

## RL78/F13, F14, F15

### Setup Procedure for LIN Communication in Master Mode (Guidance)

---

#### Introduction

This application note describes setup procedures for LIN communication, in which a device operates in master mode of the UART/LIN module (hereinafter, RLIN3) in the RL78/F13, F14, F15.

Under certain use conditions, the operations of the microcontroller might be different from examples that this document provides. Customers are required to sufficiently evaluate the use of the RLIN3 in their environment. Customers are also required to refer to the user's manual corresponding to their products for detailed functions of the RLIN3, clock generator, and interrupts.

#### Target Devices

- RL78/F13, F14, F15 products

## Contents

<b>1. Specifications of LIN Communication in Master Mode</b> .....	<b>3</b>
1.1 LIN Master Mode Usage Conditions .....	3
1.2 LIN Network Connection Example .....	4
1.3 SFR .....	5
<b>2. Overview of LIN Master Communication</b> .....	<b>7</b>
2.1 LIN Master Communication Processing Overview .....	7
2.2 LIN Frame Timing (frame separate mode) .....	9
2.3 LIN Master Communication Status and SFR .....	10
<b>3. LIN Master Driver Functions</b> .....	<b>11</b>
3.1 Function list .....	11
3.2 Driver API Functions [Public] .....	13
3.2.1 LIN Master Module Initialization .....	13
3.2.2 LIN Module De-Initialization .....	14
3.2.3 LIN Master Header Transmission .....	15
3.2.4 Before Sleep Mode Transition Processing .....	16
3.2.5 LIN Master Wake-up Signal Transmission .....	17
3.3 Driver Local Functions [Private] .....	18
3.3.1 LIN0 Port Initialization .....	18
3.3.2 LIN0 Interrupt Initialization .....	19
3.3.3 Transition to LIN Reset Mode .....	20
3.3.4 Transition to LIN Operation Mode .....	21
3.3.5 Transition to LIN Wake-up Mode .....	22
3.4 Interrupt Processing Functions .....	23
3.4.1 LIN Master Transmission Completion Interrupt Processing .....	23
3.4.2 LIN Master Reception Completion Interrupt Processing .....	25
3.4.3 LIN Master Status Error Interrupt Processing .....	26
3.4.4 LIN Master Wake-up Signal Detection Interrupt Processing .....	27
3.5 User Notification Function APIs .....	28
3.5.1 Error Notification .....	28
3.5.2 Header Transmission Completion Notification .....	29
3.5.3 Response Transmission Completion Notification .....	30
3.5.4 Response Reception Completion Notification .....	31
3.5.5 Wake-up Signal Transmission Completion Notification .....	32
3.5.6 Wake-up Signal Detection Notification .....	33
<b>4. Operation Mode Transition of LIN / UART Module (RLIN3)</b> .....	<b>34</b>
<b>5. References</b> .....	<b>35</b>
<b>6. Appendix</b> .....	<b>36</b>
<b>Revision History</b> .....	<b>37</b>

## 1. Specifications of LIN Communication in Master Mode

### 1.1 LIN Master Mode Usage Conditions

The conditions for use of LIN communication in master mode that this application note describes are shown below.

**Table 1-1 LIN Master Mode Usage Conditions**

Item		Description
CPU/peripheral hardware clock frequency (f <sub>CLK</sub> )		32 MHz
LIN clock source (f <sub>LIN</sub> )		f <sub>CLK</sub>
LTXD0		P13
LRXD0		P14
Baud rate		19200 bps
Mode		LIN master mode
Noise filter		Enabled
LIN interrupts		INTLIN0TRM, INTLIN0RVC, INTLIN0STA, INTLIN0WUP
Checksum type		Classic checksum or Enhanced checksum (User selected)
Error detection	Bit error	Enabled
	Physical bus error	Enabled
	Timeout error	Enabled
	Framing error	Enabled
	Checksum error	Enabled
	Response preparation error	Enabled
Number of bytes in frame's data field		0 to 8 bytes
Response space		0 Tbit
Inter-byte space		1 Tbit

## 1.2 LIN Network Connection Example

The LIN network connection is shown below.

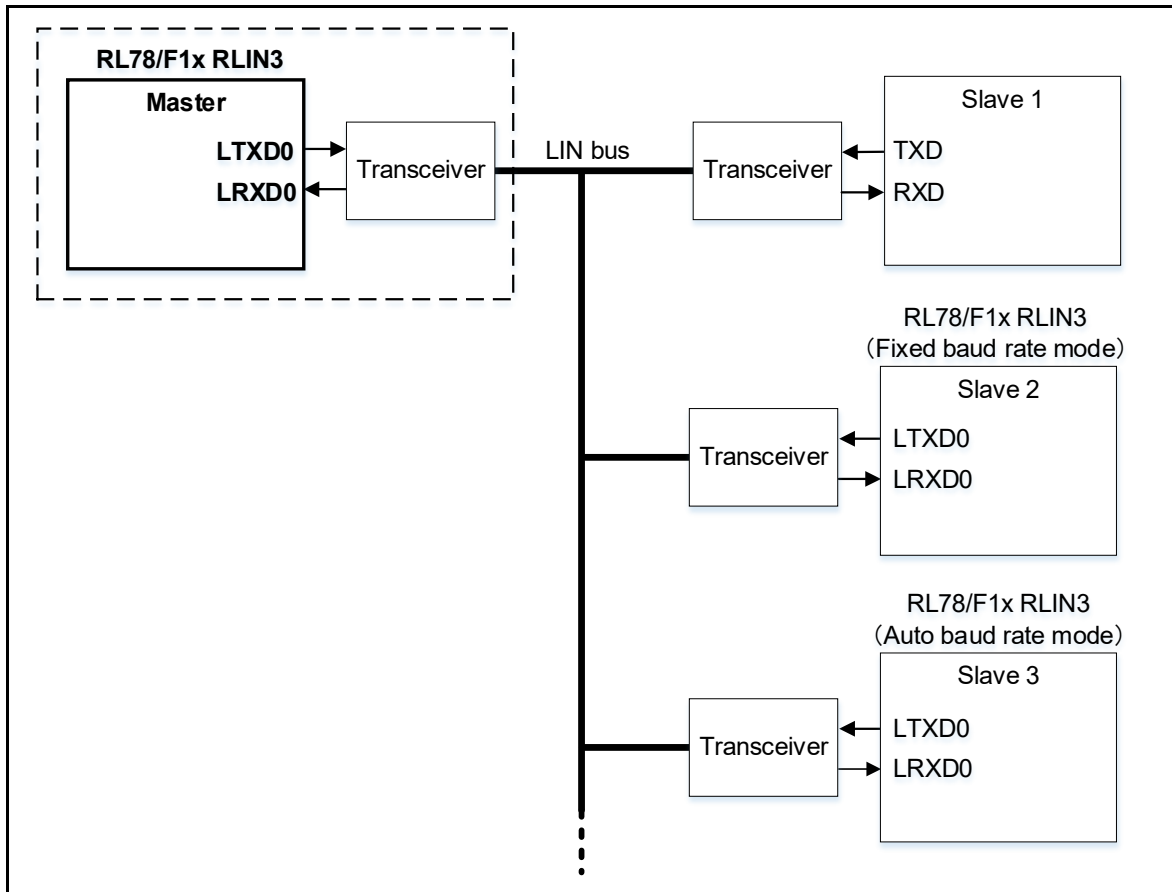


Figure 1-1 LIN Master - Slaves Network Connection (example)

### 1.3 SFR

The setting of the SFR used for LIN master communication is shown below.

**Table 1-2 SFRs Used for LIN Communication in Master Mode (1/2)**

Register		Setting Value	Conditions
Name	Symbol		
Peripheral enable register 2	PER2	LIN0EN = 0 or 1	0: Stop LIN0 clock supply, 1: LIN0 clock supply
LIN clock select register	LINCKSEL	10H	LIN0MCK=0 (LIN0 communication clock = $f_{CLK}$ ) LIN0MCKE=1 (LIN0 communication clock supply)
LIN channel select register	LCHSEL	00H	LSEL0=0 (Enable LIN0 register access)
LIN wake-up baud rate select register 0	LWBR0	01H	LWBR0=1 (Wake-up signal: LIN2.x specification) LPRS[2:0]=000B (LIN0 prescaler clock: 1/1) NSPB[3:0]=0000B (Bit sampling: 16 sampling)
LIN/UART baud rate prescaler 0 register 0	LBRP00	67H	fa = 19200bps, fb = 9600bps, fc = 2400bps
LIN/UART baud rate prescaler 1 register 0	LBRP01	5FH	fd = 10417bps
LIN self-test control register 0	LSTC0	00H	LSTM=0 (LIN self-test mode is disabled)
LIN/UART mode register 0	LMD0	10H	LMD[1:0]=00B (LIN master mode) LCKS[1:0]=00B (LIN system clock: fa (19200bps)) LIOS=1 (Enable LIN0 transmission/reception/status interrupt) LRDNFS=0 (LRXD0 noise filter enabled)
LIN/UART break field configuration register 0	LBFC0	00H	BLT[3:0]=0000B (Transmission break-low width: 13 Tbit) BDT[1:0]=00B (Transmission break delimiter width: 1 Tbit)
LIN/UART space configuration register 0	LSC0	10H	IBSH[2:0]=000B (Inter-byte space (header) / Response space: 0 Tbit) IBS[1:0]=01B (Inter-byte space: 1 Tbit)
LIN wake-up configuration register 0	LWUP0	40H	WUTL[3:0]=0100B (Wake-up transmission low width: 5 Tbit)
LIN interrupt enable register 0	LIE0	0FH	FTCIE=1 (Frame / Wake-up transmission completion interrupt: Enabled) FRCIE=1 (Frame / Wake-up reception completion interrupt: Enabled) ERRIE=1 (Error detection interrupt: Enabled) SHIE=1 (Header transmission completion interrupt: Enabled)
LIN/UART error detection enable register 0	LEDE0	8FH	BERE=1 (Bit error detection: Enabled) PBERE=1 (Physical bus error detection: Enabled) FTERE=1 (Response timeout error detection: Enabled) FERE=1 (Framing error detection: Enabled) LTES=1 (Timeout selection bit: Response timeout)
LIN/UART control register 0	LCUC0	00H, 01H, 03H	LCUC0=00H (LIN reset mode transition) LCUC0=01H (LIN wake-up mode transition) LCUC0=03H (LIN operation mode transition)
LIN/UART transmission control register 0	LTRC0	01H, 02H	LTRC0=01H (Start Header transmission / Wake-up transmission) LTRC0=02H (Start response transmission/reception)
LIN/UART mode status register 0	LMST0	Read (00H, 01H, 03H)	LMST0=00H (LIN reset mode status) LMST0=01H (LIN wake-up mode status) LMST0=03H (LIN operation mode status)
LIN/UART status register 0	LST0	Read 00H (Status clear)	Read status flag Bit 0: FTC (1: Frame / Wake-up transmission completion) Bit 1: FRC (1: Response reception completion) Bit 3: ERR (1: Status error detection) Bit 6: D1RC (1: Data[1] reception completion) Bit 7: HTRC (1: Header transmission completion)
LIN/UART error status register 0	LEST0	Read 00H (Error clear)	Read error status flag Bit 0: BER (1: Bit error detection) Bit 1: PBER (1: Physical bus error detection) Bit 2: FTER (1: Response timeout error detection) Bit 3: FER (1: Framing error detection) Bit 5: CSER (1: Checksum error detection) Bit 7: RPER (1: Response preparation error detection)
LIN/UART data field configuration register 0	LDFC0	xxH	RFDL[3:0]: Response data length (0 to 8) RFT: Response direction (0: Reception, 1: Transmission) CSM: Checksum type (0: Classic checksum, 1: Enhanced checksum) FSM=1 (Select frame separate mode) LSS=0 (Next transmission/reception data group is the last one)

**Table 1-2 SFRs Used for LIN Communication in Master Mode (2/2)**

Register		Setting Value	Conditions
Name	Symbol		
LIN/UART ID buffer register 0	LIDB0	xxH	Set transmission ID and parity bit
LIN checksum buffer register 0	LCBR0	xxH	Read reception checksum <b>Note</b>
LIN/UART data buffer [1-8] registers	LDB01 - LDB08	xxH	Set transmission data[n], or read reception data[n]

**Note:** LCBR0 register is not used in this application note. Check for checksum error detection.

**Table 1-3 SFRs Used for LIN Communication in Master Mode (Port related registers)**

Register		Setting Value	Conditions
Name	Symbol		
Input switch control register	ISC	ISC2=1	LRXD0 input pin interrupt enabled
Peripheral I/O redirection register 4	PIOR4	PIOR44=0	LRXD0: P14, LTXD0: P13 selected
Port register 1	P1	P13=1	Port P1.3 output latch = High
Port mode register 1	PM1	PM13=0, PM14=1	Port P1.3: output direction, Port P1.4: input direction
Port output mode register 1	POM1	POM13=0	Port P1.3: CMOS output
Port input mode register 1	PIM1	PIM14=0	Port P1.4: CMOS input
Port input threshold control register 1	PITHL1	PITHL14=0	Port P1.4: Schmitt1 input type selected
Pull-up resistor option register 1	PU1	PU14=0	Port P1.4: Internal pull-up disabled

