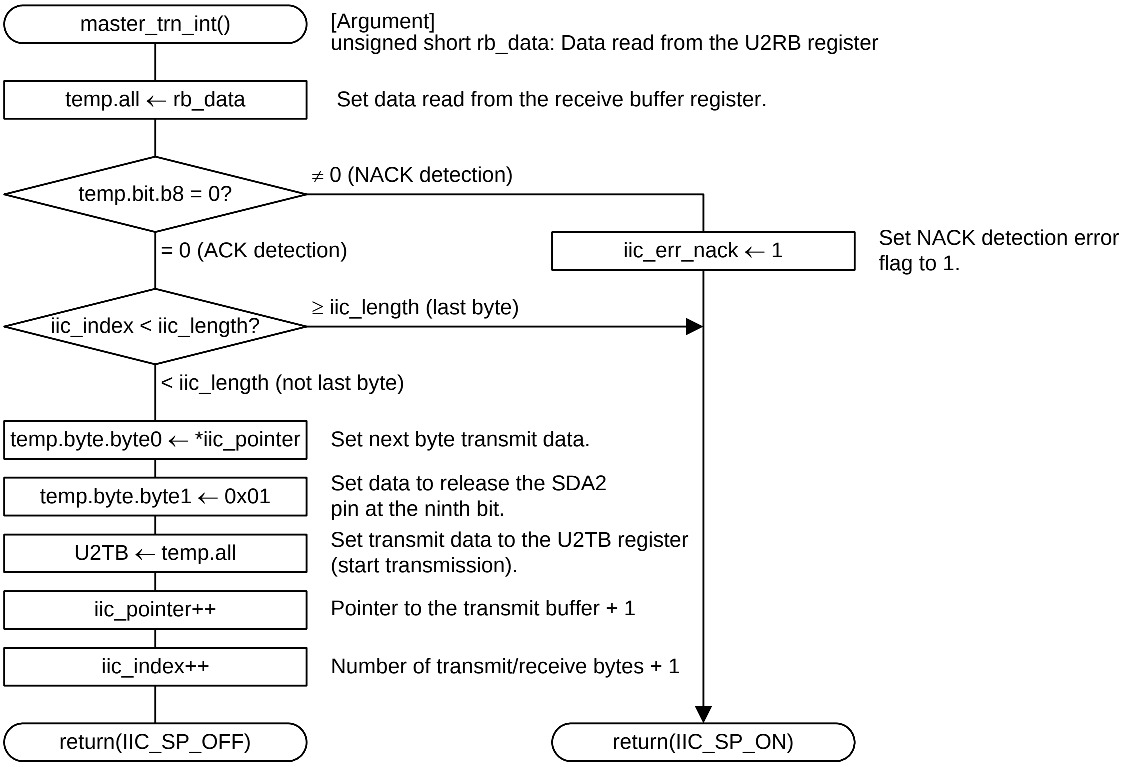
4.8 Stop Condition Detection Processing


stp_int()
U2C1 ← 0x10
Disable transmission/reception.
U2MR ← 0x00
Disable serial interface.
U2SMR3 ← 0xE0
Clock phase: No clock delay
U2SMR4 ← 0x70
Enable NACK output (release SDA2 pin), enable SCL output stop.
U2MR ← 0x02
Select I²C mode, select internal clock.
S2TIC ← 0x00
Disable UART2 transmit interrupt.
BCN2IC ← 0x00
Disable start/stop condition generation interrupt.
iic_index ← 0x00
Initialize number of transmit/receive bytes.
iic_start ← 0
Set mid-communication flag to 0.
return

