

RL78/F23, F24

R01AN6626EJ0100

Rev.1.00

Setup Procedure for LIN Communication in Slave Mode (Guidance)

2022.09.30

Introduction

This application note describes setup procedures for LIN communication, in which a device operates in slave mode of the UART/LIN module (hereinafter, RLIN3) in the RL78/F23, F24.

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/F23, F24 products

Contents

1. Specifications of LIN Communication in Slave Mode	3
1.1 LIN Slave Mode Usage Conditions	3
1.2 LIN Network Connection Example.....	4
1.3 SFR	5
2. Overview of LIN Slave Communication.....	8
2.1 LIN Slave Communication Processing Overview	8
2.2 LIN Frame Timings	10
2.3 LIN Slave Communication Status and SFR	12
3. LIN Slave Driver Functions	13
3.1 Function list	13
3.2 Driver API Functions [Public].....	15
3.2.1 LIN Slave Module Initialization	15
3.2.2 LIN Module De-Initialization	16
3.2.3 LIN Slave Header Reception Enable	17
3.2.4 Before Sleep Mode Transition Processing	18
3.2.5 LIN Slave Wake-up Signal Transmission	19
3.2.6 LIN Slave Response Transmission Start.....	20
3.2.7 LIN Slave Response Reception Start.....	21
3.2.8 LIN Slave No Response Request.....	22
3.3 Driver Local Functions [Private]	23
3.3.1 LIN0 Port Initialization	23
3.3.2 LIN0 Interrupt Initialization.....	24
3.3.3 Transition to LIN Reset Mode.....	25
3.3.4 Transition to LIN Operation Mode	26
3.3.5 Transition to LIN Wake-up Mode.....	27
3.4 Interrupt Processing Functions	28
3.4.1 LIN Slave Reception Completion Interrupt Processing	28
3.4.2 LIN Slave Transmission Completion Interrupt Processing	30
3.4.3 LIN Slave Status Error Interrupt Processing	31
3.4.4 LIN Slave Wake-up Signal Detection Interrupt Processing	32
3.5 User Notification Function APIs.....	33
3.5.1 Error Notification.....	33
3.5.2 Header Reception Completion Notification	34
3.5.3 Response Transmission Completion Notification	35
3.5.4 Response Reception Completion Notification	36
3.5.5 Wake-up Signal Transmission Completion Notification.....	37
3.5.6 Wake-up Signal Reception Notification	38
4. Operation Mode Transition of LIN / UART Module (RLIN3).....	39
5. References	40
Revision History.....	41

1. Specifications of LIN Communication in Slave Mode

1.1 LIN Slave Mode Usage Conditions

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

Table 1-1 LIN Slave Mode Usage Conditions

Item		Description
CPU/peripheral hardware clock frequency (f_{CLK})		40 MHz
LIN clock source (f_{LIN})		f_{CLK}
LTXD0		P13/LTXD0
LRXD0		P14/LRXD0
Baud rate		19200 bps
Mode		Slave mode (Fixed baud rate or auto baud rate)
Noise filter		Enabled
LIN interrupts		INTLIN0TRM, INTLIN0RVC, INTLIN0STA, INTLIN0WUP
Checksum type		Classic checksum or Enhanced checksum (User selected)
Error detection	Bit error	Enabled
	Frame/Response timeout error	Auto baud rate mode: Disabled
		Fixed baud rate mode: Enabled
	Framing error	Enabled
	Sync field error	Enabled
	Checksum error	Enabled
	ID parity error	Enabled
Response preparation error		Enabled
Number of bytes in frame's data field		0 to 8 bytes
Response space		1 Tbit
Inter-byte space		1 Tbit

1.2 LIN Network Connection Example

The LIN network connection is shown below.

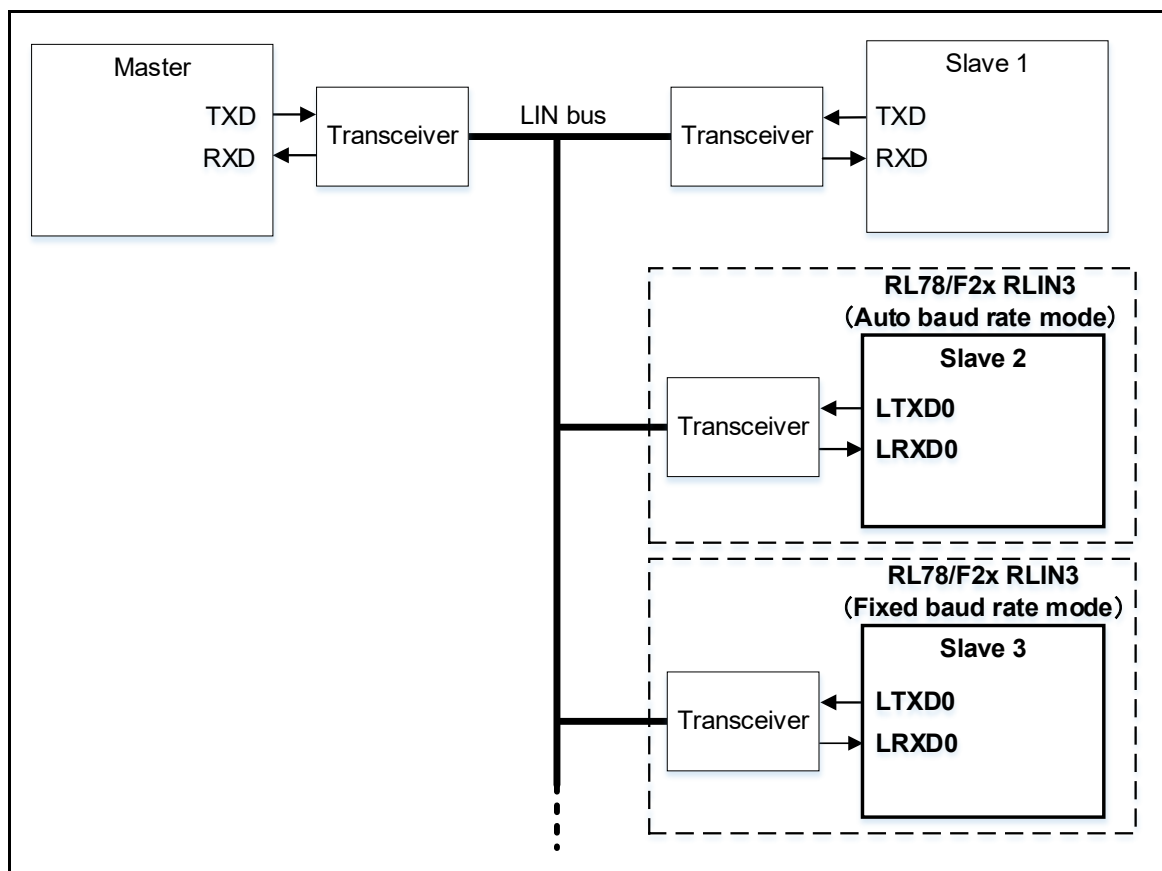


Figure 1-1 LIN Master - Slaves Network Connection (example)

1.3 SFR

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

Table 1-2 SFRs Used for LIN Communication in Slave 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} ^{Note}) 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	Auto baud rate mode: 74H	LPRS[2:0]=010B (LIN0 prescaler clock: 1/4 (10MHz)) NSPB[3:0]=0111B (Bit sampling: 8 sampling)
		Fixed baud rate mode: 00H	LPRS[2:0]=000B (LIN0 prescaler clock: 1/1 (40MHz)) NSPB[3:0]=0000B (Bit sampling: 16 sampling)
LIN/UART baud rate prescaler register 0	LBRP0	0081H	19200bps In auto baud rate mode: Updated with sync field reception
LIN self-test control register 0	LSTC0	00H	LSTM=0 (LIN self-test mode is disabled)
LIN/UART mode register 0	LMD0	Auto baud rate mode: 12H 13H (Wake-up transmission)	LMD[1:0]=10B (LIN slave mode [auto baud rate]) LIOS=1 (Enable LIN0 transmission/reception/status interrupt) LRDNFS=0 (LRXD0 noise filter enabled)
		Fixed baud rate mode: 13H	LMD[1:0]=11B (LIN slave mode [fixed baud rate]) LIOS=1 (Enable LIN0 transmission/reception/status interrupt) LRDNFS=0 (LRXD0 noise filter enabled)
LIN/UART break field configuration register 0	LBFC0	00H	BLT=0 (Break (low width) detection: 10 Tbit)
LIN/UART space configuration register 0	LSC0	10H	RS[2:0]=000B (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 reception completion interrupt: Enabled)
LIN/UART error detection enable register 0	LEDE0	Auto baud rate mode: D9H	BERE=1 (Bit error detection: Enabled) TERE=0 (Response timeout error detection: Disabled) FERE=1 (Framing error detection: Enabled) SFERE=1 (Sync field error detection: Enabled) IPERE=1 (ID parity error detection: Enabled) LTES=1 (Timeout selection bit: Response timeout)
		Fixed baud rate mode: DDH	BERE=1 (Bit error detection: Enabled) TERE=1 (Response timeout error detection: Enabled) FERE=1 (Framing error detection: Enabled) SFERE=1 (Sync field error detection: Enabled) IPERE=1 (ID parity 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, 04H	LTRC0=01H (Start Header reception / Wake-up transmission) LTRC0=02H (Start response transmission/reception) LTRC0=04H (No response request)
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 reception completion)

Note: For fixed baud rate mode, select X1 oscillator or PLL clock (source=X1) for communication clock (f_{CLK}).

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

Register		Setting Value	Conditions
Name	Symbol		
LIN/UART error status register 0	LEST0	Read 00H (Error clear)	Read error status flag Bit 0: BER (1: Bit error detection) Bit 2: TER (1: Response timeout error detection in fixed baud rate mode) Bit 3: FER (1: Framing error detection) Bit 4: SFER (1: Sync field error detection) Bit 5: CSER (1: Checksum error detection) Bit 6: IPER (1: ID parity error detection) Bit 7: RPER (1: Response preparation error detection)
LIN/UART data field configuration register 0	LDFO0	xxH	RFDL[3:0]: Response data length (0 to 8) RCDS: Response direction (0: Reception, 1: Transmission) LCS: Checksum type (0: Classic checksum, 1: Enhanced checksum) LSS=0 (Next transmission/reception data group is the last one)
LIN/UART ID buffer register 0	LIDB0	Read	Received ID + ID parity
LIN checksum buffer register 0	LCBR0	xxH	Read reception checksum ^{Note}
LIN/UART data buffer [1-8] registers	LDB01 – LDB08	xxH	Set transmit DATA[n], or read reception DATA[n]
LIN break and sync field detection status register 0	LBSS0	-	LIN break and sync field detection flag Bit 0: BRKC (1: Break field signal detection) ^{Note} Bit 1: SYCC (1: Sync field signal detection) ^{Note}
LIN response space dominant signal detection status register 0	LRSS0	-	LIN response space dominant signal detection flag Bit 0: RSDD (1: Detected dominant level in response space) ^{Note}

Note: LCBR0, LBSS0, and LRSS0 registers are not used in this application note. Checksum error is checked by checksum error detection.

Table 1-3 SFRs Used for LIN Communication in Slave 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=1	Port P1.4: Internal pull-up enabled

Table 1-4 SFRs Used for LIN Communication in Slave 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 Slave Communication

2.1 LIN Slave Communication Processing Overview

LIN slave communication example flow is shown below.

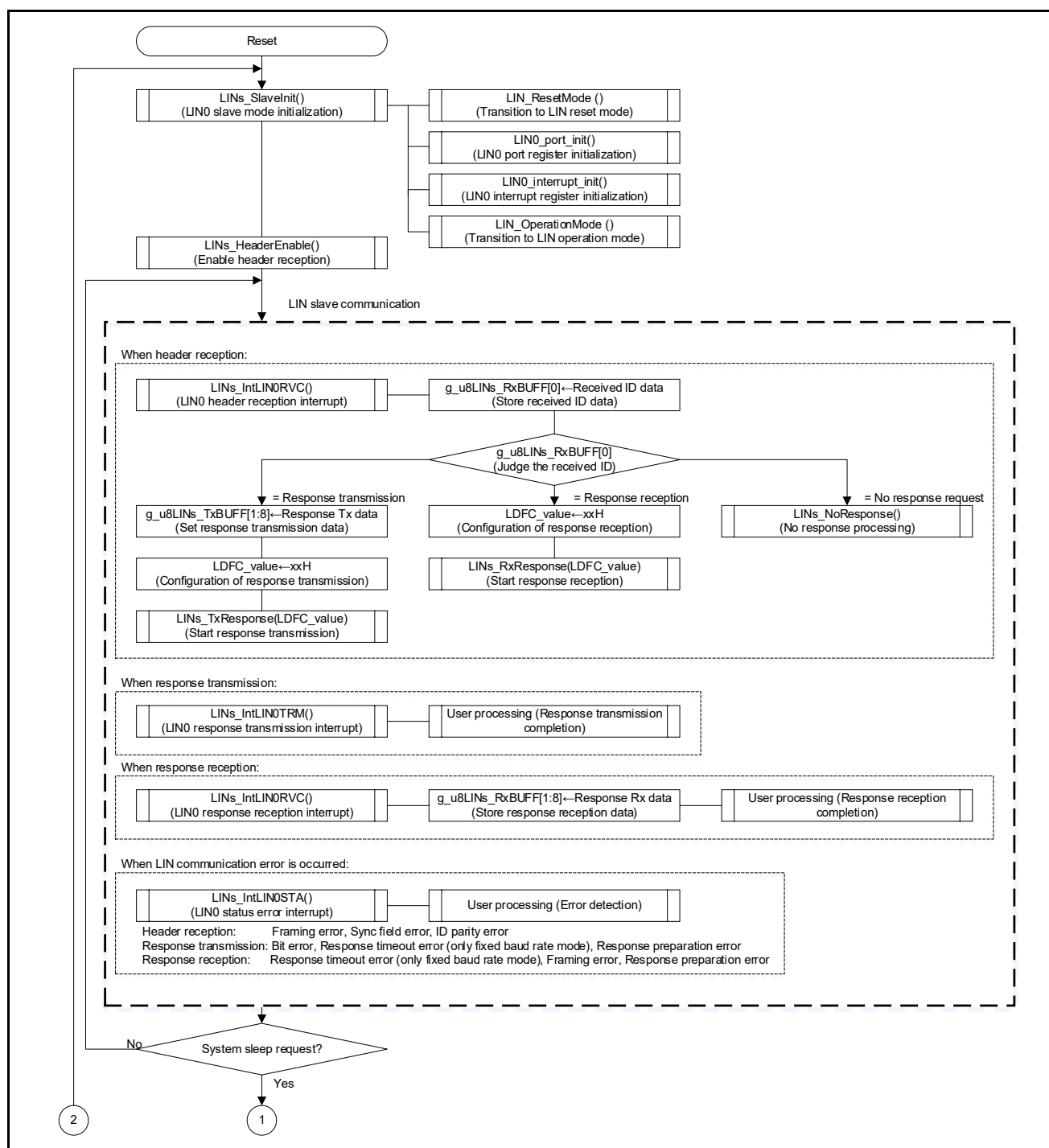


Figure 2-1 LIN Communication Process in Slave Mode (1/2)

