Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	180MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, IrDA, Microwire, QEI, MMC/SD, SPI, SSI, SSP, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, WDT
Number of I/O	49
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	200K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 4x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/lpc18s30fet100e

Table 31. ISP Write to RAM command

Command	W
Input	**Start Address:** RAM address where data bytes are to be written. This address should be a word boundary. **Number of Bytes:** Number of bytes to be written. Count should be a multiple of 4
Return Code	CMD_SUCCESS | ADDR_ERROR (Address not on word boundary) | ADDR_NOT_MAPPED | COUNT_ERROR (Byte count is not multiple of 4) | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to download data to RAM. Data should be in UU-encoded format. This command is blocked when code read protection levels CRP2 or CRP3 are enabled.
Example	"W 268435968 4<CR><LF>" writes 4 bytes of data to address 0x1000 0200.

5.7.5 Read Memory <address> <no. of bytes>

The data stream is followed by the command success return code. The check-sum is sent after transmitting 20 UU-encoded lines. The checksum is generated by adding raw data (before UU-encoding) bytes and is reset after transmitting 20 UU-encoded lines. The length of any UU-encoded line should not exceed 61 characters (bytes) i.e. it can hold 45 data bytes. When the data fits in less than 20 UU-encoded lines then the check-sum is of actual number of bytes sent. The host should compare it with the checksum of the received bytes. If the check-sum matches then the host should respond with "OK<CR><LF>" to continue further transmission. If the check-sum does not match then the host should respond with "RESEND<CR><LF>". In response the ISP command handler sends the data again.

The ISP read command can read from all internal SRAM blocks, from the external memory bank 0 (CS0 at location 0x1C00 0000), and from the internal flash.

Table 32. ISP Read Memory command

Command	R
Input	**Start Address:** Address from where data bytes are to be read. This address should be a word boundary. **Number of Bytes:** Number of bytes to be read. Count should be a multiple of 4.
Return Code	CMD_SUCCESS followed by <actual data (UU-encoded)> | ADDR_ERROR (Address not on word boundary) | ADDR_NOT_MAPPED | COUNT_ERROR (Byte count is not a multiple of 4) | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to read data from RAM or flash memory. This command is blocked when any level of code read protection is enabled.
Example	"R 268435968 4<CR><LF>" reads 4 bytes of data from address 0x1000 0200.

5.7.6 Prepare sectors for write operation <start sector number> <end sector number> <flash bank>

This command is the first step in the two-step flash write/erase operation.

Table 33. ISP Prepare sectors for write operation command

Command	P
Input	**Start Sector Number** **End Sector Number:** Should be greater than or equal to start sector number. **Flash bank**: Selects flash bank if the part supports more than on bank. 0 = flash bank A, 1 = flash bank B.
Return Code	CMD_SUCCESS | BUSY | INVALID_SECTOR | PARAM_ERROR | INVALID_FLASH_UNIT
Description	This command must be executed before executing "Copy RAM to Flash" or "Erase Sectors" command. Successful execution of the "Copy RAM to Flash" or "Erase Sectors" command causes relevant sectors to be protected again. To prepare a single sector use the same "Start" and "End" sector numbers.
Example	"P 0 0 0 <CR><LF>" prepares the flash sector 0 in flash bank A.

a. Use a 128-bit key to encrypt the image using Cypher Block Chaining (CBC) encryption and an initialization vector of 0101010... (binary). After the first block of data, each following (plain-text) block is XORed with the previous encrypted block of data. For details, see Section 7.3.4 “CMAC”.

b. Create a header using Table 68 “Boot image header description” with a dummy hash size. The actual hash size is calculated after encryption of the image and this header.

c. Encrypt the header using CBC and an initialization vector of 0.

d. Use CMAC to create a hash code and calculate the hash size of the combined encrypted header and image. See Section 7.3.4 “CMAC”.

e. Update the header with the calculated hash size.

f. Encrypt header as before using CBC and an initialization vector of 0.

2. On the LPC18Sxx, program the encryption key into the OTP memory bank 1 using the API function aes_ProgramKey1 (see Table 67).

   Remark: The encryption key itself is scrambled in OTP memory bank 1 for added security.

3. Select boot mode. See Section 4.3.

4. On parts with on-chip flash, JTAG access is not disabled. Therefore, set the appropriate CRP level in the flash memory to disable JTAG access.

   On flashless parts, JTAG access is disabled automatically once the key is programmed in OTP memory bank 1.

5. Reset the LPC18xx, and the part boots securely from the specified boot source. See Figure 27 “Boot flow for encrypted images (flashless parts)”.

Remark: To test the secure boot flow, you can create a secure image with a key of all zeros using the steps above and omitting programming the key into the OTP memory. The part then boots after reset using the zero-encrypted image.

12.6.4 PLL0AUDIO registers

The PLL0AUDIO provides a wide range of frequencies for audio applications and can be connected to multiple base clocks. The PLL0AUDIO can be used with or without a fractional divider.

See Section 12.7.4.5 for instructions on how to set up the PLL0.

