



Welcome to [E-XFL.COM](https://www.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	96MHz
Connectivity	EBI/EMI, I ² C, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I ² S, POR, PWM, WDT
Number of I/O	96
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	36K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x10b, 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam3u2ea-au

- do not branch to any instruction inside an IT block, except when returning from an exception handler
- all conditional instructions except *Bcond* must be inside an IT block. *Bcond* can be either outside or inside an IT block but has a larger branch range if it is inside one
- each instruction inside the IT block must specify a condition code suffix that is either the same or logical inverse as for the other instructions in the block.

Your assembler might place extra restrictions on the use of IT blocks, such as prohibiting the use of assembler directives within them.

12.16.3.4 Condition flags

This instruction does not change the flags.

12.16.3.5 Example

```
ITTE    NE                ; Next 3 instructions are conditional
ANDNE   R0, R0, R1        ; ANDNE does not update condition flags
ADDSNE  R2, R2, #1        ; ADDSNE updates condition flags
MOVEQ   R2, R3            ; Conditional move

CMP      R0, #9           ; Convert R0 hex value (0 to 15) into ASCII
                        ; ('0'-'9', 'A'-'F')
ITE      GT              ; Next 2 instructions are conditional
ADDGT   R1, R0, #55       ; Convert 0xA -> 'A'
ADDLE   R1, R0, #48       ; Convert 0x0 -> '0'

IT       GT              ; IT block with only one conditional instruction
ADDGT   R1, R1, #1        ; Increment R1 conditionally

ITTEE   EQ              ; Next 4 instructions are conditional
MOVEQ   R0, R1            ; Conditional move
ADDEQ   R2, R2, #10       ; Conditional add
ANDNE   R3, R3, #1        ; Conditional AND
BNE.W   dloop            ; Branch instruction can only be used in the last
                        ; instruction of an IT block

IT       NE              ; Next instruction is conditional
ADD     R0, R0, R1        ; Syntax error: no condition code used in IT block
```

- **VECTCLRACTIVE**

WO

Reserved for Debug use. This bit reads as 0. When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable.

- **VECTRESET**

WO

Reserved for Debug use. This bit reads as 0. When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable.

12.20.6.1 Binary point

The PRIGROUP field indicates the position of the binary point that splits the PRI_*n* fields in the Interrupt Priority Registers into separate *group priority* and *subpriority* fields. Table 12-31 shows how the PRIGROUP value controls this split.

Table 12-31. Priority grouping

PRIGROUP	Interrupt priority level value, PRI_ <i>N</i> [7:0]			Number of	
	Binary point ⁽¹⁾	Group priority bits	Subpriority bits	Group priorities	Subpriorities
b011	bxxxx.0000	[7:4]	None	16	1
b100	bxxx.y0000	[7:5]	[4]	8	2
b101	bxx.yy0000	[7:6]	[5:4]	4	4
b110	bx.yyy0000	[7]	[6:4]	2	8
b111	b.yyyy0000	None	[7:4]	1	16

1. PRI_*n*[7:0] field showing the binary point. x denotes a group priority field bit, and y denotes a subpriority field bit.

Determining preemption of an exception uses only the group priority field, see “Interrupt priority grouping” on page 71.

Unaligned

A data item stored at an address that is not divisible by the number of bytes that defines the data size is said to be unaligned. For example, a word stored at an address that is not divisible by four.

Undefined

Indicates an instruction that generates an Undefined instruction exception.

Unpredictable (UNP)

You cannot rely on the behavior. Unpredictable behavior must not represent security holes. Unpredictable behavior must not halt or hang the processor, or any parts of the system.

Warm reset

Also known as a core reset. Initializes the majority of the processor excluding the debug controller and debug logic. This type of reset is useful if you are using the debugging features of a processor.

Word

A 32-bit data item.

Write

Writes are defined as operations that have the semantics of a store. Writes include the Thumb instructions STM, STR, STRH, STRB, and PUSH.

17.5.2 RTC Mode Register

Name: RTC_MR

Address: 0x400E1264

Access: Read-write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	HRMOD

- **HRMOD: 12-/24-hour Mode**

0 = 24-hour mode is selected.

1 = 12-hour mode is selected.

All non-significant bits read zero.

18.5.8 Supply Controller Status Register

Name: SUPC_SR

Address: 0x400E1224

Access: Read-write

31	30	29	28	27	26	25	24
WKUPIS15	WKUPIS14	WKUPIS13	WKUPIS12	WKUPIS11	WKUPIS10	WKUPIS9	WKUPIS8
23	22	21	20	19	18	17	16
WKUPIS7	WKUPIS6	WKUPIS5	WKUPIS4	WKUPIS3	WKUPIS2	WKUPIS1	WKUPIS0
15	14	13	12	11	10	9	8
–	–	–	FWUPIS	–	–	–	–
7	6	5	4	3	2	1	0
OSCSEL	SMOS	SMS	SMRSTS	BODRSTS	SMWS	WKUPS	FWUPS

Note: Because of the asynchronism between the Slow Clock (SCLK) and the System Clock (MCK), the status register flag reset is taken into account only 2 slow clock cycles after the read of the SUPC_SR.

- **FWUPS: FWUP Wake Up Status**

0 (NO) = no wake up due to the assertion of the FWUP pin has occurred since the last read of SUPC_SR.

1 (PRESENT) = at least one wake up due to the assertion of the FWUP pin has occurred since the last read of SUPC_SR.