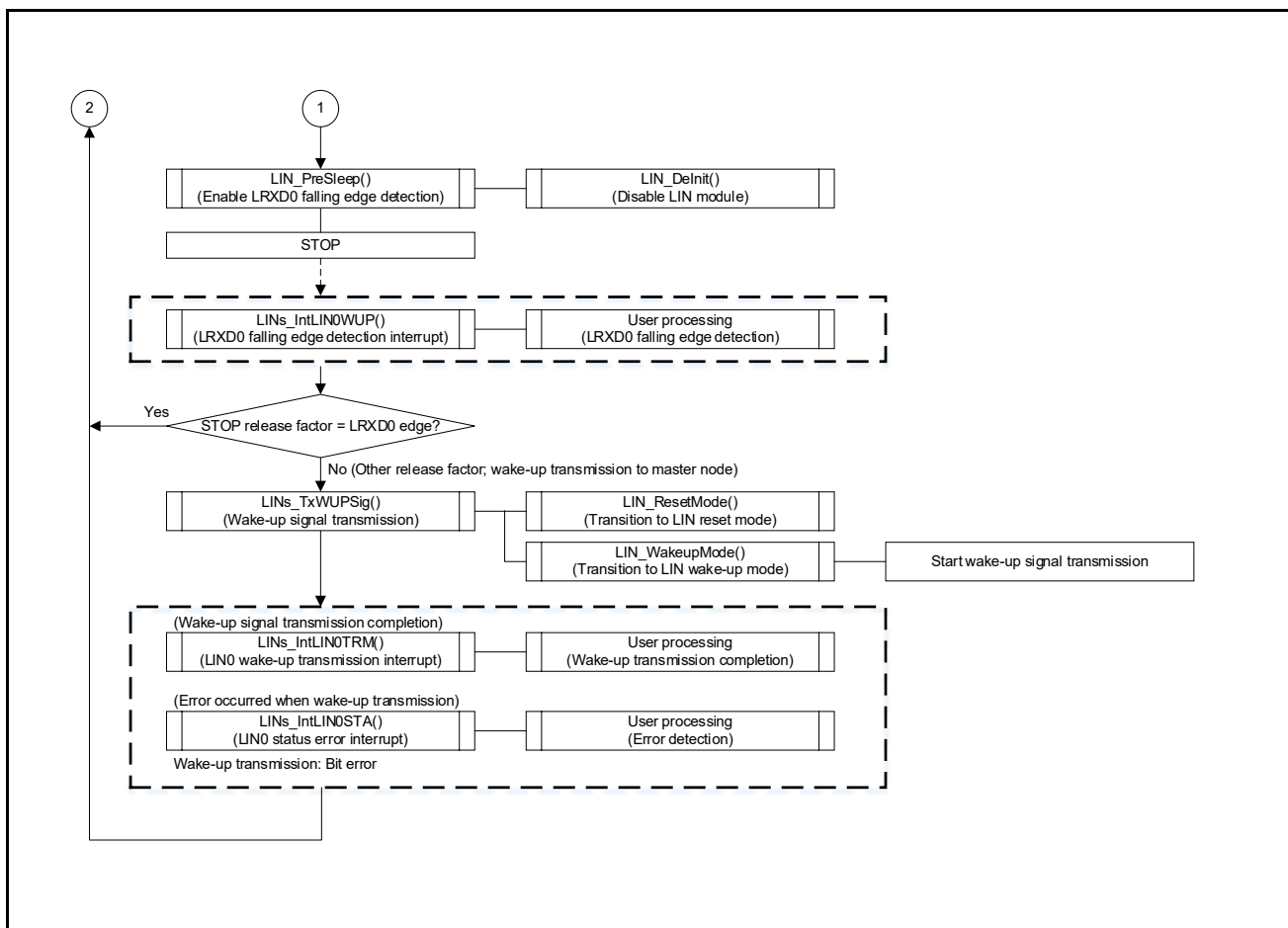


Figure 2-1 LIN Communication Process in Slave Mode (2/2)

2.2 LIN Frame Timings

The timing diagram for response transmission and reception in the LIN slave are shown below for each of the Auto baud rate and Fixed baud rate.

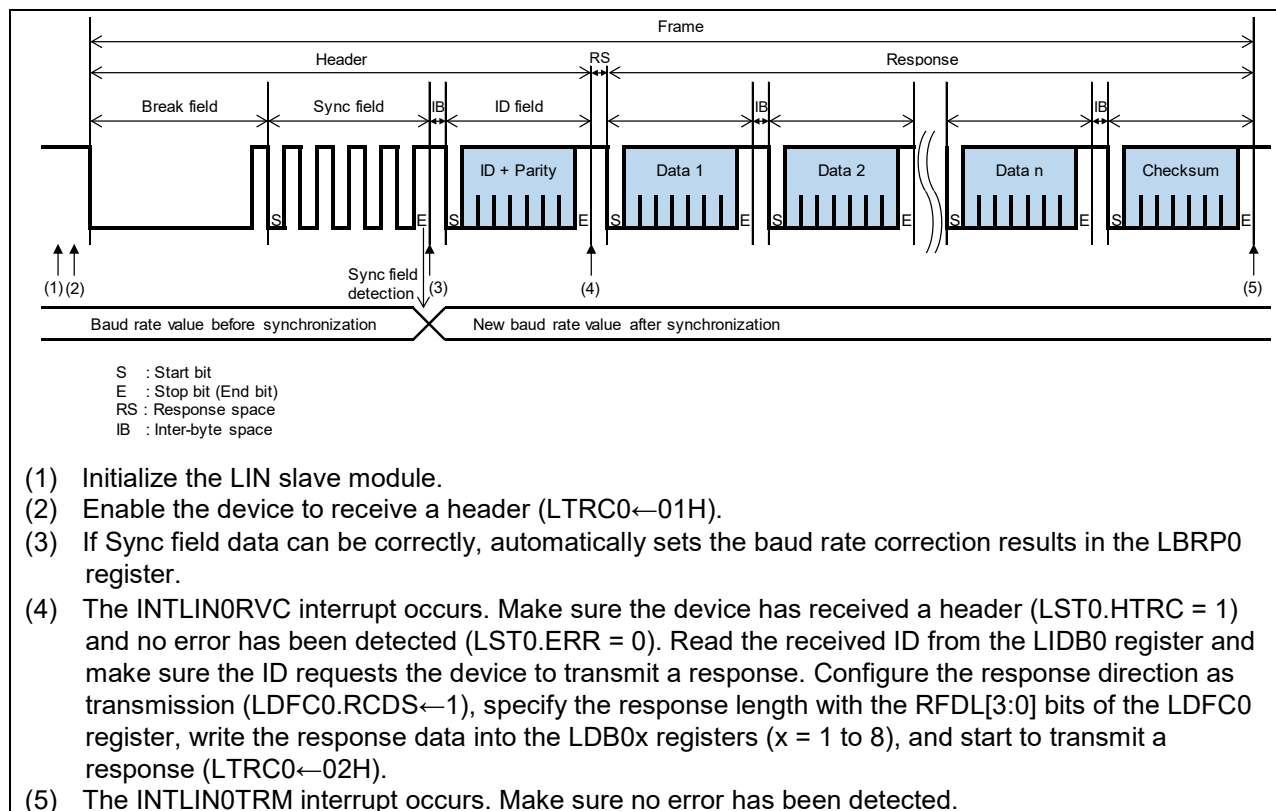


Figure 2-2 Timing Diagram of LIN Response Transmission in Slave Mode (Auto Baud Rate)

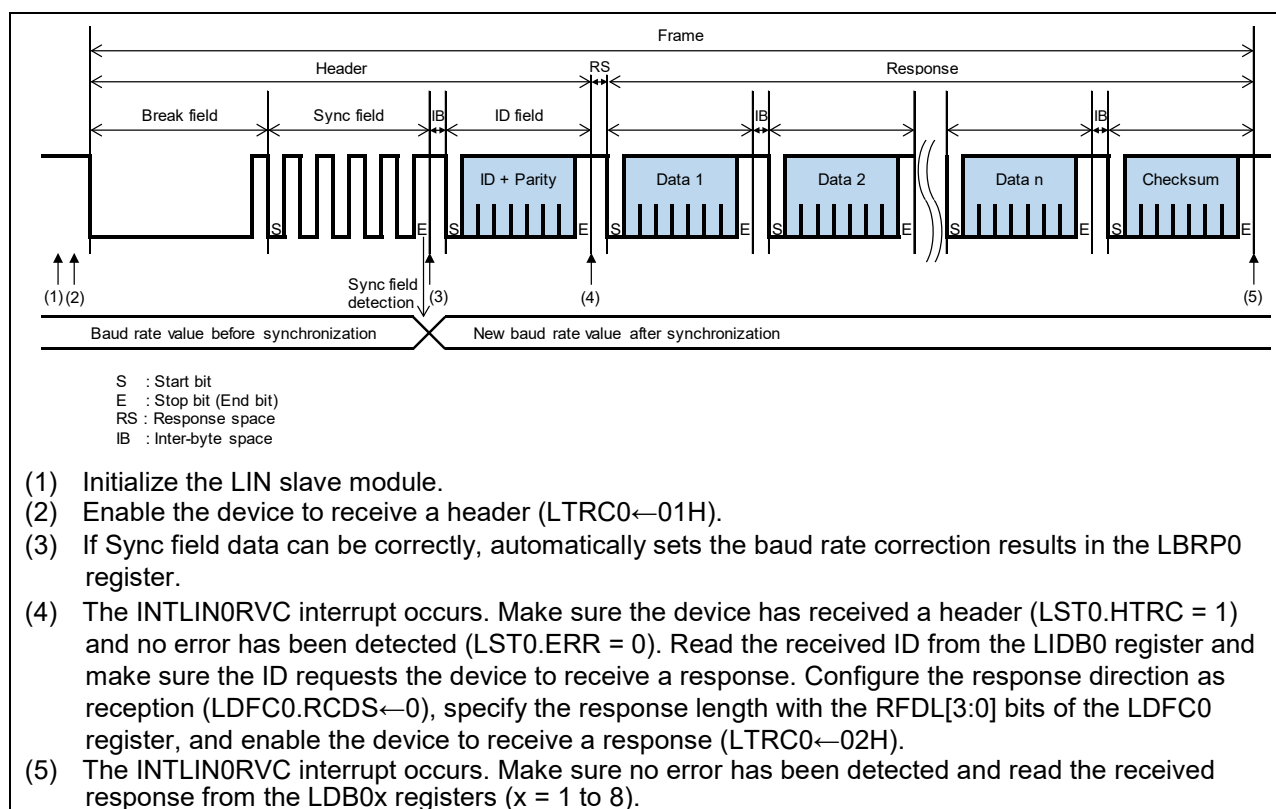


Figure 2-3 Timing Diagram of LIN Response Reception in Slave Mode (Auto Baud Rate)