12.6.4.1 PLL0AUDIO status register

Table 117. PLL0AUDIO status register (PLL0AUDIO_STAT, address 0x4005 002C) bit description

Bit	Symbol	Description	Reset value	Access
0	LOCK	PLL0 lock indicator	0	R
1	FR	PLL0 free running indicator	0	R
31:2	-	Reserved		-

12.6.4.2 PLL0AUDIO control register

Table 118. PLL0AUDIO control register (PLL0AUDIO_CTRL, address 0x4005 0030) bit description

Bit	Symbol	Value	Description	Reset value	Access
0	PD		PLL0 power down	1	R/W
		0	PLL0 enabled		
		1	PLL0 powered down		
1	BYPASS		Input clock bypass control	1	R/W
		0	CCO clock sent to post-dividers. Use this in normal operation.		
		1	PLL0 input clock sent to post-dividers (default).		
2	DIRECTI		PLL0 direct input	0	R/W
3	DIRECTO		PLL0 direct output	0	R/W
4	CLKEN		PLL0 clock enable	0	R/W
5	-		Reserved	-	-
6	FRM		Free running mode	0	R/W
7	-		Reserved	0	R/W
8	-		Reserved. Reads as zero. Do not write one to this register.	0	R/W
9	-		Reserved. Reads as zero. Do not write one to this register.	0	R/W
10	-		Reserved. Reads as zero. Do not write one to this register.	0	R/W
11	AUTOBLOCK		Block clock automatically during frequency change[1].	0	R/W
		0	Autoblocking disabled		
		1	Autoblocking enabled		

12.6.5.2 PLL1 control register

Table 123. PLL1_CTRL register (PLL1_CTRL, address 0x4005 0044) bit description

Bit	Symbol	Value	Description	Reset value	Access
0	PD		PLL1 power down	1	R/W
		0	PLL1 enabled		
		1	PLL1 powered down		
1	BYPASS		Input clock bypass control	1	R/W
		0	CCO clock sent to post-dividers. Use for normal operation.		
		1	PLL1 input clock sent to post-dividers (default).		
2	-		Reserved. Do not write one to this bit.	0	R/W
5:3	-		Reserved. Do not write one to these bits.	-	-
6	FBSEL		PLL feedback select (see Figure 36 “PLL1 block diagram”).	0	R/W
		0	CCO output is used as feedback divider input clock.		
		1	PLL output clock (clkout) is used as feedback divider input clock. Use for normal operation.		
7	DIRECT		PLL direct CCO output	0	R/W
		0	Disabled		
		1	Enabled		
9:8	PSEL		Post-divider division ratio P. The P-divider applied by the PLL is 2xP.	01	R/W
		0x0	P = 1		
		0x1	P = 2 (default)		
		0x2	P = 4		
		0x3	P = 8		
10	-		Reserved	-	-
11	AUTOBLOCK		Block clock automatically during frequency change[1].	0	R/W
		0	Autoblocking disabled		
		1	Autoblocking enabled		
13:12	NSEL		Pre-divider division ratio (N)	10	R/W
		0x0	1		
		0x1	2		
		0x2	3 (default)		
		0x3	4		
15:14	-		Reserved	-	-

these signals. All output clock Status Registers follow the format as described in Table 150 and Table 151.

Remark: The divider value for the CLK_M3_EMCDIV_CFG register is not reflected in the status register. Read the DIVSTAT bits in the CLK_M3_EMCDIV_CFG register for the divider status.

Table 150. CCU1 branch clock status register (CLK_XXX_STAT, addresses 0x4005 1104, 0x4005 110C,..., 0x4005 1A04) bit description

Bit	Symbol	Description	Reset value	Access
0	RUN	Clock enable status 0 = clock is disabled. 1 = clock is enabled.	1	R
1	AUTO	Auto (AHB disable mechanism) enable status 0 = Auto is disabled. 1 = Auto is enabled.	0	R
2	WAKEUP	Wake-up mechanism enable status. This bit reads as 1 when the power down bit has been set in the PM register (PD = 1) and the clock has been configured to run after wake-up. 0 = Wake-up is disabled. 1 = Wake-up is enabled.	0	R
4:3	-	Reserved.	-	-
5	RUN_N	Clock disable status. This bit has same functionality as the RUN bit except with the opposite polarity. 0 = clock is enabled. 1 = clock is disabled.	0	R
31:6	-	Reserved	-	-

Table 151. CCU2 branch clock status register (CLK_XXX_STAT, addresses 0x4005 2104, 0x4005 2204,..., 0x4005 2804) bit description

Bit	Symbol	Description	Reset value	Access
0	RUN	Clock enable status 0 = clock is disabled 1 = clock is enabled	1	R
1	AUTO	Auto (AHB disable mechanism) enable status 0 = Auto is disabled 1 = Auto is enabled	0	R
2	WAKEUP	Wake-up mechanism enable status. This bit reads as 1 when the power down bit has been set in the PM register (PD = 1) and the clock has been configured to run after wake-up. 0 = Wake-up is disabled 1 = Wake-up is enabled	0	R

Table 169. LPC1850/30/20/10 Pin description (flashless parts) *…continued*
LCD, Ethernet, USB0, and USB1 functions are not available on all parts.