- **WKUPS: WKUP Wake Up Status**

0 (NO) = no wake up due to the assertion of the WKUP pins has occurred since the last read of SUPC_SR.

1 (PRESENT) = at least one wake up due to the assertion of the WKUP pins has occurred since the last read of SUPC_SR.

- **SMWS: Supply Monitor Detection Wake Up Status**

0 (NO) = no wake up due to a supply monitor detection has occurred since the last read of SUPC_SR.

1 (PRESENT) = at least one wake up due to a supply monitor detection has occurred since the last read of SUPC_SR.

- **BODRSTS: Brownout Detector Reset Status**

0 (NO) = no core brownout rising edge event has been detected since the last read of the SUPC_SR.

1 (PRESENT) = at least one brownout output rising edge event has been detected since the last read of the SUPC_SR.

When the voltage remains below the defined threshold, there is no rising edge event at the output of the brownout detection cell. The rising edge event occurs only when there is a voltage transition below the threshold.

- **SMRSTS: Supply Monitor Reset Status**

0 (NO) = no supply monitor detection has generated a core reset since the last read of the SUPC_SR.

1 (PRESENT) = at least one supply monitor detection has generated a core reset since the last read of the SUPC_SR.

- **SMS: Supply Monitor Status**

0 (NO) = no supply monitor detection since the last read of SUPC_SR.

1 (PRESENT) = at least one supply monitor detection since the last read of SUPC_SR.

20.4 Functional Description

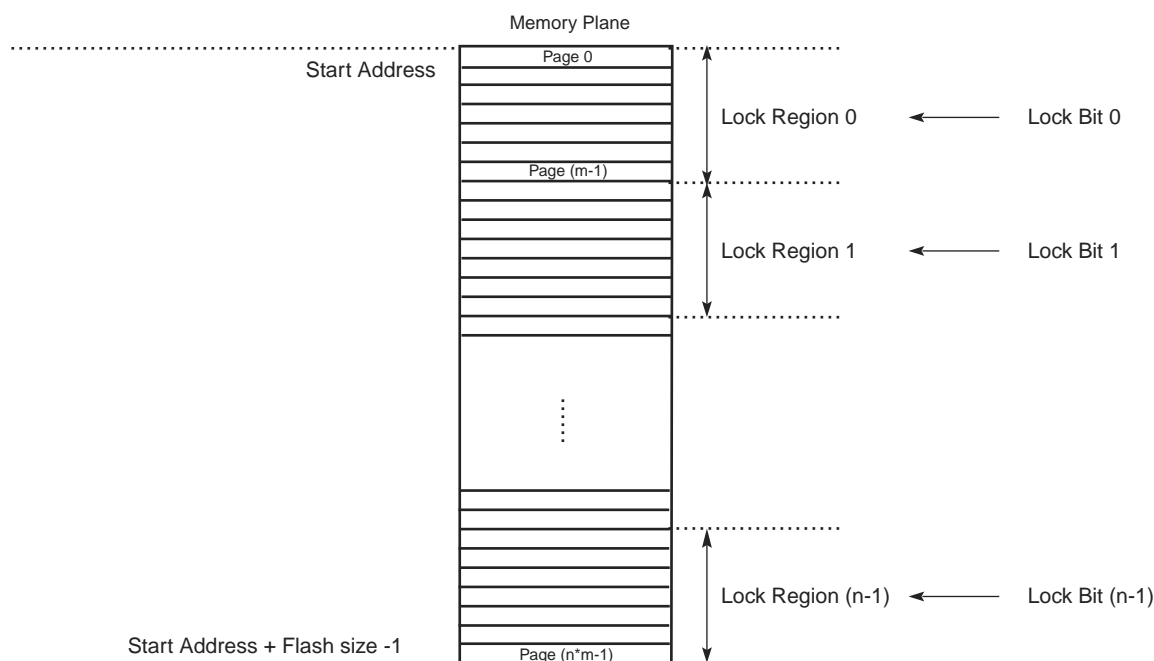
20.4.1 Embedded Flash Organization

The embedded Flash interfaces directly with the 32-bit internal bus. The embedded Flash is composed of:

- One memory plane organized in several pages of the same size.
- Two 128-bit or 64-bit read buffers used for code read optimization.
- One 128-bit or 64-bit read buffer used for data read optimization.
- One write buffer that manages page programming. The write buffer size is equal to the page size. This buffer is write-only and accessible all along the 1 MByte address space, so that each word can be written to its final address.
- Several lock bits used to protect write/erase operation on several pages (lock region). A lock bit is associated with a lock region composed of several pages in the memory plane.
- Several bits that may be set and cleared through the Enhanced Embedded Flash Controller (EEFC) interface, called General Purpose Non Volatile Memory bits (GPNVM bits).

The embedded Flash size, the page size, the lock regions organization and GPNVM bits definition are described in the product definition section. The Enhanced Embedded Flash Controller (EEFC) returns a descriptor of the Flash controlled after a get descriptor command issued by the application (see “Getting Embedded Flash Descriptor” on page 292).

Figure 20-1. Embedded Flash Organization



20.5.2 EEFC Flash Command Register

Name: EEFC_FCR

Address: 0x400E0804 (0), 0x400E0A04 (1)

Access: Write-only

Offset: 0x04

31	30	29	28	27	26	25	24
FKEY							
23	22	21	20	19	18	17	16
FARG							
15	14	13	12	11	10	9	8
FARG							
7	6	5	4	3	2	1	0
FCMD							

- **FCMD: Flash Command**

This field defines the Flash commands. Refer to “Flash Commands” on page 291.