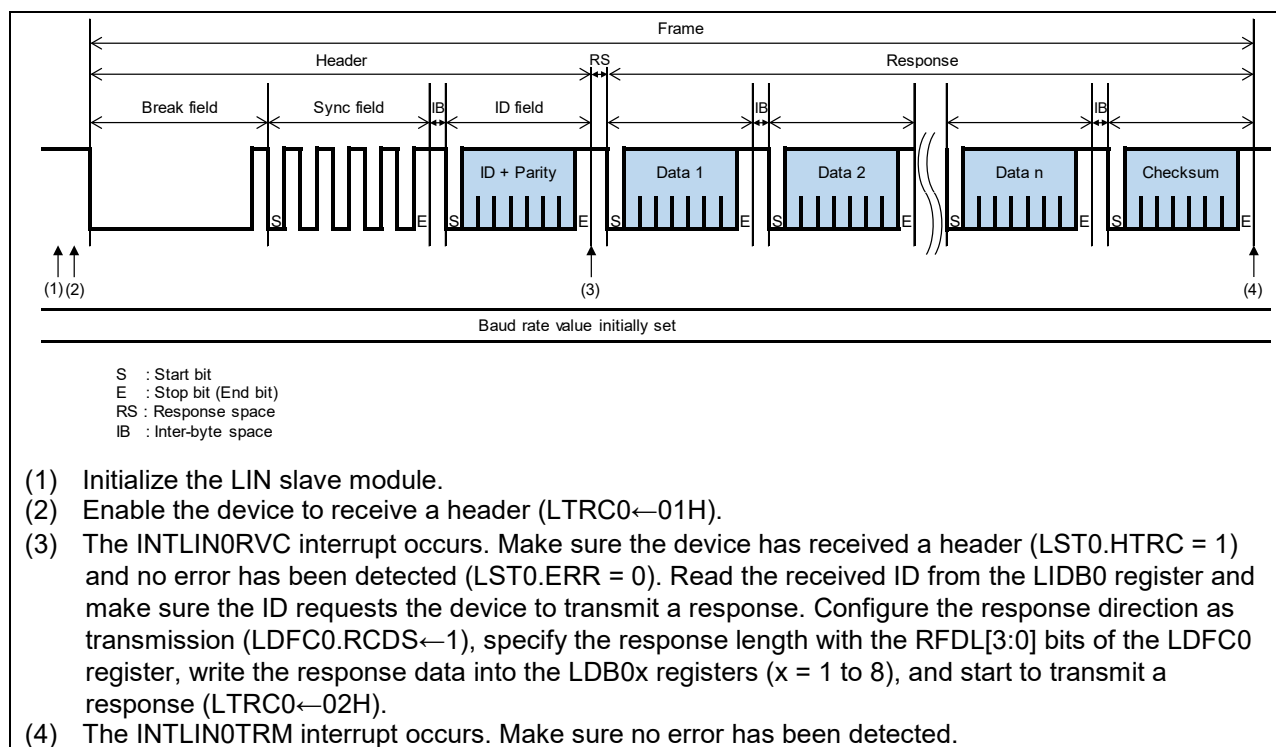


Figure 2-4 Timing Diagram of LIN Response Transmission in Slave Mode (Fixed Baud Rate)

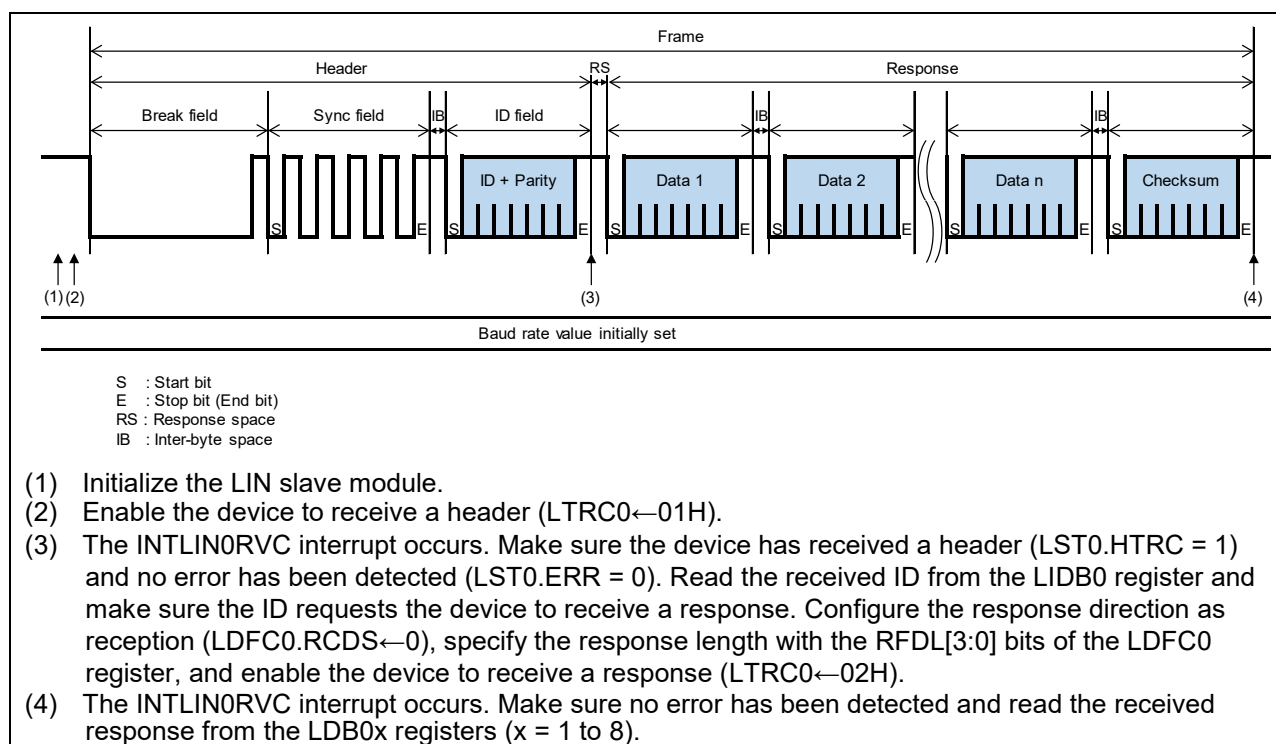


Figure 2-5 Timing Diagram of LIN Response Reception in Slave Mode (Fixed Baud Rate)

2.3 LIN Slave 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 slave communication status and SFRs are shown below.

After receiving a header, if a dominant level is detected in the LIN bus during the response space until the start of response transmission/reception, the RSDD bit of the LRSSn register becomes to "1" (dominant level detected in the response space). Error handling can also be performed by judging the RSDD bit at the start of response transmission/reception. RSDD bit is not used in this application note.

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

LIN Communication State Reception	SFRs		
	LST0		
	bit3(ERR)	bit7(HTRC)	bit1(FRC)
Communication error (Processed in the LIN0 status interrupt)	1	-	-
Header reception completion	0	1	-
Response reception completion		0	1
Illegal interrupt occurred			0

Remarks 1. -: Not judged by software.
2. For details, see section 3.4.1.

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

LIN Communication State Transmission	SFRs		
	LST0		LMST0
	bit3(ERR)	bit0(FTC)	
Communication error (Processed in the LIN0 status interrupt)	1	-	-
Response transmission completion	0	1	01H
Wake-up transmission completion		0	≠01H
Illegal interrupt occurred			-

Remarks 1. -: Not judged by software.
2. For details, see section 3.4.2.

3. LIN Slave Driver Functions

The LIN slave 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.

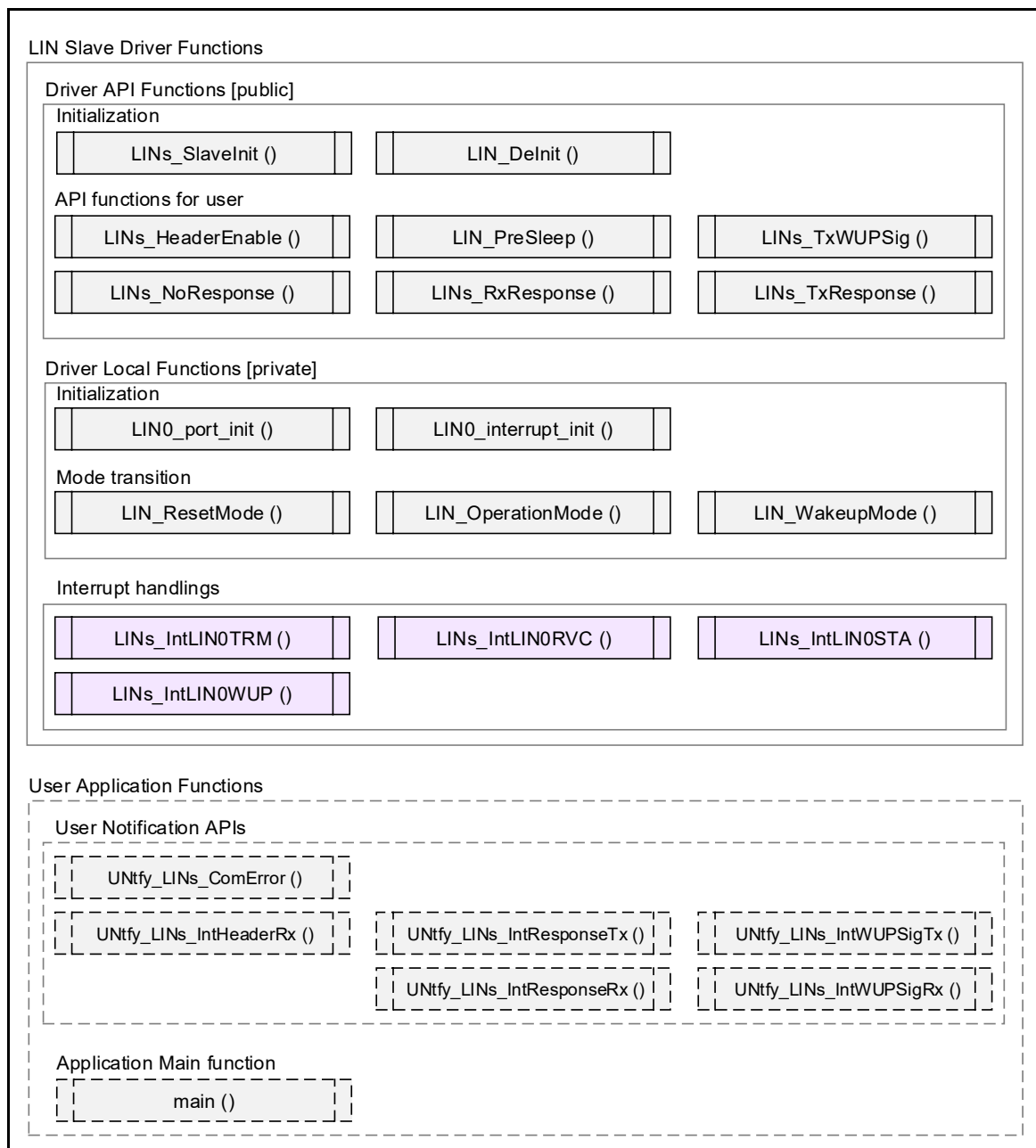


Figure 3-1 Functions of LIN Communication in Slave Mode

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

Function Name	Description	Public / Private
void LINs_SlaveInit (void)	Initialization for LIN (slave mode)	public
void LIN_DeInit (void)	Disables LIN module	public
void LINs_HeaderEnable (void)	Enable header reception	public
void LIN_PreSleep (void)	Process before transition to sleep mode	public
void LINs_TxWUPSig (void)	LIN wake-up signal transmission request	public
void LINs_TxResponse (unsigned char u8a_LDFC_value)	Response transmission request	public
void LINs_RxResponse (unsigned char u8a_LDFC_value)	Response reception request	public
void LINs_NoResponse (void)	No response request	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 LINs_IntLIN0RVC (void)	This function is the LIN0 reception completion interrupt processing. (Header reception / Response reception / Wake-up reception ^{Note})	public
void LINs_IntLIN0TRM (void)	This function is the LIN0 transmission completion interrupt processing. (Response transmission / Wake-up transmission)	public
void LINs_IntLIN0STA (void)	This function is the interrupt processing when LIN0 status error is occurred. - Response transmission: Bit error or Response timeout error detection - Header reception / Response reception: Response timeout error, Framing error, Sync field error, Checksum error or ID parity error detection	public
void LINs_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 Slave Mode

Variable Name	Description
unsigned char g_u8LINs_TxBUFF[9]	g_u8LINs_TxBUFF[0]: Not used g_u8LINs_TxBUFF[1 to 8]: Set response transmission data
unsigned char g_u8LINs_RxBUFF[9]	g_u8LINs_RxBUFF[0]: Store received ID data g_u8LINs_RxBUFF[1 to 8]: Store response reception data

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

Function API Name	Description
void UNTfy_LINs_ComError (unsigned short u16a_errorinfo)	User's processing for error detections. bit[7:0]: Store the LEST0 register value (When INTLIN0STA occurred) bit[15:0]: When INTLIN0STA occurred: 00H When INTLIN0RVC occurred: 80H (illegal interrupt) When INTLIN0TRM occurred: 40H (illegal interrupt)
void UNTfy_LINs_IntHeaderRx (void)	User's processing for header reception completion.
void UNTfy_LINs_IntResponseTx (void)	User's processing for response transmission completion.
void UNTfy_LINs_IntResponseRx (void)	User's processing for response reception completion.
void UNTfy_LINs_IntWUPSigTx (void)	User's processing for wake-up signal transmission completion.
void UNTfy_LINs_IntWUPSigRx (void)	User's processing for wake-up signal reception completion.

3.2 Driver API Functions [Public]

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

3.2.1 LIN Slave Module Initialization

Table 3-4 LIN Slave Initialization Function (LINs_SlaveInit)

Syntax	void LINs_SlaveInit (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 (slave mode), port and interrupt functions.	
Note	Call from upper layer with interrupt disabled.	