Symbol	LBGA256	TFBGA180	TFBGA100	LQFP144		Reset state [1]	Type	Description
PF_9	D6	-	-	-	[5]	N; PU	-	**R —** Function reserved.
							I/O	**U0_DIR —** RS-485/EIA-485 output enable/direction control for USART0.
							O	**CTOUT_1 —** SCT output 1. Match output 3 of timer 3.
							-	**R —** Function reserved.
							I/O	**GPIO7[23] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							AI	**ADC1_2 —** ADC1 and ADC0, input channel 2. Configure the pin as GPIO input and use the ADC function select register in the SCU to select the ADC.
PF_10	A3	-	-	-	[5]	N; PU	-	**R —** Function reserved.
							O	**U0_TXD —** Transmitter output for USART0.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**GPIO7[24] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							I	**SD_WP —** SD/MMC card write protect input.
							-	**R —** Function reserved.
							AI	**ADC0_5 —** ADC0 and ADC1, input channel 5. Configure the pin as GPIO input and use the ADC function select register in the SCU to select the ADC.
PF_11	A2	-	-	-	[5]	N; PU	-	**R —** Function reserved.
							I	**U0_RXD —** Receiver input for USART0.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**GPIO7[25] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							O	**SD_VOLT2 —** SD/MMC bus voltage select output 2.
							-	**R —** Function reserved.
							AI	**ADC1_5 —** ADC1 and ADC0, input channel 5. Configure the pin as GPIO input and use the ADC function select register in the SCU to select the ADC.

Table 170. LPC18xx Pin description (flash-based parts) *…continued*

Pin name	LBGA256	TFBGA100	LQFP144	LQFP208		Reset state [1]	Type	Description
P1_1	R2	K2	42	58	[2]	N; PU	I/O	**GPIO0[8] —** General purpose digital input/output pin. External boot pin (see Table 16).
							O	**CTOUT_7 —** SCT output 7. Match output 3 of timer 1.
							I/O	**EMC_A6 —** External memory address line 6.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**SSP0_MISO —** Master In Slave Out for SSP0.
							-	**R —** Function reserved.
							I/O	**EMC_D13 —** External memory data line 13.
P1_2	R3	K1	43	60	[2]	N; PU	I/O	**GPIO0[9] —** General purpose digital input/output pin. External boot pin (see Table 16).
							O	**CTOUT_6 —** SCT output 6. Match output 2 of timer 1.
							I/O	**EMC_A7 —** External memory address line 7.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**SSP0_MOSI —** Master Out Slave in for SSP0.
							-	**R —** Function reserved.
							I/O	**EMC_D14 —** External memory data line 14.
P1_3	P5	J1	44	61	[2]	N; PU	I/O	**GPIO0[10] —** General purpose digital input/output pin.
							O	**CTOUT_8 —** SCT output 8. Match output 0 of timer 2.
							-	**R —** Function reserved.
							O	**EMC_OE —** LOW active Output Enable signal.
							O	**USB0_IND1 —** USB0 port indicator LED control output 1.
							I/O	**SSP1_MISO —** Master In Slave Out for SSP1.
							-	**R —** Function reserved.
							O	**SD_RST —** SD/MMC reset signal for MMC4.4 card.
P1_4	T3	J2	47	64	[2]	N; PU	I/O	**GPIO0[11] —** General purpose digital input/output pin.
							O	**CTOUT_9 —** SCT output 9. Match output 3 of timer 3.
							-	**R —** Function reserved.
							O	**EMC_BLS0 —** LOW active Byte Lane select signal 0.
							O	**USB0_IND0 —** USB0 port indicator LED control output 0.
							I/O	**SSP1_MOSI —** Master Out Slave in for SSP1.
							I/O	**EMC_D15 —** External memory data line 15.
							O	**SD_VOLT1 —** SD/MMC bus voltage select output 1.

Table 170. LPC18xx Pin description (flash-based parts) *…continued*

Pin name	LBGA256	TFBGA100	LQFP144	LQFP208		Reset state [1]	Type	Description
P7_2	A16	-	115	165	[2]	N; PU	I/O	**GPIO3[10] —** General purpose digital input/output pin.
							I	**CTIN_4 —** SCT input 4. Capture input 2 of timer 1.
							I/O	**I2S0_TX_SDA —** I^2S transmit data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I^2S-bus specification*.
							O	**LCD_VD18 —** LCD data.
							O	**LCD_VD6 —** LCD data.
							-	**R —** Function reserved.
							I	**U2_RXD —** Receiver input for USART2.
							-	**R —** Function reserved.
P7_3	C13	-	117	167	[2]	N; PU	I/O	**GPIO3[11] —** General purpose digital input/output pin.
							I	**CTIN_3 —** SCT input 3. Capture input 1 of timer 1.
							-	**R —** Function reserved.
							O	**LCD_VD17 —** LCD data.
							O	**LCD_VD5 —** LCD data.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P7_4	C8	-	132	189	[5]	N; PU	I/O	**GPIO3[12] —** General purpose digital input/output pin.
							O	**CTOUT_13 —** SCT output 13. Match output 3 of timer 3.
							-	**R —** Function reserved.
							O	**LCD_VD16 —** LCD data.
							O	**LCD_VD4 —** LCD data.
							O	**TRACEDATA[0] —** Trace data, bit 0.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							AI	**ADC0_4 —** ADC0 and ADC1, input channel 4. Configure the pin as GPIO input and use the ADC function select register in the SCU to select the ADC.