**Table 1-4 SFRs Used for LIN Communication in Master Mode (Interrupt related registers)**

Register		Setting Value	Conditions
Name	Symbol		
External interrupt rising edge enable register 1	EGP1	EGP11=0	LRXD0: Falling edge detection enabled
External interrupt falling edge enable register 1	EGN1	EGN11=1	
Interrupt request flag register 0H	IF0H	LIN0TRMIF=0	Clear the LIN0 transmission interrupt request flag
Interrupt request flag register 1L	IF1L	LIN0RVCIF=0	Clear the LIN0 reception interrupt request flag
		LIN0STAIF=0	Clear the LIN0 error status interrupt request flag
Interrupt request flag register 2L	IF2L	LIN0WUPIF=0	Clear the LRXD0 falling edge detection interrupt request flag
Interrupt mask flag register 0H	MK0H	LIN0TRMMK=0,1	LIN0 transmission interrupt mask flag (0: Interrupt enabled, 1: Interrupt mask)
Interrupt mask flag register 1L	MK1L	LIN0RVCMK=0,1	LIN0 reception interrupt mask flag (0: Interrupt enabled, 1: Interrupt mask)
		LIN0STAMK=0,1	LIN0 status interrupt mask flag (0: Interrupt enabled, 1: Interrupt mask)
Interrupt mask flag register 2L	MK2L	LIN0WUPMK=0,1	LRXD0 falling edge detection interrupt mask flag (0: Interrupt enabled, 1: Interrupt mask)
Priority specification flag registers 10H/00H	PR10H, PR00H	LIN0TRMPR=01B	LIN0 transmission interrupt priority level flag 0, 1 (01B: Level-1)
Priority specification flag registers 11L/01L	PR11L, PR01L	LIN0RVCPR=01B	LIN0 reception interrupt priority level flag 0, 1 (01B: Level-1)
		LIN0STAPR=01B	LIN0 status interrupt priority level flag 0, 1 (01B: Level-1)
Priority specification flag registers 12L/02L	PR12L, PR02L	LIN0WUPPR=01B	LRXD0 falling edge detection interrupt priority level flag 0, 1 (01B: Level-1)

## 2. Overview of LIN Master Communication

### 2.1 LIN Master Communication Processing Overview

LIN master communication example flow is shown below.

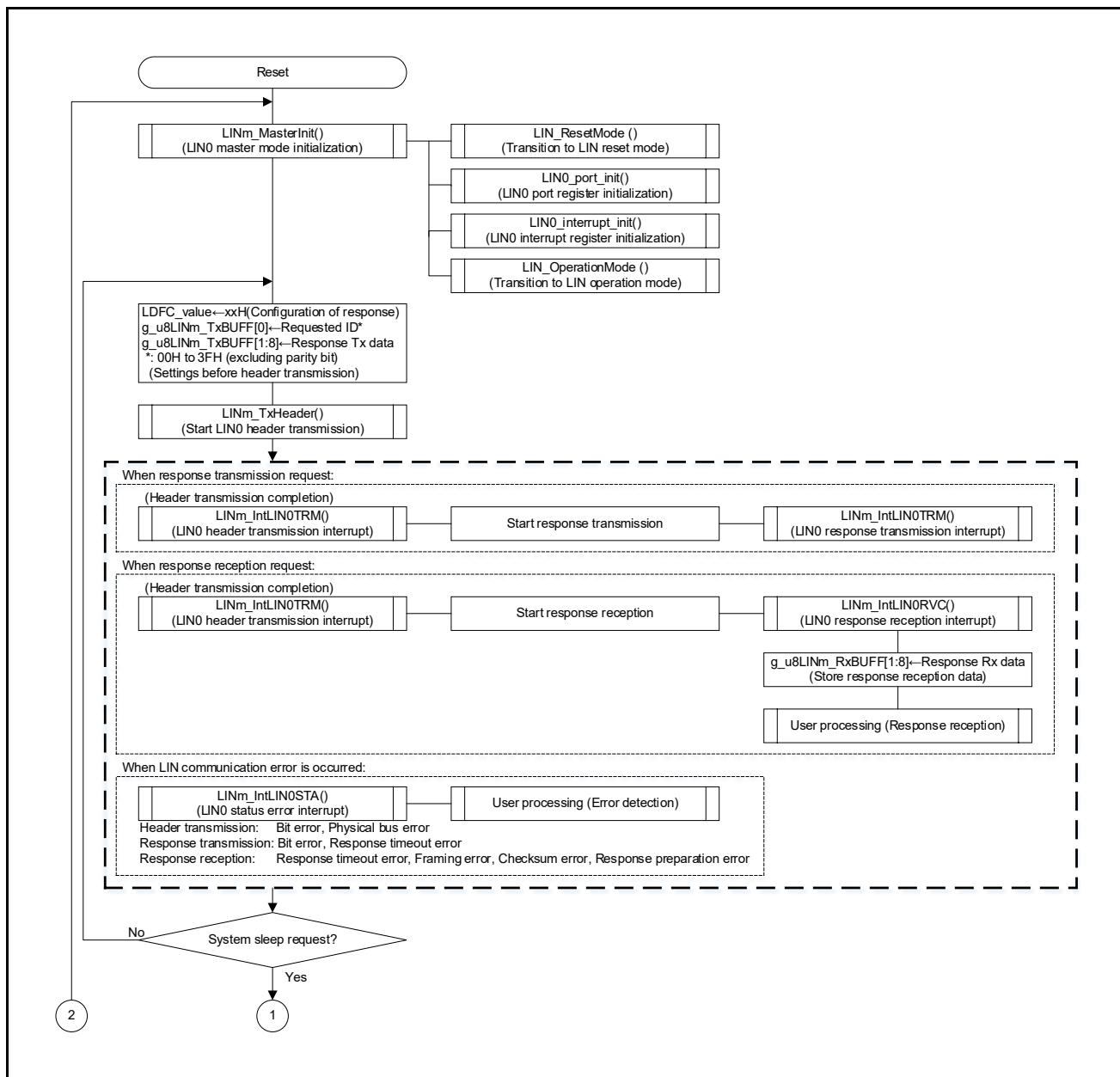


Figure 2-1 LIN Communication Process in Master Mode (1/2)

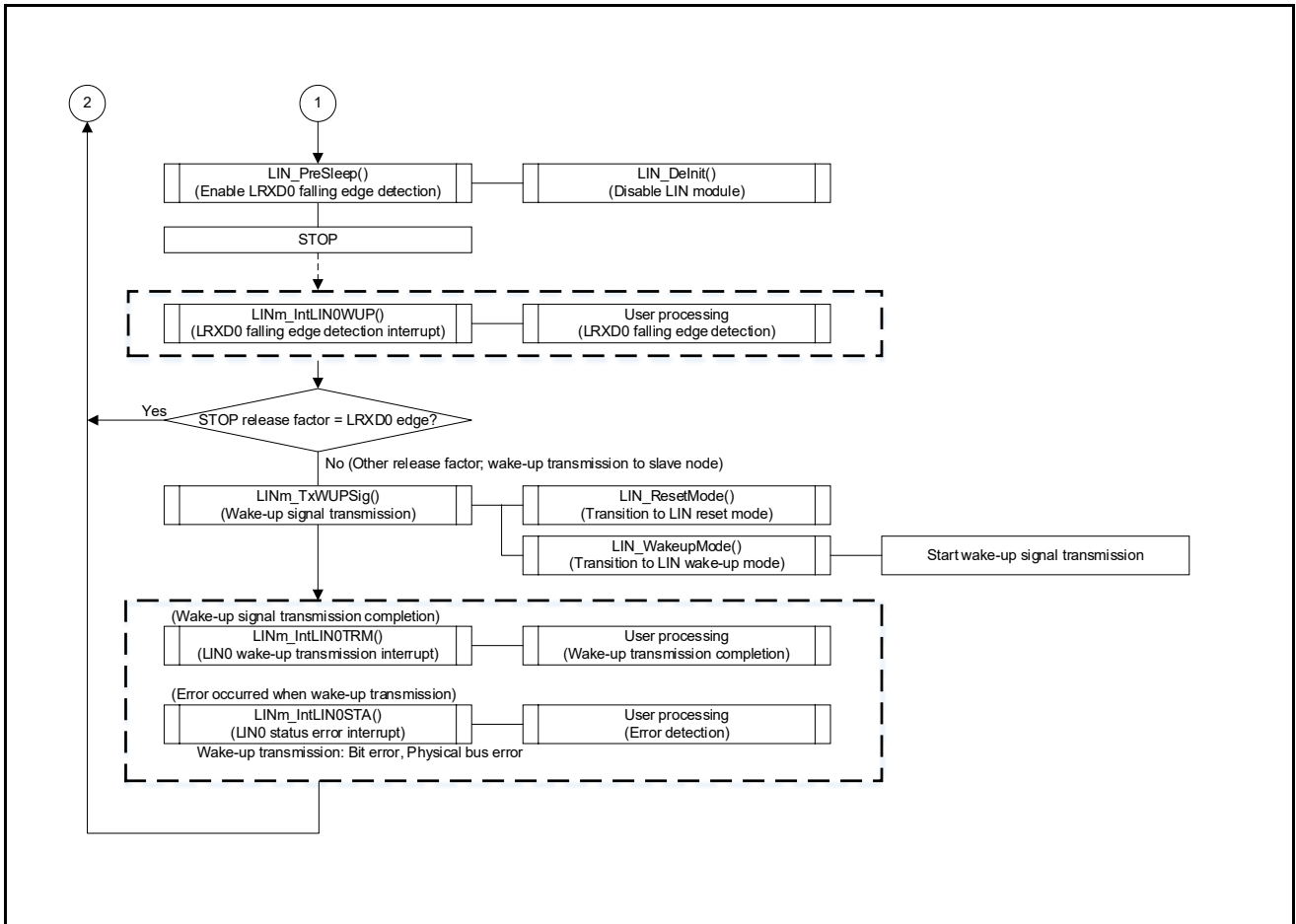


Figure 2-1 LIN Communication Process in Master Mode (2/2)



## 2.2 LIN Frame Timing (frame separate mode)

LIN master communication uses a frame-separated mode in which the header and response are started by separate transmission requests. The timing diagram for header transmission and response transmission and reception in the LIN master are shown below.

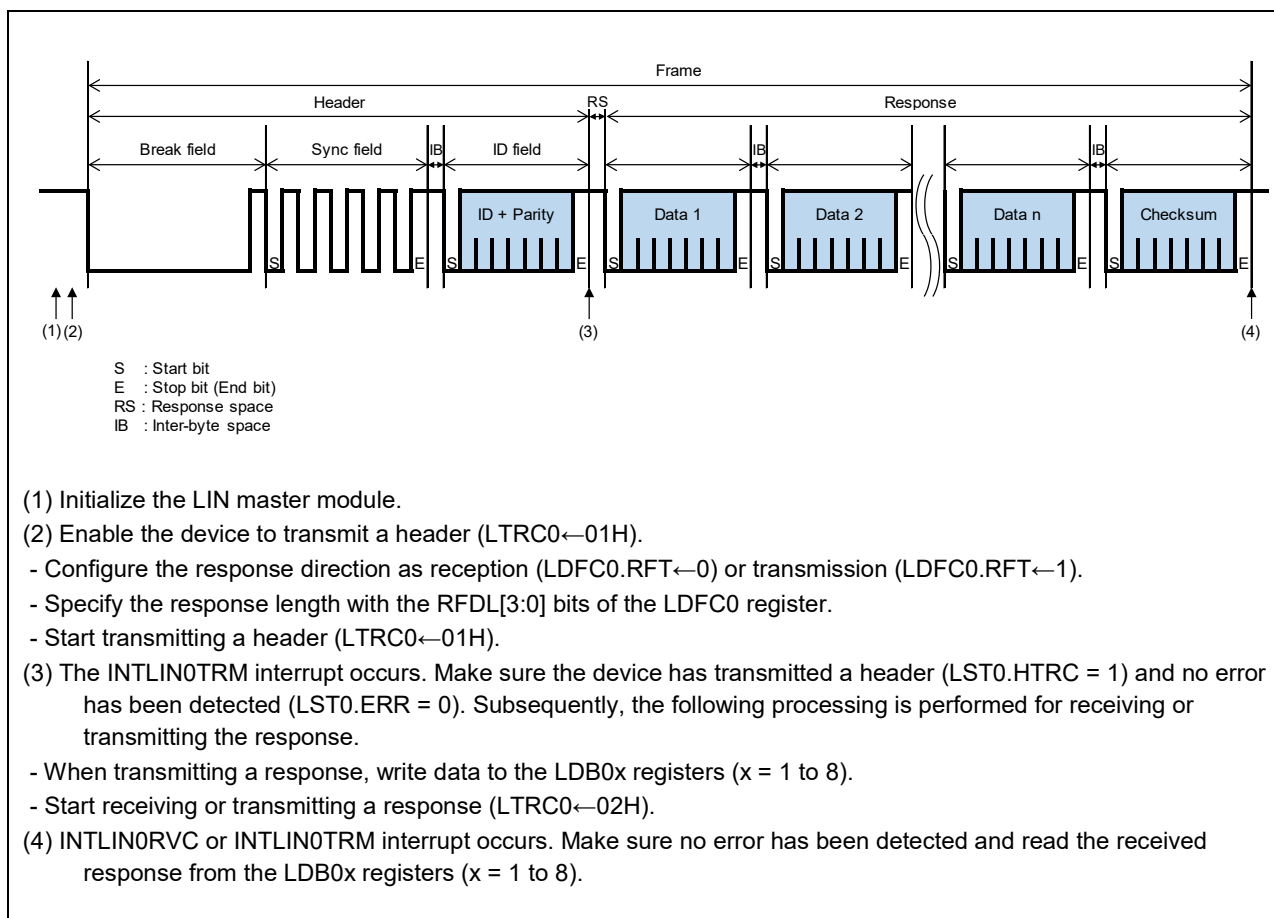


Figure 2-2 LIN Master Header Transmission and Response Transmission / Reception Timing

## 2.3 LIN Master Communication Status and SFR

The LIN/UART module (RLIN3) generates LINn successful transmission interrupt, LINn successful reception interrupt, and LINn reception status interrupt. Also, each interrupt has multiple interrupt factors. For details, refer to the user's manual of the target products. In this application note, the status register of RLIN3 module is read to determine the interrupt source. Connections between LIN master communication status and SFRs are shown below.