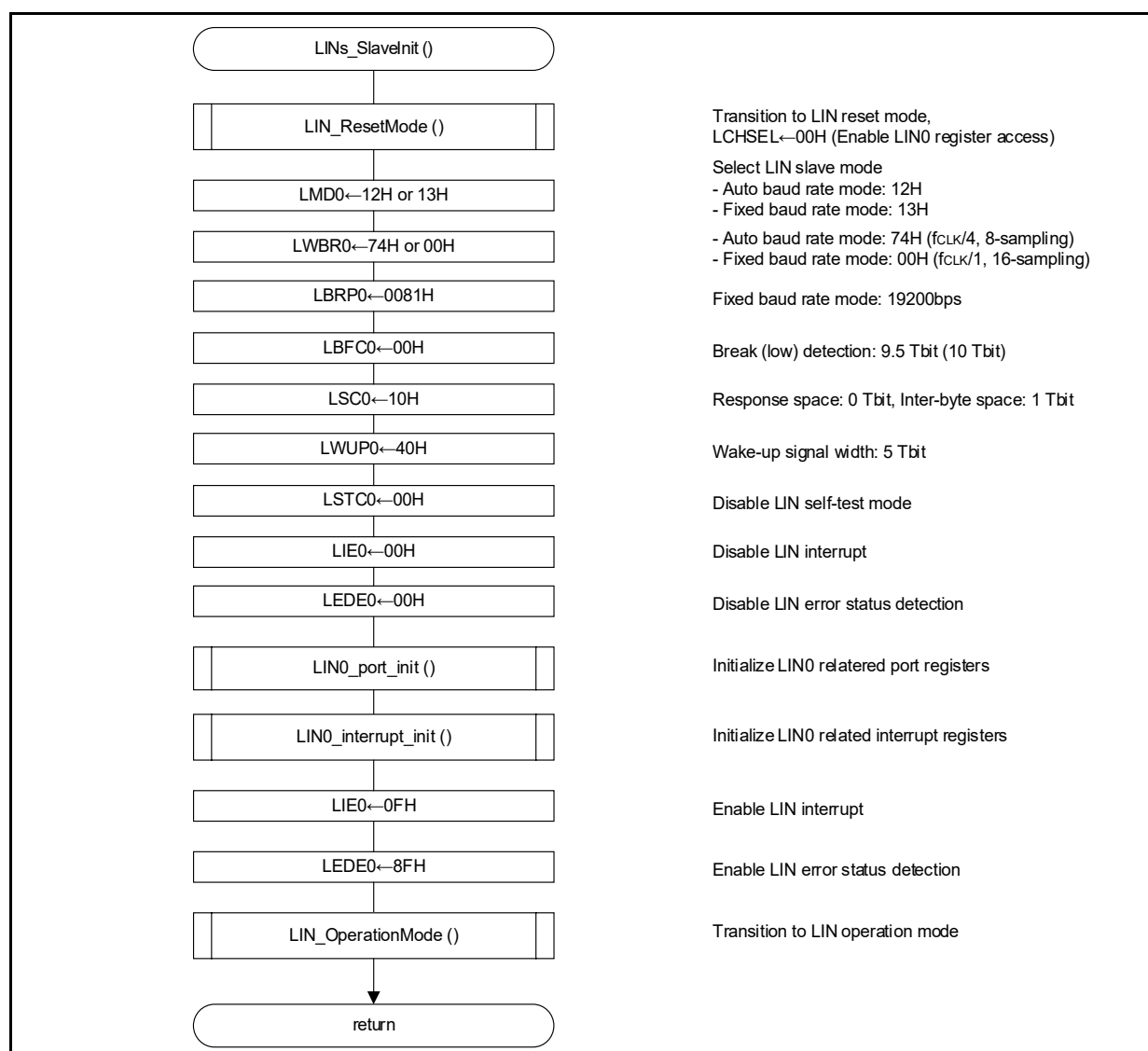


Figure 3-2 LIN Slave Initialization Procedure (LINs_SlaveInit)

3.2.2 LIN Module De-Initialization

Table 3-5 LIN De-Initialization Function (LIN_DeInit)

Syntax	void LIN_DeInit (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.	

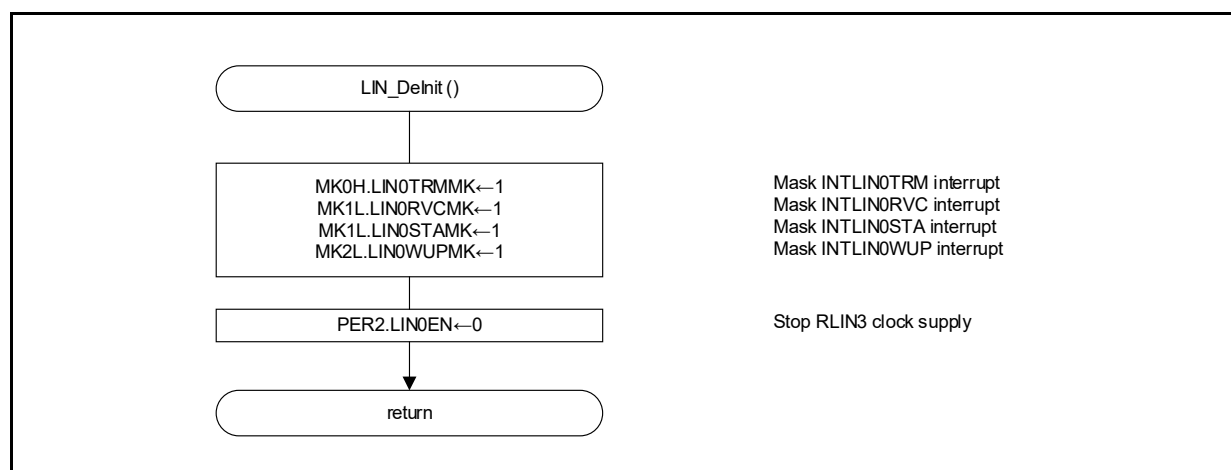


Figure 3-3 LIN De-Initialization Procedure (LIN_DeInit)

3.2.3 LIN Slave Header Reception Enable

Table 3-6 LIN Slave Header Reception Enable Function (LINs_HeaderEnable)

Syntax	void LINs_HeaderEnable (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	This function is used to enable header reception.	
Note	Call this function when GINT bit of IAWCTL register is 0.	

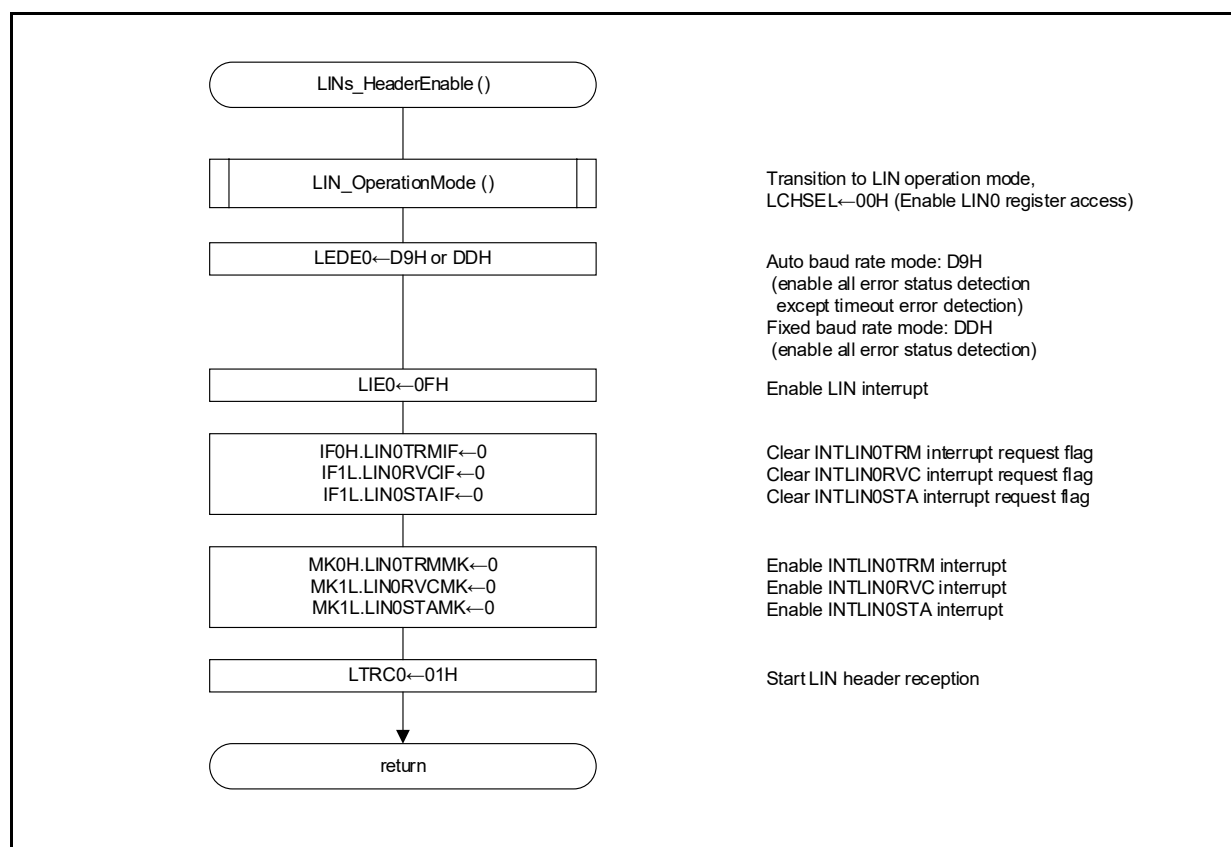


Figure 3-4 LIN Slave Header Reception Enable Procedure (LINs_HeaderEnable)

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.	

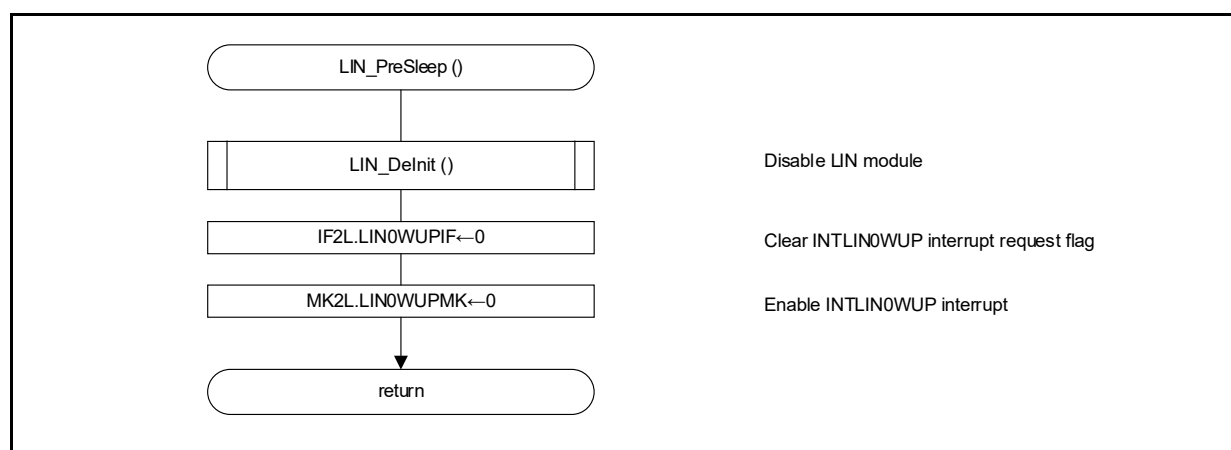


Figure 3-5 Before Sleep Mode Transition Processing Procedure (LIN_PreSleep)

3.2.5 LIN Slave Wake-up Signal Transmission

Table 3-8 LIN Slave Wake-up Signal Transmission Function (LINs_TxWUPSig)

Syntax	void LINs_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.	

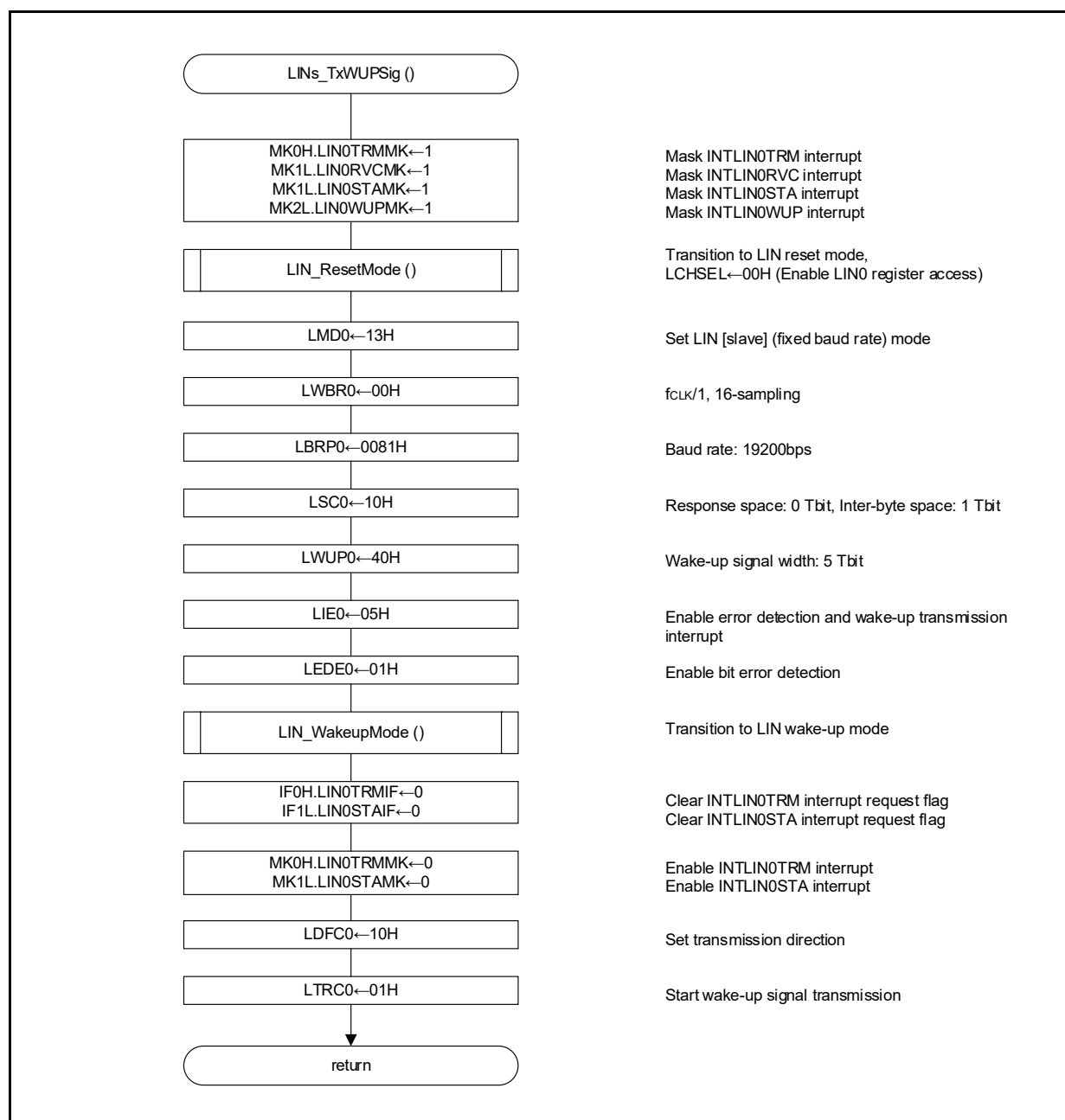


Figure 3-6 LIN Slave Wake-up Signal Transmission Procedure (LINs_TxWUPSig)