There is no data schedule mechanism for these transactions other than micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

23.8.1.9 Multiple Transaction Translators

The maximum number of embedded Transaction Translators that is currently supported is one as indicated by the N_TT field in the HCSPARAMS – Host Control Structural Parameters register.

23.8.2 Device operation

The co-existence of a device operational controller within the host controller has little effect on EHCI compatibility for host operation except as noted in this section.

23.8.2.1 USBMODE register

Given that the dual-role controller is initialized in neither host nor device mode, the USBMODE register must be programmed for host operation before the EHCI host controller driver can begin EHCI host operations.

23.8.2.2 Non-Zero register fields

Some of the reserved fields and reserved addresses in the capability registers and operational register are used in device mode.For read and write operations to these register, note the following:

- Always write zero to all EHCI reserved fields (some of which are device fields). This is an EHCI requirement of the device controller driver.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusively in device mode are undefined in host mode.

23.8.2.3 SOF interrupt

This SOF Interrupt used for device mode is also used in host mode. In host mode, this interrupt is raised every 125 μs (high-speed mode). EHCI does not specify this interrupt but it has been added for convenience and as a potential software time base. See USBSTS (Section 23.6.4) and USBINTR (Section 23.6.5) registers.

23.8.3 Deviations from EHCI

23.8.3.1 Discovery

23.8.3.1.1 Port reset

The port connect methods specified by EHCI require setting the port reset bit in the PORTSC1 register for a duration of 10 ms. Due to the complexity required to support the attachment of devices that are not high speed, there are counters already present in the design that can count the 10 ms reset pulse to alleviate the burden on the software to measure this duration. The basic connection is summarized as follows:

- [Port Change Interrupt] Port connect change occurs to notify the host controller driver that a device has attached.

Table 516. USBD_HID_INIT_PARAM class structure

Member	Description
HID_EpIn_Hdlr	ErrorCode_t(* USBD_HID_INIT_PARAM::HID_EpIn_Hdlr)(USBD_HANDLE_T hUsb, void *data, uint32_t event) Optional Interrupt IN endpoint event handler. The application software could provide Interrupt IN endpoint event handler. Application which send reports to host on interrupt endpoint should provide an endpoint event handler through this data member. This data member is ignored if the interface descriptor Parameters: 1. hUsb = Handle to the USB device stack. 2. data = Handle to HID function driver. 3. event = Type of endpoint event. See USBD_EVENT_T for more details. Returns: The call back should return ErrorCode_t type to indicate success or error condition. Return values: 1. LPC_OK = On success. 2. ERR_USBD_UNHANDLED = Event is not handled hence pass the event to next in line. 3. ERR_USBD_xxx = For other error conditions.

27.6.8 Interrupt Mask register

The INTMSK register controls whether various LCD interrupts occur.Setting bits in this register enables the corresponding raw interrupt INTRAW status bit values to be passed to the INTSTAT register for processing as interrupts.

Table 601. Interrupt Mask register (INTMSK, address 0x4000 801C) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	-
1	FUFIM	FIFO underflow interrupt enable. 0: The FIFO underflow interrupt is disabled. 1: Interrupt will be generated when the FIFO underflows.	0x0
2	LNBUIM	LCD next base address update interrupt enable. 0: The base address update interrupt is disabled. 1: Interrupt will be generated when the LCD base address registers have been updated from the next address registers.	0x0
3	VCOMPIM	Vertical compare interrupt enable. 0: The vertical compare time interrupt is disabled. 1: Interrupt will be generated when the vertical compare time (as defined by LcdVComp field in the CTRL register) is reached.	0x0
4	BERIM	AHB master error interrupt enable. 0: The AHB Master error interrupt is disabled. 1: Interrupt will be generated when an AHB Master error occurs.	0x0
31:5	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	-

27.6.9 Raw Interrupt Status register

The INTRAW register contains status flags for various LCD controller events. These flags can generate an interrupts if enabled by mask bits in the INTMSK register.

Table 602. Raw Interrupt Status register (INTRAW, address 0x4000 8020) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	-
1	FUFRIS	FIFO underflow raw interrupt status. Set when either the upper or lower DMA FIFOs have been read accessed when empty causing an underflow condition to occur. Generates an interrupt if the FUFIM bit in the INTMSK register is set.	
2	LNBURIS	LCD next address base update raw interrupt status. Mode dependent. Set when the current base address registers have been successfully updated by the next address registers. Signifies that a new next address can be loaded if double buffering is in use. Generates an interrupt if the LNBUIM bit in the INTMSK register is set.	0x0

28.7.10.1.4 Configure multiple states

1. In the EVSTATEMASK register for each event (up to 16 events, one register per event), select the state or states (up to 31) in which this event is allowed to occur. Each state can be selected for more than one event.
2. Determine how the event affects the system state:

   In the EVCTRL registers (up to 16 events, one register per event), set the new state value in the STATEV field for this event. If the event is the highest numbered in the current state, this value is either added to the existing state value or replaces the existing state value, depending on the field STATELD.