- **FARG: Flash Command Argument**

Erase all command	Field is meaningless.
Programming command	FARG defines the page number to be programmed.
Lock command	FARG defines the page number to be locked.

- **FKEY: Flash Writing Protection Key**

This field should be written with the value 0x5A to enable the command defined by the bits of the register. If the field is written with a different value, the write is not performed and no action is started.

24.14.4 NWAIT Latency and Read/Write Timings

There may be a latency between the assertion of the read/write controlling signal and the assertion of the NWAIT signal by the device. The programmed pulse length of the read/write controlling signal must be at least equal to this latency plus the 2 cycles of resynchronization + 1 cycle. Otherwise, the SMC may enter the hold state of the access without detecting the NWAIT signal assertion. This is true in frozen mode as well as in ready mode. This is illustrated on Figure 24-27.

When EXNW_MODE is enabled (ready or frozen), the user must program a pulse length of the read and write controlling signal of at least:

minimal pulse length = NWAIT latency + 2 resynchronization cycles + 1 cycle

Figure 24-27. NWAIT Latency

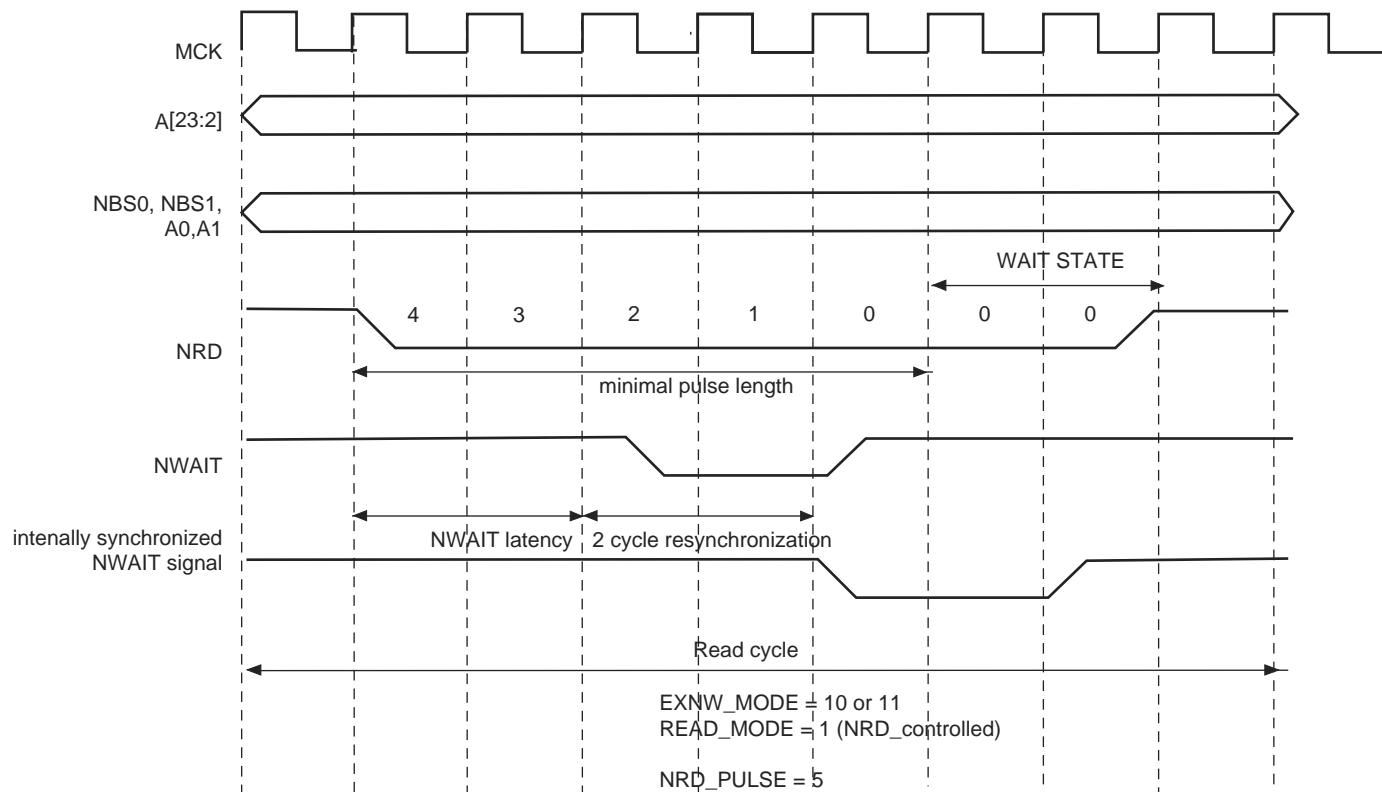
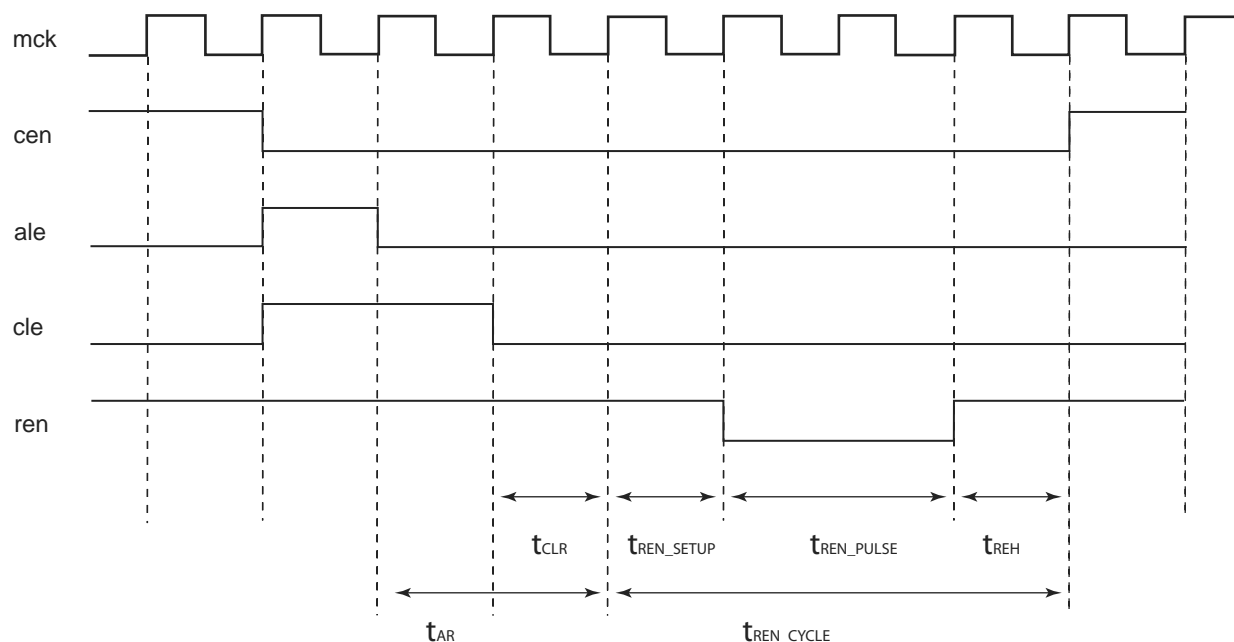