3.2.6 LIN Slave Response Transmission Start

Table 3-9 LIN Slave Response Transmission Start Function (LINs_TxResponse)

Syntax		void LINs_TxResponse (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 (RCDS):	Response direction [1: Transmission]
		- bit5 (LCS):	Checksum type [0: classical-checksum, 1: enhanced-checksum]
		- bit6:	Set to 0
		- bit7 (LSS):	0: Next transmission/reception data group is the last one
		unsigned char	g_u8LINs_TxBUFF[1:8]: Response transmission data setting
	Out	None	
Local variables		None	
Return value		None	
Call functions		None	
Description		LIN response transmission start procedure	

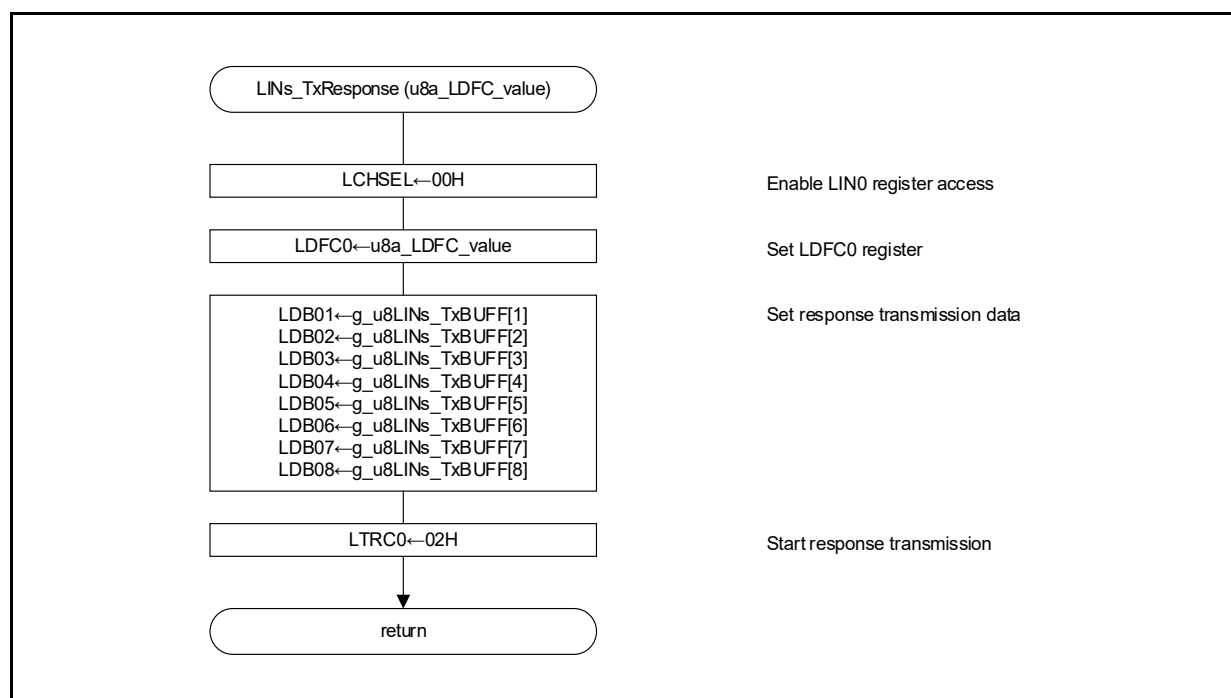


Figure 3-7 LIN Slave Response Transmission Start Procedure (LINs_TxResponse)

3.2.7 LIN Slave Response Reception Start

Table 3-10 LIN Slave Response Reception Start Function (LINS_RxResponse)

Syntax		void LINs_RxResponse (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 (RCDS): Response direction [0: Reception] - bit5 (LCS): Checksum type [0: classical-checksum, 1: enhanced-checksum] - bit6: Set to 0 - bit7 (LSS): 0: Next transmission/reception data group is the last one	
	Out	None	
Local variables		None	
Return value		None	
Call functions		None	
Description		LIN response reception start procedure	

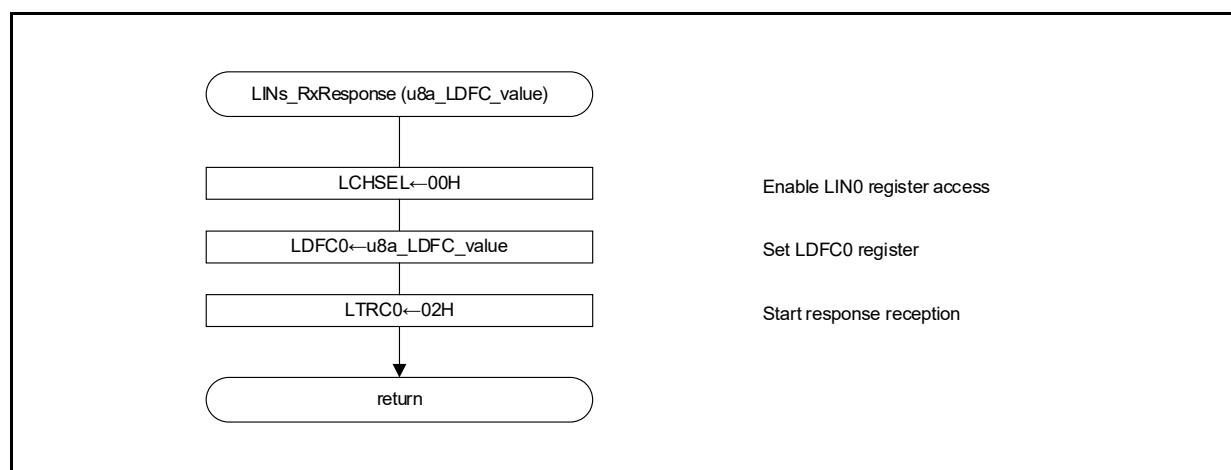


Figure 3-8 LIN Slave Response Reception Start Procedure (LINS_RxResponse)

3.2.8 LIN Slave No Response Request

Table 3-11 LIN Slave No Response Request Function (LINs_NoResponse)

Syntax	void LINs_NoResponse (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	None	
Description	LIN no response request procedure	

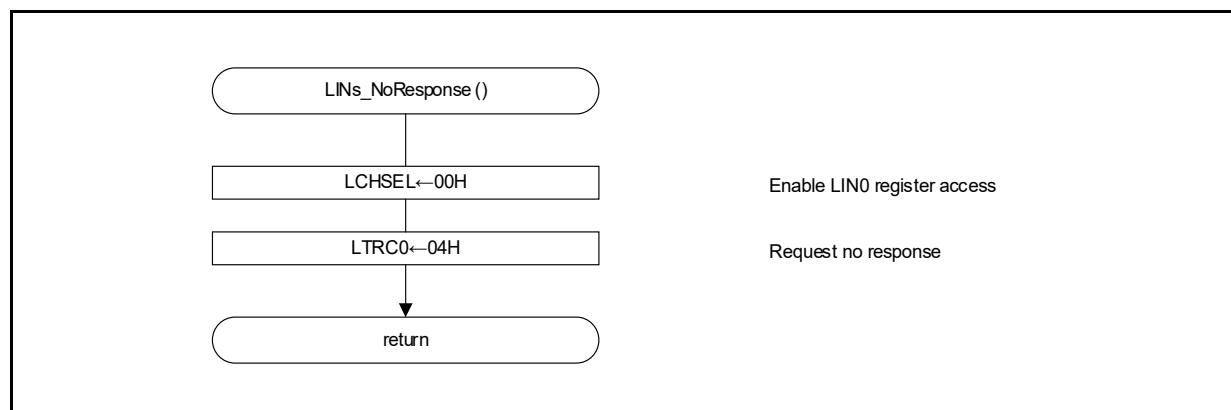


Figure 3-9 LIN Slave No Response Request Procedure (LINs_NoResponse)

3.3 Driver Local Functions [Private]

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

3.3.1 LIN0 Port Initialization

Table 3-12 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.	

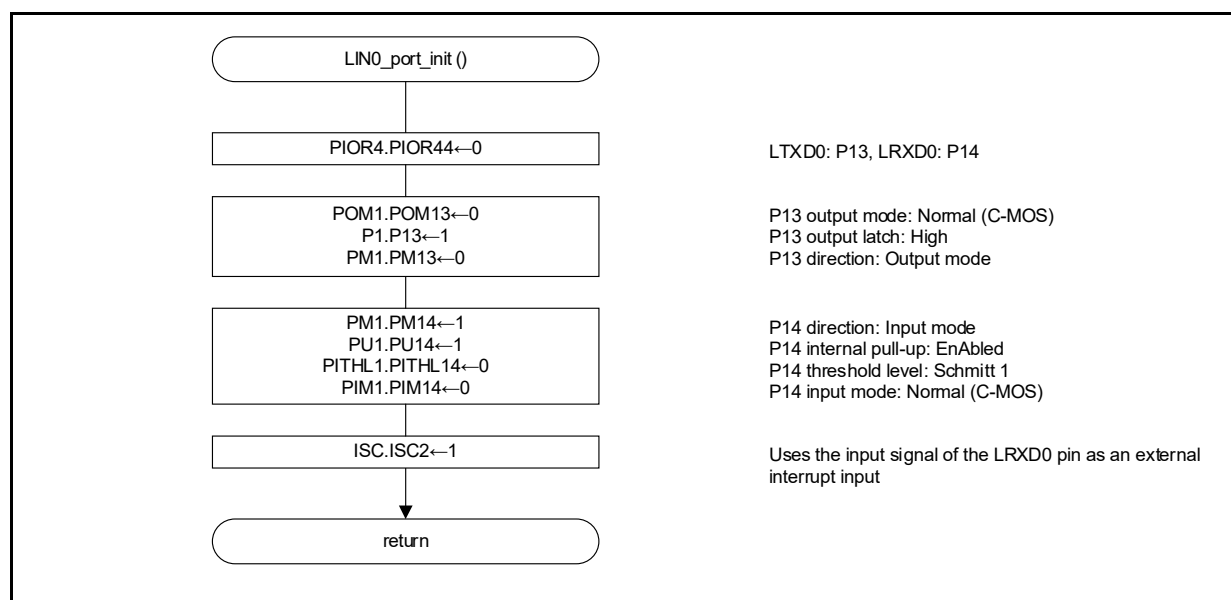


Figure 3-10 LIN0 port Initialization Procedure (LIN0_port_init)

3.3.2 LIN0 Interrupt Initialization

Table 3-13 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.	

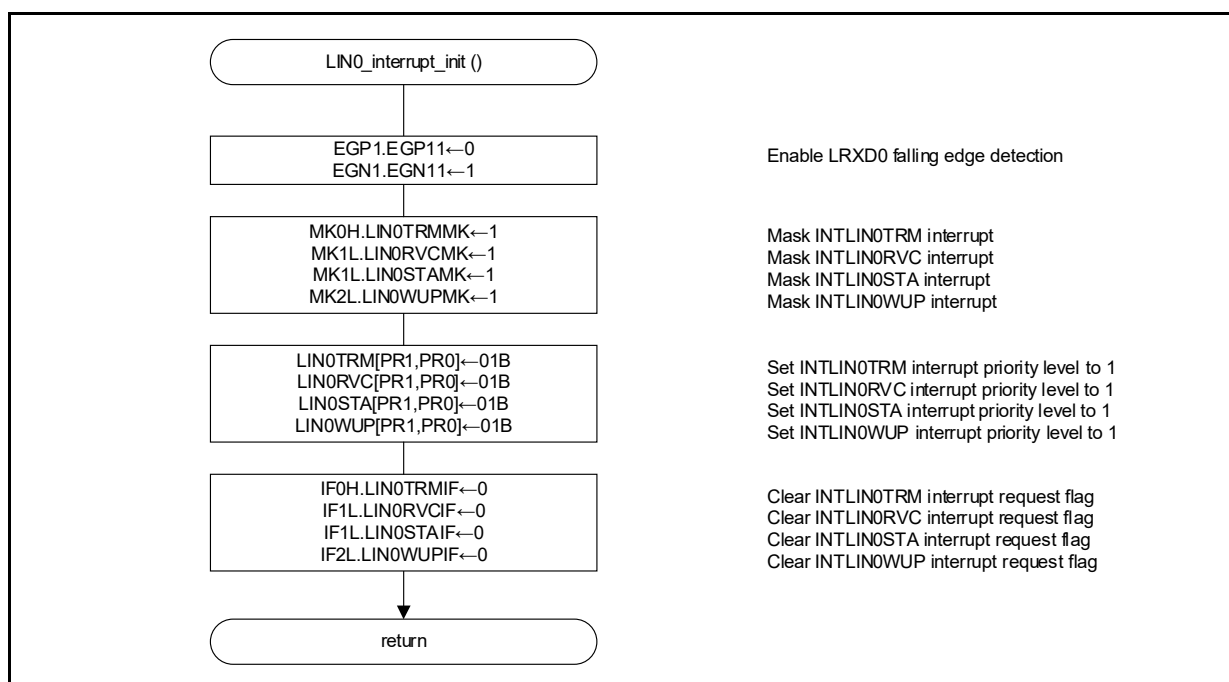


Figure 3-11 LIN0 Interrupt Initialization Procedure (LIN0_interrupt_init)

3.3.3 Transition to LIN Reset Mode

Table 3-14 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.	