   Remark: If there are higher numbered events in the current state, this event cannot change the state.

   If the STATEV and STATELD values are set to zero, the state does not change.

28.7.10.1.5 Miscellaneous options

- There are a certain (selectable) number of capture registers. Each capture register can be programmed to capture the counter contents when one or more events occur.
- If the counter is in bidirectional mode, the effect of set and clear of an output can be made to depend on whether the counter is counting up or down by writing to the OUTPUTDIRCTRL register.

28.7.10.2 Operate the SCT

1. Configure the SCT (see Section 28.7.10.1 “Configure the SCT”).
   a. Configure the counter (see Section 28.7.10.1.1).
   b. Configure the match and capture registers (see Section 28.7.10.1.2).
   c. Configure the events and event responses (see Section 28.7.10.1.3).
   d. Configure multiple states (Section 28.7.10.1.4).
2. Write to the STATE register to define the initial state. By default the initial state is state 0.
3. To start the SCT, write to the CTRL register:
   – Clear the counters.
   – Clear or set the STOP_L and/or STOP_H bits.

     Remark: The counter starts counting once the STOP bit is cleared as well. If the STOP bit is set, the SCT waits instead for an event to occur that is configured to start the counter.

   – For each counter, select unidirectional or bidirectional counting mode (field BIDIR_L and/or BIDIR_H).
   – Select the prescale factor for the counter clock (CTRL register).
   – Clear the HALT_L and/or HALT_H bit. By default, the counters are halted and no events can occur.
4. To stop the counters by software at any time, stop or halt the counter (write to STOP_L and/or STOP_H bits or HALT_L and/or HALT_H bits in the CTRL register).
   – When the counters are stopped, both an event configured to clear the STOP bit or software writing a zero to the STOP bit can start the counter again.

If the IntStatus bit is one and no interrupt is pending and the IntId bits will be zero. If the IntStatus is 0, a non auto-baud interrupt is pending in which case the IntId bits identify the type of interrupt and handling as described in Table 852. Given the status of IIR[3:0], an interrupt handler routine can determine the cause of the interrupt and how to clear the active interrupt. The IIR must be read in order to clear the interrupt prior to exiting the Interrupt Service Routine.

The USART RLS interrupt (IIR[3:1] = 011) is the highest priority interrupt and is set whenever any one of four error conditions occur on the USART RX input: overrun error (OE), parity error (PE), framing error (FE) and break interrupt (BI). The USART Rx error condition that set the interrupt can be observed via LSR[4:1]. The interrupt is cleared upon a LSR read.

The USART RDA interrupt (IIR[3:1] = 010) shares the second level priority with the CTI interrupt (IIR[3:1] = 110). The RDA is activated when the USART Rx FIFO reaches the trigger level defined in FCR7:6 and is reset when the USART Rx FIFO depth falls below the trigger level. When the RDA interrupt goes active, the CPU can read a block of data defined by the trigger level.

The CTI interrupt (IIR[3:1] = 110) is a second level interrupt and is set when the USART Rx FIFO contains at least one character and no USART Rx FIFO activity has occurred in 3.5 to 4.5 character times. Any USART Rx FIFO activity (read or write of USART RSR) will clear the interrupt. This interrupt is intended to flush the USART RBR after a message has been received that is not a multiple of the trigger level size. For example, if a peripheral wished to send a 105 character message and the trigger level was 10 characters, the CPU would receive 10 RDA interrupts resulting in the transfer of 100 characters and 1 to 5 CTI interrupts (depending on the service routine) resulting in the transfer of the remaining 5 characters.