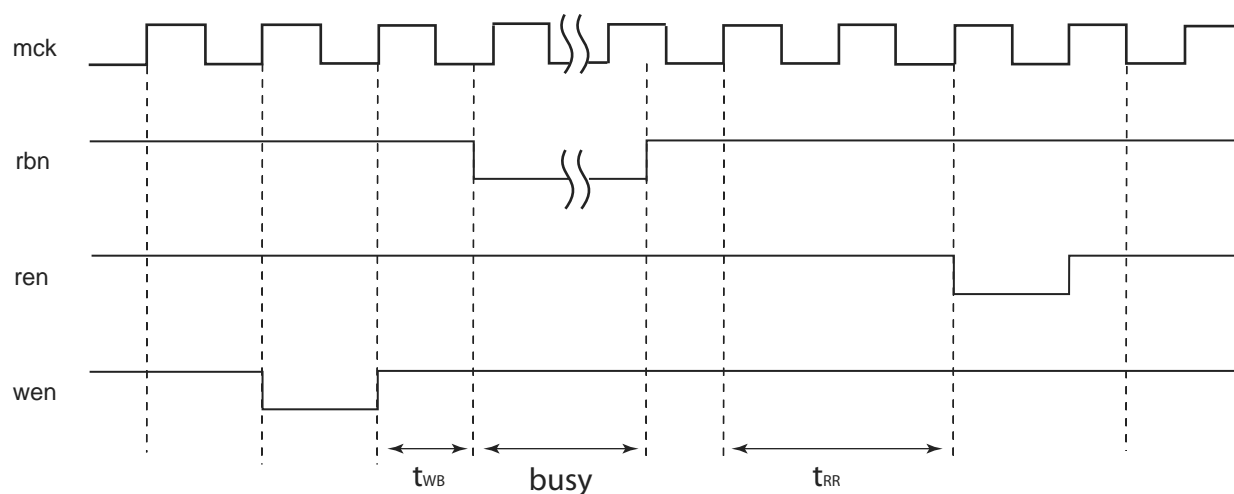
Figure 24-34. Read Enable Timing Configuration Working with NAND Flash Device



- Ready/Busy Signal Timing configuration working with a NAND Flash device

Use TWB field in SMC_TIMINGS register to configure the maximum elapsed time between the rising edge of wen signal and the falling edge of rbn signal. Use TRR field in the SMC_TIMINGS register to program the number of clock cycle between the rising edge of the rbn signal and the falling edge of ren signal.

Figure 24-35. Ready/Busy Timing Configuration



24.18.11 SMC ECC Status Register 1

Name: SMC_ECC_SR1

Address: 0x400E0028

Access: Read-only

Reset: 0x00000000

31	30	29	28	27	26	25	24
–	ECCERR7	ECCERR7	RECERR7	–	ECCERR6	ECCERR6	RECERR6
23	22	21	20	19	18	17	16
–	ECCERR5	ECCERR5	RECERR5	–	ECCERR4	ECCERR4	RECERR4
15	14	13	12	11	10	9	8
–	MULERR3	ECCERR3	RECERR3	–	MULERR2	ECCERR2	RECERR2
7	6	5	4	3	2	1	0
–	MULERR1	ECCERR1	RECERR1	–	ECCERR0	ECCERR0	RECERR0

- **RECERR0: Recoverable Error**

0: No Errors Detected.