**Table 2-1 LIN Communication State and SFRs (LINn successful transmission interrupt)**

LIN Communication State Transmission	SFRs				
	LST0			LDFC0	LMST0
	bit3(ERR)	bit7(HTRC)	bit0(FTC)	RFT	
Communication error * Processed in the LIN0 status interrupt	1	-	-	-	-
Header transmission completion	0	1	-	0	-
Response not requested				1	
Response requested				-	
Response transmission completion	0	0	1	-	03H
Wake-up transmission completion				-	≠03H
Illegal interrupt occurred				0	-

-: Not judged by software.

**Remarks:** For details, see section 3.4.1.

**Table 2-2 LIN Communication State and SFRs in Master Mode (LINn successful reception interrupt)**

LIN Communication State Reception	SFRs	
	LST0	
	bit3(ERR)	bit1(FRC)
Communication error * Processed in the LIN0 status interrupt	1	-
Response reception completion	0	1
Illegal interrupt occurred		0

-: Not judged by software.

**Remarks:** For details, see section 3.4.2.

### 3. LIN Master Driver Functions

The LIN master communication driver program example shown in this application note is explained below.

#### 3.1 Function list

The function list of this driver program and the variables used are shown below.

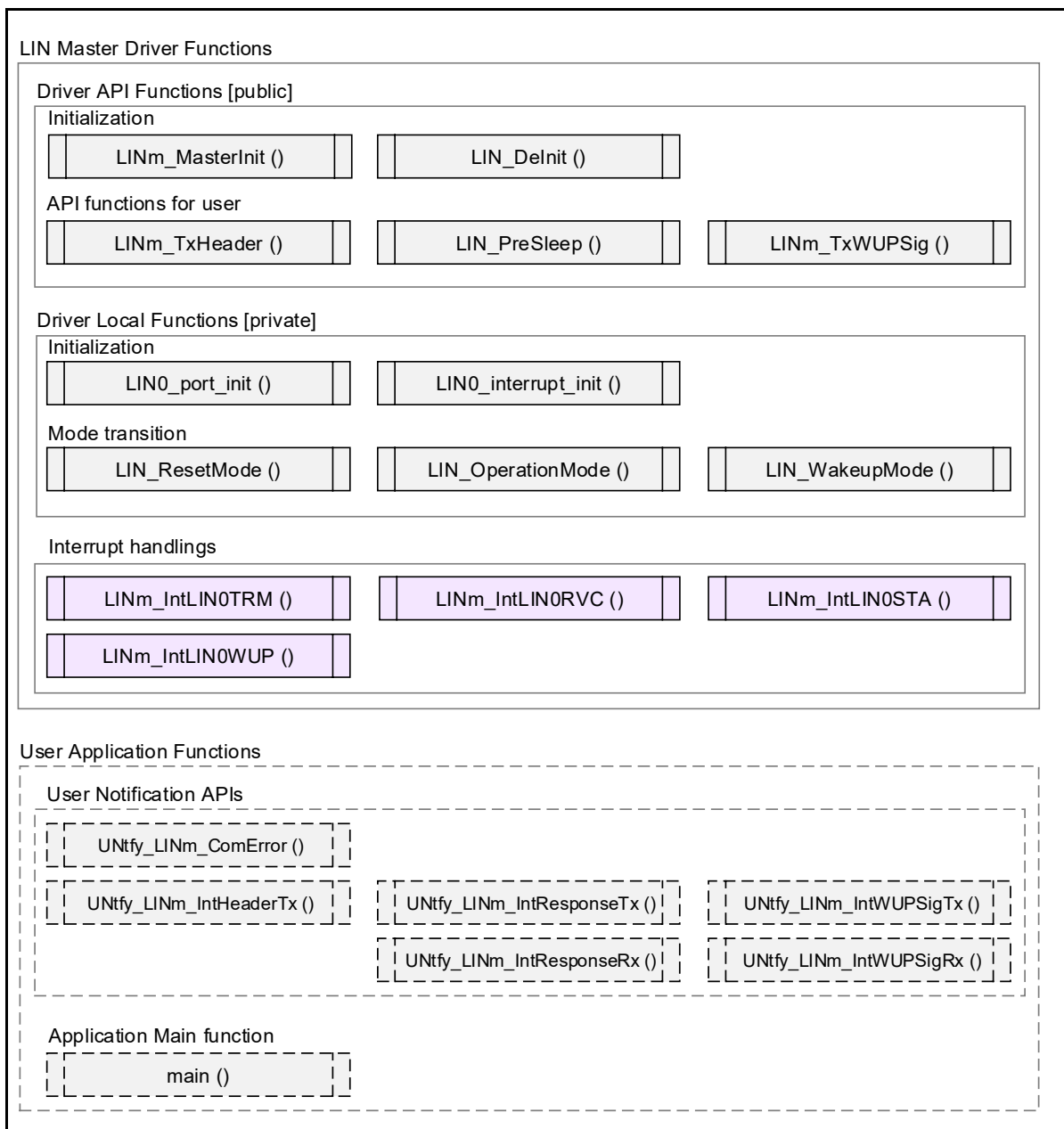


Figure 3-1 Functions of LIN Communication in Master Mode

Table 3-1 Function List for LIN Communication in Master Mode

Function Name	Description	Public / Private
void LINm_MasterInit (void)	This function initializes LIN (master mode), port and interrupt functions.	public
void LIN_DelInit (void)	This function disables LIN (master) module.	public
void LINm_TxHeader ( unsigned char u8a_LDFC_value)	This function is the transmission process of LIN header.	public
void LIN_PreSleep (void)	This function is process before transition to sleep mode.	public
void LINm_TxWUPSig (void)	This function transmits the LIN wake-up signal.	public
void LIN0_port_init (void)	Initialization for LIN related port function	private
void LIN0_interrupt_init (void)	Initialization for LIN related interrupt function	private
void LIN_ResetMode (void)	This function transitions to LIN reset mode.	private
void LIN_OperationMode (void)	This function transitions to LIN operation mode.	private
void LIN_WakeupMode (void)	This function transitions to LIN wake-up mode.	private
void LINm_IntLIN0TRM (void)	This function is the LIN0 transmission completion interrupt processing. (Header transmission / Response transmission / Wake-up transmission)	public
void LINm_IntLIN0RVC (void)	This function is the LIN0 response reception completion interrupt processing. (Response reception / Wake-up reception) <sup>Note</sup>	public
void LINm_IntLIN0STA (void)	This function is the interrupt processing when LIN0 status error is occurred. - Header transmission: Bit error / Physical bus error detection - Response transmission: Bit error / Response timeout error detection - Response reception: Response timeout error / Framing error / Checksum error detection - Wake-up transmission: Bit error / Physical bus error detection	public
void LINm_IntLIN0WUP (void)	This function is the LRXD0 falling edge detection interrupt processing.	public

**Note:** In this application note, wake-up detection is detected in LRXD0 falling edge. Wake-up reception interrupt is not used.

Table 3-2 Variables Used for LIN Communication in Master Mode

Variable Name	Description
unsigned char g_u8LINm_TxBUFF[9]	g_u8LINm_TxBUFF[0]: Set requested ID data g_u8LINm_TxBUFF[1 to 8]: Set response transmission data
unsigned char g_u8LINm_RxBUFF[9]	g_u8LINm_RxBUFF[0]: Not used g_u8LINm_RxBUSS[1 to 8]: Store response reception data

Table 3-3 User Notification Function APIs for LIN Master Driver

Function API Name	Description
void UNTfy_LINm_ComError ( unsigned short u16a_errorinfo)	User's processing for error detections. bit[7:0]: Store the LEST0 register value (When INTLIN0STA occurred) bit[15:8]: When INTLIN0STA occurred: 00H When INTLIN0RVC occurred: 80H (illegal interrupt) When INTLIN0TRM occurred: 40H (illegal interrupt)
void UNTfy_LINm_IntHeaderTx (void)	User's processing for header transmission completion.
void UNTfy_LINm_IntResponseTx (void)	User's processing for response transmission completion.
void UNTfy_LINm_IntResponseRx (void)	User's processing for response reception completion.
void UNTfy_LINm_IntWUPSigTx (void)	User's processing for wake-up signal transmission completion.
void UNTfy_LINm_IntWUPSigRx (void)	User's processing for wake-up signal reception completion.

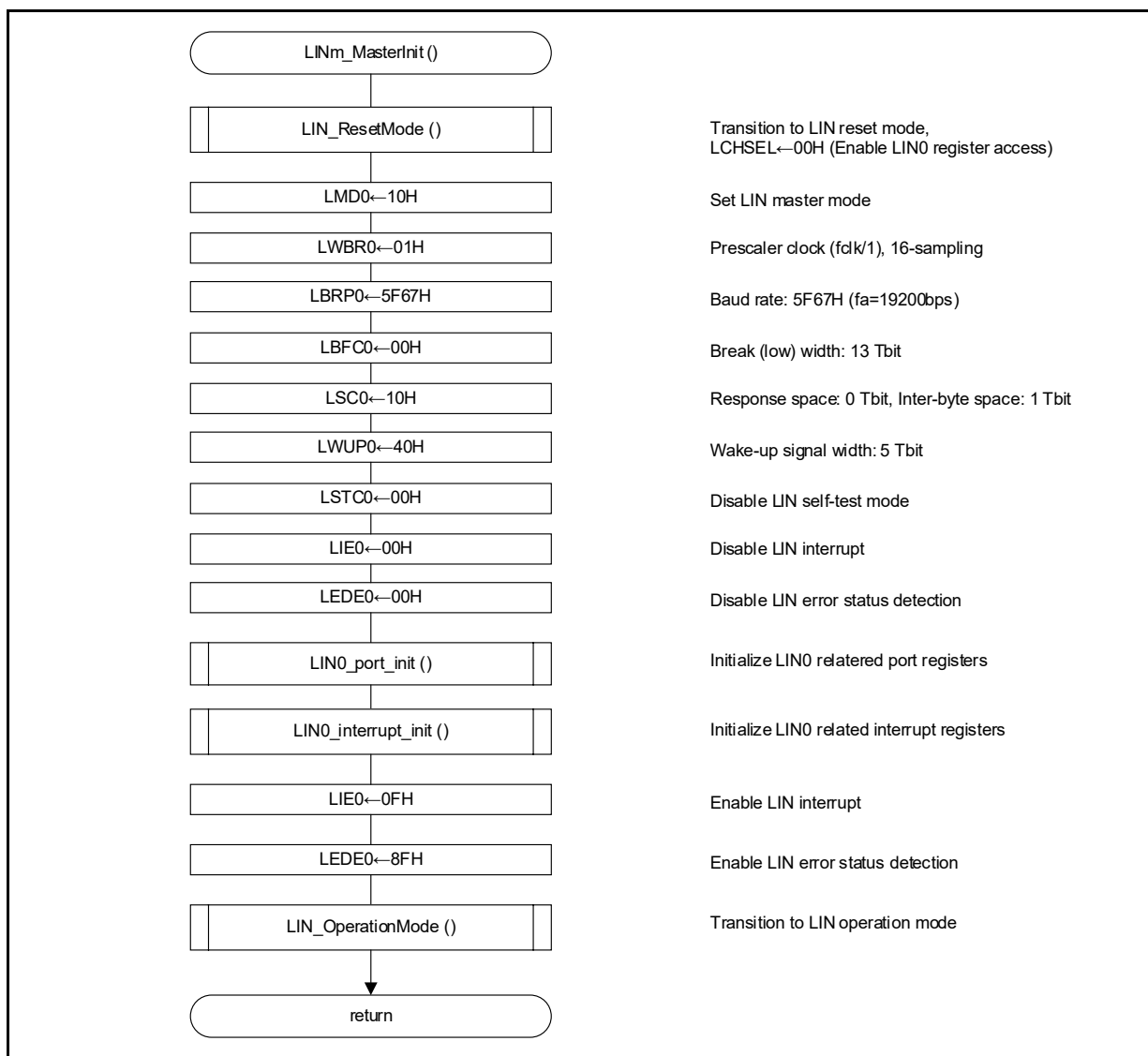
### 3.2 Driver API Functions [Public]

Each public function APIs of the LIN master communication driver program example shown in this application note is explained below.