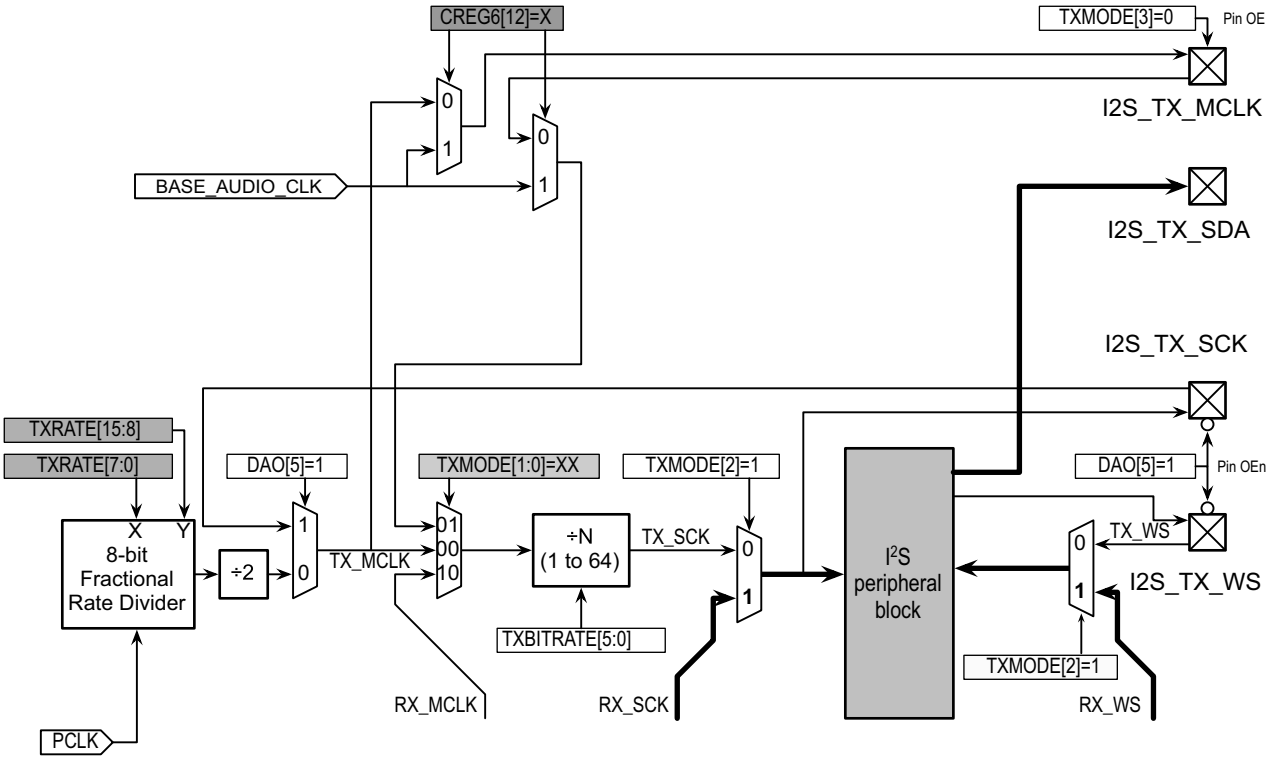
Table 852. USART Interrupt Handling

IIR[3:0] value[1]	Priority	Interrupt type	Interrupt source	Interrupt reset
0001	-	None	None	-
0110	Highest	RX Line Status / Error	OE[2] or PE[2] or FE[2] or BI[2]	LSR Read[2]

41.7.2.1.5 4-Wire Transmitter mode

Table 929. 4-Wire Transmitter mode

CREG bit 12	DAO bit 5	TXMODE bits [3:0]	Description
x	1	0 1 x x	4-wire transmitter mode sharing the receiver RX_SCK and RX_WS (4-pin mode). The I2S transmit function operates as an internal slave to the receive function. The receive function can operate in either master or slave mode, determining the operating mode of the entire I2S interface. The transmit clock source is RX_SCK. The WS used is the internally generated RX_WS. The TX_MCLK pin is not enabled for output.



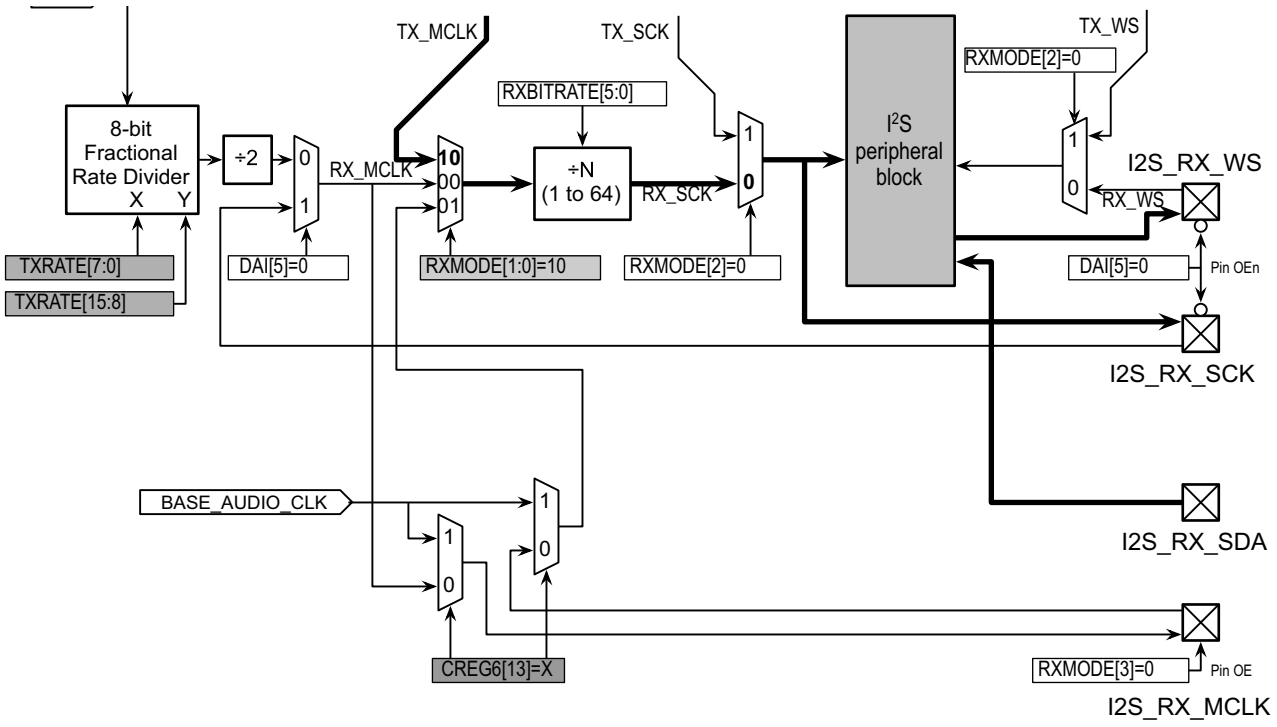
Bold lines indicate the clock path for this configuration.