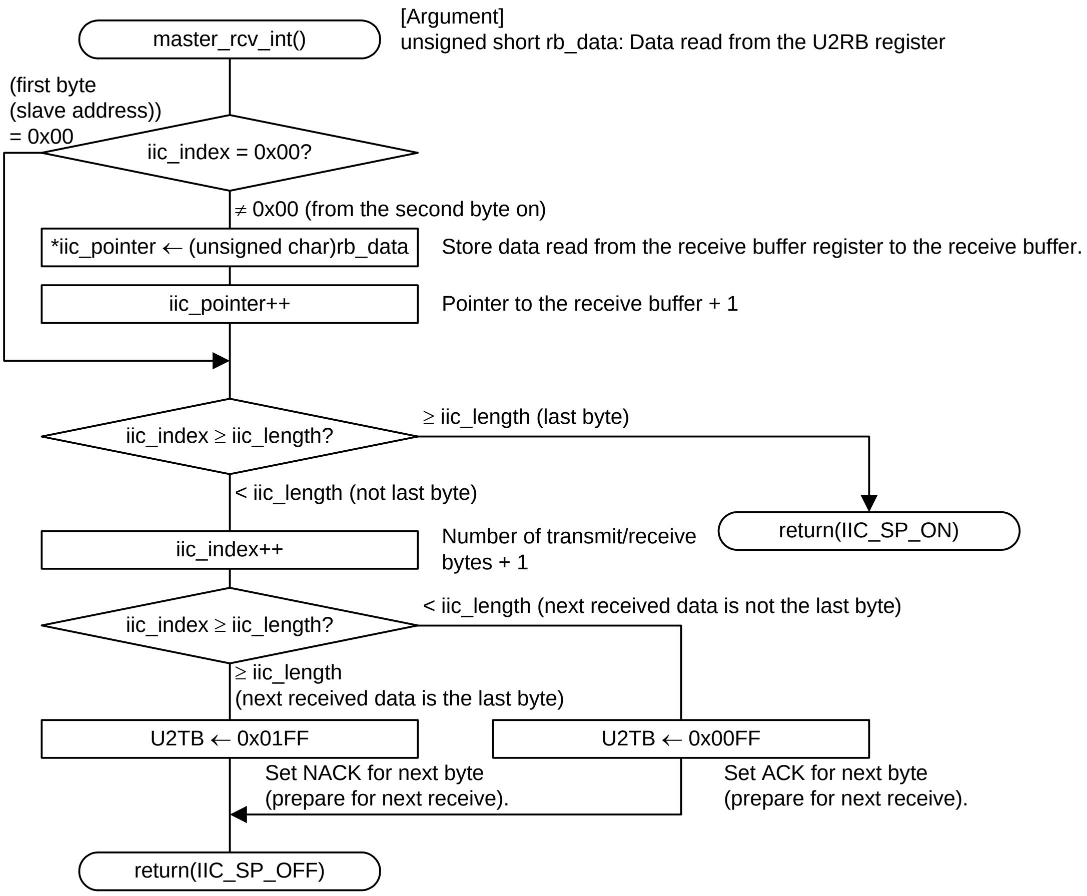
4.9 UART2 Transmit Interrupt Handling


_uart2_trans()
temp.all ← U2RB
Read from receive buffer register.
iic_index = 0x00
&&
temp.bit.b8 = 1?
NACK detection in slave address
iic_err_addr ← 1
Set address not matched error flag to 1.
stop_req ← IIC_SP_ON
Generate stop condition.
ACK detection in slave address or from the second byte on
iic_rw = 0?
= 0 (master transmit)
≠ 0 (master receive)
Master receive processing
master_rcv_int()
stop_req ← Returned value
Master transmit processing
master_trn_int()
stop_req ← Returned value
stop_req = IIC_SP_ON?
≠ IIC_SP_ON (stop condition not generated)
= IIC_SP_ON (stop condition generated)
U2SMR4 ← 0x04
U2SMR4 ← 0x3C
Generate stop condition.
return