#### 3.2.1 LIN Master Module Initialization

**Table 3-4 LIN Master Initialization Function (LINm\_MasterInit)**

Syntax	void LINm_MasterInit (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	LIN_ResetMode ():	Transition to LIN reset mode.
	LIN0_port_init ():	LIN0 port function initialization.
	LIN0_interrupt_init ():	LIN0 interrupt function initialization.
	LIN_OperationMode ():	Transition to LIN operation mode.
Description	This function initializes LIN (master mode), port and interrupt functions.	
Note	Call from upper layer with interrupt disabled.	

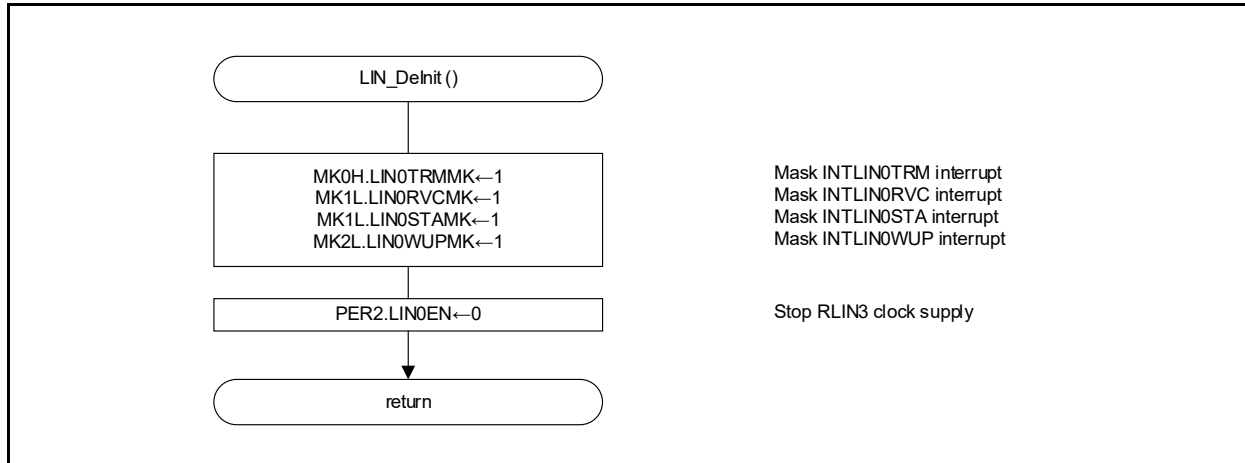


**Figure 3-2 LIN Master Initialization Procedure (LINm\_MasterInit)**

### 3.2.2 LIN Module De-Initialization

**Table 3-5 LIN De-Initialization Function (LIN\_Delnit)**

Syntax	void LIN_Delnit (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function disables LIN module.	
Note	Call this function when GCSC and GINT bits of IAWCTL register are 0.	

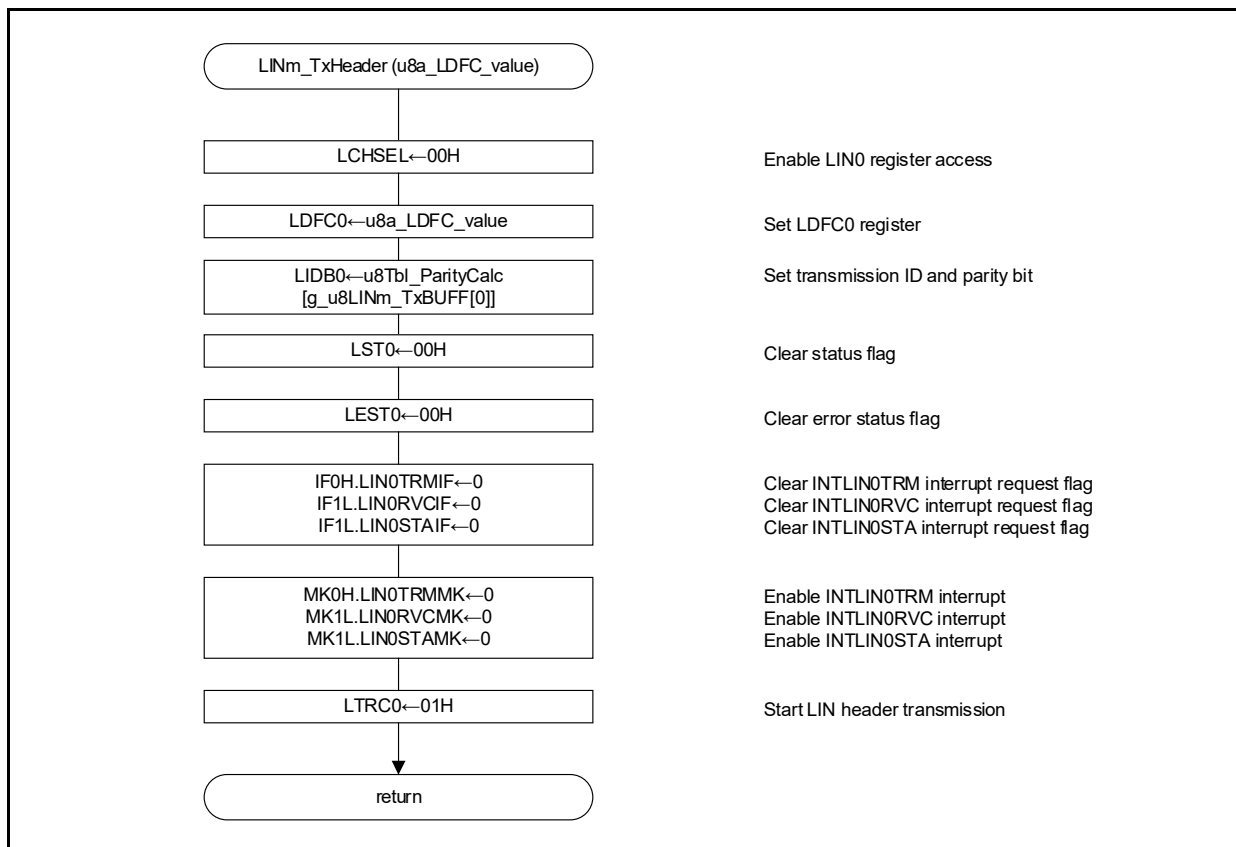


**Figure 3-3 LIN De-Initialization Procedure (LIN\_Delnit)**

### 3.2.3 LIN Master Header Transmission

**Table 3-6 LIN Master Header Transmission Function (LINm\_TxHeader)**

Syntax	void LINm_TxHeader (unsigned char u8a_LDFC_value)	
Parameters	In	unsigned char u8a_LDFC_value LDFC0 setting value - bit3-0 (RFDL[3:0]): Response field length [0 to 8] - bit4 (RFT): Response direction [0: Reception, 1: Transmission] - bit5 (CSM): Checksum type [0: classic-checksum, 1: enhanced-checksum] - bit6 (FSM): 1: Frame separate mode - bit7 (LSS): 0: Next transmission/reception data group is the last one
		unsigned char g_u8LINm_TxBUFF[0] Transmission ID setting
		unsigned char g_u8LINm_TxBUFF[1:8] Response transmission data setting
	Out	None
Local variables	const unsigned char u8TBL_ParityCalc[0x40u]	Parity bit calculation of requested ID (Refer to Chapter. 6)
Return value	None	
Call functions	None	
Description	This function is the transmission process of LIN header.	
Note	Call this function when GINT bit of IAWCTL register is 0.	

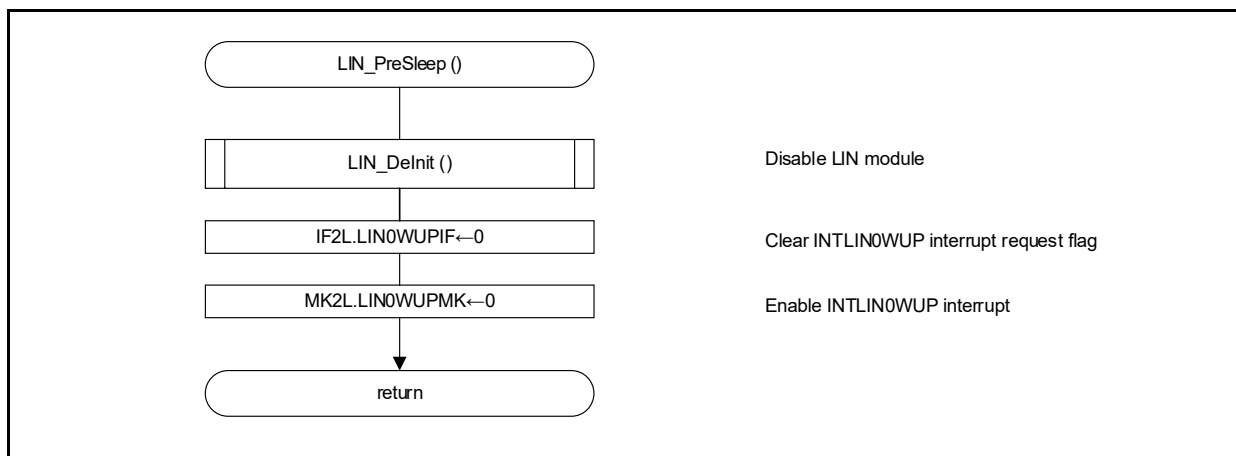


**Figure 3-4 LIN Master Header Transmission Procedure (LINm\_TxHeader)**

3.2.4 Before Sleep Mode Transition Processing

**Table 3-7 Before Sleep Mode Transition Processing Function (LIN\_PreSleep)**

Syntax	void LIN_PreSleep (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	LIN_DeInit (): Disable LIN module function	
Description	This function is process before transition to sleep mode.	
Note	Call from upper layer with interrupt disabled. Call this function when GINT bit of IAWCTL register is 0.	



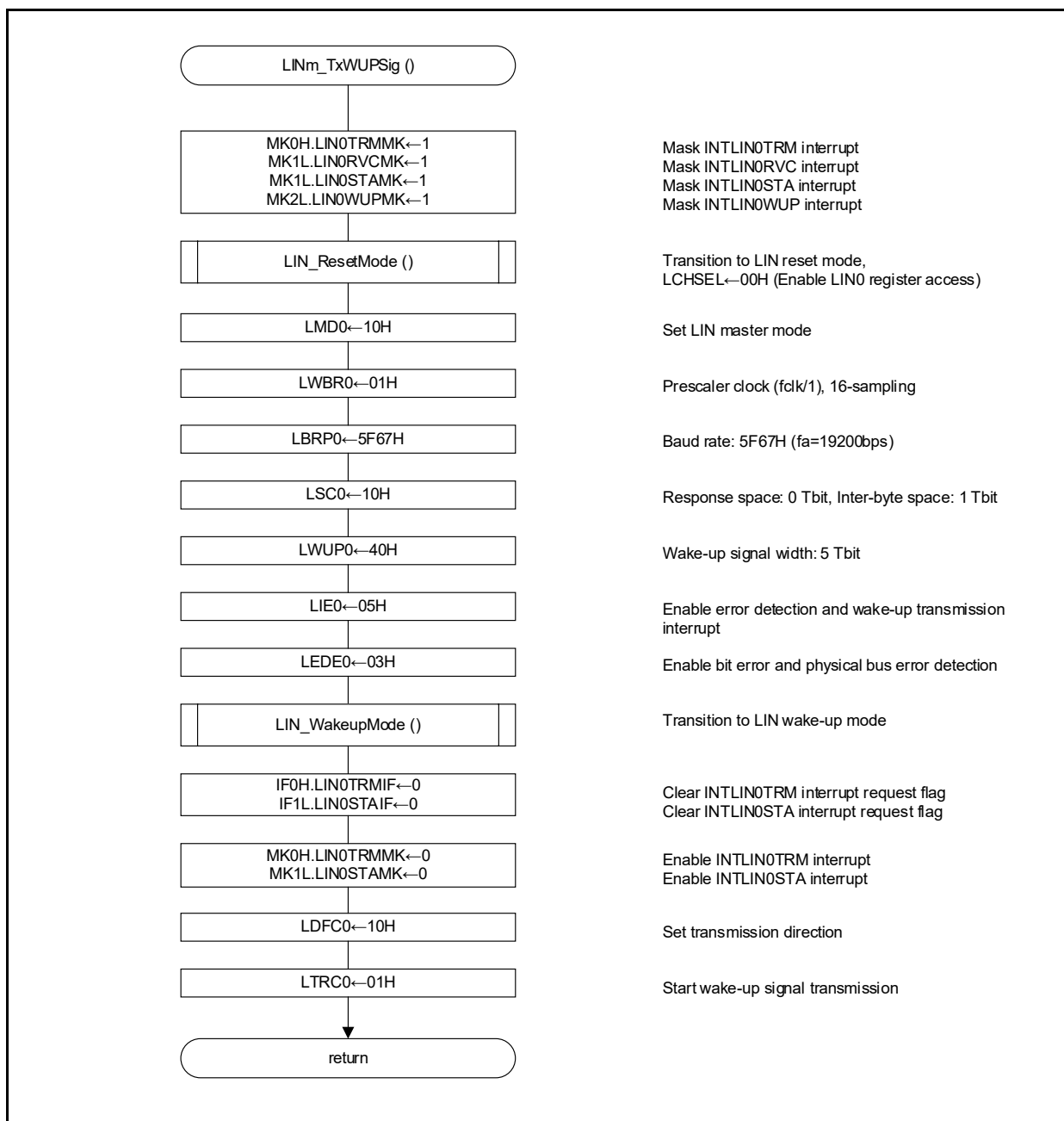
**Figure 3-5 Before Sleep Mode Transition Processing Procedure (LIN\_PreSleep)**



### 3.2.5 LIN Master Wake-up Signal Transmission

**Table 3-8 LIN Master Wake-up Signal Transmission Function (LINm\_TxWUPSig)**

Syntax	void LINm_TxWUPSig (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	LIN_ResetMode (): LIN reset mode transition function LIN_WakeupMode (): LIN wake-up mode transition function	
Description	This function transmits the LIN wake-up signal.	
Note	Call this function when GINT bit of IAWCTL register is 0.	