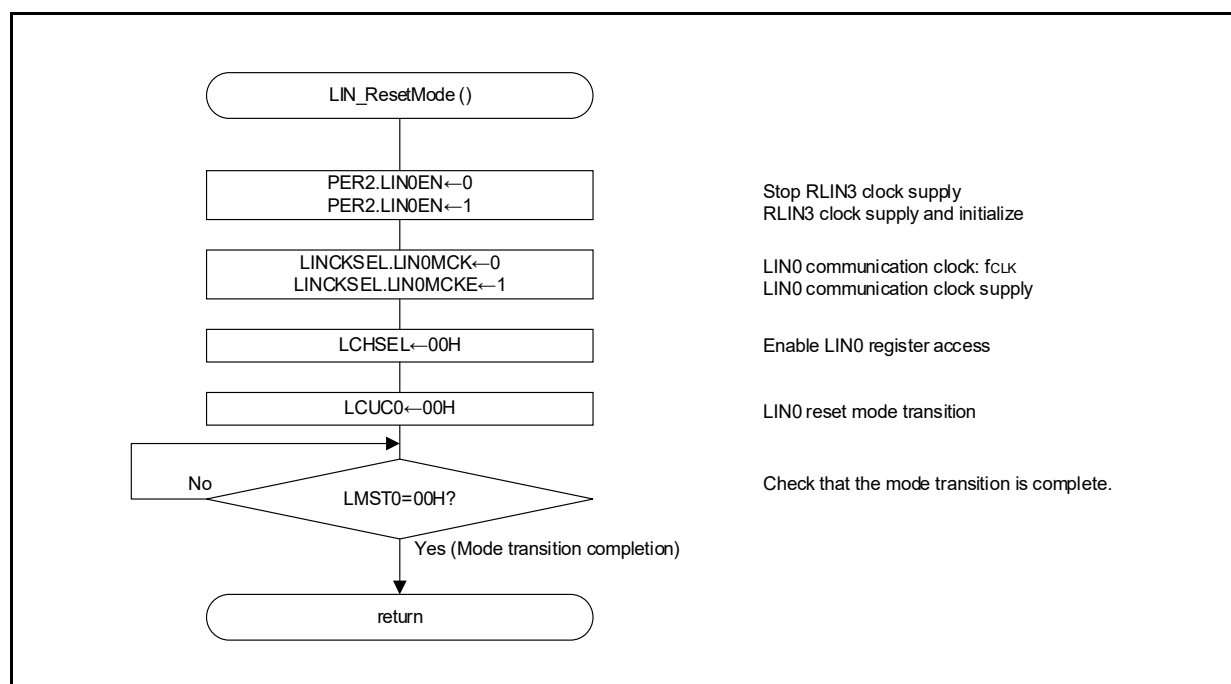


Figure 3-12 Transition to LIN Reset Mode Procedure (LIN_ResetMode)

3.3.4 Transition to LIN Operation Mode

Table 3-15 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.	

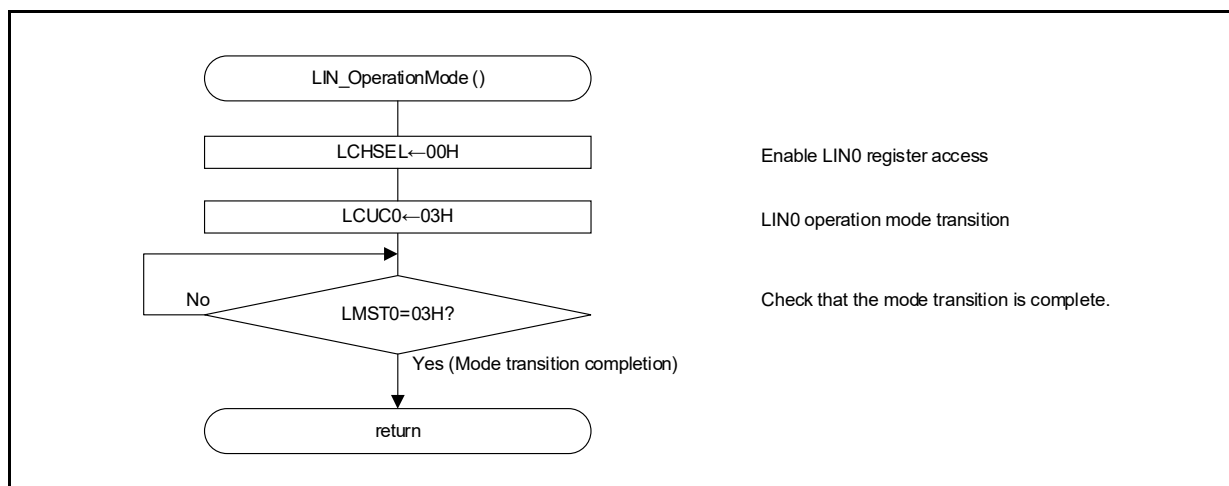


Figure 3-13 Transition to LIN Operation Mode Procedure (LIN_OperationMode)

3.3.5 Transition to LIN Wake-up Mode

Table 3-16 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.	

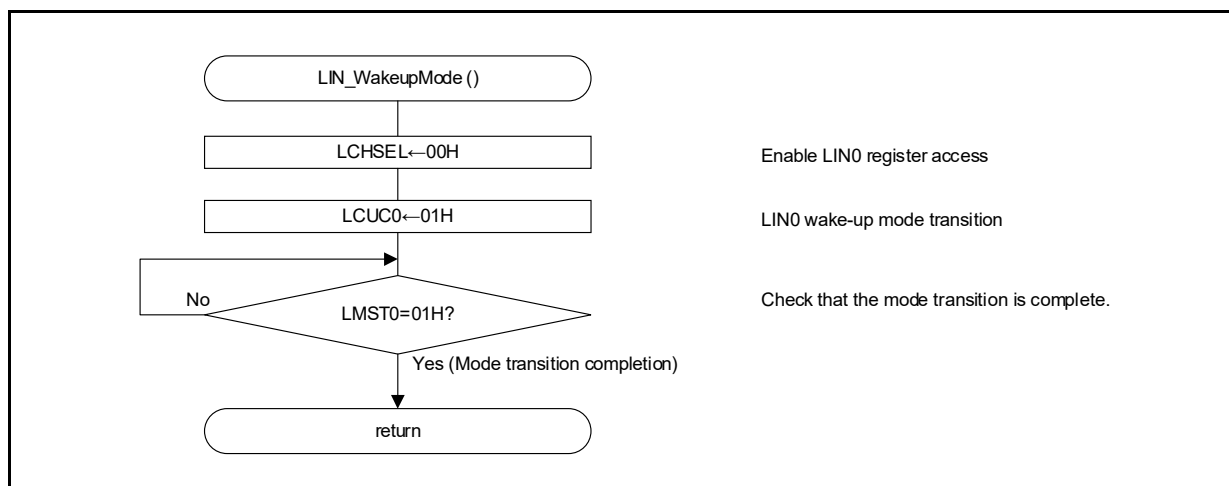


Figure 3-14 Transition to LIN Wake-up Mode Procedure (LIN_WakeupMode)

3.4 Interrupt Processing Functions

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

3.4.1 LIN Slave Reception Completion Interrupt Processing

Table 3-17 LIN Slave Reception Completion Interrupt Processing Function (LINs_IntLIN0RVC)

Syntax		void LINs_IntLIN0RVC (void)	
Parameters	In	None	
	Out	unsigned char g_u8LINs_RxBUFF[0]:	Store the received ID when receiving a header
		unsigned char g_u8LINs_RxBUFF[1:8]:	Store received data when receiving a response
Local variables		unsigned char u8a_RD_LST0:	Store LST0 register
		- bit1 (FRC):	Response reception completion
		- bit3 (ERR):	LIN communication error detection
		- bit7 (HTRC):	Header reception completion
		unsigned short u16a_errorinfo:	8000H (Illegal interrupt has occurred)
Return value		None	
Call functions		UNTfy_LINs_IntHeaderRx ():	Header reception completion processing
		UNTfy_LINs_IntResponseRx ():	Response reception completion processing
		UNTfy_LINs_ComError (u16a_errorinfo):	Illegal interrupt processing
Description		INTLIN0RVC interrupt service routine - LIN0 header reception completion - LIN0 response reception completion (Also refer to section. 3.5.2 for the UNTfy_LINs_IntHeaderRx () processing)	

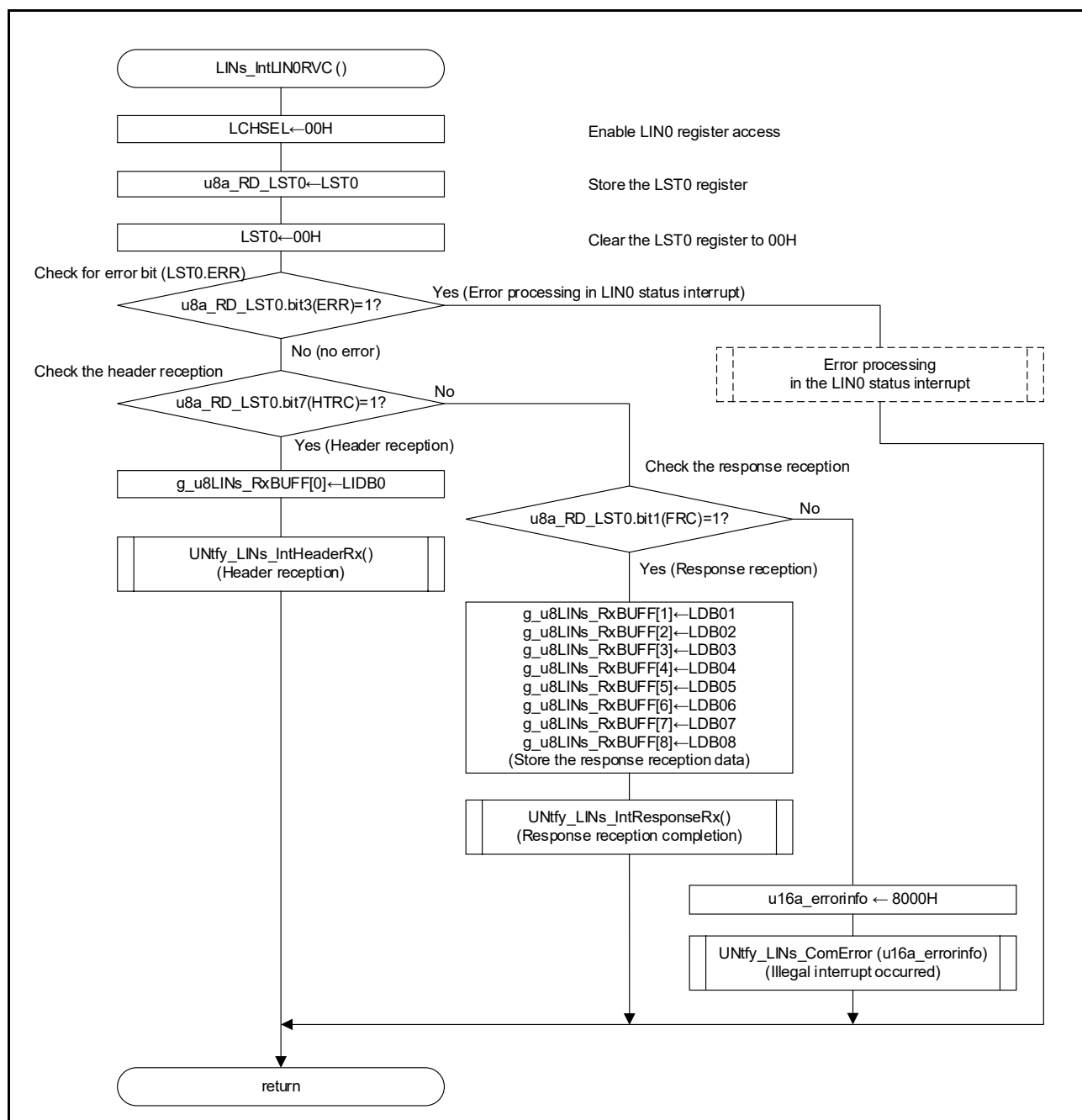


Figure 3-15 LIN Slave Reception Completion Interrupt Processing Procedure (LINs_IntLIN0RVC)

3.4.2 LIN Slave Transmission Completion Interrupt Processing

Table 3-18 LIN Slave Transmission Completion Interrupt Processing Function (LINs_IntLIN0TRM)

Syntax		void LINs_IntLIN0TRM (void)		
Parameters	In	None		
	Out	None		
Local variables		unsigned char	u8a_RD_LST0:	Store LST0 register
		- bit0 (FTC):	Response / Wake-up transmission completion	
		- bit3 (ERR):	LIN communication error detection	
		unsigned short	u16a_errorinfo:	4000H (Illegal interrupt has occurred)
Return value		None		
Call functions		UNtfy_LINs_IntResponseTx ():	Response transmission completion processing	
		UNtfy_LINs_IntWUPSigTx ():	Wake-up transmission completion processing	
		UNtfy_LINs_ComError (u16a_errorinfo):	Illegal interrupt processing	
Description		INTLIN0TRM interrupt service routine		
		- LIN0 response transmission completion		
		- LIN0 wake-up signal transmission completion		