1: Errors Detected. If MUL_ERROR is 0, a single correctable error was detected. Otherwise multiple uncorrected errors were detected.

- **ECCERR0: ECC Error**

0: No Errors Detected.

1: A single bit error occurred in the ECC bytes.

If TYPECORRECT = 0, read both ECC Parity 0 and ECC Parity 1 registers, the error occurred at the location which contains a 1 in the least significant 16 bits; else read ECC Parity 0 register, the error occurred at the location which contains a 1 in the least significant 24 bits.

- **MULERR0: Multiple Error**

0: No Multiple Errors Detected.

1: Multiple Errors Detected.

- **RECERR1: Recoverable Error in the page between the 256th and the 511th bytes or the 512nd and the 1023rd bytes**

Fixed to 0 if TYPECORREC = 0.

0: No Errors Detected.

1: Errors Detected. If MUL_ERROR is 0, a single correctable error was detected. Otherwise multiple uncorrected errors were detected.

- **ECCERR1: ECC Error in the page between the 256th and the 511th bytes or between the 512nd and the 1023rd bytes**

Fixed to 0 if TYPECORREC = 0

0: No Errors Detected.

1: A single bit error occurred in the ECC bytes.

Read ECC Parity 1 register, the error occurred at the location which contains a 1 in the least significant 24 bits.

29.7.12 PIO Controller Output Data Status Register

Name: PIO_ODSR

Address: 0x400E0C38 (PIOA), 0x400E0E38 (PIOB), 0x400E1038 (PIOC)

Access: Read-only or Read/Write

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0-P31: Output Data Status**

0 = The data to be driven on the I/O line is 0.

1 = The data to be driven on the I/O line is 1.

32.8.5 Master Receiver Mode

The read sequence begins by setting the START bit. After the start condition has been sent, the master sends a 7-bit slave address to notify the slave device. The bit following the slave address indicates the transfer direction, 1 in this case (MREAD = 1 in TWI_MMR). During the acknowledge clock pulse (9th pulse), the master releases the data line (HIGH), enabling the slave to pull it down in order to generate the acknowledge. The master polls the data line during this clock pulse and sets the **NACK** bit in the status register if the slave does not acknowledge the byte.

If an acknowledge is received, the master is then ready to receive data from the slave. After data has been received, the master sends an acknowledge condition to notify the slave that the data has been received except for the last data, after the stop condition. See Figure 32-9. When the RXRDY bit is set in the status register, a character has been received in the receive-holding register (TWI_RHR). The RXRDY bit is reset when reading the TWI_RHR.

When a single data byte read is performed, with or without internal address (**IADR**), the START and STOP bits must be set at the same time. See Figure 32-9. When a multiple data byte read is performed, with or without internal address (**IADR**), the STOP bit must be set after the next-to-last data received. See Figure 32-10. For Internal Address usage see Section 32.8.6.

Figure 32-9. Master Read with One Data Byte

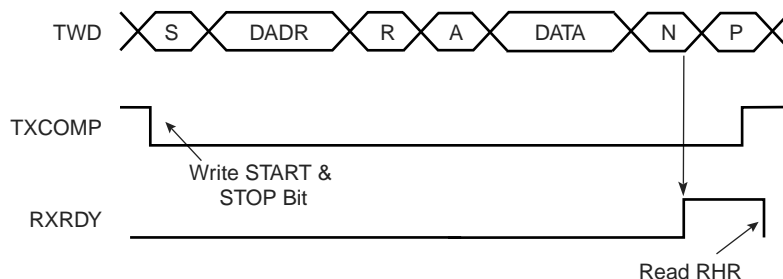
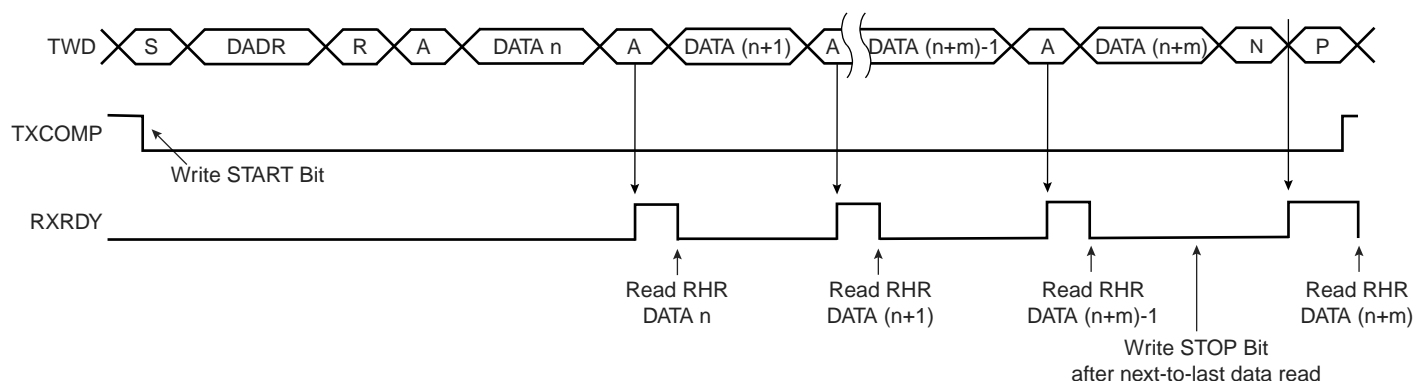


Figure 32-10. Master Read with Multiple Data Bytes



RXRDY is used as Receive Ready for the PDC receive channel.