**Figure 3-6 LIN Master Wake-up Signal Transmission Procedure (LINm\_TxWUPSig)**

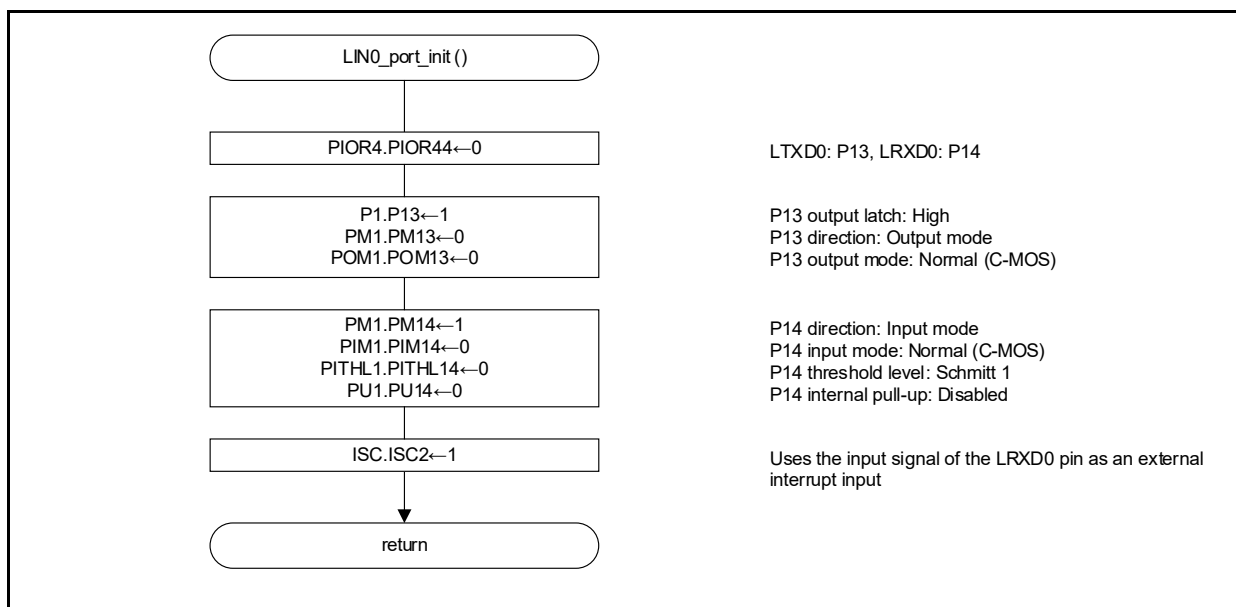
### 3.3 Driver Local Functions [Private]

Each local functions of the LIN master communication driver program example shown in this application note is explained below.

#### 3.3.1 LIN0 Port Initialization

**Table 3-9 LIN0 Port Initialization Function (LIN0\_port\_init)**

Syntax	void LIN0_port_init (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function initializes LIN0 related ports.	
Note	Call this function when GPORT bit of IAWCTL register is 0.	

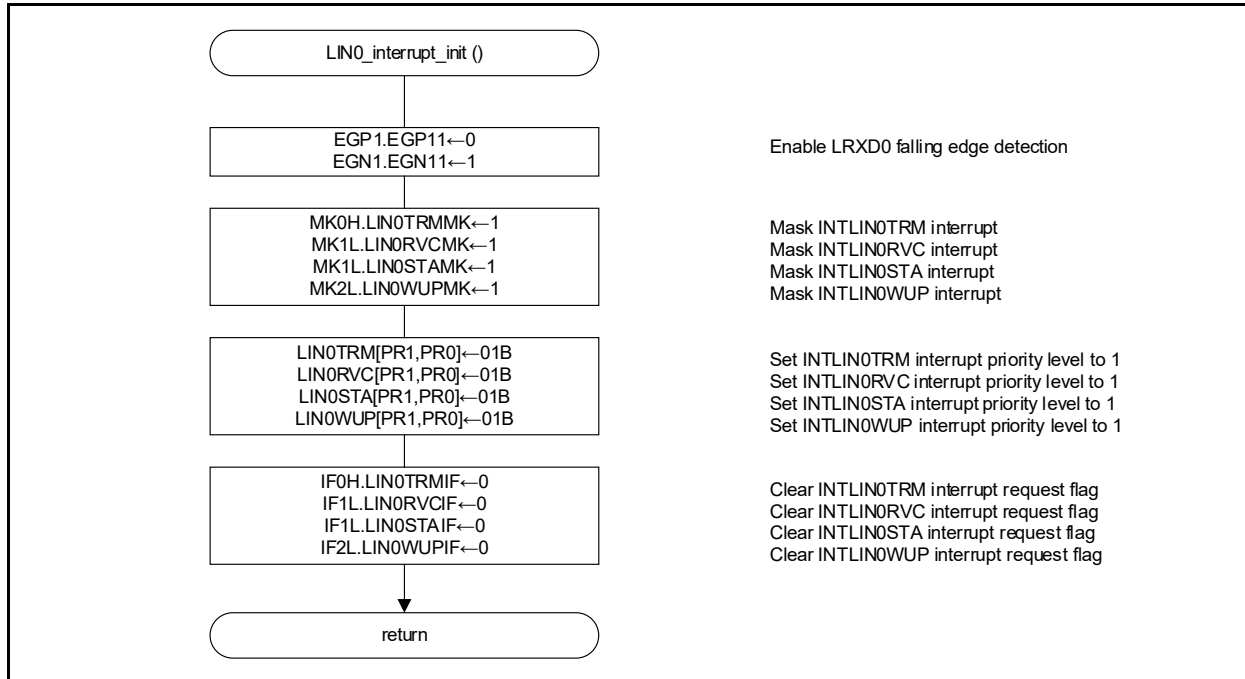


**Figure 3-7 LIN0 port Initialization Procedure (LIN0\_port\_init)**

### 3.3.2 LIN0 Interrupt Initialization

**Table 3-10 LIN0 Interrupt Initialization Function (LIN0\_interrupt\_init)**

Syntax	void LIN0_interrupt_init (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function initializes LIN0 related interrupts.	
Note	Call this function when GINT bit of IAWCTL register is 0.	

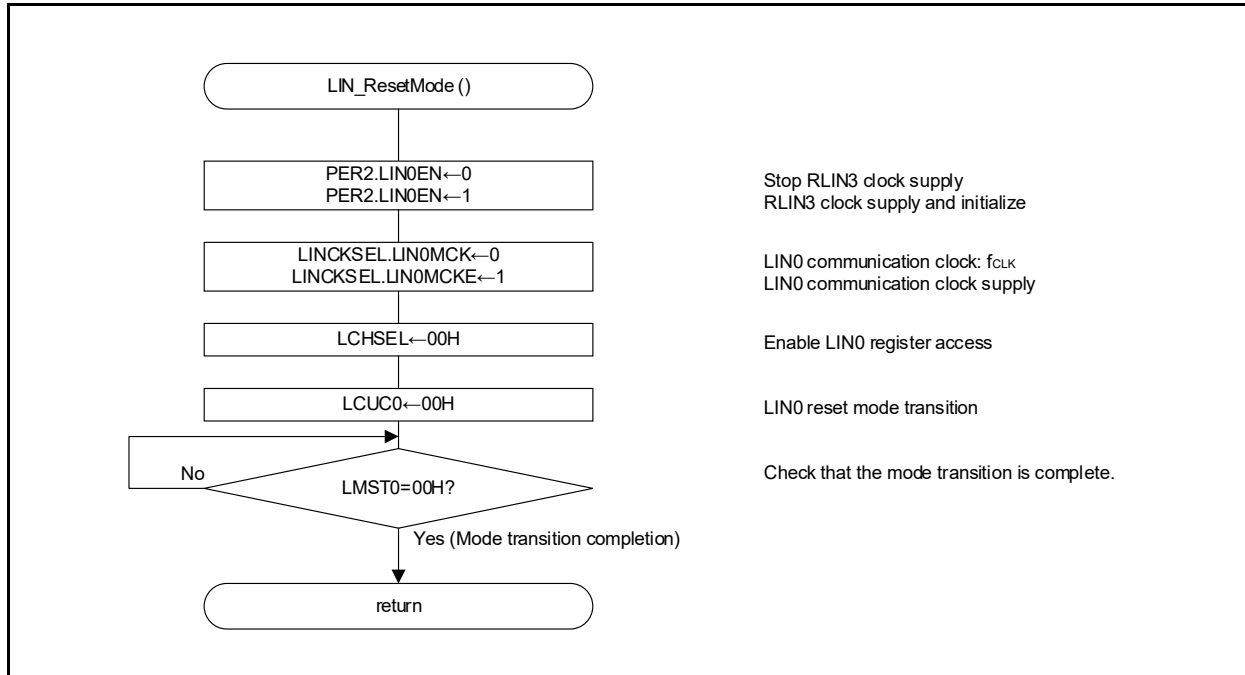


**Figure 3-8 LIN0 Interrupt Initialization Procedure (LIN0\_interrupt\_init)**

3.3.3 Transition to LIN Reset Mode

**Table 3-11 Transition to LIN Reset Mode Function (LIN\_ResetMode)**

Syntax	void LIN_ResetMode (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function transitions to LIN reset mode.	
Note	Call this function when GCSC bit of IAWCTL register is 0.	

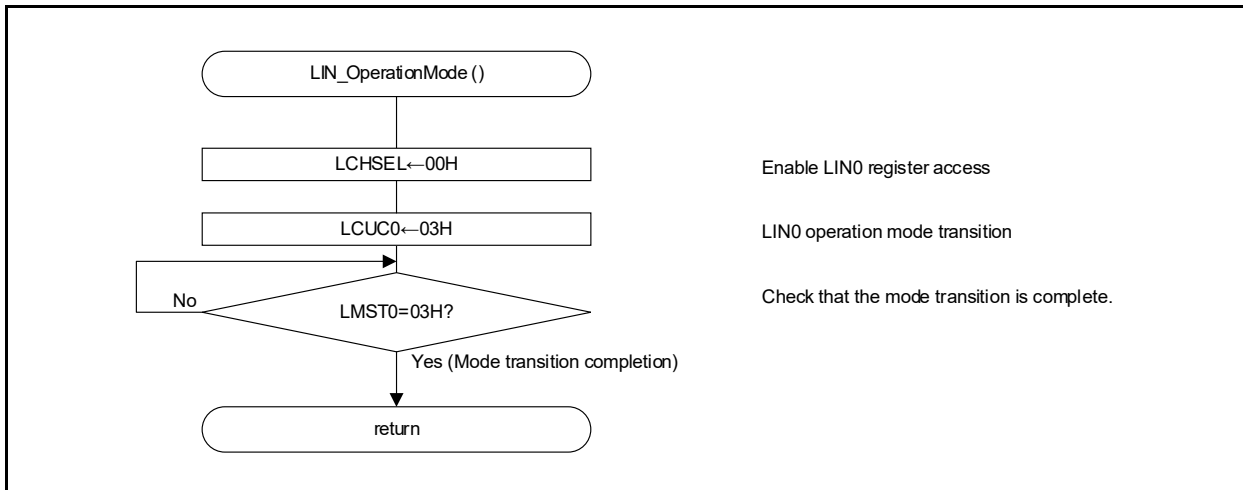


**Figure 3-9 Transition to LIN Reset Mode Procedure (LIN\_ResetMode)**

3.3.4 Transition to LIN Operation Mode

**Table 3-12 Transition to LIN Operation Mode Function (LIN\_OperationMode)**

Syntax	void LIN_OperationMode (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function transitions to LIN operation mode.	