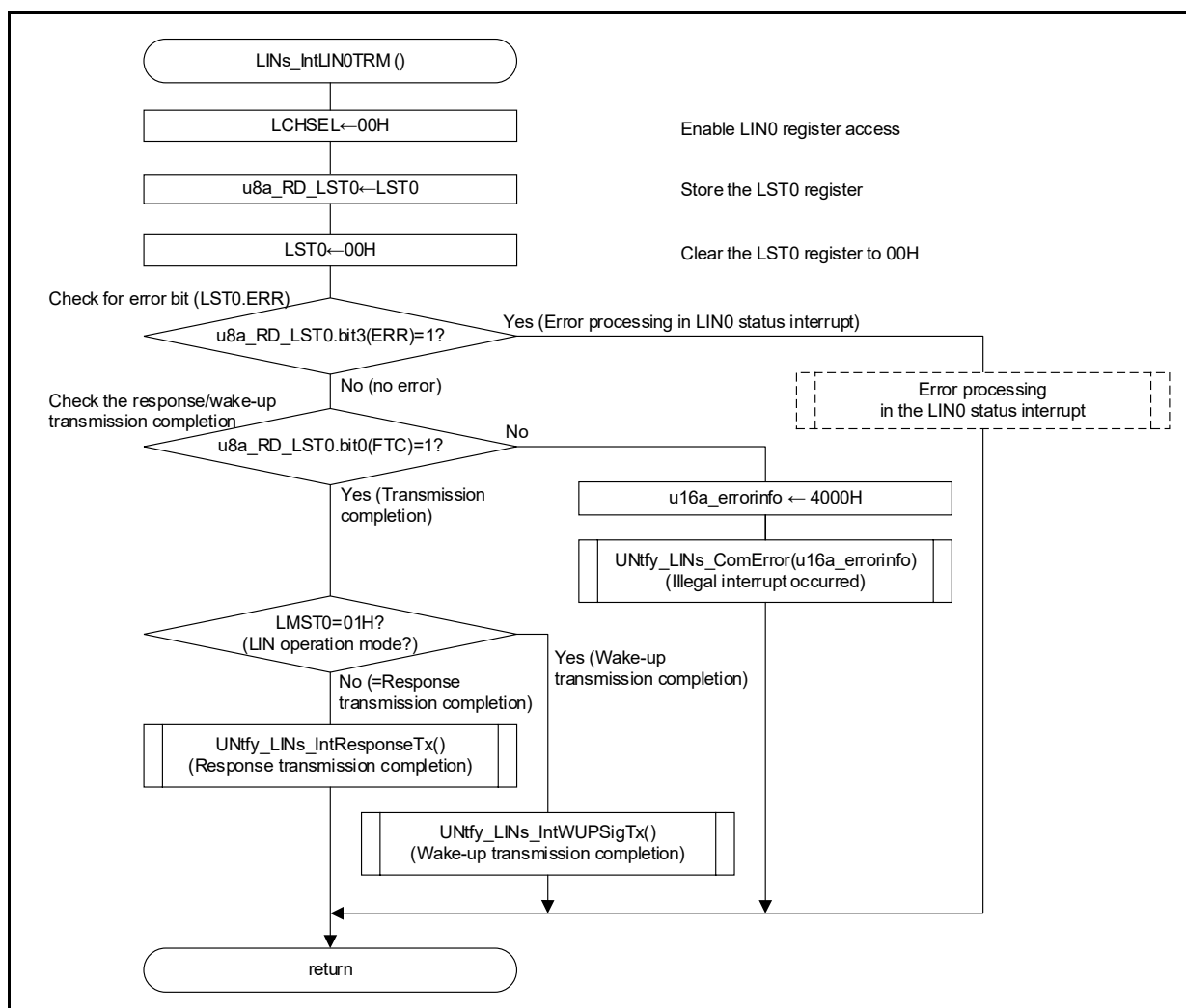


Figure 3-16 LIN Slave Transmission Completion Interrupt Processing Procedure (LINs_IntLIN0TRM)

3.4.3 LIN Slave Status Error Interrupt Processing

Table 3-19 LIN Slave Status Error Interrupt Processing Function (LINS_IntLIN0STA)

Syntax	void LINS_IntLIN0STA (void)	
Parameters	In	None
	Out	None
Local variables	unsigned char	u8a_RD_LEST0: Store LEST0 register
	- bit0 (BER):	Bit error detected during response/wake-up transmission
	- bit2 (TER):	Response timeout error detected in fixed baud rate mode
	- bit3 (FER):	Framing error detected during header/response reception
	- bit4 (SFER):	Sync field error detected during sync field reception
	- bit5 (CSER):	Checksum error detected when response reception
	- bit6 (IPER):	ID parity error detected when ID field reception
	- bit7 (RPER):	Response preparation error detected when response data 1 reception
	unsigned short	u16a_errorinfo: Error information value
	- bit[7:0]:	u8a_RD_LEST0 value
	- bit[15:8]:	00H
Return value	None	
Call functions	UNtfy_LINs_ComError (u16a_errorinfo): Error detection processing	
Description	This function is the interrupt processing when LIN0 status error is occurred.	

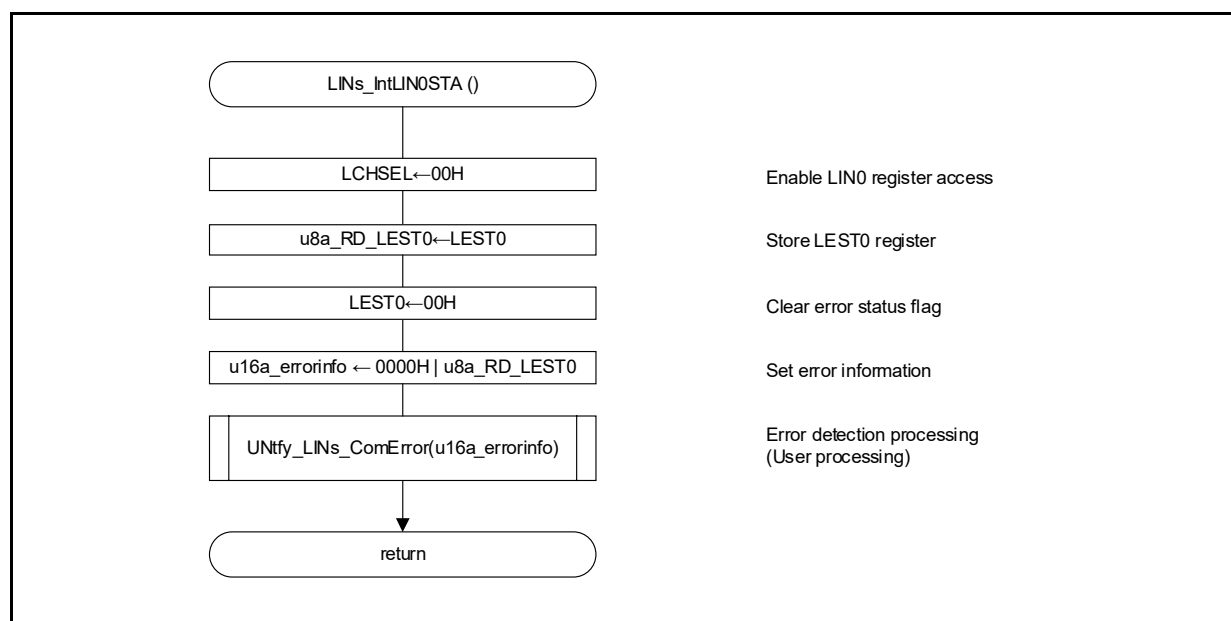


Figure 3-17 LIN Slave Status Error Interrupt Processing Procedure (LINS_IntLIN0STA)

3.4.4 LIN Slave Wake-up Signal Detection Interrupt Processing

Table 3-20 LIN Slave Wake-up Signal Detection Interrupt Processing Function (LINS_IntLIN0WUP)

Syntax	void LINS_IntLIN0WUP (void)	
Parameters	In	None
	Out	None
Local variables	None	
Return value	None	
Call functions	UNtfy_LINS_IntWUPSigRx (): LRXD0 falling edge detection processing	
Description	INTLIN0WUP interrupt service routine	

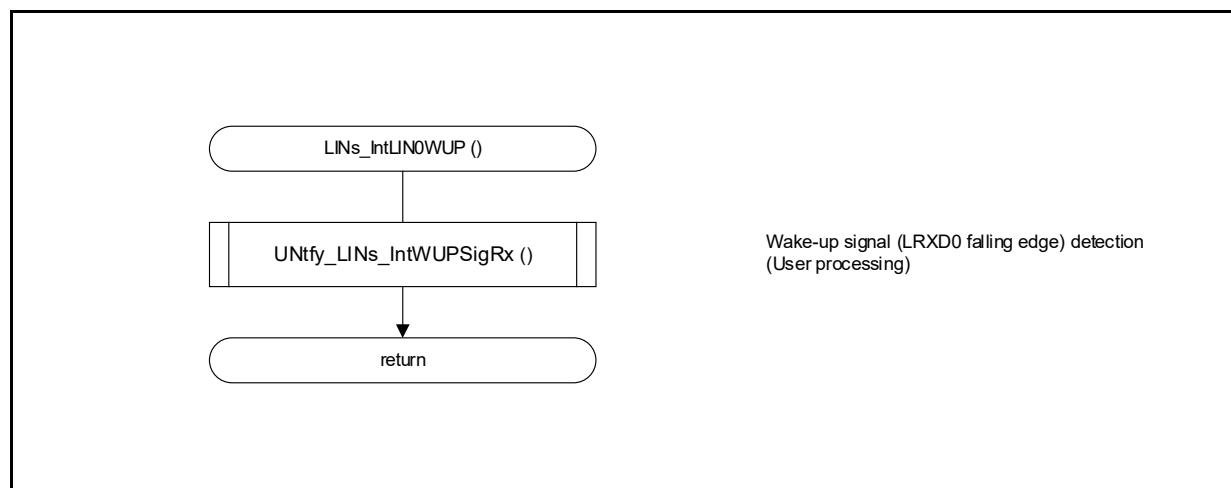


Figure 3-18 LIN Slave Wake-up Signal Detection Interrupt Processing Procedure (LINS_IntLIN0WUP)

3.5 User Notification Function APIs

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

3.5.1 Error Notification

Table 3-21 Error Notification Function API (UNtfy_LINs_ComError)

Syntax		void UNtfy_LINs_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.	

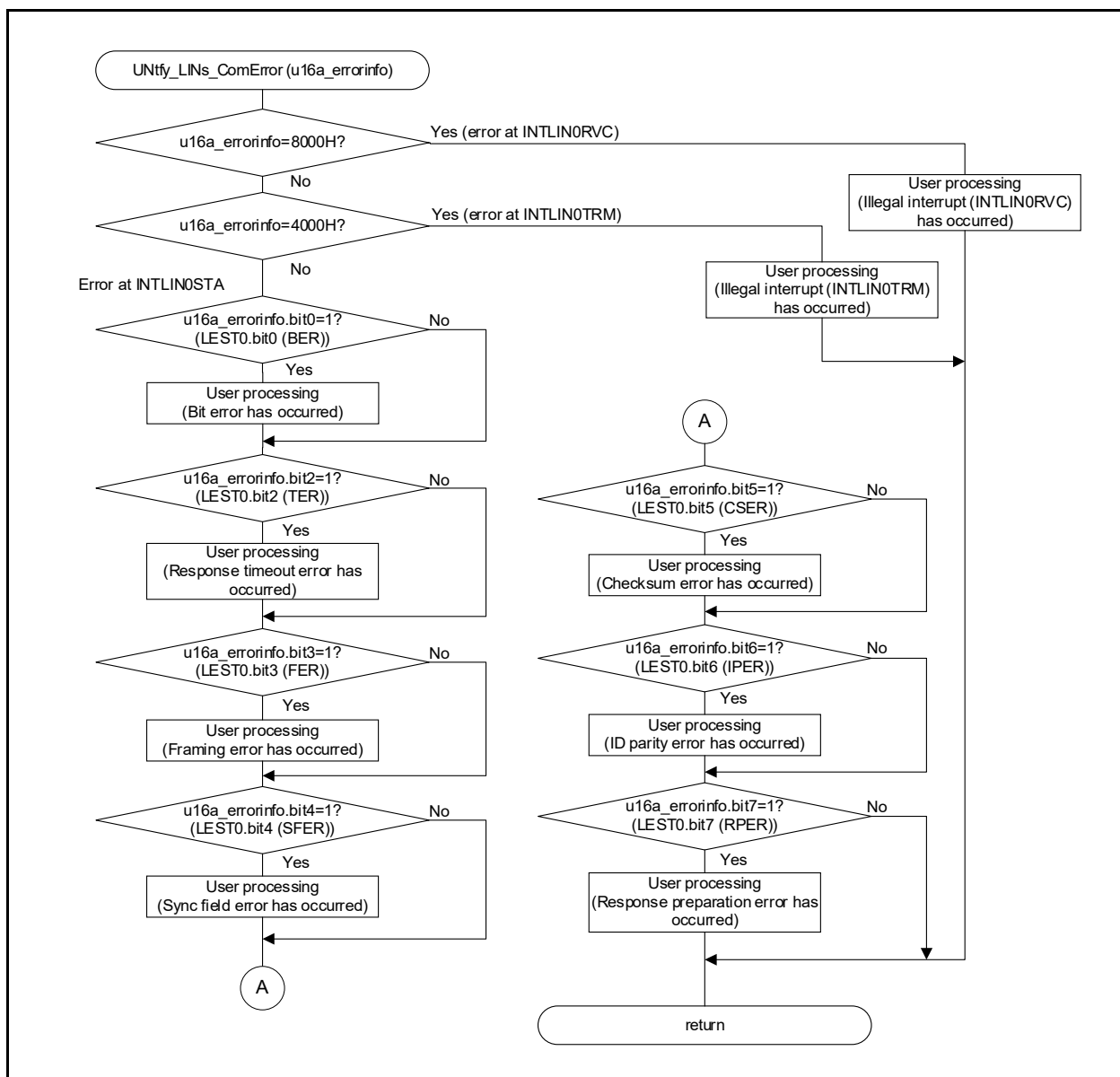


Figure 3-19 Example of Error Notification Procedure (UNtfy_LINs_ComError)

3.5.2 Header Reception Completion Notification

Table 3-22 Header Reception Completion Notification Function API (UNtfy_LINs_IntHeaderRx)

Syntax		void UNtfy_LINs_IntHeaderRx (void)	
Parameters	In	unsigned char g_u8LINs_RxBUFF[0]:	Received ID data
	Out	None	
Local variables		- (User processing)	
Return value		- (User processing)	
Call funxtiona		- (User processing)	
		LINs_TxResponse (u8a_LDFC_value):	Response transmission function
		LINs_RxResponse (u8a_LDFC_value):	Response reception function
		LINs_NoResponse ():	No response request function
Description		This function notifies that the header reception is completed.	