Fig 138. 4-Wire Transmitter mode

41.7.2.2.3 Receiver master mode, sharing TX_MCLK

Table 934. Receiver master mode, sharing TX_MCLK

CREG bit 13	DAI bit 5	RXMODE bits [3:0]	Description
x	0	0 0 1 0	Receiver master mode sharing the transmitter reference clock (TX_MCLK). The I2S receive function operates as a master. The receive clock source is TX_MCLK. The WS used is the internally generated RX_WS. The RX_MCLK pin is not enabled for output.



Bold lines indicate the clock path for this configuration.

Fig 143. Receiver master mode, sharing TX_MCLK

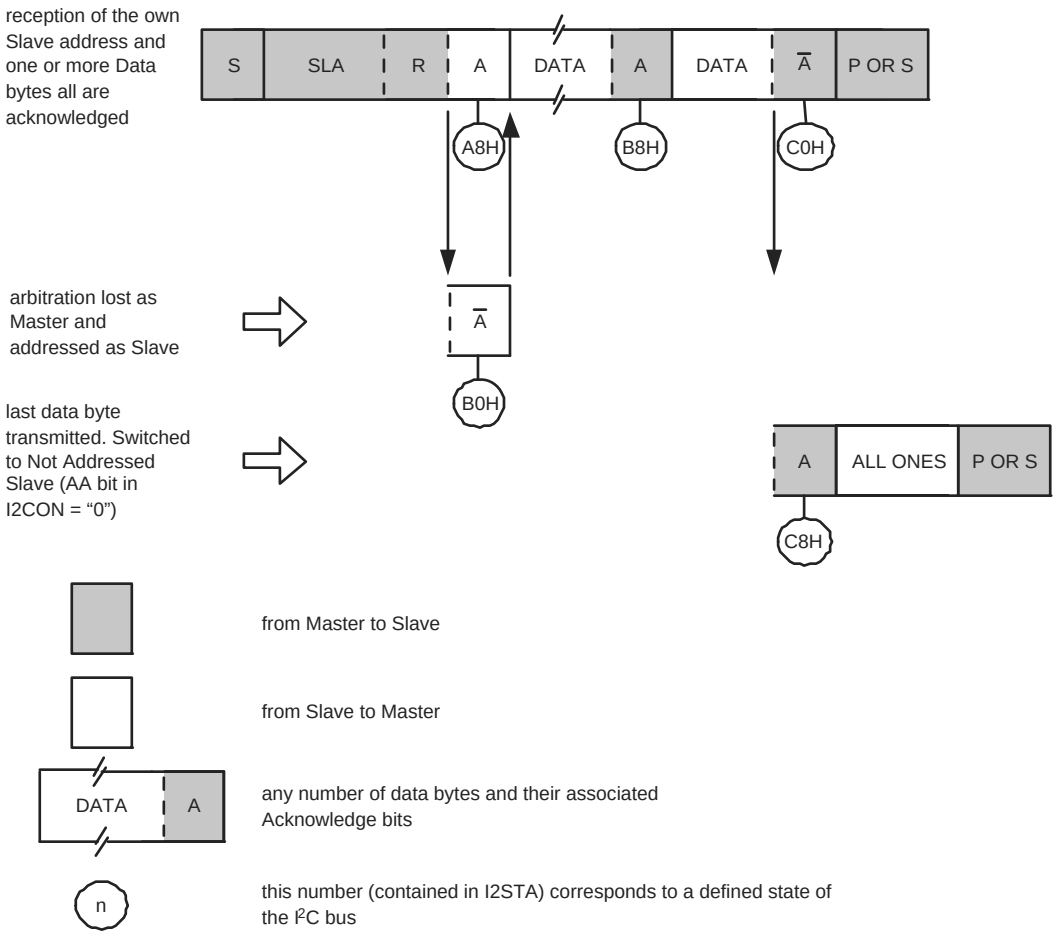


Fig 161. Format and states in the Slave Transmitter mode

42.10.5 Miscellaneous states

There are two STAT codes that do not correspond to a defined I^2C hardware state (see Table 968). These are discussed below.

42.10.5.1 STAT = 0xF8

This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when the I^2C block is not involved in a serial transfer.