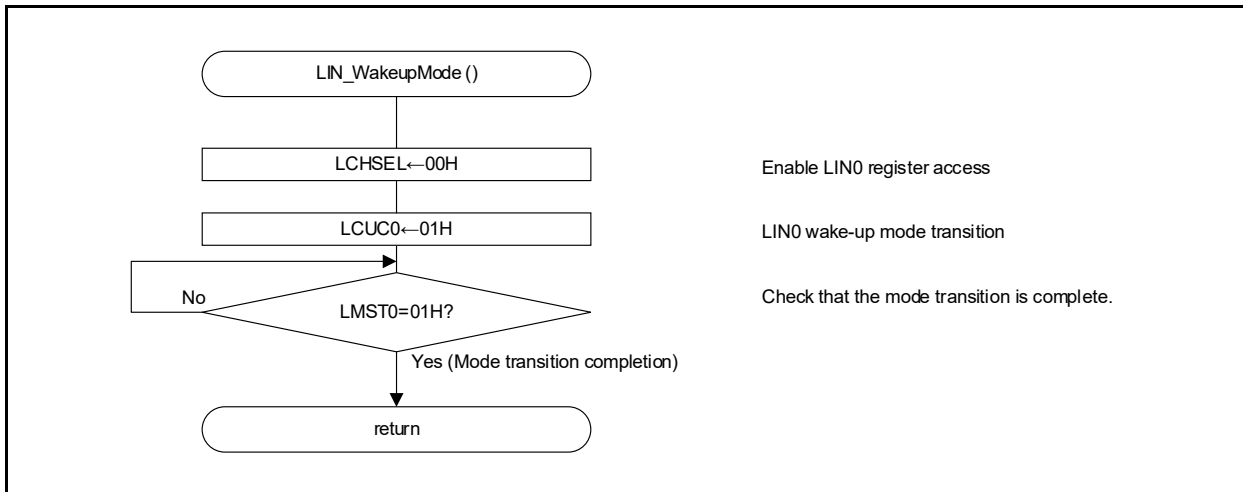


**Figure 3-10 Transition to LIN Operation Mode Procedure (LIN\_OperationMode)**

3.3.5 Transition to LIN Wake-up Mode

**Table 3-13 Transition to LIN Wake-up Mode Function (LIN\_WakeupMode)**

Syntax	void LIN_WakeupMode (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function transitions to LIN wake-up mode.	



**Figure 3-11 Transition to LIN Wake-up Mode Procedure (LIN\_WakeupMode)**

### 3.4 Interrupt Processing Functions

Each interrupt processing functions of the LIN master communication driver program example shown in this application note is explained below.

#### 3.4.1 LIN Master Transmission Completion Interrupt Processing

**Table 3-14 LIN Master Transmission Completion Interrupt Processing Function (LINm\_IntLIN0TRM)**

Syntax	void LINm_IntLIN0TRM (void)	
Parameters	In	unsigned char g_u8LINm_TxBUFF[1:8]: Response transmission data setting* *: Set before call LINm_TxHeader () function
	Out	None
Local variables	unsigned char u8a_RD_LST0:	Store LST0 register
	- bit0 (FTC): - bit3 (ERR): - bit7 (HTRC):	Response / Wake-up transmission completion LIN communication error detection Header transmission completion
	unsigned short u16a_errorinfo:	4000H (Illegal interrupt has occurred)
Return value	None	
Call functions	UNtfy_LINm_IntHeaderTx ():	Header transmission completion processing
	UNtfy_LINm_IntResponseTx ():	Response transmission completion processing
	UNtfy_LINm_IntWUPSigTx ():	Wake-up transmission completion processing
	UNtfy_LINm_ComError (u16a_errorinfo):	Illegal interrupt processing
Description	This function is the LIN0 transmission completion interrupt processing. - LIN0 header transmission completion - LIN0 response transmission completion - LIN0 wake-up signal transmission completion	

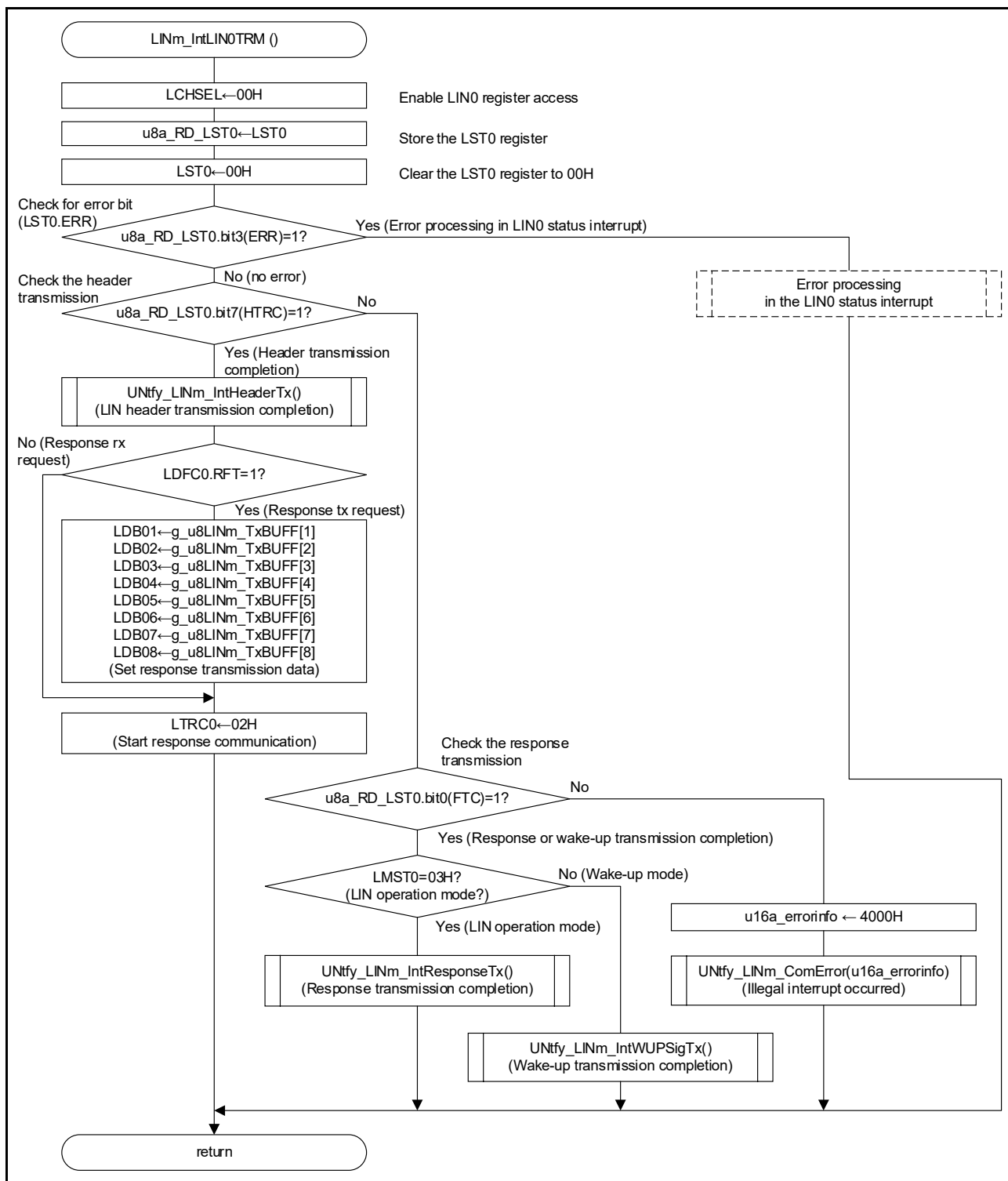


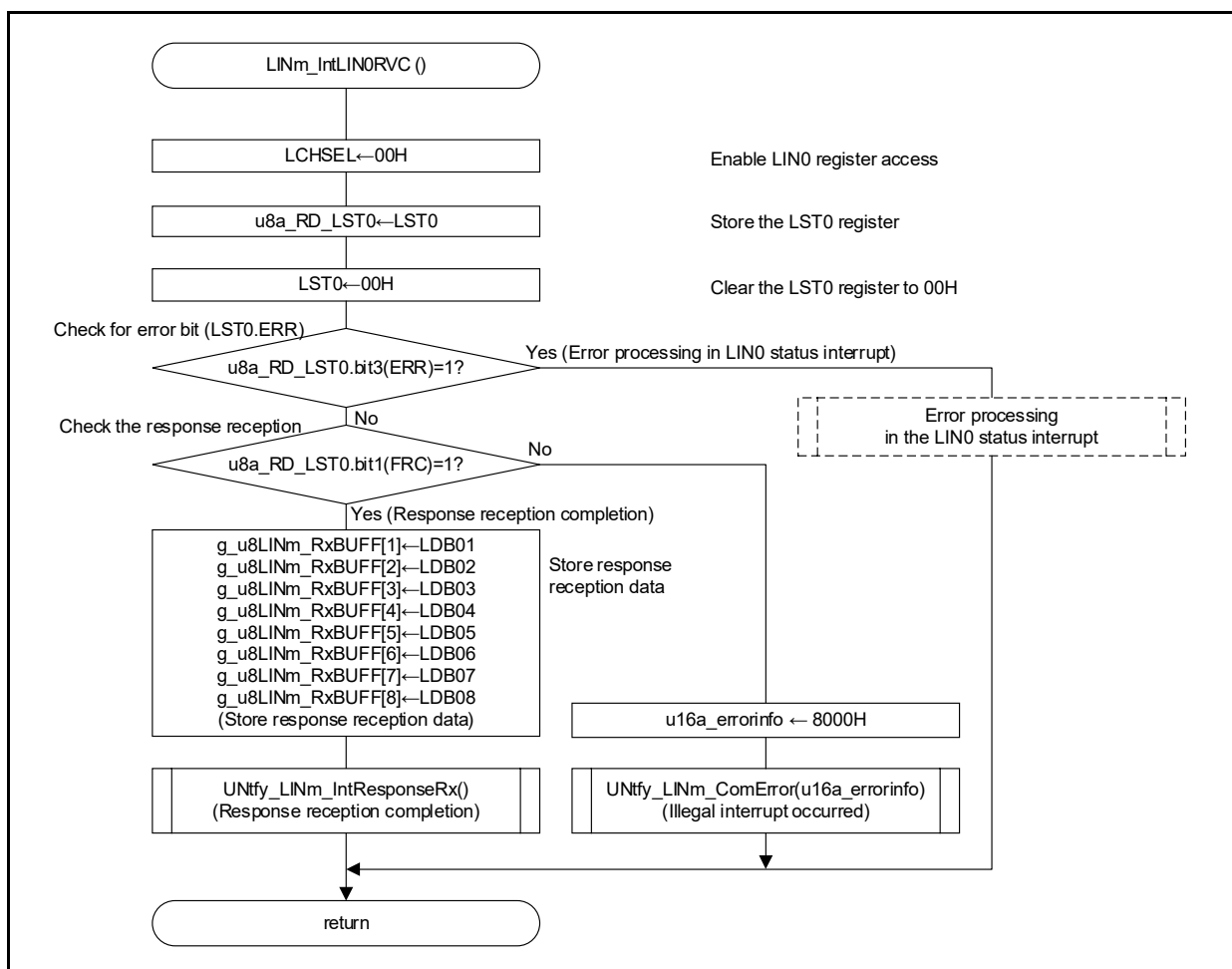
Figure 3-12 LIN Master Transmission Completion Interrupt Processing Procedure (LINm\_IntLIN0TRM)



### 3.4.2 LIN Master Reception Completion Interrupt Processing

**Table 3-15 LIN Master Reception Completion Interrupt Processing Function (LINm\_IntLIN0RVC)**

Syntax	void LINm_IntLIN0RVC (void)		
Parameters	In	None	
	Out	unsigned char	g_u8LINm_RxBUFF[1:8]: Response reception data
Local variables	unsigned char	u8a_RD_LST0:	Store LST0 register
		- bit1 (FRC): - bit3 (ERR):	Response reception completion LIN communication error detection
	unsigned short	u16a_errorinfo:	8000H (Illegal interrupt has occurred)
Return value	None		
Call functions	UNTfy_LINm_IntResponseRx ():	Response reception completion processing	
	UNTfy_LINm_ComError (u16a_errorinfo):	Illegal interrupt processing	
Description	This function is the LIN0 response reception completion interrupt processing.		