32.8.6 Internal Address

The TWI interface can perform various transfer formats: Transfers with 7-bit slave address devices and 10-bit slave address devices.

32.11.10 TWI Receive Holding Register

Name: TWI_RHR
Address: 0x40084030 (0), 0x40088030 (1)
Access: Read-only
Reset: 0x00000000

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
RXDATA							

- **RXDATA: Master or Slave Receive Holding Data**

33. Universal Asynchronous Receiver Transceiver (UART)

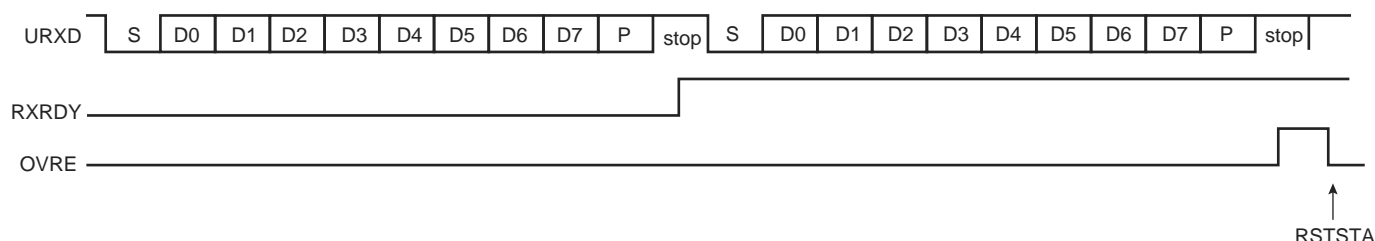
33.1 Description

The Universal Asynchronous Receiver Transmitter features a two-pin UART that can be used for communication and trace purposes and offers an ideal medium for in-situ programming solutions. Moreover, the association with two peripheral DMA controller (PDC) channels permits packet handling for these tasks with processor time reduced to a minimum.

33.2 Embedded Characteristics

- Two-pin UART
 - Implemented Features are USART Compatible
 - Independent Receiver and Transmitter with a Common Programmable Baud Rate Generator
 - Even, Odd, Mark or Space Parity Generation
 - Parity, Framing and Overrun Error Detection
 - Automatic Echo, Local Loopback and Remote Loopback Channel Modes
 - Interrupt Generation
 - Support for Two PDC Channels with Connection to Receiver and Transmitter

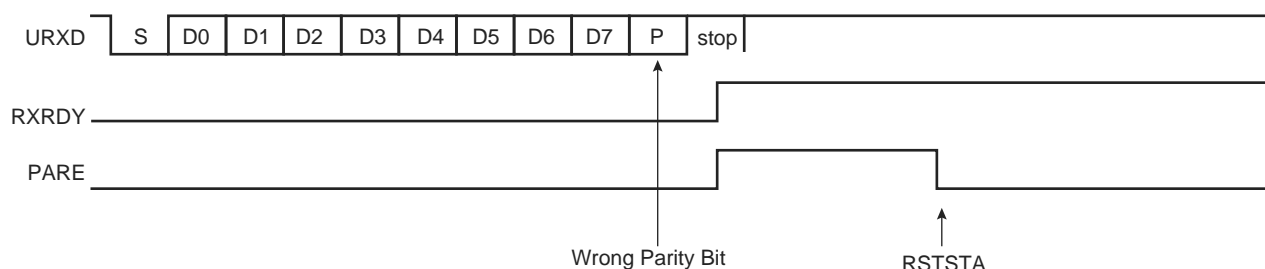
Figure 33-6. Receiver Overrun



33.5.2.5 Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with the field PAR in UART_MR. It then compares the result with the received parity bit. If different, the parity error bit PARE in UART_SR is set at the same time the RXRDY is set. The parity bit is cleared when the control register UART_CR is written with the bit RSTSTA (Reset Status) at 1. If a new character is received before the reset status command is written, the PARE bit remains at 1.

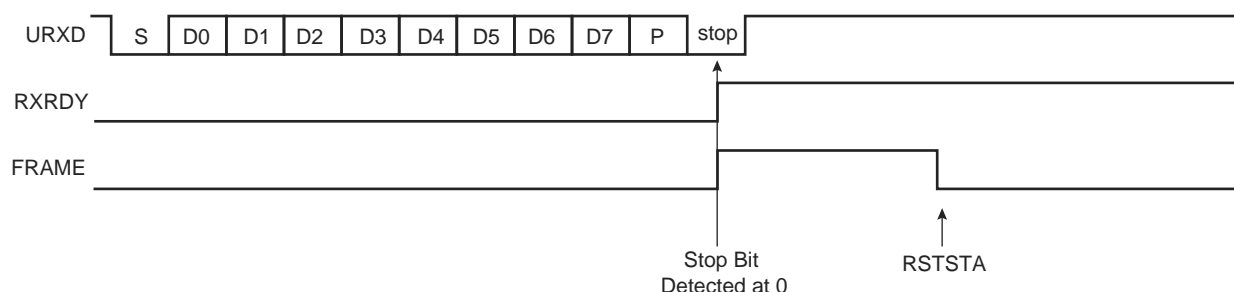
Figure 33-7. Parity Error



33.5.2.6 Receiver Framing Error

When a start bit is detected, it generates a character reception when all the data bits have been sampled. The stop bit is also sampled and when it is detected at 0, the FRAME (Framing Error) bit in UART_SR is set at the same time the RXRDY bit is set. The FRAME bit remains high until the control register UART_CR is written with the bit RSTSTA at 1.