42.10.5.2 STAT = 0x00

This status code indicates that a bus error has occurred during an I^2C serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal I^2C block signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This

Table 983. CAN message interface command request registers (IF2_CMDREQ, address 0x400E 2080 (C_CAN0) and 0x400A 4080 (C_CAN1)) bit description

Bit	Symbol	Description	Reset value	Access
5:0	Message Number	Message number 0x01 to 0x20 = Valid message numbers The message object in the message RAM is selected for data transfer. 0x00 = Not a valid message number. This value is interpreted as 0x20.[1] 0x21 to 0x3F = Not a valid message number. This value is interpreted as 0x01 - 0x1F.[1]	0x01	R/W
14:6	-	Reserved		
15	BUSY	BUSY flag Set to one by hardware when writing to this Command request register. Set to zero by hardware when read/write action to this Command request register has finished.	0	R
31:16	-	Reserved	-	-

[1] When a message number that is not valid is written into the Command request registers, the message number will be transformed into a valid value and that message object will be transferred.

43.6.2.3 CAN message interface command mask registers

The control bits of the IFx Command Mask Register specify the transfer direction and select which of the IFx Message Buffer Registers are source or target of the data transfer.The functions of the register bits depend on the transfer direction (read or write) which is selected in the WR/RD bit (bit 7) of this Command mask register.

Select the WR/RD to

one for the Write transfer direction (write to message RAM)

zero for the Read transfer direction (read from message RAM)

Transfer direction Write

Table 984. CAN message interface command mask registers write direction (IF1_CMDMSK_W, address 0x400E 2024 (C_CAN0) and 0x400A 4024 (C_CAN1)) bit description

Bit	Symbol	Value	Description	Reset value	Access
0	DATA_B		Access data bytes 4-7	0	R/W
		0	Data bytes 4-7 unchanged.		
		1	Transfer data bytes 4-7 to message object.		
1	DATA_A		Access data bytes 0-3	0	R/W
		0	Data bytes 0-3 unchanged.		
		1	Transfer data bytes 0-3 to message object.		

Table 984. CAN message interface command mask registers write direction (IF1_CMDMSK_W, address 0x400E 2024 (C_CAN0) and 0x400A 4024 (C_CAN1)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
2	TXRQST		Access transmission request bit	0	R/W
		0	No transmission request. TXRQSRT bit unchanged in IF1/2_MCTRL. **Remark:** If a transmission is requested by programming this bit, the TXRQST bit in the CANIFn_MCTRL register is ignored.		
		1	Request a transmission. Set the TXRQST bit IF1/2_MCTRL.		
3	CLRINTPND	-	This bit is ignored in the write direction.	0	R/W
4	CTRL		Access control bits	0	R/W
		0	Control bits unchanged.		
		1	Transfer control bits to message object		
5	ARB		Access arbitration bits	0	R/W
		0	Arbitration bits unchanged.		
		1	Transfer Identifier, DIR, XTD, and MSGVAL bits to message object.		
6	MASK		Access mask bits	0	R/W
		0	Mask bits unchanged.		
		1	Transfer Identifier MASK + MDIR + MXTD to message object.		
7	WR_RD	1	Write transfer Transfer data from the selected message buffer registers to the message object addressed by the command request register CANIFn_CMDREQ.	0	R/W
31:8	-	-	reserved	0	-

Table 985. CAN message interface command mask registers write direction (IF2_CMDMSK_W, address 0x400E 2084 (C_CAN0) and 0x400A 4084 (C_CAN1)) bit description

Bit	Symbol	Value	Description	Reset value	Access
0	DATA_B		Access data bytes 4-7	0	R/W
		0	Data bytes 4-7 unchanged.		
		1	Transfer data bytes 4-7 to message object.		
1	DATA_A		Access data bytes 0-3	0	R/W
		0	Data bytes 0-3 unchanged.		
		1	Transfer data bytes 0-3 to message object.		

Table 996. CAN message interface message control registers (IF1_MCTRL, address 0x400E 2038 (C_CAN0) and 0x400A 4038 (C_CAN1)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
13	INTPND		Interrupt pending	0	R/W
		0	This message object is not the source of an interrupt.		
		1	This message object is the source of an interrupt. The Interrupt Identifier in the Interrupt Register will point to this message object if there is no other interrupt source with higher priority.		
14	MSGLST		Message lost (only valid for message objects in the direction receive).	0	R/W
		0	No message lost since this bit was reset last by the CPU.		
		1	The Message Handler stored a new message into this object when NEWDAT was still set, the CPU has lost a message.		
15	NEWDAT		New data	0	R/W
		0	No new data has been written into the data portion of this message object by the message handler since this flag was cleared last by the CPU.		
		1	The message handler or the CPU has written new data into the data portion of this message object.		
31:16	-	-	Reserved	0	-

Table 997. CAN message interface message control registers (IF2_MCTRL, address 0x400E 2098 (C_CAN0) and 0x400A 4098 (C_CAN1)) bit description

Bit	Symbol	Value	Description	Reset value	Access
3:0	DLC3_0		Data length code **Remark:** The Data Length Code of a Message Object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the Message Handler stores a data frame, it will write the DLC to the value given by the received message. 0000 to 1000 = Data frame has 0 - 8 data bytes. 1001 to 1111 = Data frame has 8 data bytes.	0000	R/W
6:4	-		reserved	-	-
7	EOB		End of buffer	0	R/W
		0	Message object belongs to a FIFO buffer and is not the last message object of that FIFO buffer.		
		1	Single message object or last message object of a FIFO buffer.		