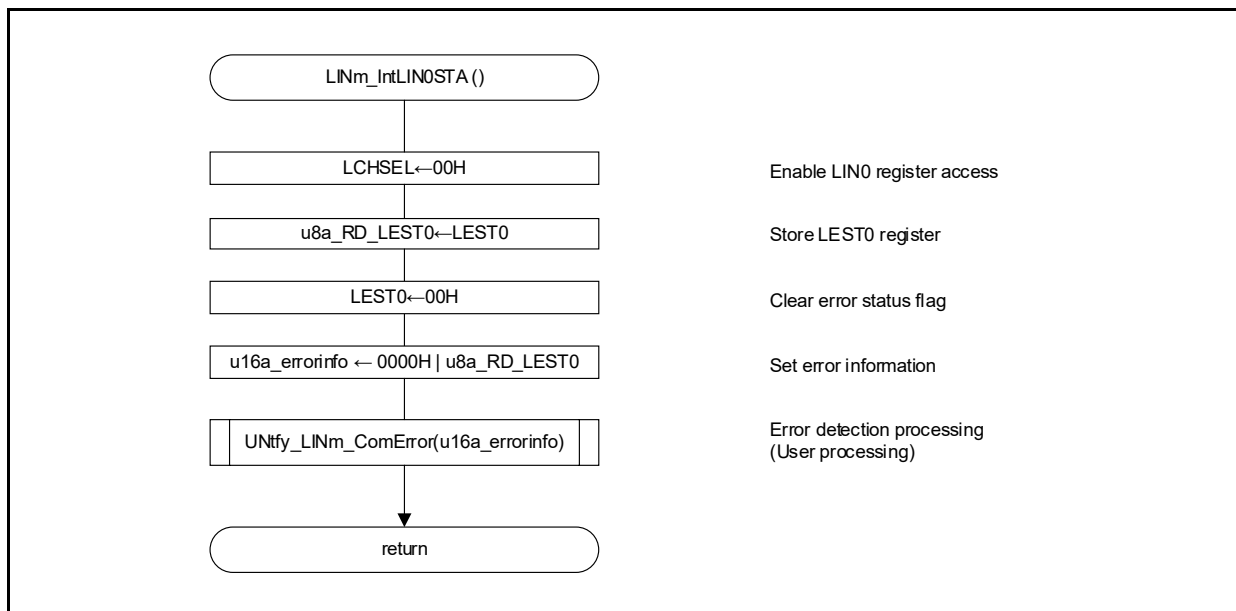


**Figure 3-13 LIN Master Reception Completion Interrupt Processing Procedure (LINm\_IntLIN0RVC)**

### 3.4.3 LIN Master Status Error Interrupt Processing

**Table 3-16 LIN Master Status Error Interrupt Processing Function (LINm\_IntLIN0STA)**

Syntax	void LINm_IntLIN0STA (void)	
Parameters	In	None
	Out	None
Local variables	unsigned char	u8a_RD_LEST0: Store LEST0 register
	- bit0 (BER): - bit1 (PBER): - bit2 (FTER): - bit3 (FER): - bit5 (CSER): - bit7 (RPER):	Bit error detection in header / response / wake-up transmission Physical bus error detection in header / wake-up transmission Response timeout error detection in response tx/rx Framing error detection in response reception Checksum error detection in response reception Response preparation error detection in response reception
Return value	unsigned short	u16a_errorinfo: Error information value
	- bit[7:0]: - bit[15:8]:	u8a_RD_LEST0 value 00H
Return value	None	
Call functions	UNtfy_LINm_ComError (u16a_errorinfo):	Error detection processing
Description	This function is the interrupt processing when LIN0 status error is occurred.	

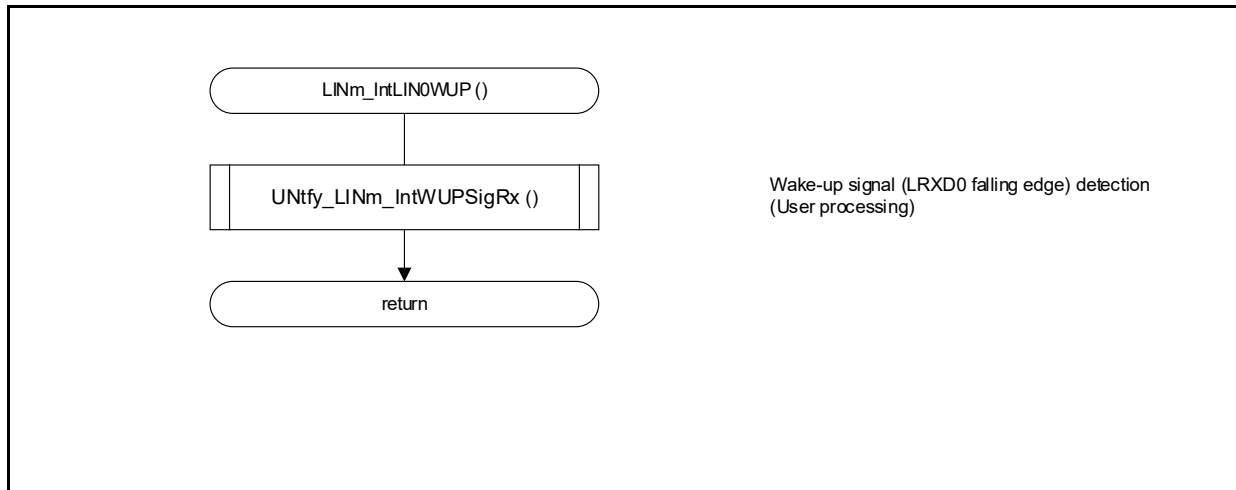


**Figure 3-14 LIN Master Status Error Interrupt Processing Procedure (LINm\_IntLIN0STA)**

3.4.4 LIN Master Wake-up Signal Detection Interrupt Processing

**Table 3-17 LIN Master Wake-up Signal Detection Interrupt Processing Function (LINm\_IntLIN0WUP)**

Syntax	void LINm_IntLIN0WUP (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	UNtfy_LINm_IntWUPSigRx (): L_RXD0 falling edge detection	
Description	INTLIN0WUP interrupt service routine	



**Figure 3-15 LIN Master Wake-up Signal Detection Interrupt Processing Procedure (LINm\_IntLIN0WUP)**

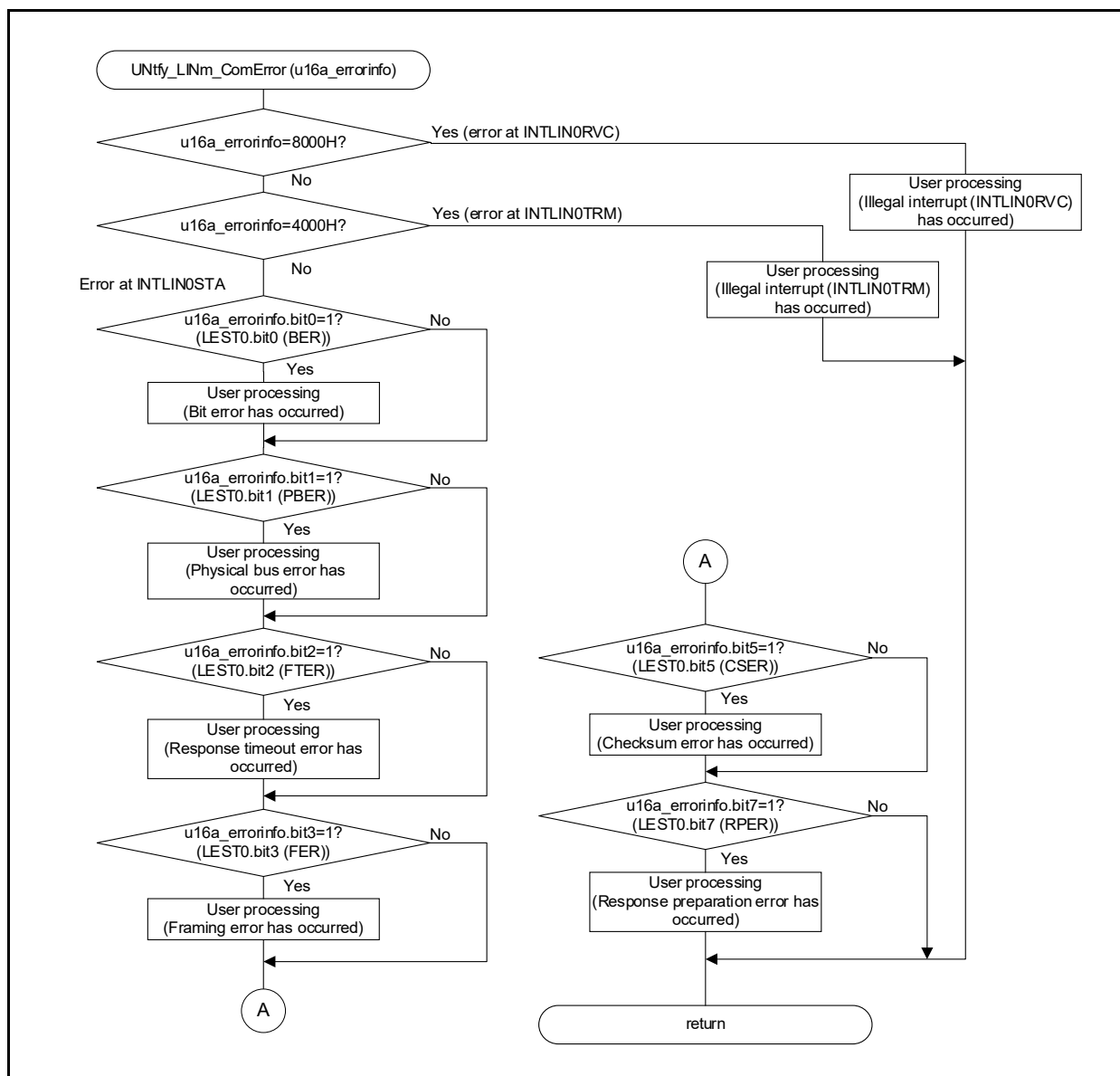
### 3.5 User Notification Function APIs

Each user notification function APIs of the LIN master communication driver program example shown in this application note is explained below.

#### 3.5.1 Error Notification

**Table 3-18 Error Notification Function API (UNtfy\_LINm\_ComError)**

Syntax	void UNtfy_LINm_ComError (unsigned short u16a_errorinfo)	
Parameters	In	unsigned short u16a_errorinfo: Error information value - 00xxH: Communication error has occurred. "xxH": LEST0 register value - 4000H: Illegal interrupt (INTLIN0TRM) has occurred. - 8000H: Illegal interrupt (INTLIN0RVC) has occurred.
	Out	None
Local variables	- (User processing)	
Return value	- (User processing)	
Call functions	- (User processing)	
Description	This function notifies that an error has occurred.	

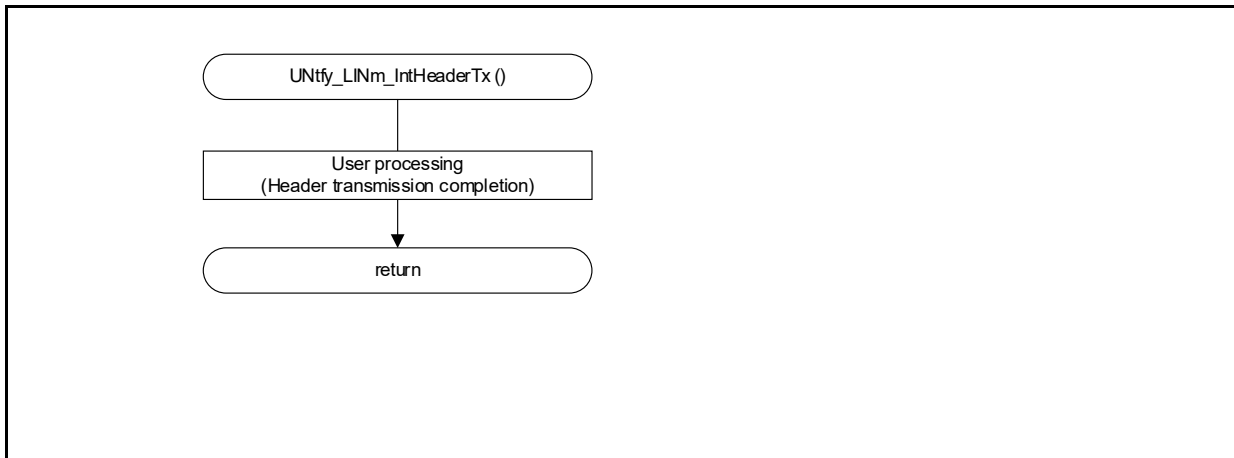


**Figure 3-16 Example of Error Notification Procedure (UNtfy\_LINm\_ComError)**

### 3.5.2 Header Transmission Completion Notification

**Table 3-19 Header Transmission Completion Notification Function API (UNtfy\_LINm\_IntHeaderTx)**

Syntax	void UNtfy_LINm_IntHeaderTx (void)	
Parameters	In	None
	Out	None
Local variables	- (User processing)	
Return value	- (User processing)	
Call functions	- (User processing)	
Description	This function notifies that the header transmission is completed.	

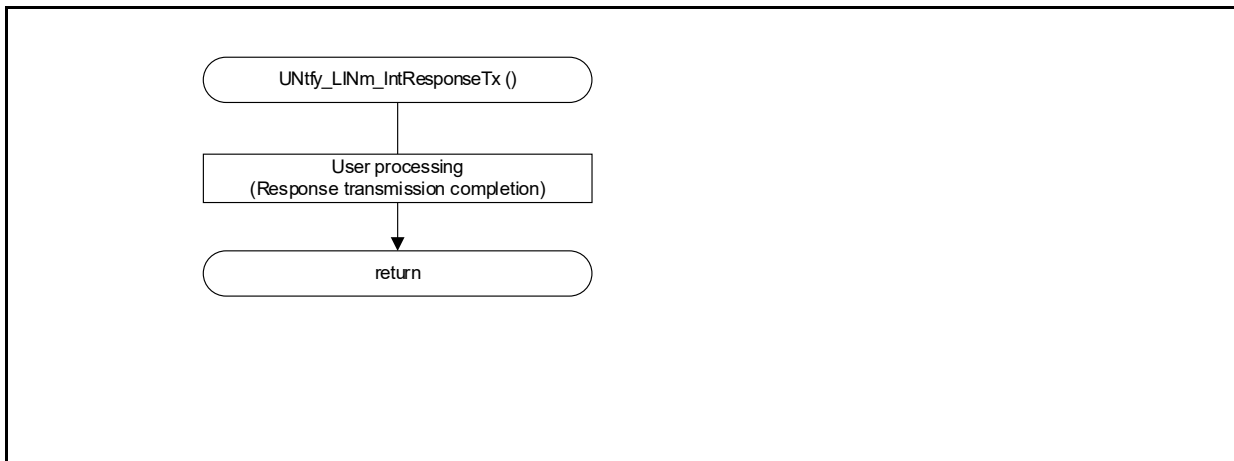


**Figure 3-17 Example of Header Transmission Completion Notification Procedure (UNtfy\_LINm\_IntHeaderTx)**

3.5.3 Response Transmission Completion Notification

**Table 3-20 Response Transmission Completion Notification Function API (UNtfy\_LINm\_IntResponseTx)**

Syntax	void UNtfy_LINm_IntResponseTx (void)	
Parameters	In	None
	Out	None
Local variables	- (User processing)	
Return value	- (User processing)	
Call functions	- (User processing)	
Description	This function notifies that the response transmission is completed.	