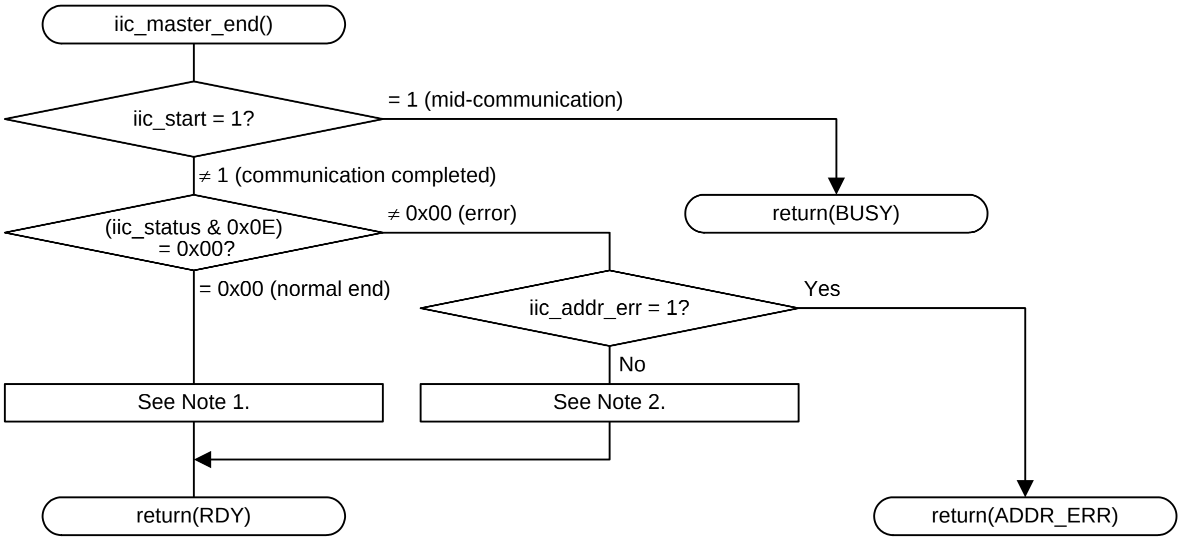
4.10 Master Transmit Processing


master_trn_int()
[Argument]
unsigned short rb_data: Data read from the U2RB register
temp.all ← rb_data
Set data read from the receive buffer register.
temp.bit.b8 = 0?
≠ 0 (NACK detection)
iic_err_nack ← 1
Set NACK detection error flag to 1.
= 0 (ACK detection)
iic_index < iic_length?
≥ iic_length (last byte)
< iic_length (not last byte)
temp.byte.byte0 ← *iic_pointer
Set next byte transmit data.
temp.byte.byte1 ← 0x01
Set data to release the SDA2 pin at the ninth bit.
U2TB ← temp.all
Set transmit data to the U2TB register (start transmission).
iic_pointer++
Pointer to the transmit buffer + 1
iic_index++
Number of transmit/receive bytes + 1
return(IIC_SP_OFF)
return(IIC_SP_ON)

4.11 Master Receive Processing


master_rcv_int()
[Argument]
unsigned short rb_data: Data read from the U2RB register
(first byte
(slave address))
= 0x00
iic_index = 0x00?
≠ 0x00 (from the second byte on)
*iic_pointer ← (unsigned char)rb_data
Store data read from the receive buffer register to the receive buffer.
iic_pointer++
Pointer to the receive buffer + 1
iic_index ≥ iic_length?
≥ iic_length (last byte)
< iic_length (not last byte)
return(IIC_SP_ON)
iic_index++
Number of transmit/receive
bytes + 1
iic_index ≥ iic_length?
< iic_length (next received data is not the last byte)
≥ iic_length
(next received data is the last byte)
U2TB ← 0x01FF
U2TB ← 0x00FF
Set NACK for next byte
(prepare for next receive).
Set ACK for next byte
(prepare for next receive).
return(IIC_SP_OFF)

4.12 Master Control Completed Processing



Notes:
1. Additional processing of communication completed normally can be added as needed.
2. Additional processing of communication completed with error can be added as needed.

5. Sample Program

A sample program can be downloaded from the Renesas Electronics website.

6. Reference Documents

R32C/118 Group User's Manual: Hardware Rev.1.00
The latest version can be downloaded from the Renesas Electronics website.

Technical News/Technical Update
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
R32C/100 Series C Compiler Package V.1.02
C Compiler User's Manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision History	R32C/100 Series I^{2}C-bus Interface Using UARTi Special Mode 1 (Master Transmit/Receive)

Rev.	Date	Description	
		Page	Summary
1.00	Aug 31, 2010	—	First edition issued

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 part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

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

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

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, 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.
7F, 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 1.0