Figure 33-8. Receiver Framing Error



33.5.3 Transmitter

33.5.3.1 Transmitter Reset, Enable and Disable

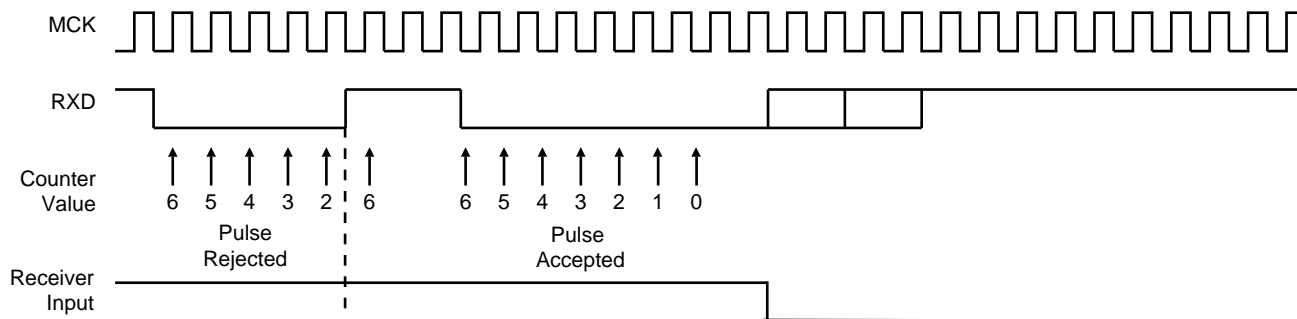
After device reset, the UART transmitter is disabled and it must be enabled before being used. The transmitter is enabled by writing the control register UART_CR with the bit TXEN at 1. From this command, the transmitter waits for a character to be written in the Transmit Holding Register (UART_THR) before actually starting the transmission.

34.7.5.3 IrDA Demodulator

The demodulator is based on the IrDA Receive filter comprised of an 8-bit down counter which is loaded with the value programmed in US_IF. When a falling edge is detected on the RXD pin, the Filter Counter starts counting down at the Master Clock (MCK) speed. If a rising edge is detected on the RXD pin, the counter stops and is reloaded with US_IF. If no rising edge is detected when the counter reaches 0, the input of the receiver is driven low during one bit time.

Figure 34-35 illustrates the operations of the IrDA demodulator.

Figure 34-35. IrDA Demodulator Operations

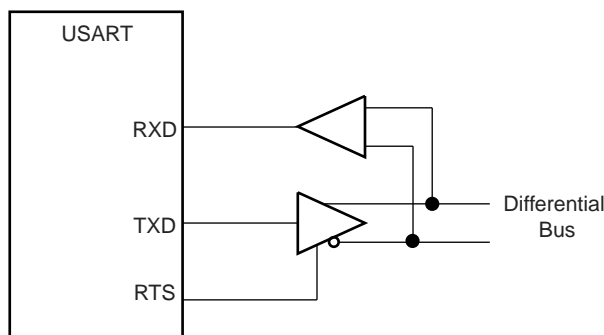


As the IrDA mode uses the same logic as the ISO7816, note that the FI_DI_RATIO field in US_FIDI must be set to a value higher than 0 in order to assure IrDA communications operate correctly.

34.7.6 RS485 Mode

The USART features the RS485 mode to enable line driver control. While operating in RS485 mode, the USART behaves as though in asynchronous or synchronous mode and configuration of all the parameters is possible. The difference is that the RTS pin is driven high when the transmitter is operating. The behavior of the RTS pin is controlled by the TXEMPTY bit. A typical connection of the USART to a RS485 bus is shown in Figure 34-36.

Figure 34-36. Typical Connection to a RS485 Bus



The USART is set in RS485 mode by programming the USART_MODE field in the Mode Register (US_MR) to the value 0x1.

The RTS pin is at a level inverse to the TXEMPTY bit. Significantly, the RTS pin remains high when a timeguard is programmed so that the line can remain driven after the last character completion. Figure 34-37 gives an example of the RTS waveform during a character transmission when the timeguard is enabled.

36.14.2 HSMCI Mode Register

Name: HSMCI_MR
Address: 0x40000004
Access: Read-write

31	30	29	28	27	26	25	24
BLKLEN							
23	22	21	20	19	18	17	16
BLKLEN							
15	14	13	12	11	10	9	8
–	PADV	FBYTE	WRPROOF	RDPROOF	PWSDIV		
7	6	5	4	3	2	1	0
CLKDIV							

This register can only be written if the WPEN bit is cleared in “HSMCI Write Protect Mode Register” on page 860.

- **CLKDIV: Clock Divider**

High Speed MultiMedia Card Interface clock (MCCK or HSMCI_CK) is Master Clock (MCK) divided by $(2 * (CLKDIV + 1))$.

- **PWSDIV: Power Saving Divider**

High Speed MultiMedia Card Interface clock is divided by $2^{(PWSDIV)} + 1$ when entering Power Saving Mode.

Warning: This value must be different from 0 before enabling the Power Save Mode in the HSMCI_CR (HSMCI_PWSEN bit).