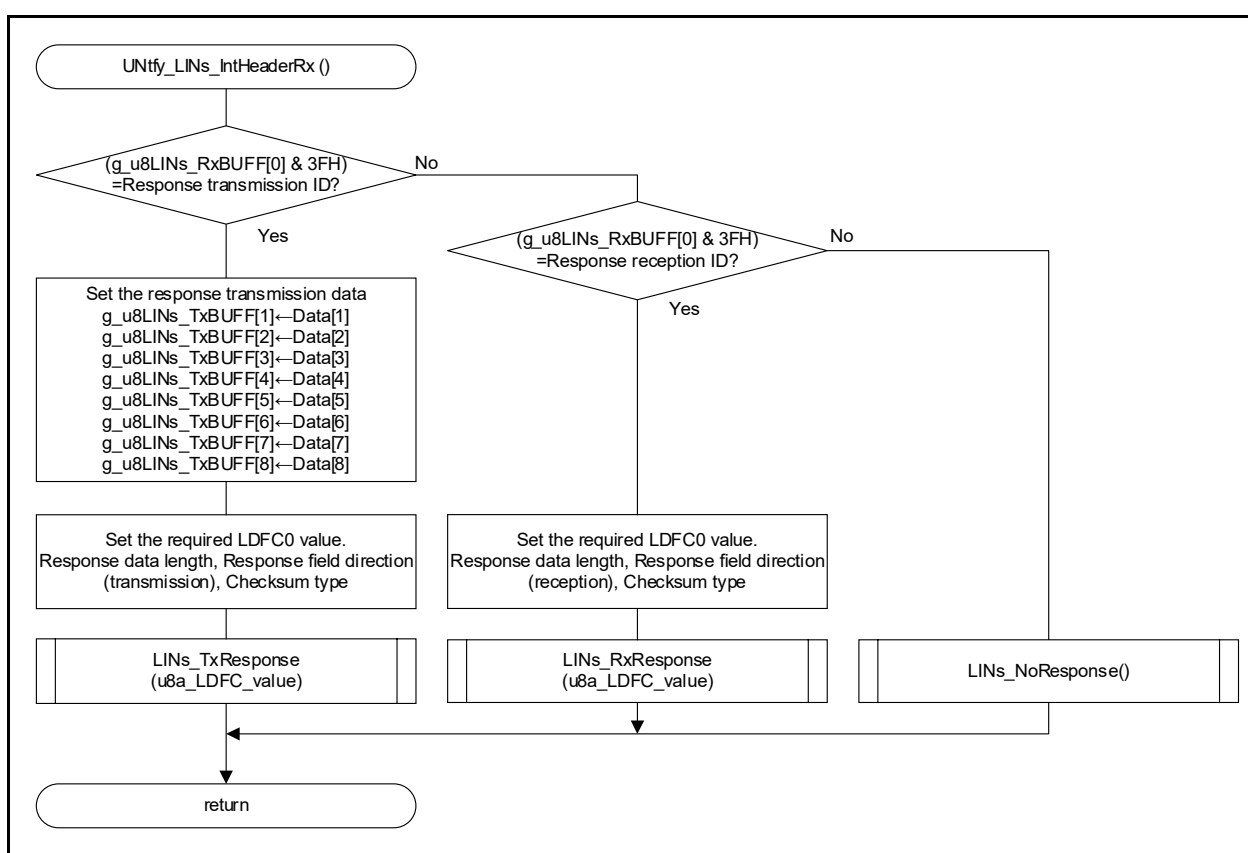
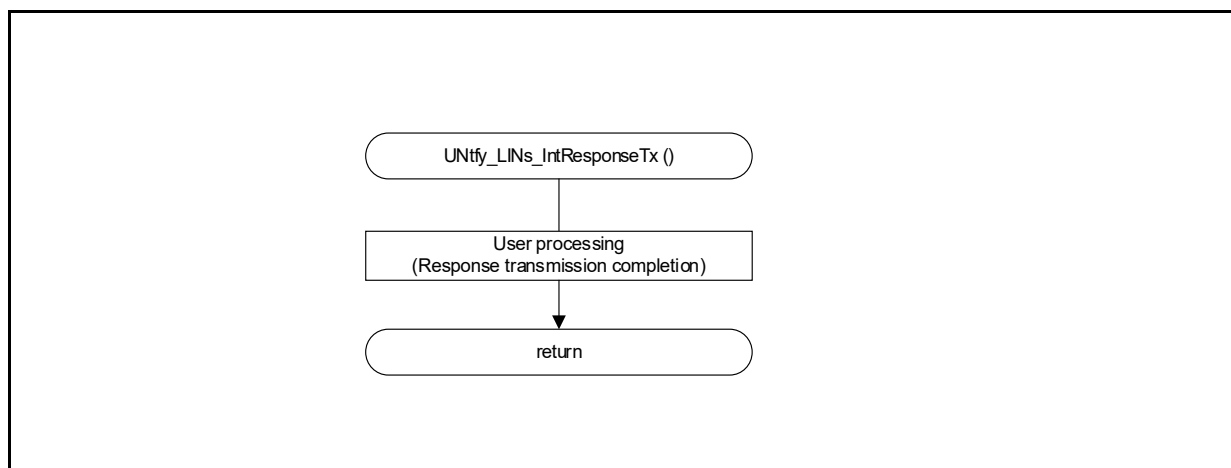


Figure 3-20 Example of Header Reception Completion Notification Procedure (UNtfy_LINs_IntHeaderRx)

3.5.3 Response Transmission Completion Notification

**Table 3-23 Response Transmission Completion Notification Function API
(UNtfy_LINs_IntResponseTx)**

Syntax		void UNtfy_LINs_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 3-21 Example of Response Transmission Completion Notification Procedure
(UNtfy_LINs_IntResponseTx)**

3.5.4 Response Reception Completion Notification

Table 3-24 Response Reception Completion Notification Function API (UNtfy_LINs_IntResponseRx)

Syntax		void UNtfy_LINs_IntResponseRx (void)	
Parameters	In	unsigned char g_u8LINs_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.	

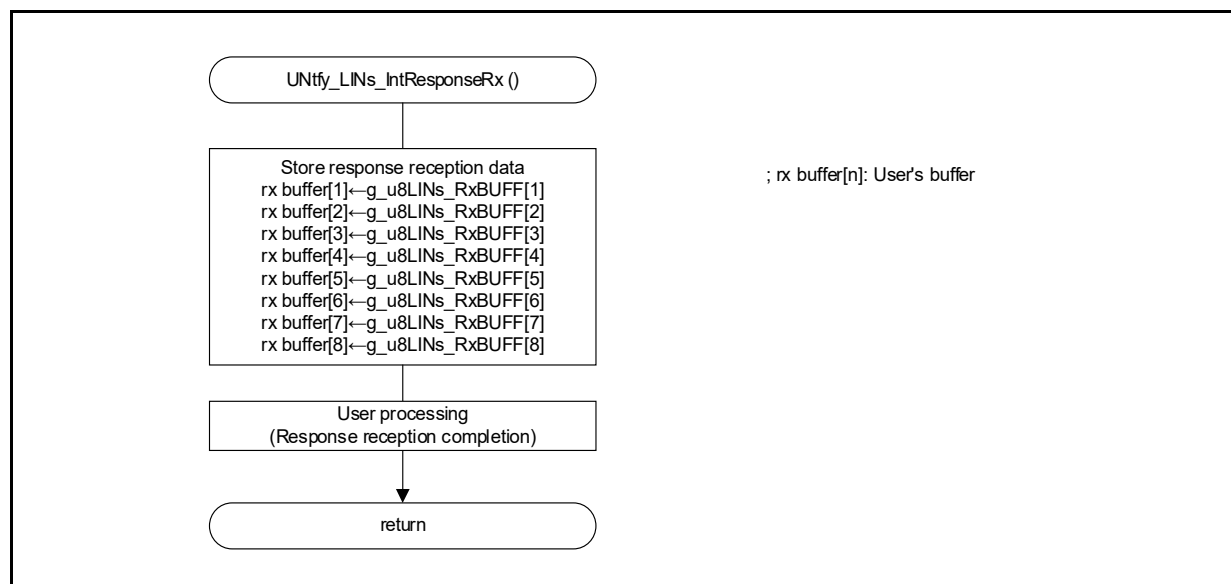
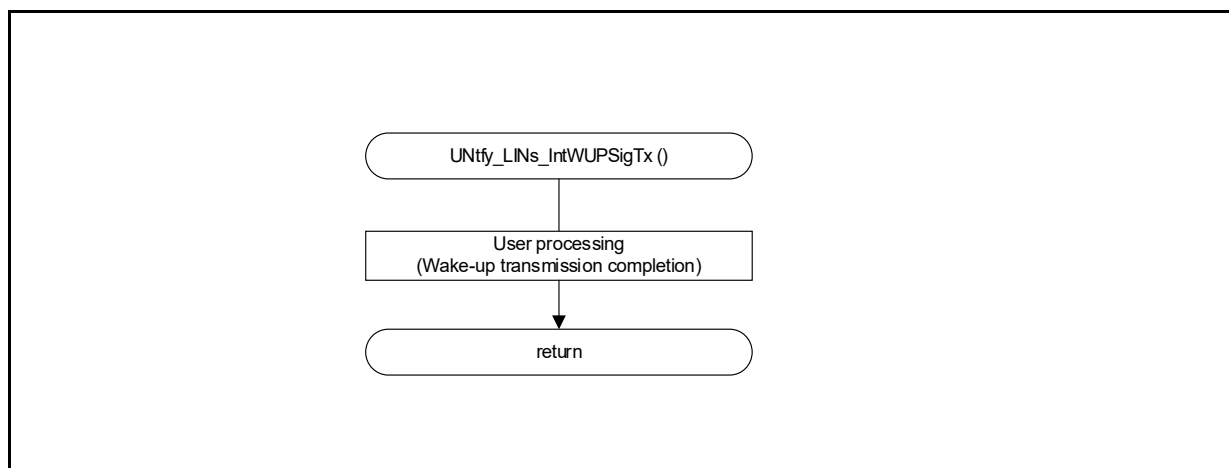


Figure 3-22 Example of Response Reception Completion Notification Procedure (UNtfy_LINs_IntResponseRx)

3.5.5 Wake-up Signal Transmission Completion Notification

**Table 3-25 Wake-up Signal Transmission Completion Notification Function API
(UNtfy_LINs_IntWUPSigTx)**

Syntax	void UNtfy_LINs_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 3-23 Example of Wake-up Signal Transmission Completion Notification Procedure
(UNtfy_LINs_IntWUPSigTx)**

3.5.6 Wake-up Signal Reception Notification

Table 3-26 Wake-up Signal Reception Notification Function API (UNtfy_LINs_IntWUPSigRx)

Syntax	void UNtfy_LINs_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.	

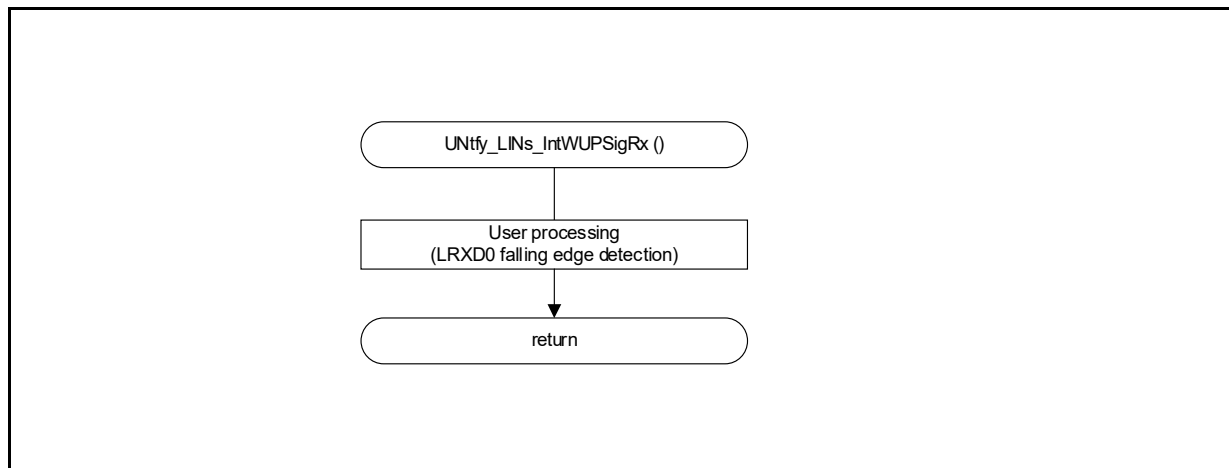


Figure 3-24 Example of Wake-up Signal Reception Notification Procedure (UNtfy_LINs_IntWUPSigRx)

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 ^{Note}
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/ F23, F24 User's Manual: Hardware Rev. 1.00
- RL78 family User's Manual: Software Rev. 2.30

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2022.09.30	-	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

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/.