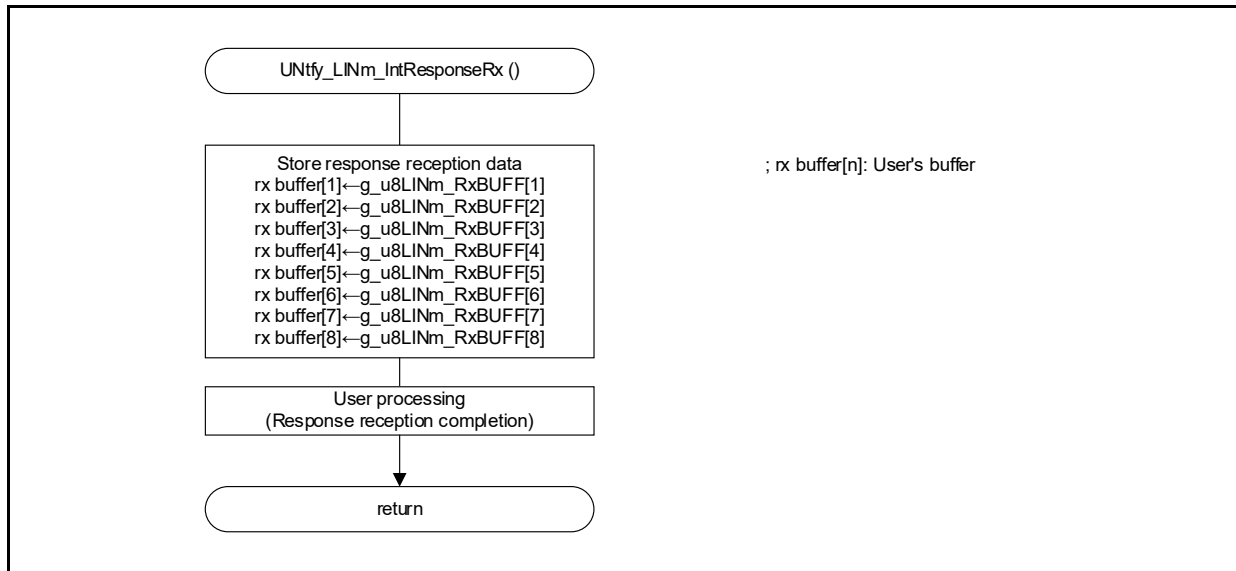


**Figure 3-18 Example of Response Transmission Completion Notification Procedure (UNtfy\_LINm\_IntResponseTx)**

### 3.5.4 Response Reception Completion Notification

**Table 3-21 Response Reception Completion Notification Function API (UNtfy\_LINm\_IntResponseRx)**

Syntax	void UNtfy_LINm_IntResponseRx (void)	
Parameters	In	unsigned char g_u8LINm_RxBUFF[1:8]: Response reception data
	Out	None
Local variables	- (User processing)	
Return value	- (User processing)	
Call functions	- (User processing)	
Description	This function notifies that the response reception is completed.	

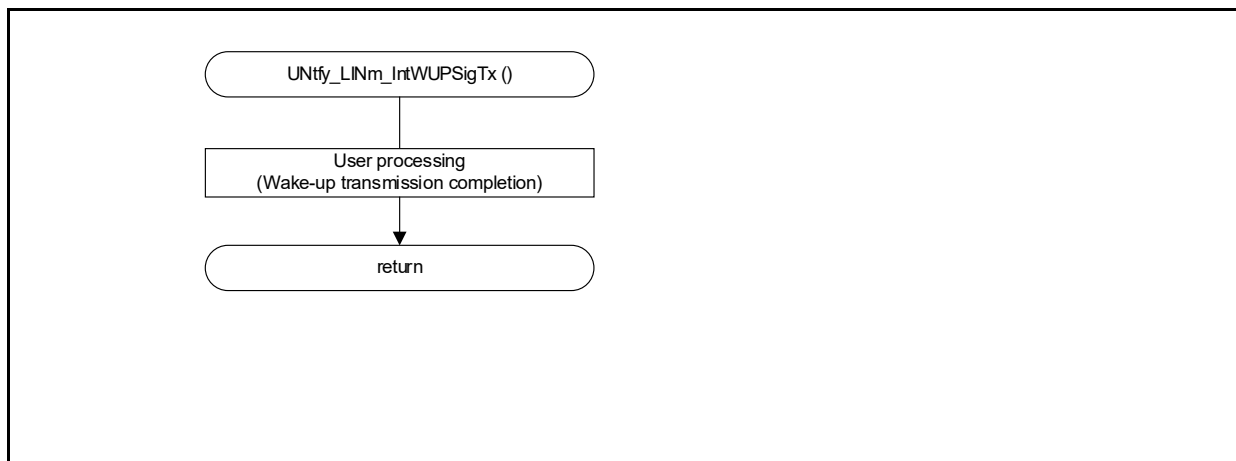


**Figure 3-19 Example of Response Reception Completion Notification Procedure (UNtfy\_LINm\_IntResponseRx)**

### 3.5.5 Wake-up Signal Transmission Completion Notification

**Table 3-22 Wake-up Signal Transmission Completion Notification Function API (UNtfy\_LINm\_IntWUPSigTx)**

Syntax	void UNtfy_LINm_IntWUPSigTx (void)	
Parameters	In	None
	Out	None
Local variables	- (User processing)	
Return value	- (User processing)	
Call functions	- (User processing)	
Description	This function notifies that the wake-up signal transmission is completed.	



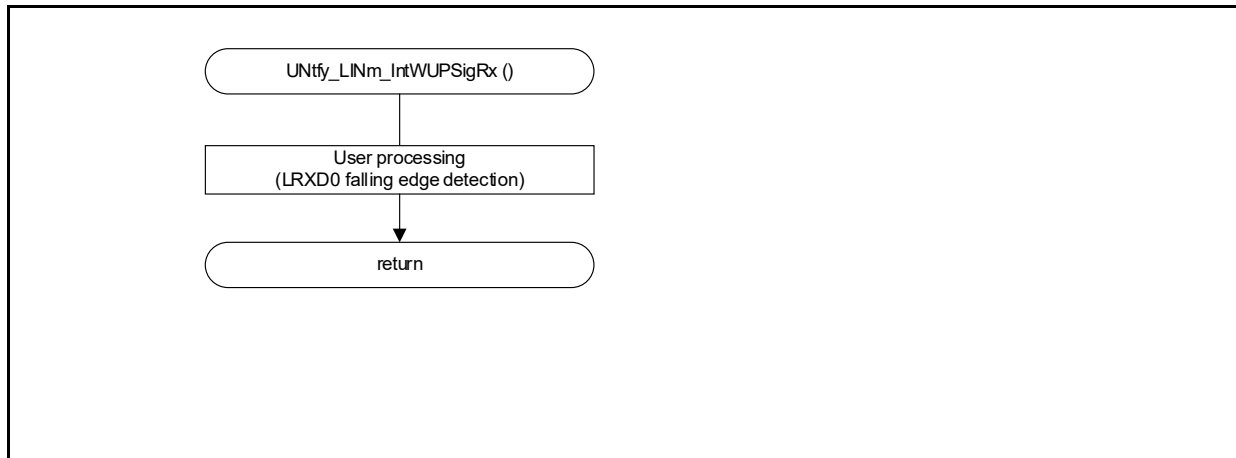
**Figure 3-20 Example of Wake-up Signal Transmission Completion Notification Procedure (UNtfy\_LINm\_IntWUPSigTx)**



### 3.5.6 Wake-up Signal Detection Notification

**Table 3-23 Wake-up Signal Reception Notification Function API (UNtfy\_LINm\_IntWUPSigRx)**

Syntax	void UNtfy_LINm_IntWUPSigRx (void)	
Parameters	In	None
	Out	None
Local variables	- (User processing)	
Return value	- (User processing)	
Call functions	- (User processing)	
Description	This function notifies that the LRXD0 falling edge is detected.	



**Figure 3-21 Example of Wake-up Signal Detection Notification Procedure (UNtfy\_LINm\_IntWUPSigRx)**

#### 4. Operation Mode Transition of LIN / UART Module (RLIN3)

Operation mode transition can be confirmed with OMM0, OMM1 bits of LMSTn register, after setting OM0, OM1 bits of LCUCn register as operation mode selection.

When setting LIN related registers after mode transition, be sure to confirm the completion of mode transition.

The maximum time of the mode transitions are shown below.

**Table 4-1 LIN / UART Module (RLIN3) Mode Transition Times**

Mode Transition	Transition Conditions	Transition Time <sup>Note</sup>
LIN reset mode → LIN operation mode	Set: LCUCn.[OM1,OM0]=11B Check: LMSTn.[OMM1, OMM0]=11B	(Max.) $3 \cdot f_{CLK} + 4 \cdot LINn$ communication clock source
LIN reset mode → LIN wake-up mode	Set: LCUCn.[OM1,OM0]=01B Check: LMSTn.[OMM1, OMM0]=01B	
LIN operation mode → LIN reset mode	Set: LCUCn.[OM1,OM0]=00B	
LIN wake-up mode → LIN reset mode	Check: LMSTn.[OMM1, OMM0]=00B	

**Note.** The time it takes for LMSTn.[OMM1,OMM0] to change after setting LCUCn.[OM1,OM0].

## 5. References

The documents referenced in this application note are shown below. When referring to these documents, make sure to obtain the latest version of each document from Renesas Electronics website.

- RL78/ F13, F14 User's Manual: Hardware Rev.2.10
- RL78/F15 User's Manual: Hardware Rev.1.00

## 6. Appendix

The following table shows the correspondence between header ID numbers (0 to 63) and ID values (with parity added) to be actually transmitted.

**Table 6-1 LIN Header ID List**

Header ID		Actual transmitted ID value <sup>(Note)</sup>	Response data length	Header ID		Actual transmitted ID value <sup>(Note)</sup>	Response data length
Decimal	Hexadecimal	Hexadecimal		Decimal	Hexadecimal	Hexadecimal	
0	0x00	0x80	2	32	0x20	0x20	4
1	0x01	0xC1	2	33	0x21	0x61	4
2	0x02	0x42	2	34	0x22	0xE2	4
3	0x03	0x03	2	35	0x23	0xA3	4
4	0x04	0xC4	2	36	0x24	0x64	4
5	0x05	0x85	2	37	0x25	0x25	4
6	0x06	0x06	2	38	0x26	0xA6	4
7	0x07	0x47	2	39	0x27	0xE7	4
8	0x08	0x08	2	40	0x28	0xA8	4
9	0x09	0x49	2	41	0x29	0xE9	4
10	0x0A	0xCA	2	42	0x2A	0x6A	4
11	0x0B	0x8B	2	43	0x2B	0x2B	4
12	0x0C	0x4C	2	44	0x2C	0xEC	4
13	0x0D	0x0D	2	45	0x2D	0xAD	4
14	0x0E	0x8E	2	46	0x2E	0x2E	4
15	0x0F	0xCF	2	47	0x2F	0x6F	4
16	0x10	0x50	2	48	0x30	0xF0	8
17	0x11	0x11	2	49	0x31	0xB1	8
18	0x12	0x92	2	50	0x32	0x32	8
19	0x13	0xD3	2	51	0x33	0x73	8
20	0x14	0x14	2	52	0x34	0xB4	8
21	0x15	0x55	2	53	0x35	0xF5	8
22	0x16	0xD6	2	54	0x36	0x76	8
23	0x17	0x97	2	55	0x37	0x37	8
24	0x18	0xD8	2	56	0x38	0x78	8
25	0x19	0x99	2	57	0x39	0x39	8
26	0x1A	0x1A	2	58	0x3A	0xBA	8
27	0x1B	0x5B	2	59	0x3B	0xFB	8
28	0x1C	0x9C	2	60	0x3C	0x3C	8
29	0x1D	0xDD	2	61	0x3D	0x7D	8
30	0x1E	0x5E	2	(62)	(reserved)	(reserved)	—
31	0x1F	0x1F	2	(63)	(reserved)	(reserved)	—

**Note:** P0(bit6) = ID0⊕ID1⊕ID2⊕ID4, P1(bit7) = !(ID1⊕ID3⊕ID4⊕ID5), P: Parity, ⊕: XOR, !: INV

**Revision History**

Rev.	Date	Description	
		Page	Summary
1.00	Sep.30.21	-	First edition issued

## 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 not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.  
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space 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.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(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 November 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://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/](http://www.renesas.com/contact/).