- **RDPROOF Read Proof Enable**

Enabling Read Proof allows to stop the HSMCI Clock during read access if the internal FIFO is full. This will guarantee data integrity, not bandwidth.

0 = Disables Read Proof.

1 = Enables Read Proof.

- **WRPROOF Write Proof Enable**

Enabling Write Proof allows to stop the HSMCI Clock during write access if the internal FIFO is full. This will guarantee data integrity, not bandwidth.

0 = Disables Write Proof.

1 = Enables Write Proof.

- **FBYTE: Force Byte Transfer**

Enabling Force Byte Transfer allow byte transfers, so that transfer of blocks with a size different from modulo 4 can be supported.

Warning: BLKLEN value depends on FBYTE.

0 = Disables Force Byte Transfer.

1 = Enables Force Byte Transfer.

37.6.2.6 Synchronous Channels

Some channels can be linked together as synchronous channels. They have the same source clock, the same period, the same alignment and are started together. In this way, their counters are synchronized together.

The synchronous channels are defined by the SYNCx bits in the “PWM Sync Channels Mode Register” (PWM_SCM). Only one group of synchronous channels is allowed.

When a channel is defined as a synchronous channel, the channel 0 is automatically defined as a synchronous channel too, because the channel 0 counter configuration is used by all the synchronous channels.

If a channel x is defined as a synchronous channel, it uses the following configuration fields of the channel 0 instead of its own:

- CPRE0 field in PWM_CMRO register instead of CPREx field in PWM_CMRx register (same source clock)
- CPRD0 field in PWM_CMRO register instead of CPRDx field in PWM_CMRx register (same period)
- CALG0 field in PWM_CMRO register instead of CALGx field in PWM_CMRx register (same alignment)

Thus writing these fields of a synchronous channel has no effect on the output waveform of this channel (except channel 0 of course).

Because counters of synchronous channels must start at the same time, they are all enabled together by enabling the channel 0 (by the CHID0 bit in PWM_ENA register). In the same way, they are all disabled together by disabling channel 0 (by the CHID0 bit in PWM_DIS register). However, a synchronous channel x different from channel 0 can be enabled or disabled independently from others (by the CHIDx bit in PWM_ENA and PWM_DIS registers).

Defining a channel as a synchronous channel while it is an asynchronous channel (by writing the bit SYNCx to 1 while it was at 0) is allowed only if the channel is disabled at this time (CHIDx = 0 in PWM_SR register). In the same way, defining a channel as an asynchronous channel while it is a synchronous channel (by writing the SYNCx bit to 0 while it was 1) is allowed only if the channel is disabled at this time.

The field UPDM (Update Mode) in the PWM_SCM register allow to select one of the three methods to update the registers of the synchronous channels:

- Method 1 (UPDM = 0): the period value, the duty-cycle values and the dead-time values must be written by the CPU in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx). The update is triggered at the next PWM period as soon as the bit UPDULOCK in the “PWM Sync Channels Update Control Register” (PWM_SCUC) is set to 1 (see “Method 1: Manual write of duty-cycle values and manual trigger of the update” on page 879).
- Method 2 (UPDM = 1): the period value, the duty-cycle values, the dead-time values and the update period value must be written by the CPU in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPD). The update of the period value and of the dead-time values is triggered at the next PWM period as soon as the bit UPDULOCK in the “PWM Sync Channels Update Control Register” (PWM_SCUC) is set to 1. The update of the duty-cycle values and the update period value is triggered automatically after an update period defined by the field UPR in the “PWM Sync Channels Update Period Register” (PWM_SCUP) (see “Method 2: Manual write of duty-cycle values and automatic trigger of the update” on page 880).
- Method 3 (UPDM = 2): same as Method 2 apart from the fact that the duty-cycle values of ALL synchronous channels are written by the Peripheral DMA Controller (PDC) (see “Method 3: Automatic write of duty-cycle values and automatic trigger of the update” on page 882). The user can choose to synchronize the PDC transfer request with a comparison match (see Section 37.6.3 “PWM Comparison Units”), by the fields PTRM and PTRCS in the PWM_SCM register.

37.7.14 PWM Interrupt Disable Register 2

Name: PWM_IDR2

Address: 0x4008C038

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
7	6	5	4	3	2	1	0
–	–	–	–	UNRE	TXBUFE	ENDTX	WRDY

- **WRDY:** Write Ready for Synchronous Channels Update Interrupt Disable
- **ENDTX:** PDC End of TX Buffer Interrupt Disable
- **TXBUFE:** PDC TX Buffer Empty Interrupt Disable
- **UNRE:** Synchronous Channels Update Underrun Error Interrupt Disable
- **CMPMx:** Comparison x Match Interrupt Disable
- **CMPUx:** Comparison x Update Interrupt Disable

37.7.37 PWM Channel Duty Cycle Update Register

Name: PWM_CDTYUPDx [x=0..3]

Address: 0x4008C208 [0], 0x4008C228 [1], 0x4008C248 [2], 0x4008C268 [3]

Access: Write-only.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CDTYUPD							
15	14	13	12	11	10	9	8
CDTYUPD							
7	6	5	4	3	2	1	0
CDTYUPD							

This register acts as a double buffer for the CDTY value. This prevents an unexpected waveform when modifying the waveform duty-cycle.

Only the first 16 bits (channel counter size) are significant.

- **CDTYUPD: Channel Duty-Cycle Update**

Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRx).